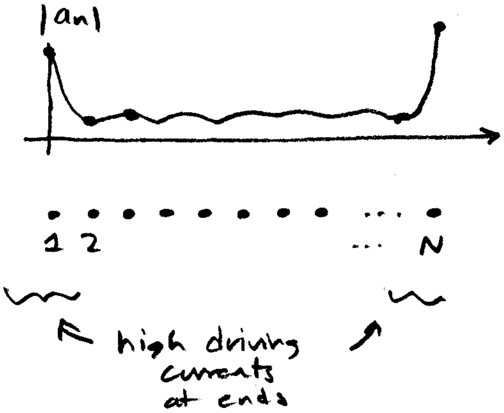
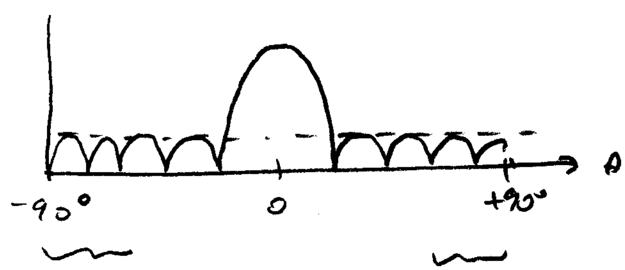
PENCIL BEAM PATTERN

The Chebychev array is one approach to synthesizing a particular type of radiation pattern, that of a narrow or pencil beam, or a highly directive pattern with low side lobes.

The Chebychev or Dolph-Chebychev array has one important limitation: as the number of elements increase, the amplitudes across the array tend to peak at the ends of the array:

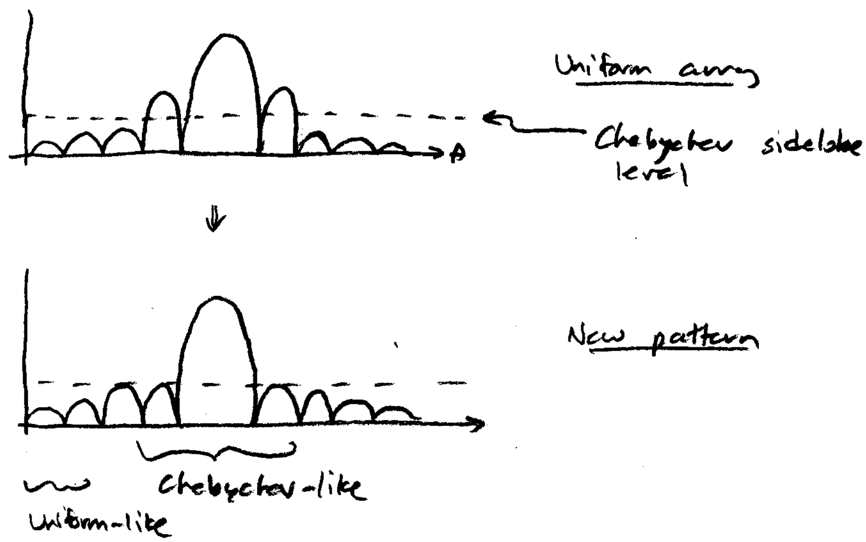


The reason for this is that the Chebychev array by design must maintain equal amplitude sidelobes, even for the "deep" sidelobes far from the main lobe:



The peaks at the array ends are like delta functions which lead to nearly constant radiation patterns in angle, keeping the "deep" sidelobes high.

An improved approach is to design a pattern that is similar to the Chebychev pattern near the main lobe, but with deep sidelobes similar to that of the uniform array.



Procedures for determining the array weights for this type of pattern can be found in the literature [Hansen, Proc. IEEE, 80, Jan. 1992, p. 141].

ARBITRARY PATTERNS

Another type of pattern synthesis is to design weights to obtain an arbitrary specified pattern. This could be a pencil beam, or something very different, as determined by the application.

There are many different ways to do this:

1. Put pattern in polynomial form:

$$F(u) = \sum_{n=0}^{N-1} I_n e^{-inkd\cos\theta}$$

$$= \sum_{n=0}^{N-1} \underbrace{I_n (w)^n}_{\text{polynomial}} , \quad w = e^{-ikd\cos\theta}$$

$$= \prod_{n=0}^{N-1} (w - w_n)$$

pattern is determined by zeros of the polynomial (null locations)

Once the zeros w_n are designed, the weights I_n can be found by multiplying out the polynomial and identifying the weights I_n.

One important way to set the zeros is to compute them iteratively, by taking the derivative of some starting pattern with respect to the zeros, and then solving a linear system for changes in the zeros such that the new pattern approaches the desired pattern.

Other methods: Lagrange interpolation of desired pattern
Fix nulls directly

2. $kd = \pi \Rightarrow -\pi \le u \le u \Rightarrow$ array factor is a Fourier series

→ expand desired pattern as a Fourier series and truncate it.

3. Trigonometric interpolation
4. other types of interpolation
5. Numerical optimization
6. Adaptive arrays (minimize noise ...)

Some of these methods can be generalized to 2D arrays.

Fourier series: If $kd = \pi$ ($d = \lambda/2$), then

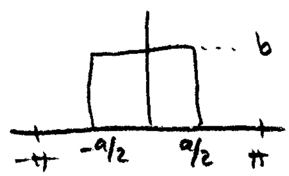
$$F(u) = \sum_{0}^{N-1} a_n e^{inu}$$

is a Fourier series. If we can expand our desired pattern $G(u)$ as a Fourier series,

$$G(u) = \sum_{0}^{\infty} b_n e^{inu}$$

and truncate at N terms, then $b_n = a_n$, $n = 0, 1, \ldots, N-1$.
(Why doesn't this work for $kd \neq \pi$?)

Example: Desired pattern is a rect:



The Fourier coefficients are

$$b_n = \frac{1}{2\pi}\int_{-a/2}^{a/2} e^{-inu}\, b\, du = \frac{1}{2\pi}\, \frac{e^{-ina/2} - e^{ina/2}}{-in}$$

$$= \frac{b}{2\pi}\, \frac{\sin(na/2)}{n/2}$$

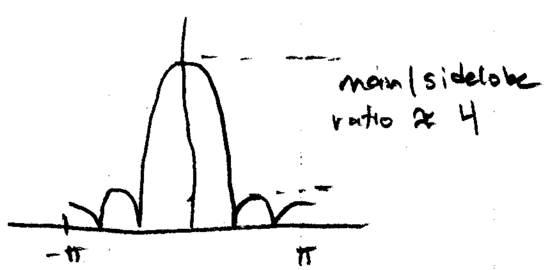
Let's shift the phase reference so that

$$F(u) = \sum_{-(N-1)/2}^{(N-1)/2} a_n e^{inu} \qquad (N \text{ odd})$$

Let's truncate at $N = 5$. Then

$$F(u) = \sum_{2}^{-2} \frac{b}{2\pi}\left(\frac{\sin na/2}{n/2}\right) e^{inu}$$

which is



Not quite a rect, but headed in that direction...
Problems: Gibb's phenomenon for piecewise continuous patterns.