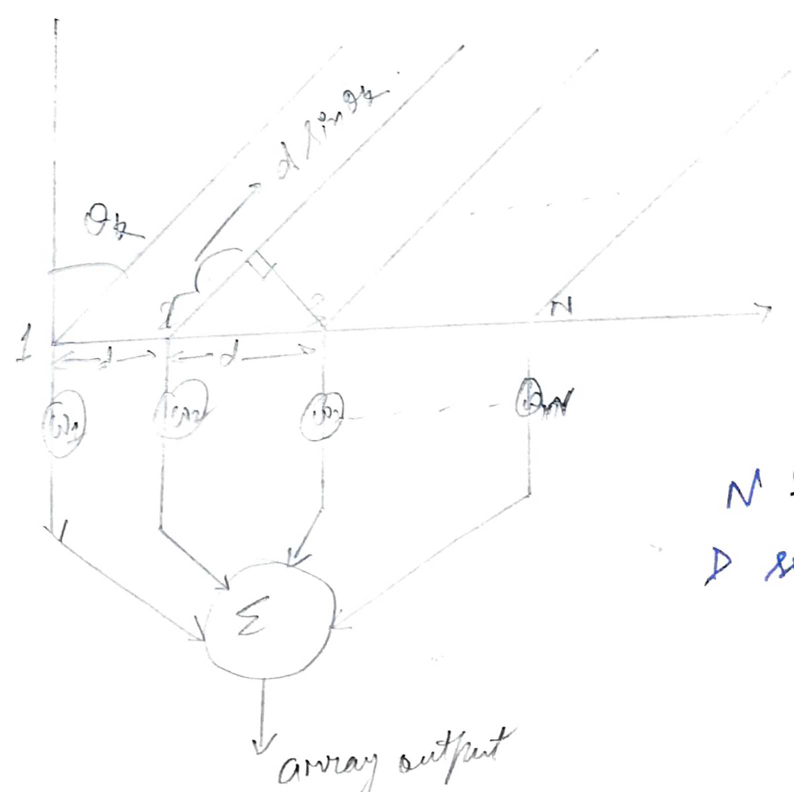
Assignment - 3

Pratyush Jaiswal
18EE35014

Capon's Minimum Variance Distortionless Response Algorithm:

Capon's MVDR Algorithm finds maximum likelihood estimation of the power of the interested signals from their directions, which means to form a beam pointing towards the looking direction while nulls the directions of interference. This method uses array ~~[illegible]~~ weights, which are obtained by minimizing, output power subject to unity constraint in the looking direction.

Mathematical Model:



N sensors
D sources

Let wavefront signal

$$S_k(t) = s_k(t) e^{j\omega_k t}$$

$$k = 1, 2, \ldots D$$

(sources are point source, narrow band, far field).

$$\therefore \quad S_k(t - t_1) = s_k(t) e^{j\omega_k (t - t_1)}$$

Taking first sensor as reference, induction signal of sensor "m" to "k^{th}" signal source is given as

$$s_k(t) e^{-j(m-1)\frac{2\pi d \sin\theta_k}{\lambda}}$$

path difference $\Rightarrow$ Phase difference

$(m-1) d \sin\theta_k$ $\quad$ $\frac{2\pi}{\lambda}(m-1) d \sin\theta_k$.

$$\therefore \quad x_m(t) = \sum_{k=1}^{D} s_k(t) e^{-j(m-1)\frac{2\pi d \sin\theta_k}{\lambda}} + N_m(t)$$

* Signal and noise are uncorrelated.

$$\Rightarrow \quad x_m(t) = \sum_{k=1}^{D} a_m(\theta_k) s_k(t) + N_m(t)$$

$$a_m(\theta_k) = e^{-j(m-1)\frac{2\pi d \sin\theta_k}{\lambda}}$$

Collecting output signals of all sensors

$$X = AS + N$$

$$X = [x_1(t) \; x_2(t) \; \ldots \; x_N(t)]^T_{N\times 1}$$

$$S = [s_1(t) \; s_2(t) \; \ldots \; s_D(t)]^T_{D\times 1}$$

$$A = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ e^{-j\phi_1} & e^{-j\phi_2} & \ldots & e^{-j\phi_D} \\ \vdots & & & \\ e^{-j(N-1)\phi_1} & e^{-j\phi_2(N-1)} & \ldots & e^{-j(N-1)\phi_D} \end{bmatrix}_{N\times D}$$

$$\phi_k = \frac{2\pi d}{\lambda} \sin\theta_k$$

$$N = [N_1(t) \; N_2(t) \; \ldots \; N_N(t)]^T_{N\times 1}$$

Minimum Output Energy:

Minimize total output energy while simultaneously keeping gain of array on the desired signal fixed. Since signal gain is fixed, any reduction in output energy is obtained by suppressing interference

$$Y = W^H X$$

$$W = [w_1 \; w_2 \; \ldots \; w_N]^T_{N\times 1}$$

$$Y = (W^H A) S + (W^H N)$$

In order for output for filter to be distortionless, gain corresponding to signal should be 1.

$$w^H A = I_{p \times p}$$

$$w^H a_\theta = 1, \quad a_\theta \text{ is a column of } A$$

$$L(w; d) = E\left[|y_\theta|^2\right] = E\left[|w^H x|^2\right] + d\left[w^H a_\theta - 1\right)$$

$$= w^H E[x x^H] w + d(w^H a_\theta - 1)$$

$$= w^H R w + d(w^H a_\theta - 1)$$

$$\frac{\partial L}{\partial w^H} = R w + d a_\theta = 0$$

$$w_{min} = -R^{-1}(d a_\theta) = -d(R^{-1} a_\theta)$$

$$w_{min}^H a_\theta = 1$$

$$(-d)\, a_\theta^H R^{-1} a_\theta = 1$$

$$-d = \frac{1}{a^H R^{-1} a_\theta}$$

$$\therefore \; w_{min} = \frac{R^{-1} a_\theta}{a_\theta^H R^{-1} a_\theta}$$

$$\text{Energy } (P) = w_{min}^H R w_{min}$$

$$= \frac{a_\theta^H R^{-1} R R^{-1} a_\theta}{(a_\theta^H R^{-1} a_\theta)(a_\theta^H R^{-1} a_\theta)} = \frac{1}{a_\theta^H R^{-1} a_\theta}$$

$$\therefore \quad P_{mvdr}(\theta) = \frac{1}{a_\theta^H R^{-1} a_\theta}$$

For θ at DOA, P_{mvdr} will be at peak.

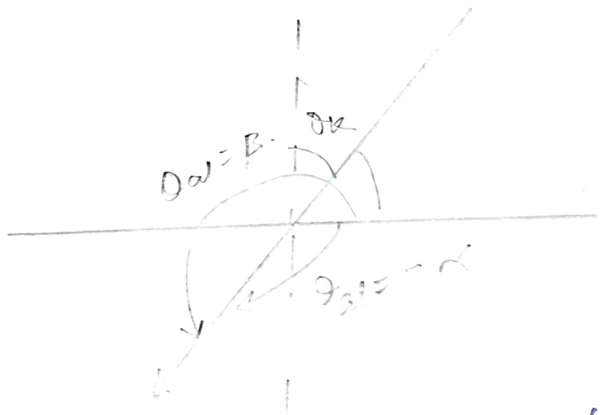
* Improvement for coherent signals,

$$R_{new} = A R_s A^H + J[A R_s A^H] + 2\sigma^2 I_{N \times N}$$

$J =$ N^{th} order anti Matrix

The modified R must be used to obtain DOA value by finding the peak.

* Conversion of azimuthal to θ_k



$$\theta_k = \cos^{-1}(\sin(\theta_{al}))$$

Applications

→ for directional audio capture in presence of multiple audio sources and noise ~~on~~ simultaneously.

→ Used in SONAR arrays for underwater signal source DOA.

→ Smart Antennas that automatically orient themselves towards direction of signal source to obtain maximum signal amplitude.

Results contain both modified & unmodified ones.

→ MVDR maximizes output SNR while keeping amplitude of interested signals from known directions unchanged because of unity constraint.

```matlab
%Initialization of parameters for MVDR estimation
clc;clear;
azimuth = [-10 20]/180*pi;
doa = azimuth;
N = 4500;            %No of sources
f = 2*10^9;          %Freq=2 GHz
snr=5;               %SNR
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
```

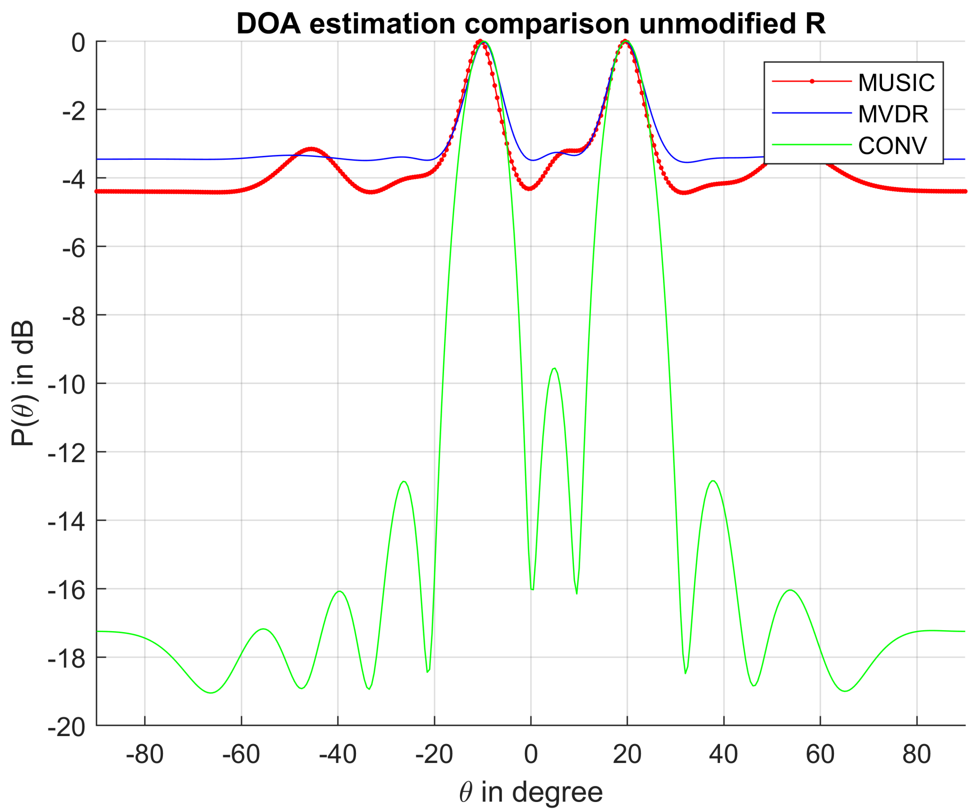
```matlab
for k=1:P
  D(k,:) = exp(-1i*2*pi*d*sin(doa(k))/lambda*(0:M-1));
end
D=D';
%Generating Signals and Noise
Xs = 2*exp(1i*(w*(1:N))); %Generating the signal
X = D*Xs;
X = awgn(X,snr);          %Insert Gaussian White Noise
R = X*X';                 %Data Covariance Matrix
```

```matlab
%Unmodified R applied MVDR and MUSIC calculations
[N,~] = 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 storing array
Pmvdr = zeros(length(theta),1);  %P mvdr storing array
Pconv = zeros(length(theta),1); %P conv storing array
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);
    PQ = SS*(inv(R))*SS';
    Pmvdr(ii) = abs(1/PQ);
    Z = SS*R;
    Z=Z*SS';
    PC = (SS*SS')/abs(Z);
    Pconv(ii) = abs(1/PC);
end
```

```matlab
%Plotting the results of theta ,P music and P mvdr functions
%In the unmodified R case
%%Plotting the results of theta and Pmusic function
```

```
figure;
hold on
Pmusic = 10*log10(Pmusic/max(Pmusic));
plot(theta,Pmusic,'.-r');
Pmvdr = 10*log10(Pmvdr/max(Pmvdr));
plot(theta,Pmvdr,'-b')
Pconv = 10*log10(Pconv/max(Pconv));
plot(theta,Pconv,'-g');
xlabel('\theta in degree');
legend({'MUSIC','MVDR','CONV'});
ylabel('P({\theta}) in dB');
title('DOA estimation comparison unmodified R');
xlim([-90 90]);
grid on;
```

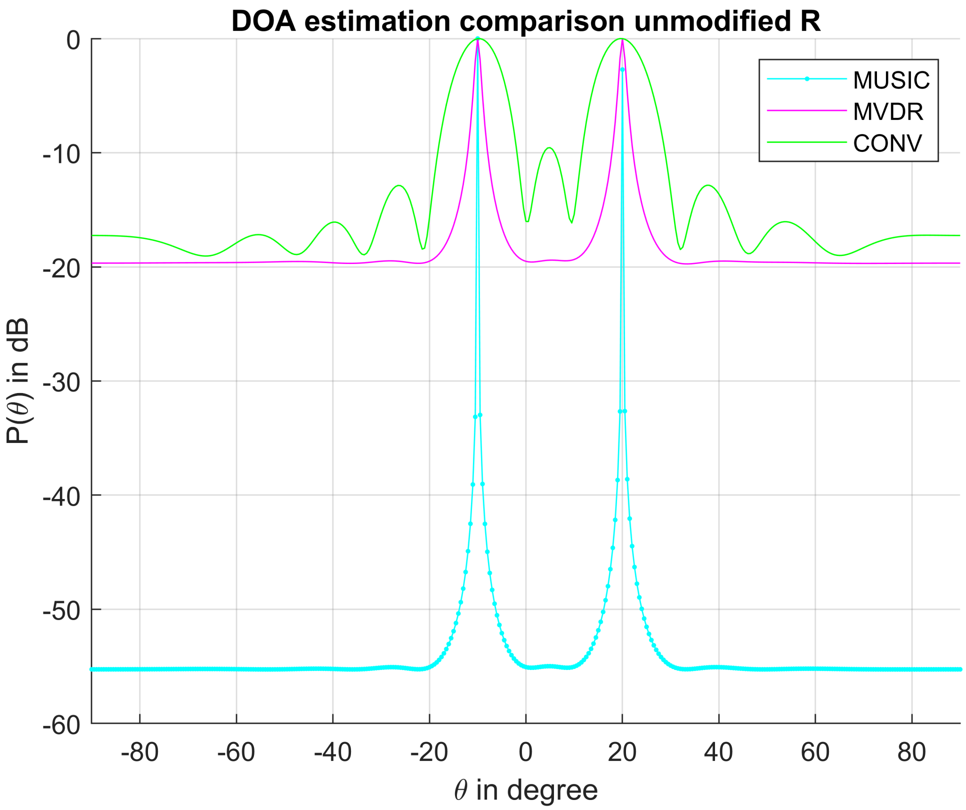



```
%MVDR and MUSIC estimations using modified R
%Unmodified R applied MVDR and MUSIC calculations
J=fliplr(eye(M));   %Anti Matrix
R = R+J*conj(R)*J;  %Modified R matrix
[N,~] = eig(R);     %Find Eigenvalues and Eigenvectors of R
NN = N(:,1:M-P);    %Estimate Noise subspace

%Theta search for peak finding
theta = -90:0.5:90; %peak search
Pmusic = zeros(length(theta),1); %P music storing array
Pmvdr = zeros(length(theta),1);  %P mvdr storing array
Pconv = zeros(length(theta),1); %P conv storing array
```

```
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);
    PQ = SS*(inv(R))*SS';
    Pmvdr(ii) = abs(1/PQ);
    Z = SS*R;
    Z=Z*SS';
    PC = (SS*SS')/abs(Z);
    Pconv(ii) = abs(1/PC);
end
```

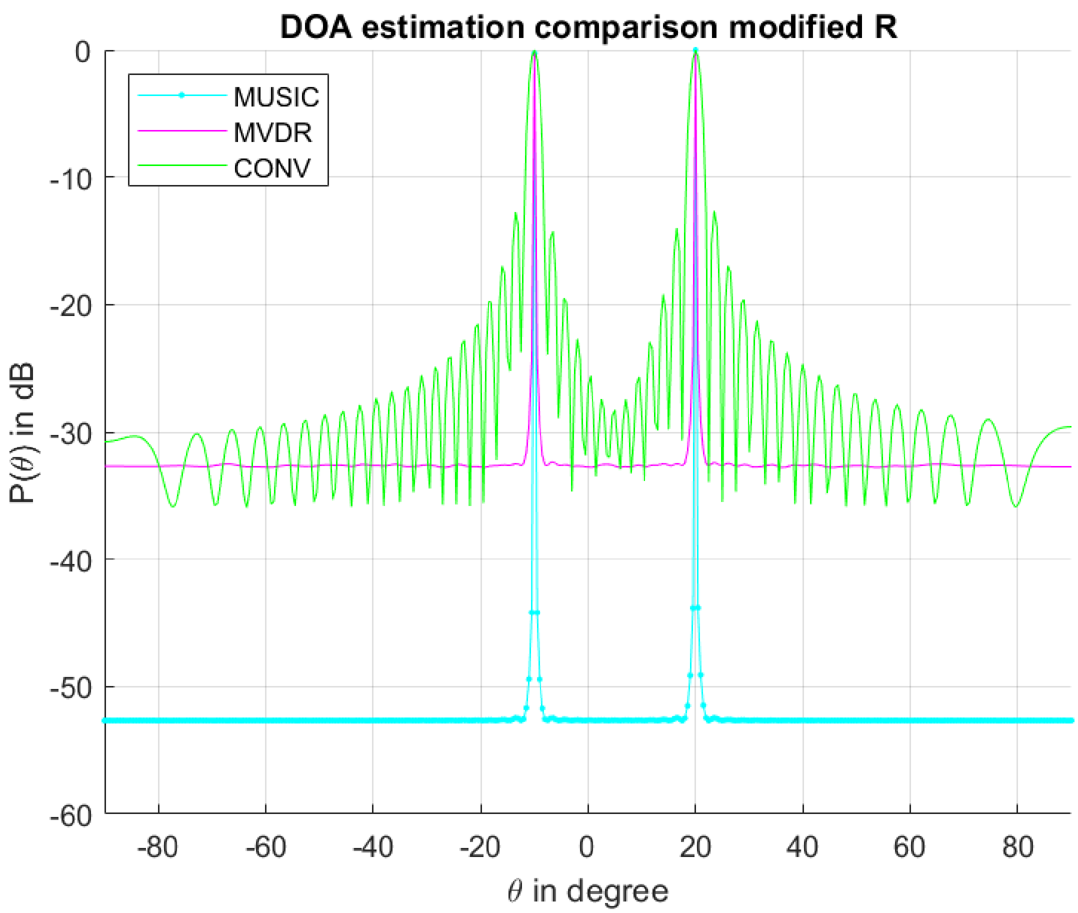
```
%Plotting the results of theta ,P music and P mvdr functions
%In the modified R case
%%Plotting the results of theta and Pmusic function
figure;
hold on
Pmusic = 10*log10(Pmusic/max(Pmusic));
plot(theta,Pmusic,'.-c');
Pmvdr = 10*log10(Pmvdr/max(Pmvdr));
plot(theta,Pmvdr,'-m')
Pconv = 10*log10(Pconv/max(Pconv));
plot(theta,Pconv,'-g');
xlabel('\theta in degree');
legend({'MUSIC','MVDR','CONV'});
ylabel('P({\theta}) in dB');
title('DOA estimation comparison unmodified R');
xlim([-90 90]);
grid on;
```


DOA estimation comparison unmodified R
MUSIC
MVDR
CONV
P(θ) in dB
θ in degree
0
-10
-20
-30
-40
-50
-60
-80
-60
-40
-20
0
20
40
60
80

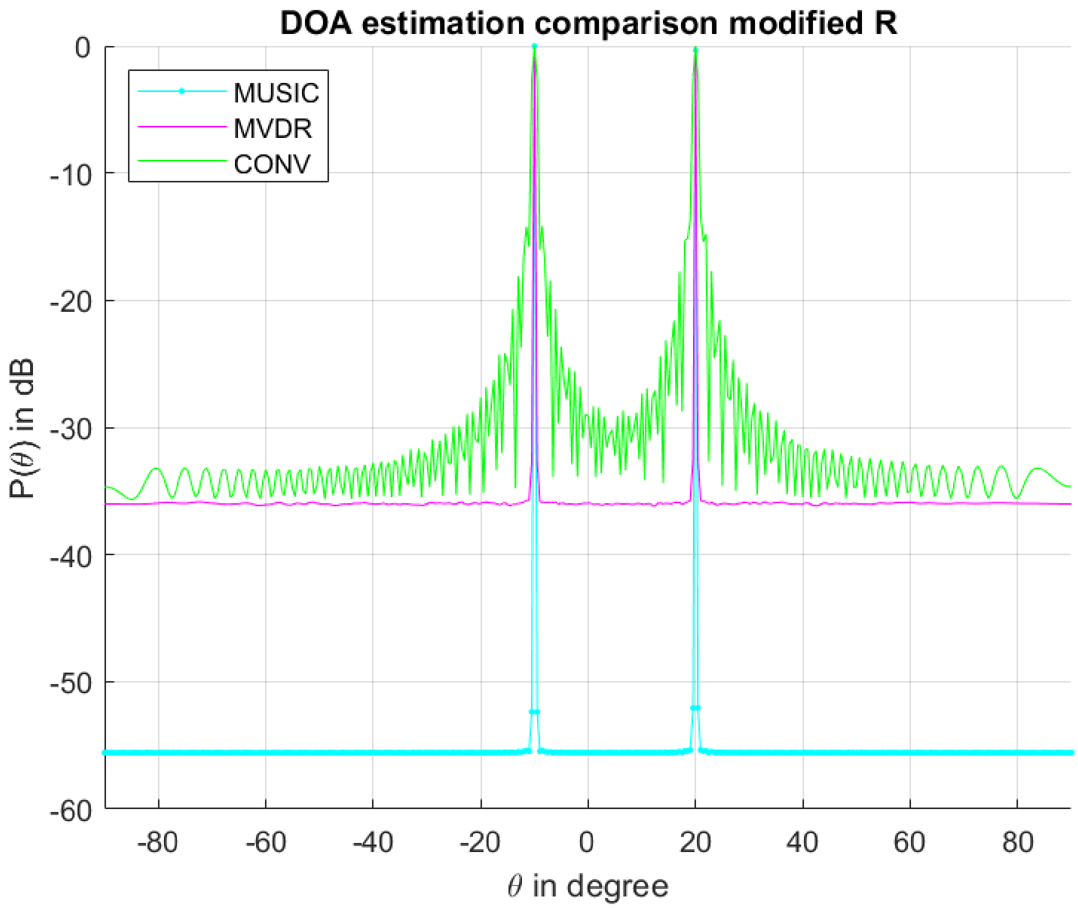
Parameter Variation DOA Estimation

Spacing Variation between Sensors

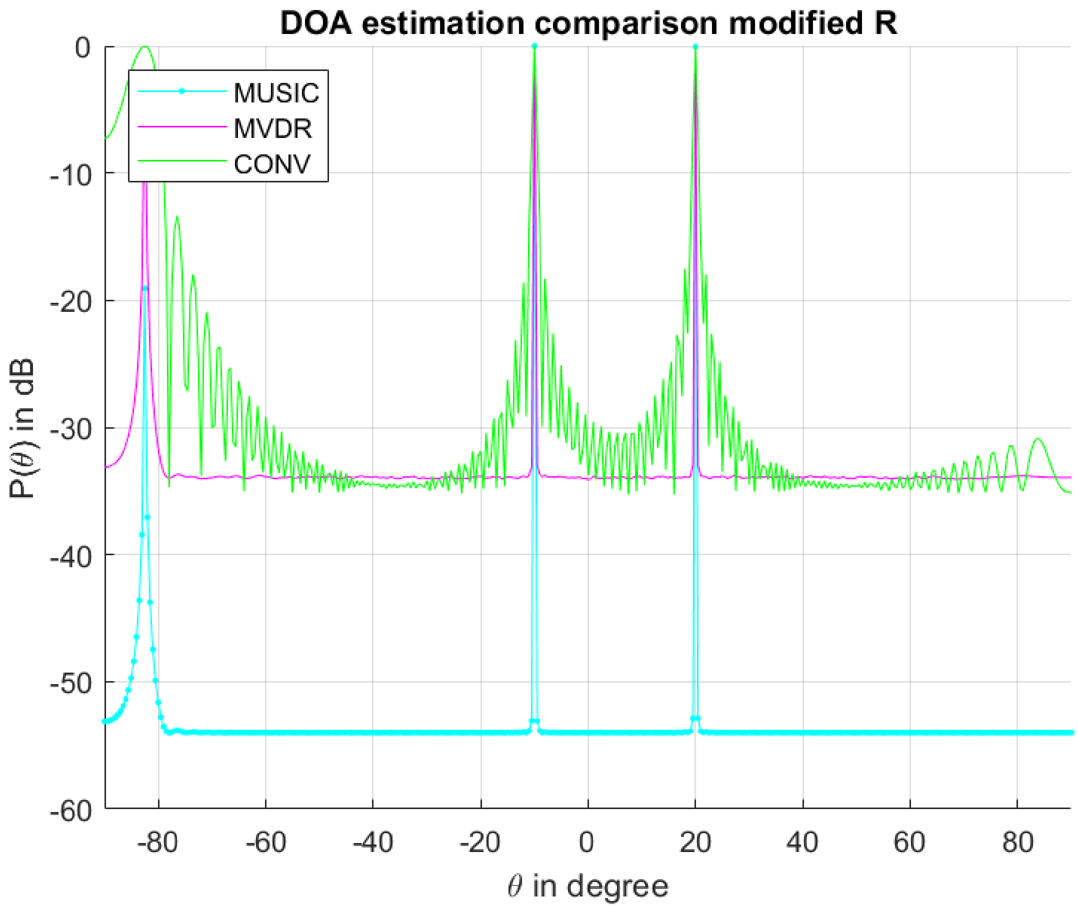
1. Spacing = Lambda/4



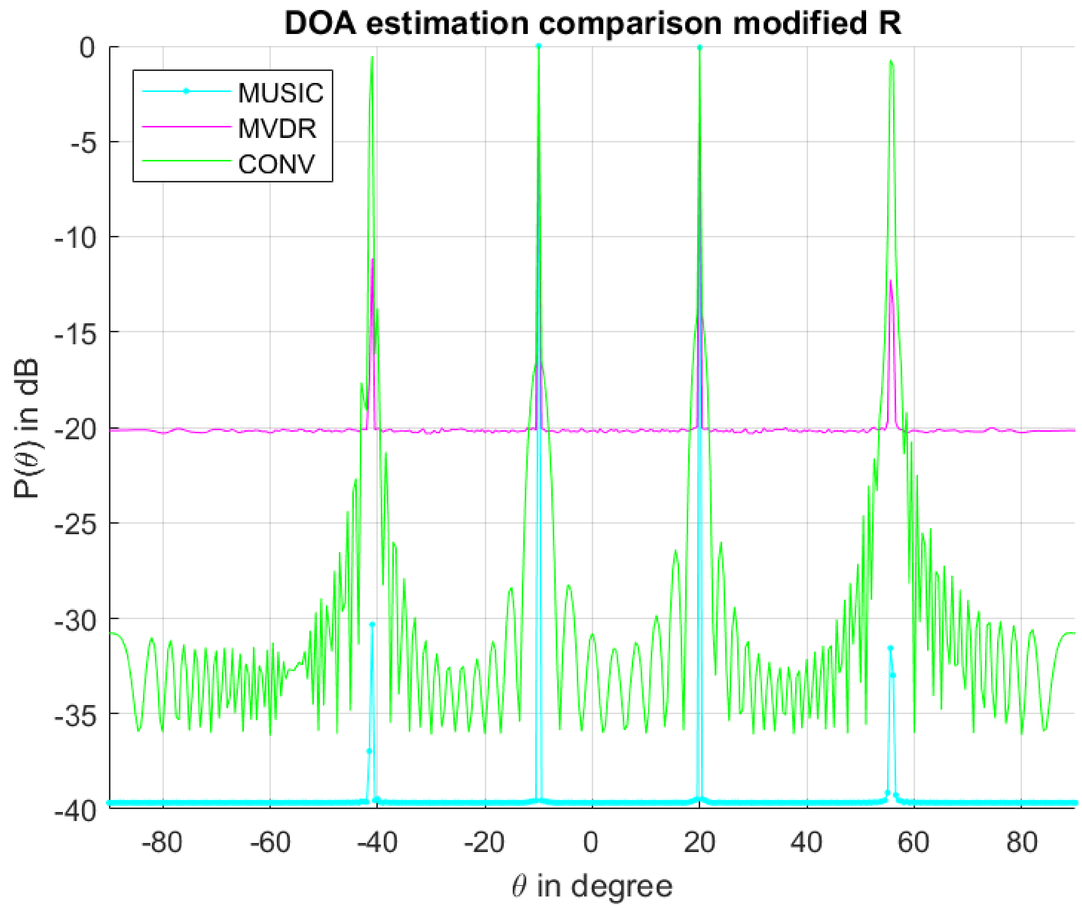
2. Spacing = Lambda/2



3. Spacing = 3*Lambda/4

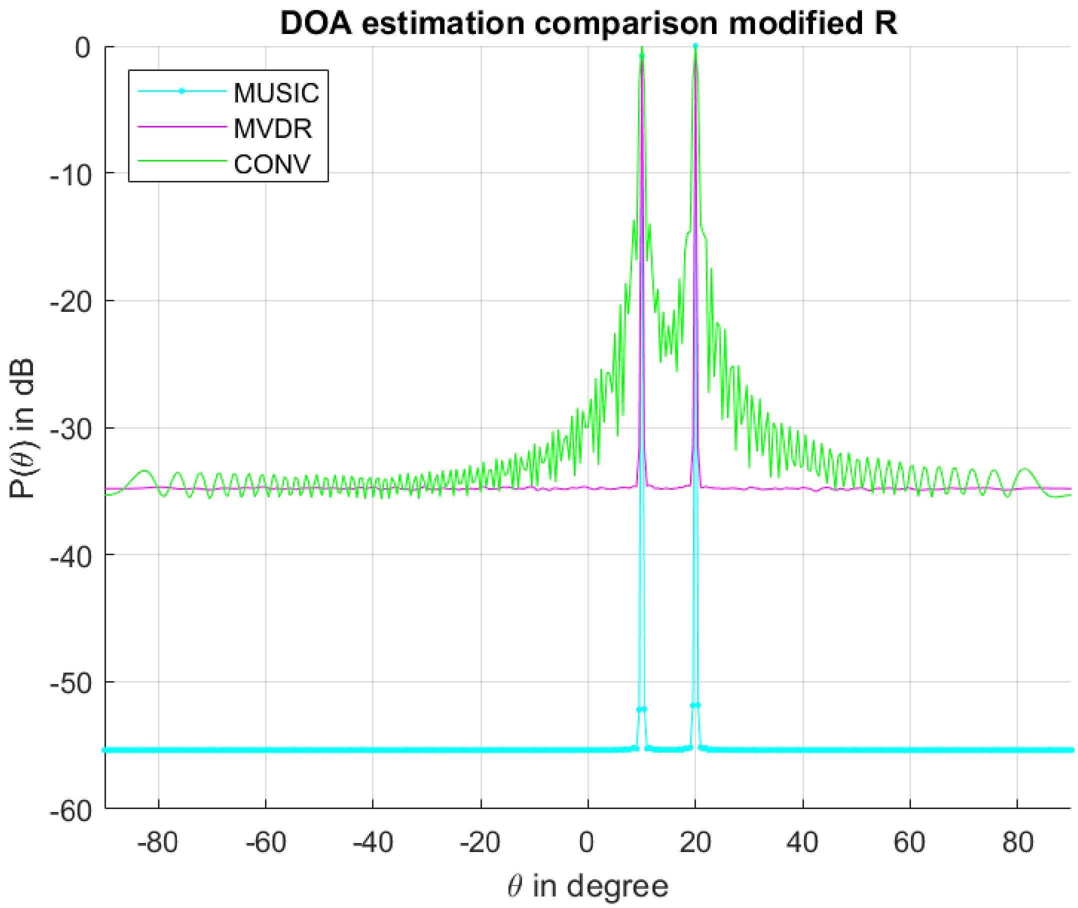


4. Spacing = Lambda

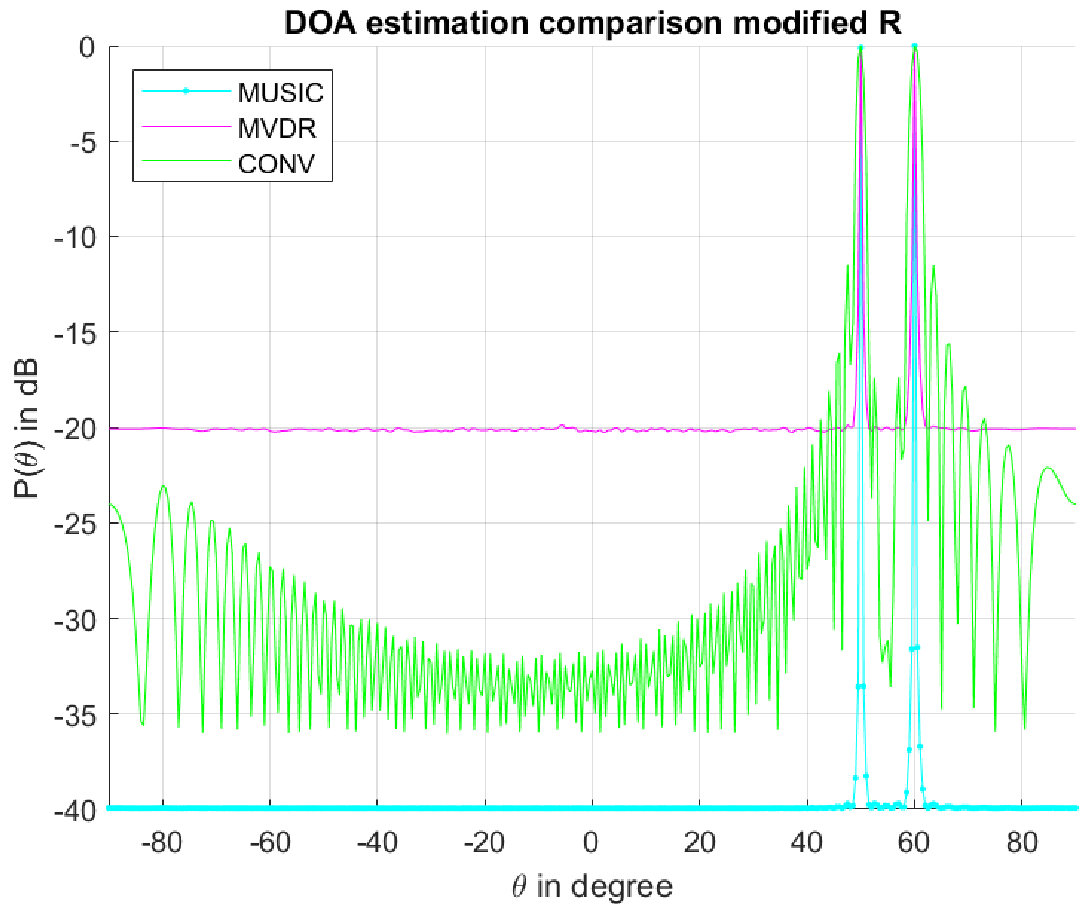


Angle Variation (Degrees)

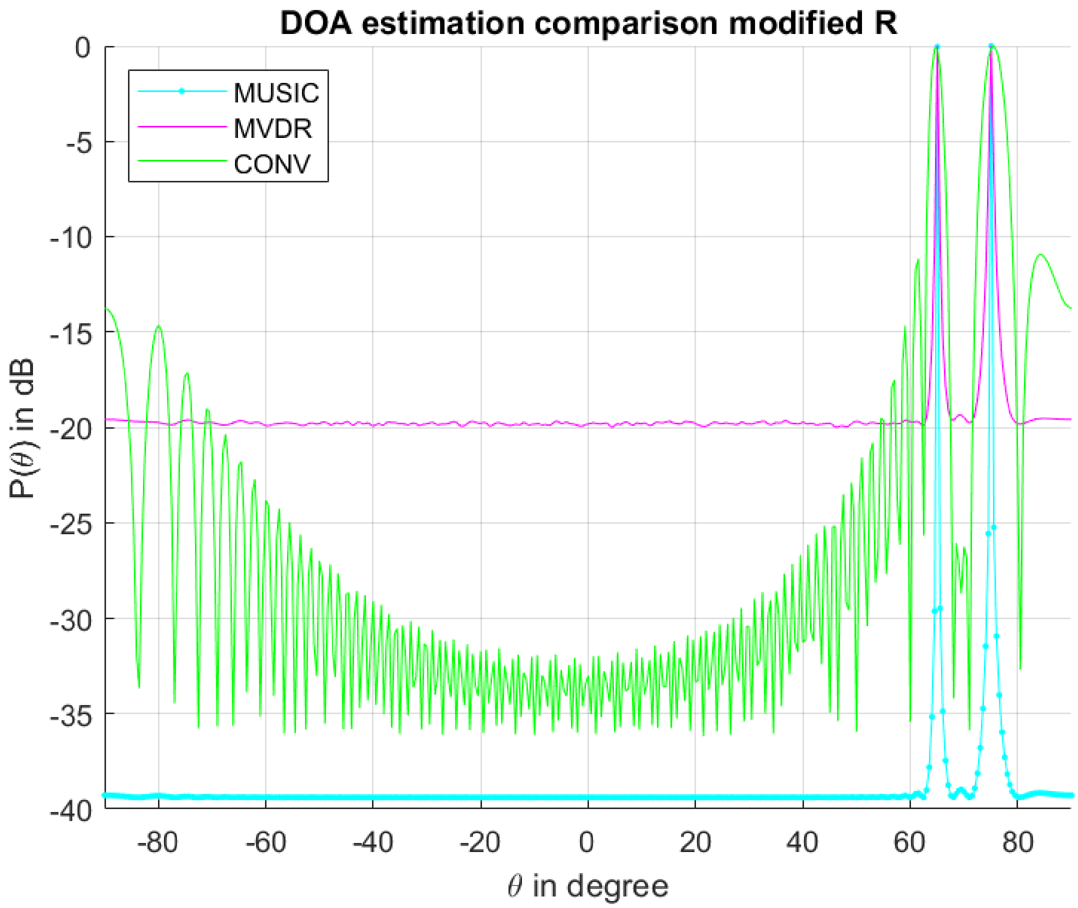
1. Theta1 = 10, Theta2 = 20



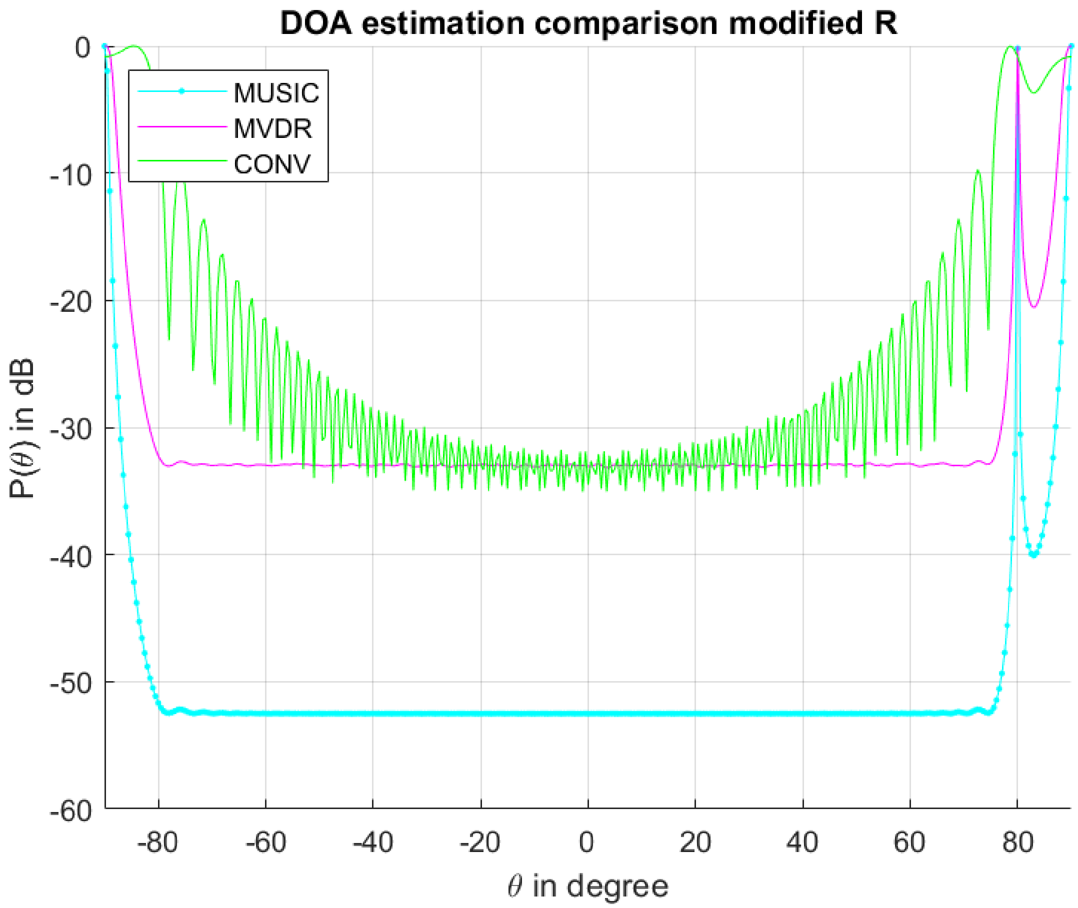
2. Theta1 = 50, Theta2 = 60



3. Theta1 = 65, Theta2 = 75

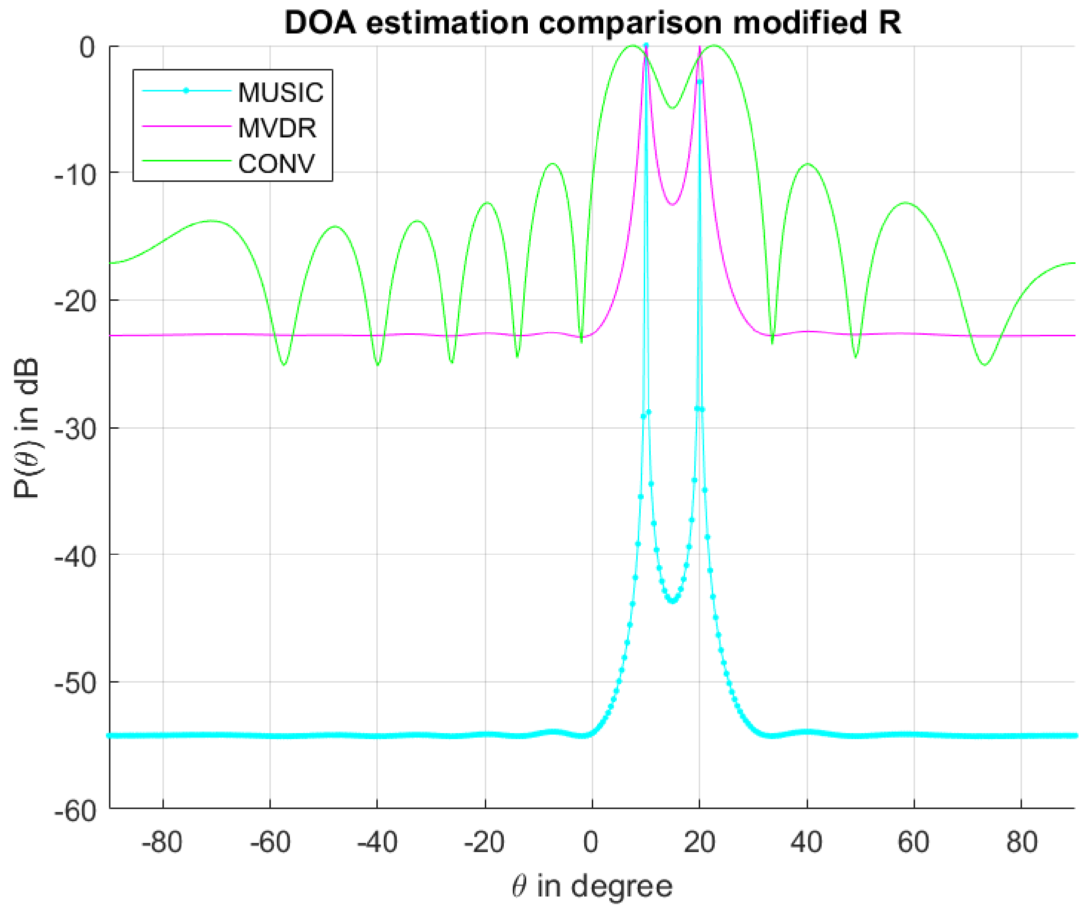


4. Theta1 = 80, Theta2 = 90

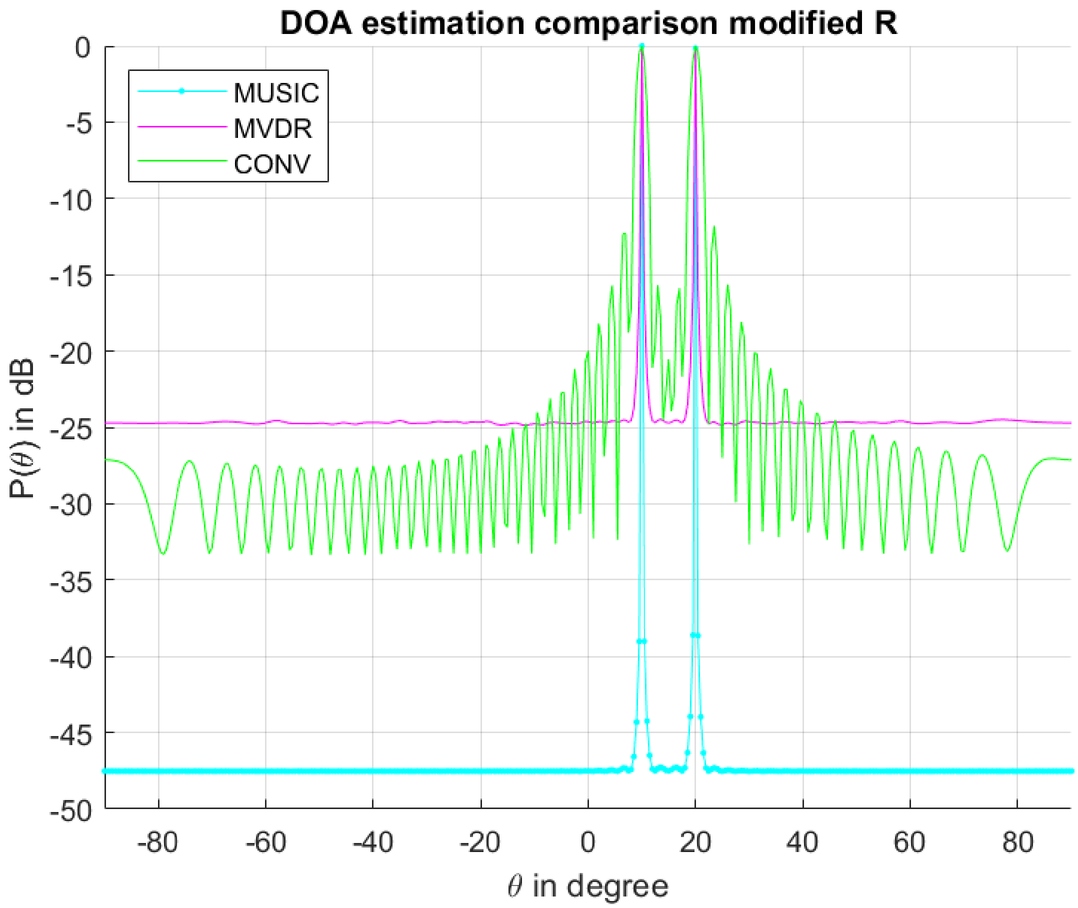


Number of Sensors

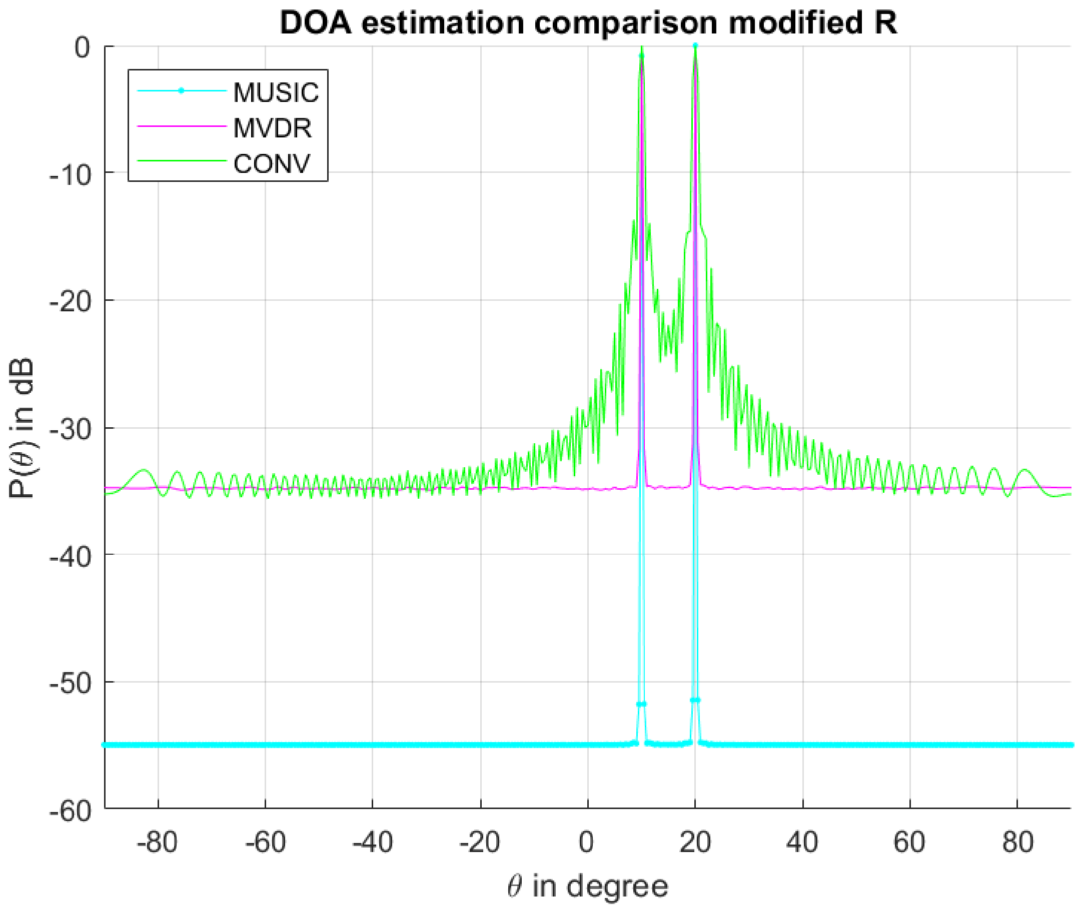
1. 10 Sensors


DOA estimation comparison modified R
MUSIC
MVDR
CONV
0
-10
-20
-30
-40
-50
-60
P(θ) in dB
-80
-60
-40
-20
0
20
40
60
80
θ in degree

2. 50 Sensors

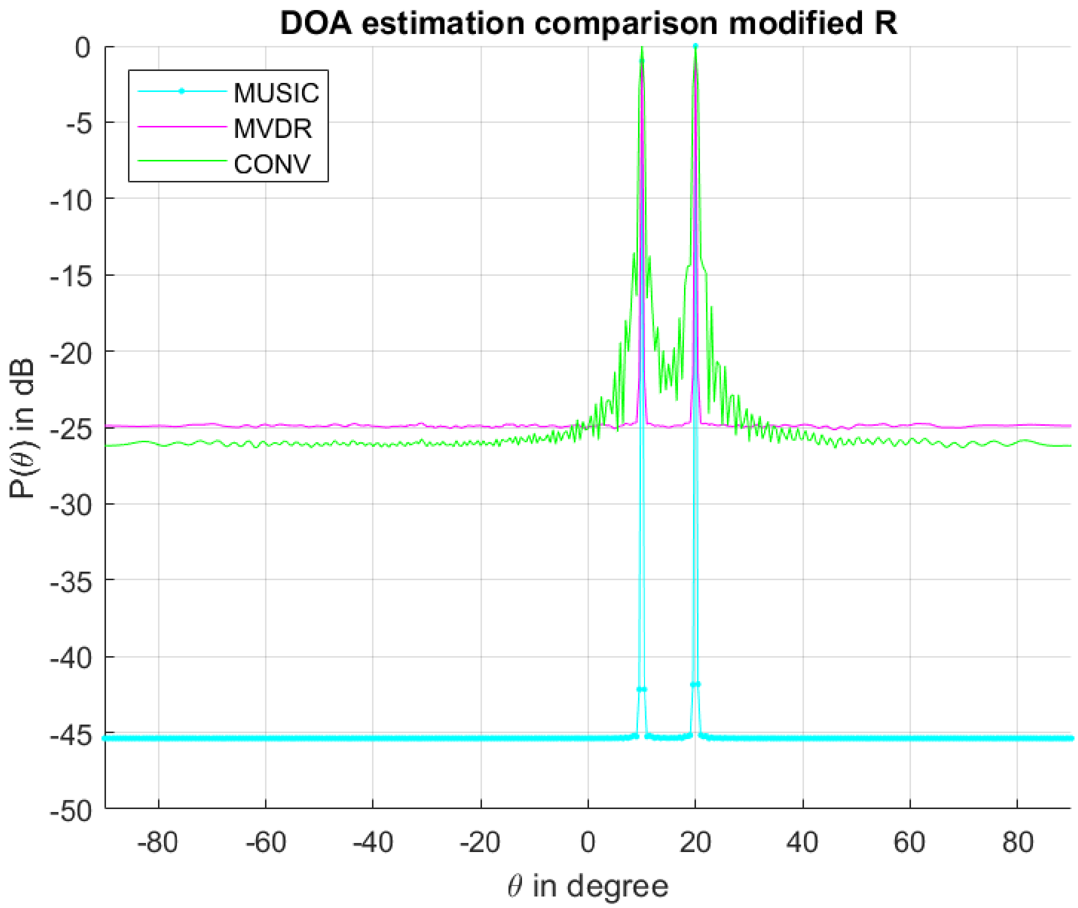


3. 100 Sensors

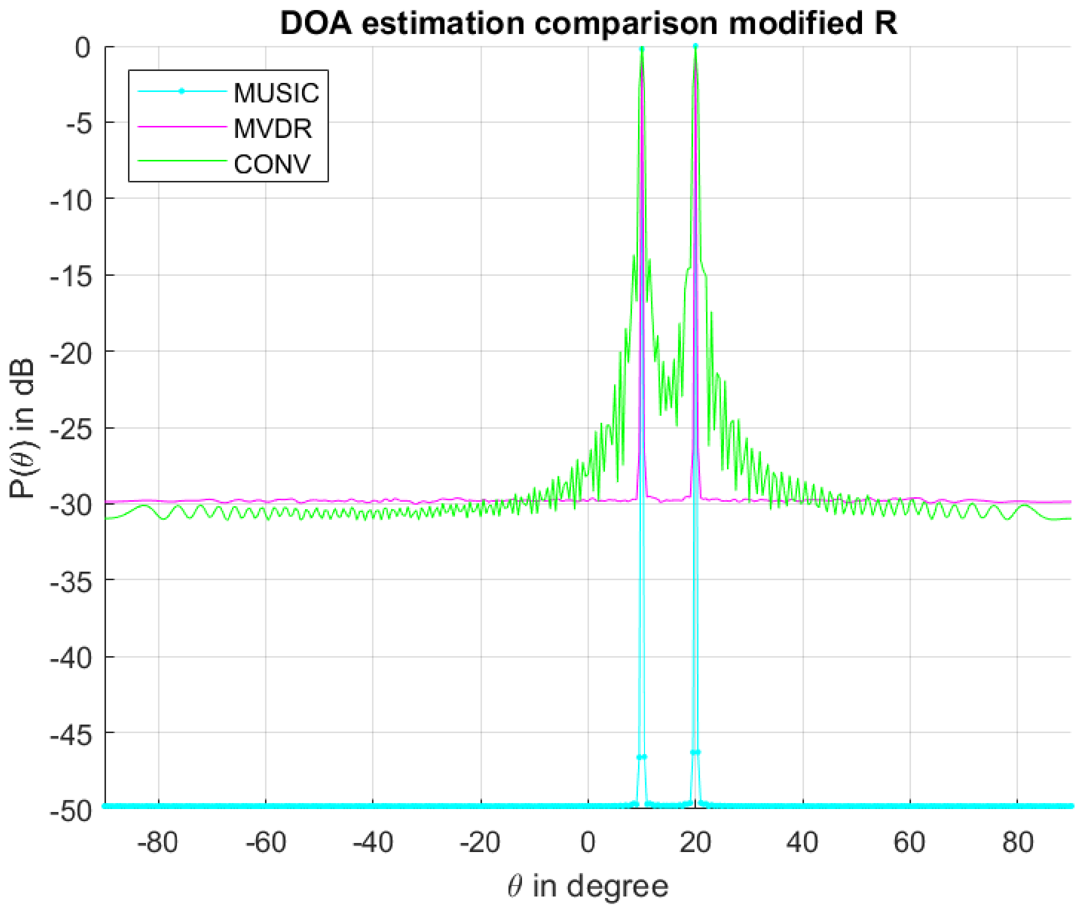


SNR Comparision

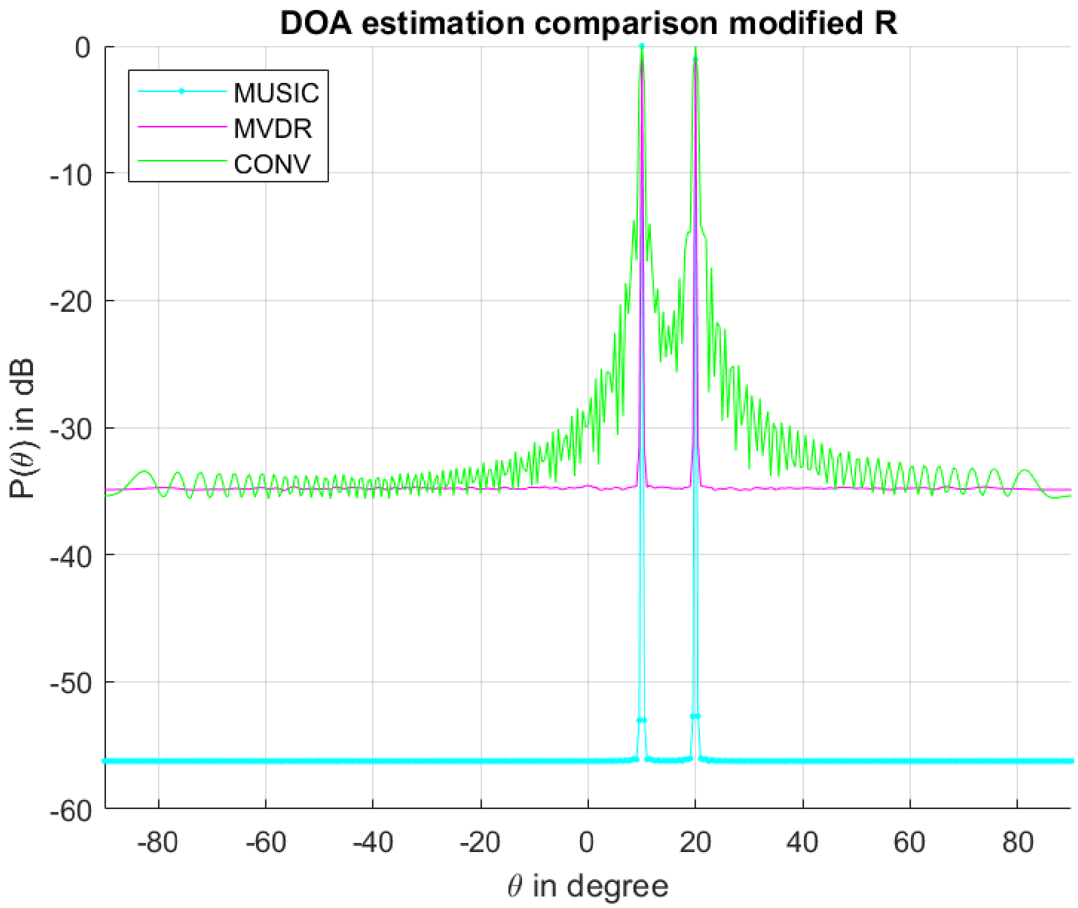
1. SNR=0.1


DOA estimation comparison modified R
MUSIC
MVDR
CONV
P(θ) in dB
θ in degree
0
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
-80
-60
-40
-20
0
20
40
60
80

2. SNR=5


DOA estimation comparison modified R
MUSIC
MVDR
CONV
0
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
P(θ) in dB
-80
-60
-40
-20
0
20
40
60
80
θ in degree

3. SNR=10



Observation with Sensor Spacing

With an increase in the spacing, the number of lobes in the estimate of conventional beamformer increases up to 3/4th of lambda, whereas the MVDR and MUSIC algo the peak sharpens. In lambda spacing, we see additional peaks also come.

Observation with number of sensors

With the number of sensors increasing, we see that the peaks of conventional beamformer become increasingly peaky and it has a lot of combs like peaks whereas the estimates of MVDR and MUSIC becomes sharpened at the direction of arrival

Observation with angle of estimation changing

With an increasing angle, the algorithms detect false peaks by increasing the space between the sensors. Algorithms are accurate with lower values of angles of arrival in all cases. MUSIC and MVDR perform better compared to the conventional beamformer.

Observation with SNR

More SNR gives sharp peaks in both MVDR and MUSIC algorithms. In conventional beamformer in SNR = 5, we have seen that conventional beamformer's estimation lobes become visible more but it becomes minute in SNR 10.