Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5320--10-9231

Amplitude Tapers for Planar Arrays Using the McClellan Transformation: Concepts and Preliminary Design Experiments

JEFFREY O. COLEMAN

Advanced Radar Systems Branch
Radar Division

April 29, 2010

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
29-04-2010	Memorandum Report	

4. TITLE AND SUBTITLE

Amplitude Tapers for Planar Arrays Using the McClellan Transformation: Concepts and Preliminary Design Experiments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61153N

6. AUTHOR(S)

Jeffrey O. Coleman

5d. PROJECT NUMBER

5e. TASK NUMBER
EW021-05-43

5f. WORK UNIT NUMBER
N0001409WX30010

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5320--10-9231

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The McClellan transformation has been widely studied in image processing since the 1970s, but it is not generally known in the phased-array community. In the array context explored here, the McClellan transformation uses a very small planar array taper — in this report examples ranged from 7 to 31 elements in size — as a "spreading function" to take the weights of a prototype line-array taper or 1D FIR filter of modest size and spread those weights out spatially to create a large planar array taper of hundreds or thousands of elements. Reasonable 2D tapers can be obtained in this way using common tools for 1D filter design and spreading functions either chosen by hand or designed using simple 2D design techniques. Examples in this report explore the design of 2D tapers of several thousand elements on the triangular grid.

The key advantage of the approach is that certain simple changes to the array pattern — modestly broadening the beam, making it elliptical, rotating that ellipse — can often be effected through simple modifications of the spreading function, with the 1D prototype filter left unchanged. Subsequent reapplication of the McClellan transformation is simple enough that such spreading-function changes allow a degree of on-the-fly beam tailoring. The key disadvantage of the approach is that approaching optimal levels of gain or taper loss appears quite difficult. Example designs here all suffered at least a 0.7 dB gain penalty relative to tapers obtained by direct optimization of the whole 2D taper to otherwise similar specifications.

15. SUBJECT TERMS
Array radar
Array taper
McClellan transformation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Jeffrey O. Coleman
Unclassified	Unclassified	Unclassified	UL	35	**19b. TELEPHONE NUMBER** *(include area code)* (202) 404-8843

Contents

List of Figures

(Short titles here differ from longer actual captions.)

List of Tables

(Short titles here differ from longer actual captions.)

1 Introduction

Radar systems engineers often obtain a taper for a planar array as the separable product of one-dimensional tapers in two orthogonal directions. While this is rarely appropriate for final design work, it is often adequate for various preliminary designs, for example for simulations or other calculations where obtaining rough array-performance numbers will settle some matter at hand. Of course the analogous separable-filter idea is well known in image processing.

In the image-processing community, however, a second design procedure, the McClellan transformation, for creating a two-dimensional filter from a one-dimensional prototype filter is both well known and, by some authors' claims, widely used. The transformation from one dimension to two is mediated by a transformation kernel—a "spreading function" in this report—whose characteristics affect the final two-dimensional design as much as does the one-dimensional prototype. The extra degrees of design freedom represented by the choice of transformation kernel represent flexibility not available in the simpler separable-filter approach, so it is natural to wonder whether the McClellan transformation might be useful in the design of planar array tapers, either as an alternative to separable tapers for quick-and-dirty designs or perhaps, if adequate performance turns out to be available, in final designs as well. Remarkably, a search of the literature for work on using the McClellan transformation to design array tapers turns up nothing but a short letter by McNamara and Botha [1] pointing out that the approach can be used to design wide-coverage beams for satellite-communication arrays.

The purpose of this report then is to present the McClellan transformation somewhat tutorially, to examine it in an array context, and to explore the associated array-taper design space on a preliminary basis. It must be cautioned though that no attempt is made here to systematically explore that design space, to do careful trade studies, or to come up with precise numbers to quantify the array-design capabilities of the approach. Though appropriate performance numbers are provided for all design examples presented, the intent here is really qualitative exploration of enough parts of the space to perhaps suggest to future researchers what systematic explorations might ultimately be appropriate, and this approach yields design experiments that stop well short of final answers. This is intentional, as detailed quantitative trade studies are better performed in specific application contexts.

No formal distinction is made in this report between transmit and receive applications, but most of the design examples presented are far more natural in a receive context. Those few examples explicitly addressing beam broadening would of course be of more interest in a transmit context, though only the still unusual type in which amplitude tapers are permitted. Phase-only transmit tapers are of no interest here, because a phase-only weight characteristic is not preserved by the McClellan transformation: a phase-only prototype transforms to a two-dimensional taper that lacks the phase-only property.

A straightforward *IEEE Xplore* subject search turns up a rich engineering research literature on the McClellan transformation, only a few papers from which will be discussed explicitly here. The variant of the transformation explored here was first presented by McClellan [2] in a paper that remains a solid starting point, even though its detailed design examples are limited to the square sampling lattice in two dimensions. A hexagonal-sampling environment, such as used in most serious planar-array work, is dealt with briefly in the presentation of the McClellan transformation in Dudgeon and Mersereau's classic text [3] on multidimensional signal processing, but the notational approach to the hexagonal lattice used is clumsy and not easy to follow. The matrix-based approach of this report is easier to work with.

One very early but now-classic treatment of the McClellan transformation is the two-part paper by Mersereau and Mecklenbräuker (and Quatieri, in the case of the first part) [4, 5]. Part I focused on

Manuscript approved November 23, 2009.

the design of the spreading-function transformation kernels, but the year was 1976, when minimizing computational complexity was important in a way that it no longer is, and the paper's resulting consideration of only the very simplest examples makes it of minimal actual relevance today. Part II worked through the details of computational efficiency in realizing the transformed filters, but the filter structures considered possessed unfortunate coefficient-sensitivity properties and so became largely irrelevant in the following year when McClellan followed up on his original work by publishing the far superior Chebyshev structure on which this report is based [2]. Part II also concluded a bit too much from the limited set of design examples that had been explored by that early date, for example asserting that "the technique is only truly valuable for designing linear phase filters with quadrilateral symmetry." This report's final design example, in Section 3.3.5, certainly shows that to be incorrect.

The McClellan transformation is traditionally considered an approach to 2D filter design, but in fact it is just as suitable for 3D design. Antialiasing filtering for staggered sampling of planar arrays [6], for example, requires cone filters in 3D, and the McClellan transformation has recently been examined for 3D cone filters [7].

In any case, the reader is encouraged to consider the present report to be suitable background reading prior to tackling the literature, rather than the other way around, if only to start out with a single notational approach rather than many.

The next section is an introduction to the key concepts in the context of a simpler 1D-to-1D transformation. Then Section 3, the bulk of the report, addresses the design of two-dimensional array tapers. The Section 4 conclusion briefly synopsizes the main lessons learned from this effort. Finally, readers wondering whether the McClellan transformation can be experimented with easily should have a quick look at the Fig. 6 matlab code on page 14. The answer is yes, but only once the ideas and notation are understood.

2 The idea and its application to 1D filters

2.1 The Chebyshev forms of a 1D linear-phase FIR filter

A 1D zero-phase filter, which is just a 1D linear-phase filter with impulse response centered at the origin, has a transfer function of the form

$$H(z) = h_0 + \sum_{k=1}^{M} h_k (z^{-k} + z^k)$$

or, equivalently, an impulse response that can be written

$$h(n) = h_0\,\delta(n) + \sum_{k=1}^{M} h_k \big(\delta(n+k) + \delta(n-k)\big).$$

Let us rewrite this with the dependence on time n suppressed for brevity. We can use δ^k rather than the usual $\delta(n-k)$ to denote a unit-sample function at time k.

$$h = h_0\,\delta^0 + \sum_{k=1}^{M} 2h_k \frac{\delta^{-k} + \delta^k}{2}. \tag{1}$$

Here δ^0 and each $\frac{1}{2}(\delta^{-k} + \delta^k)$ are building blocks from which the filter impulse response is created using simple weight-and-sum linear combining. We will be using a variety of such building blocks

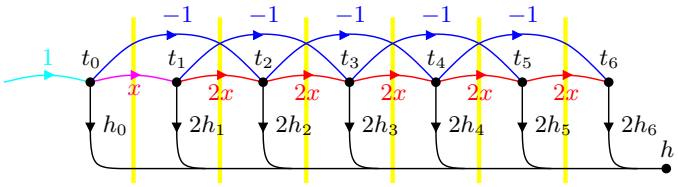


Fig. 1 Chebyshev direct form of a linear-phase FIR filter, in zero-phase form. Causality shimming will be needed at each heavy vertical line.

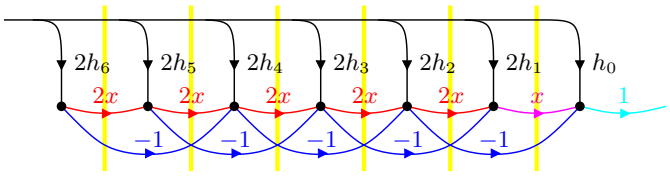


Fig. 2 Reversed Chebyshev form of a linear-phase FIR filter, in zero-phase form. Causality shimming will be needed at each heavy vertical line.

in the discussion to follow, and it will prove convenient to have them standardized to have unity DC gain. This is the reason for introducing the factors of two.

We can synthesize most of these DC-normalized impulse-pair building blocks needed from closer-spaced ones. We first convolve with $\delta^{-1}+\delta^1$ to spread a building block out in time, and we then subtract unwanted terms.

$$\begin{aligned}\frac{\delta^{-2}+\delta^2}{2} &= (\delta^{-1}+\delta^1) * \frac{\delta^{-1}+\delta^1}{2} - \delta^0 \\ \frac{\delta^{-k}+\delta^k}{2} &= (\delta^{-1}+\delta^1) * \frac{\delta^{-(k-1)}+\delta^{k-1}}{2} - \frac{\delta^{-(k-2)}+\delta^{k-2}}{2},\end{aligned} \tag{2}$$

the latter for any $k > 2$ or in fact for any $k > 1$ because the first equation here is just the $k = 2$ special case of the second.

We can clean this up with some definitions. We give the spreading function $\delta^{-1}+\delta^1$ and the DC-normalized building blocks $\frac{1}{2}(\delta^{-k}+\delta^k)$ the names $2x(n)$ and $t_k(n)$ respectively, with notational dependence on time n restored in the names. The name t_k is not meant to suggest time but respects a different naming convention. The factor of 2 in $2x(n)$ gives unity DC gain to the filter that has $x(n)$ as its impulse response, and it also puts the frequency-domain version of spreading equation (2) into a standard form. In the time and frequency domains these definitions and (2) become

$$x(n) \triangleq \frac{\delta(n+1)+\delta(n-1)}{2} \quad \leftrightarrow \quad X(f) = \frac{\mathrm{e}^{\mathrm{j}2\pi f}+\mathrm{e}^{-\mathrm{j}2\pi f}}{2} = \cos(2\pi f) \tag{3}$$

$$\begin{aligned} t_0(n) &\triangleq \delta(n) && \leftrightarrow \quad T_0(X(f)) = 1 \\ t_1(n) &\triangleq x(n) && \leftrightarrow \quad T_1(X(f)) = X(f) \\ t_k(n) &\triangleq 2x(n) * t_{k-1}(n) - t_{k-2}(n) && \leftrightarrow \quad T_k(X(f)) = 2X(f)T_{k-1}(X(f)) - T_{k-2}(X(f)), \end{aligned} \tag{4}$$

with the last line applying for $k > 1$. On the right $T_k(f)$ has been written $T_k(X(f))$ to emphasize that in fact it depends on f only through the value attained by $X(f)$. Further, written in this way each T_k is in fact a polynomial in its argument. When the f dependence of $X(f)$ is suppressed, the first few polynomials are

$$\begin{aligned} T_0 &= 1 \\ T_1 &= X \\ T_2 &= 2X^2 - 1 \\ T_3 &= 4X^3 - 3X \\ T_4 &= 8X^4 - 8X^2 + 1. \end{aligned}$$

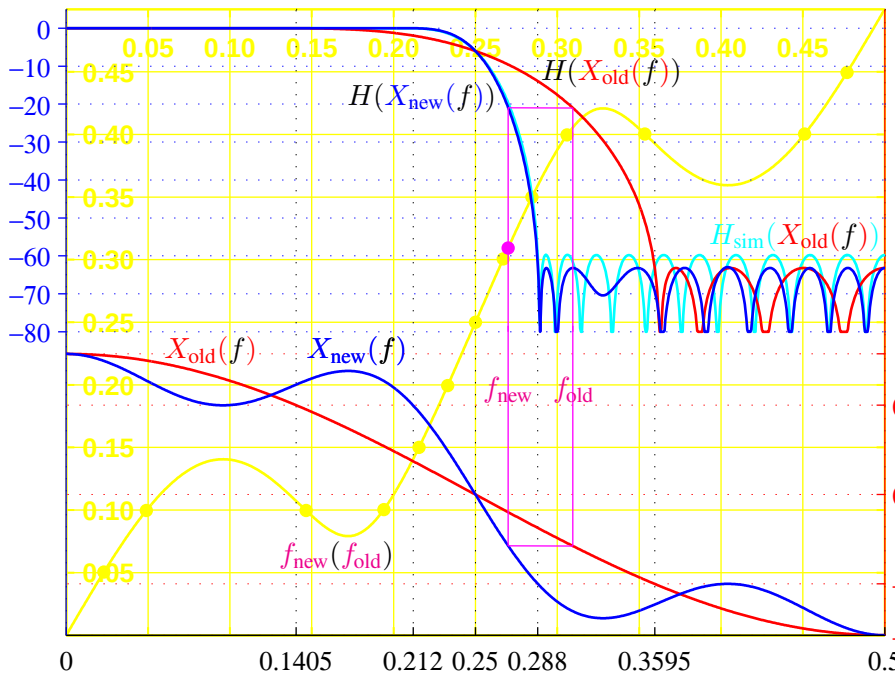


Fig. 3 Replacing the default one-add $X_{\text{old}}(f)$ with a five-add $X_{\text{new}}(f)$ (bottom, right scale) here transforms a half-band $H(X_{\text{old}}(f))$ with 4 nonredundant coefficients into $H(X_{\text{new}}(f))$ (top, left outside scale, dB). An alternative $H_{\text{sim}}(X_{\text{old}}(f))$ requires seven more coefficients than $H_{\text{sim}}(X_{\text{old}}(f))$ to perform similarly. The plot (inner scale) of $f_{\text{new}}(f_{\text{old}}) = X_{\text{old}}^{-1}(X_{\text{new}}(f_{\text{old}}))$ characterizes the net warping of frequency.

These polynomials $T_0, T_1, T_2, \ldots$ are actually the Chebyshev polynomials (of the first kind), for which the name T_k is in fact customary. Impulse response (1) can be written in both domains using these Chebyshev polynomials and their time-domain equivalents as

$$h(n) = h_0\, t_0(n) + \sum_{k=1}^{M} 2h_k t_k(n) \qquad \leftrightarrow \qquad H(X(f)) = h_0 + \sum_{k=1}^{M} 2h_k T_k(X(f)). \tag{5}$$

This relationship and the Chebyshev recursion above can be realized jointly to create a filter in what we might reasonably call *Chebyshev direct form* as shown in Fig. 1. The *reversed Chebyshev form* [8] of Fig. 2 is just the transpose of the Fig. 1 filter, the latter with every arrow of the signal flow graph reversed. The component impulse response $x(n)$ in these figures is as defined in (3) and so is not causal. Realization therefore requires shimming delays where **shown**.

2.2 A transformation: replace the linear-phase spreading function x

The temporal structure of the impulse response on the left side of (5) is completely contained in the functions $t_0, t_1, \ldots$, which contain impulses at various times, but it is the mapping in (5) from these $t_0, t_1, \ldots$ to $h(n)$ that contains all the coefficient information and hence the essence of this filter's particular frequency response. It is this separation that is interesting and useful, and it takes its cleanest form in the frequency domain. In fact, on the right in (5) the frequency response is written as $H(X(f))$ rather than the more conventional $H(f)$ to emphasize that it depends on f only through its dependence on $X(f)$. In the two-step mapping $f \to X \to H$ in effect f determines the value of $X(f)$, which then determines the value of $H(X(f))$. This means we can change $H(X(f))$ not only in the usual way, by changing the coefficients h_k that determine the second step $X \to H$, but by replacing the function $X(f)$ with a different function so as to affect the first step $f \to X$. We can replace it not only in analysis but in the implementation itself.

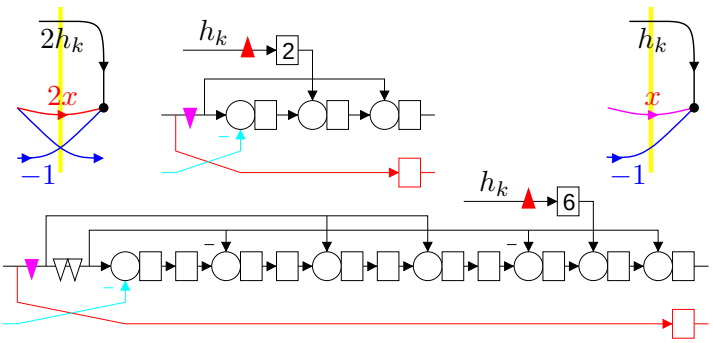


Fig. 4 The unit cells of Fig. 2 (upper corners here) can be realized as FPGA unit cells (center) realizing the standard linear-phase spreading function $x_{\text{old}}(n)$ (top) or the alternative $x_{\text{new}}(n)$ of Fig. 3 (bottom). Circles are adders, rectangles are delays, and block delays are numbered. Scalings by two and by $1/2$ are $\triangle$ and $\triangledown$ respectively. Colors denote special cases: for the left unit cell, for the right unit cell, for all cells but the first.

Consider the example in Fig. 3. There is a lot of information here, so it's best walked through slowly. Three distinct vertical scales are used, here termed the left, right, and inner scales, but the horizontal scale is the same for all curves and is just ordinary normalized frequency f. On the top the magnitude response of an example prototype filter $H(X_{\text{old}}(f))$ of length 15 is plotted against f using the left vertical dB scale. This particular prototype has been designed as a halfband filter, so on a linear scale the passband and stopband ripples are mirror images and the response is down by $1/2$ or about 6 dB at $f = 1/4$. On the bottom is plotted, against f on the linear right scale, $X_{\text{old}}(f) = \cos(2\pi f)$, the Fourier transform of the default linear-phase spreading function

$$x_{\text{old}}(n) = \tfrac{1}{2}\big(\delta(n+1) + \delta(n-1)\big).$$

Replacing $X_{\text{old}}(f)$ with $X_{\text{new}}(f)$, the Fourier transform of modified linear-phase spreading function

$$\begin{aligned} x_{\text{new}}(n) = \quad & \tfrac{1}{2}\big(\delta(n+1) + \delta(n-1)\big) \\ & - \tfrac{1}{8}\big(\delta(n+3) + \delta(n-3)\big) \\ & + \tfrac{1}{8}\big(\delta(n+5) + \delta(n-5)\big), \end{aligned}$$

replaces prototype response $H(X_{\text{old}}(f))$ with transformed response $H(X_{\text{new}}(f))$ in these plots by, in effect, warping the frequency axis. It's as if at any particular frequency f_{old} the values of $X_{\text{old}}(f_{\text{old}})$ and $H(X_{\text{old}}(f_{\text{old}}))$, one above the other in the plots, are linked as shown by the rightmost of the two vertical lines in Fig. 3 for an example f_{old}. The link is then moved horizontally (left in this example) to frequency f_{new} such that $X_{\text{old}}(f_{\text{old}}) = X_{\text{new}}(f_{\text{new}})$. Then $H(X_{\text{old}}(f_{\text{old}})) = H(X_{\text{new}}(f_{\text{new}}))$ as well.

There is another way to characterize the frequency warping graphically, one that we will prefer in subsequent discussion. Since $X_{\text{old}}(f)$ is strictly decreasing on the closed interval $[0, 0.5]$, it is invertible on that interval. Here the range of $X_{\text{new}}(f)$ is contained in (equals, in fact) that interval, so $X_{\text{old}}(f_{\text{old}}) = X_{\text{new}}(f_{\text{new}})$ implies $f_{\text{old}} = X_{\text{old}}^{-1}(X_{\text{new}}(f_{\text{new}}))$ or, for this $X_{\text{old}}(f)$ in particular, $f_{\text{old}} = \frac{1}{2\pi}\operatorname{acos}(X_{\text{new}}(f_{\text{new}}))$. In Fig. 3 this f_{old} is plotted faintly as a function of f_{new} on a third, inner vertical scale and with a • wherever this plotted f_{old} crosses a horizontal grid line. The shape of this curve reveals the character of the warping. A line with unit slope would represent no warping at all. Where this curve is steeper or shallower than that, the f_{new} frequencies are respectively being squeezed closer together or stretched further apart.

For this particular filter-design example, a preliminary resource count for Virtex-family FPGA implementation follows easily from the Fig. 4 sketches of pipelined unit-cell realizations for the reversed Chebyshev form (which may or may not be more efficient than the Chebyshev direct form). In the Virtex family one slice comprising two lookup tables is required for two wordwidth bits of a delayed adder, a pure delay, or a block delay. Taking zero coefficients into account, the new

transformed version of the Fig. 3 halfband filter requires four nontrivial coefficient multiplies and 87 slices per wordwidth bit pair. Obtaining similar performance by simply redesigning the original filter coefficients to replace prototype response $H(X_{\text{old}}(f))$ with the roughly similar response $H_{\text{sim}}(X_{\text{old}}(f))$ that is plotted for comparison in Fig. 3 requires 11 nontrivial coefficient multiplies and 94 slices per wordwidth bit pair when realized in conventional linear-phase transposed form. Relative to this conventional approach, the direct realization of transformed filter $H(X_{\text{new}}(f))$ has saved seven (nearly $2/3$) of the coefficient multiplies and seven slices (about 7%) per wordwidth bit pair (and it achieves slightly better performance).

3 The idea applied to 2D filters: the McClellan transformation

While these sorts of efficiency-improvement transformations can be useful in 1D filter design, it's in 2D filter design that the technique is the most powerful. In this context it is termed the *McClellan transformation* [2]. Here we use it to design weights for shaping the beam of a planar antenna array.

There are several stages to our transition to the 2D planar-array application. First, we restate our earlier discrete-time mathematics in continuous-time form. Second, we replace time with space and move from 1D to 2D. Finally, we explore the planar-array application through examples.

3.1 Replace discrete time with continuous time

Every discrete-time function has a continuous-time equivalent. Using a bar to indicate a function of continuous time, the continuous-time equivalent of $x(n)$, for example, is

$$\bar{x}(t) = \sum_n x(n)\,\delta(t - Sn). \tag{6}$$

The sample spacing would more typically be called T, but that letter has quite enough uses here already, so S for "Spacing" has been chosen instead.

If we likewise use the bar to denote the continuous-time frequency variable $\bar{f}$, the continuous-time Fourier transform of (6) is

$$\int \left(\sum_n x(n)\,\delta(t - Sn) \right) \mathrm{e}^{-\mathrm{j}2\pi\bar{f}t}\,dt = \sum_n x(n)\,\mathrm{e}^{-\mathrm{j}2\pi\bar{f}Sn} = X(\bar{f}S),$$

where $X(f)$ is the discrete-time Fourier transform used before. We get the continuous-time transform simply by evaluating discrete-time transform at $f = \bar{f}S$. We will write all continuous-time transforms this way and so will speak of, for example, continuous-time Fourier pair $\bar{x}(t) \leftrightarrow X(\bar{f}S)$.

Of course then the product $X(\bar{f}S)Y(\bar{f}S)$ of continuous-time transforms is just the product $X(f)Y(f)$ of discrete-time transforms evaluated at $f = \bar{f}S$, implying that $\bar{x}(t) * \bar{y}(t)$, which uses a continuous-time convolution, is the continuous-time equivalent of $x(n) * y(n)$, which uses a discrete-time convolution. Equivalence (6) is also linear in $x(n)$, so any time-domain relationship involving discrete-time functions, scaling by constants, addition and subtraction of functions, and convolution of functions has a continuous-time equivalent that can be obtained by simply replacing its constituent functions with continuous-time equivalents individually and then taking all convolutions to be in continuous time. It follows that the continuous-time equivalents of the definitions and

relationships of (4) and (5) are just

$$\begin{aligned}
\bar{t}_0(t) &\triangleq \delta(t) && \leftrightarrow && T_0(X) = 1 \\
\bar{t}_1(t) &\triangleq \bar{x}(t) && \leftrightarrow && T_1(X) = X \\
\bar{t}_k(t) &\triangleq 2\bar{x}(t) * \bar{t}_{k-1}(t) - \bar{t}_{k-2}(t) && \leftrightarrow && T_k(X) = 2XT_{k-1}(X) - T_{k-2}(X), \\
h(t) = h_0\bar{t}_0(t) + \sum_{k=1}^{M} 2h_k\bar{t}_k(t) && && \leftrightarrow && H(X(\bar{f}S)) = h_0 + \sum_{k=1}^{M} 2h_kT_k(X(\bar{f}S)).
\end{aligned} \tag{7}$$

Each occurrence of $X(\bar{f}S))$ in the recursion on the right has been written as just X for brevity and to emphasize that the form of that recursion is precisely that of the classic description of the Chebyshev polynomials.

The important thing here is that we still have a two-step frequency-domain mapping, now $\bar{f} \rightarrow X \rightarrow H$. We can modify $H(X(\bar{f}S))$ by changing filter coefficients to change the second step $X \rightarrow H$ or by replacing the linear-phase spreading function $\bar{x}(t)$ to change the first step $\bar{f} \rightarrow X$.

3.2 The planar-array application

The 2D FIR impulse and frequency responses of most interest to us are, respectively, array tapers and array factors, and here we define them in a way that makes ⟨array taper⟩ ↔ ⟨array factor⟩ a Fourier pair in various ways corresponding to different choices of normalization.

The usual periodic structure of the antenna elements of a narrowband planar antenna array can be characterized by nominal element locations that are integer-weighted combinations of 3D basis vectors, call them $\lambda\mathbf{s}_1$ and $\lambda\mathbf{s}_2$, where normalizing wavelength $\lambda = c/|\bar{f}|$ for $|\bar{f}|$ at the lower edge of the signal band. If dimensionless vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are made the columns of 3×2 element-**S**pacing basis matrix $\mathbf{S}$, every element then has a nominal location of form $\lambda\mathbf{S}\boldsymbol{n}$, where element index $\boldsymbol{n}$ is a two-element integer column vector that identifies the element.

We can arrive at a Fourier ⟨array taper⟩ ↔ ⟨array factor⟩ relationship by associating element weights to element positions normalized in one of four ways:

$$\text{weight } h(\boldsymbol{n}) \text{ associated with position } \begin{cases} \boldsymbol{n} & \text{fully normalized,} \\ \mathbf{S}\boldsymbol{n} & \text{normalized by } \lambda, \\ \pm\lambda\mathbf{S}\boldsymbol{n} & \text{no normalization.} \end{cases}$$

In the formulations that are most convenient we take $h(\boldsymbol{n})$ to be the weight associated with the element at physical location $\lambda\mathbf{S}\boldsymbol{n}$, but we can choose whether to carry full position information throughout or to instead let either λ or $\lambda\mathbf{S}$ remain unstated side information.

Fully normalized array taper $h(\boldsymbol{n})$ can be taken to be a 2D discrete-"time" (here actually space) function with 2D discrete-"time" (space) Fourier transform

$$H(\boldsymbol{f}) = \sum_{\boldsymbol{n}} h(\boldsymbol{n})\ \mathrm{e}^{-\mathrm{j}2\pi\boldsymbol{f}\boldsymbol{n}}\,. \tag{8}$$

This sum and others like it to follow are over all possible 2D element indices $\boldsymbol{n}$, but of course element weight $h(\boldsymbol{n})$ is nonzero only for those $\boldsymbol{n}$ corresponding to actively used elements. Here ⟨array taper⟩ ↔ ⟨array factor⟩ takes the form $h(\boldsymbol{n}) \leftrightarrow H(\boldsymbol{f})$, a 2D Fourier pair that makes the unit square in 2D normalized frequency $\boldsymbol{f}$ a period of fully normalized array factor $H(\boldsymbol{f})$. Of course Fourier pair $h(\boldsymbol{n}) \leftrightarrow H(\boldsymbol{f})$ mirrors exactly the structure of a 2D FIR filter and so motivates

this entire report, which is fundamentally about applying the McClellan transformation, a classic technique for 2D FIR filter design, to the design of array factors.

An array taper normalized by λ takes the form

$$\bar{h}(\mathbf{y}) \triangleq \sum_{\boldsymbol{n}} h(\boldsymbol{n})\, \delta(\mathbf{y} - \mathbf{S}\boldsymbol{n}), \tag{9}$$

where 3D column vector $\lambda\mathbf{y}$ of reals is position and $\mathbf{y}$ itself is therefore position normalized by λ. Element positions are represented by impulses with element weights as areas. Substituting (8) and (9) on the left and right readily verifies the equality in

$$H(\mathbf{aS}) = \int \bar{h}(\mathbf{y})\ \mathrm{e}^{-\mathrm{j}2\pi\mathbf{ay}}\, d\mathbf{y}, \tag{10}$$

where the integral is in 3D with $d\mathbf{y}$ a volume differential and where $\mathbf{a}$ is a real-valued 3D row vector. This provides an $\langle$array taper$\rangle \leftrightarrow \langle$array factor$\rangle$ relationship in the form $\bar{h}(\mathbf{y}) \leftrightarrow H(\mathbf{aS})$, a continuous-"time" (space) 3D Fourier pair with spatial frequency $\mathbf{a}$ as the transform variable. The curious fact that 3D function $H(\mathbf{aS})$ can be constructed from 2D function $H(\boldsymbol{f})$ is fundamentally due to the array being confined to a plane, a 2D subset of 3D.

The array factor in this λ-normalized form has natural interpretations. For a receive array, the component of the array output arising from signals with direction of $\widehat{a}$rrival (DOA) given by 3D row unit vector $\widehat{a}$ is just $H(\widehat{a}\mathbf{S})$ times the unweighted output of the single element at the spatial origin.[1] For a transmit array it is a little more complicated [9], but if we assume that the far-field radiated electromagnetic field due to a signal input to an element is everywhere linear in the product of that signal with the weight associated with that element, the electric or magnetic field at a point in the far field in direction $\widehat{a}$ is then just $H(\widehat{a}\mathbf{S})$ times the value of that same field when a single unweighted element at the spatial origin is used alone to transmit.

There's actually more than one way to state the $\langle$array taper$\rangle \leftrightarrow \langle$array factor$\rangle$ relationship with "no" normalization, because a free-space plane wave can be expressed with equal ease in terms of a radian-frequency wavenumber vector, $\frac{1}{2\pi}$ times that same vector, or the negative of either. For receive arrays that negative vector points in the DOA and so is quite natural. Most natural for us is to use spatial-frequency row vector $\mathbf{k} = \frac{1}{\lambda}\widehat{a}$, which makes the transmit-array and receive-array radian wavenumber vectors respectively $2\pi\mathbf{k}$ and $-2\pi\mathbf{k}$. Using unnormalized position $\mathbf{x} = \lambda\mathbf{y}$ and spatial frequency $\mathbf{k} = \frac{1}{\lambda}\mathbf{a}$ then, (8) yields

$$H(\mathbf{k}\lambda\mathbf{S}) = \sum_{\boldsymbol{n}} h(\boldsymbol{n})\, \mathrm{e}^{-\mathrm{j}2\pi\mathbf{k}\lambda\mathbf{S}\boldsymbol{n}} = \int \left(\sum_{\boldsymbol{n}} h(\boldsymbol{n})\, \delta(\mathbf{x} - \lambda\mathbf{S}\boldsymbol{n}) \right) \mathrm{e}^{-\mathrm{j}2\pi\mathbf{kx}}\, d\mathbf{x}, \tag{11}$$

thereby establishing Fourier-pair relationship

$$\sum_{\boldsymbol{n}} h(\boldsymbol{n})\, \delta(\mathbf{x} - \lambda\mathbf{S}\boldsymbol{n}) \ \leftrightarrow\ H(\mathbf{k}\lambda\mathbf{S})$$

between variables $\mathbf{x}$ and $\mathbf{k}$. (As a derivation strategy this direct integration of the delta functions in (11) is simpler than carrying out a change of variables on the integral in (10), because scaling the independent variable in a delta function is a little nonintuitive and for most people requires special care.)

As an aside, note that technically $\mathbf{k}$ can just as well be negated relative to this so that radian wavenumbers for a receive array take form $2\pi\mathbf{k}$, but that in this case the arguments to both h

[1] The length of this "unit vector" is precisely one only at the bottom of the signal band. For a truly narrowband array of course, that is all that matters.

and H become negated as well. This makes $h(-\boldsymbol{n})$ the weight associated with element position $\lambda\mathbf{S}\boldsymbol{n}$ and thereby moves the awkward minus sign from the spatial-frequency domain, where it only meant thinking of DOA instead of propagation direction, to the time domain, where there is just no convenient interpretation. This minus sign can be moved to many places in the formulation, but in the end it will not go away.

Overall then, the "fully normalized" formulation best ties the array application to FIR filter design, but the "normalized by λ" formulation with position $= \mathbf{S}\boldsymbol{n}$ will be the default going forward because unlike the fully normalized formulation it retains array size and shape information, makes classic element spacings like $\lambda/2$ and $\lambda/\sqrt{3}$ easy to see, makes common array-taper symmetries readily visible, and maps azimuth and elevation coordinates onto array-factor plots in an intuitive way.

3.2.1 Steering means we cannot restrict to unit vectors

Discrete-"time" Fourier-transform property $h(\boldsymbol{n})\,\mathrm{e}^{\mathrm{j}2\pi\widehat{s}\mathbf{S}\boldsymbol{n}} \leftrightarrow H(\boldsymbol{f} - \widehat{s}\mathbf{S})$ in 2D implies that using weights phase shifted in this way in the array taper results in an array factor $H\big((\widehat{a}-\widehat{s})\mathbf{S}\big)$ in which features near the origin in $H(\widehat{a}\mathbf{S})$ appear instead near some DOA steering vector $\widehat{s}$. Given this background fact, it is not necessary to explicitly consider steering here, except to realize that while $\widehat{a}$ and $\widehat{s}$ are each unit vectors, their difference $\widehat{a} - \widehat{s}$ in general is not. That difference vector may have a length approaching two. In respect of that, we generally compute and plot $H(\mathbf{aS})$ against 3D row vector $\mathbf{a}$ without requiring vector $\mathbf{a}$ to be a unit vector. Of course any boresight component in vector $\mathbf{a}$ is orthogonal to both basis vectors and therefore has no effect on product $\mathbf{aS}$, so it is enough to describe the behavior of $H(\mathbf{aS})$ versus the array-plane component of vector $\mathbf{a}$.

3.2.2 What array tapers are desired?

We will always design our unsteered $H(\mathbf{aS})$ to have a main beam at boresight, which is just the line of $\mathbf{a}$ values orthogonal to the array plane and therefore to both $\mathbf{s}_1$ and $\mathbf{s}_2$ so that $\boldsymbol{f} = \mathbf{aS} = 0$. This means we want $H(\mathbf{aS})$ to be a lowpass filter in the array-plane component of spatial frequency $\mathbf{a}$. Most often we want a narrow passband, and quite commonly we are willing to allow $H(\mathbf{aS})$ to display various kinds of symmetry, though we may not actually need such symmetry. Typical symmetries of this type are rotations about boresight and reflections through lines in the array plane that pass through the origin.

Array tapers typically have linear phase, yet another form of symmetry, though it is in no way necessary in general. It is, however, necessary if the McClellan transformation explored below is to apply. Typically the spatial origin is placed at the point of symmetry so that the $\langle$array taper$\rangle \leftrightarrow$ $\langle$array factor$\rangle$ in fact has zero phase: $\bar{h}(\mathbf{y}) = \bar{h}^*(-\mathbf{y})$ or $h(\boldsymbol{n}) = h^*(-\boldsymbol{n})$ with $H(\mathbf{aS})$ real.

Unsteered element weights are often real valued as well so that both the unsteered array taper and the unsteered array factor are real and even. This will be true of all the design examples in this report, even though the McClellan transformation does not in any way require it.

3.2.3 Taper loss

One simple characterization of a receive array is boresight SNR, the ratio of the array output power due to a plane-wave signal arriving from the boresight direction to the noise power output by the array. Suppose all array elements have identical but separate preamps with enough gain that their internally-generated noise levels dominate array output noise. In this case these array-output signal and noise power levels are respectively proportional to $|H(0)|^2$ and to $\|h\|^2 = \sum_{\boldsymbol{n}} |h_{\boldsymbol{n}}|^2$, the sum over all elements of the squared magnitudes of the weights.

One simple characterization of a transmit array is the ratio of the far-field signal power per steradian radiated in the boresight direction to the total power delivered to the elements by the transmitter. Suppose the transmitter(s) is(are) well matched to the elements at all steering angles. In this case this radiated signal-power density and the power delivered to the array by the transmitter are proportional to $|H(0)|^2$ and to $\|h\|^2$ respectively.

So for both transmit and receive arrays the ratio $|H(0)|^2/\|h\|^2$ is meaningful as a performance measure. We don't know the constants of proportionality that would enable us to directly interpret these numbers, but we can sidestep this issue by comparing this measure of array performance to the performance of a reference array described by discrete-time Fourier pair $r(\boldsymbol{n}) \leftrightarrow R(\boldsymbol{f})$. This report follows common practice and takes this reference array to be N uniformly weighted elements so that $r(\boldsymbol{n}) = 1$ and $R(0) = \|r\|^2 = N$.

In the $N = 1$ special case corresponding to a single embedded element used alone as a one-element array, the performance ratio $R(0)/\|r\|^2 = N$ goes to unity. In the more general case this ratio $R(0)/\|r\|^2$, for the reference array, or $H(0)/\|h\|^2$, for any array, can be interpreted as a performance gain relative to this single-element case. For a receive array this is classic *SNR gain*, and for a transmit array it is classic *power gain*.

While these gain figures are often given explicitly, in this report we will report them only relative to reference-array gain:

$$\frac{|H(0)|^2/\|h\|^2}{|R(0)|^2/\|r\|^2} = \frac{|H(0)|^2/\|h\|^2}{N} = \frac{|\sum_{\boldsymbol{n}} h_{\boldsymbol{n}}|^2}{\|h\|^2\, N} = \frac{|\langle h, r\rangle|^2}{\|h\|^2\|r\|^2,}$$

using on the right inner product $|\langle h, r\rangle|^2 = \sum_{\mathbf{n}} h_{\mathbf{n}} r^*_{\mathbf{n}}$ and squared norm $\|h\|^2 = \langle h, h\rangle$. The point of this seemingly odd formulation is that by the famous mathematical inequality of Cauchy, Schwarz, and Bunyakovsky in Hilbert spaces the ratio on the right cannot exceed unity. This implies that no taper can improve on the gain of a uniformly weighted array. For each design then the dB distance below this gain bound will be presented as

$$\langle \text{taper loss} \rangle = -10 \log_{10} \frac{|\langle h, r\rangle|^2}{\|h\|^2\|r\|^2.}$$

While one generally desires a small taper loss, other measures of performance are important as well. For example, one might want the best sidelobe rejection possible (by any of several measures) consistent with some tolerable level of taper loss.

3.2.4 Simple basis matrices for radar arrays

Here are clean ways to get the basis matrices of the most interest in radar, assuming we can orient the array plane relative to the coordinate frame in any way we like.

Square grid: Choose orthogonal vectors that are each zero in two components and of unit magnitude in the third. Scale the resulting matrix by $1/2$. Twenty-four distinct matrices of this type are possible, and they describe three distinct element grids with the optimal $\lambda/2$ spacing. Example: $\frac{1}{2}\begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix}$. (This one has array boresight in the $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ direction.)

Alternatively, zero any one common component of two noncollinear vectors and give the other components of each unit magnitude. Scale the resulting matrix by $1/(2\sqrt{2})$. This yields another 24 distinct matrices describing three more distinct grids with $\lambda/2$ spacing. Example: $\frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$. (This one has array boresight in the $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ direction.)

Equilateral triangular grid: Give the two vectors a single zero component each, and place that zero in different components for the two vectors. For each vector give each of the two other components unit magnitude. Scale the resulting matrix by $1/\sqrt{6}$. This yields 96 distinct matrices describing four distinct grids with the optimal $\lambda/\sqrt{3}$ spacing. Half the matrices have vectors $60°$ apart and half have them $120°$ apart. Example: $\frac{1}{\sqrt{6}}\begin{bmatrix}1 & -1\\ 1 & 0\\ 0 & 1\end{bmatrix}$. (This one has vectors $120°$ apart and has array boresight in the $\begin{bmatrix}1\\ -1\\ 1\end{bmatrix}$ direction.)

3.3 The transformation in 2D

The Chebyshev-form FIR filter structure in 1D continuous time with impulsive samples spaced by S is described by (6) and (7) and has frequency-response structure $\bar{f} \to X_{1D}(\bar{f}S) \to H(X_{1D}(\bar{f}S))$, with the latter depending on scalar continuous-time frequency $\bar{f}$. For the planar-array application we use alternative structure $\mathbf{a} \to X_{2D}(\mathbf{aS}) \to H(X_{2D}(\mathbf{aS}))$ in which the second constituent map $X \to H$ is exactly as before but is now preceded by the mapping $\mathbf{a} \to X_{2D}(\mathbf{aS})$ of a 3D row-vector frequency $\mathbf{a}$ into a scalar-valued $X_{2D}(\mathbf{aS})$ using 2D FIR-filter frequency response $X_{2D}(\boldsymbol{f})$ in discrete "time" (space) and 3×2 element-spacing basis matrix $\mathbf{S}$. The original 1D linear-phase spreading function represented by continuous-time Fourier pair $\bar{x}_{1D}(t) \leftrightarrow X_{1D}(\bar{f}S) = \cos(2\pi\bar{f}S)$ is replaced in the planar-array application with a continuous-"time" (space) transform pair $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$, where the latter 3D Fourier-pair relationship

$$X_{2D}(\mathbf{aS}) = \int \bar{x}_{3D}(\mathbf{y})\, \mathrm{e}^{-\mathrm{j}2\pi\mathbf{ay}}\, d\mathbf{y} \tag{12}$$

follows from definitions that determine both members of the pair from coefficients $x_{2D}(\boldsymbol{n})$ indexed by a two-vector $\boldsymbol{n}$ of integers:

$$X_{2D}(\boldsymbol{f}) \triangleq \sum_{\boldsymbol{n}} x_{2D}(\boldsymbol{n})\, \mathrm{e}^{-\mathrm{j}2\pi\boldsymbol{f}\boldsymbol{n}} \tag{13}$$

$$\bar{x}_{3D}(\mathbf{y}) \triangleq \sum_{\boldsymbol{n}} x_{2D}(\boldsymbol{n})\, \delta(\mathbf{y} - \mathbf{S}\boldsymbol{n}). \tag{14}$$

Here (13) makes $x_{2D}(\boldsymbol{n}) \leftrightarrow X_{2D}(\boldsymbol{f})$ a 2D discrete-time Fourier pair. Relationships (12) and (14), which move that pair relationship into 3D spatial coordinates, are structurally analogous to (10) and (9), so technically 3D Fourier pair $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$ has an ⟨array taper⟩ $\leftrightarrow$ ⟨array factor⟩ structure exactly as if $x_{2D}(\boldsymbol{n})$ were the element weights for a (presumably) tiny planar array. But of course there is no such array. Instead, the utility of $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$ is in permitting the easy construction of a ⟨array taper⟩ $\leftrightarrow$ ⟨array factor⟩ pair for a large array.

That construction of a ⟨array taper⟩ $\leftrightarrow$ ⟨array factor⟩ pair was effected originally through the Chebyshev recursion in (7), which used repeated time-domain convolution with $\bar{x}_{1D}(t)$ to spread a seed function $\bar{t}_0(t) = \delta(t)$ farther and farther out in time to provide basis functions $\bar{t}_0(t), \bar{t}_1(t), \bar{t}_2(t), \ldots$ from which an arbitrary linear-phase filter impulse response could be built. The new Chebyshev recursion uses repeated convolution in the array-plane subspace of 3D "time" (space) in a similar way. Convolution with $\bar{x}_{3D}(\mathbf{y})$ spreads a seed function $\bar{t}_0(\mathbf{y}) = \delta(\mathbf{y})$ farther and farther out in the array plane to provide basis functions $\bar{t}_0(\mathbf{y}), \bar{t}_1(\mathbf{y}), \bar{t}_2(\mathbf{y}), \ldots$ from which an arbitrary linear-phase array

taper can be built. The planar-array analog of (7),

$$\begin{aligned}
\bar{t}_0(\mathbf{y}) &\triangleq \delta(\mathbf{y}) && \leftrightarrow && T_0(X) = 1 \\
\bar{t}_1(\mathbf{y}) &\triangleq \bar{x}_{3D}(\mathbf{y}) && \leftrightarrow && T_1(X) = X \\
\bar{t}_k(\mathbf{y}) &\triangleq 2\bar{x}_{3D}(\mathbf{y}) * \bar{t}_{k-1}(\mathbf{y}) - \bar{t}_{k-2}(\mathbf{y}) && \leftrightarrow && T_k(X) = 2XT_{k-1}(X) - T_{k-2}(X), \\
\bar{h}(\mathbf{y}) &= h_0 \bar{t}_0(\mathbf{y}) + \sum_{k=1}^{M} 2h_k \bar{t}_k(\mathbf{y}) && \leftrightarrow && H(X_{2D}(\mathbf{aS})) = h_0 + \sum_{k=1}^{M} 2h_k T_k(X_{2D}(\mathbf{aS})),
\end{aligned} \quad (15)$$

follows from the new definitions of $\bar{t}_0(\mathbf{y})$ and $\bar{x}_{3D}(\mathbf{y})$, with only the latter representing a meaningful change as $\bar{t}_0(\mathbf{y})$ is and must be simply the identity element for convolution in the function space in which $\bar{x}_{3D}(\mathbf{y})$ is defined. The rest is derived from these two quantities just as before.

Because we need $X_{2D}(\mathbf{aS})$ to be real valued, $\bar{x}_{3D}(\mathbf{y})$ should be a zero-phase impulse response, which is to say that it must have conjugate symmetry $\bar{x}_{3D}(\mathbf{y}) = \bar{x}^*_{3D}(-\mathbf{y})$ or, equivalently, $x_{2D}(\boldsymbol{n}) = x^*_{2D}(-\boldsymbol{n})$. We also need the range of $X_{2D}(\mathbf{a})$ to be contained in the closed interval $[-1, 1]$, which was the range of the original $X_{1D}(\bar{f})$ in 1D, but this is easy to arrange: we can scale and offset any preliminary choice of a zero-phase $X_{2D}(\mathbf{a})$ by constants as necessary to fix its range to $[-1, 1]$. This scales $\bar{x}_{3D}(\mathbf{y})$ and offsets it by a delta function.

3.3.1 Design example: the simplest spreading function has seven samples

Let us work with a concrete example as we continue. For now suppose in particular that the basis-vector columns of matrix $\mathbf{S}$ have length $1/\sqrt{3}$ and that they are $60°$ apart (whether or not chosen in the way described earlier in Section 3.2.4). This puts the array elements on an equilateral-triangular grid, a scaled version of the so-called hexagonal lattice, with elements spaced at the classic maximum if grating lobes are to be avoided at all scan angles. Array taper $\bar{h}(\mathbf{y})$ will comprise impulses on points of the lattice as desired if and only if $\bar{x}_{3D}(\mathbf{y})$ comprises such impulses. To the extent that we can do so while staying in the lattice, let us give $\bar{x}_{3D}(\mathbf{y})$ roughly equal extent in all directions. The simplest initial choice locates unit impulses at the points of a hexagon centered on the origin, at the origin's nearest neighbors in the lattice:

$$\begin{aligned}
\bar{x}_{3D}(\mathbf{y}) = \\
&\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]) + \delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]) + \delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right]) \\
+ \delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right]) &+ \delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}0\\-1\end{smallmatrix}\right]) + \delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]).
\end{aligned}$$

This Fourier transforms to an array factor $X_{2D}(\mathbf{aS})$ comprising three cosine terms. It has range $[-3, 6]$, so scaling by $2/9$, which yields range $\left[-\frac{2}{3}, \frac{4}{3}\right]$, and subsequent addition of constant $-1/3$ fixes the range at the desired $[-1, 1]$. The final form is

$$\begin{aligned}
\bar{x}_{3D}(\mathbf{y}) = \\
&-\tfrac{1}{3}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]) \\
&+\tfrac{2}{9}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]) + \tfrac{2}{9}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]) + \tfrac{2}{9}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}-1\\1\end{smallmatrix}\right]) \\
&+\tfrac{2}{9}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right]) + \tfrac{2}{9}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}0\\-1\end{smallmatrix}\right]) + \tfrac{2}{9}\delta(\mathbf{y} - \mathbf{S}\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]).
\end{aligned} \quad (16)$$

The first seven Chebyshev-derived basis functions in the "time" (space) domain that result from this $\bar{x}_{3D}(\mathbf{y})$ are depicted in Fig. 5 in seven separate sketches. Each sketch lies entirely in the array plane, which has axes shown faintly. The top sketch represents the single unit impulse of $\bar{t}_0(\mathbf{y})$. Moving down, subsequent sketches depict $\bar{t}_1(\mathbf{y}), \ldots, \bar{t}_6(\mathbf{y})$. The numerator of each integer-ratio

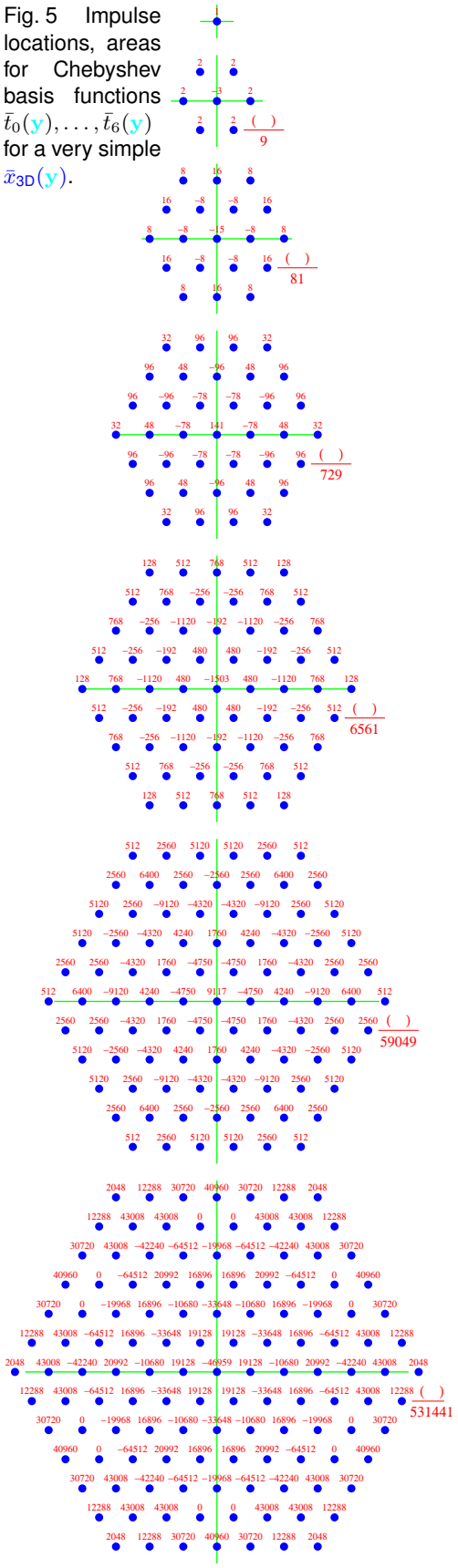


Fig. 5 Impulse locations, areas for Chebyshev basis functions $\bar{t}_0(\mathrm{y}), \ldots, \bar{t}_6(\mathrm{y})$ for a very simple $\bar{x}_{3D}(\mathrm{y})$.

impulse area is shown above its impulse-location dot •, and the denominator common to the dots of that sketch appears to the right of the sketch. Basis function $\bar{t}_k(\mathrm{y})$ has impulses on a hexagonal subset of the lattice with $k{+}1$ points on a side and containing $1{+}3k(k{+}1)$ points. The latter ties array size to prototype filter length when the McClellan transformation is used with this particular linear-phase spreading function $\bar{x}_{3D}(\mathrm{y})$.

It was apparent from the outset that the hexagonal shape of $\bar{x}_{3D}(\mathrm{y})$ would result in basis functions $\bar{t}_1(\mathrm{y}), \bar{t}_2(\mathrm{y}), \ldots$ with support confined to a hexagonal subset of the lattice, and indeed the nature of convolution often allows one to anticipate the general shape of the basis-function support. The impulse areas are another matter, however. The pattern of impulse areas in Fig. 5 quickly becomes nonintuitive as k increases. The $\bar{t}_6(\mathrm{y})$ sketch at the bottom even shows 12 areas as zero: those impulses are simply absent. The symmetries in the areas, the invariance to various rotations and reflections, are certainly no surprise given the presence of like symmetries in $\bar{x}_{3D}(\mathrm{y})$, but the only viable approach to learning anything more about the impulse areas appears to be to simply use the recursion on the left side of (15) to compute them, as was done here.

Implementing such a recursion to compute either the basis functions or the array weights themselves is not difficult. Figure 6 presents a matlab realization of a 2D taper computation integrated into the basis-function recursion by following the reversed-Chebyshev-form FIR-filter structure of Fig. 2. The Fig. 6 code as shown operates on 1D Taylor weights and the simple $\bar{x}_{3D}(\mathrm{y})$ of (16), but any 1D linear-phase taper or FIR weights can be substituted for the former, and any linear-phase $\bar{x}_{3D}(\mathrm{y})$ structured as impulses on a lattice can be substituted for the latter. McClellan transformation function `ftrans2` from matlab's Image Processing Toolbox should provide an alternative to most of the Fig. 6 code. However, this author is not a user of that toolbox and so has not experimented with that function.

Figure 8 illustrates the transformation of an example 1D array factor into 2D using the simple $\bar{x}_{3D}(\mathrm{y})$ of (16) in a McClellan transformation. The array-plane contour plot on the left of the second row shows several items together. The plane is viewed from "behind" along the boresight direction, and the origin is in the center.

```
w=1;M=39;h=taylorwin(2*M+1,12,-35);h=h/sum(h);
x=[ 0       2   2         ;...
          2  -3   2       ;...
            2   2         0]/9;cx=2;
t=cell(1,3); ct=zeros(1,3);
older=1; t{older}=0; ct(older)=1;
old=2;t{old}=2*h(1)*w; ct(old)=1;
new=3;
for m=2:M
  tot=conv2(2*x,t{old});
  c=cx+ct(old)-1; d=ct(older)-1;
  tot(c,c)=tot(c,c)+2*h(m)*w;
  tot(c-d:c+d,c-d:c+d)=tot(c-d:c+d,c-d:c+d)-t{older};
  t{new}=tot; ct(new)=c;
  next=older; older=old; old=new; new=next;
end
out=conv2(x,t{old});
c=cx+ct(old)-1; d=ct(older)-1;
out(c,c)=out(c,c)+h(M+1)*w;
out(c-d:c+d,c-d:c+d)=out(c-d:c+d,c-d:c+d)-t{older};
out=out*(1+3*M*(M+1)); %optional scaling for this x
```

Fig. 6 Matlab realization of the McClellan transformation computes $\mathtt{w}\,\bar{h}(\mathbf{y})$ from

- scalar input `w`, here unity,
- 1D linear-phase weight vector `h` of length $2\mathtt{M}+1$, here the Taylor weights from the top line of Fig. 8, and
- 2D linear-phase spreading function $\bar{x}_{\mathbf{3D}}(\mathbf{y})$, here from example (16).

The reversed-Chebyshev-form iteration of Fig. 2 is used with each 2D function represented by a square matrix and the index of its `c`enter. For example, $\bar{x}_{\mathbf{3D}}(\mathbf{y})$ here becomes matrix `x` and center index `cx` with `x(i,j)` representing $\mathtt{x(i,j)}\,\delta\big(\mathbf{y}-\mathbf{S}\left[\begin{smallmatrix}\mathtt{i-cx}\\ \mathtt{j-cx}\end{smallmatrix}\right]\big)$. The code assumes $\mathtt{M}\geq 1$. Result $\mathtt{w}\,\bar{h}(\mathbf{y})$ appears as matrix `out` and center index `c`.

Basis vectors are shown in the center with the first and second columns of basis matrix $\mathbf{S}$ in different colors.

Linear-phase spreading function $\bar{x}_{\mathbf{3D}}(\mathbf{y})$ is shown as impulse-location marks, index vectors $\left[\begin{smallmatrix}n_1\\ n_2\end{smallmatrix}\right]$ that specify those locations in $\mathbf{S}\left[\begin{smallmatrix}n_1\\ n_2\end{smallmatrix}\right]$ form, and the impulse areas in a legend at the lower left. These are intentionally made a bit hard to see, so as not to distract from the main items described below.

Having the first basis vector point downward to the right and the second point rightward allows these directions to correspond to how row and column indices increase on the page when a weight matrix is presented with indentation increasing for each successive line, in the style of `x` in Fig. 6.

The scale of the two plot items above is implied by the earlier discussion of the basis vectors used. (Recall that they have been given length $1/\sqrt{3}$ here to indicate element spacing of $\lambda/\sqrt{3}$.) These next items, in that same plot box, are on a completely separate and unrelated scale from the above. They share the box with the above only to save space.

A projected hemisphere with longitude and latitude lines at 10° intervals respectively representing azimuth and elevation relative to a boresight vector normal to the array plane. Here the array plane and the boresight vector are assumed to be oriented vertically and horizontally respectively, and azimuth increases in the clockwise direction when the array is viewed from above. This hemisphere has unit radius, so alternatively one can view it as a 3D globe with direction unit vector $\widehat{a}$ extending from its center to its surface. The viewer is in effect in the center of the globe looking through the page at the hemisphere bulging out on the far side.

The hexagonal boundary of a period of array factor $X_{\mathbf{2D}}(\mathbf{aS})$**.** It circumscribes the hemisphere. Of course frequency-warping function $X_{\mathrm{1D}}^{-1}(X_{\mathrm{2D}}(\mathbf{aS}))$ and array factor $H(X_{\mathrm{2D}}(\mathbf{aS}))$ have the same periodicity as $X_{\mathrm{2D}}(\mathbf{aS})$.

The frequency-warping function $X_{\mathrm{1D}}^{-1}(X_{\mathrm{2D}}(\mathbf{aS}))=\frac{1}{2\pi}\,\mathrm{acos}(X_{\mathrm{2D}}(\mathbf{aS}))$ in contour-plot form as a function of the array-plane component of vector $\mathbf{a}$. See Table 1 for contour values.

The top row of Fig. 8 presents the dB magnitude of 1D prototype response $H(X_{\mathrm{1D}}(\bar{f}S))$ with the vertical axis extending from -60 dB to 1 dB, with tick marks at multiples of 10 dB, and with the horizontal axis representing $\bar{f}S$ values from 0 to 0.5 if the response is interpreted as a linear-phase

Table 1 The contours of the frequency-warping plots correspond to horizontal-axis values from the transformed 1D response. That axis can be interpreted as the usual normalized discrete-time frequency of an FIR filter or as the direction cosine of a classical $\lambda/2$ spaced uniform line array.

in plane	normalized FIR frequency $\bar{f}S$		direction cosine $\lambda\bar{f}$ assuming $S = \lambda/2$	
	major contours	minor contours	major contours	minor contours
at the origin	**0**		**0**	
	0.02 0.04 0.06 0.08		0.04 0.08 0.12 0.16	
	0.1		**0.2**	
	0.12 0.14 0.16 0.18		0.24 0.28 0.32 0.36	
	0.2		**0.4**	
	0.22 0.24 0.26 0.28		0.44 0.48 0.52 0.56	
	0.3		**0.6**	
	0.32 0.34 0.36 0.38		0.64 0.68 0.72 0.76	
	0.4		**0.8**	
near peripheral extremes	0.42 0.44 0.46 0.48		0.84 0.88 0.92 0.96	
	0.5		**1.0**	

FIR-filter response and representing $\lambda\bar{f}$ values from 0 to 1 if the response is interpreted as the array factor of a uniform line array with classic element spacing $S = \lambda/2$ and a linear-phase taper. In the latter case axis value $\lambda\bar{f}$ is the direction cosine, the cosine of the angle from the line of the array or the sine of the angle from boresight. The two interpretations are offered here and in Table 1 because the filter and line-array views will likely be more natural to signal-processing and array researchers respectively. Horizontal axis values corresponding to contour levels are marked with short vertical lines just above the plot's centerline.

The top-row prototype response is actually shown for two choices of its 79 weights $h(n)$. The light, wider line corresponds to Taylor weights with a -35 dB sidelobe level and $\bar{n} = 4$. Taylor weights are not optimal in any way but may be convenient for designers lacking optimization tools or the time to deal with the learning curves of such tools. The dark, narrower line corresponds to weights optimized to minimize the energy in the sidelobe region of the response, the left edge of which for this purpose is marked with a ○ at 5 dB above the plot's bottom edge. That minimization is subject to these several constraints.

- The boresight response must not be less than 0 dB.
- Taper loss as a line array must not exceed 0.85 dB.
- Response magnitude must not exceed -35 dB anywhere on a grid of closely spaced frequencies (or direction cosines) in the sidelobe region that extends rightward from the × mark (plotted at that -35 dB level).

The first two of these constraints are satisfied with equality, of course, and the third is satisfied with equality for some grid frequencies.

This optimization specification results in a response quite similar to the Taylor response and so indirectly sheds light on the Taylor response itself: this is more or less the sense in which the Taylor response is roughly optimal. Later we'll choose optimality criteria quite differently, as there

is nothing terribly special about this combination of criteria. Meanwhile, here the Taylor response is for comparison purposes only; the optimized response is used as the prototype for the McClellan transformation to follow.

McClellan transformation (15) uses the linear-phase spreading function $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$ of (16) and is represented by the contour plot on the second line of Fig. 8 to map the optimized 1D response $H(X_{1D}(\bar{f}S))$ of the top line to a 2D array factor $H(X_{2D}(\mathbf{aS}))$ with the dB magnitude response plotted on two different scales to the right of the contour plot. In both cases plot colors correspond to dB as shown on the colorbar at the figure's lower left, but the extent of the center plot in the array plane matches the extent of the contour plot, while the plot on the right, though still centered on boresight, covers an area only about $7.5°$ square. On the latter plot contours have been added at -1, -2, and -3 dB to make it easy to visually estimate typical beamwidth measures.

The feathery look deep in the circular array-factor nulls of the rightmost of those plots is a plotting artifact, as are the center plot's apparent small discontinuities at elevations near $-35°$, $-27°$, $-12°$, $8°$, and $14°$. Similar discontinuities are visible in the rightmost plot as well, particularly in the contours. The feathers disappear when a sufficiently high plot resolution is used, but at the expense of course of burdensome plot computations and enormous plot files. The discontinuities are not present in the original plot but appear during a file format conversion.

Frequency-warping function $X_{1D}^{-1}(X_{2D}(\mathbf{aS}))$ shown using contours in Fig. 8 is analogous to the light curve in Fig. 3. In the latter the curve is a function of 1D frequency f, but here, in Fig. 8, it is a function of the 2D array-plane component of 3D spatial-frequency vector $\mathbf{a}$. The level-crossing dots • on the Fig. 3 curve are analogous to contours in Fig. 8 and convey exactly the same meaning. The contour that marks FIR frequency $\bar{f}S = \mathbf{0.4}$ or direction cosine $\lambda\bar{f} = \mathbf{0.8}$ lies along positions in the array plane where transformed array factor $H(X_{2D}(\mathbf{aS}))$ takes on the same value that $H(X_{1D}(\bar{f}S))$ has at $\bar{f}S = \mathbf{0.4}$ or $\lambda\bar{f} = \mathbf{0.8}$. Thus contours are level curves not only for the frequency-warping function but for the transformed array factor as well: the path in the array plane followed by any contour finds the transformed array factor holding absolutely constant. This gives that array factor a very unusual look, with individual sidelobes extending over long distances in the plane.

The third row of plots in Fig. 8 is like the second except that Fourier pair $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$ of the second row has been replaced with steered version $\bar{x}_{3D}(\mathbf{y})\,\mathrm{e}^{\mathrm{j}2\pi\widehat{s}\mathbf{y}} \leftrightarrow X_{2D}((\mathbf{a}-\widehat{s})\mathbf{S})$, where unit vector $\widehat{s}$ is at $40°$ azimuth and $30°$ elevation. The resulting frequency-warping function $X_{1D}^{-1}(X_{2D}(\mathbf{aS}))$ and array factor $H(X_{2D}(\mathbf{aS}))$ respectively become steered functions $X_{1D}^{-1}(X_{2D}((\mathbf{a}-\widehat{s})\mathbf{S}))$ and $H(X_{2D}((\mathbf{a}-\widehat{s})\mathbf{S}))$. The point made here is simple: in those systems in which the McClellan transformation is used to compute weights in real time, explicit phase-shift steering of the perhaps thousands of weights of taper $\bar{h}(\mathbf{y})$ is unnecessary. It is enough to steer the very few weights of spreading function $\bar{x}_{3D}(\mathbf{y})$. The McClellan transformation handles the rest.

To put the performance of the array factor obtained here through the McClellan transformation into context, the bottom row of plots in Fig. 8 presents an array factor for comparison for which no McClellan transformation was used. Instead the corresponding 2D taper was optimized directly. This array is not hexagonal in outline but more or less circular: elements are placed at all points of the $\lambda/\sqrt{3}$ spaced hexagonal lattice that lie at distance $\lambda\sqrt{1237/3}$ or less from the origin. This yields an array with 4507 elements, making it some 3.7% smaller than the design of Fig. 8 plot rows two and three. The taper was made invariant to rotations by multiples of $60°$ and to reflections through axes at multiples of $30°$ from the azimuth axis. This meant only 404 distinct weights required optimization, with the twelvefold symmetry determining the rest.

The optimization maximized SNR gain by minimizing noise gain while constraining the boresight array factor to at least unity, and additional constraints were imposed to bound the array-factor magnitude to no more than -35 dB outside an array-plane circle centered on the origin and extend-

ing to 2.5° in azimuth and elevation. The latter required individual constraints at approximately 8,000 specific points in the array plane, all in (the closure of) a wedge covering $1/12$ of an array-factor period with the rest of the array plane taken care of by the imposed symmetries. The optimization specification was converted to a second-order cone program (SOCP) by our Opt matlab toolbox [10] and solved by the SeDuMi numerical solver [11, 12] (The latter tools were also used for the earlier 1D optimization.)

The optimized array factor in the bottom row appears superior to the McClellan-transformation design of the second row in several ways. It has a very slightly narrower beamwidth. It has similar sidelobes near the main beam but considerably smaller sidelobes everywhere far from the main beam. It uses fewer elements and yet has some 0.83 dB more SNR gain by virtue of having much lower taper loss. It is not clear from this one example whether the McClellan-transformation approach is seriously inferior to 2D optimization or whether we just need to be more clever in applying it. The next few designs will move towards better McClellan designs in hopes of resolving this question.

The taper-loss numbers in Fig. 8 do demonstrate nicely that taper loss is not preserved by the McClellan transformation from 1D to 2D. From the Section 3.2.3 discussion, the increase in taper loss in going from 1D to 2D should be

$$10 \log_{10} \frac{N_{\text{2D}} \|h_{\text{2D}}\|^2}{N_{\text{1D}} \|h_{\text{1D}}\|^2}$$

in terms of the numbers N_{1D} and N_{2D} of array elements in 1D and 2D and the sums $\|h_{\text{1D}}\|^2$ of $\|h_{\text{2D}}\|^2$ of squared magnitudes of the 1D and 2D weights. Obtaining zero taper-loss increase would require

$$\|h_{\text{2D}}\|^2 = \frac{N_{\text{1D}}}{N_{\text{2D}}} \|h_{\text{1D}}\|^2,$$

which in this particular case would amount to a "noise power beamwidth" (in analogy to the noise power bandwidth of a filter) of 1.94°. This is fairly close to the 3 dB beamwidth obtained on the right in plot row two, so the observed taper-loss deterioration in going from 1D to 2D of 0.96 dB, a power ratio of about 5/4, likely arises largely from the contribution of the large 2D sidelobe area to $\|h_{\text{2D}}\|$. This suggests sidelobe suppression as important to improve in future McClellan-transformation experiments.

3.3.2 Design example: a larger spreading function allows beam spoiling

To obtain more control over Fourier pair $\bar{x}_{\text{3D}}(\mathrm{y}) \leftrightarrow X_{\text{2D}}(\mathbf{aS})$ and hence, through the McClellan transformation, over transformed array factor $H(X_{\text{2D}}(\mathbf{aS}))$, in Fig. 9 we increase the spatial extent of $\bar{x}_{\text{3D}}(\mathrm{y})$ from seven impulses on element-location lattice points to nineteen impulses on two concentric hexagons of points around the origin. Full 12-fold symmetry on the impulse weights reduces the number of design degrees of freedom to four, and setting the overall scale and origin-impulse area to give $X_{\text{2D}}(\mathbf{aS})$ a range of $[-1, 1]$ further reduces it to two. These two are in effect the ratio of the side impulse area to the corner impulse area on the outer hexagon, the "side ratio," and the ratio of the corner area on the outer hexagon to the corner area on the inner hexagon, the "corner ratio."

Two frequency-warping functions $X_{\text{1D}}^{-1}(X_{\text{2D}}(\mathbf{aS}))$ are presented.

narrow beam: In the second plot row the side and corner ratios are each set to $1/2$ to give a bit of taper to the outermost impulses. This $\bar{x}_{\text{3D}}(\mathrm{y}) \leftrightarrow X_{\text{2D}}(\mathbf{aS})$ pair, chosen by trial and error, yields a narrow beam in transformed array factor $H(X_{\text{2D}}(\mathbf{aS}))$ and gives it a simple sidelobe structure.

wide beam: In the third plot row the side ratio is set to zero to give pair $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$ a "third band" structure [13], which with small sidelobes will automatically force a wide, flat beam in $X_{2D}(\mathbf{aS})$. The corner ratio is set experimentally to about -0.179 to obtain low outer sidelobes in $X_{2D}(\mathbf{aS})$. A wide beam in transformed array factor $H(X_{2D}(\mathbf{aS}))$ results.

The top-row prototype response is again shown in Taylor and optimized versions, but now the latter has the taper-loss bound loosened to 1.25 dB to give the minimization more latitude to suppress sidelobe energy in hopes of lowering the taper-loss penalty in going from 1D to 2D. This strategy does not appear particularly successful, but further investigation would be required in order to say anything definitive on this point. All we can really say here is that the transformed 2D sidelobes now appear at a more generally reasonable level for the array application.

The main purpose of this example is to show that a modest change in the spreading function can be used to broaden or "spoil" the beam of transformed array factor $H(X_{2D}(\mathbf{aS}))$. Here the 3 dB beamwidth was increased by a factor of about 3.4. Unfortunately, this spoiling feature cannot be used for phase-only transmit tapers, because the McClellan transformation transforms phase-only 1D tapers to 2D tapers that lack the phase-only property.

The taper loss shown for the broadened-beam case is over 12 dB, but most of this is simply an unavoidable geometrical phenomenon tied to the beam broadening itself in a very fundamental way: beamwidth broadening by $3.4\times$ would naturally increase the portion of the taper loss associated with the main beam by some $20 \log_{10} 3.4 \approx 10.6$ dB. The taper loss here increases by less than that amount in going from the narrow-beam case to the wide-beam case, indicating that a significant part of the narrow-beam taper loss comes from the energy in the sidelobe structure. Discussion of taper loss in greater depth and an approach to modifying the measure appropriately for wide-beam cases is detailed in [14].

How much can full optimization improve on the third-row design? The optimized design presented in the bottom row of Fig. 9 begins with array elements at the 4315 points of the $\lambda/\sqrt{3}$ spaced hexagonal lattice that lie at distance $\lambda\sqrt{1191/3}$ or less from the origin, making the array about 0.4% smaller than the design of Fig. 9 plot rows two and three. The taper is again constrained to be twelvefold symmetric and is optimized to minimize noise gain with additional constraints bounding the array-factor magnitude to no more than -35 dB outside a circle with a 10° radius.

Recall that the bottom-row optimization in Fig. 8 was similar but additionally required the array factor to exceed unity at the origin so that the minimization of noise gain would not simply zero all the weights. Here the array factor is instead required, everywhere inside a circle of radius 3.3°, approximating the 3 dB beamwidth, to exceed the broadened third-row array factor but to not exceed unity. Sandwiching the main beam between these bounds forces a beam shape very similar to that shown in the third row. We can then compare the sidelobe levels to get a sense of how much better true optimization is than the McClellan approach used for row three.

The optimized fourth-row taper turns out to be quite comparable to the third-row McClellan-transformed design in 3 dB beamwidth, in taper loss, and in first-null beamwidth, but its sidelobe performance is so much better than the McClellan-transformed design that a different color scale was used to plot it, one that covers 50% more dB.

So while the ready availability of simple beamspoiling is an attractive feature of the McClellan transformation, the question of ultimately obtainable performance for a given number of elements appears to be serious and suggests the next experiment.

3.3.3 Design example: trading ripple for beamwidth

The Fig. 10 design experiment has four aims.

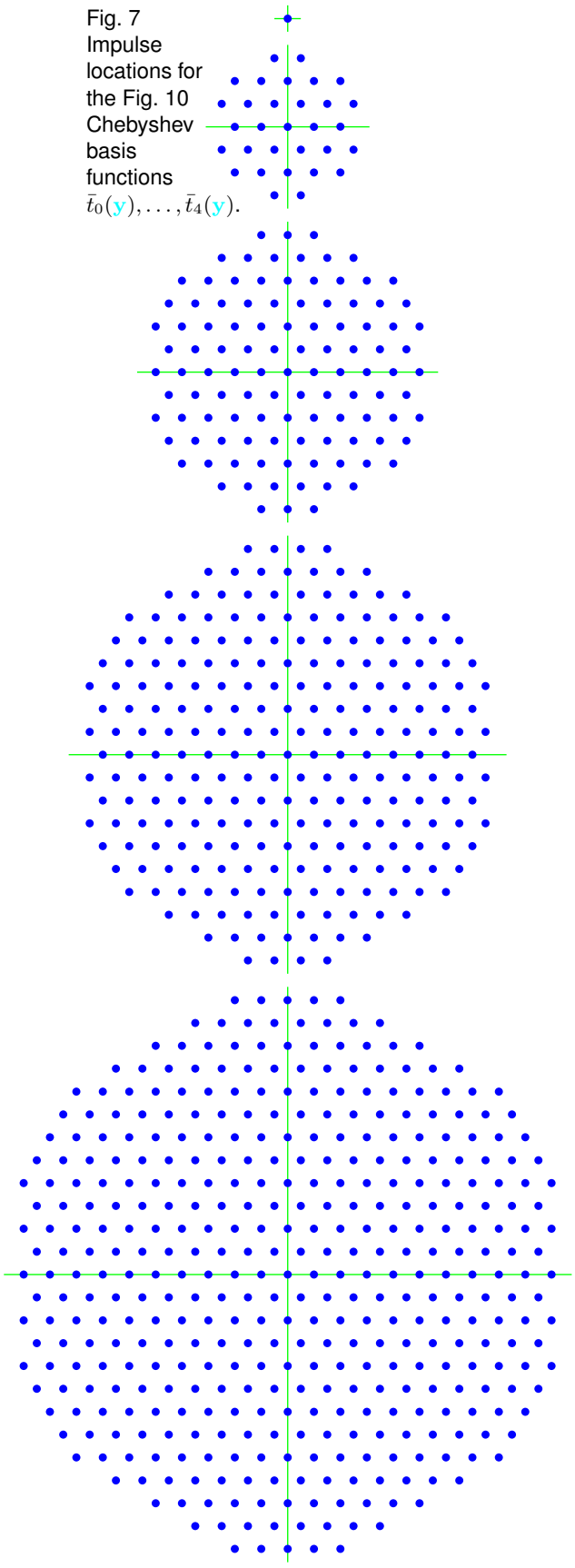


Fig. 7
Impulse locations for the Fig. 10 Chebyshev basis functions $\bar{t}_0(\mathbf{y}), \ldots, \bar{t}_4(\mathbf{y})$.

1. Make the array shape more circular by moving away from hexagonally shaped spreading functions.

2. Experiment with a less enormous array.

3. Examine the tradeoff between ripple and beamwidth in the design of spreading functions.

4. Improve the design process by jointly optimizing the 1D prototype filter and the 2D McClellan-transformed array factor.

Aim 1 is addressed through constructing a spreading function $\bar{t}_1(\mathbf{y}) = \bar{x}_{3D}(\mathbf{y})$ with impulses located on the 31 lattice points shown in Fig. 7. These points form three concentric hexagons, all centered on the origin, with sides of two, three, and four points, but with the six corner points of the outer hexagon removed. This is the least number of impulses that yields nonhexagonal array shapes while retaining twelvefold symmetry. The basis functions $\bar{t}_1(\mathbf{y}), \bar{t}_2(\mathbf{y}), \ldots$ and therefore of the array itself have dodecagonal (12-sided) outlines. They are hexagons in the spirit of Fig. 5 but with the six corners "filed down." The hope was that by not wasting elements on the corners where not really needed the 2D taper losses could be improved.

Aim 2 is addressed by setting $M = 10$, resulting in an array size equal to the number of impulses in $\bar{t}_{10}(\mathbf{y})$ or 2461 elements.

Aim 3 is addressed in the Fig. 10 experiment by carrying out two versions of the design. The spreading-function pairs $\bar{x}_{3D}(\mathbf{y}) \leftrightarrow X_{2D}(\mathbf{aS})$ of plot rows two and three are each optimized to have $X_{2D}(0) = 1$ and for $\mathbf{a}$ further from the origin than some stopband radius to have $X_{2D}(\mathbf{aS})$ contained in some interval $[-1, b]$ with bound b minimized. It's a shifted and scaled version of a classic lowpass-filter design problem, here with a single-point passband at the origin. The difference is that here the usual near-zero stopband is replaced by a region near -1 for which the stopband terminology we will use is at best appropriate by analogy.

There is a tradeoff between the stopband-radius design parameter and the optimal value of b

that results. The small radius of Fig. 10 row two yields a relatively high bound $b \approx \cos(2\pi{\times}0.1764)$ so that the stopband-ripple swings are larger. The high radius of row three yields a very low bound $b \approx \cos(2\pi \times 0.4666)$, making the stopband-ripple swings very small. In either case the obtained bounds b on $X_{\mathrm{2D}}(\mathbf{aS})$ imply bounds b', here 0.1764 and 0.4666 in the two cases, on frequency-warping function $X_{\mathrm{1D}}^{-1}(X_{\mathrm{2D}}(\mathbf{aS}))$, and the latter bounds are then used to set up the top-row optimization of the 1D prototype filter.

Aim 4, joint optimization of the prototype filter and the McClellan-transformed array factor, is accomplished by optimizing the common set of weights that determine them both using some specifications in 1D and some specifications in 2D.

- The prototype filter's DC gain $H(X_{\mathrm{1D}}(0))$, which is also the boresight array factor $H(X_{\mathrm{2D}}(0))$, must be at least unity or 0 dB,
- prototype filter $H(X_{\mathrm{1D}}(\bar{f}S))$ is limited to -30 dB stopband ripple for $0.08 \leq \bar{f}S \leq b'$,
- prototype filter $H(X_{\mathrm{1D}}(\bar{f}S))$ is limited to -50 dB stopband ripple for $b' \leq \bar{f}S \leq 0.5$, and
- the 2D taper loss of array factor $H(X_{\mathrm{2D}}(\mathbf{aS}))$ is minimized.

The idea is that the McClellan transformation guarantees that the ripple magnitudes are the same in 1D and 2D, but bounding that magnitude is easier in 1D because far fewer constraints are needed there. The taper loss that matters, however, is the 2D taper loss.

Joint 1D and 2D optimization would seem to be ideal, but embedded in that design is the constraining effect of the hand-picked spreading-function stopband radius. By determining the spreading function, that radius determines the basis set used to build the 2D array factor and therefore the subspace to which its design is restricted. Aim 3 is therefore key here: is it better for the 1D prototype to have a smaller outer stopband to get the narrowest main lobe possible? Or would a larger outer stopband and a slightly wider main lobe be preferable?

Comparing Fig. 10 rows two and three answers this quite clearly for the design of conventional pencil beams. A small spreading-function stopband radius is preferable, even though a large spreading-function ripple results. That large ripple requires a concomitantly low outer-stopband cutoff in the 1D prototype filter, which leads to a larger 1D main beam, but the latter penalty is more than overcome when the McClellan transformation maps that main beam into a much smaller 2D beam. Beamwidth is reduced in the favored approach by over 20%, and the width of the -30 dB inner sidelobe shelf is markedly reduced as well.

The optimized design of the bottom row of Fig. 10 is also twelvefold symmetric with -30 dB inner sidelobes, with -50 dB outer sidelobes, and with the transition between constraint regions occurring at 4° and 8° radii for comparability to the design of row two. Relative to that McClellan-transformed design, however, the optimized bottom-row design has a very slightly narrower beam, clearly lower average sidelobe levels, and 0.37 dB more SNR gain, and it achieves this with over 14% fewer elements, as it has elements only at the 2107 points of the $\lambda/\sqrt{3}$ spaced hexagonal lattice that lie at distance $\lambda\sqrt{579/3}$ or less from the origin. The taper is optimized to minimize noise gain subject to constraints on peak sidelobe levels.

3.3.4 Design example: small spreading function vs. large

If questions of array shape or beam spoiling, etc. are not at issue, is there any inherent performance advantage, in the beamshaping application, in a spreading function comprising a smaller or larger number of impulses? The design examples of Figure 11 address this question. The hexagonal supports of the spreading functions used in the designs of the second and third rows contain seven

and 37 points respectively. These are used with $M = 30$ and $M = 10$ respectively so that by starting with prototype filters of different lengths, the McClellan transformation yields a hexagonally shaped array of 2791 elements in each case in spite of the difference in spreading-function size. Each spreading function is optimally designed exactly as in the Fig. 10 designs discussed in Section 3.3.4 except that in each Figure 11 design the spreading-function stopband radius was set to $10°$. The very different spreading-function sizes then naturally led to very different ripple ranges $[-1, b]$ for spreading function $X_{\mathrm{2D}}(\mathbf{aS})$ in the two cases, and therefore to very different stopbands $[b', 0.5]$ for frequency-warping function $X_{\mathrm{1D}}^{-1}(X_{\mathrm{2D}}(\mathbf{aS}))$ as well. The b' values of 0.0817 and 0.2361 shown in Figure 11 for the second- and third-row designs then become the stopband cutoff frequencies used to optimize the associated prototype filters of the first row of Figure 11.

A closer look at the significance of the $10°$ spreading-function stopband radius is enlightening. That radius actually divides the plane into an inner region where the $[0, b']$ interval in 1D frequency for the prototype filter maps monotonically to radius in 2D and an outer region where there is no such monotonicity, where 1D frequency moves back and forth across the $[b', 0.5]$ stopband in various ways as 2D radius increases beyond $10°$. Because of that complexity in the outer region, it makes little sense to design for any particular detailed stopband behavior there, and these designs simply use an upper bound on array-factor magnitude there. The monotonicity in the inner region, however, makes it reasonable to attempt some shaping of the 2D stopband behavior there through shaping of the 1D prototype filter's stopband behavior in the associated interval.

In the Fig. 10 designs discussed earlier in Section 3.3.4, the inner sidelobes were set to a constant level of -30 dB, while the outer sidelobes were bounded to -50 dB and below. Here in Figure 11, however, only the innermost sidelobe is permitted to reach -30 dB, and in the first-row 1D prototypes the bounding level used to control the other inner sidelobes is ramped linearly down to the -50 dB outer-sidelobe level. The resulting decreasing heights of the inner sidelobe rings in the McClellan-transformed array factors is a generally desirable feature, and this experiment also satisfies some basic curiosity as to whether such a feature might reduce the penalty suffered by the McClellan approach in taper loss and SNR gain relative to optimized designs. (The answer: perhaps, but not dramatically.)

The bottom-row array factor was for the same hexagonal array and was directly optimized

- to lower bound its boresight magnitude at 0 dB,
- to upper bound its magnitude at -30 dB from $3.5°$ to $4.1°$ in radius,
- to upper bound its magnitude with a decibel-scale ramp extending from -30 dB at $4.1°$ radius to -50 dB at $10°$ radius,
- to upper bound its magnitude at -50 dB everywhere outside $10°$ radius, and
- to minimize taper loss.

The differences in taper loss among the three designs appears entirely accounted for by their differences in beamwidth. The taper loss is, as expected, best for the fully optimized design, by a substantial margin of 0.72 dB. That fully optimized design also had far lower sidelobe levels over well over half the hemisphere, though such levels are quite pointless because hardware weighting errors would prevent their realization in practice. Better would be to specify the optimization in a way that used the extra degrees of design freedom to more useful effect.

The two McClellan-transformed designs were surprisingly close in performance, differing in taper loss by only 0.14 dB. It was no surprise, however, that the slightly better of the two was the design based on the small spreading function, simply because that optimization worked with $M+1 = 31$ basis functions as opposed to 11.

3.3.5 Design example: a tilted-ellipse beamshape

Reddy and Hazra [15] used a McClellan transformation to create elliptical passbands or beamshapes with the ellipses oriented arbitrarily in the plane. They assumed a square sampling (element layout) lattice in the plane, however, and they considered only a very simple nine-sample spreading function. Let us see what we can do along these lines but for the hexagonal lattice and using optimization tools to favor high-performance beamshapes.

The design experiment presented in Fig. 12 investigates broadening the beam in one arbitrarily chosen direction only. To do this it combines the geometry of the Fig. 10 design, in particular

- the 31-point spreading-function support and the resulting dodecagonal array shape and
- an array sized by setting $M = 10$ to yield 2461 array elements,

with the basic structure of the Fig. 11 design, in particular

- inner-sidelobe heights that ramp down from -30 dB at a fixed 1D cutoff frequency of $\bar{f}S = 0.098$ to -50 dB at the outer-sidelobe bandedge b' that results from the spreading-function optimization as discussed in Section 3.3.3.

These features from prior designs are combined with one new aspect:

- the stopband-edge circle of $10°$ radius used to constrain previous spreading-function designs is here replaced with a stopband edge that is elliptical in the two dimensions of $\mathbf{aS}$. That ellipse
  - has a minor radius equal to the $10°$ radius of the circle used in earlier designs,
  - has a major radius double that—the doubling is in $\|\mathbf{aS}\|$ rather than angle so that the major radius is slightly larger than $20°$ in angle—and
  - the ellipse is placed in the $\mathbf{aS}$ plane with its semi-major axis at an angle to the horizontal (azimuth) axis that begins at zero in row two of Fig. 12 and that increments in $5°$ increments in successive rows of the figure.

Obtaining angles at $5°$ increments beyond $30°$ is a matter of taking one of the designs in the $[0°, 30°]$ interval and either rotating its spreading-function weights by a multiple of $60°$ or reflecting those weights through an axis at some multiple of $30°$ from horizontal. Establishing the behavior of the design over seven angles at $5°$ increments in the $[0°, 30°]$ interval therefore effectively establishes its behavior at 72 multiples of $5°$ spaced evenly around the circle.

The first row of Figure 12 presents the overlaid magnitude responses of the separately optimized 1D prototype filters of all seven designs, along with their stopband bounding lines. The seven prototypes are sufficiently similar that it appears entirely practical to choose one fixed prototype design to use at all angles.

Figure 12 breaks from this report's pattern of showing the entire hemisphere in the center column and instead covers only about $\pm 20°$ relative to boresight in each direction, primarily to simplify examination of the stopband-boundary ellipse, shown in red, and the inner-sidelobe region. As in earlier designs the outer sidelobes in the rest of the hexagonal period of array factor $X_{2D}(\mathbf{aS})$ simply comprise extended ripple ridges that peak at -50 dB as before. Those outer sidelobes move and change their ridge geometries with ellipse angle but are otherwise not terribly interesting.

The third column of Figure 12 shows the center portion of the main beam up very close, closer than for other designs, so that the shapes of the contours can be visually compared to the shapes of the stopband-boundary ellipses in the center column.

As the angle of the stopband-boundary ellipse is incremented, what can be seen is that even though the -3 dB main-beam contour is not explicitly shape controlled in the optimization and obtains its shape only indirectly from the stopband-boundary ellipse, that main-beam contour generally becomes elliptical and tilts right along with the stopband-boundary ellipse. However, that angle tracking is neither perfect nor does it maintain the $2:1$ axis ratio consistently. The 20° and 25° tilts in particular have main-beam ellipses that are visibly less eccentric. The designs for each of these two tilts, and to a lesser extent the design for the 15° tilt as well, also show changes in how the center-column stopband-boundary ellipse relates to the sidelobe ridges immediately around it. Somehow the stopband-boundary ellipse "loses traction" on the design a bit at these middle tilt angles.

Roughly 3 dB of the taper losses shown in the third column of course are the organic result of realizing the intended beam broadening by a factor of two in area and do not represent some sort of greatly increased design inferiority. The last two tilts, 25° and 30°, result in very different mainbeam ellipticities and correspondingly very different taper losses. This shape sensitivity makes these numbers much less useful here than in earlier, pencil-beam designs. They are included here for completeness.

The take-away lesson from the designs for this series of seven tilts is that the general idea of simply changing the spreading-function weights to effect beam broadening, whether in 1D as here or in 2D as in the Fig. 9 design, appears to be quite viable. Further, it appears that given a particular desired elliptical beam shape, a single prototype design is likely viable for all ellipse tilt angles. The detailed design procedure used here is probably not quite adequate in the elliptical-beam case, but it should not be particularly difficult to modify the optimization formulation to better preserve the beam shape as the ellipse tilt is varied, but we can safely leave demonstration of that to the relevant application design.

In a real application making the tilt a continuously variable parameter is apt to be an attractive option, and the methods recently proposed by Yeung and Chan [16] and by Shyu et al. [17] to combine the McClellan-transformation approach to 2D filters with the Farrow structure, or variations thereon, for realizing variable FIR filters should be examined closely.

4 Conclusions

Broadly speaking, the McClellan transformation seems well suited to quick design of 2D array tapers of medium to large size. In this report most of the 1D prototype filters transformed are designed using optimization approaches, but the real purpose of such optimization here is to learn something about obtainable limits to performance. In practice the strength of McClellan transformation is that it permits easy creation of large 2D tapers by using ordinary design tools to design a 1D taper and using ordinary 2D-filter design tools to design a very small 2D spreading function. Reasonable results can even be obtained using very small spreading functions designed by hand and simple Taylor or Remez-exchange design of prototypes.

This report's design experiments reveal only one fundamentally important limitation of the McClellan-transformation approach to 2D taper design: taper loss. Every combination of spreading-function and prototype-filter design criteria yielded a taper loss at least 0.7 dB greater than obtained for the same size array through direct optimization of the taper.[2] This is a significant limitation and may limit the use of the McClellan transformation in final designs to those rare settings in which the approach's enabling of real-time changes to the 2D taper through simple modifications to

[2] In the Fig. 10 design of Section 3.3.3, SNR performance only 0.37 dB worse than the fully optimized reference design was achieved, but the latter design used some 14.4% fewer elements, which amounts to another 0.67 dB.

the spreading function is of some special benefit. Also, in preliminary design work in which optimization code capable of handling large 2D problems is unavailable, the McClellan-transformation approach might reasonably be used to get early estimates of pattern shape, perhaps including in SNR or beamwidth estimates some rough fudge factor to compensate for further improvements likely to be obtained through eventual full-out optimization.

Mersereau and Mecklenbräuker [5] assert that the McClellan approach can save an enormous number of coefficient multiplies in implementation. This claim needs to be interpreted carefully in the array context, however. Basically, the defining recursion for the Chebyshev polynomials leads to a structurally similar recursion at the heart of the McClellan transformation. The latter recursion[3] can take any of three related forms, each with an apparently natural application to arrays.

Receive-array implementation: An FIR-filter recursion in Chebyshev direct form, as presented in Fig. 1, can incorporate the McClellan recursion into the real-time computation of a receive-array beam sum. Each prototype-filter coefficient is used in multiplication just once.

Transmit-array implementation: An FIR-filter recursion in reversed Chebyshev form, as presented in Fig. 2, can incorporate the McClellan recursion into the real-time computation of element-drive signals. Each prototype-filter coefficient is used in multiplication just once.

Offline design, for transmit or receive: A McClellan recursion can be used to compute the taper weights offline, and those weights can then be used conventionally to compute receive beam sum or transmit element drives in real time.

The relatively obvious third approach treats the McClellan transformation as an easily computed design procedure unrelated to issues of implementation. The first two approaches build the McClellan recursion right into the implementation so that it operates on actual signals and not just on taper weights, so the taper weights are never explicitly computed at all. This saves many coefficient multiplies if one counts only multiplications by the weights of the prototype filter. In the first two approaches above, however, the 2D spatial convolutions with the spreading function, denoted by x in Figs. 1 and 2 but by $\bar{x}_{3D}(\mathbf{y})$ later in array-specific discussion, dominate the required implementation. Some careful application-specific operation counting would seem necessary before pronouncing on any implementation savings to be had.

References

[1] D. McNamara and E. Botha, "Transformation-based synthesis technique for planar arrays with contoured beams," *Electronics Letters*, vol. 27, no. 17, pp. 1502–1504, Aug. 1991, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=132772&k2dockey=132772@ieejrns.

[2] J. McClellan and D. Chan, "A 2-D FIR filter structure derived from the Chebyshev recursion," *IEEE Trans. Circuits and Systems*, vol. 24, no. 7, pp. 372–378, July 1977. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=23470&arnumber=1084354

[3] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. Englewood Cliffs NJ, USA: Prentice-Hall, 1984.

[4] R. Mersereau, W. Mecklenbräuker, and T. Quatieri Jr., "McClellan transformations for two-dimensional digital filtering—Part I: Design," *IEEE Trans. Circuits and Systems*,

[3] It is odd to associate the term "recursion" with FIR filters, but here the recursion is in the node number, not the sample index. For example, the signal at each node of a delay line is computed from the signal at the previous node.

vol. 23, no. 7, pp. 405–414, Jul 1976, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1084236&k2dockey=1084236@ieeejrns.

[5] W. F. G. Mecklenbräuker and R. Mersereau, "McClellan transformations for two-dimensional digital filtering—Part II: Implementation," *IEEE Trans. Circuits and Systems*, vol. 23, no. 7, pp. 414–422, Jul 1976, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1084237&k2dockey=1084237@ieeejrns.

[6] J. O. Coleman, "Three-phase sample timing on a wideband triangular array of 4/3 the usual density reduces the Nyquist rate for far-field signals by two thirds," in *38th IEEE Asilomar Conf. on Signals, Systems and Computers*, vol. 1, Nov. 2004, pp. 284–288, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1399137&k2dockey=1399137@ieeecnfs.

[7] G. Mollova and W. Mecklenbräuker, "A design method for 3-D FIR cone-shaped filters based on the McClellan transformation," *IEEE Trans. Signal Processing*, vol. 57, no. 2, pp. 551–564, Feb. 2009, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=4667658&k2dockey=4667658@ieeejrns.

[8] B. Lien and G. Tang, "Reversed Chebyshev implementation of McClellan transform and its roundoff error," *IEEE Trans. Acoustics, Speech and Signal Proc.*, vol. 35, no. 10, pp. 1435–1439, Oct. 1987, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1165045&k2dockey=1165045@ieeejrns.

[9] D. Scholnik and J. Coleman, "Optimal array-pattern synthesis for wideband digital transmit arrays," *IEEE J. Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 660–677, Dec. 2007, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=4407773&k2dockey=4407773@ieeejrns.

[10] J. O. Coleman, D. P. Scholnik, and J. J. Brandriss, "A specification language for the optimal design of exotic FIR filters with second-order cone programs," in *IEEE Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2002, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1197203&k2dockey=1197203@ieeecnfs.

[11] J. F. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653 (versions 1.02/1.03), 1999, updated for version 1.05 online in 2001, http://www.optimization-online.org/DB_HTML/2001/10/395.html.

[12] I. Pólik and T. Terlaky. SeDuMi 1.21. Cor@l Lab: Computational Optimization Research at Lehigh. Note that Jos Sturm developed the original SeDuMi code. http://sedumi.ie.lehigh.edu/.

[13] J. O. Coleman, K. R. McPhail, P. E. Cahill, and D. P. Scholnik, "Efficient subarray realization through layering," in *Antenna Applications Symposium:* http://www.ecs.umass.edu/ece/allerton/., Monticello IL, USA, Sept. 21–23, 2005.

[14] K. R. McPhail, J. O. Coleman, and D. P. Scholnik, "Experiments in weight design for a sombrero array pattern," Naval Research Laboratory, NRL Memo Report 9232, sometime in early 2010.

[15] M. Reddy and S. Hazra, "Design of elliptically symmetric two-dimensional FIR filters with arbitrary orientation," *Electronics Letters*, vol. 23, no. 18, pp. 964–966, 27 Aug, 1987, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=4258746&k2dockey=4258746@ieejrns.

[16] K. Yeung and S. Chan, "Design and implementation of multiplier-less tunable 2-D FIR filters using McClellan transformation," in *IEEE International Symp. on Circuits and Systems (ISCAS 2002)*, vol. 5, 2002, pp. V761–V764, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=1010815&k2dockey=1010815@ieeecnfs.

[17] J.-J. Shyu, S.-C. Pei, and Y.-D. Huang, "Design of variable two-dimensional FIR digital filters by McClellan transformation," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 56, no. 3, pp. 574–582, Mar. 2009, http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=4663646&k2dockey=4663646@ieeejrns.

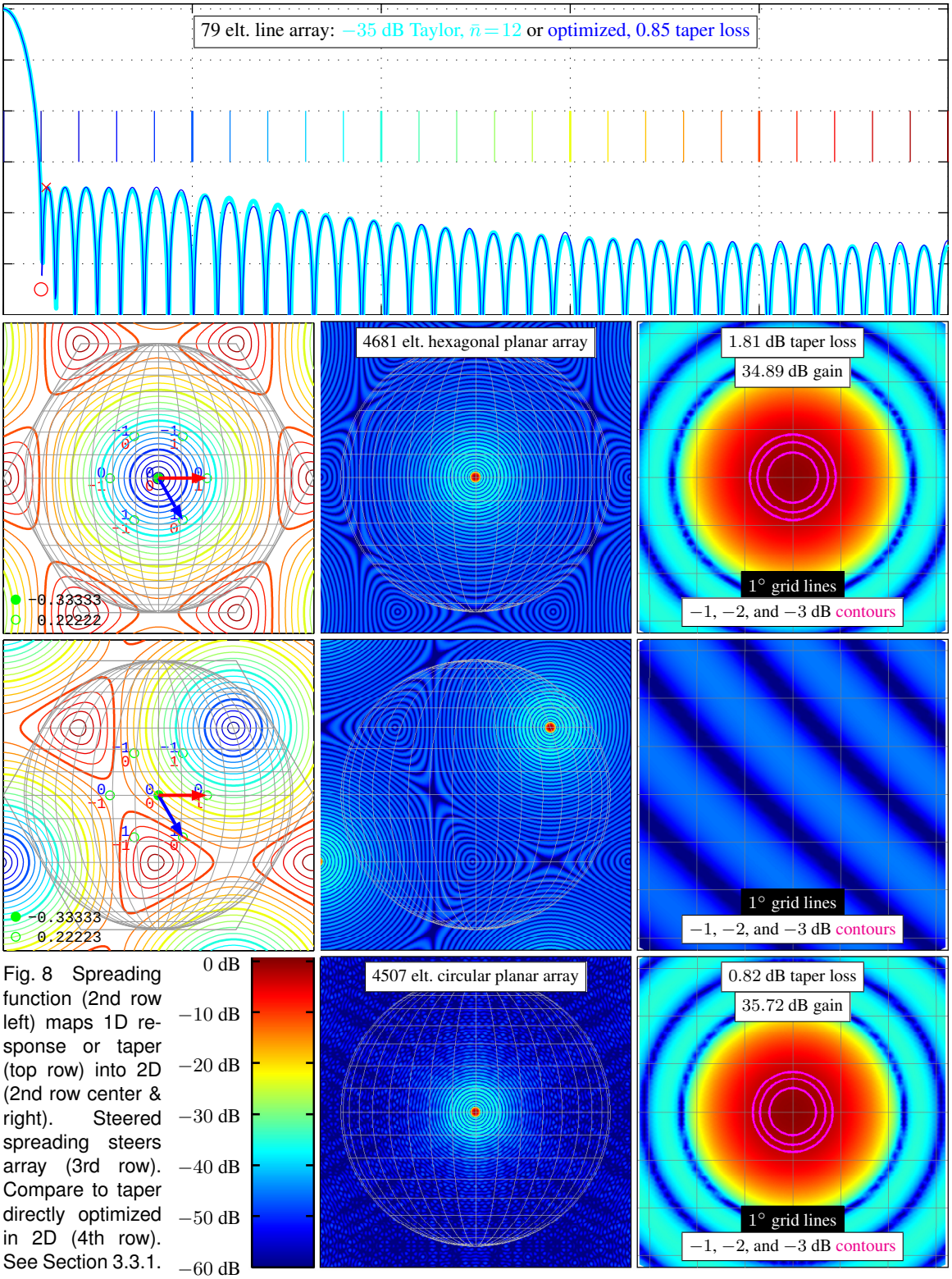


Fig. 8 Spreading function (2nd row left) maps 1D response or taper (top row) into 2D (2nd row center & right). Steered spreading steers array (3rd row). Compare to taper directly optimized in 2D (4th row). See Section 3.3.1.

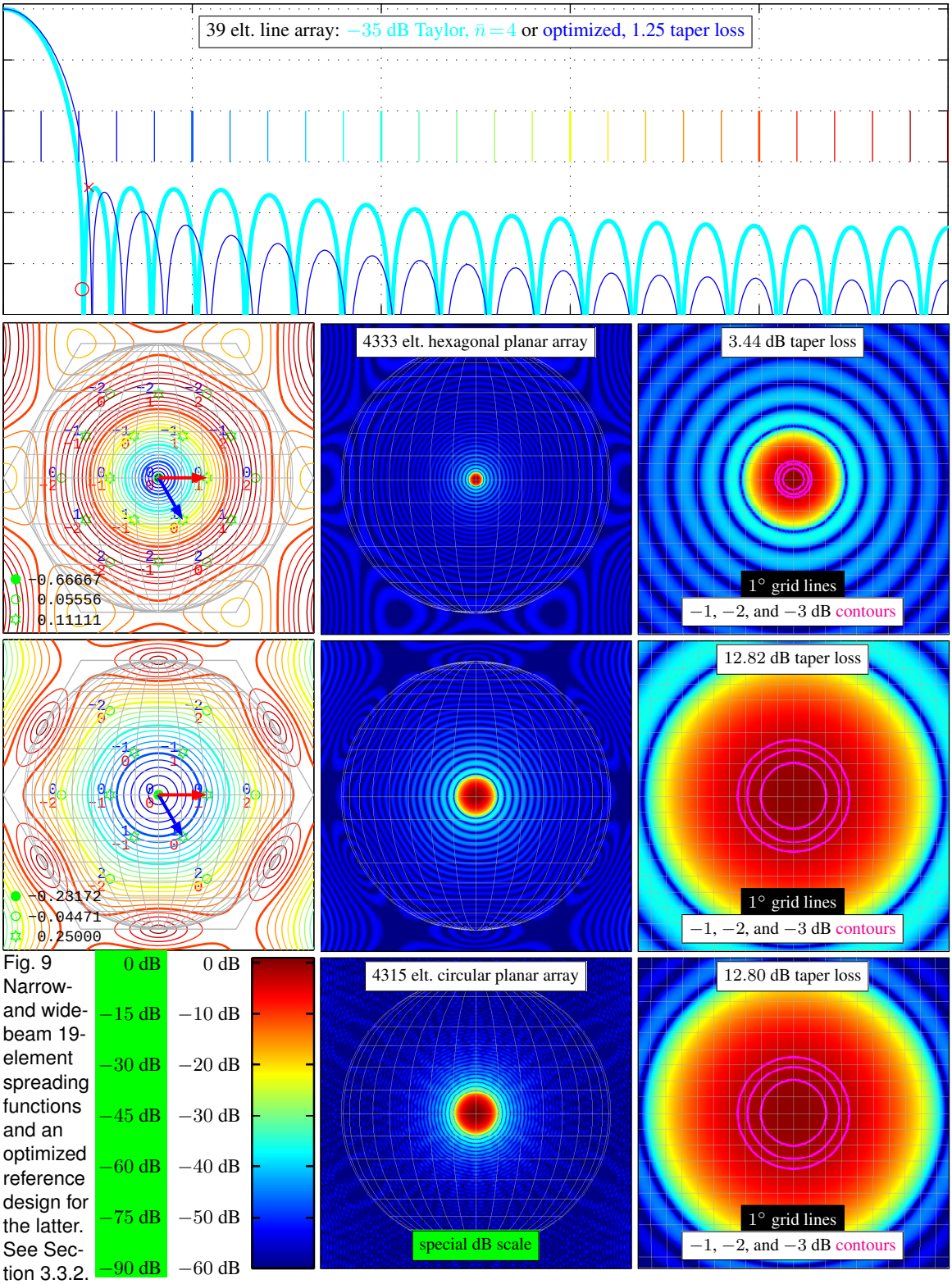


Fig. 9 Narrow- and wide-beam 19-element spreading functions and an optimized reference design for the latter. See Section 3.3.2.

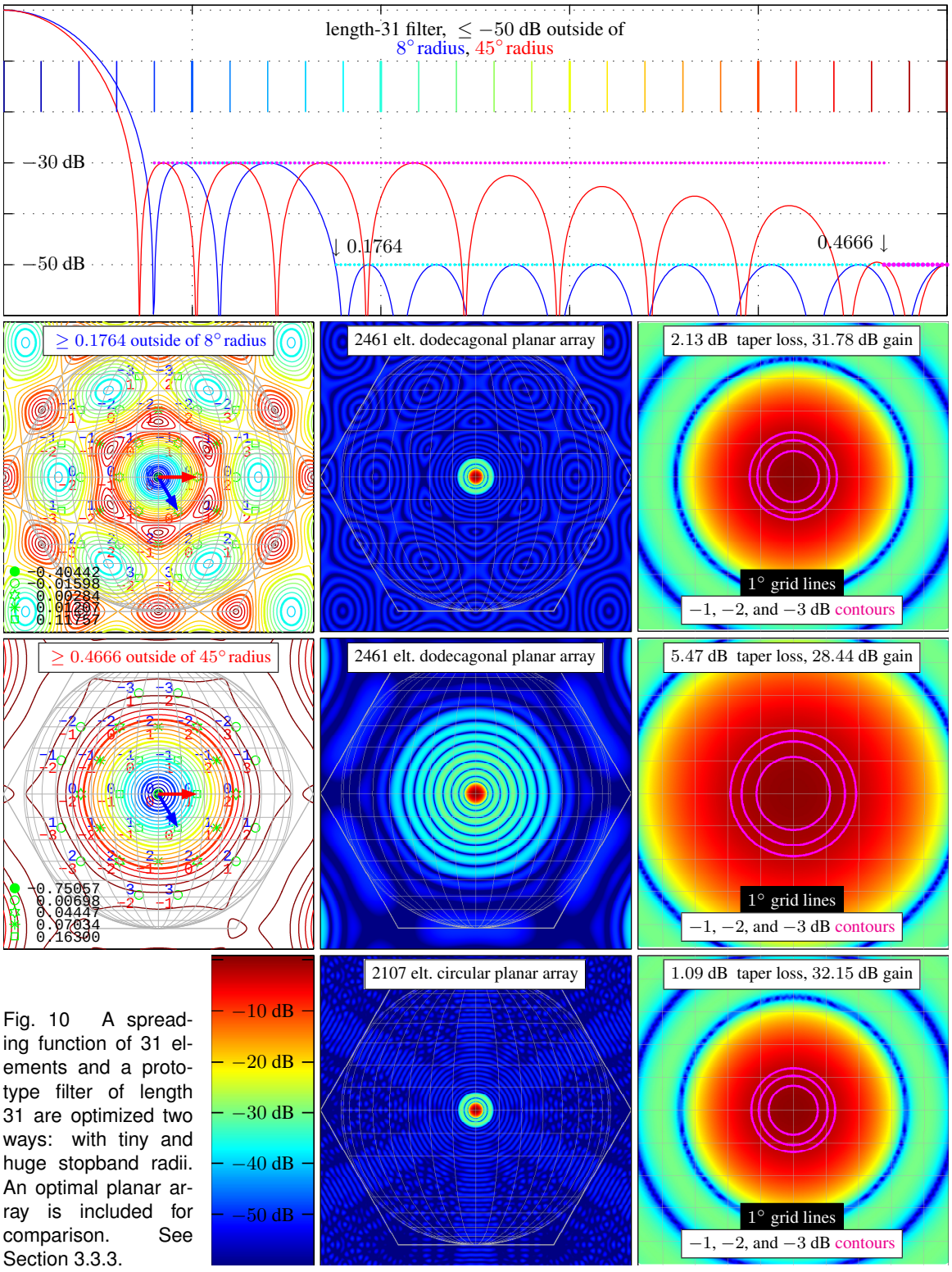


Fig. 10 A spreading function of 31 elements and a prototype filter of length 31 are optimized two ways: with tiny and huge stopband radii. An optimal planar array is included for comparison. See Section 3.3.3.

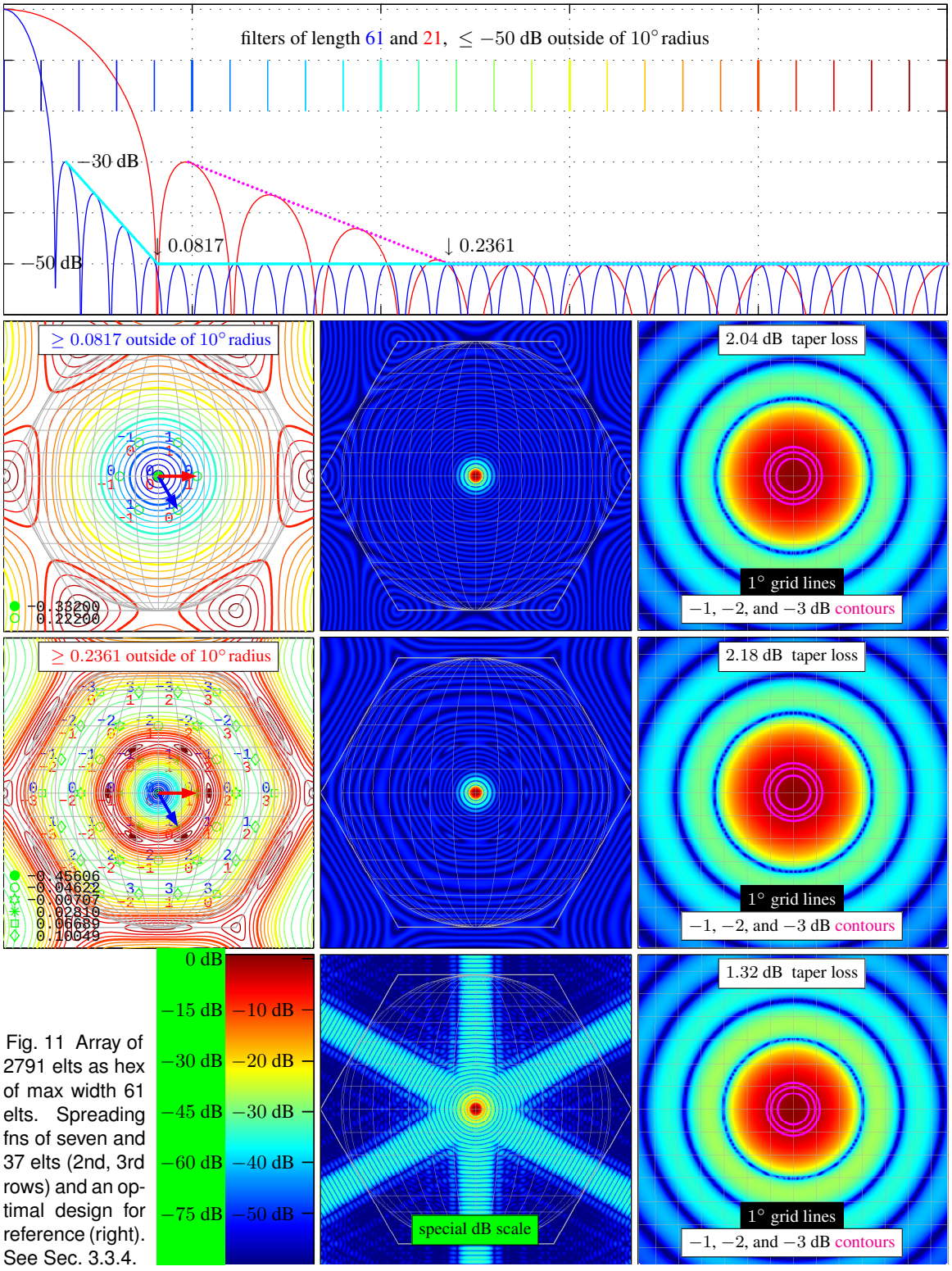


Fig. 11 Array of 2791 elts as hex of max width 61 elts. Spreading fns of seven and 37 elts (2nd, 3rd rows) and an optimal design for reference (right). See Sec. 3.3.4.

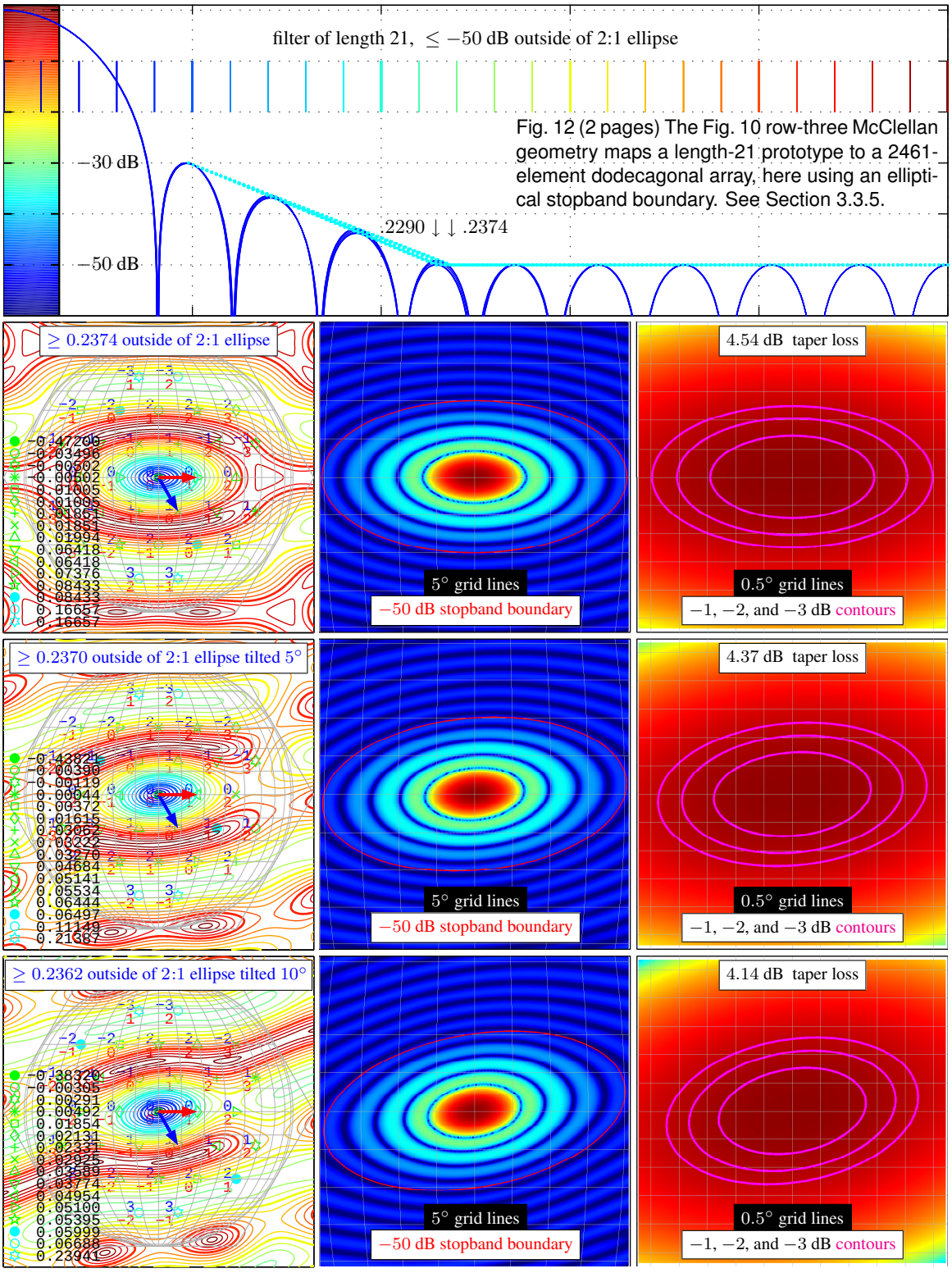
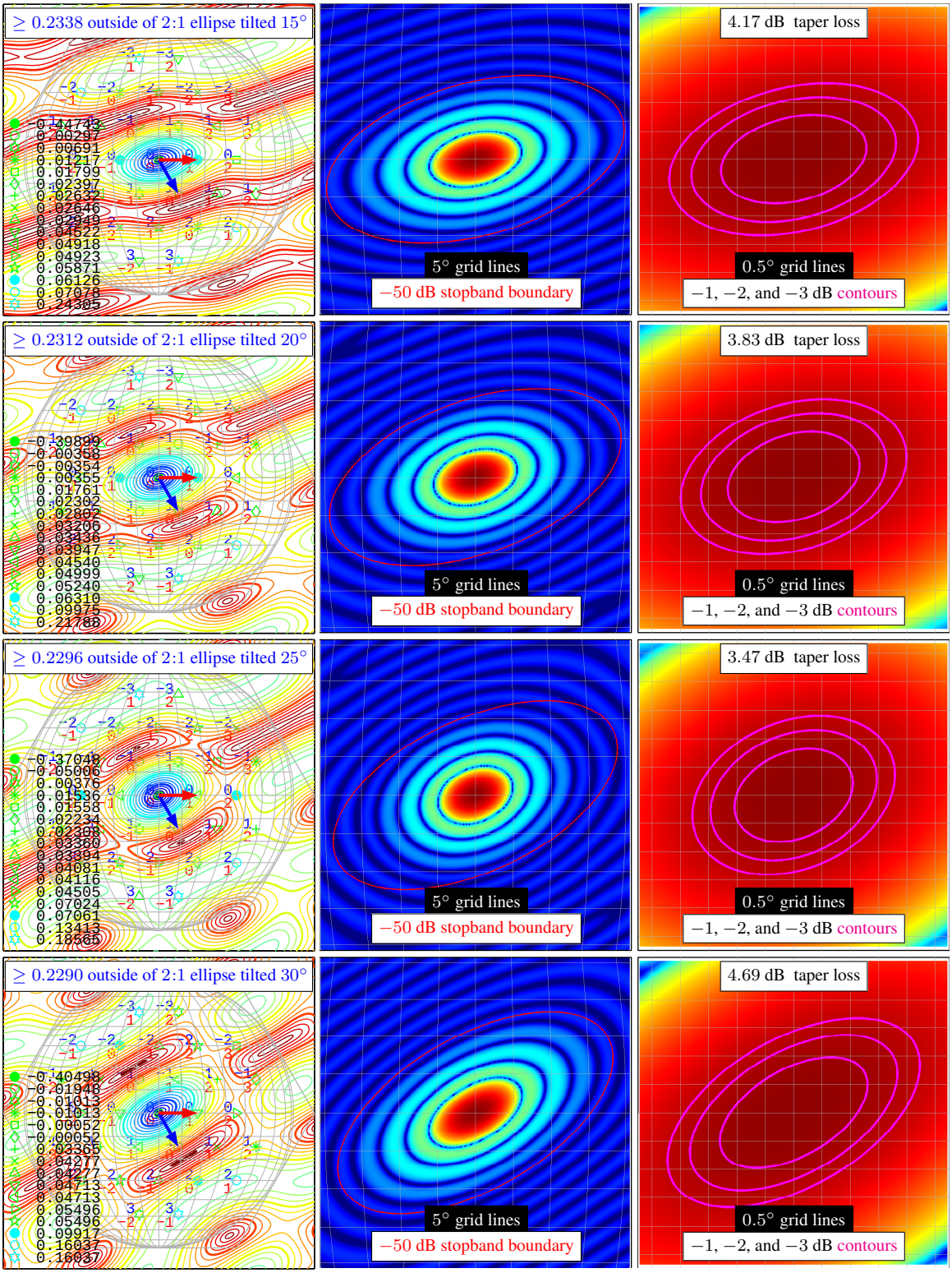


Fig. 12 (2 pages) The Fig. 10 row-three McClellan geometry maps a length-21 prototype to a 2461-element dodecagonal array, here using an elliptical stopband boundary. See Section 3.3.5.


≥ 0.2338 outside of 2:1 ellipse tilted 15°
5° grid lines
−50 dB stopband boundary
4.17 dB taper loss
0.5° grid lines
−1, −2, and −3 dB contours
≥ 0.2312 outside of 2:1 ellipse tilted 20°
5° grid lines
−50 dB stopband boundary
3.83 dB taper loss
0.5° grid lines
−1, −2, and −3 dB contours
≥ 0.2296 outside of 2:1 ellipse tilted 25°
5° grid lines
−50 dB stopband boundary
3.47 dB taper loss
0.5° grid lines
−1, −2, and −3 dB contours
≥ 0.2290 outside of 2:1 ellipse tilted 30°
5° grid lines
−50 dB stopband boundary
4.69 dB taper loss
0.5° grid lines
−1, −2, and −3 dB contours