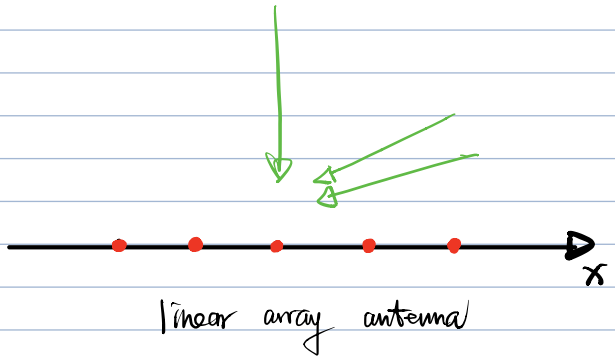
Experiment AM

Main MATLAB codes are at the end of this logbook

Array Communications & Processing

5 sensors along x-axis

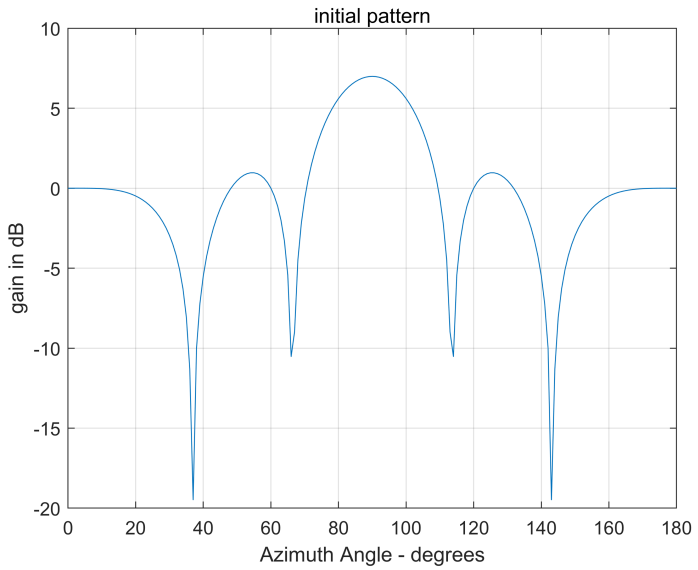
$$\text{array} = \begin{bmatrix} -2 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$



direction of sources

$$\text{direction} = \begin{bmatrix} 30 & 0 \\ 31 & 0 \\ 90 & 0 \end{bmatrix}$$

1. Pattern of the array



If there is no beamforming, the pattern of the array is shown on the left, where the maximum gain is obtained at the degree of 90°

2. Theoretical Covariance Matrix

Array manifold vector

$$\underline{s}(\theta,\phi) = e^{-j\cdot r^T \cdot k(\theta,\phi)} \qquad S = [\underline{s}_1, \underline{s}_2 \dots \underline{s}_L]$$

Received signal

$$\underline{x}(t) = S \cdot \underline{m}(t) + \underline{n}(t)$$

$$\underline{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \qquad R_{mm} = \mathbb{E}[\underline{m} \cdot \underline{m}^*]$$

$$= \begin{bmatrix} \mathbb{E}[m_1 m_1^*] & \mathbb{E}[m_1 m_2^*] & \mathbb{E}[m_1 m_3^*] \\ \mathbb{E}[m_2 m_1^*] & \mathbb{E}[m_2 m_2^*] & \mathbb{E}[m_2 m_3^*] \\ \mathbb{E}[m_3 m_1^*] & \mathbb{E}[m_3 m_2^*] & \mathbb{E}[m_3 m_3^*] \end{bmatrix}$$

As different sources are uncorrelated and the power is 1

$$\Rightarrow R_{mm} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{xx} = \mathbb{E}[\underline{x}(t)\,\underline{x}(t)^H]$$

$$= \mathbb{E}\left[\left[S\underline{m}(t) + \underline{n}(t)\right] \cdot \left[S\underline{m}(t) + \underline{n}(t)\right]^H\right]$$

$$= \mathbb{E}[S\underline{m}(t) \cdot \underline{m}(t)^H \cdot S^H] + \mathbb{E}[\underline{n}(t) \cdot \underline{n}(t)^H] + \mathbb{E}[S\underline{m}(t) \cdot n(t)^H] + \mathbb{E}[n(t) \cdot m(t)^H \cdot S(t)]$$

If $\underline{n}(t)$ is isotropic AWGN, $\mathbb{E}[\underline{n}(t) \cdot \underline{n}(t)^H] = \sigma_n^2 I_N$

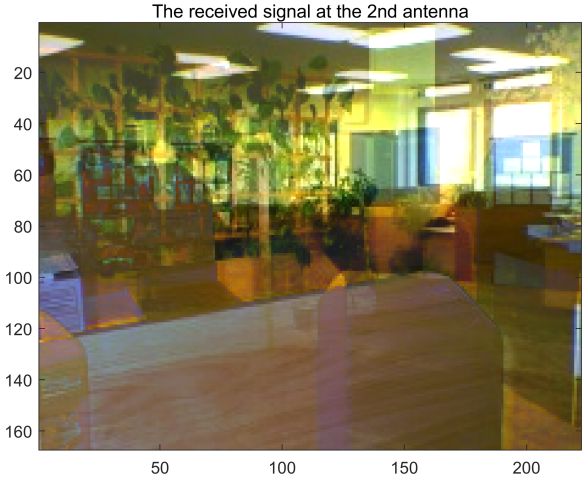
If $\underline{n}(t)$ and $\underline{m}(t)$ are uncorrelated, $\mathbb{E}[S\underline{m}(t)\,n(t)^H] = \mathbb{E}[n(t)\,\underline{m}(t)^H S(t)] = 0$

$$\Rightarrow R_{xx} = \mathbb{E}[S\underline{m}(t) \cdot \underline{m}(t)^H S^H] + \sigma_n^2 I_N$$

$$= S \cdot R_{mm} \cdot S^H + \sigma_n^2 I_N$$

3. Practical Covariance Matrix

The image received at the second antenna.



For the received signal with finite snapshots

$$X = [\underline{x}_1(t), \underline{x}_2(t), \underline{x}_3(t), \cdots \underline{x}_L(t)]$$

The covariance matrix can be calculated by

$$R_{xx} = \frac{1}{L} X \cdot X^H$$

5. Detection Problem

Observing the theoritical R_{XX} matrix

$$R_{XX} = S \cdot R_{sss} \cdot S^H + \sigma_n^2 I_N$$

$$= R_{signal} + \sigma_n^2 I_N$$

As there are M sources

$$\text{rank}\{R_{signal}\} = M$$

i.e. R_{signal} has $N-M$ duplicate eigenvalue $\lambda = 0$

Therefore, the eigen-decomposition of R_{signal} is

$$R_{signal} = E \Lambda E^H$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_M \end{bmatrix}_{N \times N}$$

For R_{XX}

$$R_{XX} = R_{signal} + \sigma_n^2 I_N$$

$$= E \Lambda E^H + \sigma_n^2 \cdot E E^H$$

$$= E [\Lambda + \sigma_n^2 I_N] E^H$$

The $D = \Lambda + \sigma_n^2 I_N$ is the eigenvalue matrix of R_{XX}

$$D = \begin{bmatrix} \lambda_1 + \sigma_n^2 & & & 0 \\ & \lambda_2 + \sigma_n^2 & & \\ & & \ddots & \\ & & & \lambda_M + \sigma_n^2 \\ & & & & \sigma_n^2 \\ 0 & & & & & \ddots \\ & & & & & & \sigma_n^2 \end{bmatrix}$$

The power of noise can be estimated by the average of $N-M$ smallest values of eigenvalues of the estimated covariance matrix of received signals

The number of sources can be estimated by

$M = N$ - multiplicity of minimum eigenvalues of R_{XX}

The eigenvalues of R_{XX} - theoretical are

$[9.9369, 4.9631, 0.0993, 0.0001, 0.0001]$

The eigenvalues of R_{XX} - our is

$[4.382 \times 10^5, 1.072 \times 10^4, 1.733 \times 10^2, 0.0101, 0.0100]$ (0.0101, 0.0100: noise)

$M = 3$, $\sigma_n^2 = \frac{0.0101 + 0.0100}{2} \approx 0.01$

The eigenvalues of R_{XX} - im is

$[2.178 \times 10^5, 2.039 \times 10^4, 1.909 \times 10^2, 7.061 \times 10^{-11}, -9.492 \times 10^{-11}]$ (7.061×10^{-11}, -9.492×10^{-11}: noise)

$M = 3$, $\sigma_n^2 = \frac{7.061 \times 10^{-11} + -9.492 \times 10^{-11}}{2}$

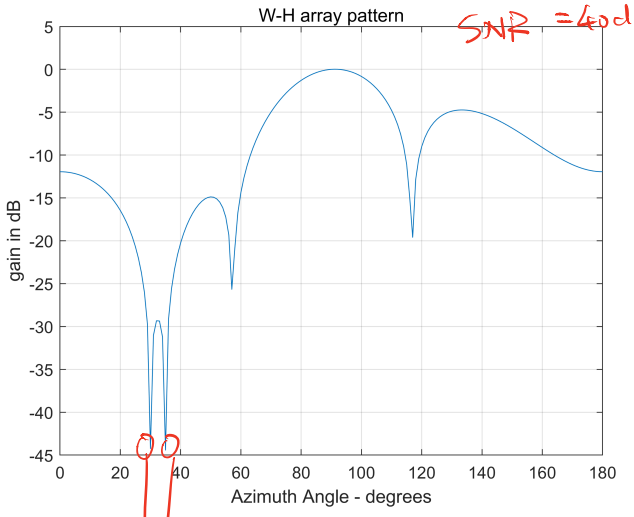
$= -1.216 \times 10^{-11}$

6. Estimation

The WIENER-HOPF Beamformer is used

the weights

$$\underline{w} = c \cdot R_{int}^{-1} S_d$$



SNR = 40dB

The pattern of WIENER-HOPF is shown on the left

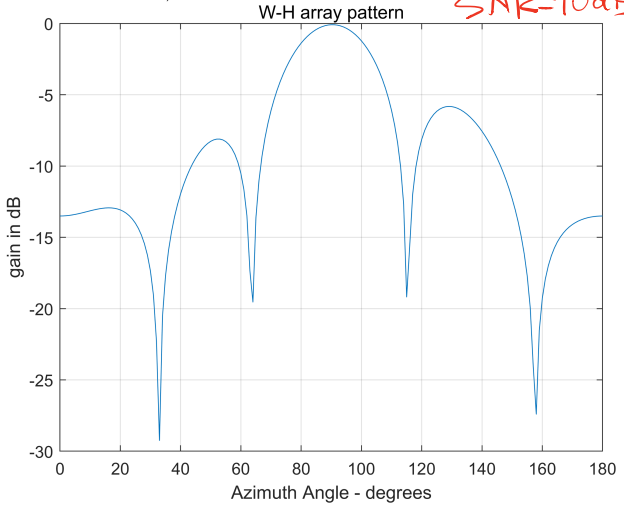
We can notice that the gains at 30° and 37° degrees are smallest, therefore the directions of the interferences are 30° and 37°

the interferences

For the noise level 10dB below the level of the sources

i.e. $\sigma_n^2 = \frac{1}{10} \cdot P_s = 0.1$

The result of WIENER-HOPF is



SNR=10dB

Compared with the situation where $\sigma_n^2 = 0.0031$. the results is bad, as the pattern is not changed too much from the initial pattern.

This result shows that WIENER-HOPF beamformer performs badly with 10dB SNR.

9. MUSIC Algorithm.

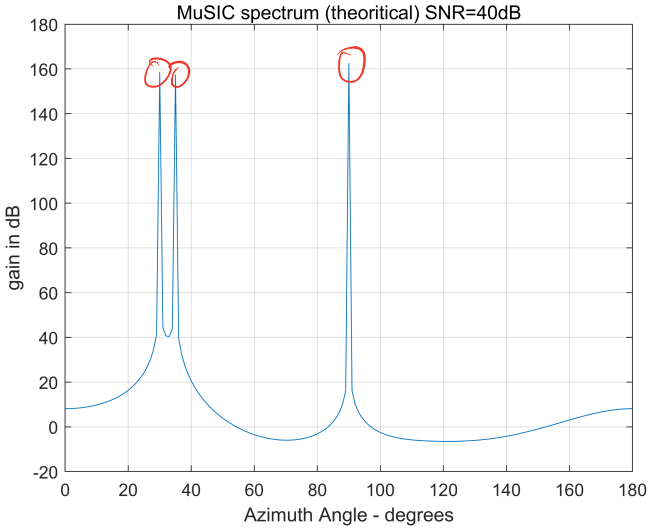
$$P(\rho) = \underline{S}(\rho)^H \cdot P_n \cdot \underline{S}(\rho) \quad \forall \rho$$

P_n is the projection matrix of $\lambda_i E_n$?

This equation can be simplified to

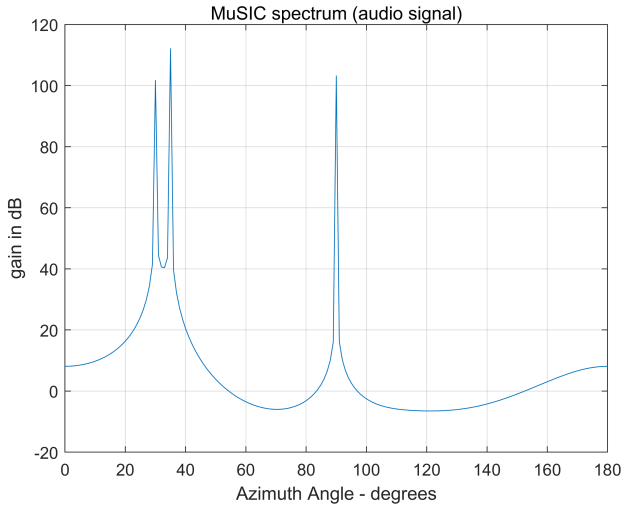
$$P(\rho) = \underline{S}(\rho)^H \cdot E_n \cdot E_n^H \cdot \underline{S}(\rho)$$

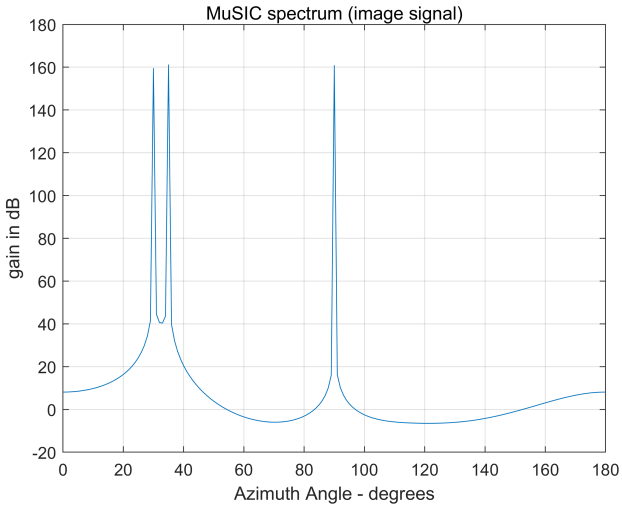
Matrix E_n is the eigenvector matrix of the $N-M$ smallest eigenvalues of R_{xx}



In result figure, we can notice that there are three DOAs, which are (30°,0°) (37°,0°) (90°,0°)

Apply the MUSIC Algorithm on audio and image signals





Both of the signals have the same DOAs, which are

$(35°, 0°)$ $(37°, 0°)$ $(98°, 0°)$

But the gains of each DOA are different for the two signals

N. Coherent Sources

$$R_{mm} = \mathbb{E}[\underline{m} \cdot \underline{m}^*]$$

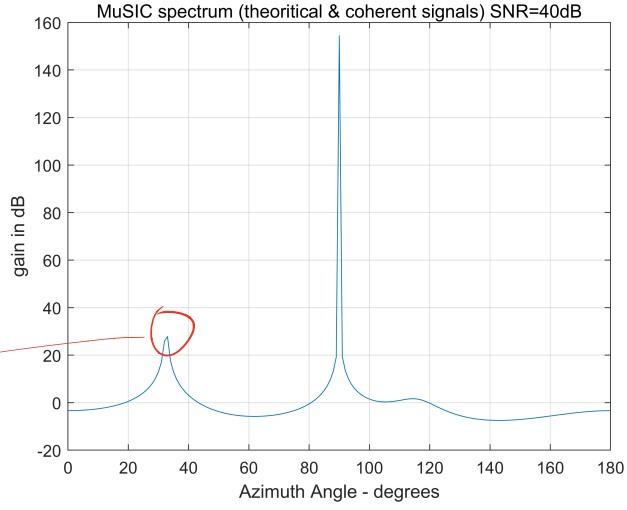
$$= \begin{bmatrix} \mathbb{E}[m_1 m_1^*] & \mathbb{E}[m_1 m_2^*] & \mathbb{E}[m_1 m_3^*] \\ \mathbb{E}[m_2 m_1^*] & \mathbb{E}[m_2 m_2^*] & \mathbb{E}[m_2 m_3^*] \\ \mathbb{E}[m_3 m_1^*] & \mathbb{E}[m_3 m_2^*] & \mathbb{E}[m_3 m_3^*] \end{bmatrix}$$

m_1 is the signal arriving at 35°
m_2 is the signal arriving at 37°.

Assuming m_1 and m_2 are fully correlated

we have $\mathbb{E}[m_1 m_2^*] = \mathbb{E}[m_2 m_1^*] = 1$

$$\Rightarrow R_{mm} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



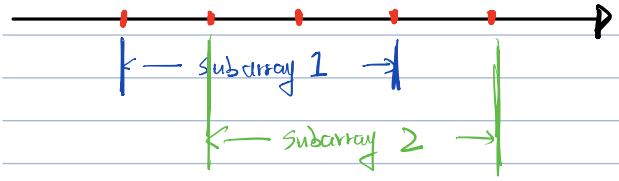
If the two sources from 30° and 35° are fully correlated,

MUSIC cannot seperate this two sources

MUSIC with Spatial Smoothing

$K=4$

Define the subarrays with the length K



There are L elements in the ULA

$\Rightarrow$ # subarrays $= L-K+1$

$N = L-K+1$

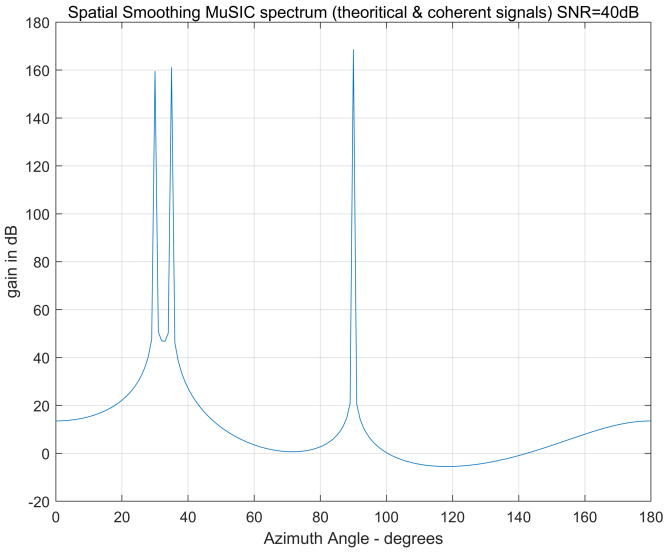
Using the smoothed covariance R_{xx_smooth} instead of the original R_{xx}

$$R_{xx\_smooth} = \frac{1}{N}\sum_{s=1}^{N} R_s$$

where R_s is the covariance of each subarray

$$R_s = R_{xx}(s:s+k-1, s:s+k-1)$$

The result of smoothing MUSIC is shown on the right.



We can notice that the sources from 30° and 31° can be separated

However, the size of smoothed covariance is KxK, which means that the aperture of array is reduced.

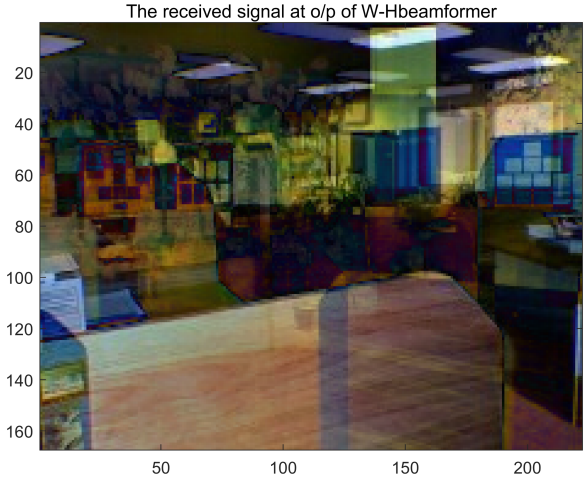
Therefore, the spatial smoothing technique is capable of addressing the coherent problem, with the penalty of reducing the aperture of array

12. Superresolution Techniques

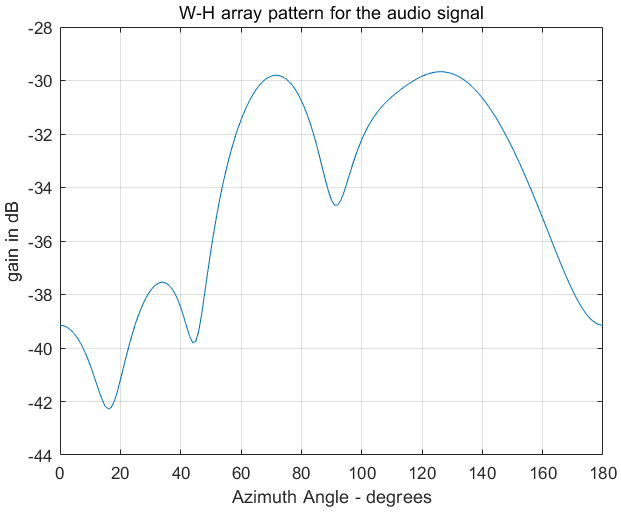
Applying the W-H beamformer to the audio signals with the desired direction of (90°, 0).

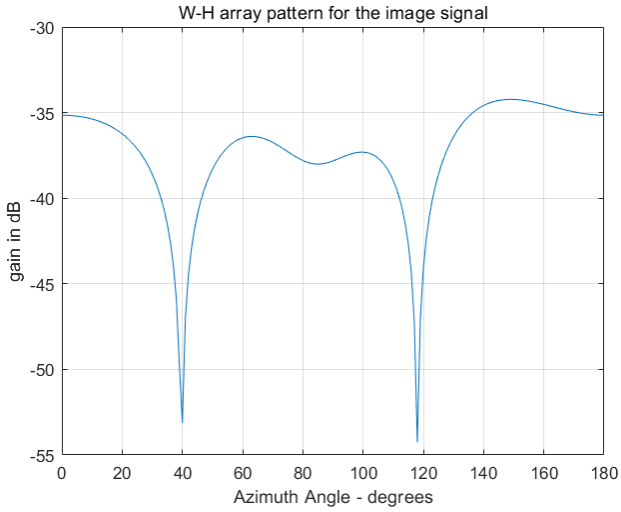
The received audio signal by W-H beamformer is not clear, and there is the same problem for the image signal.

The result of image signal received by W-H beamformer is shown on the right. There are lots of interference in this image and the image is not clear.



Check the W-H array pattern for both audio and image signal





These results indicate that the W-H array pattern is not good enough for these audio and image signals with desired direction of (90°, 0°)

Superresolution Beamformer

1°. Superresolution Beamformer based on DOA estimation

$$\underline{W} = P_{S_J}^{\perp} \cdot S_{\substack{\text{desired} \\ \text{signal}}}$$

According to the MUSIC Algorithm question 9, the DOAs of audio and image signals are

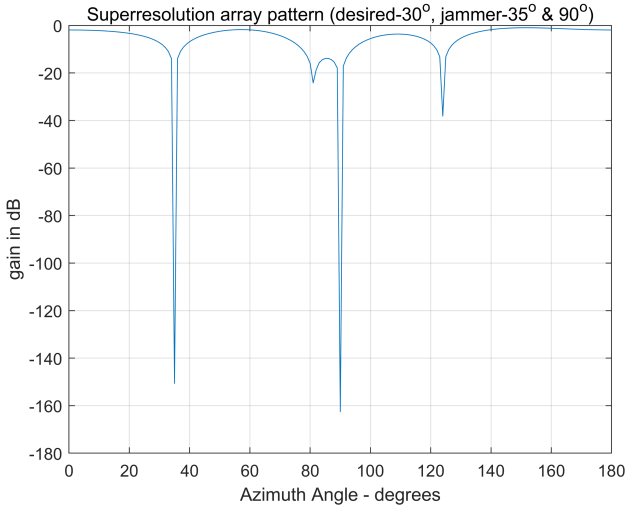
(30°, 0°) (35°, 0°) (90°, 0°)

If the signal with DOA of (30°, 0) is the desired signal, $S_{\substack{\text{desired} \\ \text{signal}}} = \underline{S}(30°, 0°)$ and $S_J = [\underline{S}(35°, 0°) \ \underline{S}(90°, 0°)]$

$$P_{S_J}^{\perp} = I_N - S_J \cdot (S_J^H \cdot S_J)^{-1} \cdot S_J^H$$

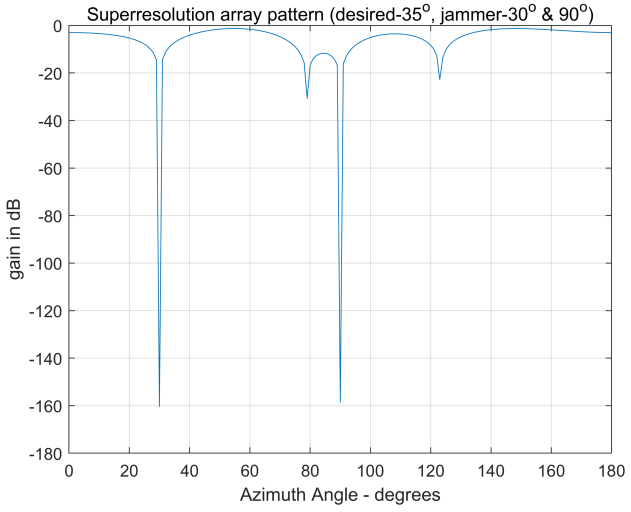
The pattern of the arrays with this superresolution beamformer and the result image is shown below

30°

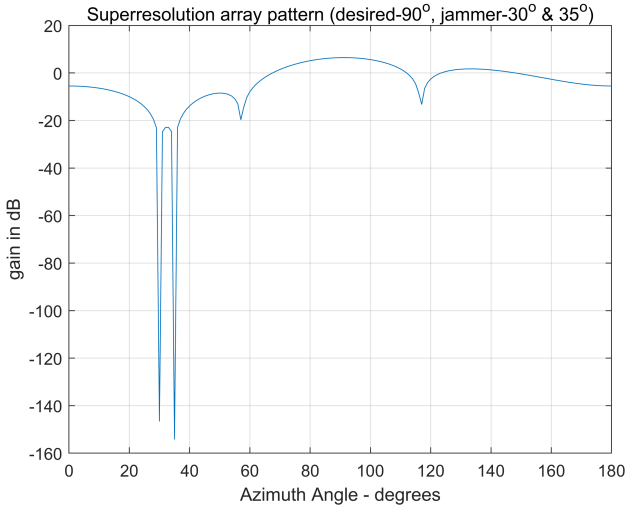


Do the same operations by setting the DOA of (35°,0) and (90°,0) as the DOA of desired signals. The result is shown below

35°



90°



2° Superresolution Beamformer not based on DOA estimation of interference

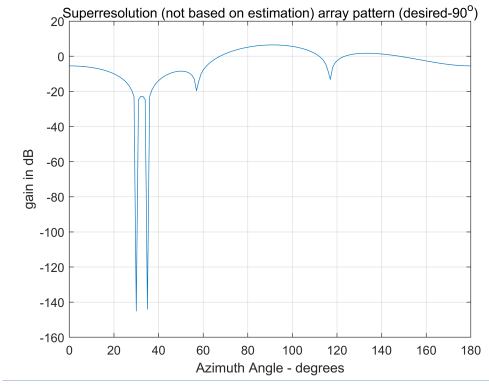
$$\underline{W} = P_{Enj}^{+} \cdot S_{\text{desired signal}}$$

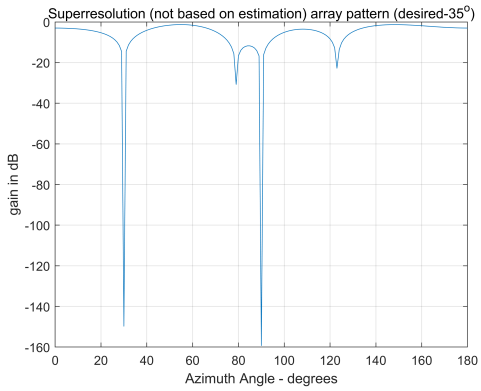
where E_{nj} is the interference subspace of R_{nj}

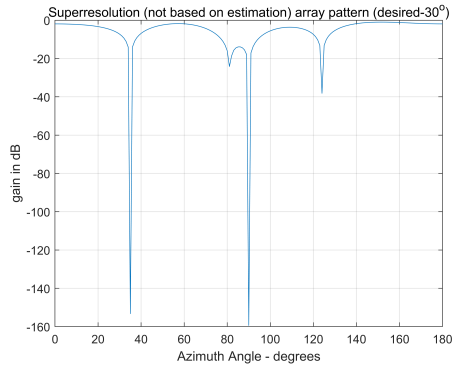
and R_{nj} is the covariance matrix generated by removing the effect of desired signal from R_{xx}

i.e. $R_{nj} = R_{xx} - \underbrace{P_d}_{\text{power of desired signal}} \cdot S_d \cdot S_d^H$

① Apply this beamformer on the Rxx-theoretical







We can notice that, although we don't know the DOAs of interference signals, the superresolution beamformer for interference cancellation can also be obtained.

② For Image & Audio signal

For the Rxx-theoretical, we know the power of desired signal Pd = 1. However, for Rxx-im. or Rxx-cal, we don't know the Pd. Therefore, it requires the estimation of Pd.

13. Pratical Detection Criterion

generate the 250 snapshots

```
array=[-2 0 0; -1 0 0; 0 0 0; 1 0 0; 2 0 0];
directions = [30, 0; 35, 0 ; 90, 0];

% random source signal from 3 directions
L = 250;
N = size(array, 1);
M = size(directions,1);
m = (randn(M,L) + 1i*randn(M,L)) / sqrt(2);

% random noise
sigma2 = 0.1;
noise = sqrt(sigma2) * (randn(N,L) + 1i*randn(N,L)) / sqrt(2);
S = spv(array,directions);

% received signal
X = S * m + noise;

% covariance matrix
Rxx = X*X' / length(X(1,:));
```

chang σ_n^2 to evaluate the performance under different SNR

1° when $\sigma^2 = 0.0001$, i.e. SNR = 40dB
the eigenvalues of Rxx are
10.7398, 4.8704, 0.1072, 0.0001, 0.0001
It is easy to identify there are M=3 sources

2° However, when $\sigma^2 = 0.1$, i.e. SNR = 10 dB.
the eigenvalues of Rxx are
10.7387, 4.8214, 0.1235, 0.1034, 0.0890
It is no easy to identify there are M=3 sources, because the last 3 eigenvalues are similar

Using AIC and MDL methods to estimate the number of sources M

$$AIC(k) = -2\ln\left(\frac{\prod_{i=k+1}^{N} d_i^{\frac{1}{(N-k)}}}{\frac{1}{N-k}\cdot\sum_{i=k+1}^{N} d_i}\right)^{(N-k)L} + 2k(2N-k)$$

$$\hat{M} = \arg\min_k \{AIC(k)\}$$

or $\arg\min_k \{MDL(k)\}$

$$MDL(k) = -\ln\left(\frac{\prod_{i=k+1}^{N} d_i^{\frac{1}{(N-k)}}}{\frac{1}{N-k}\cdot\sum_{i=k+1}^{N} d_i}\right)^{(N-k)L} + \frac{1}{2}k(2N-k)\ln L$$

For both high SNR (e.g. 40dB) or log SNR (e.g. 10dB)

both AIC and MDL methods estimate the correct result $M=3$

MATLAB Codes (Main Parts)

1 & 2

```
4 —    array=[-2 0 0; -1 0 0; 0 0 0; 1 0 0; 2 0 0]; % locations of array sensors
5 —    directions = [30, 0; 35, 0 ; 90, 0]; % DOAs of sources
6 —    Z = my_pattern(array);  % array pattern
7 —    figure(1);
8 —    plot2d3d(Z,[0:180],0,'gain in dB','initial pattern');
9
10 —   S = spv(array,directions); % array manifold vectors
11
12 —   sigma2 = 0.0001; % noise power
13 —   Rmm = eye(3);
14 —   Rxx_theoretical = S*Rmm*S' + sigma2*eye(5,5); % covariance matirx of received signals
```

3

```
19 —   load Xaudio/Xaudio.mat
20 —   load Ximage/Ximage.mat
21 —   soundsc(real(X_au(2,:)), 11025);
22 —   displayimage(X_im(2,:),image_size, 201,'The received signal at the 2nd antenna');
23
24     % covariance matrix in practice
25 —   Rxx_au = X_au*X_au' / length(X_au(1,:));
26 —   Rxx_im = X_im*X_im' / length(X_im(1,:));
```

4

```
28     %estimated parameters
29 —   directions = [];
30 —   Rmm = [];
31 —   S = [];
32 —   sigma2 = [];
```

6

```
%% Estimation Problem
Sd = spv(array, [90,0]);
wopt = inv(Rxx_theoretical) * Sd;
Z = my_pattern(array, wopt);
plot2d3d(Z,[0:180],0,'gain in dB','W-H array pattern');
```

9.

```
function nc = music(array, Rxx, M, K)
% MUSCI algorithm
N = size(Rxx,1);
if nargin<=3
    K = N; % for spatital smoothing techniques
end

[E, D] = eig(Rxx); %eigenvalues and eigenvectors
D = diag(D);
[~,I] = sort(D);

E = E(:,I);
En = []; % eigenvectors of nosie subspace
for i=1:N-M
    En = [En E(:,N-M-i+1)];
end

nc = []; % gain of different directions
for azimuth = 0:180
    S = spv(array,[azimuth,0]);
    S = S(1:K);
    nc = [nc S'*(En*En')*S];
end

nc = -10*log10(nc);
end
```

11.

```m
function nc = smooth_music(array, Rxx, M, K)
% MUSCI algprithm with spatitial smoothing technique

L = size(Rxx, 1);
N = L-K+1;
Rxxs = zeros(K,K);
for i=1:N
    Rxxs=Rxxs+Rxx(i:i+K-1,i:i+K-1);
end
Rxxs = Rxxs/N; % smoothed covariance matrix

nc = music(array, Rxxs, M, K);
end
```

12 Based on DOA estimation

```m
array=[-2 0 0; -1 0 0; 0 0 0; 1 0 0; 2 0 0];
load Xaudio/Xaudio.mat
load Ximage/Ximage.mat

%% WH beamformer
Rxx_au = X_au*X_au' / length(X_au(1,:));
Sd = spv(array, [90,0]);
wopt = inv(Rxx_au) * Sd;
Z = my_pattern(array, wopt);
plot2d3d(Z,[0:180],0,'gain in dB','W-H array pattern for the audio signal');

yt=wopt'*X_au;
% soundsc(real(yt), 11025);

Rxx_im = X_im*X_im' / length(X_im(1,:));
Sd = spv(array, [90,0]);
wopt = inv(Rxx_im) * Sd;
yt=wopt'*X_im;
yt = abs(yt);
yt = (yt-min(yt)) * 255/(max(yt)-min(yt)); % normalize the image to [0,255]
displayimage(yt, image_size, 202,'The received signal at o/p of W-Hbeamformer');
Z = my_pattern(array, wopt);
plot2d3d(Z,[0:180],0,'gain in dB','W-H array pattern for the image signal');
```

```
%% superresolution techniques
array=[-2 0 0; -1 0 0; 0 0 0; 1 0 0; 2 0 0];
directions = [30, 0; 35, 0 ; 90, 0];
% DOAs - 30, 35 and 90;
S = spv(array,directions);
Sd = S(:,2); % desired signal is 35

S_J = [S(:,1) S(:,3)]; % DOA of jammer signal

P_J = S_J*inv(S_J'*S_J)*S_J';
P_J_orth = eye(size(P_J))-P_J;

wsuper = P_J_orth*Sd;
Z = my_pattern(array, wsuper);
plot2d3d(Z,[0:180],0,'gain in dB','Superresolution array pattern (desired-35^o, jammer-30^o & 90^o)');

yt=wsuper'*X_im;
yt = abs(yt);
yt = (yt-min(yt)) * 255/(max(yt)-min(yt)); % normalize the image to [0,255]
displayimage(yt, image_size, 202,'The received signal at o/p of Superresolution Beamformer (35^o)');
```

Not Based on DOA estimation of interference

```
%% not based on DOA estimation (Theoretical Signal)
array=[-2 0 0; -1 0 0; 0 0 0; 1 0 0; 2 0 0];
directions = [30, 0; 35, 0 ; 90, 0];
S = spv(array,directions);
sigma2 = 0.1; % noise power SNR=10dB

Rmm = eye(3);
Rxx_theoretical = S*Rmm*S' + sigma2*eye(5,5);
Sd = S(:,3); % desired signale - 90

R_nJ = Rxx_theoretical - Sd*Sd';

[E, D] = eig(R_nJ);
D = diag(D);
[D,I] = sort(D,'descend');
E = E(:,I);
Ej = []; % eigenvectors of interference subspace
Ej = E(:,1:2);

P_nJ = Ej*inv(Ej'*Ej)*Ej';
P_nJ_orth = eye(size(P_nJ))-P_nJ;
wsuper = P_nJ_orth*Sd;
Z = my_pattern(array, wsuper);
plot2d3d(Z,[0:180],0,'gain in dB','Superresolution not based on estimation array pattern (desired-90^o)');
```

13

```
array=[-2 0 0; -1 0 0; 0 0 0; 1 0 0; 2 0 0];
directions = [30, 0; 35, 0 ; 90, 0];

% random source signal from 3 directions
L = 250;
N = size(array, 1);
M = size(directions,1);
m = (randn(M,L) + 1i*randn(M,L)) / sqrt(2);

% random noise
sigma2 = 0.1;
noise = sqrt(sigma2) * (randn(N,L) + 1i*randn(N,L)) / sqrt(2);
S = spv(array,directions);

% received signal
X = S * m + noise;

% covariance matrix
Rxx = X*X' / length(X(1,:));

eigenv = sort(real(eig(Rxx)), 'descend')

%% AIC and MDL criterion
AIC = zeros(1,N);
MDL = zeros(1,N);
for k = 0:N-1
    a = vpa(prod(eigenv(k+1:N) .^(1/(N-k))),6);
    b = (1/(N-k)) * sum(eigenv(k+1:N));
    AIC(k+1) = -2 * log((a / b).^((N-k)*L)) + 2*k*(2*N-k);
    MDL(k+1) = -1 * log((a / b).^((N-k)*L)) + 1/2 * k*(2*N-k) * log(L);
end

[~,M_AIC] = min(AIC);
[~,M_MDL] = min(MDL);

M_AIC = M_AIC - 1;
M_MDL = M_MDL - 1;

```