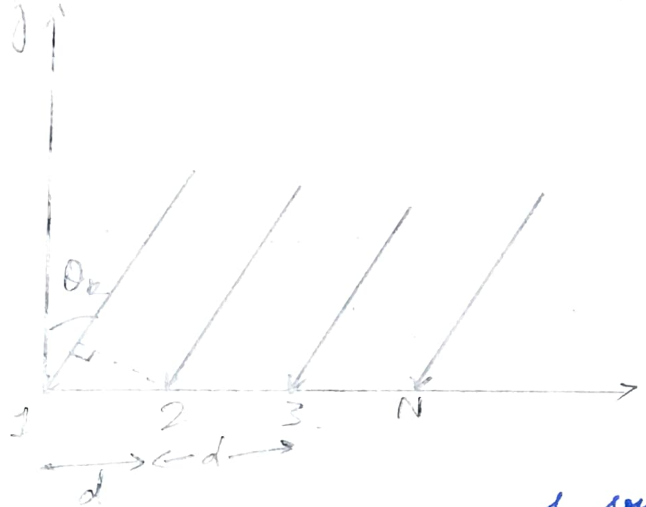
Assignment-2

Real Time Signal Processing

Pratyush Jaiswal
18EE35014

MUSIC (Multiple Signal Classification) is an algorithm used for frequency estimation and radio direction finding. Mostly signal processing problems have an objective of estimating measurements from a set of constants parameter upon which the received signals depend. Pisarenko (1973) has one of the first to exploit the structure of data model and ~~the~~ tried to eliminate limitations of earlier approaches by doing so in the context of estimation of parameter of complex sinusoids in additive noise using a covariance approach. Some exploited measurement model in case of sensor arrays of arbitrary form. Schmidt accomplished this first by deriving a complete geometric solution in absence of noise resulting Music Algorithm.

Mathematical Model of DoA Estimation :-



θ_k = direction angle

$d \leq \frac{\lambda}{2}$ small

Let N = no. of sensors, D = no. of sources (N>D).
Given sources are point sources, narrow band and far field, therefore we can assume that wave-fronts are planar. Let wavefront signal be $S_k(t) = s_k(t) e^{j\omega_k t}$; $k = 1, 2, \ldots D$ where $s_k(t)$ is complex envelope of $S_k(t)$.

$s_k(t)$ → signal strength ; ω_k = angular frequency of $S_k(t)$.

Let wavefront signal's be taking t_1 time for narrowband $S_k(t-t_1) \simeq S_k(t)$ and delay sees in exponential.

Using 1st array element as reference point, at moment 't' induction signal of array elements 'm' to the k^{th} signal source is given as $s_k(t) e^{-j(m-1)\frac{2\pi d \sin\theta_k}{\lambda}}$

where $(m-1)\frac{d\sin\theta_k}{\lambda}$ is the phase difference between m^{th} & 1st element.

$$\therefore \underset{\text{O/P signal}}{X_m(t)} = \sum_{k=1}^{D} s_k(t) e^{-j(m-1)\frac{2\pi d \sin\theta_k}{\lambda}} + \underset{\text{Noise}}{N_m(t)}$$

Now in the MUSIC method this x_m is signal measured by m^{th} elements collecting all signals in a signal vector x, considering of P complex exponentials whose frequencies ω are unknown in the presence of Gaussian white noise N is given by linear model, $X = AS + N$,

$A = [a(\omega_1) \ldots a(\omega_P)]$ is $M \times P$ Vandermonde matrix of steering vectors $a(\omega) = [1, e^{j\omega}, e^{2j\omega}, \ldots, e^{j(M-1)\omega}]^T$

and $S = [s_1 \ldots s_P]^T$ is the amplitude vector A. A crucial assumption is that no. of sources $P <$ no. of elements in measurement vector.

$M \times M$ autocorrelation matrix of X is given as

$$R_x = A R_S A^H + \sigma^2 I$$

$I = M \times M$ Identity Matrix

σ^2 = Noise Variance

R_S = $P \times P$ autocorrelation matrix of S.

$$\hat{R}_x = \frac{1}{N} x x^H, \quad N > M$$

Given the estimate of R_x, MUSIC estimates the frequency count of signal / autocorrelation matrix using eigenspace method.

R_x is Hermitian, and all M eigenvectors, are orthogonal; Sorting eigenvalues of R_x in descending order, the eigenvectors corresponding to largest P eigenvalues span the signal subspace U_S. Remaining $M-P$ eigenvalues span noise subspace U_N. $U_S \perp U_N$.

MUSIC method uses all the eigenvectors spanning the noise subspace to improve the performance of the Pisarenko estimation.

Any signal vector $e \in U_s$ is $\perp$ to U_N. So $e \perp v_i$ ∃ eigenvectors $\{v_i\}_{i=P+1}^{M}$ spanning noise subspace.

In order to measure the degree of orthogonality of e with respect to all $v_i \in U_N$. music algo. defines a squared ~~form~~ norm

$$d^2 = \| U_N^H e \|^2 = e^H U_N U_N^H e = \sum_{i=P+1}^{M} |e^H v_i|^2$$

U_N = matrix of eigenvectors spanning of U_N. If $e \in U_s$, $d^2 = 0$. Taking a reciprocal of the squared norm expression creates sharp peaks at the signal frequencies. The frequency estimation function of MUSIC is

$$\hat{P}_{MU}(e^{j\omega}) = \frac{1}{e^H U_N U_N^H e} = \frac{1}{\sum_{i=P+1}^{M} |e^H v_i|^2}$$ where

v_i are noise eigen vectors and $e = [1 \; e^{j\omega} \; e^{2j\omega} \cdots e^{j(M-1)\omega}]^T$ is

the candidate steering vector. Locations of the p largest peaks of the estimation function gives the frequency estimates for the P signal components.

$$\hat{\omega} = \arg\max_{\omega} \hat{P}_{MU}(e^{j\omega})$$

Characteristics

→ MUSIC algorithm simultaneously measures multiple signals.

→ High precision measurement ~~&~~ high resolution for antenna signals.

→ It outperforms simple methods such as picking peaks of DFT spectra in presence of noise when the no. of components is known in advance exploiting knowledge of this number to ignore the noise in final report.

→ One main disadvantage is that it requires the no. of components to be known in advance so ~~that~~ the original method cannot be used in more general case.

→ It assumes coexistent sources to be uncorrelated which limits pratical applications.

Uses

1. It is used in frequency estimation & radio direction finding.

2. It is used in modified version Time-Reversal MUSIC applied to computational time-reversal imaging.

3. fast detection of the DTMF frequencies (Dual tone multi-freq signalling) in form of C library - libmusic.

Pseudocode :

1. Simulate Setup.

2. i) Compute Covariance matrix S.

ii) Compute eigen decomposition of $S = E \Lambda E^{-1}$

iii) Estimate number of sources from step (ii)

iv) Based on (iii) partition $E = [E_s \ E_N]$ E_N : noise part.

v) Sweep $P(\theta) = \dfrac{1}{a_\theta^H E_N E_N^H a(\theta)}$ in the range

of (θ) and find peaks & we get

estimate of θ_1 & θ_2.

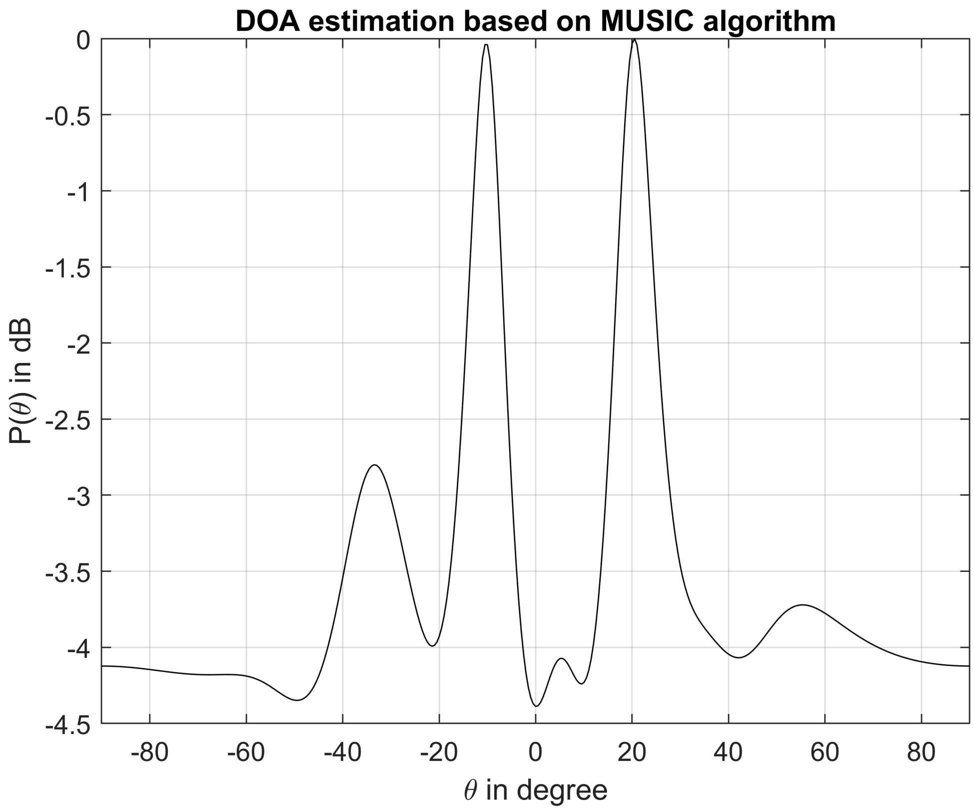
DISCUSSION :

1. When θ coincides with the actual direction of arrival $P_{mu}(\theta)$ goes to zero when there is perfect estimation of covariance matrix.

2. The resolution of the MUSIC algorithm increases with increase in sensors, difference between incident angle & number of snapshots. If $d > \lambda/2$, the algorithm ~~reports~~ reports a false peak.

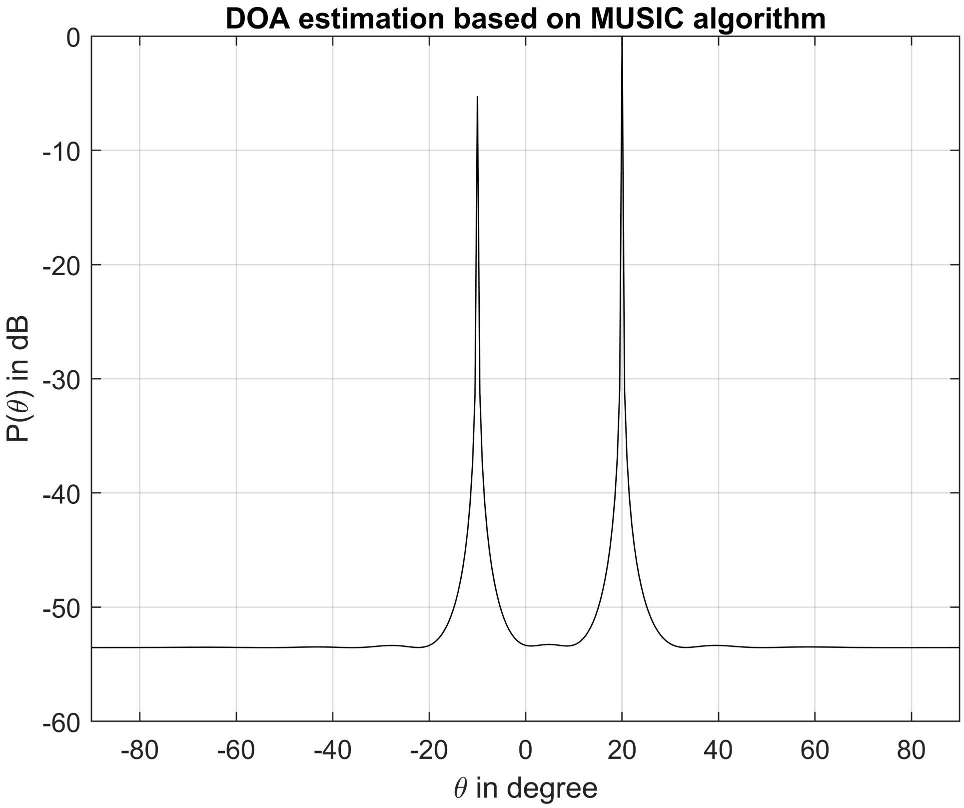
3. The MUSIC algorithm only works when $M > D$, because that is when the signal & noise subspace are different else when $M < D$ the noise & signal subspace get mixed.

```
%MUSIC Algorithm for DOA
clc;
clear;

azimuth = [-10 20]/180*pi;
doa = azimuth;

N = 4500;
f = 2*10^9;
snr=5;
w = 2*pi*f*[1 1]'; %Angular frequency
M = 10;             %Number of array elements
P = length(w);      %Number of signal
lambda = 150/1000;  %Wavelength
d = lambda/2;       %Element spacing
D = zeros(P,M);     %Creating a zero matrix with P rows and M columns

for k=1:P
  D(k,:) = exp(-1i*2*pi*d*sin(doa(k))/lambda*(0:M-1));
end
D=D';
%Generating Signals and Noise
Xs = 2*exp(1i*(w*(1:N))); %Generating the signal
X = D*Xs;
X = awgn(X,snr); %Insert gaussian White Noise
R = X*X';
[N,V] = eig(R);     %Find Eigenvalues and Eigenvectors of R
NN = N(:,1:M-P);    %Estimate Noise subspace

%Theta search for peak finding
theta = -90:0.5:90; %peak search
Pmusic = zeros(length(theta),1); %P music function
for ii=1:length(theta)
    SS = zeros(1,length(M));
    for jj=0:M-1
        SS(1+jj) = exp(-1i*2*jj*pi*d*sin(theta(ii)/180*pi)/lambda);
    end
    PP = SS*(NN*NN')*SS';
    Pmusic(ii) = abs(1/PP);
end
%%Plotting the results of theta and Pmusic function
figure;
Pmusic = 10*log10(Pmusic/max(Pmusic));
plot(theta,Pmusic,'-k');
xlabel('\theta in degree');
ylabel('P(\theta) in dB');
title('DOA estimation based on MUSIC algorithm');
xlim([-90 90]);
grid on;
```




```
%Modification in MUSIC Algorithm for Coherent Sources
J = fliplr(eye(M)); %anti Matrix
R = R+J*conj(R)*J;  %Modified R matrix
[N,V] = eig(R);     %Find Eigenvalues and Eigenvectors of R
NN = N(:,1:M-P);    %Estimate Noise subspace

%Theta search for peak finding
theta = -90:0.5:90; %peak search
Pmusic = zeros(length(theta),1); %P music function
for ii=1:length(theta)
    SS = zeros(1,length(M));
    for jj=0:M-1
        SS(1+jj) = exp(-1i*2*jj*pi*d*sin(theta(ii)/180*pi)/lambda);
    end
    PP = SS*(NN*NN')*SS';
    Pmusic(ii) = abs(1/PP);
end
%%Plotting the results of theta and Pmusic function
figure;
Pmusic = 10*log10(Pmusic/max(Pmusic));
plot(theta,Pmusic,'-k');
xlabel('\theta in degree');
ylabel('P(\theta) in dB');
title('DOA estimation based on MUSIC algorithm');
xlim([-90 90]);
```

```
grid on;
```



DOA estimation based on MUSIC algorithm
0
-10
-20
-30
-40
-50
-60
P(θ) in dB
-80
-60
-40
-20
0
20
40
60
80
θ in degree