EE 779 Advanced topics in signal processing

Assignment 4 simulations

Swrangsar Basumatary Roll: 09d07040

Given, theta1 = 0 degrees. We carry out the experiment for two values of 'theta2' i.e 15 degrees and 7.5 degrees.

The functions:

```
function ulaData = generateULAData(numberOfSensors,
numberOfSnapshots, secondAngle)

centerFrequecncy = 4000;
spacing = 0.5;
firstAngle = 0;

sourceMatrix1 = getSourceMatrix(numberOfSensors,
numberOfSnapshots, firstAngle, spacing, centerFrequecncy);
sourceMatrix2 = getSourceMatrix(numberOfSensors,
numberOfSnapshots, secondAngle, spacing, centerFrequecncy);
sourceMatrix = sourceMatrix1 + sourceMatrix2;

ulaData = sourceMatrix + getErrorMatrix(numberOfSensors,
numberOfSnapshots);

end

%% make array transfer vector a(theta)

function directionVector = getDirectionVector(angle, spacing,
numberOfSensors, centerFrequency)

wavelength = 1;
speedOfPropagation = centerFrequency * wavelength;
wc = 2 * pi * centerFrequency;
angle = angle * pi / 180;
k = 1:numberOfSensors;
delay = (k - 1) .* (spacing * sin(angle) / speedOfPropagation);
directionVector = exp(-1i * wc * delay);
```

```
directionVector = directionVector(:);

end

%% get a source signal

function sourceSignal = getSourceSignal(numberOfSnapshots)

sourceSignal = wgn(1, numberOfSnapshots, 1, 'linear',
'complex');

end

%% get an error signal

function errorSignal = getErrorSignal(numberOfSnapshots)

errorSignal = wgn(1, numberOfSnapshots, 1, 'linear', 'complex');

end

%% get an error matrix

function errorMatrix = getErrorMatrix(numberOfSensors,
numberOfSnapshots)

errorSignalCell = cell(numberOfSensors, 1);
for k = 1:numberOfSensors
    errorSignalCell{k} = getErrorSignal(numberOfSnapshots);
end
errorMatrix = cat(1, errorSignalCell{:});

end

%% get a source matrix

function sourceMatrix = getSourceMatrix(numberOfSensors,
numberOfSnapshots, angle, spacing, centerFrequency)

sourceSignal = getSourceSignal(numberOfSnapshots);
directionVector = getDirectionVector(angle, spacing,
numberOfSensors, centerFrequency);
sourceMatrix = directionVector * sourceSignal;

end
```

```
function avgUlaData = getMonteCarloUlaData(numberOfRealizations,
secondAngle)

m = 10; N = 100;

avgUlaData = generateULAData(m, N, secondAngle);
for k = 1:numberOfRealizations-1
    avgUlaData = avgUlaData + generateULAData(m, N,
secondAngle);
end
avgUlaData = avgUlaData ./ numberOfRealizations;

end


function estimatedSpectrum = getSpectrum(frequencyEstimates)

N=100;
t = 1:N;

x = zeros(size(t));
for k = 1:length(frequencyEstimates)
    x = x + (exp(1i * frequencyEstimates(k) .* t));
end

% now get the spectrum using fft

M = 2 ^ nextpow2(4 * length(x));
estimatedSpectrum = 10 * log10(abs(fftshift(fft(x, M))));
L = length(estimatedSpectrum); % we want only -pi/2 to pi/2
estimatedSpectrum = estimatedSpectrum(round(L/4):round(3 * L/
4));

end
```

Plotting functions:

```
function plotSpectraOfUlaData(secondAngle, numberOfAngleSamples)


%% do beamforming

ULAData = getMonteCarloUlaData(50, secondAngle);
```

```
phi = beamform(ULAData, numberOfAngleSamples, 0.5);
figure;
plot(phi); axis tight;
title(['Beamforming data for ',
num2str(numberOfAngleSamples), ...
    ' angle samples at \theta_1 = 0 and \theta_2 = ',
num2str(secondAngle), ' degrees']);

%% do Capon beamforming

phiCapon = capon_sp(ULAData, numberOfAngleSamples, 0.5);
figure;
plot(phiCapon); axis tight;
title(['Capon beamforming data for ',
num2str(numberOfAngleSamples), ...
    ' angle samples at \theta_1 = 0 and \theta_2 = ',
num2str(secondAngle), ' degrees']);

%% do root MUSIC and plot spectrum

doaMusic = root_music_doa(ULAData, 2, 0.5) .* (pi/180);
rootMusicSpectrum = getSpectrum(doaMusic);
w = -(length(rootMusicSpectrum)/2):(length(rootMusicSpectrum)/
2)-1;
w = 180 * w ./ length(rootMusicSpectrum);
figure;
plot(w, rootMusicSpectrum);
axis tight;
title(['Root MUSIC spectrum for ',
num2str(numberOfAngleSamples), ...
    ' angle samples at \theta_1 = 0 and \theta_2 = ',
num2str(secondAngle), ' degrees']);
xlabel('Angles in degrees');

%% do  ESPRIT and plot spectrum

doaESPRIT = esprit_doa(ULAData, 2, 0.5) .* (pi/180);
espritSpectrum = getSpectrum(doaESPRIT);
w = -(length(espritSpectrum)/2):(length(espritSpectrum)/2)-1;
w = 180 * w ./ length(espritSpectrum);
figure;
plot(w, espritSpectrum);
axis tight;
title(['ESPRIT spectrum for ',
num2str(numberOfAngleSamples), ...
```

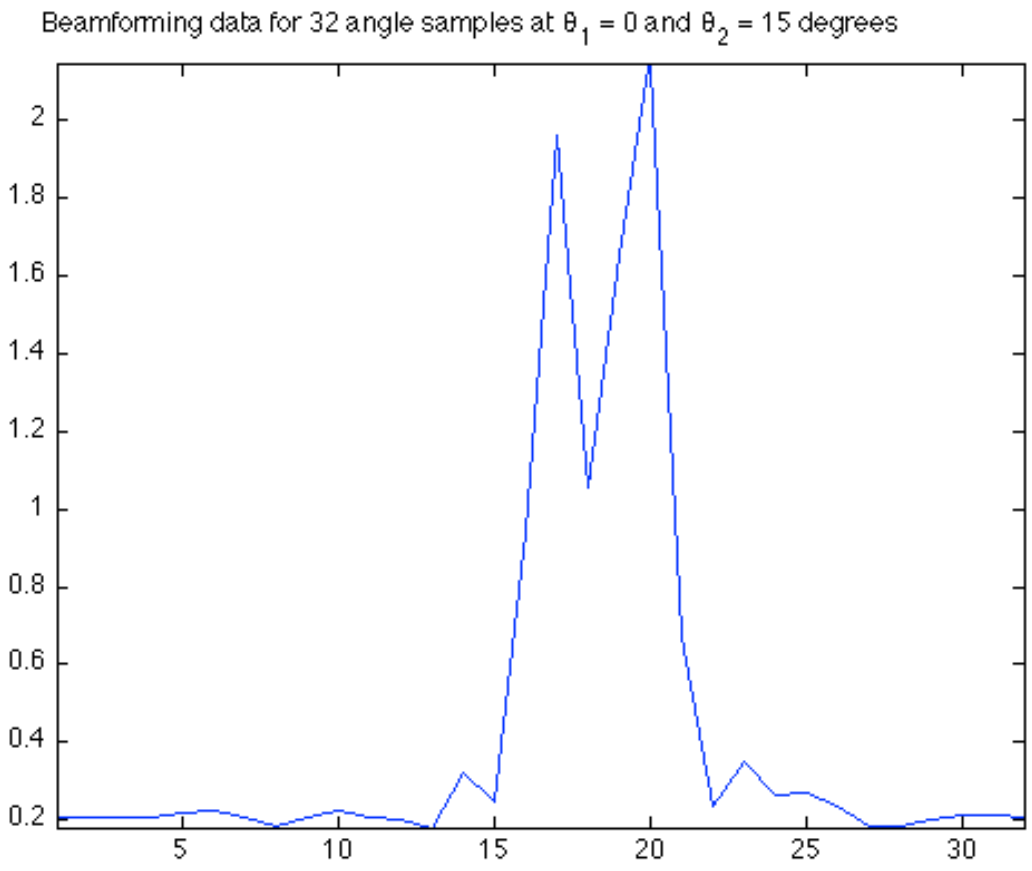
```
    ' angle samples at \theta_1 = 0 and \theta_2 = ',
num2str(secondAngle), ' degrees']);
xlabel('Angles in degrees');


end
```

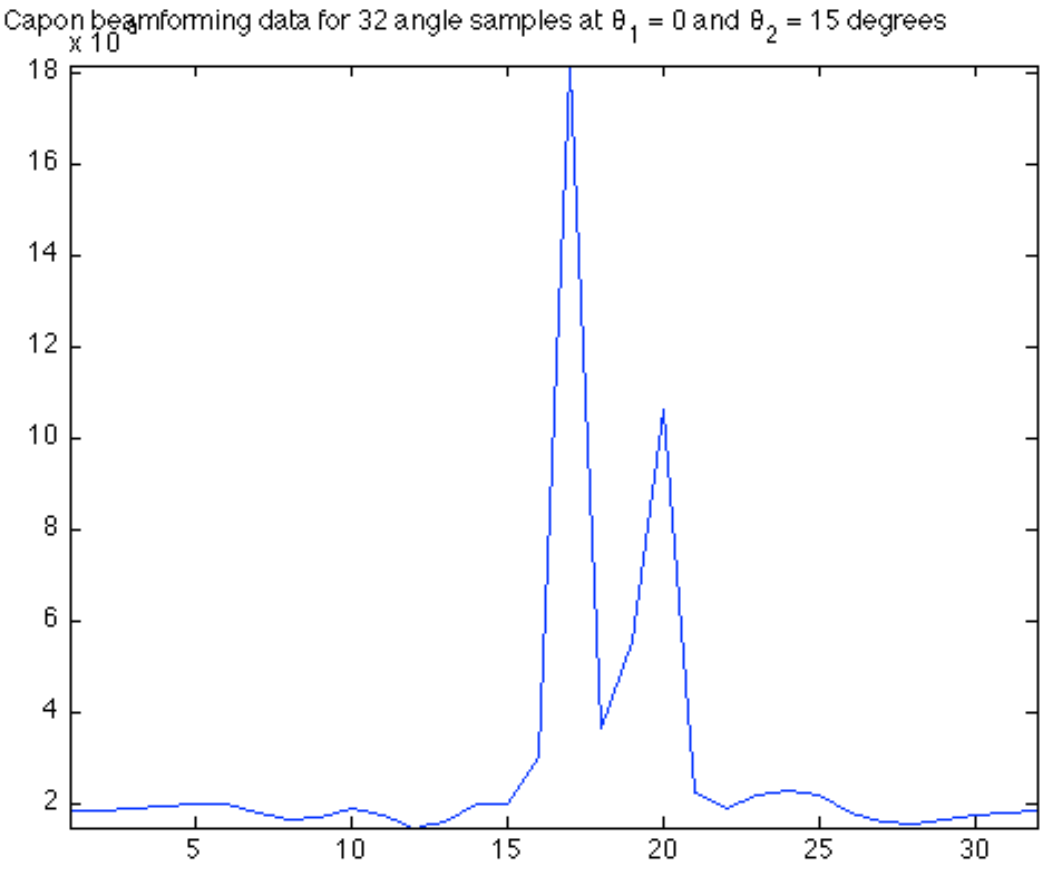
Plotting script:

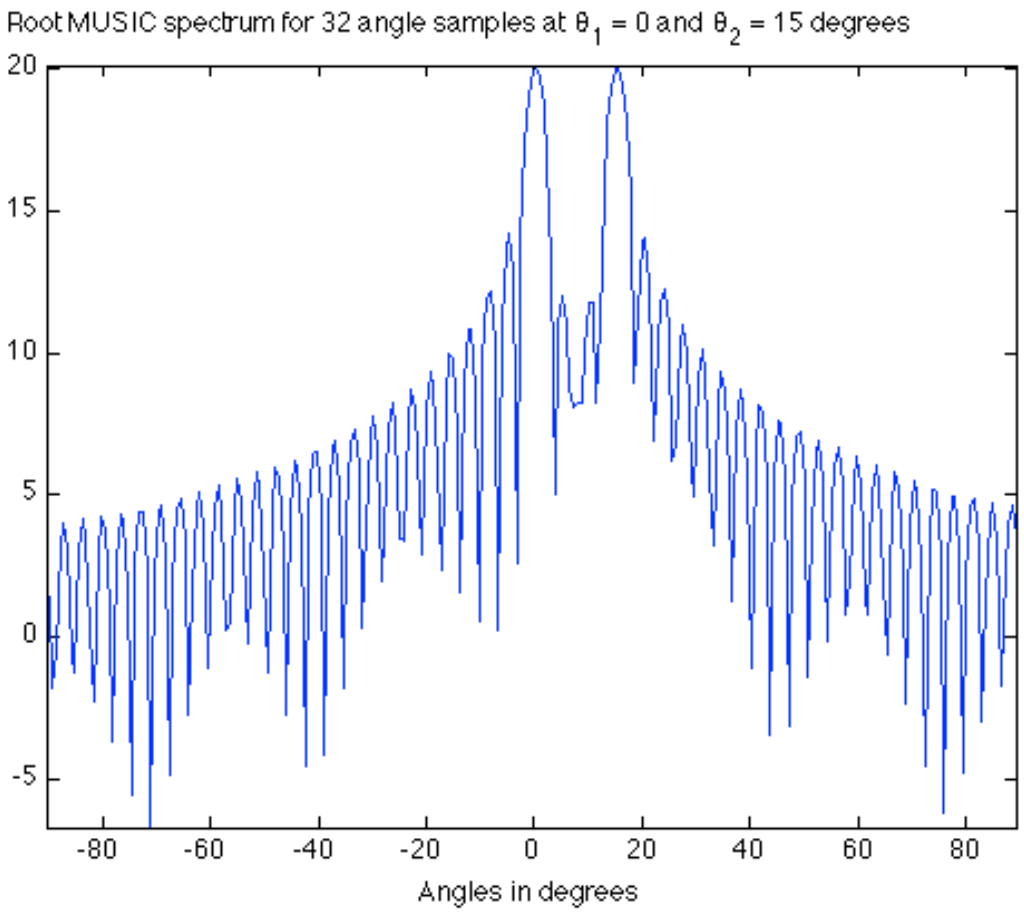
```
close all; clear all;

addpath /Users/swrangsarbasumatary/Desktop/advSignalProcAsgn4/
ch6/
addpath /Users/swrangsarbasumatary/Desktop/advSignalProcAsgn4/

plotSpectraOfUlaData(15, 32);
plotSpectraOfUlaData(15, 64);
plotSpectraOfUlaData(7.5, 64);
plotSpectraOfUlaData(7.5, 512);


rmpath /Users/swrangsarbasumatary/Desktop/advSignalProcAsgn4/
rmpath /Users/swrangsarbasumatary/Desktop/advSignalProcAsgn4/
ch6/
```

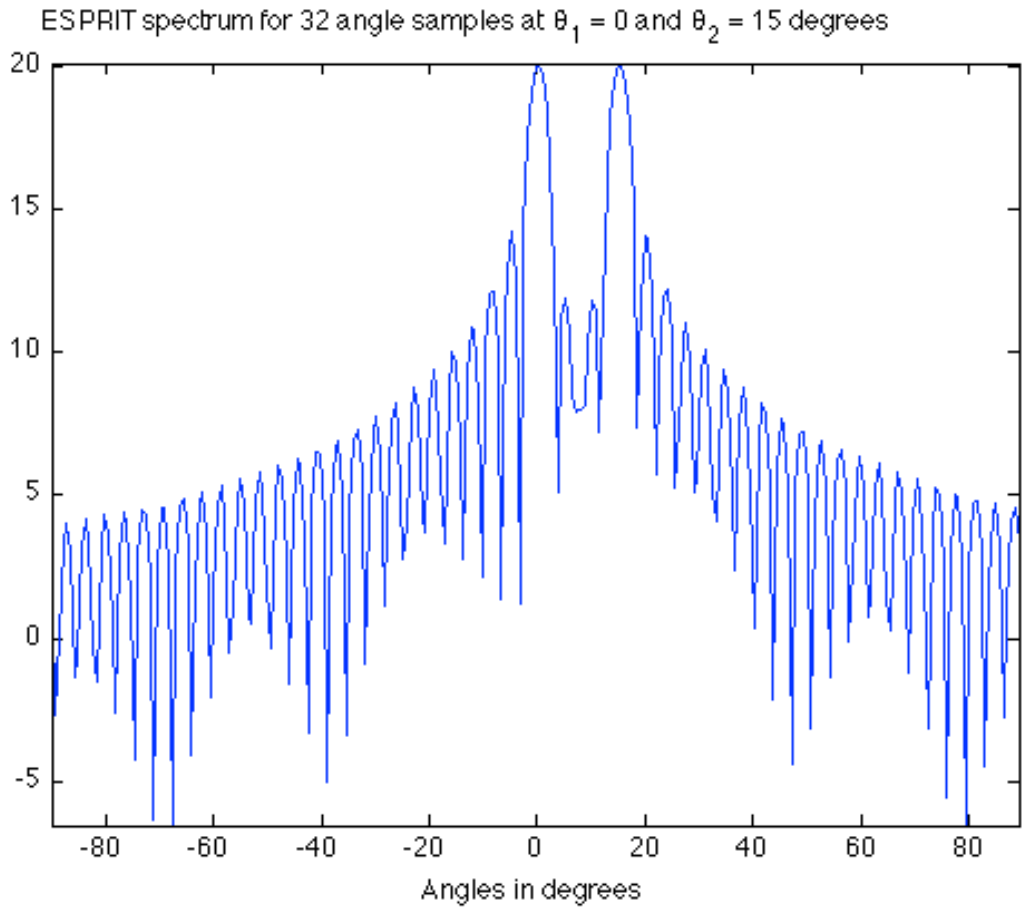
The plots:

For 32 samples of angles in [-90, 90] (in degrees) and theta2 = 15 degrees

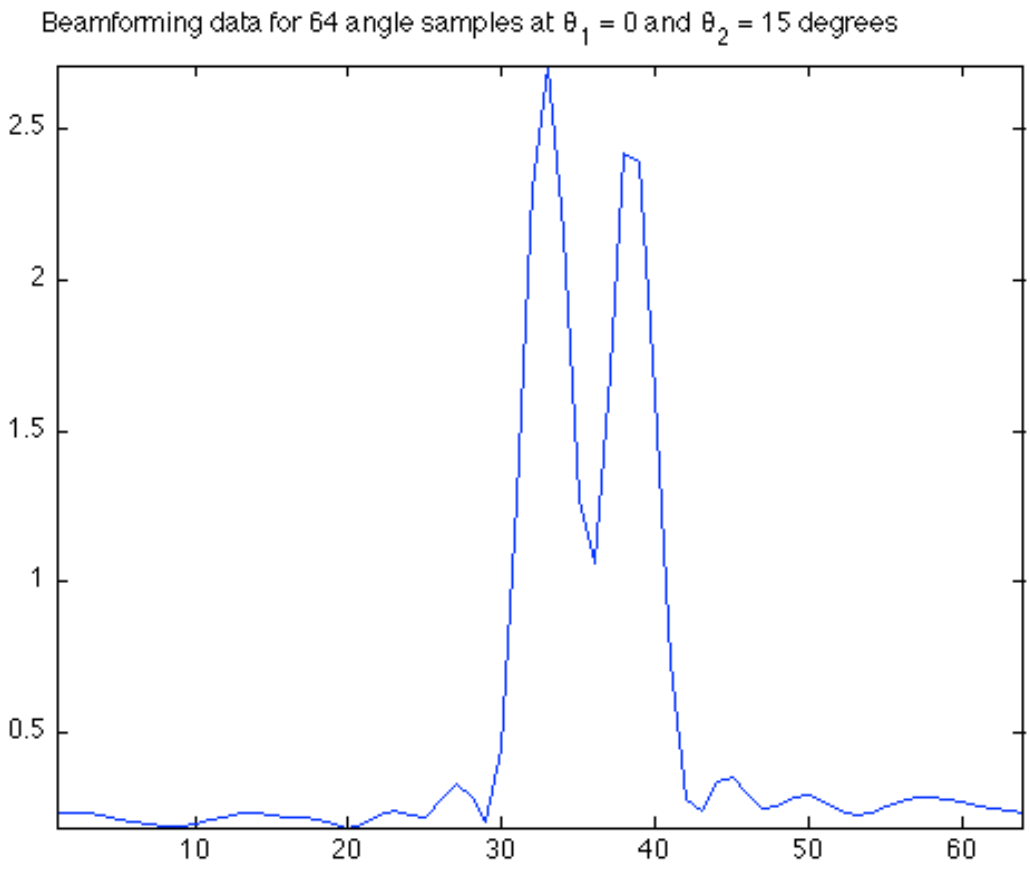

Beamforming data for 32 angle samples at θ1 = 0 and θ2 = 15 degrees
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
5
10
15
20
25
30

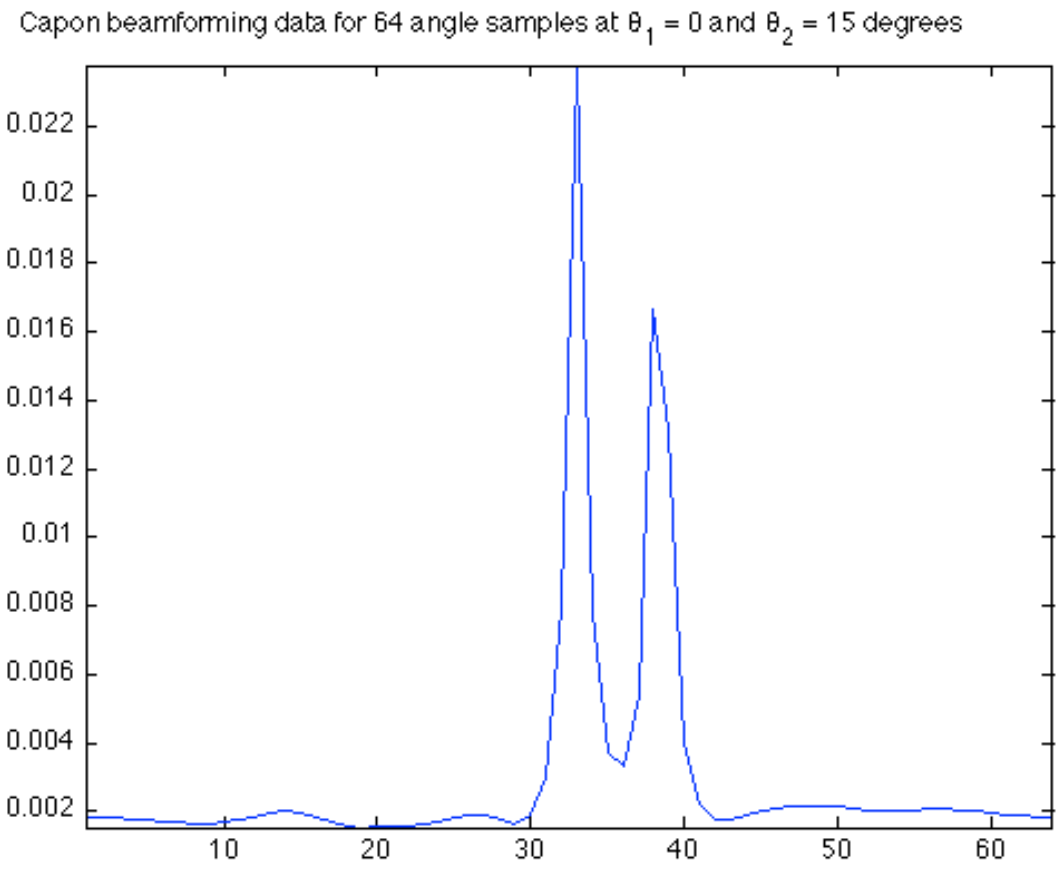

Capon beamforming data for 32 angle samples at θ1 = 0 and θ2 = 15 degrees
x 10^3
18
16
14
12
10
8
6
4
2
5
10
15
20
25
30

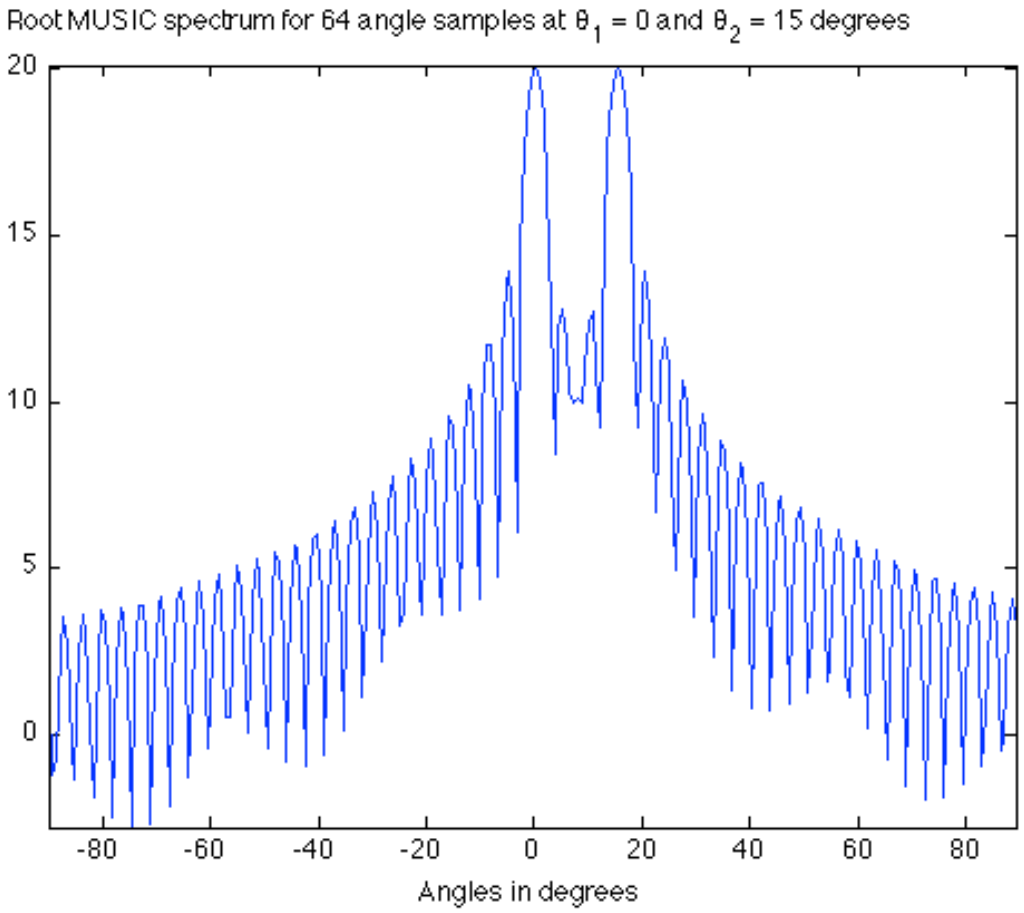

Root MUSIC spectrum for 32 angle samples at θ1 = 0 and θ2 = 15 degrees
20
15
10
5
0
-5
-80
-60
-40
-20
0
20
40
60
80
Angles in degrees

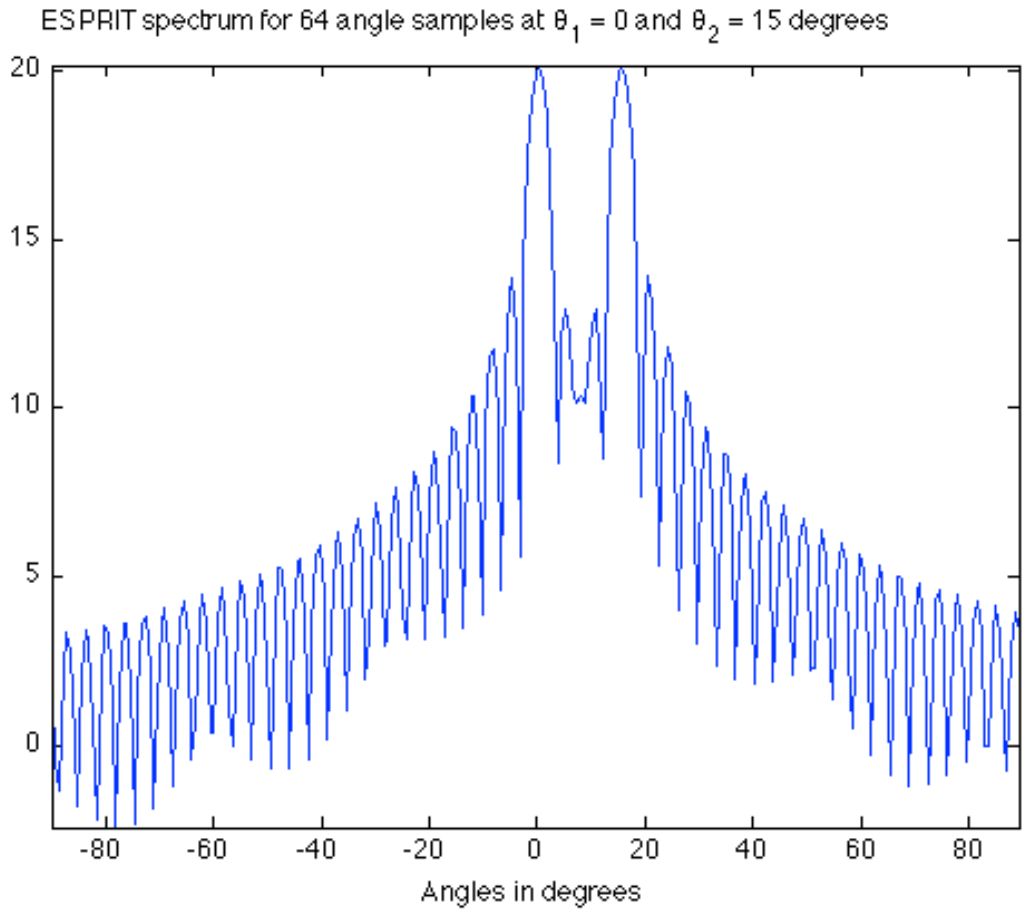


For 64 samples of angles in [-90, 90] (in degrees) and theta2 = 15 degrees

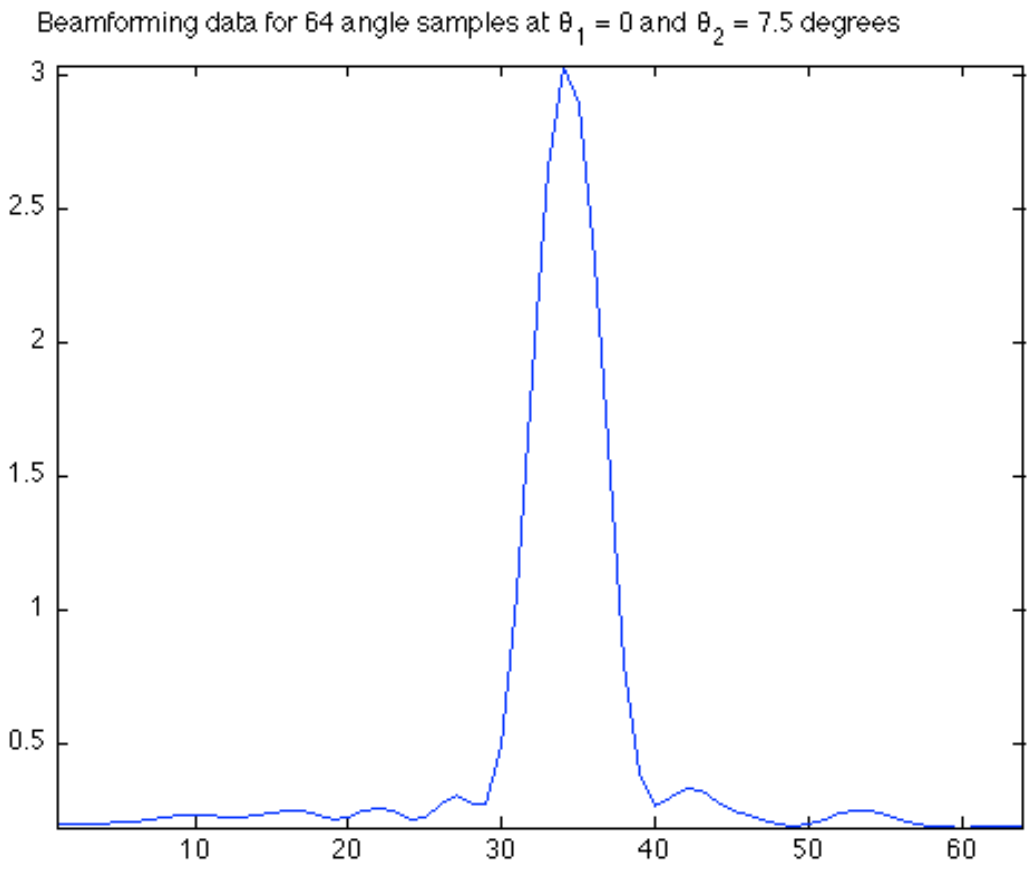

Beamforming data for 64 angle samples at θ1 = 0 and θ2 = 15 degrees
2.5
2
1.5
1
0.5
10
20
30
40
50
60

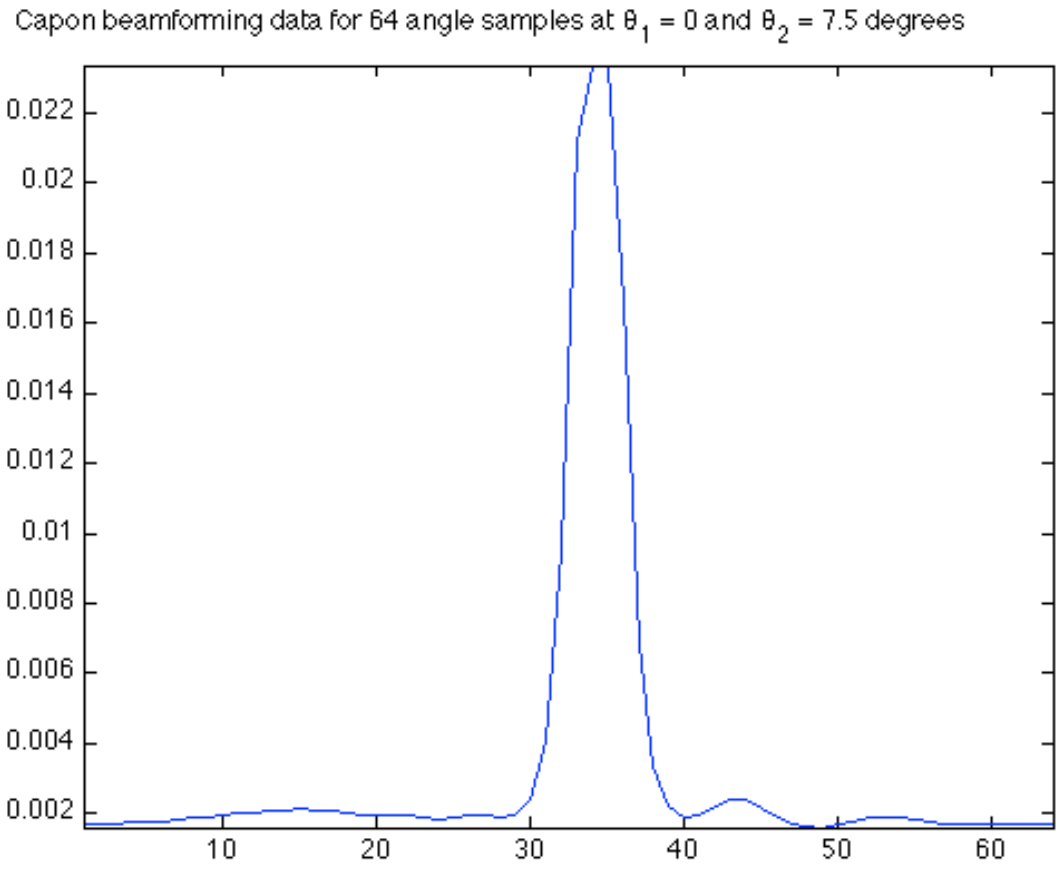

Capon beamforming data for 64 angle samples at θ1 = 0 and θ2 = 15 degrees
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
10
20
30
40
50
60

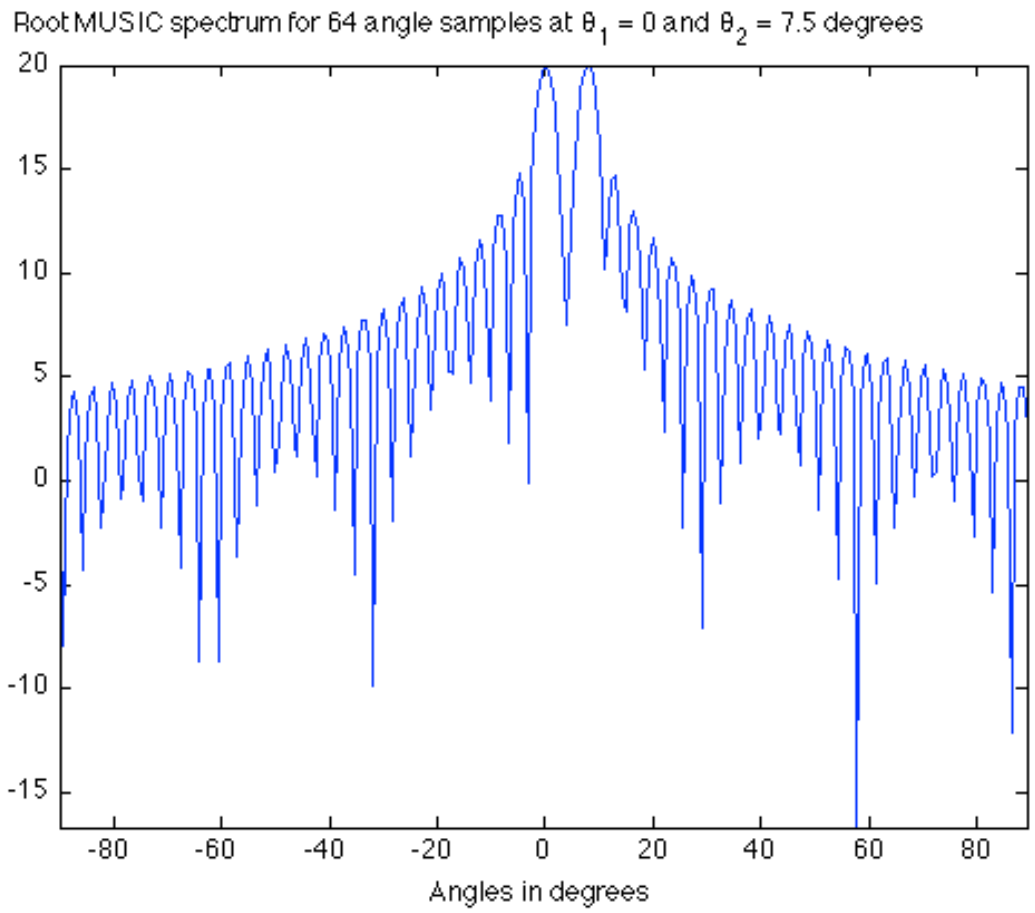

Root MUSIC spectrum for 64 angle samples at θ1 = 0 and θ2 = 15 degrees
20
15
10
5
0
-80
-60
-40
-20
0
20
40
60
80
Angles in degrees

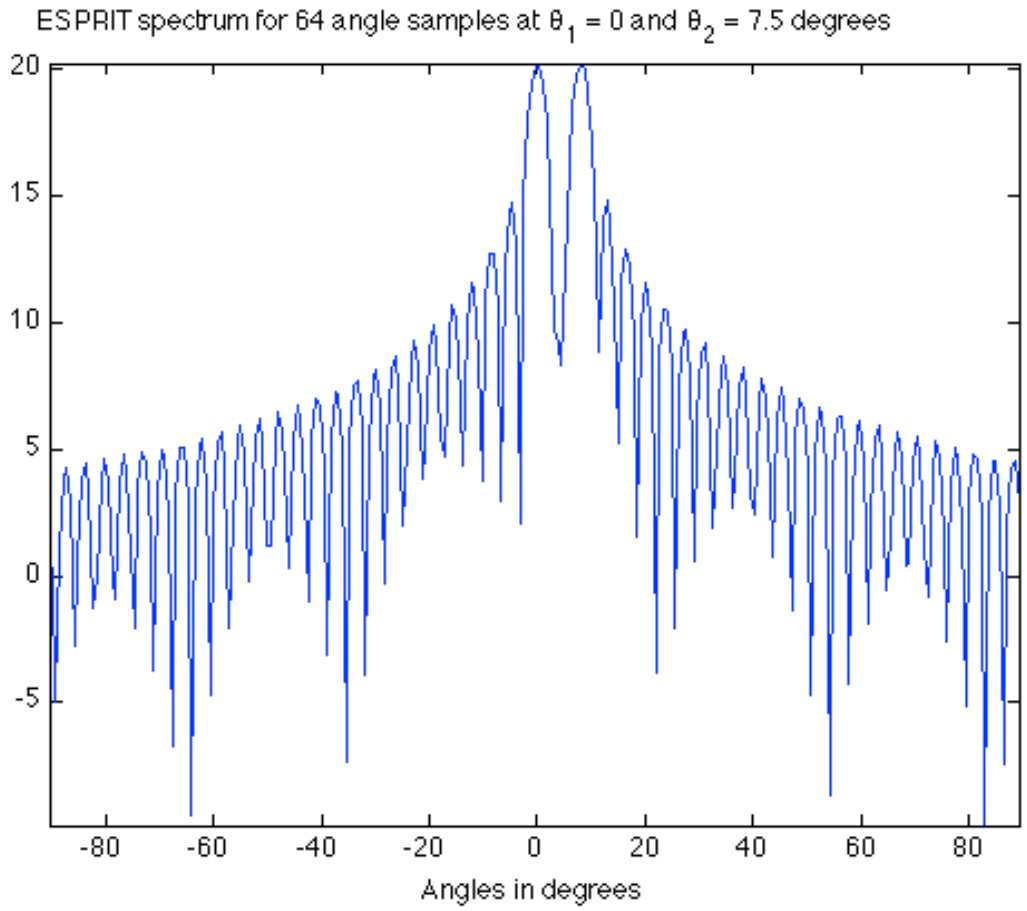


For 64 samples of angles in [-90, 90] (in degrees) and theta2 = 7.5 degrees

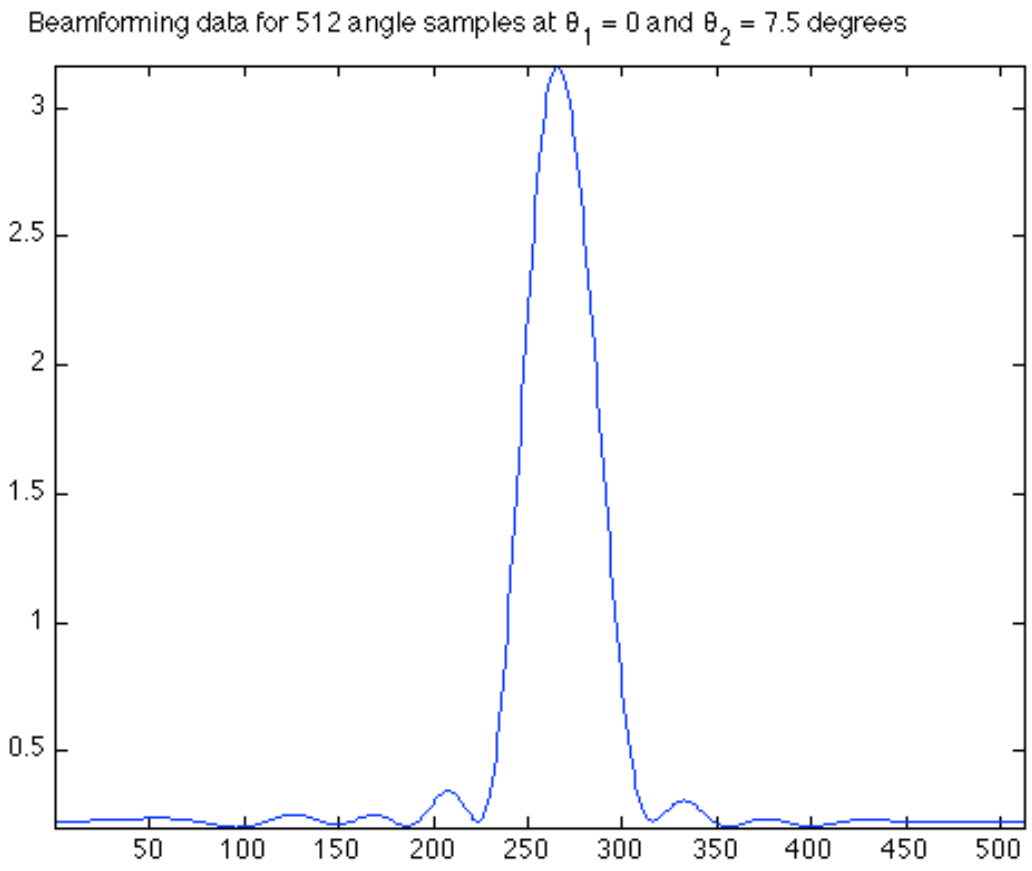

Beamforming data for 64 angle samples at θ1 = 0 and θ2 = 7.5 degrees
3
2.5
2
1.5
1
0.5
10
20
30
40
50
60

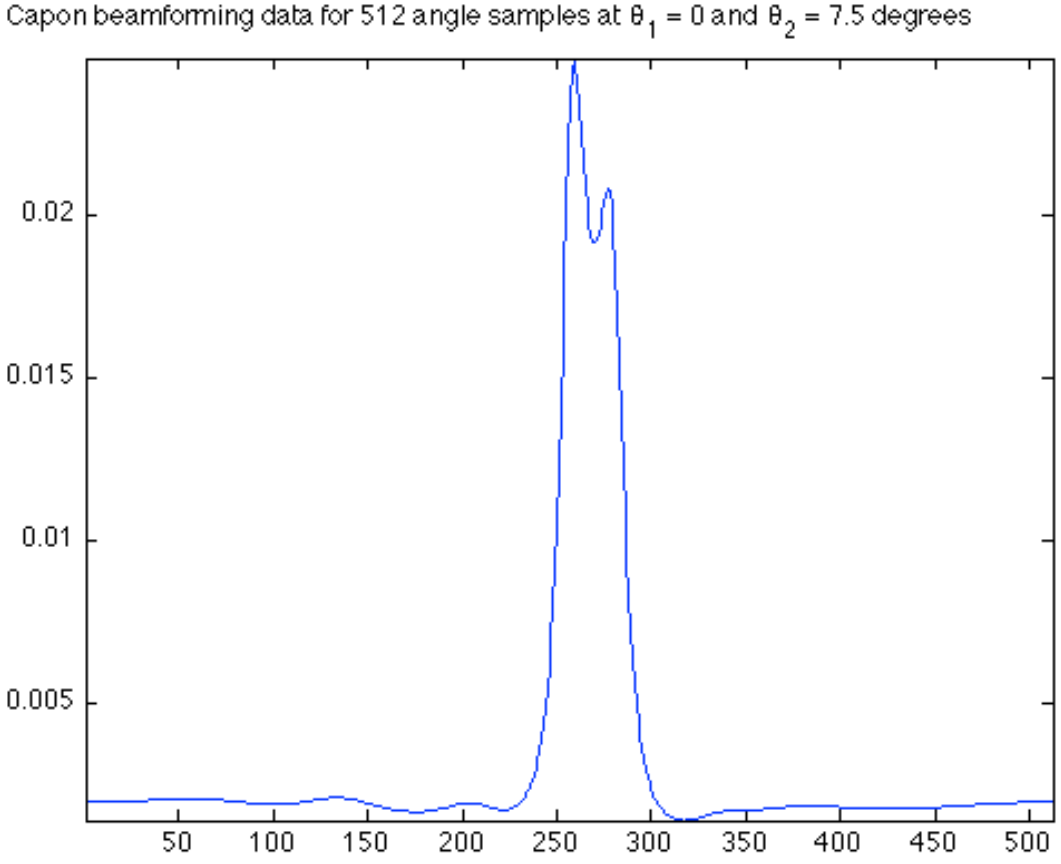

Capon beamforming data for 64 angle samples at θ1 = 0 and θ2 = 7.5 degrees
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
10
20
30
40
50
60

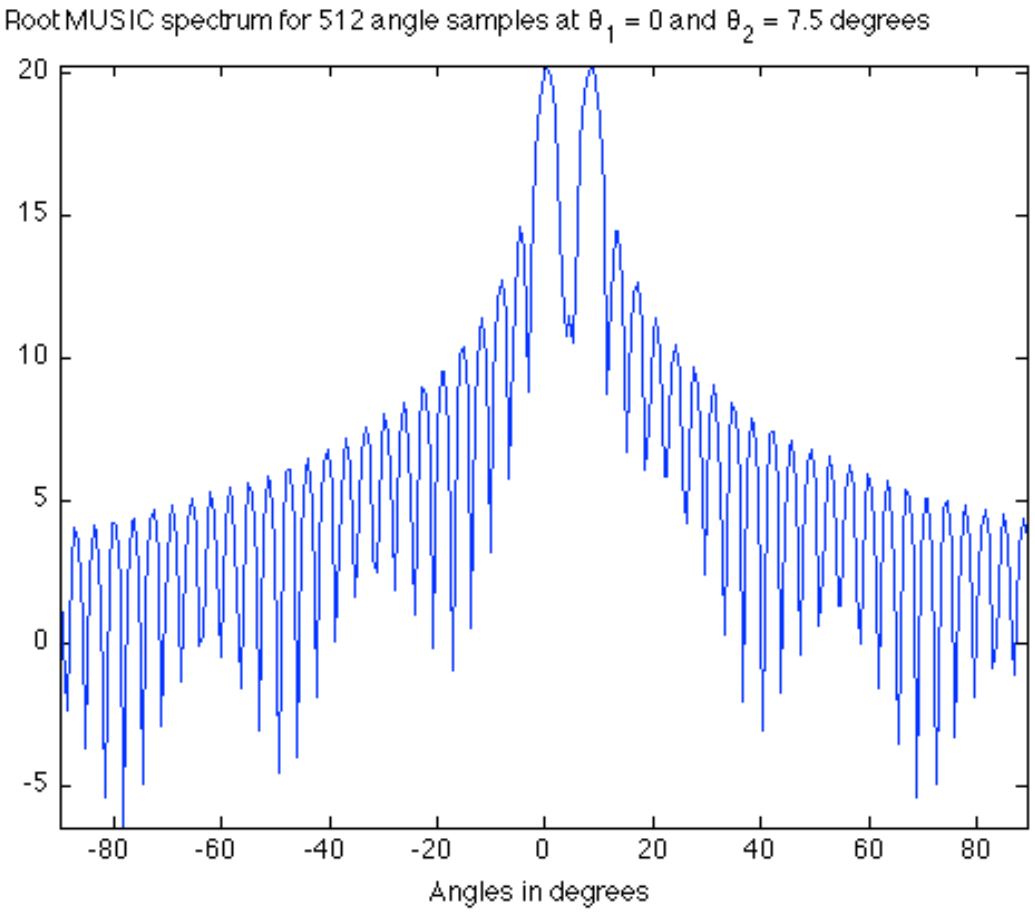

Root MUSIC spectrum for 64 angle samples at θ1 = 0 and θ2 = 7.5 degrees
20
15
10
5
0
-5
-10
-15
-80
-60
-40
-20
0
20
40
60
80
Angles in degrees

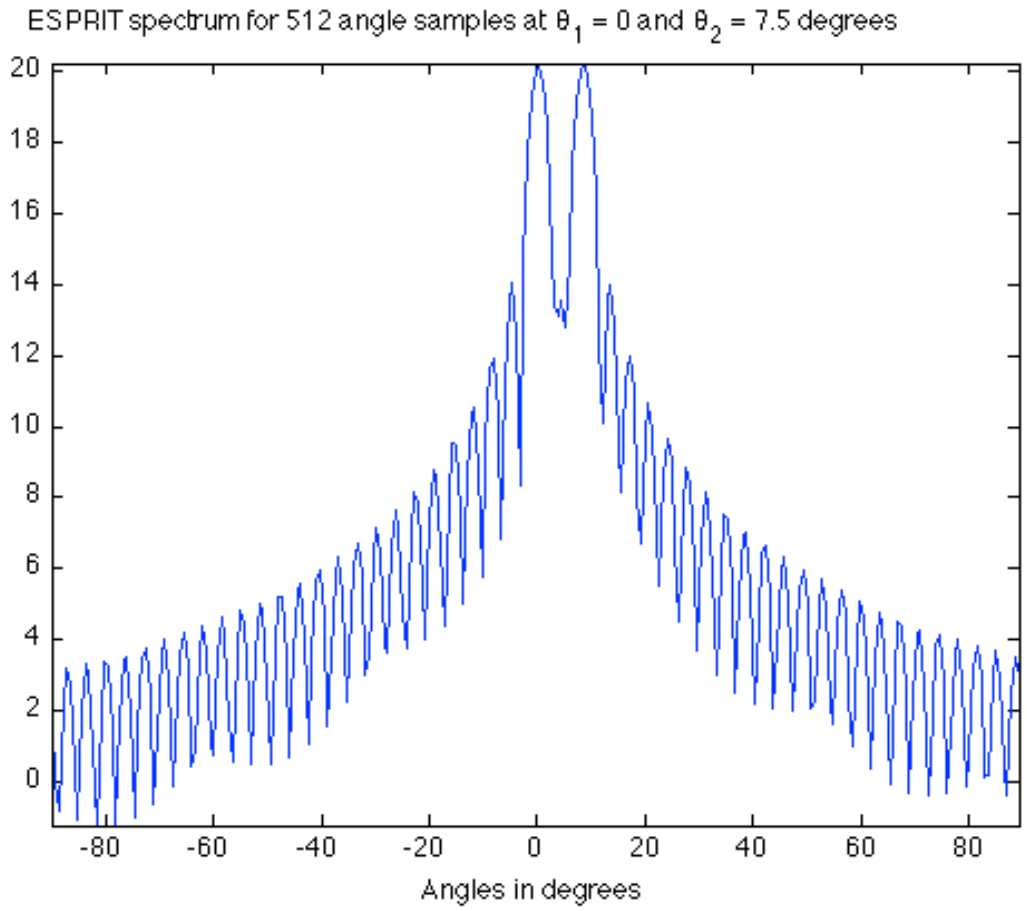


For 512 samples of angles in [-90, 90] (in degrees) and theta2 = 15 degrees


Beamforming data for 512 angle samples at θ1 = 0 and θ2 = 7.5 degrees
3
2.5
2
1.5
1
0.5
50
100
150
200
250
300
350
400
450
500


Capon beamforming data for 512 angle samples at θ1 = 0 and θ2 = 7.5 degrees
0.02
0.015
0.01
0.005
50
100
150
200
250
300
350
400
450
500


Root MUSIC spectrum for 512 angle samples at θ1 = 0 and θ2 = 7.5 degrees
20
15
10
5
0
-5
-80
-60
-40
-20
0
20
40
60
80
Angles in degrees



Discussion:

For theta2 = 15 the two spatial frequencies are resolved well for angle samples of 32 and more. '32' is not the minimum but if we keep decreasing the number of angle samples the two different sources cannot be resolved because the peaks start to merge into a single peak.

For theta2 = 7.5 the two spatial frequencies(signal sources) are not resolved by both Beamforming method and the Capon beamforming method even at 32 angle samples. But at 512 angle samples atleast the Capon beamforming method atleast begins to be able to resolve the two sources.

So Capon can be considered more powerful than the normal beamforming method. But MUSIC method is better than them both because at the above number of angle samples and theta2 values under discussion the MUSIC method is able to resolve the two sources. The ESPRIT method is even better. We know

that the ESPRIT method is better than the MUSIC method from the previous assignments.