Solution to Mini MATLAB Project: Beampattern and Beamwidth Analysis of Two Uniform Linear Arrays

Beamwidth Calculation

For a uniform planar array with $M = N^2$ antennas, the array gain is given by:

$$\text{Array Gain} = \frac{1}{M} \left| \sum_{n=1}^{N} \sum_{m=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{x_n^2}{2z}+\frac{y_m^2}{2z}+z\right)} e^{j\Phi_{n,m}} \right|^2 \tag{1}$$

For a uniform linear array (ULA) with N antennas, the normalized array gain (NAG) simplifies to:

$$\text{NAG} = \frac{1}{(N)^2} \left| \sum_{n=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{x_n^2}{2z}+z\right)} e^{j\Phi_n} \right|^2 \tag{2}$$

To analyze the beamwidth, we consider the user's location shift along the x-axis by x_t, with the focal point at $(0,0,F)$ and $z = F$. The normalized array gain is then given by:

$$\text{NAG} = \frac{1}{(2N)^2} \left| \sum_{n=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{(x_n-x_t)^2}{2F}+F\right)} e^{j\frac{2\pi}{\lambda}(\frac{x_n^2}{2F}+F)} \right|^2 \tag{3}$$

where $x_n = (n - \frac{N+1}{2})\Delta$ ensures that the ULA is centered at zero.

For two ULAs separated by a distance B, each array is considered separately by shifting their centers. Defining the shift as $-\bar{B}\Delta$ and $+\bar{B}\Delta$, where $\bar{B} = \frac{(N-1)}{2} + \frac{B}{2}$, the normalized array gain is expressed as:

$$\text{NAG} = \frac{1}{(2N)^2} \left| \sum_{n=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{(x_n-\bar{B}\Delta-x_t)^2}{2F}+F\right)} e^{j\frac{2\pi}{\lambda}\left(\frac{(x_n-\bar{B}\Delta)^2}{2F}+F\right)} \right. \tag{4}$$

$$\left. + \sum_{m=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{(x_m+\bar{B}\Delta-x_t)^2}{2F}+F\right)} e^{j\frac{2\pi}{\lambda}\left(\frac{(x_m+\bar{B}\Delta)^2}{2F}+F\right)} \right|^2 \tag{5}$$

Next, by expanding the exponents, we have:

$$\text{NAG} = \frac{1}{(2N)^2}\left|\sum_{n=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{(x_n-\bar{B}\Delta)^2+x_t^2-2(x_n-\bar{B}\Delta)x_t}{2F}-\frac{(x_n-\bar{B}\Delta)^2}{2F}\right)}\right. \tag{6}$$

$$\left.+\sum_{m=1}^{N} e^{-j\frac{2\pi}{\lambda}\left(\frac{(x_m+\bar{B}\Delta)^2+x_t^2-2(x_m+\bar{B}\Delta)x_t}{2F}-\frac{(x_m+\bar{B}\Delta)^2}{2F}\right)}\right|^2 \tag{7}$$

Then, we can simplify it to:

$$\text{NAG} = \frac{1}{(2N)^2}\left|e^{-j\frac{2\pi}{\lambda}\frac{x_t^2}{2F}}\left(\sum_{n=1}^{N} e^{j\frac{2\pi}{\lambda}\frac{(x_n-\bar{B}\Delta)x_t}{F}}+\sum_{m=1}^{N} e^{j\frac{2\pi}{\lambda}\frac{(x_m+\bar{B}\Delta)x_t}{F}}\right)\right|^2 \tag{8}$$

Since the phase shift $e^{-j\frac{2\pi}{\lambda}\frac{x_t^2}{2F}}$ does not affect the magnitude, we can remove it and write NGA as follows:

$$\text{NAG} = \frac{1}{(2N)^2}\left|e^{-j\frac{2\pi}{\lambda}\frac{\bar{B}\Delta x_t}{F}}\sum_{n} e^{j\frac{2\pi}{\lambda}\frac{x_n x_t}{F}}+e^{j\frac{2\pi}{\lambda}\frac{\bar{B}\Delta x_t}{F}}\sum_{m} e^{j\frac{2\pi}{\lambda}\frac{x_m x_t}{F}}\right|^2 \tag{9}$$

Then:

$$\text{NAG} = \frac{1}{(2N)^2}\left|\left(e^{-j\frac{2\pi}{\lambda}\frac{\bar{B}\Delta x_t}{F}}+e^{j\frac{2\pi}{\lambda}\frac{\bar{B}\Delta x_t}{F}}\right)\sum_{n=1}^{N} e^{j\frac{2\pi}{\lambda}\frac{x_n x_t}{F}}\right|^2 \tag{10}$$

Since $e^{-j\theta}+e^{j\theta}=2\cos(\theta)$:

$$\text{NAG} = \frac{1}{(2N)^2}\left|2\cos\left(\frac{2\pi\bar{B}\Delta x_t}{\lambda F}\right)\sum_{n=1}^{N} e^{j\frac{2\pi}{\lambda}\frac{x_n x_t}{F}}\right|^2 \tag{11}$$

We approximate the summation with an integral by defining $x_n = n\Delta$, where $n \in \left(-\frac{N}{2}, \frac{N}{2}\right)$, as follows[**?**]:

$$\text{NAG} = \frac{1}{(2N)^2}\left|2\cos\left(\frac{2\pi\bar{B}\Delta x_t}{\lambda F}\right)\int_{-N/2}^{N/2} e^{j\frac{2\pi}{\lambda}\frac{n\Delta x_t}{F}}\,dn\right|^2 \tag{12}$$

Solving the integral expression:

$$\text{NAG} = \frac{1}{(2N)^2}\left|2\cos\left(\frac{2\pi\bar{B}\Delta x_t}{\lambda F}\right)\frac{1}{j\frac{2\pi}{\lambda}\frac{\Delta x_t}{F}}\left(e^{j\frac{2\pi}{\lambda}\frac{N\Delta x_t}{2F}}-e^{-j\frac{2\pi}{\lambda}\frac{N\Delta x_t}{2F}}\right)\right|^2 \tag{13}$$

Using the sinc function:

$$\text{NAG} = \frac{1}{(2N)^2}\left|2\cos\left(\frac{2\pi\bar{B}\Delta x_t}{\lambda F}\right)N\text{sinc}\left(\frac{N\Delta x_t}{F\lambda}\right)\right|^2 \tag{14}$$

Thus, we arrive at the final expression for the normalized array gain.

$$\text{NAG} = \left| \cos\left(\frac{2\pi \bar{B} \Delta x_t}{\lambda F}\right) \text{sinc}\left(\frac{N \Delta x_t}{F \lambda}\right) \right|^2 \tag{15}$$

MATLAB Simulation

The figures below illustrate the beampattern analysis of two Uniform Linear Arrays (ULAs) separated by different distances B. The first column of plots represents the Normalized Array Gain (NAG) as a function of the spatial coordinates (x, z), while the second column displays the Beampattern along the x-axis, including the upper bound, the computed beampattern, and the corresponding -3 dB beamwidth.

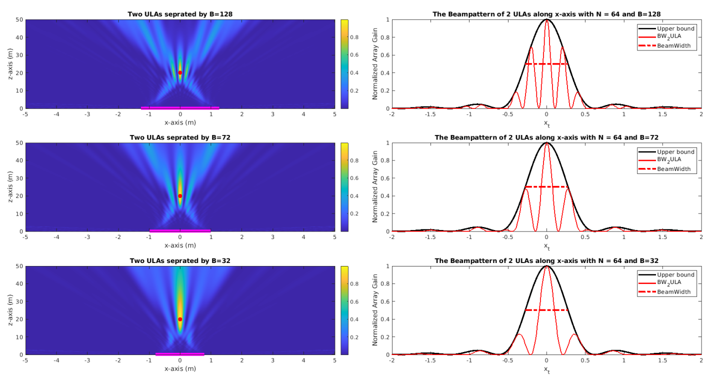

Figure 1: Beam pattern and beamwidth analysis results from MATLAB simulation, illustrating the effects of antenna configuration on the normalized array gain.

MATLAB Code

MATLAB script for computing and visualizing the beampattern of two ULAs.

```
% ######################### Feb 23, 2024
% ######################### Mini MATLAB Project:
   Beampattern and Beamwidth Analysis of Two Uniform
   Linear Arrays
% ######################### Written by Shima Mashhadi,
    PhD student at Rochester Institute of Technology,
   supervised by Professor Alireza Vahid.
% ######################### EEEE-789 Spectrum Sharing
% ######################### Beam Focusing of Two ULAs

clear; clc;
```

```
% Constants
N = 64;
F = 20; % Focal Point
c = 3 * 10^8; % Speed of light
f_c = 15 * 10^9; % Carrier frequency
wave_length = c / f_c;
space_Ant_elements = wave_length * 0.5; % Spacing between two adjacent elements
B = [128, 72, 32]; % Distance between two ULAs in multiples of space_Ant_elements
B_bar = ((B + (N - 1)) / 2) * space_Ant_elements; % The center shift of the ULAs from 0 to -B_bar and +B_bar

%% Fraunhofer distance of each ULA
dF = 2 * (space_Ant_elements * N).^2 / wave_length;

%% Antenna axis
xn = ((1:N) - ((N + 1) / 2)) * space_Ant_elements;

%% Define spatial coordinates
x = linspace(-5, 5, 1000);
z = linspace(0, 50, 1000);

%% Initialize Normalized Array Gain (NAG) matrix
NAG = zeros(length(z), length(x));

% Compute the beampattern
for k = 1:length(B_bar)
    for i = 1:length(x)
        for j = 1:length(z)
            ULA1 = exp(-1i * (2 * pi / wave_length) * ((xn - B_bar(k) - x(i)).^2 / (2 * z(j)))) .* ...
                exp(1i * (2 * pi / wave_length) * ((xn - B_bar(k)).^2 / (2 * F)));
            ULA2 = exp(-1i * (2 * pi / wave_length) * ((xn + B_bar(k) - x(i)).^2 / (2 * z(j)))) .* ...
                exp(1i * (2 * pi / wave_length) * ((xn + B_bar(k)).^2 / (2 * F)));
            NAG(j, i, k) = (1 / ((2 * N)^2)) * (abs(sum(ULA1) + sum(ULA2)).^2);
        end
    end
```

```
end

%% Compute Beamwidth with calculated equation
BW_NF = zeros(length(x), length(B_bar));
for i = 1:length(B_bar)
    BW_NF(:, i) = abs(sinc(N * x / (2 * F)) .* (cos(2
        * pi * B_bar(i) * x / (wave_length * F)))).^2;
end
BW_UB = (sinc(N * x / (2 * F))).^2;
BW_3dB = 1.77 * F / N;

%% Plot the results
figure;
tiledlayout(3, 2)

% Plot for B = 128
nexttile
imagesc(x, z, NAG(:, :, 1));
set(gca, 'YDir', 'normal');
colorbar;
xlabel('x-axis (m)');
ylabel('z-axis (m)');
title(['Two ULAs separated by B = ', num2str(B(1))]);
hold on;
scatter(0, F, 50, 'r', 'filled');
plot([xn(1) - B_bar(1), xn(end) + B_bar(1)], [0, 0], '
    -m', 'LineWidth', 5);

nexttile
hold on
plot(x, BW_UB, '-k', 'LineWidth', 2.5, 'DisplayName',
    'Upper Bound');
plot(x, BW_NF(:, 1), '-r', 'LineWidth', 2, '
    DisplayName', 'BW_2ULA');
plot([-BW_3dB / 2, BW_3dB / 2], [0.5, 0.5], '-.r', '
    LineWidth', 3, 'DisplayName', 'Beamwidth');
title('Beampattern along x-axis with N = 64 and B =
    128');
xlim([-2, 2]);
xlabel('x_t');
ylabel('Normalized Array Gain');
box on;
legend('show', 'FontSize', 10, 'Location', 'northeast'
    );

% Plot for B = 72
```

```
nexttile
imagesc(x, z, NAG(:, :, 2));
set(gca, 'YDir', 'normal');
colorbar;
xlabel('x-axis (m)');
ylabel('z-axis (m)');
title(['Two ULAs separated by B = ', num2str(B(2))]);
hold on;
scatter(0, F, 50, 'r', 'filled');
plot([xn(1) - B_bar(2), xn(end) + B_bar(2)], [0, 0], '
    -m', 'LineWidth', 5);

nexttile
hold on
plot(x, BW_UB, '-k', 'LineWidth', 2.5, 'DisplayName',
    'Upper Bound');
plot(x, BW_NF(:, 2), '-r', 'LineWidth', 2, '
    DisplayName', 'BW_2ULA');
plot([-BW_3dB / 2, BW_3dB / 2], [0.5, 0.5], '-.r', '
    LineWidth', 3, 'DisplayName', 'Beamwidth');
title('Beampattern along x-axis with N = 64 and B = 72
    ');
xlim([-2, 2]);
xlabel('x_t');
ylabel('Normalized Array Gain');
box on;
legend('show', 'FontSize', 10, 'Location', 'northeast'
    );

% Plot for B = 32
nexttile
imagesc(x, z, NAG(:, :, 3));
set(gca, 'YDir', 'normal');
colorbar;
xlabel('x-axis (m)');
ylabel('z-axis (m)');
title(['Two ULAs separated by B = ', num2str(B(3))]);
hold on;
scatter(0, F, 50, 'r', 'filled');
plot([xn(1) - B_bar(3), xn(end) + B_bar(3)], [0, 0], '
    -m', 'LineWidth', 5);

nexttile
hold on
plot(x, BW_UB, '-k', 'LineWidth', 2.5, 'DisplayName',
    'Upper Bound');
```

```
plot(x, BW_NF(:, 3), '-r', 'LineWidth', 2, '
    DisplayName', 'BW_2ULA');
plot([-BW_3dB / 2, BW_3dB / 2], [0.5, 0.5], '-.r', '
    LineWidth', 3, 'DisplayName', 'Beamwidth');
title('Beampattern along x-axis with N = 64 and B = 32
    ');
xlim([-2, 2]);
xlabel('x_t');
ylabel('Normalized Array Gain');
box on;
legend('show', 'FontSize', 10, 'Location', 'northeast'
    );
```

References