CT509:

Direction of Arrival Estimation Algorithms

November 1 2020

1 Spectral Resolution and Leakage

1.1 Case1:

K=3 and angle of arrivals are [-1° 0° 1°]. In all the following figures,

- PSD indicates FFT approach
- MF indicates Matched Filter approach
- Window indicates FIR filter approach
- EM stands for Expectation Maximization
- ESPRIT is an algebraic algorithm.

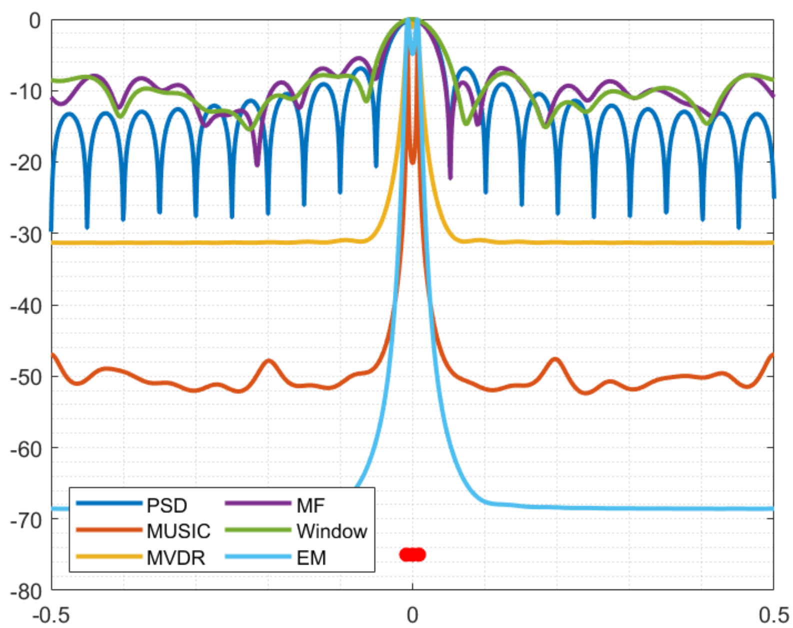


Figure 1: PSD in dB

Algorithm	Estimated AoA
MVDR	[-1 1]
MUSIC	[-1 1]
EM	[-1 1]
ESPRIT	[-1 0 1]

Table 1: Estimated Angle of arrivals for different algorithms

Observations

- FFT,MF,FIR filter methods doesn't perform well
- We can see peaks in MUSIC, MVDR, EM algorithms, but as frequencies are very close [-0.0087 0 0.0087] we were able to see only two peaks.
- MUSIC outperforms other algorithms in terms of resolution.
- Leakage is least in EM, followed by MUSIC and MVDR.
- ESPRIT calculates angle of arrivals accurately. We can generalize that ESPRIT can calculate AoA precisely for other higher angles (and K=3)

1.2 Case2:

K=3 and angles are [-d 0 d] where d $\in$ {2,4,5,6,8,10}

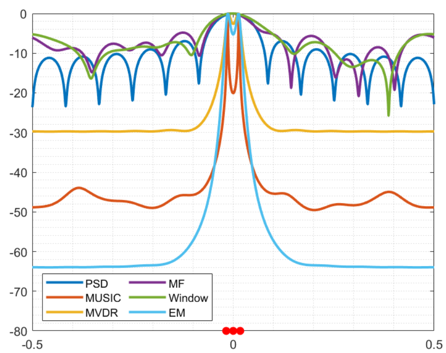


Figure 2: [-2 0 2] (d=2)

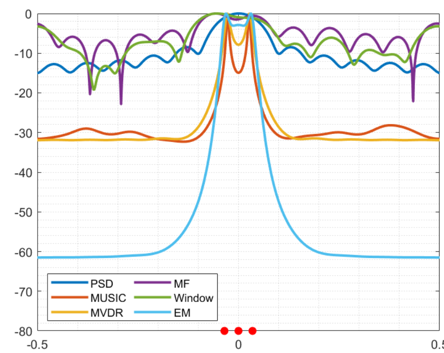


Figure 3: [-4 0 4] (d=4)

- In the above two cases signals are inseparable.
- These are typical examples of smearing

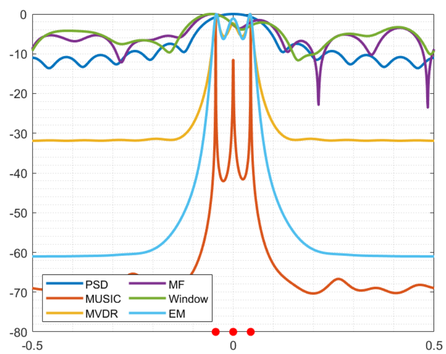


Figure 4: [-5 0 5] (d=5)

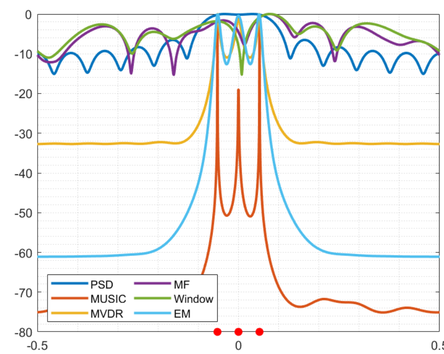


Figure 5: [-6 0 6] (d=6)

- As d is increasing, MUSIC is clearly able to distinguish different users.
- Expectation maximization is starting to produce better results

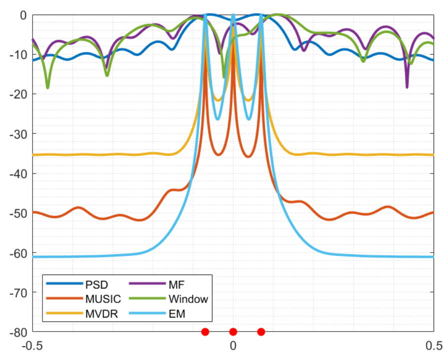


Figure 6: [-8 0 8] (d=8)

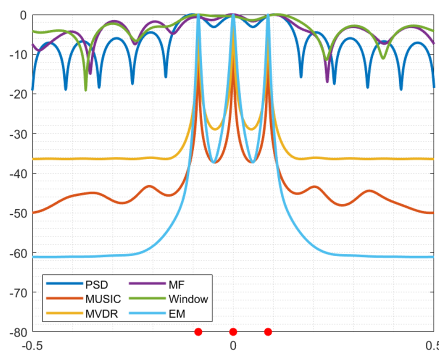
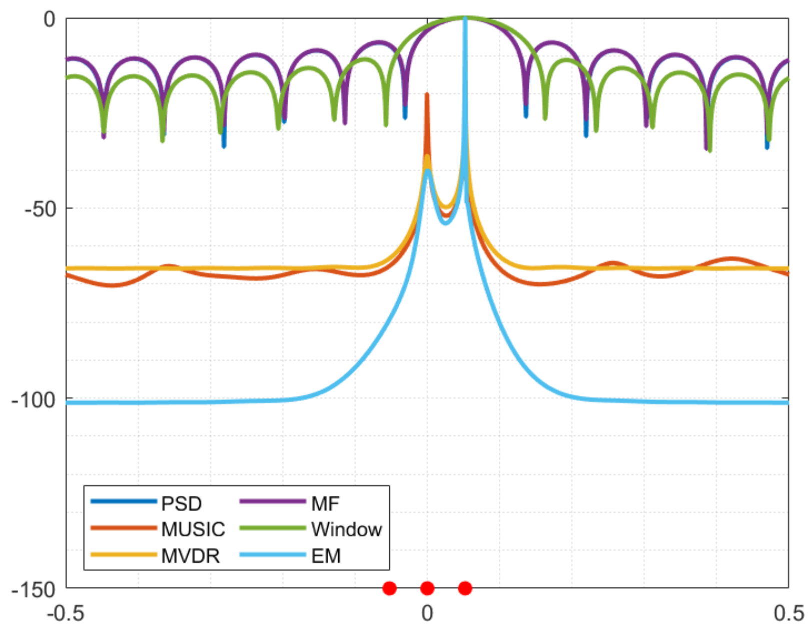


Figure 7: [-10 0 10] (d=10)

- MUSIC,MVDR and EM are totally able to separate the signals.
- FFT based approach is also shaping out but rest are not upto the mark.

1.3 Case3:

In this case, d is chosen as 6 and power levels are set to {-20dB 0dB 20dB}

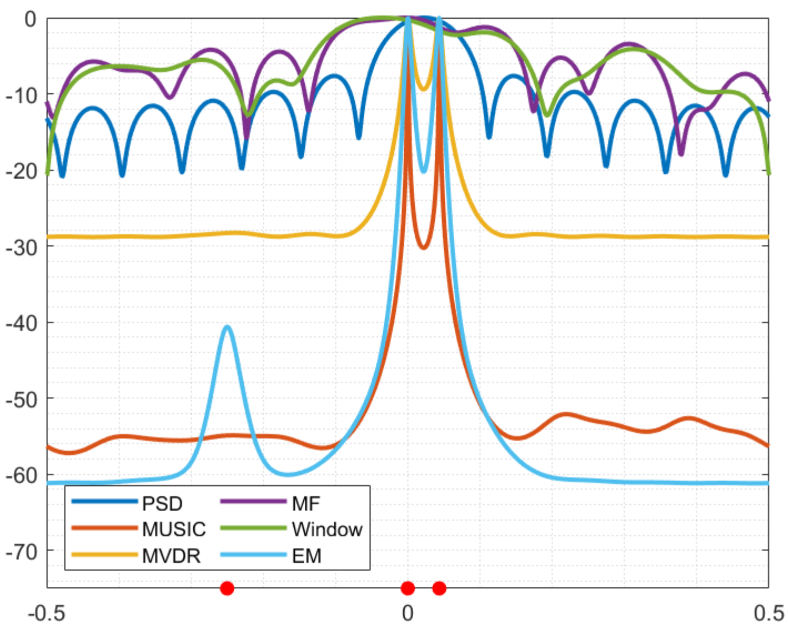


Observations:

- In MUSIC,MVDR and EM, we observe two peaks.No algorithm is able to separate three signals.
- ESPRIT is able to calculate AoA's precisely
- EM,MVDR are sensitive to received power signals

1.4 Case4:

K=3, angles [-30 0 5] (in degrees), power levels are set to {-20dB 0dB 0dB}



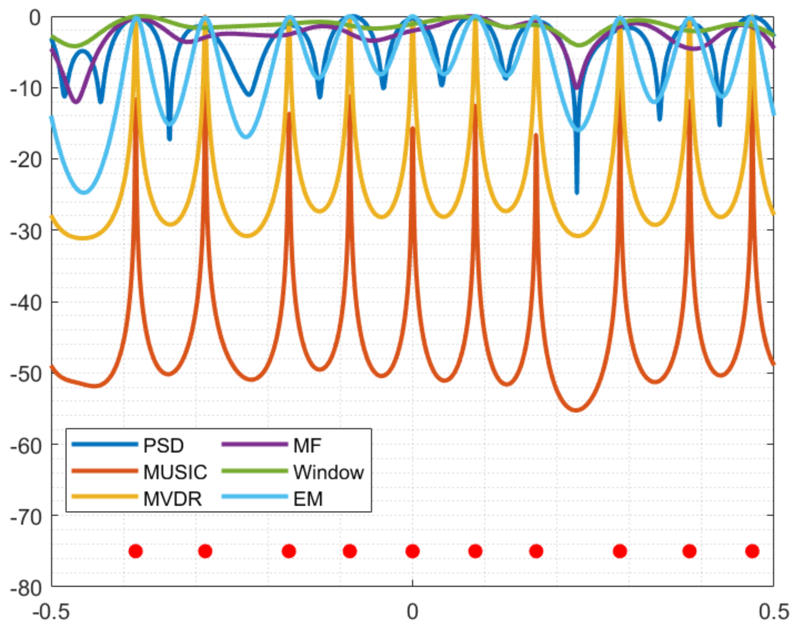
Observations:

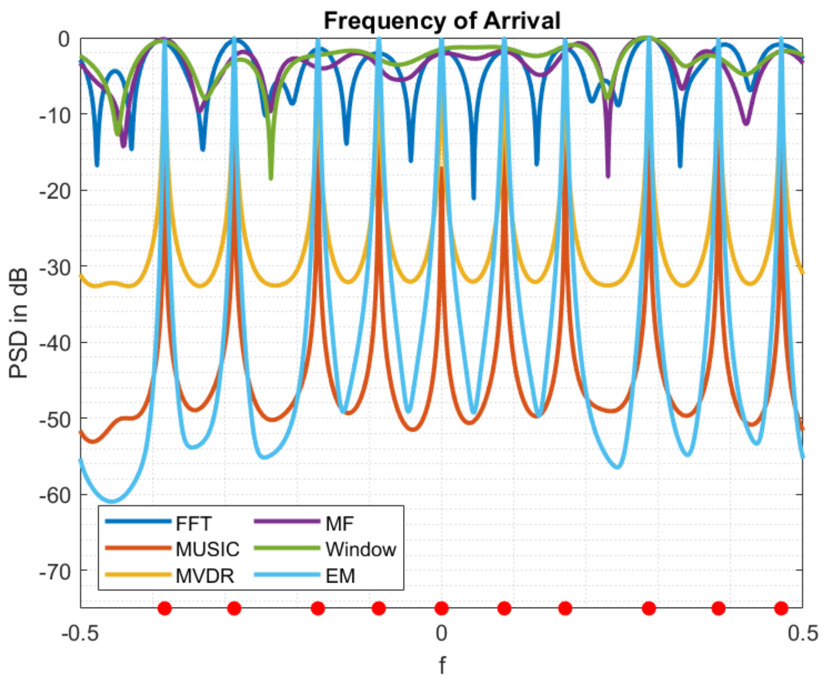
- Again, Similar comments can be made on MUSIC, MVDR, EM.
- However, EM is showing a faint presence of third user, other two were able to distinguish only two users.

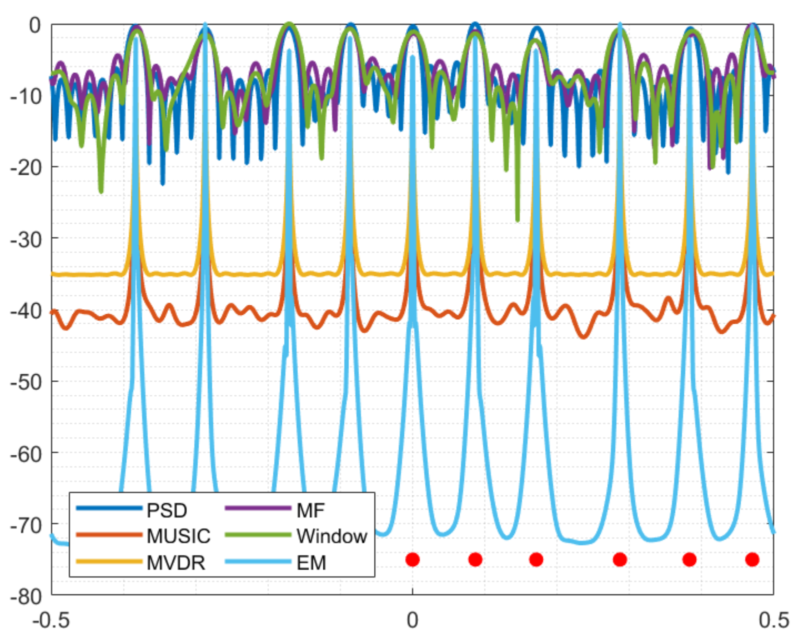
2 A Performance Study:

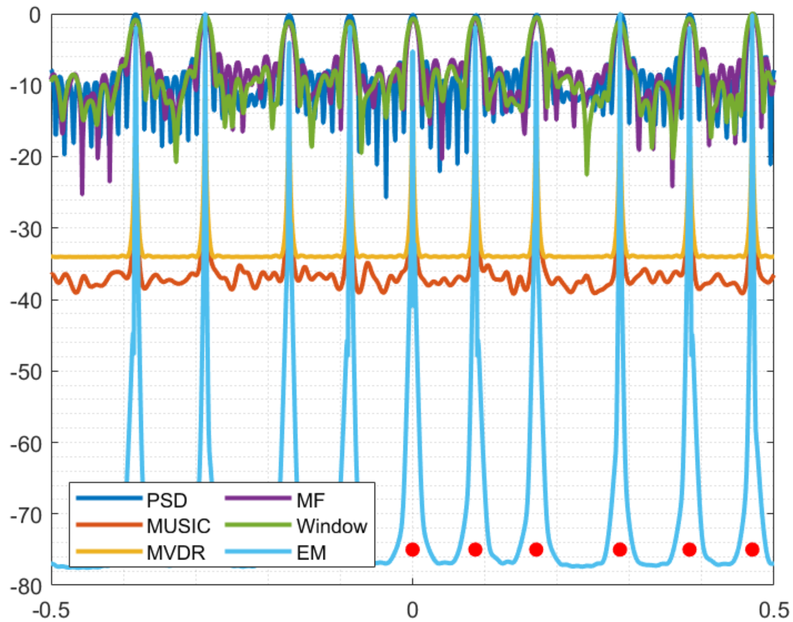
2.1 Performance of all algorithms with increase in number of elements

Number of users are set to 10, DoAs are the following angles and N is varied as [15 20 50 80] respectively
$\theta° = \{-50°, -35°, -20°, -10°, 0°, 10°, 20°, 35°, 50°, 70°\}$
And signals are received at power level of 1W




Frequency of Arrival
PSD in dB
0
-10
-20
-30
-40
-50
-60
-70
-0.5
0
0.5
f
FFT
MUSIC
MVDR
MF
Window
EM


0
-10
-20
-30
-40
-50
-60
-70
-80
-0.5
0
0.5
PSD
MUSIC
MVDR
MF
Window
EM



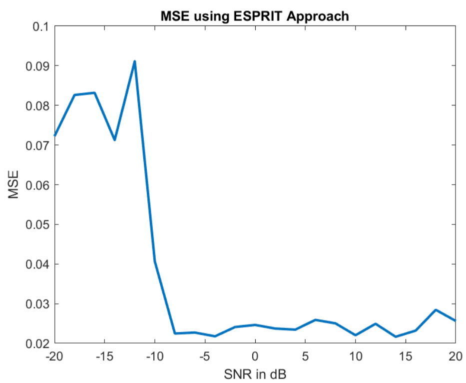
Observations

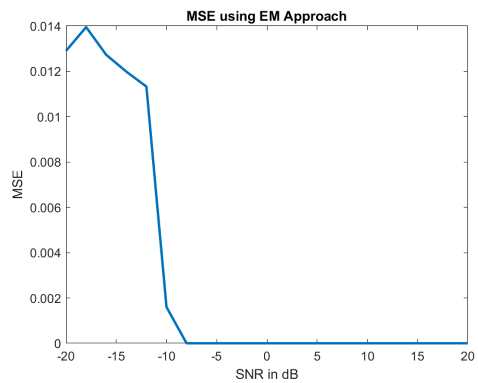
- With increase in number of Elements(N) in ULA, We observe,
  - Peaks become more prominent, i.e. better resolution
  - Spectral leakage is reduced. i.e The power leaked in the side-lobes is reduced.
- Expectation maximization algorithm gives accurate results with increase in number of elements
- However, with increase in number of elements, receivers become expensive.
- MUSIC algorithm gives expected results with lesser number of elements.

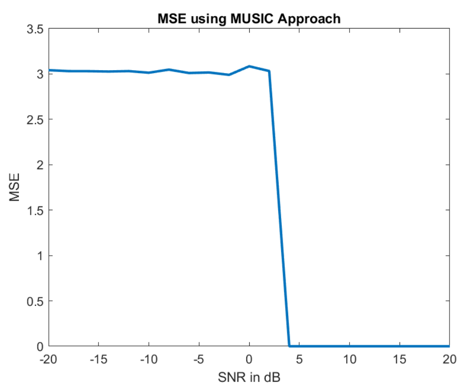
2.2 Monte-Carlo Simulations

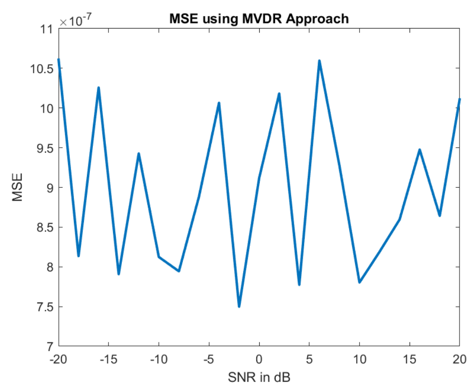
- Number of sources be 5, K=5
- SNR is varied from -20 to 20 (in the steps of 2)
- Number of MonteCarlo Simulations is considered to be 50.

case1 let the separation between any two closest sources be 20°.

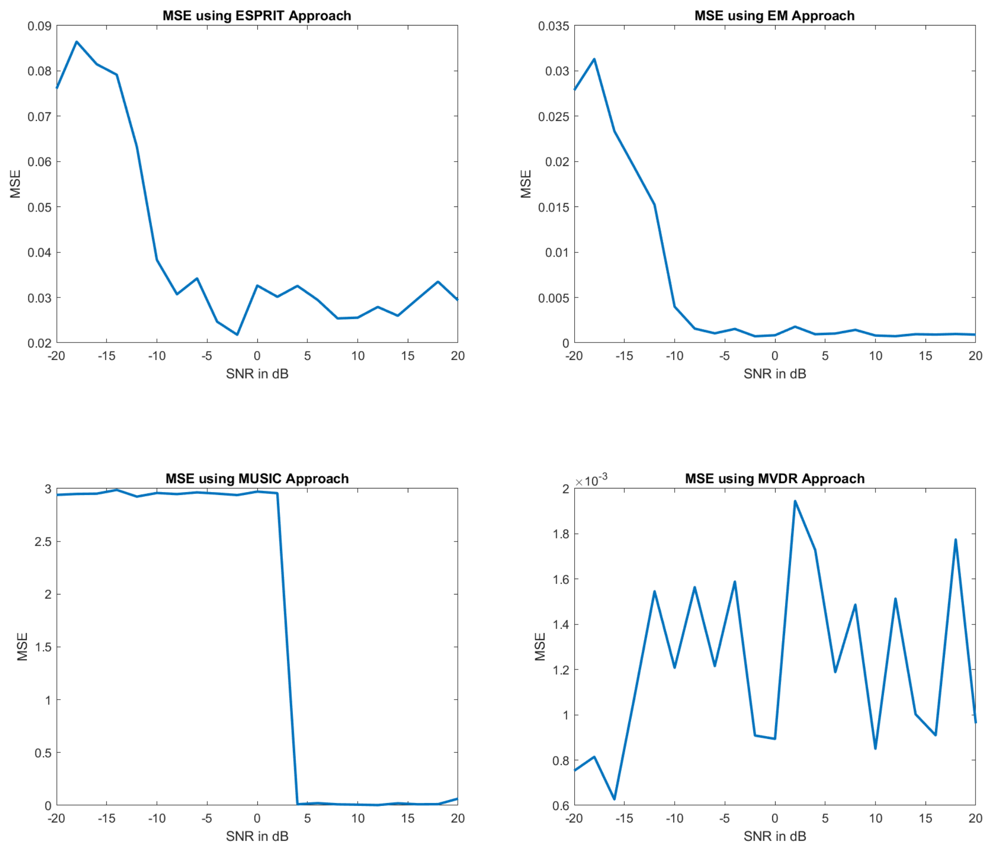








case2 let the separation between any two closest sources be 15°.



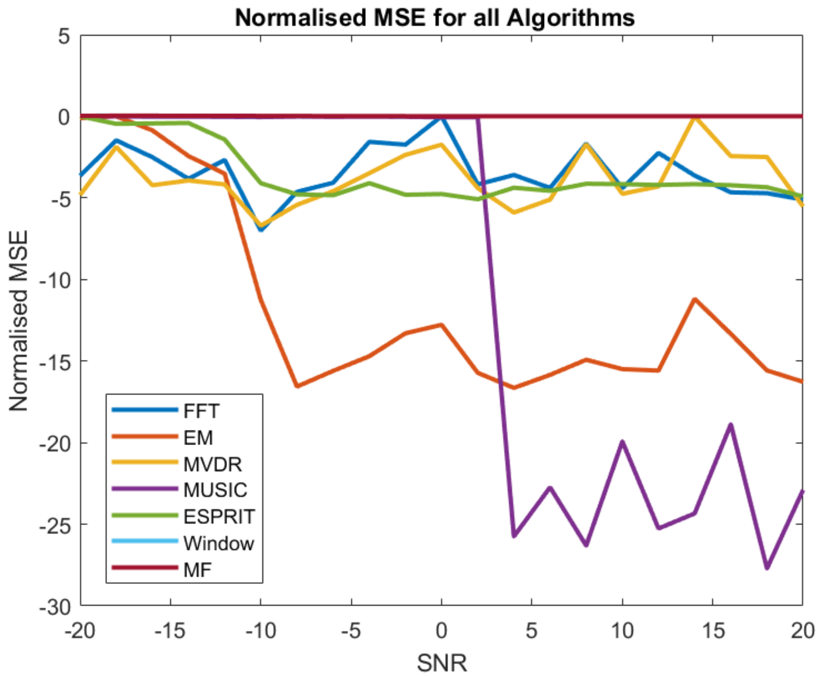


Figure 8: MSE vs SNR for all algorithms when $\Delta = 20°$

Calculations

- MSE is calculated as follows

$$MSE = \Sigma_{k=1}^{K}(\theta_{estimated}^{k} - \theta_{actual}^{k})^2 \tag{1}$$

- Normalised MSE

$$NormalisedMSE = 10log_{10}(MSE/max(MSE)) \tag{2}$$

3 Inferences

- MVDR algorithm has relatively better resolution, compared to MF,FFT and Window algorithms.
- MVDR and EM algorithms depend on the power of the received signals, while MUSIC and ESPRIT are less dependent.
- All algorithms are dependent on the number of signals processed
- Both MUSIC and ESPRIT produce better results in detecting signals that are close. i.e. when the DoAs are close to each other.
- These two algorithms have superior resolution and accuracy.
- In case of Low SNR, performance of MUSIC deteriorates.

4 Appendix

4.1 CODE

```
clearvars;close all;clc;
N=20;n=0:N-1;K=5;
a_k = ones(K,1);mse = zeros(1,21);
M=1024;t=0;
snr = -20:2:20;
mse_psd=zeros(1,length(snr));
mse_PEM=zeros(1,length(snr))
;mse_mvdr=zeros(1,length(snr));
mse_music=zeros(1,length(snr));
mse_esprit=zeros(1,length(snr));
for ii =1:length(snr)
for nsim = 1:50
SNR_dB=snr(ii);gamma=10^(SNR_dB/10);sigma=1/sqrt(gamma);
theta1 = 15*rand()-60;
theta2 = 15*rand()-30;
theta3 = 15*rand();
theta4 = -15*rand()+30;
theta5 = -15*rand()+60;
thetass =[theta1 theta2 theta3 theta4 theta5];
theta=thetass.*(pi/180);
f=sin(theta)/2;
phi_k=2*pi*rand(K,1)-pi;f=f(:);
xn_k = a_k.*exp(1j*2*pi*theta').*exp(1j*2*pi*f*(0:N-1));
%% PSD
for kk=1:N
   xnk=xn_k(:,kk);
   R=xnk*xnk';
   t=t+R;
   f_k=f.';
   W=exp(-1j*2*pi*n'*f_k);
   u = sigma*randn(N,1)+1j*randn(N,1);
   xn = W*xnk+u;
end
```

```
R_ss=t/N;
Rss = diag(a_k.^2);Rxx=W*R_ss*W'+sigma^2*eye(N);
fo = linspace(-0.5,0.5,M);
r = fftshift(fft(Rxx(1,:),M));
r1 = abs(r);r1 = 10*log10(r1/max(r1));

a1 = [-70;-50];a2 = [-40;-20];a3 = [-10;10];a4 = [20;40];a5 =
    [50;70];
f_one = sin([a1 a2 a3 a4 a5]*pi/180)/2;
aa1 = find(abs(fo-f_one(1,1))<0.0005);ab1 =
    find(abs(fo-f_one(2,1))<0.0005);
aa2 = find(abs(fo-f_one(1,2))<0.0005);ab2 =
    find(abs(fo-f_one(2,2))<0.0005);
aa3 = find(abs(fo-f_one(1,3))<0.0005);ab3 =
    find(abs(fo-f_one(2,3))<0.0005);
aa4 = find(abs(fo-f_one(1,4))<0.0005);ab4 =
    find(abs(fo-f_one(2,4))<0.0005);
aa5 = find(abs(fo-f_one(1,5))<0.0005);ab5 =
    find(abs(fo-f_one(2,5))<0.0005);
for oo = 1:length(fo)
if oo>=aa1 && oo<ab1
   r11(oo) = r1(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f1_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa2 && oo<ab2
   r11(oo) = r1(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f2_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa3 && oo<ab3
   r11(oo) = r1(oo);
```

```
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f3_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa4 && oo<ab4
   r11(oo) = r1(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f4_est = fo(nn);

for oo = 1:length(fo)
if oo>=aa5 && oo<ab5
   r11(oo) = r1(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f5_est = fo(nn);
f_estimated = [f1_est f2_est f3_est f4_est f5_est];
aoa = asin(2*f_estimated)*180/pi;
theta11 = sort(thetass)';
aoa11 = sort(aoa);
mmse_psd(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
%% MVDR
W1 = exp(-1j*2*pi*n'*fo);
for i = 1:M
P_MVDR(i) = 1/(W1(:,i)'*inv(Rxx)*W1(:,i));
end
Pmvdr=10*log10(abs(P_MVDR)/max(abs(P_MVDR))); %Spatial spectrum
    function
for oo = 1:length(fo)
if oo>=aa1 && oo<ab1
   r11(oo) = Pmvdr(oo);
else
```

```
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f1_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa2 && oo<ab2
   r11(oo) = Pmvdr(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f2_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa3 && oo<ab3
   r11(oo) = Pmvdr(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f3_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa4 && oo<ab4
   r11(oo) = Pmvdr(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f4_est = fo(nn);

for oo = 1:length(fo)
if oo>=aa5 && oo<ab5
   r11(oo) = Pmvdr(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f5_est = fo(nn);
f_estimated = [f1_est f2_est f3_est f4_est f5_est];
```

```
aoa = asin(2*f_estimated);
theta11 = sort(theta)';
aoa11 = sort(aoa);
mmse_mvdr(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);

%% MUSIC
[Q,D]=eig(Rxx);
k=sum(sum(round(D)~=0));
K1 = N-k;
V = Q(:,[k+1:end]);
W1 = exp(-1j*2*pi*n'*fo);
for i = 1:length(fo)
PsePSD(i) = 1/((W1(:,i)'*(V*V')*W1(:,i)));
end
Pmusic=10*log10(abs(PsePSD)/max(abs(PsePSD))); %Spatial spectrum function
for oo = 1:length(fo)
if oo>=aa1 && oo<ab1
   r11(oo) = Pmusic(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f1_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa2 && oo<ab2
   r11(oo) = Pmusic(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f2_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa3 && oo<ab3
   r11(oo) = Pmusic(oo);
else
   r11(oo) = -80;
end
```

```
end
[mm,nn]=max(r11);f3_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa4 && oo<ab4
   r11(oo) = Pmusic(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f4_est = fo(nn);

for oo = 1:length(fo)
if oo>=aa5 && oo<ab5
   r11(oo) = Pmusic(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f5_est = fo(nn);
f_estimated = [f1_est f2_est f3_est f4_est f5_est];
aoa = asin(2*f_estimated);
theta11 = sort(theta)';
aoa11 = sort(aoa);
mmse_music(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
%% EM
W1 = exp(-1j*2*pi*n'*fo);%Reference filter bank
%Random initialization of Rzz
rnf = rand(M,1);
Rzz = diag(rnf.^2);
Xn = W*xn_k+sigma^2*eye(N);
% Rzz = eye(M);
for i = 1:20
   Rxx1 = W1*Rzz*W1';
   Rxx1 = Rxx1/N;
   Rxx_inv = inv(Rxx1);
   for mm = 1:M
      w_m =W1(:,mm);
      w_mvdr = Rxx_inv*w_m/(w_m'*Rxx_inv*w_m);
```

```
      y_em = w_mvdr'*Xn;
      %       for aa = 1:12
%          y_em(:,aa) = w_mvdr'*Xn(:,aa);
%       end
      Rzz(mm,mm) = y_em*y_em';
   end
end
P_em = abs(diag(Rzz));
P_EM=10*log10((P_em)/max((P_em)));

for oo = 1:length(fo)
if oo>=aa1 && oo<ab1
   r11(oo) = P_EM(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f1_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa2 && oo<ab2
   r11(oo) = P_EM(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f2_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa3 && oo<ab3
   r11(oo) = P_EM(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f3_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa4 && oo<ab4
   r11(oo) = P_EM(oo);
else
```

```
    r11(oo) = -80;
end
end
[mm,nn]=max(r11);f4_est = fo(nn);

for oo = 1:length(fo)
if oo>=aa5 && oo<ab5
    r11(oo) = P_EM(oo);
else
    r11(oo) = -80;
end
end
[mm,nn]=max(r11);f5_est = fo(nn);
f_estimated = [f1_est f2_est f3_est f4_est f5_est];
aoa = asin(2*f_estimated);
theta11 = sort(theta)';
aoa11 = sort(aoa);
mmse_PEM(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
%% ESPRIT
[U,S,V] = svd(Xn);
MM = diag(S);
[rr,cc]=find(abs(MM)-sigma^2<0.00001);
kapa = sum(cc);
KK = N-kapa;
U_est = U(:,[1:K]);val = 1;
J_x = [eye(N-val) zeros(N-val,1)];
J_y = [zeros(N-val,1) eye(N-val)];

Ux = J_x*U_est;
Uy = J_y*U_est;
Z = pinv(Ux)*Uy;
[T,thetas] = eig(Z);
ang = asin(angle(thetas)/pi)*180/pi;
ang = diag(ang)*pi/180;
W_1 =T*U_est';
theta11 = sort(thetass)'*pi/180;
aoa11 = sort(ang);
mmse_esprit(nsim,ii) = (1/K)*sum((theta11-aoa11).^2);
```

```
%% MF
y = xn'*W1;
y_mf=10*log10(abs(y)/max(abs(y)));
for oo = 1:length(fo)
if oo>=aa1 && oo<ab1
   r11(oo) = y_mf(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f1_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa2 && oo<ab2
   r11(oo) = y_mf(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f2_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa3 && oo<ab3
   r11(oo) = y_mf(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f3_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa4 && oo<ab4
   r11(oo) = y_mf(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f4_est = fo(nn);

for oo = 1:length(fo)
if oo>=aa5 && oo<ab5
```

```
    r11(oo) = y_mf(oo);
else
    r11(oo) = -80;
end
end
[mm,nn]=max(r11);f5_est = fo(nn);
f_estimated = [f1_est f2_est f3_est f4_est f5_est];
aoa = asin(2*f_estimated);
theta11 = sort(theta)';
aoa11 = sort(aoa);
mmse_MF(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);

%% Window
wc = kaiser(N,2.4);
xn = xn.*wc;
y = xn'*W1;
y_win=10*log10(abs(y)/max(abs(y)));
for oo = 1:length(fo)
if oo>=aa1 && oo<ab1
    r11(oo) = y_win(oo);
else
    r11(oo) = -80;
end
end
[mm,nn]=max(r11);f1_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa2 && oo<ab2
    r11(oo) = y_win(oo);
else
    r11(oo) = -80;
end
end
[mm,nn]=max(r11);f2_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa3 && oo<ab3
    r11(oo) = y_win(oo);
else
    r11(oo) = -80;
```

```
end
end
[mm,nn]=max(r11);f3_est = fo(nn);
for oo = 1:length(fo)
if oo>=aa4 && oo<ab4
   r11(oo) = y_win(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f4_est = fo(nn);

for oo = 1:length(fo)
if oo>=aa5 && oo<ab5
   r11(oo) = y_win(oo);
else
   r11(oo) = -80;
end
end
[mm,nn]=max(r11);f5_est = fo(nn);
f_estimated = [f1_est f2_est f3_est f4_est f5_est];
aoa = asin(2*f_estimated);
theta11 = sort(theta)';
aoa11 = sort(aoa);
mmse_Window(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);

end
end
%%
% mmse = mmse/mnop;
mmse_psd = sum(mmse_psd)/nsim;
mmse_PEM = sum(mmse_PEM)/nsim;
mmse_mvdr = sum(mmse_mvdr)/nsim;
mmse_music = sum(mmse_music)/nsim;
mmse_esprit = sum(mmse_esprit)/nsim;
mmse_Window = sum(mmse_Window)/nsim;
mmse_MF = sum(mmse_MF)/nsim;
```

```
mmse_psd = 10*log10(mmse_psd/max(mmse_psd));
mmse_PEM = 10*log10(mmse_PEM/max(mmse_PEM));
mmse_mvdr = 10*log10(mmse_mvdr/max(mmse_mvdr));
mmse_music = 10*log10(mmse_music/max(mmse_music));
mmse_esprit = 10*log10(mmse_esprit/max(mmse_esprit));
figure(1);
plot(snr,(mmse_psd),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using FFT Approach');
% figure(2);
plot(snr,(mmse_PEM),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using EM Approach');
% figure(3);
plot(snr,(mmse_mvdr),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using MVDR Approach');
% figure(4);
plot(snr,(mmse_music),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using MUSIC Approach');
% figure(5);
plot(snr,(mmse_esprit),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using ESPRIT Approach');
plot(snr,(mmse_Window),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using Window Approach');
plot(snr,(mmse_MF),'linewidth',2);hold on;
% xlabel('SNR in dB');ylabel('MSE');
% title('MSE using MF Approach');
legend('FFT','EM','MVDR','MUSIC','ESPRIT','Window','MF')
% saveas(figure(1),'FFT.png');saveas(figure(2),'EM.png');
% saveas(figure(4),'MUSIC.png');
saveas(figure(3),'MVDR.png');
saveas(figure(5),'ESPRIT.png');
```