E9 231: Digital Array Signal Processing

Scribe: Nishkam Jain
Dept. of EE and ECE
Indian Institute of Science
Bangalore 560 012, India
nishkamjain@ee.iisc.ernet.in

Class held on: 04 Sept 2008

1 Topics

- Rectangular Array design.
- Null Constraint based design.
- Least Squares design.
- Seperable design.
- Circular Symmetric design.

2 Class Notes

2.1 Designing Rectangular Arrays

$$w_{nm}^* = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{2\pi} B(\psi_x,\psi_y)\ e^{j(\frac{N-1}{2}\psi_x+\frac{M-1}{2}\psi_y)}\ e^{-j(n\psi_x+m\psi_y)} d\psi_x\psi_y$$

w_{nm} will need to use windowing or truncation. Here we borrow results from window based - 2D FIR filter design techniques. A good reference for that is **"*Multi-dimentional Signal Processing*"** by **Dudgeon and Meneseran**.

2.2 Another representation of $B(\psi_x, \psi_y)$.

$$B(\psi_x,\psi_y) = \ e^{-jphase}\sum_{m=0}^{M-1} w_m^* V_m(\psi)$$

$$phase = -\left(\left(\frac{N-1}{2}\right)\psi_x + \left(\frac{M-1}{2}\right)\psi_y\right)$$

where

$$\psi = (\psi_x, \psi_y)$$

$$V_m(\psi) = \begin{bmatrix} e^{jm\psi_y} \\ e^{j(\psi_x+m\psi_y)} \\ e^{j(\overline{N-1}\psi_x+m\psi_y)} \end{bmatrix}_{N\times 1}$$

$$w_m = \begin{bmatrix} w_{0,m} \\ \vdots \\ w_{N-1,m} \end{bmatrix}_{N\times 1}$$

$$W = \begin{bmatrix} w_0, w_1 \dots w_{M-1} \end{bmatrix}_{N\times 1}$$

$$V(\psi) = \begin{bmatrix} v_o(\psi), \dots v_{M-1}(\psi) \end{bmatrix}_{N\times M}$$

$$vecW = \begin{bmatrix} w_0 \\ \vdots \\ w_{M-1} \end{bmatrix}_{MN\times 1}$$

$$vecV(\psi) = \begin{bmatrix} v_0(\psi) \\ \vdots \\ v_{M-1}(\psi) \end{bmatrix}_{MN\times 1}$$

$$B(\psi) = e^{jphase}.vecW.vecV(\psi) \tag{1}$$

2.3 Null Constraint based design

Assume that there exists a desired beam pattern $B_d(\psi)$ which can be realized using weights $W_{d(N\times M)}$. We would like to minimise the mean square error between $B(\psi)$ and $B_d(\psi)$ while satisfying some null constraints. In (1) form, the null constraints leads to linear constraints. This leads us to an optimization problem of the form

$$min\, \|vec(\mathbf{W}) - vec(\mathbf{W}_d)\|^2$$

subject to $vec(\mathbf{W})^H C = \underline{0}$
like in one dimentional case we have the following solution:

$$vec(\mathbf{W})_{opt} = P_c^{\perp} vec(\mathbf{W}_d)$$

$$P_c^{\perp} = \left(I - C\left(C^H C\right)^T C^H\right)$$

this is similar to imposing null constraints in one dimentional case.

2.4 Least Squares Design

Given a $B_d(\theta,\phi)$ and we want to try to get as close to it as possible, hence we want to minimise

$$\frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi} \mid B_d(\theta,\phi) - B(\theta,\phi) \mid^2 \sin\theta d\theta d\phi$$

here we include e^{jphase} in $B_d(\theta,\phi)$

$$\frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi} \mid B_d(\theta,\phi) - vec(W)^H vecV(\theta,\phi)|^2 \sin\theta d\theta d\phi$$

if we solve the integrand we get

$$(B_d(\theta,\phi) - vec(W)^H vecV(\theta,\phi)).(B_d^*(\theta,\phi) - vecV(\theta,\phi)^H vec(W))$$

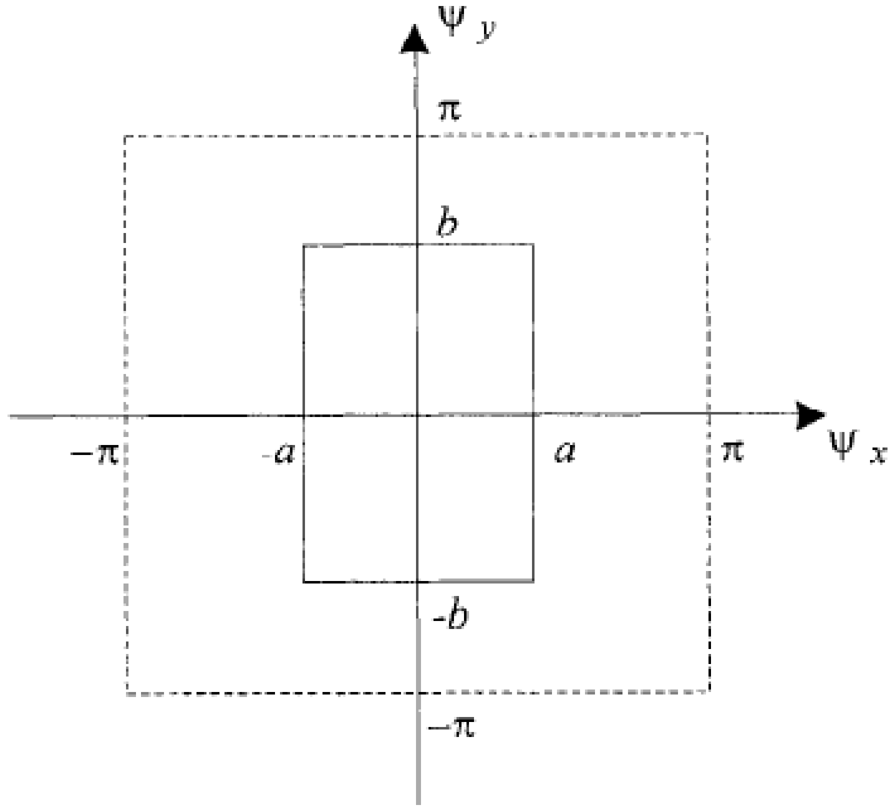

Figure 1: Ex1. SURA

$$= constant - vec(W)^H vecV(\theta,\phi)B_d^*(\theta,\phi) - B_d(\theta,\phi)vecV(\theta,\phi)^H vec(W) + vec(W)^H vecV(\theta,\phi)vecV(\theta,\phi)^H vec(W)$$

now if we differentiate wrt $vec(W)^H$ and equate it to zero and solve it (Note: look at appendix for notes on complex derivatives), we get the objective function as

$$= constant - vec(W)^H \underline{p} - \underline{p}^H vecW + vecW^H B_1 vecW$$

where

$$\underline{p} := \frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi} vecV(\theta,\phi)B_d^*(\theta,\phi)\sin\theta d\theta d\phi$$

$$B_1 := \frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi} vecV(\theta,\phi)vecV(\theta,\phi)^H \sin\theta d\theta d\phi$$

after solving we get

$$vecW_{opt} = B_1^{-1}\underline{p} \tag{2}$$

Ex1. I want to design a standard uniform rectangular array (SURA)(Fig:1)

$$B_d(\psi_x,\psi_y) = \begin{cases} 1 & -\psi_a \le \psi_x \le \psi_a \ and \ -\psi_b \le \psi_y \le \psi_b \\ 0 & otherwise \end{cases},$$

Remember: Uniform in $(\psi_x,\psi_y) \not\Rightarrow$ a nice region in the (θ,ϕ) space

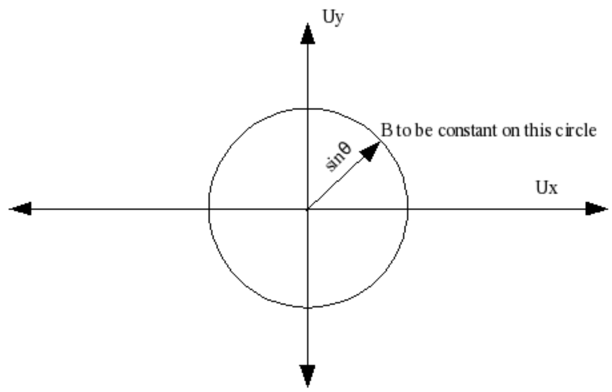

Figure 2: Response independent of ϕ

now taking the inv FT we get:

$$w^*_{nm} = \frac{1}{4\pi^2}\int_{-\psi_a}^{\psi_a}\int_{-\psi_b}^{\psi_b} 1.e^{-jphase}e^{-j(n\psi_x+m\psi_y)}d\psi_y d\psi_x$$

$$w^*_{nm} = \frac{1}{4\pi^2}\int_{-\psi_a}^{\psi_a}\int_{-\psi_b}^{\psi_b} 1.e^{-j\left[(n-\frac{N-1}{2})\psi_x+(m-\frac{M-1}{2})\psi_y\right]}d\psi_y d\psi_x$$

$$w^*_{nm} = \frac{\sin((n-\frac{N-1}{2})\psi_a)}{(n-\frac{N-1}{2})\pi}.\frac{\sin((m-\frac{M-1}{2})\psi_b)}{(m-\frac{M-1}{2})\pi} \tag{3}$$

Note here that the weights are separable and real.
Similar to the 1D case, for finite arrays, there is an overshoot in the beam pattern due to Gibbs phenomenon
To alleviate the overshoot apply a window (A filter design problem)

$$\text{Design weights} = w^*_{nm}\times \text{ Window}$$
$$= \frac{\sin((n-\frac{N-1}{2})\psi_a)}{(n-\frac{N-1}{2})\pi}\times 1D\ window\ of\ length\ N$$
$$= \frac{\sin((m-\frac{M-1}{2})\psi_b)}{(m-\frac{M-1}{2})\pi}\times 1D\ window\ of\ length\ M$$

Kaiser window is a good choice for window

2.5 Circularly Symmetric Design

We want the beam pattern to be uniform in ϕ. that is for a given θ the response is independent of ϕ.
Recall: for SURA

$$\psi_x = \pi\sin\theta\cos\phi$$
$$\psi_y = \pi\sin\theta\sin\phi$$

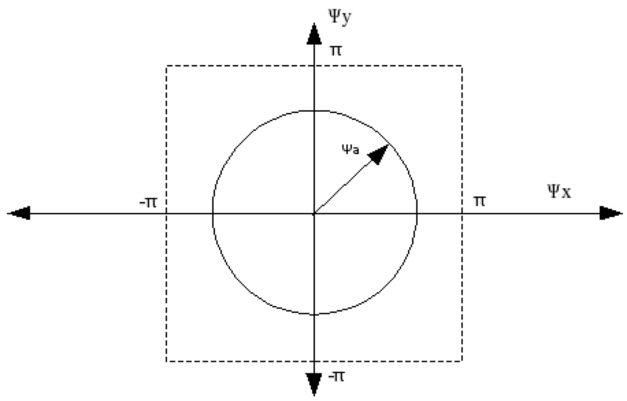

Figure 3: Ex:2

if the beam pattern is to be independent of ϕ we must have

$$B(\psi_x, \psi_y) = B_{\psi_R}(\sqrt{\psi_x^2 + \psi_y^2}) \tag{4}$$

$\psi_R = \sqrt{\psi_x^2 + \psi_y^2} :=$ radial wave number
Observe: side lobe structure will be rings (see textbook)
Beam pattern satisfying (4) have a circular sidelobe structure and (in general) nm seperable weighting functions.

$$B_\psi(\psi_x, \psi_y) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} a_{nm}^* e^{j(n\psi_x + m\psi_y)}$$

note here that this change of indexing removes the e^{jphase} term
Claim: for circularly symmetric $B_\psi(\psi_x, \psi_y)$ the weights a_{nm} will also have a circular symmetry.

$$a_{nm} = a(\sqrt{n^2 + m^2}) = a_{\sqrt{n^2+m^2},0}$$

we can design $a_{n,o}$ and simply substitute $n = \sqrt{n^2 + m^2}$
Example 2: Let us have the following desired beam:(Fig. 3)

$$B_d(\psi_x, \psi_y) = \begin{cases} 1 & \sqrt{\psi_x^2 + \psi_y^2} \le \psi_R \\ 0 & otherwise \end{cases},$$

take Inv FT

$$a_{n0}^* = \frac{1}{4\pi^2} \int_{-\psi_R}^{\psi_R} \int_{-\sqrt{\psi_R^2 - \psi_x^2}}^{\sqrt{\psi_R^2 - \psi_x^2}} 1.e^{-j(n\psi_x + m\psi_y)} d\psi_y d\psi_x$$

now put $m = 0$

$$a_{n0}^* = \frac{1}{4\pi^2} \int_{-\psi_R}^{\psi_R} \int_{-\sqrt{\psi_R^2 - \psi_x^2}}^{\sqrt{\psi_R^2 - \psi_x^2}} e^{-j(n\psi_x)} d\psi_y d\psi_x$$

$$a_{n0}^* = \frac{1}{4\pi^2}\int_{-\psi_R}^{\psi_R}(2\sqrt{\psi_R^2 - \psi_x^2})e^{-j(n\psi_x)}d\psi_x$$

substituting $\psi_x = \psi_R \sin\phi$ above we get,

$$a_{n0}^* = \frac{1}{4\pi^2}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}(2\psi_R^2\cos^2\phi)e^{-j(n\psi_R\sin\phi)}d\phi$$

$$a_{n0}^* = \frac{\psi_R}{2\pi n}J_1(\psi_R, n)$$

where $J_1(\psi_R, n)$ is the first order modified bessel function of first kind.

$$J_m(x) := x^m \sum_{l=0}^{\infty}\frac{(-1)^l x^{2l}}{2^{(2l+m)}l!(l+m)!}$$

if x is small $2^{(2l+m)}l!(l+m)!$ terms decay rapidly, thus we get real weights,

$$a_{nm} = \frac{\psi_R J_1(\psi_R\sqrt{n^2+m^2})}{2\pi\sqrt{n^2+m^2}} \tag{5}$$

Thus we get the circularly symmetric beam pattern, but it needs to be truncated for finite arrays

2.6 Several choices of truncation

Let us consider an example with N=M=21

1. Trivial truncation

$$a_{nm} = \begin{cases} \frac{\psi_R J_1(\psi_R\sqrt{n^2+m^2})}{2\pi\sqrt{n^2+m^2}} & -10 \le n \le 10 \ and \ -10 \le m \le 10 \\ 0 & otherwise \end{cases},$$

   it is essentially a rectangular window which implies that we end up with large sidelobes. may be some loss of circular symmetry

2. Also fairly trivial

$$a_{nm} = \begin{cases} \frac{\psi_R J_1(\psi_R\sqrt{n^2+m^2})}{2\pi\sqrt{n^2+m^2}} & -10 \le \sqrt{n^2+m^2} \le 10 \\ 0 & otherwise \end{cases},$$

   will use ~ 314 elements instead of 441.

3. Apply a window (Kaiser)

$$a_{nm} = \frac{\psi_R J_1(\psi_R\sqrt{n^2+m^2})w(\sqrt{n^2+m^2})}{2\pi\sqrt{n^2+m^2}}$$

   here $w(\sqrt{n^2+m^2})$ is a 1D window. we see that we can design 2D windows using transformations of 1D window.