

Array Design Methods

20.1 Array Design Methods

As we mentioned in Sec. 19.4, the array design problem is essentially equivalent to the problem of designing FIR digital filters in DSP. Following this equivalence, we discuss several array design methods, such as:

1. Schelkunoff's zero placement method
2. Fourier series method with windowing
3. Woodward-Lawson frequency-sampling design
4. Narrow-beam low-sidelobe design methods
5. Multi-beam array design

Next, we establish some common notation. One-dimensional equally-spaced arrays are usually considered *symmetrically* with respect to the origin of the array axis. This requires a slight redefinition of the array factor in the case of *even* number of array elements. Consider an array of N elements at locations x_m along the x -axis with element spacing d . The array factor will be:

$$A(\phi) = \sum_m a_m e^{jk_x x_m} = \sum_m a_m e^{jk x_m \cos \phi}$$

where $k_x = k \cos \phi$ (for polar angle $\theta = \pi/2$.) If N is odd, say $N = 2M + 1$, we can define the element locations x_m symmetrically as:

$$x_m = md, \quad m = 0, \pm 1, \pm 2, \dots, \pm M$$

This was the definition we used in Sec. 19.4. The array factor can be written then as a discrete-space Fourier transform or as a spatial z -transform:

$$\begin{aligned} A(\psi) &= \sum_{m=-M}^M a_m e^{jm\psi} = a_0 + \sum_{m=1}^M [a_m e^{jm\psi} + a_{-m} e^{-jm\psi}] \\ A(z) &= \sum_{m=-M}^M a_m z^m = a_0 + \sum_{m=1}^M [a_m z^m + a_{-m} z^{-m}] \end{aligned} \quad (20.1.1)$$

where $\psi = k_x d = kd \cos \phi$ and $z = e^{j\psi}$. On the other hand, if N is even, say $N = 2M$, in order to have symmetry with respect to the origin, we must place the elements at the *half-integer* locations:

$$x_{\pm m} = \pm \left(md - \frac{d}{2} \right) = \pm \left(m - \frac{1}{2} \right) d, \quad m = 1, 2, \dots, M$$

The array factor will be now:

$$\begin{aligned} A(\psi) &= \sum_{m=1}^M \left[a_m e^{j(m-1/2)\psi} + a_{-m} e^{-j(m-1/2)\psi} \right] \\ A(z) &= \sum_{m=1}^M \left[a_m z^{m-1/2} + a_{-m} z^{-(m-1/2)} \right] \end{aligned} \quad (20.1.2)$$

In particular, if the array weights a_m are symmetric with respect to the origin, $a_m = a_{-m}$, as they are in most design methods, then the array factor can be simplified into the cosine forms:

$$\begin{aligned} A(\psi) &= a_0 + 2 \sum_{m=1}^M a_m \cos(m\psi), \quad (N = 2M + 1) \\ A(\psi) &= 2 \sum_{m=1}^M a_m \cos((m-1/2)\psi), \quad (N = 2M) \end{aligned} \quad (20.1.3)$$

In *both* the odd and even cases, Eqs. (20.1.1) and (20.1.2) can be expressed as the left-shifted version of a right-sided z-transform:

$$A(z) = z^{-(N-1)/2} \tilde{A}(z) = z^{-(N-1)/2} \sum_{n=0}^{N-1} \tilde{a}_n z^n \quad (20.1.4)$$

where $\mathbf{a} = [\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_{N-1}]$ is the vector of array weights reindexed to be right-sided. In terms of the original symmetric weights, we have:

$$\begin{aligned} [\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_{N-1}] &= [a_{-M}, \dots, a_{-1}, a_0, a_1, \dots, a_M], \quad (N = 2M + 1) \\ [\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_{N-1}] &= [a_{-M}, \dots, a_{-1}, a_1, \dots, a_M], \quad (N = 2M) \end{aligned} \quad (20.1.5)$$

In time-domain DSP, a factor of z represents a time-advance or left shift. But in the spatial domain, a left shift is represented by z^{-1} because of the opposite sign convention in the definition of the z-transform. Thus, the factor $z^{-(N-1)/2}$ represents a left shift by a distance $(N-1)d/2$, which places the middle of the right-sided array at the origin. For instance, see Examples 19.3.1 and 19.3.2.

The corresponding array factors in ψ -space are related in a similar fashion. Setting $z = e^{j\psi}$, we have:

$$A(\psi) = e^{-j\psi(N-1)/2} \tilde{A}(\psi) = e^{-j\psi(N-1)/2} \sum_{n=0}^{N-1} \tilde{a}_n e^{jn\psi} \quad (20.1.6)$$

Working with $\tilde{A}(\psi)$ is more convenient for programming purposes, as it can be computed by an ordinary DTFT routine, such as that in Ref. [48], $\tilde{A}(\psi) = \text{dtft}(\mathbf{a}, -\psi)$. The phase factor $e^{-j\psi(N-1)/2}$ does not affect the power gain of the array; indeed, we have $|A(\psi)|^2 = |\tilde{A}(\psi)|^2 = |\text{dtft}(\mathbf{a}, -\psi)|^2$.

Some differences arise also for steered array factors. Given a steering phase $\psi_0 = kd \cos \phi_0$, we define the steered array factor as $A'(\psi) = A(\psi - \psi_0)$. Then, we have:

$$A'(\psi) = A(\psi - \psi_0) = e^{-j(\psi - \psi_0)(N-1)/2} \tilde{A}(\psi - \psi_0) = e^{-j\psi(N-1)/2} \tilde{A}'(\psi)$$

It follows that the steered version of $\tilde{A}(\psi)$ will be:

$$\tilde{A}'(\psi) = e^{j\psi_0(N-1)/2} \tilde{A}(\psi - \psi_0) \quad (20.1.7)$$

which implies for the weights:

$$\tilde{a}'_n = \tilde{a}_n e^{-j\psi_0(n-(N-1)/2)}, \quad n = 0, 1, \dots, N-1 \quad (20.1.8)$$

This simply means that the progressive phase is measured with respect to the middle of the array. Again, the common phase factor $e^{j\psi_0(N-1)/2}$ is usually unimportant. One case where it is important is the case of *multiple* beams steered towards different angles; these are discussed in Sec. 20.14. In the *symmetric* notation, the steered weights are as follows:

$$\begin{aligned} a'_m &= a_m e^{-jm\psi_0}, \quad m = 0, \pm 1, \pm 2, \dots, \pm M, \quad (N = 2M + 1) \\ a'_{\pm m} &= a_{\pm m} e^{\mp j(m-1/2)\psi_0}, \quad m = 1, 2, \dots, M, \quad (N = 2M) \end{aligned} \quad (20.1.9)$$

The MATLAB functions `scan` and `steer` perform the desired progressive phasing of the weights according to Eq. (20.1.8). Their usage is as follows:

```
ascan = scan(a, psi0);           % scan array with given scanning phase  $\psi_0$ 
asteer = steer(d, a, phi0);      % steer array towards given angle  $\phi_0$ 
```

Example 20.1.1: For the cases $N = 7$ and $N = 6$, we have $M = 3$. The symmetric and right-sided array weights will be related as follows:

$$\begin{aligned} \mathbf{a} &= [\tilde{a}_0, \tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \tilde{a}_4, \tilde{a}_5, \tilde{a}_6] = [a_{-3}, a_{-2}, a_{-1}, a_0, a_1, a_2, a_3] \\ \mathbf{a} &= [\tilde{a}_0, \tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \tilde{a}_4, \tilde{a}_5] = [a_{-3}, a_{-2}, a_{-1}, a_1, a_2, a_3] \end{aligned}$$

For $N = 7$ we have $(N-1)/2 = 3$, and for $N = 6$, $(N-1)/2 = 5/2$. Thus, the array locations along the x-axis will be:

$$\begin{aligned} x_m &= \{-3d, -2d, -d, 0, d, 2d, 3d\} \\ x_m &= \left\{ -\frac{5}{2}d, -\frac{3}{2}d, -\frac{1}{2}d, \frac{1}{2}d, \frac{3}{2}d, \frac{5}{2}d \right\} \end{aligned}$$

Eq. (20.1.4) reads as follows in the two cases:

$$\begin{aligned} A(z) &= a_{-3}z^{-3} + a_{-2}z^{-2} + a_{-1}z^{-1} + a_0 + a_1z + a_2z^2 + a_3z^3 \\ &= z^{-3} [a_{-3} + a_{-2}z + a_{-1}z^2 + a_0z^3 + a_1z^4 + a_2z^5 + a_3z^6] = z^{-3} \tilde{A}(z) \\ A(z) &= a_{-3}z^{-5/2} + a_{-2}z^{-3/2} + a_{-1}z^{-1/2} + a_1z^{1/2} + a_2z^{3/2} + a_3z^{5/2} \\ &= z^{-5/2} [a_{-3} + a_{-2}z + a_{-1}z^2 + a_1z^3 + a_2z^4 + a_3z^5] = z^{-5/2} \tilde{A}(z) \end{aligned}$$

If the arrays are steered, the weights pick up the progressive phases:

$$\begin{aligned} & [a_{-3}e^{j3\psi_0}, a_{-2}e^{j2\psi_0}, a_{-1}e^{j\psi_0}, a_0, a_1e^{-j\psi_0}, a_2e^{-j2\psi_0}, a_3e^{-j3\psi_0}] \\ &= e^{j3\psi_0} [a_{-3}, a_{-2}e^{-j\psi_0}, a_{-1}e^{-2j\psi_0}, a_0e^{-3j\psi_0}, a_1e^{-4j\psi_0}, a_2e^{-5j\psi_0}, a_3e^{-6j\psi_0}] \\ & [a_{-3}e^{j5\psi_0/2}, a_{-2}e^{j3\psi_0/2}, a_{-1}e^{j\psi_0/2}, a_1e^{-j\psi_0/2}, a_2e^{-j3\psi_0/2}, a_3e^{-j5\psi_0/2}] \\ &= e^{j5\psi_0/2} [a_{-3}, a_{-2}e^{-j\psi_0}, a_{-1}e^{-2j\psi_0}, a_1e^{-3j\psi_0}, a_2e^{-j4\psi_0}, a_3e^{-j5\psi_0}] \end{aligned}$$

where $\psi_0 = kd \cos \phi_0$ is the steering phase. \square

Example 20.1.2: The uniform array of Sec. 19.7, was defined as a right-sided array. In the present notation, the weights and array factor are:

$$\mathbf{a} = [\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_{N-1}] = \frac{1}{N} [1, 1, \dots, 1], \quad \tilde{A}(z) = \frac{1}{N} \frac{z^N - 1}{z - 1}$$

Using Eq. (20.1.4), the corresponding symmetric array factor will be:

$$A(z) = z^{-(N-1)/2} \tilde{A}(z) = z^{-(N-1)/2} \frac{1}{N} \frac{z^N - 1}{z - 1} = \frac{1}{N} \frac{z^{N/2} - z^{-N/2}}{z^{1/2} - z^{-1/2}}$$

Setting $z = e^{j\psi}$, we obtain

$$A(\psi) = \frac{\sin\left(\frac{N\psi}{2}\right)}{N \sin\left(\frac{\psi}{2}\right)} \quad (20.1.10)$$

which also follows from Eqs. (19.7.3) and (20.1.6). \square

20.2 Schelkunoff's Zero Placement Method

The array factor of an N -element array is a polynomial of degree $N - 1$ and therefore it has $N - 1$ zeros:

$$\tilde{A}(z) = \sum_{n=0}^{N-1} \tilde{a}_n z^n = (z - z_1)(z - z_2) \cdots (z - z_{N-1}) \tilde{a}_{N-1} \quad (20.2.1)$$

By proper placement of the zeros on the z -plane, a desired array factor can be designed. Schelkunoff's paper of more than 45 years ago [1067] discusses this and the Fourier series methods.

As an example consider the uniform array that has zeros equally spaced around the unit circle at the N -th roots of unity, that is, at $z_i = e^{j\psi_i}$, where $\psi_i = 2\pi i/N$, $i = 1, 2, \dots, N - 1$. The index $i = 0$ is excluded as $z = 1$ or $\psi = 0$ corresponds to the mainlobe peak of the array. Depending on the element spacing d , it is possible that not all of these zeros lie within the visible region and, therefore, they may not correspond to actual nulls in the angular pattern. This happens when $d < \lambda/2$ for a broadside array, which has a visible region that covers less than the full unit circle, $\psi_{\text{vis}} = 2kd < 2\pi$.

20.2. Schelkunoff's Zero Placement Method

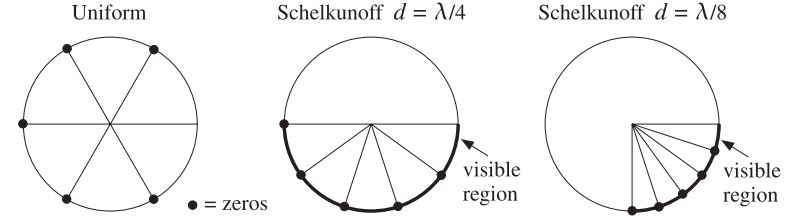


Fig. 20.2.1 Endfire array zeros and visible regions for $N = 6$, and $d = \lambda/4$ and $d = \lambda/8$.

Schelkunoff's design idea was to place all $N - 1$ zeros of the array within the visible region, for example, by equally spacing them within it. Fig. 20.2.1 shows the visible regions and array zeros for a six-element *endfire* array with element spacings $d = \lambda/4$ and $d = \lambda/8$.

The visible region is determined by Eq. (19.9.5). For an endfire ($\phi_0 = 0$) array with $d = \lambda/4$ or $kd = \pi/2$, the steered wavenumber will be $\psi' = kd(\cos \phi - \cos \phi_0) = (\cos \phi - 1)\pi/2$ and the corresponding visible region, $-\pi \leq \psi' \leq 0$. Similarly, when $d = \lambda/8$ or $kd = \pi/4$, we have $\psi' = (\cos \phi - 1)\pi/4$ and visible region, $-\pi/2 \leq \psi' \leq 0$.

The uniform array has five zeros. When $d = \lambda/4$, only three of them lie in the visible region, and when $d = \lambda/8$ only one of them does. By contrast Schelkunoff's design method places all five zeros within the visible regions.

Fig. 20.2.2 shows the gains of the two cases and compares them to the gains of the corresponding uniform array. The presence of more zeros in the visible regions results in a narrower mainlobe and smaller sidelobes.

The angular nulls corresponding to the zeros that lie in the visible region may be observed in these graphs for both the uniform and Schelkunoff designs.

Because the visible region is in both cases $-2kd \leq \psi' \leq 0$, the five zeros are chosen as $z_i = e^{j\psi_i}$, where $\psi_i = -2kdi/5$, $i = 1, 2, \dots, 5$. The array weights can be obtained by expanding the zero factors of Eq. (20.2.1). The following MATLAB statements will perform and plot the design:

```
d=1/4; kd=2*pi*d;
i = 1:5;
psi = -2*kd*i/5;
zi = exp(j*psi);
a = flipplr(poly(zi));
a = steer(d, a, 0);
[g, ph] = array(d, a, 400);
dbz(ph, g, 45, 40);
```

The function `poly` computes the expansion coefficients. But because it lists them from the higher coefficient to the lowest one, that is, from z^{N-1} to z^0 , it is necessary to reverse the vector by `flipplr`. When the weight vector is symmetric with respect to its middle, such reversal is not necessary.

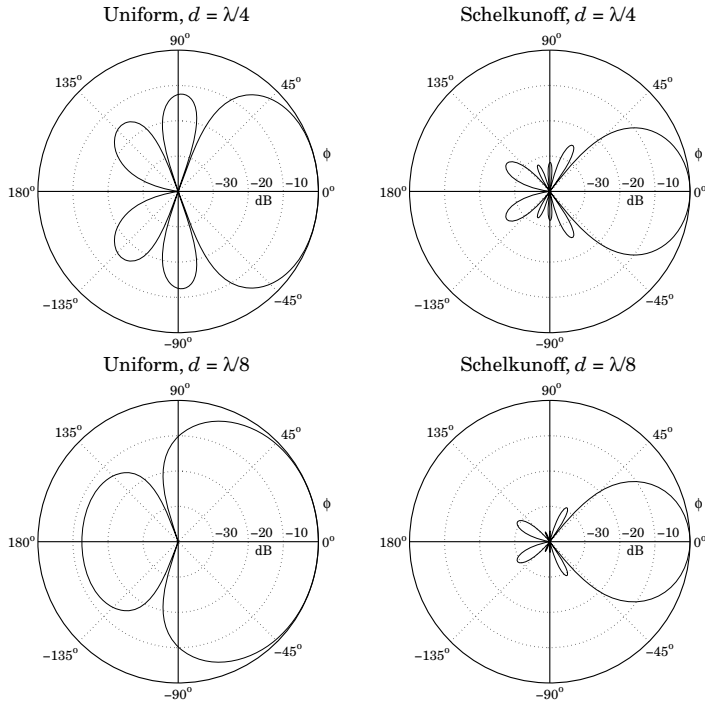


Fig. 20.2.2 Gain of six-element endfire array with $d = \lambda/4$ and $d = \lambda/8$.

20.3 Fourier Series Method with Windowing

The Fourier series design method is identical to the same method in DSP for designing FIR digital filters [47,48]. The method is based on the inverse discrete-space Fourier transforms of the array factor.

Eqs. (20.1.1) and (20.1.2) may be thought of as the truncated or *windowed* versions of the corresponding infinite Fourier series. Assuming an infinite and convergent series, we have for the “odd” case:

$$A(\psi) = a_0 + \sum_{m=1}^{\infty} [a_m e^{jm\psi} + a_{-m} e^{-jm\psi}] \quad (20.3.1)$$

Then, the corresponding inverse transform will be:

$$a_m = \frac{1}{2\pi} \int_{-\pi}^{\pi} A(\psi) e^{-jm\psi} d\psi, \quad m = 0, \pm 1, \pm 2, \dots \quad (20.3.2)$$

20.4 Sector Beam Array Design

Similarly, in the “even” case we have:

$$A(\psi) = \sum_{m=1}^{\infty} [a_m e^{j(m-1/2)\psi} + a_{-m} e^{-j(m-1/2)\psi}] \quad (20.3.3)$$

with inverse transform:

$$a_{\pm m} = \frac{1}{2\pi} \int_{-\pi}^{\pi} A(\psi) e^{\mp j(m-1/2)\psi} d\psi, \quad m = 1, 2, \dots \quad (20.3.4)$$

In general, a desired array factor requires an infinite number of coefficients a_m to be represented exactly. Keeping only a finite number of coefficients in the Fourier series introduces unwanted ripples in the desired response, known as the Gibbs phenomenon [47,48]. Such ripples can be minimized using an appropriate window, but at the expense of wider transition regions.

The Fourier series method may be summarized as follows. Given a desired response, say $A_d(\psi)$, pick an odd or even window length, for example $N = 2M + 1$, and calculate the N ideal weights by evaluating the inverse transform:

$$a_d(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} A_d(\psi) e^{-jm\psi} d\psi, \quad m = 0, \pm 1, \dots, \pm M \quad (20.3.5)$$

then, the final weights are obtained by windowing with a length- N window $w(m)$:

$$a(m) = w(m) a_d(m), \quad m = 0, \pm 1, \dots, \pm M \quad (20.3.6)$$

This method is convenient only when the required integral (20.3.5) can be done exactly, as when $A_d(\psi)$ has a simple shape such as an ideal lowpass filter. For arbitrarily shaped $A_d(\psi)$ one must evaluate the integrals approximately using an inverse DFT as is done in the Woodward-Lawson frequency-sampling design method discussed in Sec. 20.5.

In addition, the method requires that $A_d(\psi)$ be specified over one complete Nyquist interval, $-\pi \leq \psi \leq \pi$, regardless of whether the visible region $\psi_{\text{vis}} = 2kd$ is more or less than one Nyquist period.

20.4 Sector Beam Array Design

As an example of the Fourier series method, we discuss the design of an array with angular pattern confined into a desired angular sector.

First, we consider the design in ψ -space of an ideal *bandpass* array factor centered at wavenumber ψ_0 with bandwidth of $2\psi_b$. We will see later how to map these specifications into an actual angular sector. The ideal bandpass response is defined over $-\pi \leq \psi \leq \pi$ as follows:

$$A_{\text{BP}}(\psi) = \begin{cases} 1, & \psi_0 - \psi_b \leq \psi \leq \psi_0 + \psi_b \\ 0, & \text{otherwise} \end{cases}$$

For the odd case, the corresponding ideal weights are obtained from Eq. (20.3.2):

$$a_{BP}(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} A_{BP}(\psi) e^{-jm\psi} d\psi = \frac{1}{2\pi} \int_{\psi_0 - \psi_b}^{\psi_0 + \psi_b} 1 \cdot e^{-jm\psi} d\psi$$

which gives:

$$a_{BP}(m) = e^{-jm\psi_0} \frac{\sin(\psi_b m)}{\pi m}, \quad m = 0, \pm 1, \pm 2, \dots \quad (20.4.1)$$

This problem is equivalent to designing an ideal *lowpass* response with cutoff frequency ψ_b and then translating it by $A_{BP}(\psi) = A_{LP}(\psi') = A_{LP}(\psi - \psi_0)$, where $\psi' = \psi - \psi_0$. The lowpass response is defined as:

$$A_{LP}(\psi') = \begin{cases} 1, & -\psi_b \leq \psi' \leq \psi_b \\ 0, & \text{otherwise} \end{cases}$$

and its ideal weights are:

$$a_{LP}(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} A_{LP}(\psi') e^{-jm\psi'} d\psi' = \frac{1}{2\pi} \int_{-\psi_b}^{\psi_b} 1 \cdot e^{-jm\psi'} d\psi' = \frac{\sin(\psi_b m)}{\pi m}$$

Thus, as expected, the ideal weights for the bandpass and lowpass designs are related by a scanning phase: $a_{BP}(m) = e^{-jm\psi_0} a_{LP}(m)$.

A more realistic design of the bandpass response is to prescribe “brickwall” specifications, that is, defining a passband range over which the response is essentially flat and a stopband range over which the response is essentially zero. These ranges are defined by the bandedge frequencies ψ_p and ψ_s , such that the passband is $|\psi - \psi_0| \leq \psi_p$ and the stopband $|\psi - \psi_0| \geq \psi_s$. The specifications of the equivalent lowpass response are shown in Fig. 20.4.1.

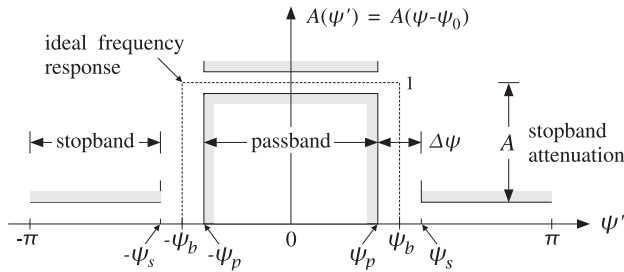


Fig. 20.4.1 Specifications of equivalent lowpass response.

Over the stopband, the attenuation is required to be greater than a *minimum* value, say A dB. The attenuation over the passband need not be specified, because the window method always results in extremely flat passbands for reasonable values of A , e.g., for $A > 35$ dB. Indeed, the maximum passband attenuation is related to A by the approximate formula $A_{\text{pass}} = 17.4\delta$ dB, where $\delta = 10^{-A/20}$ (see Ref. [48].)

Most windows do not allow a user-defined choice for the stopband attenuation. For example, the Hamming window has $A = 54$ dB and the rectangular window $A = 21$ dB.

The *Kaiser window* is the best and simplest of a small class of windows that allow a *variable* choice for A .

Thus, the design specifications are the quantities $\{\psi_p, \psi_s, A\}$. Alternatively, we can take them to be $\{\psi_p, \Delta\psi, A\}$, where $\Delta\psi = \psi_s - \psi_p$ is the transition width. We prefer the latter choice. The design steps for the bandpass response using the Kaiser window are summarized below:

1. From the stopband attenuation A , calculate the so-called D -factor of the window (similar to the broadening factor):

$$D = \begin{cases} \frac{A - 7.95}{14.36}, & \text{if } A > 21 \\ 0.922, & \text{if } A \leq 21 \end{cases} \quad (20.4.2)$$

and the window's shape parameter α :

$$\alpha = \begin{cases} 0.1102(A - 8.7), & \text{if } A \geq 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21), & \text{if } 21 < A < 50 \\ 0, & \text{if } A \leq 21 \end{cases} \quad (20.4.3)$$

2. From the transition width $\Delta\psi$, calculate the length of the window by choosing the smallest *odd* integer $N = 2M + 1$ that satisfies:

$$\Delta\psi = \frac{2\pi D}{N - 1} \quad (20.4.4)$$

Alternatively, if N is given, calculate the transition width $\Delta\psi$.

3. Calculate the samples of the Kaiser window:

$$w(m) = \frac{I_0(\alpha \sqrt{1 - m^2/M^2})}{I_0(\alpha)}, \quad m = 0, \pm 1, \dots, \pm M \quad (20.4.5)$$

where $I_0(x)$ is the modified Bessel function of first kind and zeroth order.

4. Calculate the ideal cutoff frequency ψ_b by taking it to be at the middle between the passband and stopband frequencies:

$$\psi_b = \frac{1}{2}(\psi_p + \psi_s) = \psi_p + \frac{1}{2}\Delta\psi \quad (20.4.6)$$

5. Calculate the final windowed array weights from $a(m) = w(m)a_{BP}(m)$:

$$a(m) = w(m) e^{-jm\psi_0} \frac{\sin(\psi_b m)}{\pi m}, \quad m = 0, \pm 1, \dots, \pm M \quad (20.4.7)$$

Next, we use the above bandpass design in ψ -space to design an array with an angular sector response in ϕ -space. The ideal array will have a pattern that is uniformly flat over an angular sector $[\phi_1, \phi_2]$:

$$A(\phi) = \begin{cases} 1, & \phi_1 \leq \phi \leq \phi_2 \\ 0, & \text{otherwise} \end{cases}$$

Alternatively, we can define the sector by means of its center angle and its width, $\phi_c = (\phi_1 + \phi_2)/2$ and $\phi_b = \phi_2 - \phi_1$. Thus, we have the equivalent definitions of the angular sector:

$$\begin{aligned} \phi_c &= \frac{1}{2}(\phi_1 + \phi_2) & \phi_1 &= \phi_c - \frac{1}{2}\phi_b \\ \phi_b &= \phi_2 - \phi_1 & \phi_2 &= \phi_c + \frac{1}{2}\phi_b \end{aligned} \quad (20.4.8)$$

For a practical design, we may take $[\phi_1, \phi_2]$ to represent the *passband* of the response and assume an angular stopband with attenuation of at least A dB that begins after a small angular transition width $\Delta\phi$ on either side of the passband.

In filter design, the stopband attenuation and the transition width are used to determine the window length N . But in the array problem, because we are usually limited in the number N of available array elements, we must assume that N is given and determine the transition width $\Delta\phi$ from A and N .

Thus, our design specifications are the quantities $\{\phi_1, \phi_2, N, A\}$, or alternatively, $\{\phi_c, \phi_b, N, A\}$. These specifications must be mapped into equivalent ones in ψ -space using the steered wavenumber $\psi' = kd(\cos\phi - \cos\phi_0)$.

We require that the angular passband $[\phi_1, \phi_2]$ be mapped onto the lowpass passband $[-\psi_p, \psi_p]$ in ψ' -space. Thus, we have the conditions:

$$\begin{aligned} \psi_p &= kd \cos \phi_1 - \psi_0 \\ -\psi_p &= kd \cos \phi_2 - \psi_0 \end{aligned}$$

They may be solved for ψ_p and ψ_0 as follows:

$$\begin{aligned} \psi_p &= \frac{1}{2}kd(\cos \phi_1 - \cos \phi_2) \\ \psi_0 &= \frac{1}{2}kd(\cos \phi_1 + \cos \phi_2) \end{aligned} \quad (20.4.9)$$

Using Eq. (20.4.8) and some trigonometry, we have equivalently:

$$\begin{aligned} \psi_p &= kd \sin(\phi_c) \sin\left(\frac{\phi_b}{2}\right) \\ \psi_0 &= kd \cos(\phi_c) \cos\left(\frac{\phi_b}{2}\right) \end{aligned} \quad (20.4.10)$$

Setting $\psi_0 = kd \cos \phi_0$, we find the effective steering angle ϕ_0 :

$$\cos \phi_0 = \cos(\phi_c) \cos\left(\frac{\phi_b}{2}\right) \Rightarrow \phi_0 = \arccos(\cos(\phi_c) \cos(\phi_b/2)) \quad (20.4.11)$$

Note that ϕ_0 is not equal to ϕ_c , except for very narrow widths ϕ_b .

The design procedure is then completed as follows. Given the attenuation A , we calculate the window parameters D, α from Eqs. (20.4.2) and (20.4.3). Since N is given, we calculate the transition width $\Delta\psi$ directly from Eq. (20.4.4). Then, the ideal lowpass frequency ψ_b is calculated from Eq. (20.4.6), that is,

$$\psi_b = \psi_p + \frac{1}{2}\Delta\psi = kd \sin(\phi_c) \sin\left(\frac{\phi_b}{2}\right) + \frac{\pi D}{N-1} \quad (20.4.12)$$

Finally, the array weights are obtained from Eq. (20.4.7). The transition width $\Delta\phi$ can be approximated by linearizing $\psi = kd \cos \phi$ around ϕ_1 , or around ϕ_2 , or around ϕ_c . We prefer the latter choice, giving:

$$\Delta\phi = \frac{\Delta\psi}{kd \sin \phi_c} = \frac{2\pi D}{kd(N-1) \sin \phi_c} \quad (20.4.13)$$

The design method can be extended to the case of even $N = 2M$. The integral (20.3.4) can still be done exactly. The Kaiser window expression (20.4.5) remains the same for $m = \pm 1, \pm 2, \dots, \pm M$. We note the symmetry $w(-m) = w(m)$. After windowing and scanning with ψ_0 , we get the final designed weights:

$$a(\pm m) = w(m) e^{\mp j(m-1/2)\psi_0} \frac{\sin(\psi_b(m-1/2))}{\pi(m-1/2)}, \quad m = 1, 2, \dots, M \quad (20.4.14)$$

The MATLAB function `sector` implements the above design steps for either even or odd N . Its usage is as follows:

```
[a, dph] = sector(d, ph1, ph2, N, A); % A=stopband attenuation in dB
```

Fig. 20.4.2 shows four design examples having sector $[\phi_1, \phi_2] = [45^\circ, 75^\circ]$, or center $\phi_c = 60^\circ$ and width $\phi_b = 30^\circ$. The number of array elements was $N = 21$ and $N = 41$, with half-wavelength spacing $d = \lambda/2$. The stopband attenuations were $A = 20$ and $A = 40$ dB. The two cases with $A = 20$ dB are equivalent to using the rectangular window. They have visible Gibbs ripples in their passband. Some typical MATLAB code for generating these graphs is as follows:

```
d=0.5; ph1=45; ph2=75; N=21; A=20;
[a, dph] = sector(d, ph1, ph2, N, A);
[g, ph] = array(d, a, 400);
dbz(ph,g, 30, 80);
addray(ph1, '--'); addray(ph2, '--');
```

The basic design tradeoff is between N and A and is captured by Eq. (20.4.4). Because D is linearly increasing with A , the transition width will increase with A and decrease with N . As A increases, the passband exhibits no Gibbs ripples but at the expense of larger transition width.

20.5 Woodward-Lawson Frequency-Sampling Design

As mentioned earlier, the Fourier series method is feasible only when the inverse transform integrals (20.3.2) and (20.3.4) can be done exactly. If not, we may use the

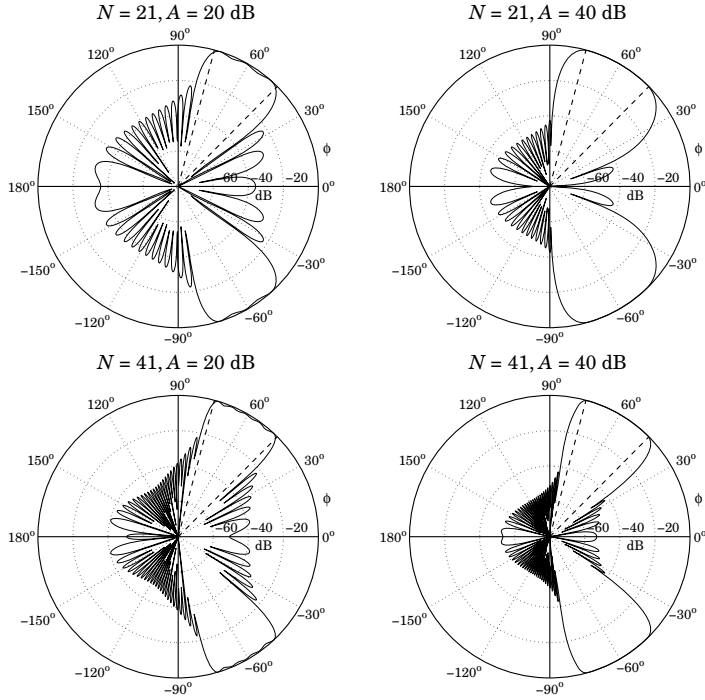


Fig. 20.4.2 Angular sector array design with the Kaiser window.

frequency-sampling design method of DSP [47,48]. In the array context, the method is referred to as the *Woodward-Lawson* method.

For an N -element array, the method is based on performing an inverse N -point DFT. It assumes that N samples of the desired array factor $A(\psi)$ are available, that is, $A(\psi_i)$, $i = 0, 1, \dots, N-1$, where ψ_i are the N DFT frequencies:

$$\psi_i = \frac{2\pi i}{N}, \quad i = 0, 1, \dots, N-1, \quad (\text{DFT frequencies}) \quad (20.5.1)$$

The frequency samples $A(\psi_i)$ are related to the array weights via the *forward* N -point DFT's obtained by evaluating Eqs. (20.1.1) and (20.1.2) at the N DFT frequencies:

$$A(\psi_i) = a_0 + \sum_{m=1}^M [a_m e^{jm\psi_i} + a_{-m} e^{-jm\psi_i}], \quad (N = 2M + 1) \quad (20.5.2)$$

$$A(\psi_i) = \sum_{m=1}^M [a_m e^{j(m-1/2)\psi_i} + a_{-m} e^{-j(m-1/2)\psi_i}], \quad (N = 2M)$$

where ψ_i are given by Eq. (20.5.1). The corresponding *inverse* N -point DFT's are as follows. For odd $N = 2M + 1$,

$$a_m = \frac{1}{N} \sum_{i=0}^{N-1} A(\psi_i) e^{-jm\psi_i}, \quad m = 0, \pm 1, \pm 2, \dots, \pm M \quad (20.5.3)$$

and for even $N = 2M$,

$$a_{\pm m} = \frac{1}{N} \sum_{i=0}^{N-1} A(\psi_i) e^{\mp j(m-1/2)\psi_i}, \quad m = 1, 2, \dots, M \quad (20.5.4)$$

There is an alternative definition of the N DFT frequencies ψ_i for which the forms of the forward and inverse DFT's, Eqs. (20.5.2)–(20.5.4), remain the same. For either even or odd N , we define:

$$\psi_i = \frac{2\pi(i-K)}{N}, \quad (\text{alternative DFT frequencies}) \quad (20.5.5)$$

where $i = 0, 1, \dots, N-1$ and $K = (N-1)/2$.

This definition makes a difference only for even N , in which case the index $i-K$ takes on all the *half-integer* values in the *symmetric* interval $[-K, K]$. For odd N , Eq. (20.5.5) amounts to a re-indexing of Eq. (20.5.1), with $i-K$ taking values now over the symmetric integer interval $[-K, K]$.

For both the standard and the alternative sets, the N complex numbers $z_i = e^{j\psi_i}$ are equally spaced around the unit circle. For odd N , they are the N -th roots of unity, that is, the solutions of the equation $z^N = 1$. For the alternative set with even N , they are the N solutions of the equation $z^N = -1$.

The alternative set is usually preferred in array processing. In DSP, it leads to the discrete cosine transform. The MATLAB function `woodward` implements the inverse DFT operations (20.5.3) and (20.5.4), for either the standard or the alternative definition of ψ_i . Its usage is as follows:

```
a = woodward(A, alt); % alt=0,1 for standard or alternative
```

The frequency-sampling array design method is summarized as follows: Given a set of N frequency response values $A(\psi_i)$, $i = 0, 1, \dots, N-1$, calculate the N array weights $a(m)$ using the inverse DFT formulas (20.5.3) or (20.5.4). Then, replace the weights by their windowed versions using any symmetric length- N window. The final expressions for the windowed weights are, for odd $N = 2M + 1$,

$$a(m) = w(m) \frac{1}{N} \sum_{i=0}^{N-1} A(\psi_i) e^{-jm\psi_i}, \quad m = 0, \pm 1, \pm 2, \dots, \pm M \quad (20.5.6)$$

and for even $N = 2M$,

$$a(\pm m) = w(\pm m) \frac{1}{N} \sum_{i=0}^{N-1} A(\psi_i) e^{\mp j(m-1/2)\psi_i}, \quad m = 1, 2, \dots, M \quad (20.5.7)$$

As an example, consider the design of a sector beam with edges at $\phi_1 = 45^\circ$ and $\phi_2 = 75^\circ$. Thus, the beam is centered at $\phi_c = 60^\circ$ and has width $\phi_b = 30^\circ$.

As ϕ ranges over $[\phi_1, \phi_2]$, the wavenumber $\psi = kd \cos \phi$ will range over $kd \cos \phi_2 \leq \psi \leq kd \cos \phi_1$. For all DFT frequencies ψ_i that lie in this interval, we set $A(\psi_i) = 1$, otherwise, we set $A(\psi_i) = 0$. Assuming the alternative definition for ψ_i , we have the passband condition:

$$kd \cos \phi_2 \leq \frac{2\pi(i-K)}{N} \leq kd \cos \phi_1$$

Setting $kd = 2\pi d/\lambda$ and solving for the DFT index $i - K$, we find:

$$j_1 \leq i - K \leq j_2, \quad \text{where} \quad j_1 = \frac{Nd}{\lambda} \cos \phi_2, \quad j_2 = \frac{Nd}{\lambda} \cos \phi_1$$

This range determines the DFT indices i for which $A(\psi_i) = 1$. The inverse DFT summation over i will then be restricted over this subset of i 's. Fig. 20.5.1 shows the response of a 20-element array with half-wavelength spacing, $d = \lambda/2$, designed with a rectangular and a Hamming window. The MATLAB code for generating the right graph was as follows:

```
d=0.5; N=20; ph1=45; ph2=75; alt=1; K=(N-1)/2;
j1 = N*d*cos(ph2*pi/180);
j2 = N*d*cos(ph1*pi/180);
i = (0:N-1); % DFT index
j = i - alt*K; % alternative DFT index
A = (j>=j1)&(j<=j2); % equals 1, if j1 ≤ j ≤ j2, and 0, otherwise
a = woodward(A, alt); % inverse DFT
w = 0.54 - 0.46*cos(2*pi*i/(N-1)); % Hamming window
awind = a .* w; % windowed weights
[g,ph] = array(0.5, awind, 400); % array gain
dbz(ph, g, 30, 80);
addray(ph1,'--'); addray(ph2,'--');
```

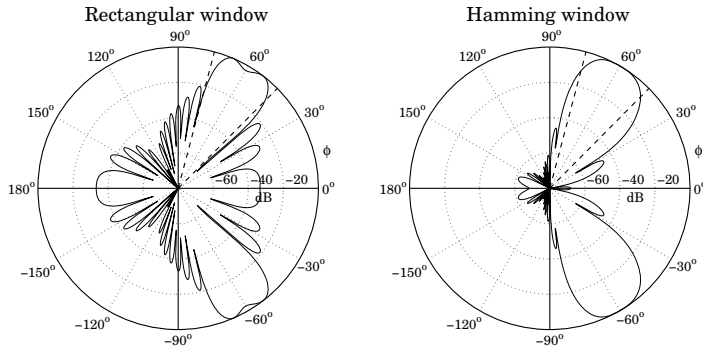


Fig. 20.5.1 Angular sector array design with Woodward-Lawson method.

The sidelobes of the Hamming window are down approximately at the expected 54-dB level (they reach 54 dB for larger N .) The design is comparable to that of Fig. 20.4.2.

The power of this method lies in the ability to specify *any* shape for the array factor through its frequency samples. The method works well for half-wavelength spacing $d = \lambda/2$, because all N DFT frequencies ψ_i lie within the visible region, which coincides in this case with the full Nyquist interval, $-\pi \leq \psi \leq \pi$.

As another example, we consider the design of an array with a secant-squared gain pattern, which is relevant in air search radars as discussed in Sec. 15.11. We consider an array of N elements along the z -direction with half-wavelength spacing $d = \lambda/2$. The corresponding wavenumber ψ will be $\psi = k_z d$, or

$$\psi = kd \cos \theta$$

The design of the secant-squared gain pattern requires that the array factor itself have a secant dependence. Indeed,

$$g(\theta) = |A(\psi)|^2 = \frac{K}{\cos^2 \theta} \Rightarrow |A(\psi)| = \frac{K^{1/2}}{|\cos \theta|}$$

Because the secant pattern is defined only up to an angle θ_{\max} , we may define the theoretical array factor in the normalized form:

$$A(\theta) = \begin{cases} \frac{\cos \theta_{\max}}{\cos \theta}, & \text{if } 0 \leq \theta \leq \theta_{\max} \\ 1, & \text{if } \theta_{\max} < \theta \leq 90^\circ \end{cases} \quad (20.5.8)$$

As θ varies over $[0, \theta_{\max}]$, the wavenumber $\psi = kd \cos \theta$ will vary over $[\psi_{\max}, kd]$, where $\psi_{\max} = kd \cos \theta_{\max}$. Because $d = \lambda/2$, we have $kd = \pi$ and the ψ -range becomes $[\psi_{\max}, \pi]$. Noting that $\cos \theta_{\max} / \cos \theta = \psi_{\max} / \psi$, we can rewrite Eq. (20.5.8) in terms of ψ :

$$A(\psi) = \begin{cases} \frac{\psi_{\max}}{\psi}, & \text{if } \psi_{\max} \leq \psi \leq \pi \\ 1, & \text{if } 0 \leq \psi < \psi_{\max} \end{cases} \quad (20.5.9)$$

We symmetrize $A(-\psi) = A(\psi)$ to cover the entire 2π Nyquist interval in ψ . Evaluating Eq. (20.5.9) at the N DFT frequencies $\psi_i = 2\pi i/N$, we obtain the array weights by doing an inverse DFT and then windowing the array coefficients with a Hamming window. Fig. 20.5.2 shows a design case with $N = 21$ and $\theta_{\max} = 70^\circ$. The figure compares the Hamming and rectangular window designs to the exact expression (20.5.8). The details of the design are indicated in the MATLAB code:

```
N=21; K=(N-1)/2; d=0.5; thmax=70;
psmax = 2*pi*d * cos(thmax*pi/180);
Ai = ones(1,K+1);
psi = 2*pi*(0:K)/N; % half of DFT frequencies
j = find(psi); % non-zero ψ's
Ai(j) = psmax*(psi(j)>=psmax)./psi(j) + (psi(j)<psmax); % half of the DFT values
Ai = [Ai, Ai(K:-1:1)]; % all the DFT values
a = woodward(Ai, 0) / N; % inverse DFT with alt=0
```



```

aw = a .* (0.54 - 0.46*cos(2*pi*(0:N-1)/(N-1))); % Hamming

th = (0:200) * 90 / 200;
ps = 2*pi*d * cos(th*pi/180);

A = abs(dtft(a, -ps)); % rectangular design
Aw = abs(dtft(aw, -ps)); % Hamming design
A0 = psmax*(ps>=psmax) ./ ps + (ps<psmax); % exact pattern

```

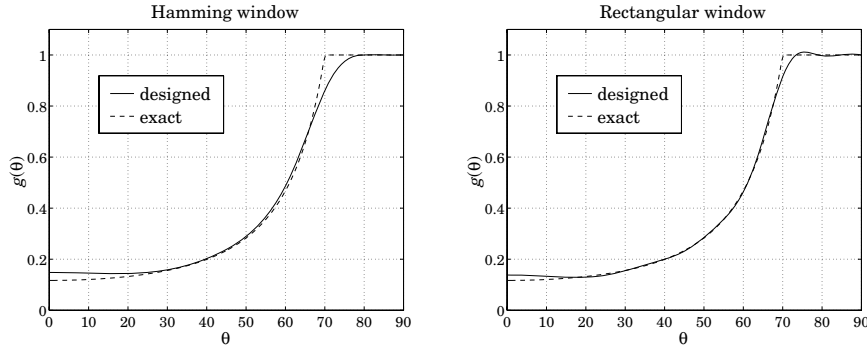


Fig. 20.5.2 Woodward-Lawson design of secant-squared array gain.

20.6 Discretization of Continuous Line Sources

One-dimensional arrays may be thought of as arising from the *spatial sampling* of continuous line current distributions. Consider, for example, a current $I(x)$ flowing along the x -axis. Its current density is $J_x(x, y, z) = I(x)\delta(y)\delta(z)$, where the delta functions confine the current on the x -axis. The corresponding radiation vector will have only an x -component:

$$\begin{aligned}
 F_x(k_x, k_y, k_z) &= \int J_x(x, y, z) e^{jk_x x + jk_y y + jk_z z} dx dy dz \\
 &= \int I(x) \delta(y) \delta(z) e^{jk_x x + jk_y y + jk_z z} dx dy dz = \int_{-\infty}^{\infty} I(x) e^{jk_x x} dx
 \end{aligned}$$

Thus, $F_x(k_x)$ depends only on the k_x wavevector component and is the spatial Fourier transform of the line current $I(x)$:

$$F_x(k_x) = \int_{-\infty}^{\infty} I(x) e^{jk_x x} dx \quad (20.6.1)$$

In spherical coordinates, k_x is given by $k_x = k \sin \theta \cos \phi$, with $k = 2\pi/\lambda$. The range of k_x values when θ, ϕ vary over $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$ is “visible region”. The inversion of the Fourier transform, however, requires knowledge of $F_x(k_x)$ over all k_x ,

20.6. Discretization of Continuous Line Sources

and in such case the inverse is:

$$I(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_x(k_x) e^{-jk_x x} dk_x \quad (20.6.2)$$

Suppose now that the current $I(x)$ is sampled at the regular intervals $x_m = md$ with spacing d and integer m . The sampled current may be represented as the sum of impulses:

$$\hat{I}(x) = \sum_{m=-\infty}^{\infty} I(x_m) \delta(x - x_m) = \sum_{m=-\infty}^{\infty} I_m \delta(x - md) \quad (20.6.3)$$

where we set $I_m = I(x_m) = I(md)$. Then, the corresponding Fourier transform will be:

$$\hat{F}_x(k_x) = \int_{-\infty}^{\infty} \hat{I}(x) e^{jk_x x} dx = \sum_{m=-\infty}^{\infty} I_m e^{jm k_x d} = \sum_{m=-\infty}^{\infty} I_m e^{jm \psi} \quad (20.6.4)$$

This has precisely the form of an array factor with $\psi = k_x d$. The pattern $\hat{F}_x(k_x)$ is periodic in k_x with period $k_s = 2\pi/d$, which is the sampling frequency in units of radians/meter. Equivalently, $\hat{F}_x(k_x)$ is periodic in ψ with period 2π . The Poisson summation formula [47] relates $\hat{F}_x(k_x)$ to the unsampled pattern $F_x(k_x)$ as a sum of shifted replicas:

$$\hat{F}_x(k_x) = \frac{1}{d} \sum_{n=-\infty}^{\infty} F_x(k_x - n k_s) \quad (20.6.5)$$

Aliasing, that is, the overlapping of the spectral replicas, can be avoided only if $F_x(k_x)$ is bandlimited to within the Nyquist interval, $|k_x| \leq k_s/2$. This would imply that $I(x)$ have infinite extent.

In practice, $I(x)$ is assumed to be space-limited with a finite extent, say, over an interval $-l/2 \leq x \leq l/2$. In this case, $F_x(k_x)$ cannot be bandlimited and therefore, aliasing will always occur. However, if the pattern $F(k_x)$ attenuates with large k_x , aliasing may be minimized by selecting a small enough d .

Eqs. (20.6.4) and (20.6.5) provide two equivalent ways to express the spectrum of the sampled current. Eq. (20.6.4) can be inverted to recover the current samples I_m :

$$I_m = \frac{1}{k_s} \int_{-k_s/2}^{k_s/2} \hat{F}_x(k_x) e^{-jm k_x d} dk_x = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{F}_x(\psi) e^{-jm \psi} d\psi \quad (20.6.6)$$

which is the inverse discrete-space Fourier transform that we introduced in (19.4.8). By using the z -domain variable $z = e^{j\psi}$, (20.6.4) can also be written as the spatial z -transform:

$$\hat{F}_x(z) = \sum_{m=-\infty}^{\infty} I_m z^m \quad (20.6.7)$$

Next, we focus on finite line sources $I(x)$, $-l/2 \leq x \leq l/2$. Then, (20.6.1) reads:

$$F_x(k_x) = \int_{-l/2}^{l/2} I(x) e^{jk_x x} dx \quad (20.6.8)$$

It proves convenient to define a normalized wavenumber variable u by:

$$u = \frac{l k_x}{2\pi} \Leftrightarrow k_x = \frac{2\pi u}{l} \Leftrightarrow u = \frac{l}{\lambda} \sin \theta \cos \phi \quad (20.6.9)$$

and define a scaled pattern $F(u) = F_x(k_x)/l$. Then, we have the Fourier relationships:

$$F(u) = \frac{1}{l} \int_{-l/2}^{l/2} I(x) e^{j2\pi ux/l} dx \Leftrightarrow I(x) = \int_{-\infty}^{\infty} F(u) e^{-j2\pi ux/l} du \quad (20.6.10)$$

If $I(x)$ were periodic with period l , then $2\pi/l$ would be its fundamental harmonic and $2\pi u/l$ would be interpreted as the u th harmonic. Indeed, the continuous-line version of the Woodward-Lawson method gives u just such an interpretation. Let us define the periodic extension of the space-limited $I(x)$ with period l to be the sum of its replicas:

$$\tilde{I}(x) = \sum_{n=-\infty}^{\infty} I(x - nl) \quad (20.6.11)$$

Then, $\tilde{I}(x)$, being periodic, could be expanded in a Fourier series with coefficients:

$$\tilde{I}(x) = \sum_{p=-\infty}^{\infty} c_p e^{-j2\pi px/l}, \quad c_p = \frac{1}{l} \int_{-l/2}^{l/2} \tilde{I}(x) e^{j2\pi px/l} dx \quad (20.6.12)$$

Because $\tilde{I}(x) = I(x)$ over the period $-l/2 \leq x \leq l/2$, the above integral for the p th coefficient implies from (20.6.10) that $c_p = F(u)$ with $u = p$. Thus, restricting x over its basic period, we have the representation:

$$I(x) = \sum_{p=-\infty}^{\infty} F(p) e^{-j2\pi px/l}, \quad -\frac{l}{2} \leq x \leq \frac{l}{2} \quad (20.6.13)$$

The pattern $F(u)$ may itself be expressed in terms of its samples $F(p)$. We have from (20.6.13):

$$F(u) = \frac{1}{l} \int_{-l/2}^{l/2} I(x) e^{j2\pi ux/l} dx = \sum_{p=-\infty}^{\infty} F(p) \frac{1}{l} \int_{-l/2}^{l/2} e^{j2\pi(u-p)x/l} dx, \quad \text{or,} \quad (20.6.14)$$

$$F(u) = \sum_{p=-\infty}^{\infty} F(p) \frac{\sin(\pi(u-p))}{\pi(u-p)}$$

Eqs. (20.6.13) and (20.6.14) are the continuous-line version of the Woodward-Lawson method, which is of course equivalent to the application of Shannon's sampling theorem to the space-limited function $I(x)$, and our derivation is nothing more than the proof of that theorem.

For discrete arrays, we must sample in space $x_m = md$, not in frequency. By taking N samples over the length l , that is, $d = l/N$, and truncating the summation in (20.6.13) to $p = 0, 1, \dots, N-1$, we obtain the practical version of the Woodward-Lawson method that we used in the previous section.

For an N -element finite array, the z -transform $\hat{F}_x(z)$ of Eq. (20.6.7) becomes a polynomial of degree $N-1$ in z . Such an array can be designed directly in discrete-space domain, or it can be designed by mapping a given continuous line source pattern to the discrete case. This can be accomplished approximately by mapping $N-1$ zeros of the

continuous pattern to $N-1$ zeros of the array using the mapping $z = e^{j\psi} = e^{jk_x d}$. Since $d = l/N$, the mapping from u -space to ψ -space becomes $\psi = k_x d = 2\pi u d/l = 2\pi u/N$:

$$\psi = k_x d = \frac{2\pi u}{N} \quad (20.6.15)$$

Therefore, if u_n , $n = 1, 2, \dots, N-1$ are the $N-1$ zeros of the pattern $F(u)$ on which the design is to be based, then, we may define the corresponding zeros of the array by:

$$\psi_n = \frac{2\pi u_n}{N} \Rightarrow z_n = e^{j\psi_n} = e^{j2\pi u_n/N}, \quad n = 1, 2, \dots, N-1 \quad (20.6.16)$$

and construct the array pattern polynomial from these zeros:

$$A(z) = \prod_{n=1}^{N-1} (z - z_n) \quad (20.6.17)$$

The method is an approximation because $F(u)$ generally has an infinity of zeros. However, good results are obtained if N is large (e.g., $N > 10$).

To clarify the above definitions and Fourier relationships, we consider three examples: (a) the uniform line source and how it relates to the uniform array, (b) Taylor's one-parameter line source and its use to design Taylor-Kaiser arrays, and (c) Taylor's ideal line source, which is an idealization of the Chebyshev array, and leads to the so-called Taylor's \bar{n} distribution. A uniform line source has constant current:

$$I(x) = \begin{cases} 1, & \text{if } -l/2 \leq x \leq l/2 \\ 0, & \text{otherwise} \end{cases} \quad (20.6.18)$$

Its pattern is:

$$F(u) = \frac{1}{l} \int_{-l/2}^{l/2} I(x) e^{j2\pi ux/l} dx = \frac{1}{l} \int_{-l/2}^{l/2} e^{j2\pi ux/l} dx = \frac{\sin(\pi u)}{\pi u} \quad (20.6.19)$$

Its zeros are at the non-zero integers $u_n = \pm n$, for $n = 1, 2, \dots$. By selecting the first $N-1$ of these, $u_n = n$, for $n = 1, 2, \dots, N-1$, we may map them to the $N-1$ zeros of the uniform array:

$$z_n = e^{j2\pi u_n/N} = e^{j2\pi n/N}, \quad n = 1, 2, \dots, N-1$$

The constructed array polynomial will be then,

$$A(z) = \frac{1}{N} \prod_{n=1}^{N-1} (z - z_n) = \frac{1}{N} \prod_{n=1}^{N-1} (z - e^{j2\pi n/N}) = \frac{1}{N} \frac{\prod_{n=0}^{N-1} (z - e^{j2\pi n/N})}{z - 1}$$

where we introduced a scale factor $1/N$ and multiplied and divided by the factor $(z-1)$. But the numerator polynomial, being a monic polynomial and having as roots the N th roots of unity, must be equal to $z^N - 1$. Thus,

$$A(z) = \frac{1}{N} \frac{z^N - 1}{z - 1} = \frac{1}{N} (1 + z + z^2 + \dots + z^{N-1})$$

which has uniform array weights, $a_m = 1/N$. Replacing $z = e^{j\psi} = e^{j2\pi u/N}$, we have:

$$A(\psi) = \frac{1}{N} \frac{e^{j\psi N} - 1}{e^{j\psi} - 1} = \frac{\sin(N\psi/2)}{N \sin(\psi/2)} e^{j\psi(N-1)/2} = \frac{\sin(\pi u)}{N \sin(\pi u/N)} e^{j\pi u(N-1)/N}$$

For large N and fixed value of u , we may use the approximation $\sin x \simeq x$ in the denominator which tends to $N \sin(\pi u/N) \simeq N(\pi u/N) = \pi u$, thus, approximating the $\sin \pi u / \pi u$ pattern of the continuous line case.

Taylor's one-parameter continuous line source [1085] has current $I(x)$ and corresponding pattern $F(u)$ given by the Fourier transform pair [156]:

$$F(u) = \frac{\sinh(\pi\sqrt{B^2 - u^2})}{\pi\sqrt{B^2 - u^2}} \Leftrightarrow I(x) = I_0\left(\pi B\sqrt{1 - (2x/l)^2}\right) \quad (20.6.20)$$

where $-l/2 \leq x \leq l/2$ and $I_0(\cdot)$ is the modified Bessel function of first kind and zeroth order, and B is a positive parameter that controls the sidelobe level. For $u > B$, the pattern becomes a sinc-pattern in the variable $\sqrt{u^2 - B^2}$, and for large u , it tends to the pattern of the uniform line source. We will discuss this further in Sec. 20.10.

Taylor's ideal line source [1086] also has a parameter that controls the sidelobe level and is defined by the Fourier pair [156]:

$$F(u) = \cosh(\pi\sqrt{A^2 - u^2})$$

$$I(x) = \frac{I_1(\pi A\sqrt{1 - (2x/l)^2})}{\sqrt{1 - (2x/l)^2}} \frac{\pi A}{l} + \delta\left(x - \frac{l}{2}\right) + \delta\left(x + \frac{l}{2}\right) \quad (20.6.21)$$

where $I_1(\cdot)$ is the modified Bessel function of first kind and first order. Van der Maas [1074] showed first that this pair is the limit of a Dolph-Chebyshev array in the limit of a large number of array elements. We will explore it further in Sec. 20.12.

20.7 Narrow-Beam Low-Sidelobe Designs

The problem of designing arrays having narrow beams with low sidelobes is equivalent to the DSP problem of spectral analysis of *windowed sinusoids*. A single beam corresponds to a single sinusoid, multiple beams to multiple sinusoids.

To understand this equivalence, suppose one wants to design an infinitely narrow beam toward some look direction $\phi = \phi_0$. In ψ -space, the array factor (spatial or wavenumber spectrum) should be the infinitely thin spectral line:[†]

$$A(\psi) = 2\pi\delta(\psi - \psi_0)$$

where $\psi = kd \cos \phi$ and $\psi_0 = kd \cos \phi_0$. Inserting this into the inverse DSFT of Eq. (20.3.2), gives the double-sided infinitely-long array, for $-\infty < m < \infty$:

$$a(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} A(\psi) e^{-jm\psi} d\psi = \frac{1}{2\pi} \int_{-\pi}^{\pi} 2\pi\delta(\psi - \psi_0) e^{-jm\psi} d\psi = e^{-j\psi_0 m}$$

[†]To be periodic in ψ , all the Nyquist replicas of this term must be added. But they are not shown here because ψ_0 and ψ are assumed to lie in the central Nyquist interval $[-\pi, \pi]$.

20.7. Narrow-Beam Low-Sidelobe Designs

This is the spatial analog of an infinite sinusoid $a(n) = e^{j\omega_0 n}$ whose spectrum is the sharp spectral line $A(\omega) = 2\pi\delta(\omega - \omega_0)$. A finite-duration sinusoid is obtained by windowing with a length- N time window $w(n)$ resulting in $a(n) = w(n)e^{j\omega_0 n}$.

In the frequency domain, the effect of windowing is to replace the spectral line $\delta(\omega - \omega_0)$ by its smeared version $W(\omega - \omega_0)$, where $W(\omega)$ is the DTFT of the window $w(n)$. The spectrum $W(\omega - \omega_0)$ exhibits a main lobe at $\omega = \omega_0$ and sidelobes. The main lobe gets narrower with increasing N .

A finite N -element array with a narrow beam and low sidelobes, and steered towards an angle ϕ_0 , can be obtained by windowing the infinite narrow-beam array with an appropriate length- N spatial window $w(m)$. For odd $N = 2M + 1$, or even $N = 2M$, we define respectively:

$$\begin{aligned} a(m) &= e^{-jm\psi_0} w(m), \quad m = 0, \pm 1, \pm 2, \dots, \pm M \\ a(\pm m) &= e^{\mp j(m-1/2)\psi_0} w(\pm m), \quad m = 1, 2, \dots, M \end{aligned} \quad (20.7.1)$$

In both cases, the array factor of Eqs. (20.1.1) and (20.1.2) becomes:

$$A(\psi) = W(\psi - \psi_0) \quad (\text{narrow beam array factor}) \quad (20.7.2)$$

where $W(\psi)$ is the DSFT of the window, defined for odd or even N as:

$$\begin{aligned} W(\psi) &= w(0) + \sum_{m=1}^M [w(m)e^{jm\psi} + w(-m)e^{-jm\psi}] \\ W(\psi) &= \sum_{m=1}^M [w(m)e^{j(m-1/2)\psi} + w(-m)e^{-j(m-1/2)\psi}] \end{aligned} \quad (20.7.3)$$

Assuming a symmetric window, $w(-m) = w(m)$, we can rewrite:

$$\begin{aligned} W(\psi) &= w(0) + 2 \sum_{m=1}^M w(m) \cos(m\psi) \quad (N = 2M + 1) \\ W(\psi) &= 2 \sum_{m=1}^M w(m) \cos((m-1/2)\psi) \quad (N = 2M) \end{aligned} \quad (20.7.4)$$

At broadside, $\psi_0 = 0$, $\phi_0 = 90^\circ$, Eq. (20.7.1) reduces to $a(m) = w(m)$ and the array factor becomes $A(\psi) = W(\psi)$. Thus, the weights of a broadside narrow beam array *are* the window samples $a(m) = w(m)$. The steered weights (20.7.1) can be calculated with the help of the MATLAB function `scan`, or `steer`:

```
a = scan(w, psi0);
a = steer(d, w, phi0);
```

The primary issue in choosing a window function $w(m)$ is the tradeoff between frequency resolution and frequency leakage, that is, between *main-lobe width* and *sidelobe level* [47,48]. Ideally, one would like to meet, as best as possible, the two conflicting requirements of having a very narrow mainlobe and very small sidelobes.

Fig. 20.7.1 shows four narrow-beam design examples illustrating this tradeoff. All designs are 7-element arrays with half-wavelength spacing, $d = \lambda/2$, and steered towards 90° . The Dolph-Chebyshev and Taylor-Kaiser arrays were designed with sidelobe level of $R = 20$ dB.

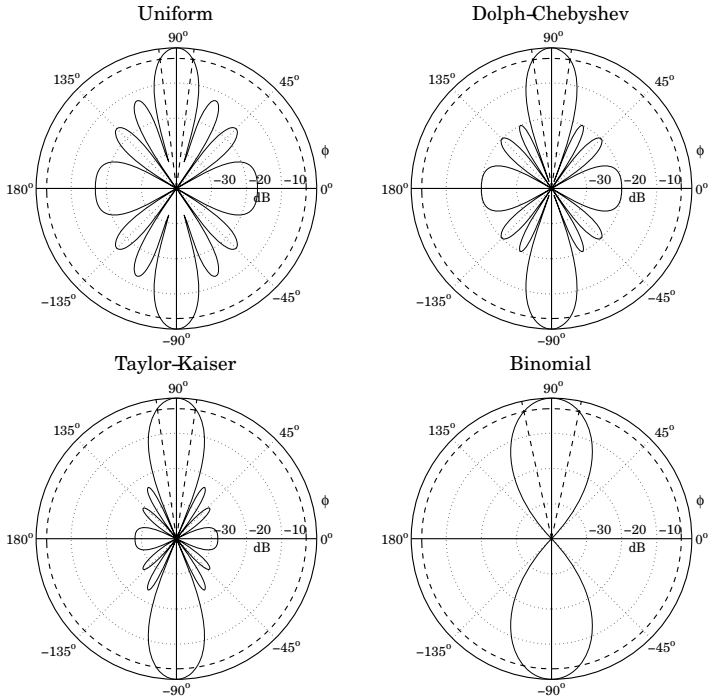


Fig. 20.7.1 Narrow beam design examples.

Shown on the graphs are also the half-power 3-dB circles being intersected by the angular rays at the 3-dB angles. For comparison, we list below the designed array weights (normalized to unity at their endpoints) and the corresponding 3-dB angular widths (in degrees):

Uniform	Dolph-Chebyshev	Taylor-Kaiser	Binomial
1	1.0000	1.0000	1
1	1.2764	1.8998	6
1	1.6837	2.6057	15
1	1.8387	2.8728	20
1	1.6837	2.6057	15
1	1.2764	1.8998	6
1	1.0000	1.0000	1
14.5°	16.4°	16.8°	24.6°

The uniform array has the narrowest mainlobe but also the highest sidelobes. The Dolph-Chebyshev is optimum in the sense that, for the given sidelobe level of 20 dB, it has the narrowest width. The Taylor-Kaiser is somewhat wider than the Dolph-Chebyshev, but it exhibits better sidelobe behavior. The binomial array has the widest mainlobe but no sidelobes at all.

Fig. 20.7.2 shows another set of examples. All designs are 21-element arrays with half-wavelength spacing, $d = \lambda/2$, and scanned towards 60° .

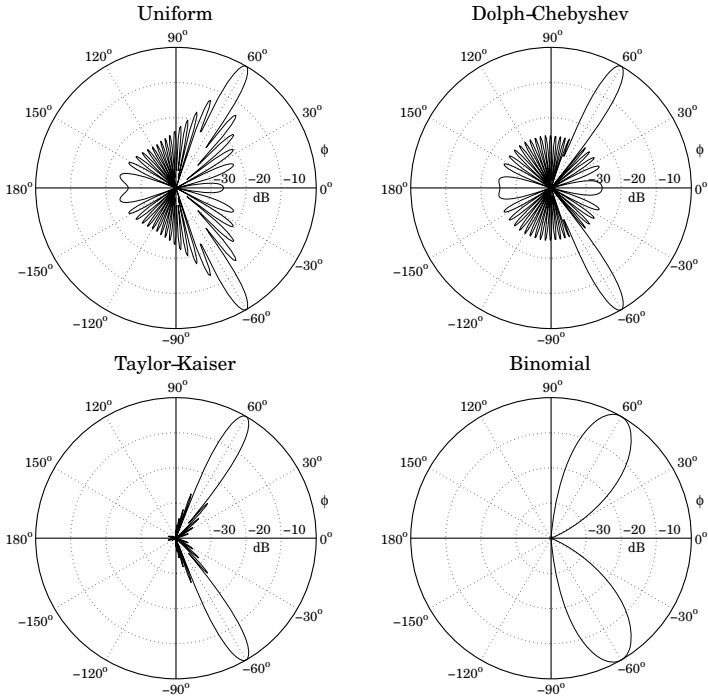


Fig. 20.7.2 Comparison of steered 21-element narrow-beam arrays.

The Dolph-Chebyshev and Taylor arrays were designed with sidelobe level of $R = 25$ dB. The uniform array has sidelobes at $R = 13$ dB. Because N is higher than in Fig. 20.7.1, the beams will be much narrower. The 3-dB beamwidths are in the four cases:

- $\Delta\phi_{3\text{dB}} = 5.58^\circ$ Uniform
- $\Delta\phi_{3\text{dB}} = 6.44^\circ$ Dolph-Chebyshev
- $\Delta\phi_{3\text{dB}} = 7.03^\circ$ Taylor-Kaiser
- $\Delta\phi_{3\text{dB}} = 15.64^\circ$ Binomial

The two key parameters characterizing a window are the 3-dB width of its main lobe, $\Delta\psi_{3\text{dB}}$, and its sidelobe level R (in dB). For some windows, such as Dolph-Chebyshev

and binomial, $\Delta\psi_{3\text{dB}}$ can be calculated exactly. In others, such as Taylor-Kaiser and Hamming, it can be calculated approximately by Eq. (19.10.2), that is,

$$\Delta\psi_{3\text{dB}} = 0.886 \frac{2\pi b}{N} \quad (3\text{-dB width in } \psi\text{-space}) \quad (20.7.5)$$

where b is a broadening factor that depends on the choice of window and increases with the sidelobe attenuation R . As discussed in Sec. 19.10, once $\Delta\psi_{3\text{dB}}$ is known, the angular 3-dB width of the steered array can be computed approximately by:

$$\Delta\phi_{3\text{dB}} = \begin{cases} \frac{\Delta\psi_{3\text{dB}}}{kd \sin \phi_0}, & \text{for } 0^\circ < \phi_0 < 180^\circ \\ 2\sqrt{\frac{\Delta\psi_{3\text{dB}}}{kd}}, & \text{for } \phi_0 = 0^\circ, 180^\circ \end{cases} \quad (20.7.6)$$

This is an adequate approximation in practice. In succeeding sections, we discuss the binomial, Dolph-Chebyshev, and Taylor-Kaiser arrays in more detail. In addition, we discuss prolate arrays, Taylor's \bar{n} distribution, and Villeneuve arrays.

We finish this section by summarizing the uniform array, which is based on the rectangular window and has $b = 1$ and sidelobe level $R = 13$ dB. Its weights, symmetric DSFT, and symmetric z -transform were determined in Example 20.1.2:

$$\begin{aligned} \mathbf{w} &= \frac{1}{N} [1, 1, \dots, 1] \\ W(\psi) &= \frac{\sin\left(\frac{N\psi}{2}\right)}{N \sin\left(\frac{\psi}{2}\right)} \\ W(z) &= \frac{1}{N} \frac{z^{N/2} - z^{-N/2}}{z^{1/2} - z^{-1/2}} = z^{-(N-1)/2} \frac{1}{N} \frac{z^N - 1}{z - 1} \end{aligned} \quad (20.7.7)$$

20.8 Binomial Arrays

The weights of an N -element binomial array are the binomial coefficients:

$$w(m) = \frac{(N-1)!}{m!(N-1-m)!}, \quad m = 0, 1, \dots, N-1 \quad (20.8.1)$$

For example, for $N = 4$ and $N = 5$ they are:

$$\begin{aligned} \mathbf{w} &= [1, 3, 3, 1] \\ \mathbf{w} &= [1, 4, 6, 4, 1] \end{aligned}$$

The binomial weights are the expansion coefficients of the polynomial $(1+z)^{N-1}$. Indeed, the symmetric z -transform of the binomial array is defined as:

$$W(z) = (z^{1/2} + z^{-1/2})^{N-1} = z^{-(N-1)/2} (1+z)^{N-1} \quad (20.8.2)$$

20.9 Dolph-Chebyshev Arrays

Setting $z = e^{j\psi}$, we find the array factor in ψ -space:

$$W(\psi) = (e^{j\psi/2} + e^{-j\psi/2})^{N-1} = \left[2 \cos\left(\frac{\psi}{2}\right) \right]^{N-1} \quad (20.8.3)$$

This response falls monotonically on either side of the peak at $\psi = 0$ until it becomes zero at the Nyquist frequency $\psi = \pm\pi$. Indeed, the z -transform has a multiple zero of order $N-1$ at $z = -1$.

Thus, the binomial response has no sidelobes. This is, of course, at the expense of a fairly wide mainlobe. The 3-dB width $\Delta\psi_{3\text{dB}}$ can be determined by finding the 3-dB frequencies $\pm\psi_3$ that satisfy the half-power condition:

$$\frac{|W(\psi_3)|^2}{|W(0)|^2} = \frac{1}{2} \quad \Rightarrow \quad \left[\cos\left(\frac{\psi_3}{2}\right) \right]^{2(N-1)} = \frac{1}{2}$$

The solution is:

$$\psi_3 = 2 \arccos(2^{-0.5/(N-1)})$$

Therefore, the 3-dB width will be $\Delta\psi_{3\text{dB}} = 2\psi_3$:

$$\Delta\psi_{3\text{dB}} = 4 \arccos(2^{-0.5/(N-1)}) \quad (20.8.4)$$

Once $\Delta\psi_{3\text{dB}}$ is found, the 3-dB width $\Delta\phi_{3\text{dB}}$ in angle space, for an array steered towards an angle ϕ_0 , can be found from Eq. (20.7.6). The MATLAB function `binomial` generates the array weights (steered towards ϕ_0) and 3-dB width. Its usage is:

```
[a, dph] = binomial(d, ph0, N); % binomial array coefficients and beamwidth
```

For example, the fourth graph of the binomial response of Fig. 20.7.1 was generated by the MATLAB code:

```
[a, dph] = binomial(0.5, 90, 5); % array weights and 3-dB width
[g, ph] = array(0.5, a, 200); % compute array gain
dbz(ph, g, 45, 40); % plot gain in dB with 40-dB scale
addcirc(3, 40, '--'); % add 3-dB grid circle
addray(90 + dph/2, '-'); % add rays at 3-dB angles
addray(90 - dph/2, '-');
```

20.9 Dolph-Chebyshev Arrays

Most windows have largest sidelobes near the main lobe. If a window is designed to achieve a minimum sidelobe attenuation of R dB, then typically R will be the attenuation of the sidelobes nearest to the mainlobe; the sidelobes further away will have attenuations higher than R .

Because of the tradeoff between mainlobe width and sidelobe attenuation, the extra attenuation of the furthest sidelobes will come at the expense of increased mainlobe width. If the attenuation of these sidelobes could be decreased (up to the level of the minimum R), then the mainlobe width would narrow.

It follows that for a given *minimum* desired sidelobe level R , the *narrowest* mainlobe width will be achieved by a window whose sidelobes are all *equal* to R . Conversely,

for a given *maximum* desired mainlobe width, the *largest* sidelobe attenuation will be achieved by a window with equal sidelobe levels.

This “optimum” window is the Dolph-Chebyshev window, which is constructed with the help of Chebyshev polynomials. The m th Chebyshev polynomial $T_m(x)$ is:

$$T_m(x) = \cos(m \arccos(x)) \quad (20.9.1)$$

If $|x| > 1$, the inverse cosine $\arccos(x)$ becomes imaginary, and the expression can be rewritten in terms of hyperbolic cosines: $T_m(x) = \cosh(m \operatorname{acosh}(x))$.

Setting $x = \cos \theta$, or $\theta = \arccos(x)$, we see that $T_m(x) = \cos(m\theta)$. Using trigonometric identities, the quantity $\cos(m\theta)$ can always be expanded as a polynomial in powers of $\cos \theta$. The expansion coefficients are precisely the coefficients of the powers of x of the Chebyshev polynomial. For example, we have:

$$\begin{aligned} \cos(0\theta) &= 1 & T_0(x) &= 1 \\ \cos(1\theta) &= \cos \theta & T_1(x) &= x \\ \cos(2\theta) &= 2 \cos^2 \theta - 1 & \Rightarrow T_2(x) &= 2x^2 - 1 \\ \cos(3\theta) &= 4 \cos^3 \theta - 3 \cos \theta & T_3(x) &= 4x^3 - 3x \\ \cos(4\theta) &= 8 \cos^4 \theta - 8 \cos^2 \theta + 1 & T_4(x) &= 8x^4 - 8x^2 + 1 \end{aligned}$$

For $|x| < 1$, the Chebyshev polynomial has equal ripples, whereas for $|x| > 1$, it increases like x^m . Moreover, $T_m(x)$ is even in x if m is even, and odd in x if m is odd. Fig. 20.9.1 depicts the Chebyshev polynomials $T_9(x)$ and $T_{10}(x)$.

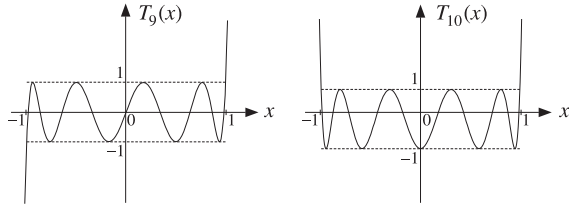


Fig. 20.9.1 Chebyshev polynomials of orders nine and ten.

The Dolph-Chebyshev window is defined such that its sidelobes will correspond to a portion of the equi-ripple range $|x| \leq 1$ of the Chebyshev polynomial, whereas its mainlobe will correspond to a portion of the range $x > 1$.

For either even or odd N , Eq. (20.7.4) implies that any window spectrum $W(\psi)$ can be written in general as a polynomial of degree $N - 1$ in the variable $u = \cos(\psi/2)$. Indeed, we have for the m th terms:

$$\begin{aligned} \cos(m\psi) &= \cos\left(2m \frac{\psi}{2}\right) = T_{2m}(u) \\ \cos((m-1/2)\psi) &= \cos\left((2m-1) \frac{\psi}{2}\right) = T_{2m-1}(u) \end{aligned}$$

Thus in the odd case, the summation in Eq. (20.7.4) will result in a polynomial of maximal degree $2M = N - 1$ in the variable u , and in the even case, it will result into a polynomial of degree $2M - 1 = N - 1$.

The Dolph-Chebyshev [1068] array factor is defined by the Chebyshev polynomial of degree $N - 1$ in the scaled variable $x = x_0 \cos(\psi/2)$, that is,

$$W(\psi) = T_{N-1}(x), \quad x = x_0 \cos\left(\frac{\psi}{2}\right) \quad (\text{Dolph-Chebyshev array factor}) \quad (20.9.2)$$

The scale factor x_0 is always $x_0 > 1$ and is determined below. For a broadside design, as the azimuthal angle ϕ ranges over the interval $0^\circ \leq \phi \leq 180^\circ$, the wavenumber $\psi = kd \cos \phi$ will range over the visible region $-kd \leq \psi \leq kd$. The quantity $x = x_0 \cos(\psi/2)$ will range from $x_{\min} = x_0 \cos(kd/2)$ to the value $x = x_0$, which is reached broadside at $\phi = 90^\circ$ or $\psi = 0$, and then x will move back to x_{\min} . Thus, the range of variation of x will be $x_{\min} \leq x \leq x_0$.

Assuming that x_{\min} is in the interval $-1 \leq x_{\min} \leq 1$, we can split the interval $[x_{\min}, x_0]$ into the two subintervals: $[x_{\min}, 1]$ and $[1, x_0]$, as shown in Fig. 20.9.2. We require that the subinterval $[x_{\min}, 1]$ coincide with the sidelobe interval of the array factor $W(\psi)$, and that the subinterval $[1, x_0]$ coincide with the mainlobe interval. The zeros of the Chebyshev polynomial within $[x_{\min}, 1]$ become the sidelobe zeros of the array factor and get repeated twice as ϕ varies over $[0^\circ, 180^\circ]$.

In Fig. 20.9.2, for spacing $d = \lambda/2$, we have $kd = \pi$ and $x_{\min} = x_0 \cos(kd/2) = x_0 \cos(\pi/2) = 0$. Similarly, we have $x_{\min} = x_0 \cos(3\pi/4) = -0.707x_0$ for $d = 3\lambda/4$, and $x_{\min} = x_0 \cos(\pi/4) = 0.707x_0$ for $d = \lambda/4$.

The relative sidelobe attenuation level in absolute units and in dB is defined in terms of the ratio of the mainlobe to the sidelobe heights:

$$R_a = \frac{W_{\text{main}}}{W_{\text{side}}}, \quad R = 20 \log_{10}(R_a), \quad R_a = 10^{R/20}$$

Because the mainlobe peak occurs at $\psi = 0$ or $x = x_0$, we will have $W_{\text{main}} = T_{N-1}(x_0)$, and because the sidelobe level is equal to the Chebyshev level within $|x| \leq 1$, we will have $W_{\text{side}} = 1$. Thus, we find:

$$R_a = T_{N-1}(x_0) = \cosh((N-1) \operatorname{acosh}(x_0)) \quad (20.9.3)$$

which can be solved for x_0 in terms of R_a :

$$x_0 = \cosh\left(\frac{\operatorname{acosh}(R_a)}{N-1}\right) \quad (20.9.4)$$

Once the scale factor x_0 is determined, the window samples $w(m)$ can be computed by constructing the z-transform of the array factor from its zeros and then doing an inverse z-transform. The $N - 1$ zeros of $T_{N-1}(x)$ are easily found to be:

$$T_{N-1}(x) = \cos((N-1) \arccos(x)) = 0 \quad \Rightarrow \quad x_i = \cos\left(\frac{(i-1/2)\pi}{N-1}\right)$$

for $i = 1, 2, \dots, N - 1$. Solving for the corresponding wavenumbers through $x_i = x_0 \cos(\psi_i/2)$, we find the pattern zeros:

$$\psi_i = 2 \arccos\left(\frac{x_i}{x_0}\right), \quad z_i = e^{j\psi_i}, \quad i = 1, 2, \dots, N - 1$$

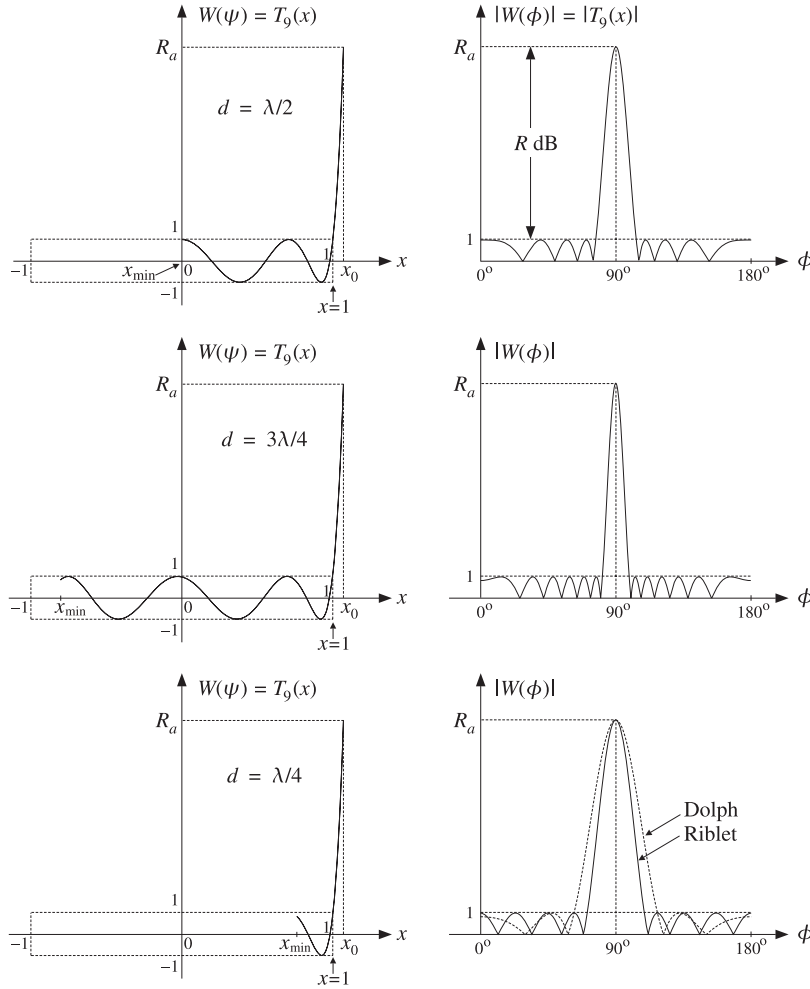


Fig. 20.9.2 Chebyshev polynomials and array factors for $d = \lambda/2$, $d = 3\lambda/4$, and $d = \lambda/4$.

We note that the zeros x_i do not have to lie within the sidelobe range $[x_{\min}, 1]$ and the corresponding ψ_i do not all have to be in the visible region.

The symmetric z-transform of the window is constructed in terms of the one-sided transform using Eq. (20.1.4) as follows:

$$W(z) = z^{-(N-1)/2} \tilde{W}(z) = z^{-(N-1)/2} \prod_{i=1}^{N-1} (z - z_i) \quad (20.9.5)$$

The inverse z-transform of $W(z)$ are the window coefficients $w(m)$. The MATLAB

function `dolph.m` of Appendix I implements this design procedure with the help of the function `poly2.m`, which calculates the coefficients from the zeros.[†] The typical MATLAB code in `dolph.m` is as follows:

```
N1 = N-1; % number of zeros
Ra = 10^(R/20); % sidelobe level in absolute units
x0 = cosh(acosh(Ra)/N1); % scaling factor

i = 1:N1;
xi = cos(pi*(i-0.5)/N1); % N1 zeros of Chebyshev polynomial
psi = 2 * acos(xi/x0); % N1 array pattern zeros in psi-space
zi = exp(j*psi); % N1 zeros of array polynomial

a = real(poly2(zi)); % zeros-to-polynomial form (N coefficients)
```

The window coefficients resulting from definition (20.9.5) are normalized to unity values at their end-points. This definition differs from that of Eq. (20.9.2) by the scale factor $x_0^{N-1}/2$.

The function `dolph.m` also returns the 3-dB width of the main lobe. The 3-dB frequency ψ_3 is defined by the half-power condition:

$$W(\psi_3) = T_{N-1}(x_3) = \frac{T_{N-1}(x_0)}{\sqrt{2}} = \frac{R_a}{\sqrt{2}} \Rightarrow \cosh((N-1)\cosh(x_3)) = \frac{R_a}{\sqrt{2}}$$

Solving for x_3 and the corresponding 3-dB angle, $x_3 = x_0 \cos(\psi_3/2)$, we have:

$$x_3 = \cosh\left(\frac{\cosh(R_a/\sqrt{2})}{N-1}\right), \quad \psi_3 = 2 \arccos\left(\frac{x_3}{x_0}\right) \quad (20.9.6)$$

which yields the 3-dB width in ψ -space, $\Delta\psi_{3dB} = 2\psi_3$. The 3-dB width in angle space, $\Delta\phi_{3dB}$, is then computed from Eq. (20.7.6) or (19.10.6).

There exist several alternative methods for calculating the Chebyshev array coefficients [1072-1080,1082] and have been compared in [1081]. One particularly accurate and effective method is that of Bresler [1075], which has recently been implemented by Simon [1077] with the MATLAB function `chebarray.m`.

Example 20.9.1: The second graph of Fig. 20.7.1 was generated by the MATLAB commands:

```
[a, dph] = dolph(0.5, 90, 5, 20); % array weights and 3-dB width
[g, ph] = array(0.5, a, 200); % compute array gain
dbz(ph, g, 45); % plot gain in dB
addcirc(3, 40, '--'); % add 3-dB gain circle
addray(90 + dph/2, '--'); % add 3-dB angles
addray(90 - dph/2, '--');
```

The array weights and 3-dB width were given previously in the table of Fig. 20.7.1. The weights are constructed as follows. The scale parameter x_0 is found to be $x_0 = 1.2933$. The zeros x_i , ψ_i , and z_i are found to be:

i	x_i	ψ_i	z_i
1	0.9239	1.5502	$0.0206 + 0.9998j$
2	0.3827	2.5408	$-0.8249 + 0.5653j$
3	-0.3827	3.7424	$-0.8249 - 0.5653j$
4	-0.9239	4.7330	$0.0206 - 0.9998j$

[†] See Sec. 6.8 regarding the accuracy of `poly2` versus `poly`.

It follows that the one-sided array polynomial will be:

$$\tilde{W}(z) = (z - z_1)(z - z_2)(z - z_3)(z - z_4) = z^4 + 1.6085z^3 + 1.9319z^2 + 1.6085z + 1$$

and the symmetric z -transform:

$$W(z) = z^{-2} \tilde{W}(z) = z^2 + 1.6085z + 1.9319 + 1.6085z^{-1} + z^{-2}$$

resulting in the array weights $\mathbf{w} = [1.0000, 1.6085, 1.9319, 1.6085, 1.0000]$. We note that the array zeros come in conjugate pairs. Only the first two x_i and ψ_i lie in the visible region and show up as pattern zeros in the array factor. \square

Example 20.9.2: The second graph of Fig. 20.7.2 was generated by the MATLAB commands:

```
[a, dph] = dolph(0.5, 60, 21, 25);
[g, ph] = array(0.5, a, 200);
dbz(ph, g);
```

The function `dolph.m` was called with the parameters $N = 21$, $R = 20$ dB and was steered towards the angle $\phi_0 = 60^\circ$. \square

Example 20.9.3: As another example, consider the design of a nine-element broadside Dolph-Chebyshev array with half-wavelength spacing and sidelobe attenuation level of $R = 20$ dB. The array factor is shown in Fig. 20.9.2.

The absolute attenuation level is $R_a = 10^{R/20} = 10^{20/20} = 10$, that is, if the peak is normalized to height $R_a = 10$, the sidelobes will have height of unity. The scale factor x_0 is found to be $x_0 = 1.0708$, and the array weights:

$$\mathbf{w} = [1.0000, 1.0231, 1.3503, 1.5800, 1.6627, 1.5800, 1.3503, 1.0231, 1.0000]$$

The array zeros are constructed as follows:

i	x_i	ψ_i	z_i
1	0.9808	0.8260	$0.6778 + 0.7352j$
2	0.8315	1.3635	$0.2059 + 0.9786j$
3	0.5556	2.0506	$-0.4616 + 0.8871j$
4	0.1951	2.7752	$-0.9336 + 0.3583j$
5	-0.1951	3.5080	$-0.9336 - 0.3583j$
6	-0.5556	4.2326	$-0.4616 - 0.8871j$
7	-0.8315	4.9197	$0.2059 - 0.9786j$
8	-0.9808	5.4572	$0.6778 - 0.7352j$

The 3-dB width is found from Eq. (20.9.6) to be $\Delta\phi_{3\text{dB}} = 12.51^\circ$. \square

In order for the Chebyshev interval $[x_{\min}, 1]$ to be mapped onto the sidelobe region of the array factor, we must require that $x_{\min} \geq -1$.

If $d < \lambda/2$, then this condition is automatically satisfied because $kd < \pi/2$ and $x_{\min} = x_0 \cos(kd/2) > 0$. (In this case, we must also demand that $x_{\min} \leq 1$. However, as we discuss below, when $d < \lambda/2$ Dolph's construction is no longer optimal and is replaced by the alternative procedure of Riblet.)

If $\lambda/2 < d < \lambda$, then $\pi < kd < 2\pi$ and $x_{\min} < 0$ and can exceed the left limit $x = -1$. This requires that for the given sidelobe level R , the array spacing may not exceed a *maximum* value that satisfies $x_{\min} = x_0 \cos(kd_{\max}/2) = -1$. This gives:

$$kd_{\max} = 2 \arccos\left(-\frac{1}{x_0}\right) \Rightarrow d_{\max} = \frac{\lambda}{\pi} \arccos\left(-\frac{1}{x_0}\right) \quad (20.9.7)$$

An alternative way of phrasing the condition $x_{\min} \geq -1$ is to say that for the given value of the array spacing d (such that $\lambda/2 < d < \lambda$), there is a *maximum sidelobe attenuation* that may be designed. The corresponding maximum value of x_0 will satisfy $x_{\min} = x_{0,\max} \cos(kd/2) = -1$, which gives:

$$x_{0,\max} = -\frac{1}{\cos(kd/2)} \Rightarrow R_{a,\max} = T_{N-1}(x_{0,\max}) \quad (20.9.8)$$

Example 20.9.4: Consider the case $d = 3\lambda/4$, $R = 20$ dB, $N = 9$. Then for the given R , the maximum element spacing that we can have is $d_{\max} = 0.8836\lambda$.

Alternatively, for the given spacing $d = 3\lambda/4$, the maximum sidelobe attenuation that we can have is $R_{a,\max} = 577$, or, $R_{\max} = 55.22$ dB.

An array designed with the maximum spacing $d = d_{\max}$ will have the narrowest mainlobe, because its total length will be the longest possible. For example, the following two calls to the function `dolph` will calculate the required 3-dB beamwidths:

```
[w, dph1] = dolph(0.75, 90, 9, 20); % spacing d = 3/4
[w, dph2] = dolph(0.8836, 90, 9, 20); % spacing d = dmax
```

We find $\Delta\phi_1 = 8.34^\circ$ and $\Delta\phi_2 = 7.08^\circ$. The array weights \mathbf{w} are the *same* in the two cases and equal to those of Example 20.9.3. The gains are shown in Fig. 20.9.3. \square

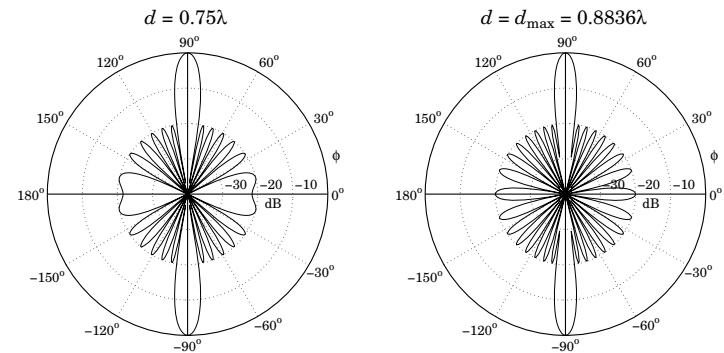


Fig. 20.9.3 Chebyshev arrays with $N = 9$, $R = 20$ dB, $d = 3\lambda/4$ and $d = 0.8836\lambda$.

As pointed out by Riblet [1069], Dolph's procedure is optimal only for element spacings that are greater than half a wavelength, $d \geq \lambda/2$. For $d < \lambda/2$, it is possible to find another set of window coefficients that would result into a narrower main lobe.

Riblet modified Dolph's method to obtain an optimal design for both cases, $d < \lambda/2$ and $d \geq \lambda/2$, but only for an *odd* number of array elements, $N = 2M + 1$.

It follows from Eq. (20.7.4) that if N is odd, the array factor $W(\psi)$ can be expressed either as a polynomial in the variable $\cos(\psi/2)$ or as a polynomial in the variable $\cos \psi$.

Dolph's original definition of Eq. (20.9.2) used a Chebyshev polynomial $T_{2M}(x)$ of order $2M = N - 1$ in the variable $x = x_0 \cos(\psi/2)$. Riblet used instead a Chebyshev polynomial $T_M(y)$ of order M in the new variable $y = A \cos \psi + B$, where the constants A, B are to be determined from the desired spacing d and sidelobe attenuation R . The array factor is defined as:

$$W(\psi) = T_M(y), \quad y = A \cos \psi + B \quad (\text{Riblet's modification}) \quad (20.9.9)$$

The mainlobe peak of height R_a at $\phi = 90^\circ$ (or $\psi = 0$) will correspond to a value y_0 such that:

$$R_a = T_M(y_0) = \cosh(M \operatorname{acosh}(y_0)) \quad (20.9.10)$$

which may be solved for y_0 :

$$y_0 = \cosh\left(\frac{\operatorname{acosh}(R_a)}{M}\right) \quad (20.9.11)$$

We note that y_0 is related to x_0 of Eq. (20.9.3) by $y_0 = 2x_0^2 - 1$. This follows from the general property of Chebyshev polynomials that:

$$y = 2x^2 - 1 \quad \Rightarrow \quad T_{2M}(x) = T_M(y) \quad (20.9.12)$$

Indeed, setting $x = \cos \theta$ and $y = \cos(2\theta) = 2\cos^2 \theta - 1 = 2x^2 - 1$, we have $\theta = \operatorname{acos}(x)$ and $2\theta = \operatorname{acos}(y)$, and therefore:

$$T_{2M}(x) = \cos((2M)\theta) = \cos(M(2\theta)) = T_M(y)$$

As the azimuthal angle ϕ varies over $0^\circ \leq \phi \leq 180^\circ$ and the wavenumber ψ over the visible region $-kd \leq \psi \leq kd$, the quantity $c = \cos \psi$ will vary from $c = \cos(kd)$ at $\phi = 0^\circ$ to $c = 1$ at $\phi = 90^\circ$, and then back to $c = \cos(kd)$ at $\phi = 180^\circ$.

If $\lambda/2 \leq d \leq \lambda$, then $\pi \leq kd \leq 2\pi$ and $\psi = kd \cos \phi$ will pass through the value $\psi = \pi$ before it reaches the value $\psi = kd$. It follows that the quantity c will go through $c = -1$ before it reaches $c = \cos(kd)$. Thus, in this case the widest range of variation of $c = \cos \psi$ is $-1 \leq c \leq 1$.

On the other hand, if $d < \lambda/2$, then $kd < \pi$ and c never reaches the value $c = -1$. Its minimum value is $c = \cos(kd)$, and the range of c is $[\cos(kd), 1]$. To summarize, the range of variation of c will be the interval $[c_0, 1]$, where

$$c_0 = \begin{cases} -1, & \text{if } d \geq \lambda/2 \\ \cos(kd), & \text{if } d < \lambda/2 \end{cases} \quad (20.9.13)$$

Assuming $A > 0$, it follows that the range of variation of $y = A \cos \psi + B$ will be the interval $[Ac_0 + B, A + B]$. The parameters A, B are fixed by requiring that this interval coincide with the interval $[-1, y_0]$ so that the right end will correspond to the mainlobe

peak, while the left end will ensure that we use the maximum size of the equi-ripple interval of the Chebyshev variable y . Thus, we require the conditions:

$$\begin{aligned} Ac_0 + B &= -1 \\ A + B &= y_0 \end{aligned} \quad (20.9.14)$$

which may be solved for A, B :

$$\begin{aligned} A &= \frac{1 + y_0}{1 - c_0} \\ B &= -\frac{1 + y_0 c_0}{1 - c_0} \end{aligned} \quad (20.9.15)$$

For $d \geq \lambda/2$, the method coincides with Dolph's original method. In this case, $c_0 = -1$, and A, B become:

$$\begin{aligned} A &= \frac{y_0 + 1}{2} = x_0^2 \\ B &= \frac{y_0 - 1}{2} = x_0^2 - 1 \end{aligned} \quad (20.9.16)$$

where we used $y_0 = 2x_0^2 - 1$, as discussed above. It follows that the y variable will be related to the Dolph variable $x = x_0 \cos(\psi/2)$ by:

$$y = x_0^2 \cos \psi + x_0^2 - 1 = x_0^2 (\cos \psi + 1) - 1 = 2x_0^2 \cos^2\left(\frac{\psi}{2}\right) - 1 = 2x^2 - 1$$

and therefore, Eq. (20.9.12) implies that $W(\psi) = T_M(y) = T_{2M}(x)$.

Once the parameters A, B are determined, the window $w(m)$ may be constructed from the zeros of the Chebyshev polynomials. The M zeros of $T_M(y)$ are:

$$y_i = \cos\left(\frac{(i - 1/2)\pi}{M}\right), \quad i = 1, 2, \dots, M$$

The corresponding wavenumbers are found by inverting $y_i = A \cos \psi_i + B$:

$$\psi_i = \operatorname{acos}\left(\frac{y_i - B}{A}\right), \quad i = 1, 2, \dots, M$$

The $2M = N - 1$ zeros of the z-transform of the array are the conjugate pairs:

$$\{e^{j\psi_i} \ e^{-j\psi_i}\}, \quad i = 1, 2, \dots, M$$

The symmetrized z-transform will be then:

$$W(z) = z^{-M} \tilde{W}(z) = z^{-M} \prod_{i=1}^M ((z - e^{j\psi_i})(z - e^{-j\psi_i}))$$

The inverse z-transform of $W(z)$ will be the desired array weights $w(m)$. This procedure is implemented by the MATLAB function `dolph2.m` of Appendix I. We note again that this definition differs from that of Eq. (20.9.9) by the scale factor $A^M/2$.

The function `dolph2` also returns the 3-dB width of the main lobe. The 3-dB frequency ψ_3 is computed from the half-power condition:

$$W(\psi_3) = T_M(y_3) = \frac{T_M(y_0)}{\sqrt{2}} = \frac{R_a}{\sqrt{2}} \Rightarrow y_3 = \cosh\left(\frac{\operatorname{acosh}(R_a/\sqrt{2})}{M}\right)$$

Inverting $y_3 = A \cos \psi_3 + B$, we obtain the 3-dB width in ψ -space:

$$\psi_3 = \operatorname{acos}\left(\frac{y_3 - B}{A}\right), \quad \Delta\psi_{3\text{dB}} = 2\psi_3 \quad (20.9.17)$$

For the case $d \geq \lambda/2$, the maximum element spacing given by Eq. (20.9.7) can also be expressed in terms of the variable y_0 as follows:

$$d_{\max} = \lambda \left[1 - \frac{1}{2\pi} \operatorname{acos}\left(\frac{3 - y_0}{1 + y_0}\right) \right] \quad (20.9.18)$$

This follows from the condition $x_{\min} = x_0 \cos(kd_{\max}/2) = -1$. The corresponding value of y will be $y = 2x_{\min}^2 - 1 = 1$. Using Eq. (20.9.16), this condition reads:

$$y = \frac{y_0 + 1}{2} \cos(kd_{\max}) + \frac{y_0 - 1}{2} = 1 \Rightarrow \cos(kd_{\max}) = \frac{3 - y_0}{1 + y_0}$$

Because the function acos always returns a value in the range $[0, \pi]$, and we want a value $kd_{\max} > \pi$, we must invert the cosine as follows:

$$kd_{\max} = 2\pi - \operatorname{acos}\left(\frac{3 - y_0}{1 + y_0}\right)$$

which implies Eq. (20.9.18).

Example 20.9.5: The bottom two graphs of Fig. 20.9.2 show the array factor designed using Dolph's and Riblet's methods for the case $N = 9$, $R = 20$ dB, and $d = \lambda/4$. The Dolph weights are the same as those given in Example 20.9.3. The Riblet weights computed by `dolph2` are:

$$\mathbf{w} = [1, -3.4884, 7.8029, -11.7919, 13.6780, -11.7919, 7.8029, -3.4884, 1]$$

The corresponding array gains in dB are shown in Fig. 20.9.4. The 3-dB widths of the Dolph and Riblet designs are $\Delta\phi_{3\text{dB}} = 25.01^\circ$ and $\Delta\phi_{3\text{dB}} = 17.64^\circ$. \square

Next, we discuss steered arrays [1070]. We assume a steering angle $0^\circ < \phi_0 < 180^\circ$. The endfire case $\phi_0 = 0^\circ, 180^\circ$ will be treated separately [1071]. The steered wavenumber will be:

$$\psi' = \psi - \psi_0 = kd(\cos \phi - \cos \phi_0) \quad (20.9.19)$$

where $\psi_0 = kd \cos \phi_0$. The corresponding array weights and array factor will be:

$$\begin{aligned} a(m) &= e^{-jm\psi_0} w(m), \quad -M \leq m \leq M \\ A(\psi) &= W(\psi - \psi_0) = W(\psi') = T_M(y'), \quad y' = A \cos \psi' + B \end{aligned} \quad (20.9.20)$$

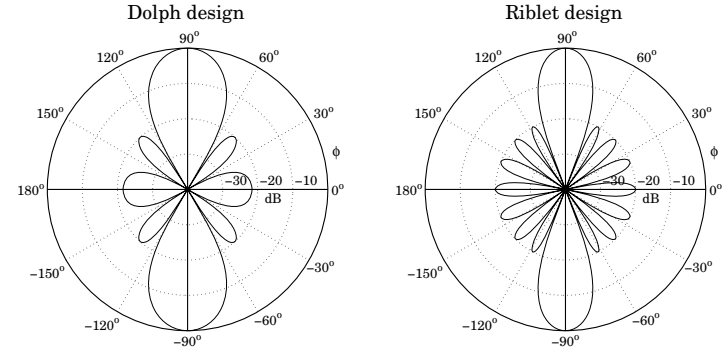


Fig. 20.9.4 Dolph and Riblet designs of Chebyshev array with $N = 9$, $R = 20$ dB, $d = \lambda/4$.

where we assumed that N is odd, $N = 2M + 1$. The visible region becomes now:

$$kd(1 - |\cos \phi_0|) \leq \psi' \leq kd(1 + |\cos \phi_0|)$$

In order to avoid grating lobes, the element spacing must be less than the maximum:

$$d_0 = \frac{\lambda}{1 + |\cos \phi_0|} \quad (20.9.21)$$

which satisfies $kd_0(1 + |\cos \phi_0|) = 2\pi$.

The Chebyshev design method is carried out in the same way, except instead of using the half-wavelength spacing $\lambda/2$ as the dividing line between the Riblet and the Dolph methods, we must use $d_0/2$. Thus, the variable $c = \cos \psi' = \cos(\psi - \psi_0)$ will vary in the interval $[c_0, 1]$, where Eq. (20.9.13) is now replaced by

$$c_0 = \begin{cases} -1, & \text{if } d \geq d_0/2 \\ \cos(kd(1 + |\cos \phi_0|)), & \text{if } d < d_0/2 \end{cases}$$

Replacing $1 + |\cos \phi_0| = \lambda/d_0$, we can rewrite this as follows:

$$c_0 = \begin{cases} -1, & \text{if } d \geq d_0/2 \\ \cos\left(\frac{2\pi d}{d_0}\right), & \text{if } d < d_0/2 \end{cases} \quad (20.9.22)$$

The solutions for A, B will still be given by Eq. (20.9.15) with this new value for c_0 . Note that when $d < d_0/2$ the quantities A, B , and hence the array weights $w(m)$, will depend on ϕ_0 . Therefore, the weights must be redesigned for each new value of ϕ_0 , instead of simply steering the broadside weights [1070].

When $d \geq d_0/2$, we have $c_0 = -1$ and the weights $w(m)$ become *independent* of ϕ_0 . In this case, the steered weights are obtained by steering the broadside weights.

Example 20.9.6: Fig. 20.9.5 shows the gain of an array steered towards $\phi_0 = 60^\circ$, with $N = 9$, $R = 20$ dB, and element spacing $d = \lambda/4$.

The grating lobe spacing is $d_0 = \lambda / (1 + \cos(60^\circ)) = 2\lambda/3$, and the dividing line between Dolph and Riblet designs will be $d_0/2 = \lambda/3$. The second graph shows the gain of a broadside array, which is steered towards 60° . It demonstrates that the plain steering of a broadside design will not work for $d < d_0/2$. The array weights were computed by the MATLAB commands:

```
a1 = dolph2(1/4, 60, 9, 20);    % steered array
w  = dolph2(1/4, 90, 9, 20);    % broadside array
a2 = steer(1/4, w, 60);        % steered broadside array
```

The 3-dB width was $\Delta\phi_{3\text{dB}} = 26.66^\circ$. It was obtained using Eq. (20.9.17) and the approximation Eq. (20.7.6). The first graph also shows the 3-dB gain circle intersecting the rays at the 3-dB angles $\phi_0 \pm \Delta\phi_{3\text{dB}}/2$, that is, at 46.67° and 73.33° . We note also that the broadside weights **w** were given in Example 20.9.5. \square

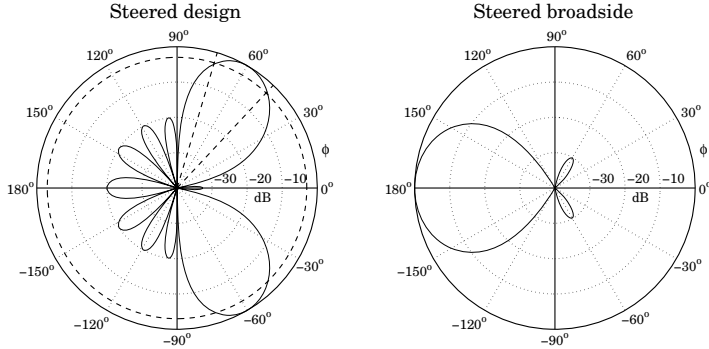


Fig. 20.9.5 Nine-element array with $d = \lambda/4$ steered towards 60° .

Endfire Dolph-Chebyshev arrays require special treatment. DuHamel has shown how to modify Riblet's design for this purpose [1071]. The key idea is not to use a steering angle $\phi_0 = 0^\circ$ or $\phi_0 = 180^\circ$, but rather to make ϕ_0 , and the corresponding steering phase $\psi_0 = kd \cos \phi_0$, a free design parameter.

The steered wavenumber will still be $\psi' = kd \cos \phi - kd \cos \phi_0 = kd \cos \phi - \psi_0$ and the array factor and array weights will still be given by Eq. (20.9.20).

The three parameters $\{A, B, \psi_0\}$ are determined by the following conditions. For a forward endfire array (with mainlobe peak towards $\phi = 0^\circ$), we require that $y' = y_0$ at $\phi = 0$, or, at $\psi' = kd - \psi_0$. Moreover, we require that the two endpoints $y' = -1$ and $y' = 1$ of the equi-ripple range of the Chebyshev polynomial are reached at $\psi' = 0$ and at $\phi = 180^\circ$, or, $\psi' = -kd - \psi_0$. These three conditions can be stated as follows:

$$\begin{aligned} A \cos(kd - \psi_0) + B &= y_0 \\ A + B &= -1 \\ A \cos(kd + \psi_0) + B &= 1 \end{aligned} \quad (20.9.23)$$

For a backward endfire array (with mainlobe towards $\phi = 180^\circ$), we must replace ψ_0 by $-\psi_0$. The solution of Eqs. (20.9.23) is:

$$\begin{aligned} A &= -\frac{y_0 + 3 + 2 \cos(kd) \sqrt{2(y_0 + 1)}}{2 \sin^2(kd)} \\ B &= -1 - A \\ \psi_0 &= \pm \arcsin\left(\frac{y_0 - 1}{2A \sin(kd)}\right) \end{aligned} \quad (20.9.24)$$

where in the solution for ψ_0 , the plus (minus) sign is chosen for the forward (backward) endfire array. Bidirectional endfire arrays can also be designed. In that case, we set $\psi_0 = 0$ and only require the first two conditions in (20.9.23), which become

$$\begin{aligned} A \cos(kd) + B &= y_0 \\ A + B &= -1 \end{aligned} \quad (20.9.25)$$

with solution:

$$\begin{aligned} A &= -\frac{y_0 + 1}{1 - \cos(kd)} \\ B &= \frac{y_0 + \cos(kd)}{1 - \cos(kd)} \end{aligned} \quad (20.9.26)$$

In all three of the above endfire designs, we must assume $d \leq \lambda/2$ in order to avoid grating lobes. The MATLAB function `dolph3.m` of Appendix I implements all three cases.

Example 20.9.7: Fig. 20.9.6 shows three endfire designs for a nine-element array with quarter-wavelength spacing $d = \lambda/4$, and sidelobe level of $R = 20$ dB. The array weights and 3-dB widths were computed as follows:

```
[a1, dph1] = dolph3(1, 1/4, 9, 20);    % forward endfire
[a2, dph2] = dolph3(-1, 1/4, 9, 20);   % backward endfire
[a3, dph3] = dolph3(2, 1/4, 9, 20);    % bidirectional endfire
```

The first argument of `dolph3` takes on one of the three values $\{1, -1, 2\}$, for forward, backward, and bidirectional designs. In the forward and backward cases, the array weights are already scanned by the effective scanning phase $\pm\psi_0$. The calculated array weights are in the three cases:

weights	forward	backward	bidirectional
a_0	18.3655	18.3655	20.4676
$a_1 = a_{-1}^*$	$-15.8051 - 1.0822j$	$-15.8051 + 1.0822j$	-17.5583
$a_2 = a_{-2}^*$	$9.8866 + 1.3603j$	$9.8866 - 1.3603j$	10.8723
$a_3 = a_{-3}^*$	$-4.1837 - 0.8703j$	$-4.1837 + 0.8703j$	-4.5116
$a_4 = a_{-4}^*$	$0.9628 + 0.2701j$	$0.9628 - 0.2701j$	1.0000

Because the backward case is obtained by the replacement $\psi_0 \rightarrow -\psi_0$, its weights will be the conjugates of those of the forward case.

The 3-dB widths are in the three cases: $\Delta\phi_{3\text{dB}} = 22.85^\circ, 22.85^\circ, 22.09^\circ$. The graphs also show the 3-dB gain circles intersecting the gains at the 3-dB angles. \square

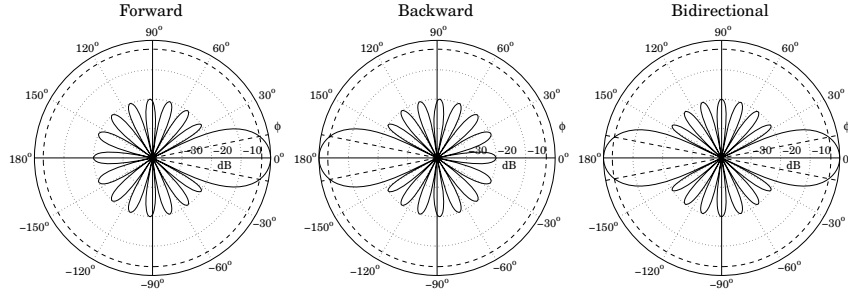


Fig. 20.9.6 Forward, backward, and bidirectional endfire designs.

20.10 Taylor One-Parameter Source

In Sec. 20.4, we used the Kaiser window to design a sector array pattern. That design problem was equivalent to designing an FIR lowpass digital filter using the window method. Here, we use the Kaiser window to design a narrow beam array—a problem equivalent to the spectral analysis of windowed sinusoids [47,48,1090].

The broadside array weights are equal to the window coefficients $a(m) = w(m)$, defined up to an overall normalization constant by:

$$w(m) = I_0 \left(\alpha \sqrt{1 - m^2/M^2} \right) \quad (20.10.1)$$

where $m = \pm 1, \pm 2, \dots, \pm M$, or $m = 0, \pm 1, \pm 2, \dots, \pm M$, for even or odd number of array elements, $N = 2M$ or $N = 2M + 1$.

This window is based on Taylor's one-parameter continuous line source [1085], and is obtained by setting $x_m = md$ with $d = l/(2M)$ in Eq. (20.6.20), so that $2x_m/l = m/M$,

$$I(x_m) = I_0 \left(\pi B \sqrt{1 - (2x_m/l)^2} \right) = I_0 \left(\pi B \sqrt{1 - (m/M)^2} \right)$$

Thus, we note that the Kaiser window shape parameter α is related to Taylor's parameter B by $\alpha = \pi B$. The parameter B or α control the sidelobe level. The continuous line pattern of (20.6.20),

$$F(u) = \frac{\sinh(\pi \sqrt{B^2 - u^2})}{\pi \sqrt{B^2 - u^2}} = \frac{\sin(\pi \sqrt{u^2 - B^2})}{\pi \sqrt{u^2 - B^2}} \quad (20.10.2)$$

has a first null at $u_0 = \sqrt{B^2 + 1}$, and therefore, the first sidelobe will occur for $u > u_0$. For this range, we must use the sinc-form of $F(u)$ and to find the maximum sidelobe level, we must find the maximum of the sinc function (for argument other than zero). This can be determined, for example, by the MATLAB command:[†]

```
x0=fminbnd('sinc(x)', 1,2, optimset('TolX',eps)); r0 = abs(sinc(x0));
```

[†]MATLAB's sinc function is defined as $\text{sinc}(x) = \sin \pi x / \pi x$.

20.10. Taylor One-Parameter Source

which yields the values:

$$\begin{aligned} x_0 &= 1.4302966532 \\ r_0 &= |\text{sinc}(x_0)| = 0.2172336282 \\ R_0 &= -20 \log_{10}(r_0) = 13.2614588840 \text{ dB} \end{aligned} \quad (20.10.3)$$

The sidelobe level R_a (in absolute units) is defined as the ratio of the pattern at $u = 0$ to the maximum sidelobe level r_0 , that is,

$$R_a = \frac{1}{r_0} \frac{\sinh(\pi B)}{\pi B} \quad (20.10.4)$$

and in dB, $R = 20 \log_{10}(R_a)$,

$$R = R_0 + 20 \log_{10} \left(\frac{\sinh(\pi B)}{\pi B} \right) \quad (20.10.5)$$

To avoid having to solve (20.10.4) for B for a given R_a , Kaiser and Schafer [1090] have developed an empirical formula in terms of the sidelobe level R in dB, which is valid across the range $13 < R < 120$ dB:

$$\pi B = \begin{cases} 0, & R \leq 13.26 \\ 0.76609(R - 13.26)^{0.4} + 0.09834(R - 13.26), & 13.26 < R \leq 60 \\ 0.12438(R + 6.3), & 60 < R < 120 \end{cases} \quad (20.10.6)$$

For $R \leq 13.26$, $w(m)$ becomes the rectangular window. The broadening factor b , and the 3-dB width in ψ -space can also be expressed in terms of the dB sidelobe level R by the following empirical formula valid for $20 < R < 100$ dB:

$$b = 0.01330R + 0.9761, \quad \Delta\psi_{3\text{dB}} = 0.886 \frac{2\pi b}{N} \quad (20.10.7)$$

The 3-dB width in angle space, $\Delta\phi_{3\text{dB}}$, is then calculated from Eq. (20.7.6). The 3-dB beam width may be more accurately calculated by finding it in u -space, say Δu , and then transforming it to ψ -space using Eq. (20.6.15), $\Delta\psi_{3\text{dB}} = 2\pi\Delta u/N$. The width Δu is given by $\Delta u = 2u_3$, where u_3 is the solution of the half-power condition:

$$|F(u_3)|^2 = \frac{1}{2} |F(0)|^2 \Rightarrow \frac{\sinh(\pi \sqrt{B^2 - u_3^2})}{\pi \sqrt{B^2 - u_3^2}} = \frac{1}{\sqrt{2}} \frac{\sinh(\pi B)}{\pi B} \quad (20.10.8)$$

For small values of B , the right-hand side becomes less than one, and we must switch the left-hand side to its sinc form. This happens when $B \leq B_c$, where

$$\frac{1}{\sqrt{2}} \frac{\sinh(\pi B_c)}{\pi B_c} = 1 \Rightarrow B_c = 0.4747380492 \quad (20.10.9)$$

which, through (20.10.5), corresponds to a sidelobe attenuation of $R_c = 16.27$ dB. Rather than using the above empirical formulas, Eqs. (20.10.4) and (20.10.8) may be solved numerically in MATLAB. The function `taylorbw` implements the solution, returning the values of B and Δu , for any vector of sidelobe attenuations R :

```
[B,Du] = taylorbw(R); % Taylor parameter B and beamwidth Δu
```

It is built on the functions `sinhc` and `asinhc` for computing the hyperbolic sinc function and its inverse:

```
y = sinhc(x); % hyperbolic sinc function, sinhc(x) = sinh(πx)/πx
x = asinhc(y); % inverse function, finds the x that satisfies sinhc(x) = y
```

For small x , the equation $y = \sinh(x)/x$ is solved for x by using the Taylor series expansion $y = \sinh(x)/x \approx 1 + x^2/6 + x^4/120$; for larger x , it is solved by the iteration $\sinh(x_n)/x_{n-1} = y$, or $x_n = \text{asinh}(yx_{n-1})$, for $n = 1, 2, \dots$

Once the B -parameter is determined, the array weights $w(m)$ can be computed from (20.10.1) using the built-in function `besselj`, and then steered towards an angle ϕ_0 using Eq. (20.7.1). In this case, to avoid grating lobes, the element spacing must be less than the maximum:

$$d_0 = \frac{\lambda}{1 + |\cos \phi_0|} \quad (20.10.10)$$

As discussed in Sec. 20.9, in order for the visible region in ψ -space to cover at least one Nyquist period, the element spacing d must be in the range:

$$\frac{d_0}{2} \leq d < d_0 \quad (20.10.11)$$

The MATLAB function `taylor1p` of Appendix I implements this design procedure and invokes the function `taylorbw`. The outputs of the function are the steered array weights and the 3-dB width. It has usage:

```
[a, dph] = taylor1p(d, ph0, N, R); % Taylor 1-parameter line source
```

Example 20.10.1: Fig. 20.10.1 depicts the gain of a 14- and a 15-element Taylor-Kaiser array with half-wavelength spacing $d = \lambda/2$, steered towards $\phi_0 = 60^\circ$. The sidelobe level was $R = 20$ dB. The array weights were obtained by:

```
[a1, dph1] = taylor1p(0.5, 60, 14, 20);
[a2, dph2] = taylor1p(0.5, 60, 15, 20);
```

The graphs in Fig. 20.10.1 can be produced by the following commands:

```
[g1, ph1] = gain1d(0.5, a1, 720); % compute normalized gain at 720 angles
dbz(ph1, g1); % make azimuthal plot of the gain
addcirc(3); % add 3-dB circle
addray(60-dph1/2); addray(60+dph1/2); % add rays at 3-dB angles
```

The array weights are already steered towards 60° . The designed unsteered weights were in the two cases:

```
w1 = [1.0000, 1.3903, 1.7762, 2.1339, 2.4401, 2.6749, 2.8224,
      2.8224, 2.6749, 2.4401, 2.1339, 1.7762, 1.3903, 1.0000]
```

```
w2 = [1.0000, 1.3903, 1.7762, 2.1339, 2.4401, 2.6749, 2.8224, 2.8728,
      2.8224, 2.6749, 2.4401, 2.1339, 1.7762, 1.3903, 1.0000]
```

The corresponding 3-dB widths were $\Delta\phi_{3\text{dB}} = 9.68^\circ$ and $\Delta\phi_{3\text{dB}} = 9.03^\circ$, with the second being slightly narrower because the array is slightly longer. The graphs show how the rays at the two 3-dB angles intersect the 3-dB gain circles. The maximum and minimum array spacings are from (20.10.10): $d_0 = 2\lambda/3$ and $d_0/2 = \lambda/3$. □

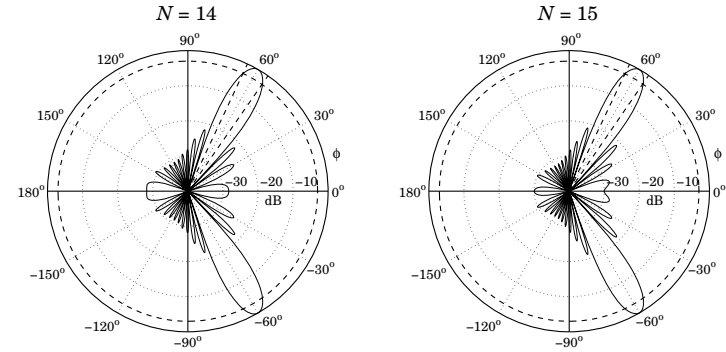


Fig. 20.10.1 Taylor-Kaiser arrays with $N = 14$ and $N = 15$, and $d = \lambda/2$.

Example 20.10.2: Fig. 20.10.2 depicts the gain of a 31-element endfire array with spacing $d = \lambda/4$ and sidelobe level $R = 20$ dB, steered towards the forward direction, $\phi_0 = 0^\circ$, and the backward one, $\phi_0 = 180^\circ$.

The maximum and minimum array spacings, calculated from Eq. (20.10.10) for $\phi_0 = 0^\circ$ and $\phi_0 = 180^\circ$, are $d_0 = \lambda/2$ and $d_0/2 = \lambda/4$. We have chosen $d = d_0/2 = \lambda/4$.

The 3-dB widths are in both cases $\Delta\phi_{3\text{dB}} = 43.12^\circ$. The graphs also show the 3-dB circle intersecting the 3-dB angle rays. □

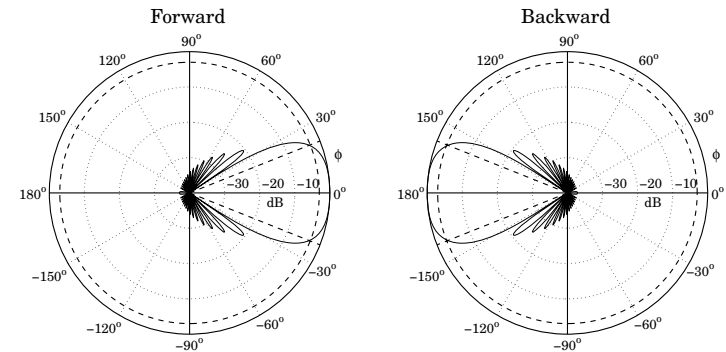


Fig. 20.10.2 Taylor-Kaiser endfire arrays with $N = 31$ and $d = \lambda/4$.

The design method of the section was based on sampling the current distribution directly, as in Eq. (20.10.1), rather than using the procedure of mapping the pattern

zeros as outlined in Eqs. (20.6.15)–(20.6.17). A variation of the design method array that uses the latter procedure is discussed in Problem 20.1.

20.11 Prolate Array

Kaiser has noted [1089,1090] that the Kaiser window function (20.10.1) is an excellent approximation to the 0th order discrete prolate spheroidal sequence that maximizes the energy concentration in a given frequency interval [1092–1101].

Using the prolate sequence as a window for array design provides a slight improvement over the Taylor-Kaiser case in the sense of having a slightly narrower beamwidth while meeting the sidelobe specification more precisely. The prolate array can be designed very quickly using the inverse power iteration.

Given an N -dimensional array $\mathbf{a} = [a_0, a_1, \dots, a_{N-1}]^T$ with array pattern $A(\psi)$, the discrete prolate spheroidal performance index which measures the concentration of energy within the wavenumber interval $[-\psi_c, \psi_c]$ is defined by:

$$\mathcal{J} = \frac{\frac{1}{2\pi} \int_{-\psi_c}^{\psi_c} |A(\psi)|^2 d\psi}{\frac{1}{2\pi} \int_{-\pi}^{\pi} |A(\psi)|^2 d\psi} \quad (20.11.1)$$

The integration range in the denominator may be changed to be the visible region $[-kd, kd]$ if so desired [1094], but the design technique remains essentially the same. Inserting the array pattern,

$$A(\psi) = \sum_{n=0}^{N-1} a_n e^{j\psi n} \quad (20.11.2)$$

into (20.11.1), we may express the performance index \mathcal{J} as a Rayleigh quotient involving the so-called *prolate matrix* [1093,1101]:

$$\mathcal{J} = \frac{\mathbf{a}^\dagger \mathbf{A} \mathbf{a}}{\mathbf{a}^\dagger \mathbf{a}} \quad (20.11.3)$$

where the dagger denotes the Hermitian conjugate and the prolate matrix is defined by its matrix elements:

$$A_{nm} = \frac{\sin(\psi_c(n-m))}{\pi(n-m)} = \frac{\sin(2\pi W(n-m))}{\pi(n-m)}, \quad n, m = 0, 1, \dots, N-1 \quad (20.11.4)$$

where we set $\psi_c = 2\pi W$ for later convenience. This matrix is the convolution matrix arising from the impulse response of the ideal lowpass filter with cutoff ψ_c .

The problem of maximum energy concentration is to find that finite sequence \mathbf{a} that maximizes the performance index \mathcal{J} . This problem has been studied extensively both for discrete and continuous time sequences, see [1095] for a nice review. The maximization of the Rayleigh quotient is realized by the maximum eigenvector of the prolate matrix \mathbf{A} , that is, the eigenvector belonging to the maximum eigenvalue, say, λ_0 :

$$\mathbf{A} \mathbf{a} = \lambda_0 \mathbf{a} \quad (20.11.5)$$

20.11. Prolate Array

The prolate matrix is notoriously ill-conditioned having approximately $2NW$ eigenvalues that are very near one, and the remaining eigenvalues decreasing rapidly to zero. The following table lists the eigenvalues in decreasing order for the case $N = 21$ and $W = 0.2$, so that $2NW = 8.4$, its condition number being, $\text{cond}(\mathbf{A}) = 5.1063 \times 10^{16}$:

i	λ_i	i	λ_i
0	0.99999999998517786000	11	0.00131552671490021500
1	0.999999999795514627000	12	0.00007986915605618046
2	0.99999987170540139000	13	0.00000365494381482577
3	0.99999517388508363000	14	0.00000012731149204486
4	0.99987947149714795000	15	0.00000000336154097643
5	0.99792457099956200000	16	0.00000000006621668668
6	0.97588122145542644000	17	0.00000000000094327944
7	0.83446090480119717000	18	0.00000000000000920186
8	0.45591142240913063000	19	0.00000000000000004034
9	0.11887181858959120000	20	0.00000000000000001958
10	0.01567636516215985600		

These were generated by the following MATLAB code:

```
N = 21; W = 0.2;
n = 0:N-1; f = 2*W*sinc(2*W*n);
A = toeplitz(f,f);
[la = svd(A);
```

The eigenvectors of the prolate matrix are referred to as the *discrete prolate spheroidal sequences* (DPSS), and the first $2NW$ of them are relevant in multitaper methods of spectral analysis [1100].

For the array problem, we are interested only in the maximum eigenvector. A simple way to compute it is by the power iteration, that is, $\mathbf{a}_n = \mathbf{A} \mathbf{a}_{n-1} = \mathbf{A}^n \mathbf{a}_0$. However, because the corresponding eigenvalue λ_0 and the next highest one are so close to unity, the iteration will be very slow converging.

A more efficient approach is to apply the inverse power iteration on the matrix $\mathbf{Q} = \mathbf{I} - \mathbf{A}$, that is, $\mathbf{a}_{n+1} = \mathbf{Q}^{-1} \mathbf{a}_n = \mathbf{Q}^{-n} \mathbf{a}_0$. This iteration converges to the minimum eigenvector of \mathbf{Q} , which is the same as the maximum eigenvector of \mathbf{A} . The minimum eigenvalue of \mathbf{Q} is $1 - \lambda_0$, which is very small and its inverse $(1 - \lambda_0)^{-1}$ very large, causing the iteration to converge very fast.

For the array problem one needs to know the relationship of the bandwidth parameter W to the desired sidelobe level R and the array length N . Because the Taylor-Kaiser window is a good approximation to the maximum eigenvector, one expects to have a relationship among the parameters W, B, N, R . As pointed out by Kaiser and Walden [1089,1097], this relationship is approximately $W = B/N$.

We have improved this relationship slightly by using the results of [1101] to arrive at the following empirical formula, which works well over the range $14 \leq R \leq 120$ dB:

$$W = \frac{0.95B + 0.14}{N} \quad (20.11.6)$$

This leads to the following design procedure. Given N and R , we calculate B using the function `tailorbw`, described in Sec. 20.10, then calculate W from (20.11.6), and

construct the prolate matrix A , and $Q = I - A$, then, apply the inverse power iteration initialized with the unit vector $\mathbf{a}_0 = [1, 0, 0, \dots, 0]^T$:

$$\begin{aligned} \mathbf{a} &= [1, 0, 0, \dots, 0]^T \\ \text{for } n &= 1, 2, \dots, N_{\text{iter}}, \\ \mathbf{a} &= Q^{-1}\mathbf{a} \end{aligned}$$

The algorithm is insensitive to the choice of the initial vector \mathbf{a}_0 and converges extremely fast, requiring about 1-3 iterations (we use 3 by default).

We determine the 3-dB width by simply equating it to that of the Taylor array, that is, $\Delta\psi = 2\pi\Delta u/N$, where Δu is also obtained from the function `taylorbw`. Even so, the prolate array's mainlobe, as a whole, is slightly narrower than that of the Taylor array. The MATLAB function `pro1` implements the above procedure:

```
[a, dph] = pro1(d, ph0, N, R); % prolate array
```

The function `pro1mat` constructs the prolate matrix for given N, W :

```
A = pro1mat(N, W); % prolate matrix
```

Fig. 20.11.1 shows a design example with $N = 21$ and $R = 30$. The left graph plots the array patterns $|A(\psi)|$ for the prolate and Taylor designs at broadside. The right graph shows the same prolate array steered towards 60° . The 3-dB width is also indicated on the figure. We note that the Taylor array has a slightly wider mainlobe and slightly lower sidelobes, whereas the prolate design meets the sidelobe specification exactly. The graphs were computed by the following MATLAB code:

```
d = 0.5; N = 21; R = 30; ph0 = 90;

[at, Dt] = taylor1p(d, ph0, N, R); % Taylor design
[ap, Dp] = pro1(d, ph0, N, R); % Prolate design

f = linspace(-1, 1, 1001); psi = pi*f; % normalized wavenumber ψ

At = 20*log10(abs(dtft(at, -psi))); At = At - max(At); % compute pattern
Ap = 20*log10(abs(dtft(ap, -psi))); Ap = Ap - max(Ap);

figure; plot(f, Ap, '-', f, At, '--');

ap = steer(d, ap, 60); % redesign steered towards 60°
[gp, phi] = gain1d(d, ap, 720); % normalized gain

figure; dbz(phi, gp);
```

Fig. 20.11.2 compares the corresponding array weights of the prolate and Taylor designs for the cases $N = 21$ and $N = 41$, and $R = 30$ dB.

20.12 Taylor Line Source

Taylor's ideal line source pattern [1086], given in Eq. (20.6.21), has a mainlobe when $|u| \leq A$ and equiripple sidelobes when $|u| \geq A$. The equiripple behavior arises from the fact that the pattern switches to its cosine form when $|u| \geq A$, as shown below:

20.12. Taylor Line Source

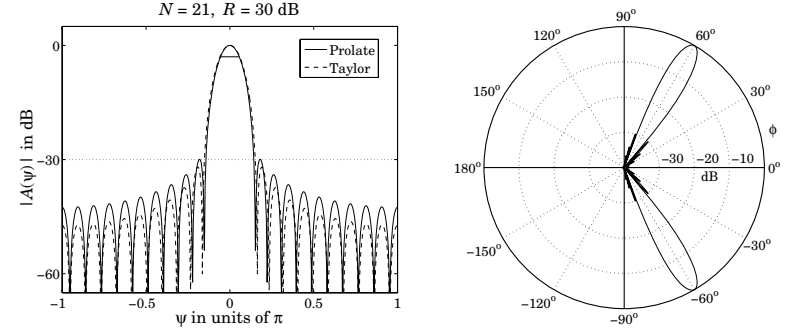


Fig. 20.11.1 Prolate array design.

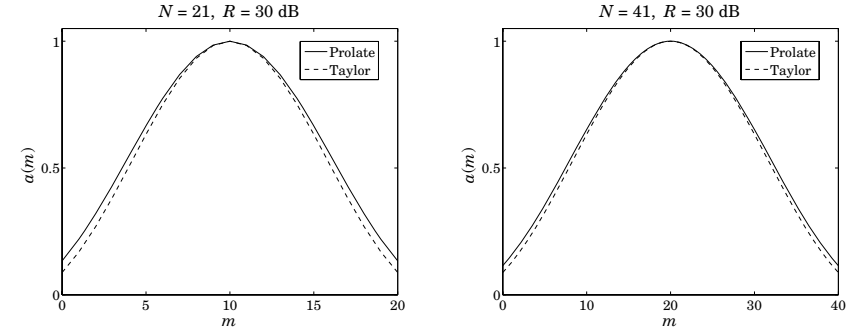
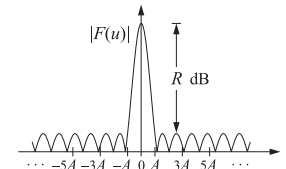


Fig. 20.11.2 Comparison of prolate and Taylor array weights.



$$F(u) = \begin{cases} \cosh(\pi\sqrt{A^2 - u^2}), & \text{if } |u| \leq A \\ \cos(\pi\sqrt{u^2 - A^2}), & \text{if } |u| \geq A \end{cases} \quad (20.12.1)$$

The sidelobe level (in absolute units) is the ratio of the mainlobe peak height $|F(0)| = \cosh(\pi A)$ to the sidelobe height, which is unity:

$$R_a = \cosh(\pi A) \Rightarrow A = \frac{1}{\pi} \operatorname{acosh}(R_a) \quad (20.12.2)$$

The pattern $F(u)$ may be thought of as a limiting form of the Chebyshev array when the number N of array elements becomes large [1074]. To see this, we consider the Chebyshev polynomial $T_N(x) = \cosh(N\xi)$, where $x = \cosh \xi$. In the limit of large N

and small ξ , with the product $N\xi$ kept constant, we can use the approximation

$$x = \cosh \xi \approx 1 + \frac{1}{2}\xi^2 \Rightarrow \xi \approx \sqrt{2x-2} \Rightarrow T_N(x) \approx \cosh(N\sqrt{2x-2}) \quad (20.12.3)$$

For an N -element array, the Chebyshev pattern is defined by Eqs. (20.9.2) and (20.9.3):

$$A(\psi) = T_{N-1}\left(x_0 \cos \frac{\psi}{2}\right), \quad R_a = T_{N-1}(x_0) \quad (20.12.4)$$

Recalling that $\psi = 2\pi u/N$, it follows that ψ will be small in the limit of large N and fixed u , thus, we may apply the approximation $\cos(\psi/2) \approx 1 - \psi^2/8$. Then, using the Chebyshev approximation (20.12.3), we have:

$$\begin{aligned} A(\psi) &\approx \cosh\left((N-1)\sqrt{2x_0 \cos \frac{\psi}{2} - 2}\right) = \cosh\left((N-1)\sqrt{2x_0\left(1 - \frac{\psi^2}{8}\right) - 2}\right) \\ &= \cosh\left(\sqrt{(N-1)^2(2x_0 - 2) - \left(\frac{(N-1)\psi}{2}\right)^2}\right) \end{aligned}$$

We also have $R_a = T_{N-1}(x_0) \approx \cosh((N-1)\sqrt{2x_0 - 2})$. Comparing with (20.12.2), we may identify $\pi A = (N-1)\sqrt{2x_0 - 2}$. We also note that in the large- N limit:

$$\frac{(N-1)\psi}{2} = \frac{(N-1)2\pi u}{2N} \approx \pi u$$

It follows that the limiting form of $A(\psi)$ is precisely the pattern (20.12.1).

Taylor introduced a modification of the ideal pattern so the first few sidelobes, say the first \bar{n} , are essentially equiripple at the given sidelobe level R , while the remaining ones follow the $\sin \pi u/\pi u$ attenuation rate of the uniform array. The method essentially preserves the mainlobe width and sidelobe level of the Chebyshev array, while allowing the far sidelobes to decay faster.

The zeros of the $\sin \pi u/\pi u$ pattern occur at the integers $u_n = \pm n$, $n = 1, 2, \dots$, whereas the zeros of the ideal pattern (20.12.1) occur at the locations:

$$\sqrt{u_n^2 - A^2} = (n - 0.5) \Rightarrow u_n = \pm \sqrt{A^2 + (n - 0.5)^2}, \quad n = 1, 2, \dots \quad (20.12.5)$$

Taylor defined a new pattern such that its zeros are:

$$u_n = \begin{cases} \pm \sqrt{A^2 + (n - 0.5)^2}, & \text{for } n = 1, 2, \dots, \bar{n} - 1 \\ \pm n, & \text{for } n \geq \bar{n} \end{cases} \quad (20.12.6)$$

The scale parameter σ is selected to allow a smooth transition between the two sets of zeros, that is, requiring the matching condition:

$$\sigma \sqrt{A^2 + (\bar{n} - 0.5)^2} = \bar{n} \Rightarrow \sigma = \frac{\bar{n}}{\sqrt{A^2 + (\bar{n} - 0.5)^2}} \quad (20.12.7)$$

The 3-dB width of the ideal pattern is obtained from the condition:

$$\cosh\left(\pi \sqrt{A^2 - u_3^2}\right) = \frac{1}{\sqrt{2}} \cosh(\pi A) \Rightarrow \Delta u = 2u_3 \quad (20.12.8)$$

The 3-dB width of the modified pattern is σ times larger:

$$\Delta u = 2u_3\sigma = 2\sigma \sqrt{A^2 - \frac{1}{\pi^2} \operatorname{acosh}^2\left(\frac{1}{\sqrt{2}} \cosh(\pi A)\right)} \quad (20.12.9)$$

As discussed by Taylor [1086], the minimum acceptable value for \bar{n} should be such that $\partial\sigma/\partial\bar{n} < 0$. This gives the constraint:

$$\bar{n} \geq 2A^2 + 1 \quad (20.12.10)$$

Eqs. (20.12.6)–(20.12.10) define completely the properties of the modified pattern. An N -element array approximating Taylor's modified pattern can be designed by the procedure outlined in Eqs. (20.6.16) and (20.6.17), that is, selecting the first $N - 1$ zeros of the continuous pattern as the zeros of the discrete pattern.

In particular, given the parameters N, R, \bar{n} , we calculate A from (20.12.2) and σ from (20.12.7), and define the $N - 1$ conjugate zeros:

$$u_n = \begin{cases} \sigma \sqrt{A^2 + (n - 0.5)^2}, & \text{for } 1 \leq n \leq \bar{n} - 1 \\ n, & \text{for } \bar{n} \leq n \leq N - \bar{n} \\ -\sigma \sqrt{A^2 + (N - n - 0.5)^2}, & \text{for } N - \bar{n} + 1 \leq n \leq N - 1 \end{cases} \quad (20.12.11)$$

Then, we define the array's zeros in ψ -space and z -domain:

$$\psi_n = \frac{2\pi u_n}{N}, \quad z_n = e^{j\psi_n}, \quad n = 1, 2, \dots, N - 1 \quad (20.12.12)$$

and convolve them to get the array pattern polynomial:

$$A(z) = \prod_{n=1}^{N-1} (z - z_n) \quad (20.12.13)$$

The first and last $\bar{n} - 1$ zeros are conjugate pairs by construction; the middle ones come in conjugate pairs because for each n in the range $\bar{n} \leq n \leq N - \bar{n}$, the integer $N - n$ is also in the same range and has a conjugated zero: $z_{N-n} = e^{j2\pi(N-n)/N} = e^{-j2\pi n/N} = z_n^*$. An exception is in the case when N is even, for which $n = N/2$ corresponds to a real zero.

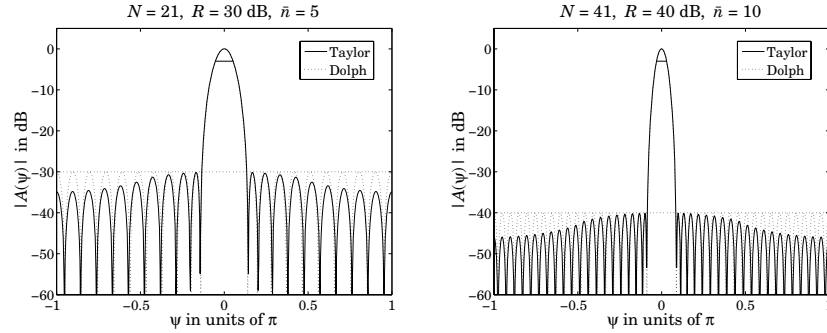
The 3-dB width of the array is calculated from $\Delta\psi = 2\pi\Delta u/N$. This design method is implemented by the MATLAB function `taylorlnb` with usage:

```
[a,dph] = taylorlnb(d,ph0,N,R,nbar); % Taylor's n-bar line source array design
```

Fig. 20.12.1 shows two design examples. The left graph has $N = 21$, $R = 30$ dB, and $\bar{n} = 5$, while the right graph has $N = 41$, $R = 40$ dB, and $\bar{n} = 10$. The required minimum values for \bar{n} , calculated from (20.12.10), were $\bar{n} = 4.48$ and $\bar{n} = 6.69$, respectively.

In order for the middle range of ns in (20.12.11) to be nontrivial, we must necessarily have $\bar{n} \leq N/2$, which combined with the restriction (20.12.10) implies a minimum value for the array length:

$$N \geq 4A^2 + 2 \quad (20.12.14)$$

Fig. 20.12.1 Taylor \bar{n} line source array design.

The following table lists some representative values of the minimum N :

R dB	\bar{n}	N	R dB	\bar{n}	N
15	2	4	50	9	18
20	2	5	55	11	22
25	3	7	60	12	25
30	4	8	65	14	29
35	5	11	70	16	33
40	6	13	75	18	37
45	7	15	80	20	41

20.13 Villeneuve Arrays

Taylor's \bar{n} pattern was based on the ideal continuous line source distribution (20.12.1), which was the limit of a Dolph-Chebyshev array. The design of an N -element array was accomplished by the usual method of mapping $N - 1$ continuous-case zeros to the $N - 1$ zeros of the array.

Villeneuve [1102] introduced an alternative design method whose starting point was a true N -element Chebyshev array, instead of the ideal limiting form. The modified array was designed by choosing its first (and last) $\bar{n} - 1$ zeros to coincide with the (stretched) zeros of the Chebyshev array, and the remaining zeros to coincide with zeros of an N -element uniform array.

We recall from Sec. 20.9 that the $N - 1$ zeros of and N -element Chebyshev array are constructed by:

$$x_n = \cos\left(\frac{(2n-1)\pi}{2(N-1)}\right), \quad \psi_n = 2 \operatorname{acos}\left(\frac{x_n}{x_0}\right), \quad z_n = e^{j\psi_n} \quad (20.13.1)$$

for $n = 1, 2, \dots, N - 1$, where x_0 is determined by $T_N(x_0) = R_a$, and R_a is the sidelobe level in absolute units. Villeneuve modified the above zeros as follows:

20.14. Multibeam Arrays

$$\psi_n = \begin{cases} +\sigma 2 \operatorname{acos}\left(\frac{x_n}{x_0}\right), & \text{for } 1 \leq n \leq \bar{n} - 1 \\ \frac{2\pi n}{N}, & \text{for } \bar{n} \leq n \leq N - \bar{n} \\ -\sigma 2 \operatorname{acos}\left(\frac{x_{N-n}}{x_0}\right), & \text{for } N - \bar{n} + 1 \leq n \leq N - 1 \end{cases} \quad (20.13.2)$$

where x_n are given as in (20.13.1). By construction, we have $\psi_{N-n} = -\psi_n$, for $1 \leq n \leq \bar{n} - 1$, which implies that the first and last \bar{n} zeros are conjugate pairs.

The scale factor σ is fixed by requiring a smooth transition between the two sets of zeros at $n = \bar{n}$, that is,

$$\sigma 2 \operatorname{acos}\left(\frac{x_{\bar{n}}}{x_0}\right) = \frac{2\pi\bar{n}}{N}, \quad \text{where } x_{\bar{n}} = \cos\left(\frac{(2\bar{n}-1)\pi}{2(N-1)}\right) \quad (20.13.3)$$

With $z_n = e^{j\psi_n}$, the array polynomial is then formed by

$$A(z) = \prod_{n=1}^{N-1} (z - z_n)$$

The 3-dB width of the new design is taken to be σ times greater than that of the Dolph-Chebyshev case. The MATLAB function `villc` implements this method:

```
[a,dph] = villc(d,ph0,N,R,nbar); % Villeneuve array design
```

The method applied to the two examples of Fig. 20.12.1 produces virtually identical graphs, and we do not repeat them here.

20.14 Multibeam Arrays

An array can form multiple narrow beams towards different directions. For example, suppose it is desired to form three beams towards the steering angles ϕ_1 , ϕ_2 , and ϕ_3 . The weights for such a multibeam array can be obtained by superimposing the weights of a single broadside array, say $w(m)$, steered towards the three angles. Defining the corresponding scanning phases $\psi_i = kd \cos \phi_i$, $i = 1, 2, 3$, we have:

$$a(m) = A_1 e^{-jm\psi_1} w(m) + A_2 e^{-jm\psi_2} w(m) + A_3 e^{-jm\psi_3} w(m)$$

where $m = 0, \pm 1, \pm 2, \dots, \pm M$ and we assumed an odd number of array elements $N = 2M + 1$. The complex amplitudes A_1, A_2, A_3 represent the relative importance of the three beams. The corresponding array factor becomes:

$$A(\psi) = A_1 W(\psi - \psi_1) + A_2 W(\psi - \psi_2) + A_3 W(\psi - \psi_3)$$

and will exhibit narrow peaks towards the three steering angles. More generally, we can form L beams towards the angles ϕ_i , $i = 1, 2, \dots, L$ by superimposing the steered beams:

$$a(m) = \sum_{i=1}^L A_i e^{-jm\psi_i} w(m), \quad m = 0, \pm 1, \pm 2, \dots, \pm M \quad (20.14.1)$$

where $\psi_i = kd \cos \phi_i$, $i = 1, 2, \dots, L$. For an even number of array elements, $N = 2M$, we replace Eq. (20.14.1) with:

$$a(\pm m) = \sum_{i=1}^L A_i e^{\mp j(m-1/2)\psi_i} w(\pm m), \quad m = 1, 2, \dots, M \quad (20.14.2)$$

For either even or odd N , the corresponding array factor will be the superposition:

$$A(\psi) = \sum_{i=1}^L A_i W(\psi - \psi_i) \quad (\text{multi-beam array factor}) \quad (20.14.3)$$

The basic broadside array weights $w(m)$ can be designed to achieve a desired sidelobe level or beam width. As the broadside beam $w(m)$ is steered away from 90° , the beamwidths will broaden. To avoid grating lobes, the element spacing d must be less than the quantity d_0 (and greater than $d_0/2$):

$$d_0 = \min_i d_i, \quad \text{where } d_i = \frac{\lambda}{1 + |\cos \phi_i|}, \quad i = 1, 2, \dots, L$$

This minimum is realized at the beam angle closest to endfire. If the steering angles are closer to each other than about one 3-dB beamwidth, the mainlobes will begin to merge with each other reducing the resolvability of the individual beams. This behavior is analogous to the problem of frequency resolution of multiple sinusoids.

The MATLAB function `multbeam.m` of Appendix I implements Eqs. (20.14.1) and (20.14.2). Its inputs are the vector of broadside array weights \mathbf{w} —which can be designed beforehand using for example `dolph2` or `taylor1p`—and the beam angles and amplitudes ϕ_i, A_i .

Example 20.14.1: Fig. 20.14.1 shows the gains of two 21-element three-beam arrays with half-wavelength spacing, and steered towards the three angles of $45^\circ, 90^\circ$, and 120° . The broadside array was designed as a Taylor-Kaiser array with sidelobe level of $R = 20$ and $R = 30$ dB.

The relative amplitudes of the three beams were equal to unity. The MATLAB code used to generate the right figure was:

```
w = taylor1p(0.5, 90, 21, 30); % unsteered weights
a = multbeam(0.5, w, [1,1,1], [45, 90, 120]); % equal-amplitude beams
[g, ph] = array(d, a, 400); % compute gain
dbz(ph, g); % plot gain in dB
addray(45); addray(-45); % add ± 45° grid rays
```

We note the broadening of the beam widths of the larger beam angles. The left array has narrower mainlobes than the right one because its sidelobe attenuation is less. But, it also exhibits more constructive interference between mainlobes causing somewhat smaller sidelobe attenuations than the desired one of 20 dB. □

Equations (20.14.1) and (20.14.2) generalize the Woodward-Lawson frequency sampling design equations (20.5.6) and (20.5.7) in the sense that the steering phases ψ_i can be arbitrary and do not have to be the DFT frequencies.

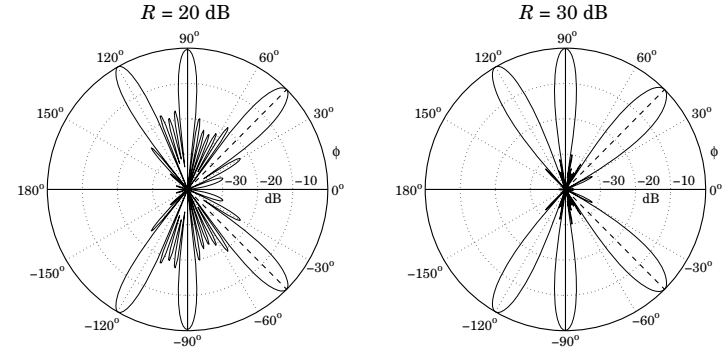


Fig. 20.14.1 Multi-beam arrays with $R = 20$ and $R = 30$ dB sidelobes.

However, if the ψ_i are chosen to be the DFT frequencies given by Eq. (20.5.1) or (20.5.5), and the broadside array is chosen to be a length- N uniform array, $w(m) = 1$, then the inverse DFT expressions (20.5.6) and (20.5.7) can be thought of as defining N beams—called the Woodward-Lawson-Butler beams—steered towards the DFT angles $\phi_i = \text{acos}(\psi_i/kd)$, that is, towards

$$\phi_i = \text{acos}\left(\frac{\psi_i}{kd}\right) = \text{acos}\left(\frac{2\pi i}{Nkd}\right) = \text{acos}\left(\frac{\lambda i}{Nd}\right), \quad i = 0, 1, \dots, N-1 \quad (20.14.4)$$

The array weights will be given then by the inverse DFT:

$$a(m) = \frac{1}{N} \sum_{i=0}^{N-1} A(\psi_i) e^{-jm\psi_i} \quad (20.14.5)$$

and the corresponding array factor by:

$$A(\psi) = \frac{1}{N} \sum_{i=0}^{N-1} A(\psi_i) W(\psi - \psi_i) \quad (20.14.6)$$

where $W(\psi) = \sin(N\psi/2) / \sin(\psi/2)$ is the array factor of the uniform window. The DFT values are identified as the relative beam weights $A_i = A(\psi_i)/N$.

A single Butler beam, say the j th beam, can be turned on by choosing $A_i = \delta_{ij}$. By successively turning on the Butler beams one by one, the array will act as a scanning array. Fig. 20.14.2 depicts such a multi-beam array structure. The inverse DFT box implements Eq. (20.14.5). The inputs are the “beams” A_i and the outputs are the weights $a(m)$.

Somewhat before the advent of the FFT algorithm, Butler proposed a hardware realization of the inverse DFT network, which was quickly recognized to be equivalent to the FFT algorithm [1103,1104,1106–1109]. The DFT matrix realization of this network is called the Blass matrix in the antenna array context [18,10].

Example 20.14.2: Fig. 20.14.3 shows the individual Butler beams turned on successively for an eight-element array. Both the standard and alternative DFT frequency sets are shown.

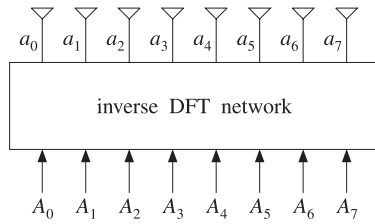


Fig. 20.14.2 Woodward-Lawson-Butler beam matrix network for $N = 8$.

There are eight beams in each graph. For the standard DFT set, the two endfire beams count as one, that is, the $i = 0$ beam.

The sidelobes are at the 13-dB level because these are scanned versions of the uniform array. The mainlobes intersect exactly half-way between the DFT frequencies ψ_i , that is, the i th beam intersects the neighboring ones at $\psi = \psi_i + \pi/N = 2\pi(i + 0.5)/N$. These intersection points are approximately 4 dB down (3.92 dB to be exact) from the main peaks. The 4-dB gain circle intersects the gain curves at these points. \square

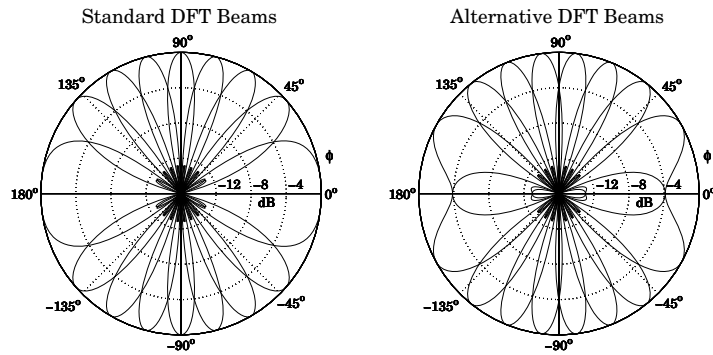


Fig. 20.14.3 Woodward-Lawson-Butler beams for $N = 8$.

20.15 Problems

20.1 *Computer Experiment—Taylor's one-parameter/ \bar{n} array design.* Taylor's \bar{n} distribution of Sec. 20.12 can also be applied to Taylor's one-parameter continuous distribution of Sec. 20.10. First, show that the zeros of Eq. (20.10.2) occur at

$$u_n = \sqrt{B^2 + n^2}, \quad n = 1, 2, \dots$$

20.15. Problems

Then, choose \bar{n} such that $\bar{n} \leq N/2$ and define an N -element array by its $N - 1$ zeros:

$$u_n = \begin{cases} \sigma\sqrt{B^2 + n^2}, & \text{for } 1 \leq n \leq \bar{n} - 1 \\ n, & \text{for } \bar{n} \leq n \leq N - \bar{n} \\ -\sigma\sqrt{B^2 + (N - n)^2}, & \text{for } N - \bar{n} + 1 \leq n \leq N - 1 \end{cases} \quad (20.15.1)$$

Fix the parameter σ such that $\sigma\sqrt{B^2 + \bar{n}^2} = \bar{n}$. Then, define the array polynomial:

$$A(z) = \prod_{n=1}^{N-1} (z - z_n), \quad \psi_n = \frac{2\pi u_n}{N}, \quad z_n = e^{j\psi_n}, \quad n = 1, 2, \dots, N - 1 \quad (20.15.2)$$

Write a MATLAB function that implements this procedure, and takes as input the parameters N, R, \bar{n} and outputs the array weights and 3-dB width.

Apply your function to the following example $N = 21, R = 30$ dB, $\bar{n} = 5$, with half-wavelength spacing $d = \lambda/2$. You will notice that, like the prolate array, the mainlobe is slightly narrower and the sidelobe level slightly better matched than the Taylor-Kaiser array. On the same graph, plot the array patterns $|A(\psi)|$ in dB for the present design, the Taylor-Kaiser and the prolate arrays designed with the same specifications. Vary \bar{n} to understand its effect on the design.

20.2 *Computer Experiment—Villeneuve array design.* Redesign the examples shown in Fig. 20.12.1 using Villeneuve's array design method and plot the array responses together with those of that figure. Vary the parameters N, R, \bar{n} and compare the range of similarity of the Villeneuve versus the Taylor \bar{n} method.