

CT509:
Direction of Arrival Estimation
Algorithms

November 1 2020

1 Spectral Resolution and Leakage

1.1 Case1:

$K=3$ and angle of arrivals are $[-1^\circ \ 0^\circ \ 1^\circ]$. 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.

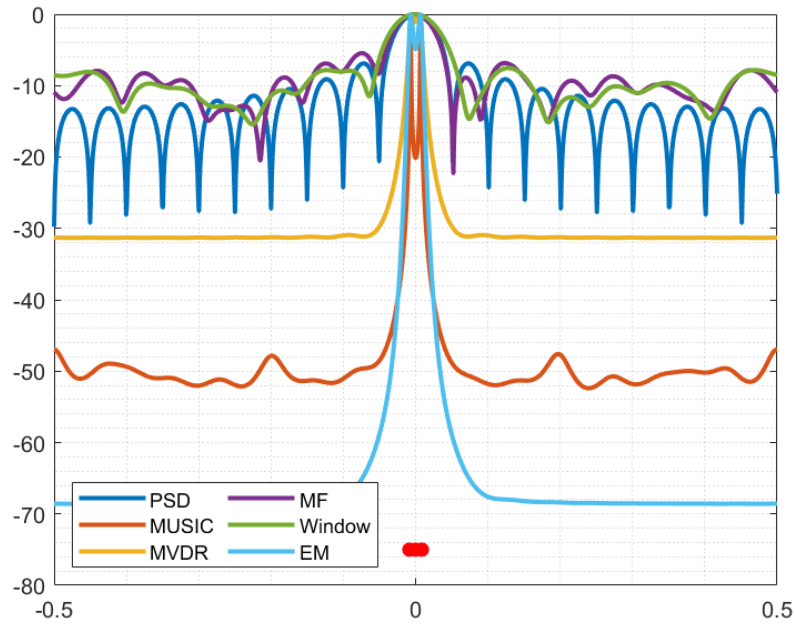


Figure 1: PSD in dB

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 $d \in \{2,4,5,6,8,10\}$

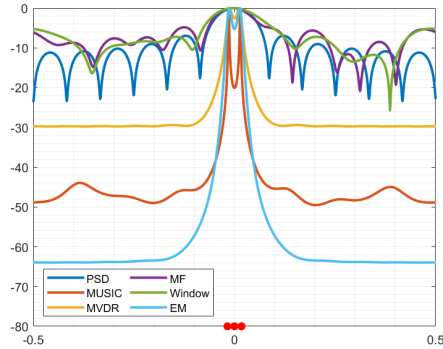


Figure 2: $[-2 \ 0 \ 2]$ ($d=2$)

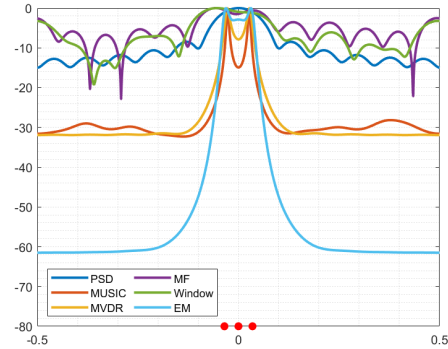


Figure 3: $[-4 \ 0 \ 4]$ ($d=4$)

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

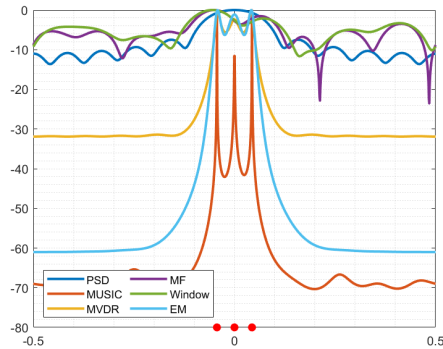


Figure 4: $[-5 \ 0 \ 5]$ ($d=5$)

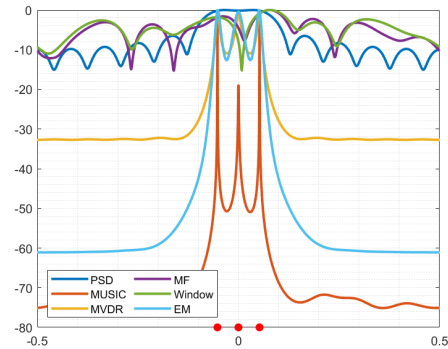


Figure 5: $[-6 \ 0 \ 6]$ ($d=6$)

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

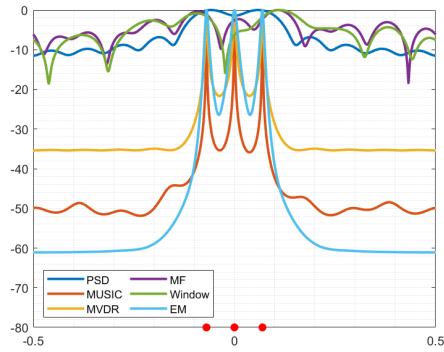


Figure 6: $[-8 \ 0 \ 8]$ ($d=8$)

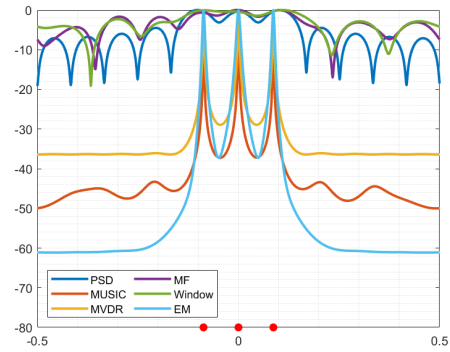
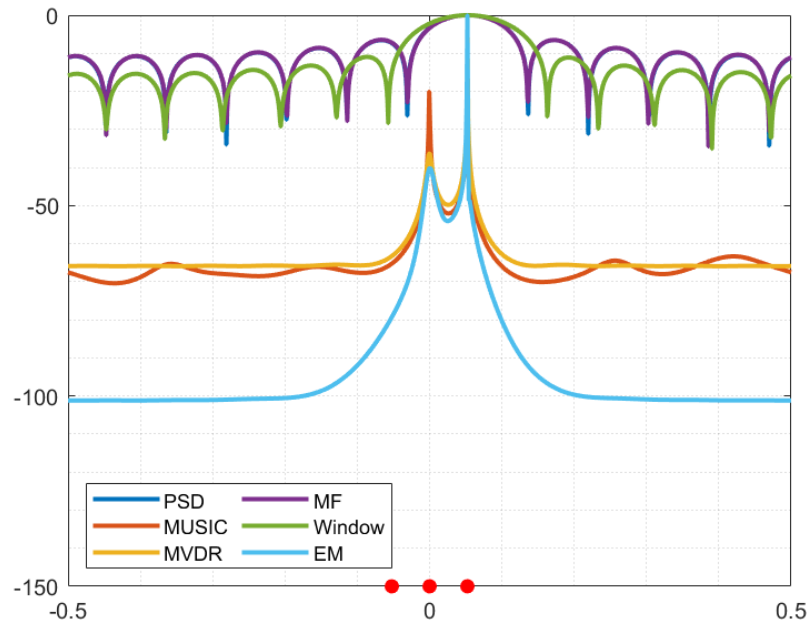


Figure 7: $[-10 \ 0 \ 10]$ ($d=10$)

- 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 $\{-20\text{dB } 0\text{dB } 20\text{dB}\}$

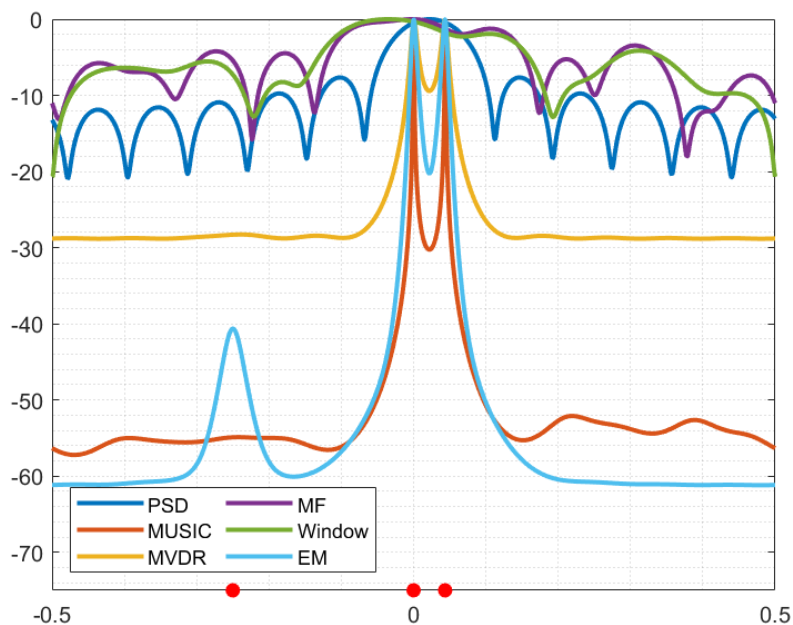


Observations:

- 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 $\{-20\text{dB } 0\text{dB } 0\text{dB}\}$



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.

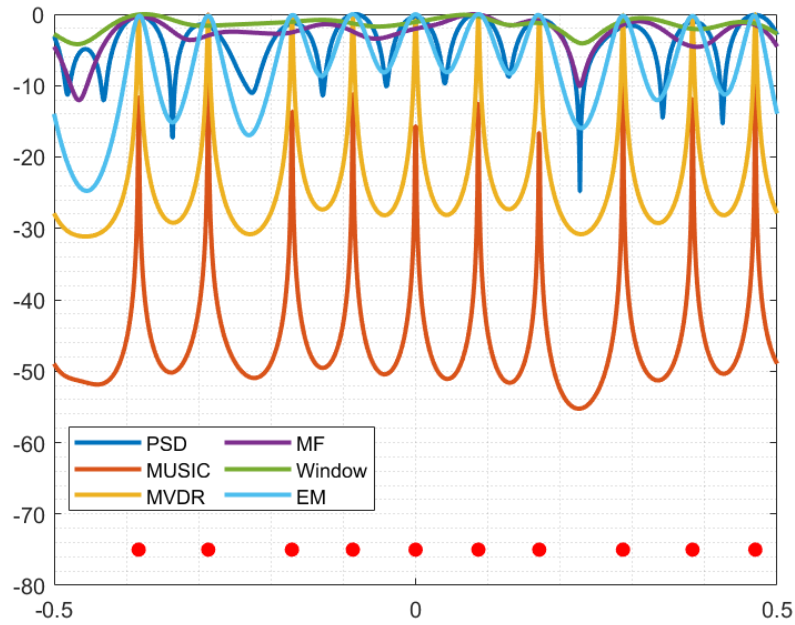
2 A Performance Study:

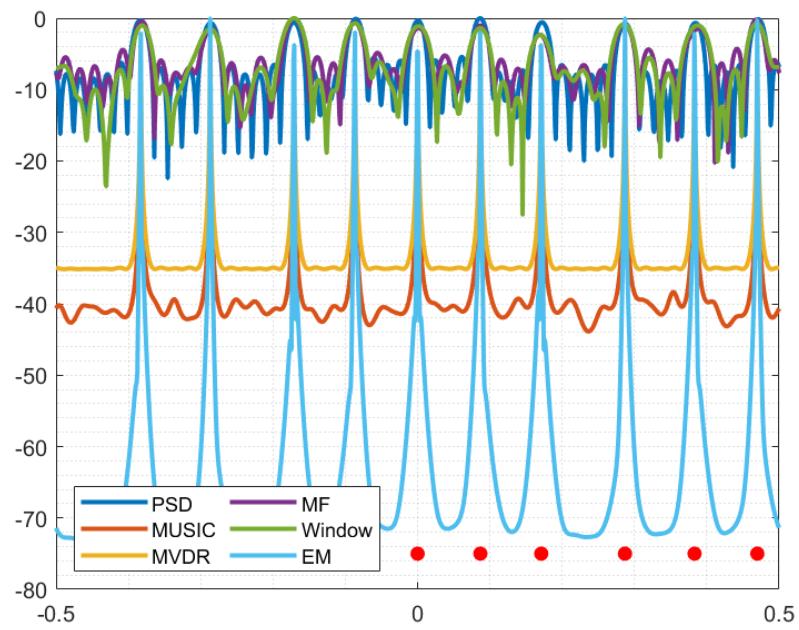
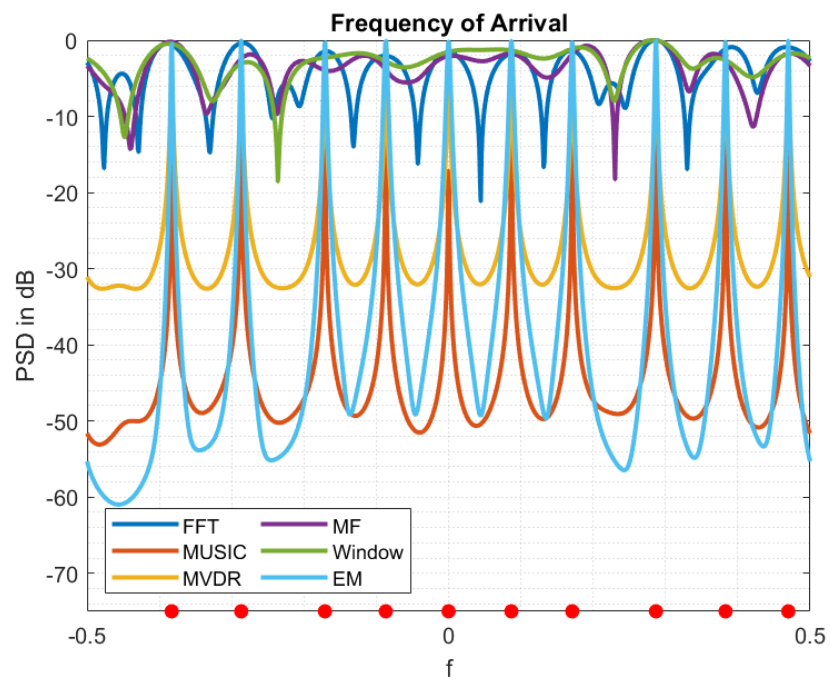
2.1 Performance of all algorithms with increase in number of elements

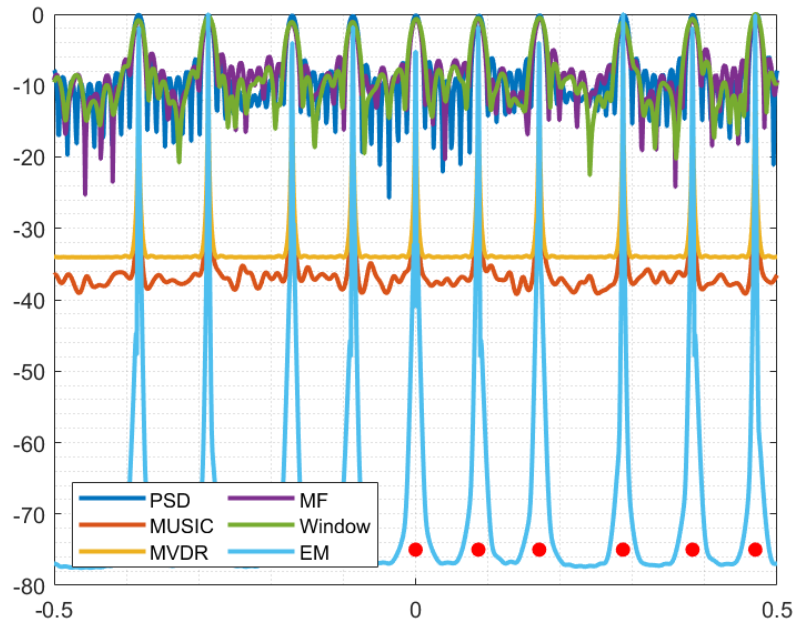
Number of users are set to 10, DoAs are the following angles and N is varied as [15 20 50 80] respectively

$$\theta^\circ = \{-50^\circ, -35^\circ, -20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ, 35^\circ, 50^\circ, 70^\circ\}$$

And signals are received at power level of 1W







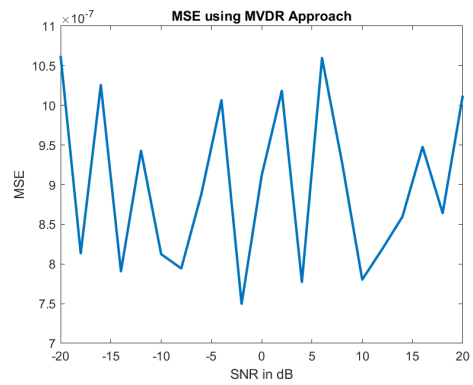
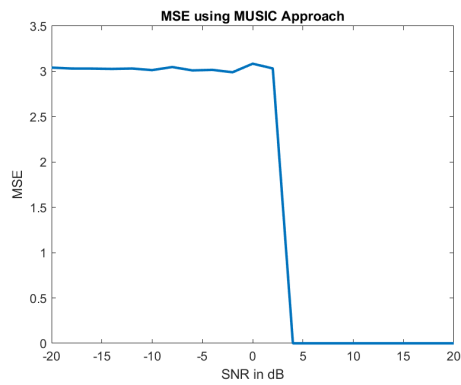
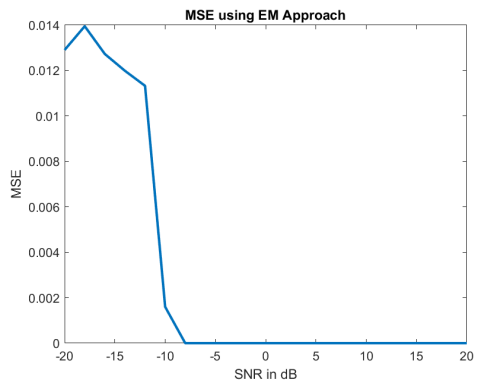
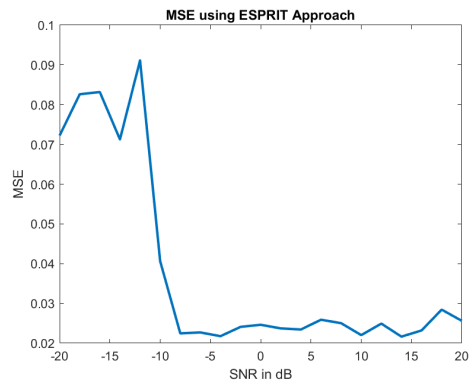
Observations

- 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.

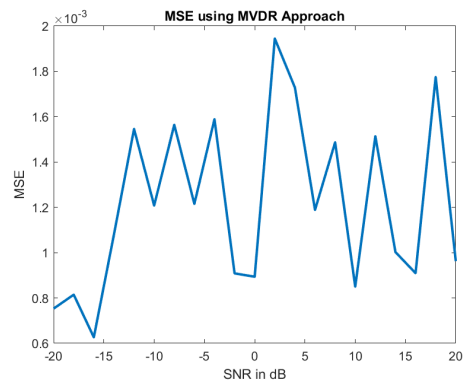
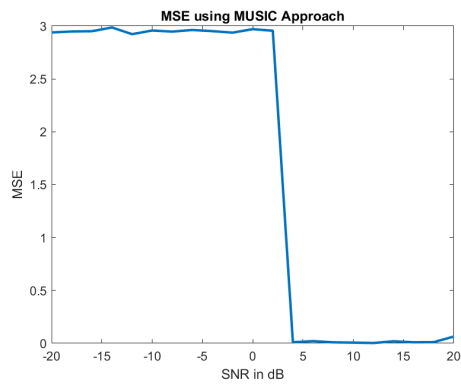
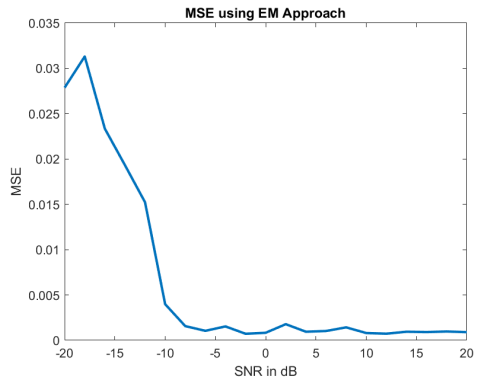
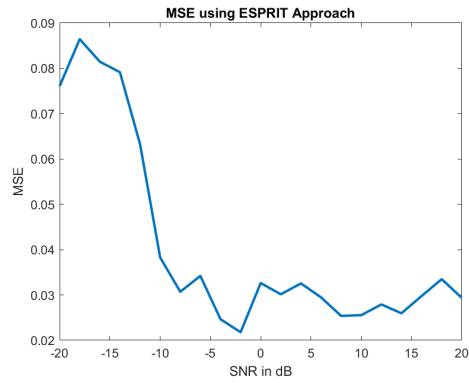
2.2 Monte-Carlo Simulations

- 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° .



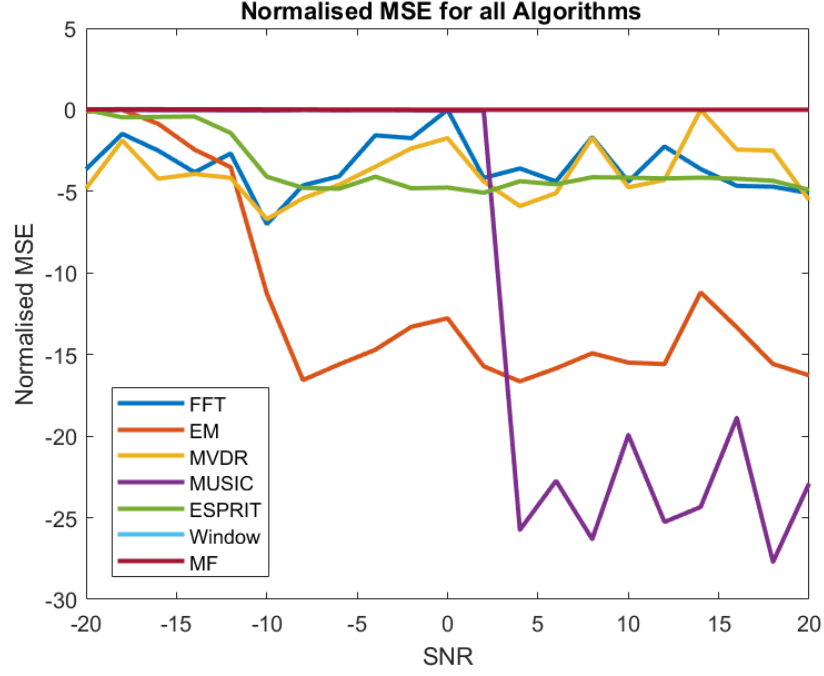


Figure 8: MSE vs SNR for all algorithms when $\Delta = 20^\circ$

Calculations

- MSE is calculated as follows

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

- Normalised MSE

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

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

```
1 clearvars;close all;clc;
2 N=20;n=0:N-1;K=5;
3 a_k = ones(K,1);mse = zeros(1,21);
4 M=1024;t=0;
5 snr = -20:2:20;
6 mse_psd=zeros(1,length(snr));
7 mse_PEM=zeros(1,length(snr));
8 ;mse_mvdr=zeros(1,length(snr));
9 mse_music=zeros(1,length(snr));
10 mse_esprit=zeros(1,length(snr));
11 for ii =1:length(snr)
12     for nsim = 1:50
13         SNR_dB=snr(ii);gamma=10^(SNR_dB/10);sigma=1/sqrt(gamma);
14         theta1 = 15*rand()-60;
15         theta2 = 15*rand()-30;
16         theta3 = 15*rand();
17         theta4 = -15*rand()+30;
18         theta5 = -15*rand()+60;
19         thetass =[theta1 theta2 theta3 theta4 theta5];
20         theta=thetass.*(pi/180);
21         f=sin(theta)/2;
22         phi_k=2*pi*rand(K,1)-pi;f=f(:);
23         xn_k = a_k.*exp(1j*2*pi*theta').*exp(1j*2*pi*f*(0:N-1));
24         %% PSD
25         for kk=1:N
26             xnk=xn_k(:,kk);
27             R=xnk*xnk';
28             t=t+R;
29             f_k=f.';
30             W=exp(-1j*2*pi*n'*f_k);
31             u = sigma*randn(N,1)+1j*randn(N,1);
32             xn = W*xnk+u;
33     end
```

```

34 R_ss=t/N;
35 Rss = diag(a_k.^2);Rxx=W*R_ss*W'+sigma^2*eye(N);
36 fo = linspace(-0.5,0.5,M);
37 r = fftshift(fft(Rxx(1,:),M));
38 r1 = abs(r);r1 = 10*log10(r1/max(r1));
39
40 a1 = [-70;-50];a2 = [-40;-20];a3 = [-10;10];a4 = [20;40];a5 =
    [50;70];
41 f_one = sin([a1 a2 a3 a4 a5]*pi/180)/2;
42 aa1 = find(abs(fo-f_one(1,1))<0.0005);ab1 =
    find(abs(fo-f_one(2,1))<0.0005);
43 aa2 = find(abs(fo-f_one(1,2))<0.0005);ab2 =
    find(abs(fo-f_one(2,2))<0.0005);
44 aa3 = find(abs(fo-f_one(1,3))<0.0005);ab3 =
    find(abs(fo-f_one(2,3))<0.0005);
45 aa4 = find(abs(fo-f_one(1,4))<0.0005);ab4 =
    find(abs(fo-f_one(2,4))<0.0005);
46 aa5 = find(abs(fo-f_one(1,5))<0.0005);ab5 =
    find(abs(fo-f_one(2,5))<0.0005);
47 for oo = 1:length(fo)
48     if oo>=aa1 && oo<ab1
49         r11(oo) = r1(oo);
50     else
51         r11(oo) = -80;
52     end
53 end
54 [mm,nn]=max(r11);f1_est = fo(nn);
55 for oo = 1:length(fo)
56     if oo>=aa2 && oo<ab2
57         r11(oo) = r1(oo);
58     else
59         r11(oo) = -80;
60     end
61 end
62 [mm,nn]=max(r11);f2_est = fo(nn);
63 for oo = 1:length(fo)
64     if oo>=aa3 && oo<ab3
65         r11(oo) = r1(oo);

```



```

66 else
67     r11(oo) = -80;
68 end
69 end
70 [mm,nn]=max(r11);f3_est = fo(nn);
71 for oo = 1:length(fo)
72     if oo>=aa4 && oo<ab4
73         r11(oo) = r1(oo);
74     else
75         r11(oo) = -80;
76     end
77 end
78 [mm,nn]=max(r11);f4_est = fo(nn);
79
80 for oo = 1:length(fo)
81     if oo>=aa5 && oo<ab5
82         r11(oo) = r1(oo);
83     else
84         r11(oo) = -80;
85     end
86 end
87 [mm,nn]=max(r11);f5_est = fo(nn);
88 f_estimated = [f1_est f2_est f3_est f4_est f5_est];
89 aoa = asin(2*f_estimated)*180/pi;
90 theta11 = sort(thetass)';
91 aoa11 = sort(aoa);
92 mmse_psd(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
93 %% MVDR
94 W1 = exp(-1j*2*pi*n'*fo);
95 for i = 1:M
96     P_MVDR(i) = 1/(W1(:,i)'*inv(Rxx)*W1(:,i));
97 end
98 Pmvdr=10*log10(abs(P_MVDR)/max(abs(P_MVDR))); %Spatial spectrum
    function
99 for oo = 1:length(fo)
100     if oo>=aa1 && oo<ab1
101         r11(oo) = Pmvdr(oo);
102     else

```

```

103     r11(oo) = -80;
104 end
105 end
106 [mm,nn]=max(r11);f1_est = fo(nn);
107 for oo = 1:length(fo)
108     if oo>=aa2 && oo<ab2
109         r11(oo) = Pmvdr(oo);
110     else
111         r11(oo) = -80;
112     end
113 end
114 [mm,nn]=max(r11);f2_est = fo(nn);
115 for oo = 1:length(fo)
116     if oo>=aa3 && oo<ab3
117         r11(oo) = Pmvdr(oo);
118     else
119         r11(oo) = -80;
120     end
121 end
122 [mm,nn]=max(r11);f3_est = fo(nn);
123 for oo = 1:length(fo)
124     if oo>=aa4 && oo<ab4
125         r11(oo) = Pmvdr(oo);
126     else
127         r11(oo) = -80;
128     end
129 end
130 [mm,nn]=max(r11);f4_est = fo(nn);
131
132 for oo = 1:length(fo)
133     if oo>=aa5 && oo<ab5
134         r11(oo) = Pmvdr(oo);
135     else
136         r11(oo) = -80;
137     end
138 end
139 [mm,nn]=max(r11);f5_est = fo(nn);
140 f_estimated = [f1_est f2_est f3_est f4_est f5_est];

```

```

141 aoa = asin(2*f_estimated);
142 theta11 = sort(theta)';
143 aoa11 = sort(aoa);
144 mmse_mvdr(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
145
146 %% MUSIC
147 [Q,D]=eig(Rxx);
148 k=sum(sum(round(D)~=0));
149 K1 = N-k;
150 V = Q(:, [k+1:end]);
151 W1 = exp(-1j*2*pi*n'*fo);
152 for i = 1:length(fo)
153   PsePSD(i) = 1/((W1(:,i)')*(V*V')*W1(:,i)));
154 end
155 Pmusic=10*log10(abs(PsePSD)/max(abs(PsePSD))); %Spatial spectrum
      function
156 for oo = 1:length(fo)
157   if oo>=aa1 && oo<ab1
158     r11(oo) = Pmusic(oo);
159   else
160     r11(oo) = -80;
161   end
162 end
163 [mm,nn]=max(r11);f1_est = fo(nn);
164 for oo = 1:length(fo)
165   if oo>=aa2 && oo<ab2
166     r11(oo) = Pmusic(oo);
167   else
168     r11(oo) = -80;
169   end
170 end
171 [mm,nn]=max(r11);f2_est = fo(nn);
172 for oo = 1:length(fo)
173   if oo>=aa3 && oo<ab3
174     r11(oo) = Pmusic(oo);
175   else
176     r11(oo) = -80;
177   end

```

```

178 end
179 [mm,nn]=max(r11);f3_est = fo(nn);
180 for oo = 1:length(fo)
181 if oo>=aa4 && oo<ab4
182     r11(oo) = Pmusic(oo);
183 else
184     r11(oo) = -80;
185 end
186 end
187 [mm,nn]=max(r11);f4_est = fo(nn);
188
189 for oo = 1:length(fo)
190 if oo>=aa5 && oo<ab5
191     r11(oo) = Pmusic(oo);
192 else
193     r11(oo) = -80;
194 end
195 end
196 [mm,nn]=max(r11);f5_est = fo(nn);
197 f_estimated = [f1_est f2_est f3_est f4_est f5_est];
198 aoa = asin(2*f_estimated);
199 theta11 = sort(theta)';
200 aoa11 = sort(aoa);
201 mmse_music(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
202 %% EM
203 W1 = exp(-1j*2*pi*n'*fo);%Reference filter bank
204 %Random initialization of Rzz
205 rnf = rand(M,1);
206 Rzz = diag(rnf.^2);
207 Xn = W*xn_k+sigma^2*eye(N);
208 % Rzz = eye(M);
209 for i = 1:20
210     Rxx1 = W1*Rzz*W1';
211     Rxx1 = Rxx1/N;
212     Rxx_inv = inv(Rxx1);
213     for mm = 1:M
214         w_m =W1(:,mm);
215         w_mvdr = Rxx_inv*w_m/(w_m'*Rxx_inv*w_m);

```

```

216         y_em = w_mvdr'*Xn;
217         %         for aa = 1:12
218 %             y_em(:,aa) = w_mvdr'*Xn(:,aa);
219 %         end
220         Rzz(mm,mm) = y_em*y_em';
221     end
222 end
223 P_em = abs(diag(Rzz));
224 P_EM=10*log10((P_em)/max((P_em)));
225
226 for oo = 1:length(fo)
227     if oo>=aa1 && oo<ab1
228         r11(oo) = P_EM(oo);
229     else
230         r11(oo) = -80;
231     end
232 end
233 [mm,nn]=max(r11);f1_est = fo(nn);
234 for oo = 1:length(fo)
235     if oo>=aa2 && oo<ab2
236         r11(oo) = P_EM(oo);
237     else
238         r11(oo) = -80;
239     end
240 end
241 [mm,nn]=max(r11);f2_est = fo(nn);
242 for oo = 1:length(fo)
243     if oo>=aa3 && oo<ab3
244         r11(oo) = P_EM(oo);
245     else
246         r11(oo) = -80;
247     end
248 end
249 [mm,nn]=max(r11);f3_est = fo(nn);
250 for oo = 1:length(fo)
251     if oo>=aa4 && oo<ab4
252         r11(oo) = P_EM(oo);
253     else

```

```

254     r11(oo) = -80;
255 end
256 end
257 [mm,nn]=max(r11);f4_est = fo(nn);
258
259 for oo = 1:length(fo)
260 if oo>=aa5 && oo<ab5
261     r11(oo) = P_EM(oo);
262 else
263     r11(oo) = -80;
264 end
265 end
266 [mm,nn]=max(r11);f5_est = fo(nn);
267 f_estimated = [f1_est f2_est f3_est f4_est f5_est];
268 aoa = asin(2*f_estimated);
269 theta11 = sort(theta)';
270 aoa11 = sort(aoa);
271 mmse_PEM(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
272 %% ESPRIT
273 [U,S,V] = svd(Xn);
274 MM = diag(S);
275 [rr,cc]=find(abs(MM)-sigma^2<0.00001);
276 kapa = sum(cc);
277 KK = N-kapa;
278 U_est = U(:, [1:K]);val = 1;
279 J_x = [eye(N-val) zeros(N-val,1)];
280 J_y = [zeros(N-val,1) eye(N-val)];
281
282 Ux = J_x*U_est;
283 Uy = J_y*U_est;
284 Z = pinv(Ux)*Uy;
285 [T,thetas] = eig(Z);
286 ang = asin(angle(thetas)/pi)*180/pi;
287 ang = diag(ang)*pi/180;
288 W_1 =T*U_est';
289 theta11 = sort(thetass) '*pi/180;
290 aoa11 = sort(ang);
291 mmse_esprit(nsim,ii) = (1/K)*sum((theta11-aoa11).^2);

```

```

292 %% MF
293 y = xn'*W1;
294 y_mf=10*log10(abs(y)/max(abs(y)));
295 for oo = 1:length(fo)
296     if oo>=aa1 && oo<ab1
297         r11(oo) = y_mf(oo);
298     else
299         r11(oo) = -80;
300     end
301 end
302 [mm,nn]=max(r11);f1_est = fo(nn);
303 for oo = 1:length(fo)
304     if oo>=aa2 && oo<ab2
305         r11(oo) = y_mf(oo);
306     else
307         r11(oo) = -80;
308     end
309 end
310 [mm,nn]=max(r11);f2_est = fo(nn);
311 for oo = 1:length(fo)
312     if oo>=aa3 && oo<ab3
313         r11(oo) = y_mf(oo);
314     else
315         r11(oo) = -80;
316     end
317 end
318 [mm,nn]=max(r11);f3_est = fo(nn);
319 for oo = 1:length(fo)
320     if oo>=aa4 && oo<ab4
321         r11(oo) = y_mf(oo);
322     else
323         r11(oo) = -80;
324     end
325 end
326 [mm,nn]=max(r11);f4_est = fo(nn);
327
328 for oo = 1:length(fo)
329     if oo>=aa5 && oo<ab5

```

```

330     r11(oo) = y_mf(oo);
331 else
332     r11(oo) = -80;
333 end
334 end
335 [mm,nn]=max(r11);f5_est = fo(nn);
336 f_estimated = [f1_est f2_est f3_est f4_est f5_est];
337 aoa = asin(2*f_estimated);
338 theta11 = sort(theta)';
339 aoa11 = sort(aoa);
340 mmse_MF(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
341
342 %% Window
343 wc = kaiser(N,2.4);
344 xn = xn.*wc;
345 y = xn'*w1;
346 y_win=10*log10(abs(y)/max(abs(y)));
347 for oo = 1:length(fo)
348     if oo>=aa1 && oo<ab1
349         r11(oo) = y_win(oo);
350     else
351         r11(oo) = -80;
352     end
353 end
354 [mm,nn]=max(r11);f1_est = fo(nn);
355 for oo = 1:length(fo)
356     if oo>=aa2 && oo<ab2
357         r11(oo) = y_win(oo);
358     else
359         r11(oo) = -80;
360     end
361 end
362 [mm,nn]=max(r11);f2_est = fo(nn);
363 for oo = 1:length(fo)
364     if oo>=aa3 && oo<ab3
365         r11(oo) = y_win(oo);
366     else
367         r11(oo) = -80;

```



```

368 end
369 end
370 [mm,nn]=max(r11);f3_est = fo(nn);
371 for oo = 1:length(fo)
372 if oo>=aa4 && oo<ab4
373     r11(oo) = y_win(oo);
374 else
375     r11(oo) = -80;
376 end
377 end
378 [mm,nn]=max(r11);f4_est = fo(nn);
379
380 for oo = 1:length(fo)
381 if oo>=aa5 && oo<ab5
382     r11(oo) = y_win(oo);
383 else
384     r11(oo) = -80;
385 end
386 end
387 [mm,nn]=max(r11);f5_est = fo(nn);
388 f_estimated = [f1_est f2_est f3_est f4_est f5_est];
389 aoa = asin(2*f_estimated);
390 theta11 = sort(theta)';
391 aoa11 = sort(aoa);
392 mmse_Window(nsim,ii) = (1/K)*sum((theta11'-aoa11).^2);
393
394 end
395 end
396 %%
397 % mmse = mmse/mnop;
398 mmse_psd = sum(mmse_psd)/nsim;
399 mmse_PEM = sum(mmse_PEM)/nsim;
400 mmse_mvdr = sum(mmse_mvdr)/nsim;
401 mmse_music = sum(mmse_music)/nsim;
402 mmse_esprit = sum(mmse_esprit)/nsim;
403 mmse_Window = sum(mmse_Window)/nsim;
404 mmse_MF = sum(mmse_MF)/nsim;
405

```

```

406 mmse_psd = 10*log10(mmse_psd/max(mmse_psd));
407 mmse_PEM = 10*log10(mmse_PEM/max(mmse_PEM));
408 mmse_mvdr = 10*log10(mmse_mvdr/max(mmse_mvdr));
409 mmse_music = 10*log10(mmse_music/max(mmse_music));
410 mmse_esprit = 10*log10(mmse_esprit/max(mmse_esprit));
411 figure(1);
412 plot(snr,(mmse_psd),'linewidth',2);hold on;
413 % xlabel('SNR in dB');ylabel('MSE');
414 % title('MSE using FFT Approach');
415 % figure(2);
416 plot(snr,(mmse_PEM),'linewidth',2);hold on;
417 % xlabel('SNR in dB');ylabel('MSE');
418 % title('MSE using EM Approach');
419 % figure(3);
420 plot(snr,(mmse_mvdr),'linewidth',2);hold on;
421 % xlabel('SNR in dB');ylabel('MSE');
422 % title('MSE using MVDR Approach');
423 % figure(4);
424 plot(snr,(mmse_music),'linewidth',2);hold on;
425 % xlabel('SNR in dB');ylabel('MSE');
426 % title('MSE using MUSIC Approach');
427 % figure(5);
428 plot(snr,(mmse_esprit),'linewidth',2);hold on;
429 % xlabel('SNR in dB');ylabel('MSE');
430 % title('MSE using ESPRIT Approach');
431 plot(snr,(mmse_Window),'linewidth',2);hold on;
432 % xlabel('SNR in dB');ylabel('MSE');
433 % title('MSE using Window Approach');
434 plot(snr,(mmse_MF),'linewidth',2);hold on;
435 % xlabel('SNR in dB');ylabel('MSE');
436 % title('MSE using MF Approach');
437 legend('FFT','EM','MVDR','MUSIC','ESPRIT','Window','MF')
438 % saveas(figure(1),'FFT.png');saveas(figure(2),'EM.png');
439 % saveas(figure(4),'MUSIC.png');
440 saveas(figure(3),'MVDR.png');
441 saveas(figure(5),'ESPRIT.png');

```
