EE 779 Advanced Topics in Signal Processing
Assignment 3 simulations

Name: Swrangsar Basumatary Roll: 09d07040

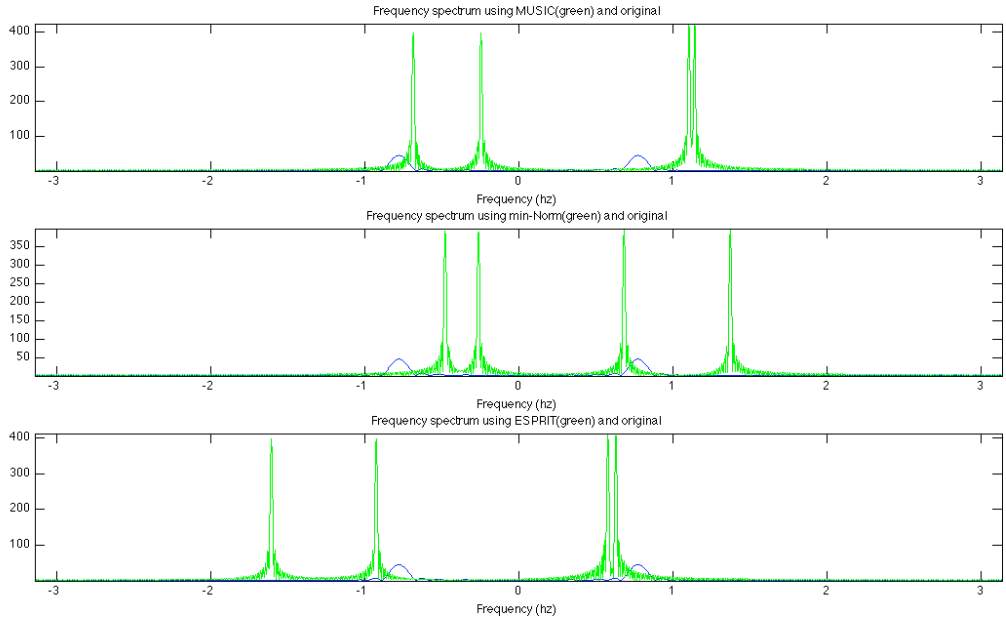
Answer to C4.12

a)
Considering a variance of value '1' for this part.

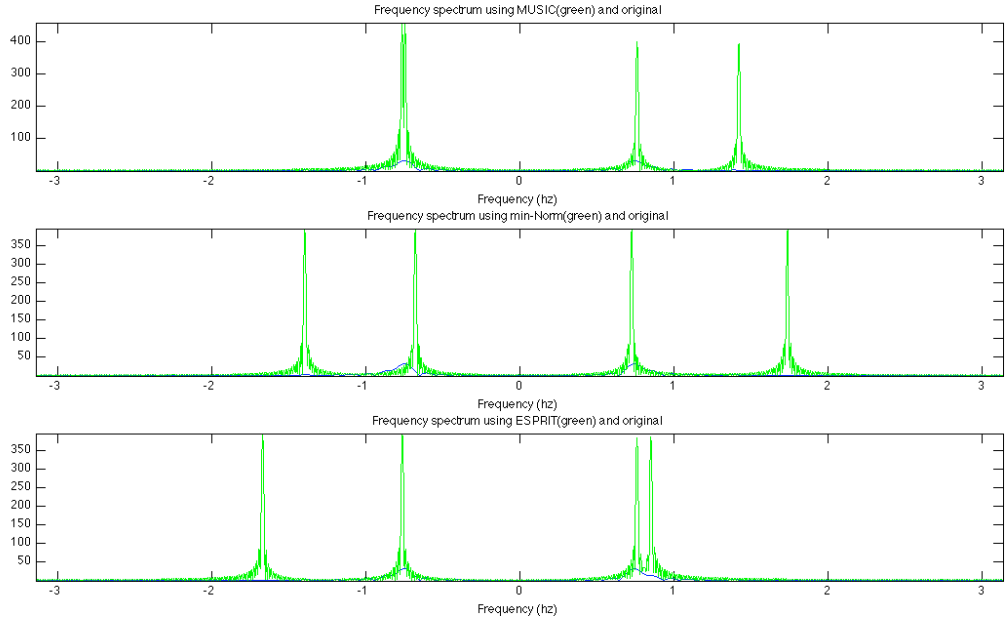
For values of 'm' from 5 to 9 all three methods i.e. MUSIC, min-norm and ESPRIT either have peaks at inaccurate values of frequencies or they are not resolvable (meaning the peaks merge into one). At m = 10 we see some change for the min-norm and the ESPRIT case, they start to show peaks at close to original frequencies while being resolvable at the same time. The ESPRIT case is more resolvable at this 'm'. At m = 12, the MUSIC method gives close to original peaks that are resolvable.

(Large size plots are included in this folder)

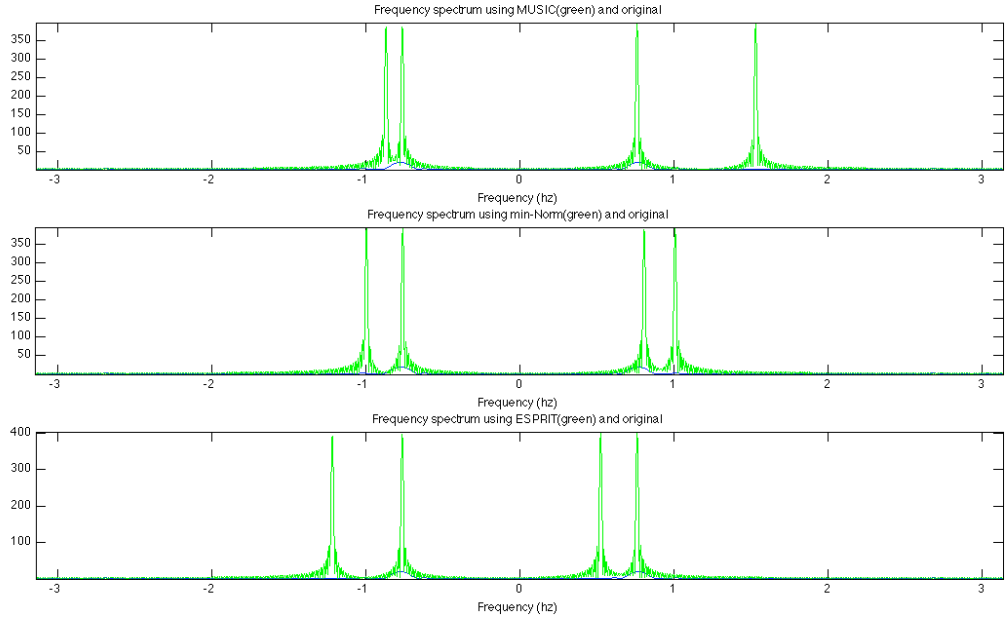
The plot for m = 5



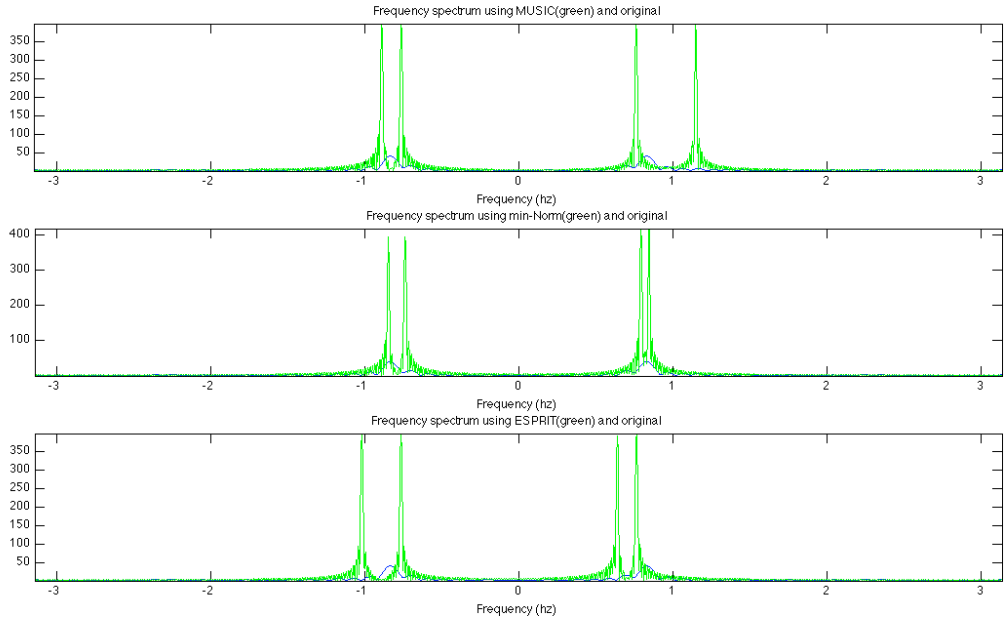
The plot for m = 7



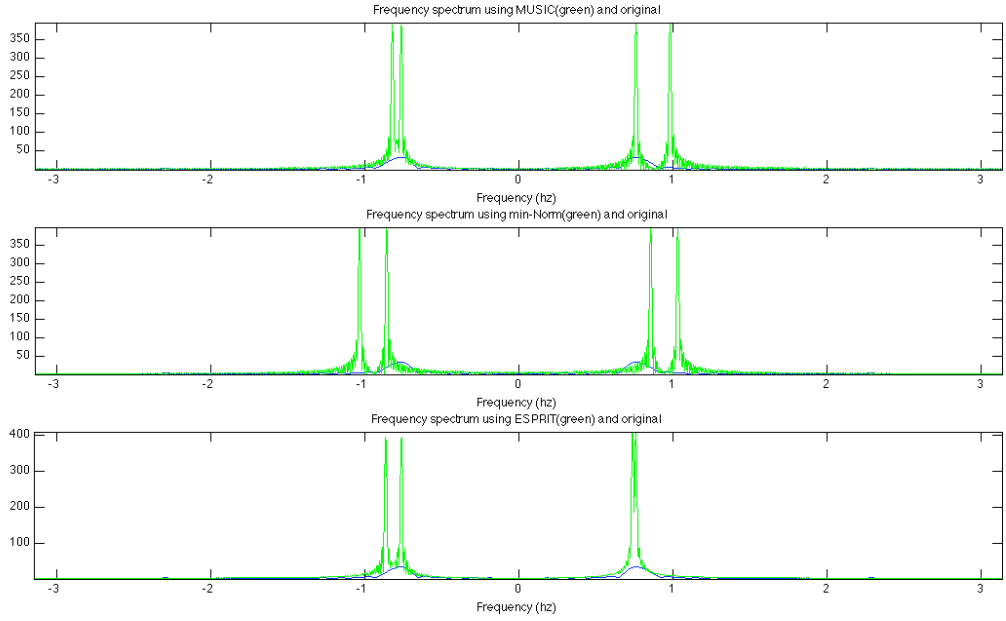
The plot for m=9



The plot for m=10



The plot for m=12

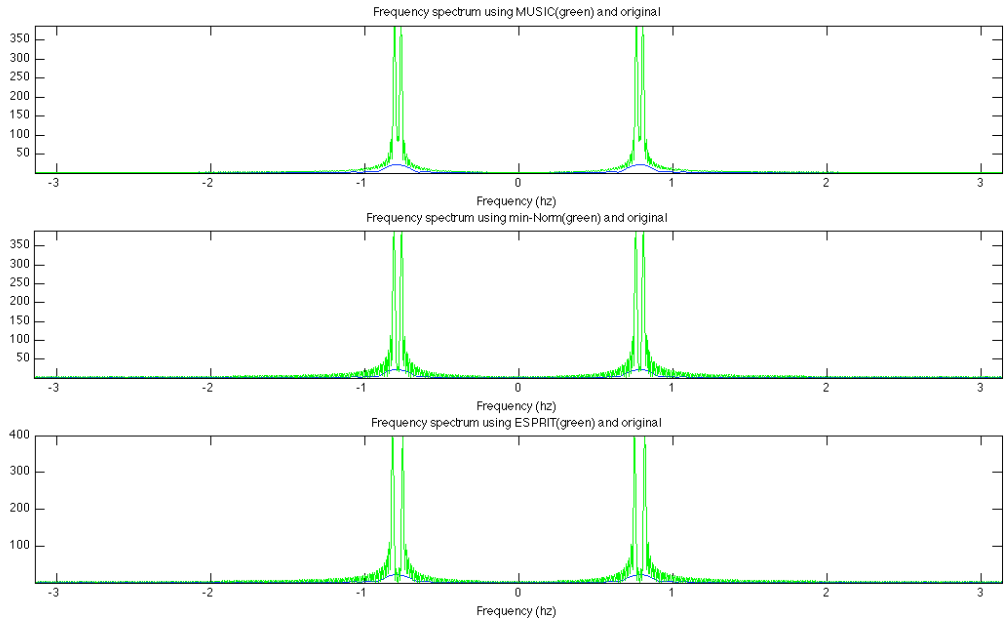


b)
This is for the infinite SNR case or the case when variance is ‘zero’.

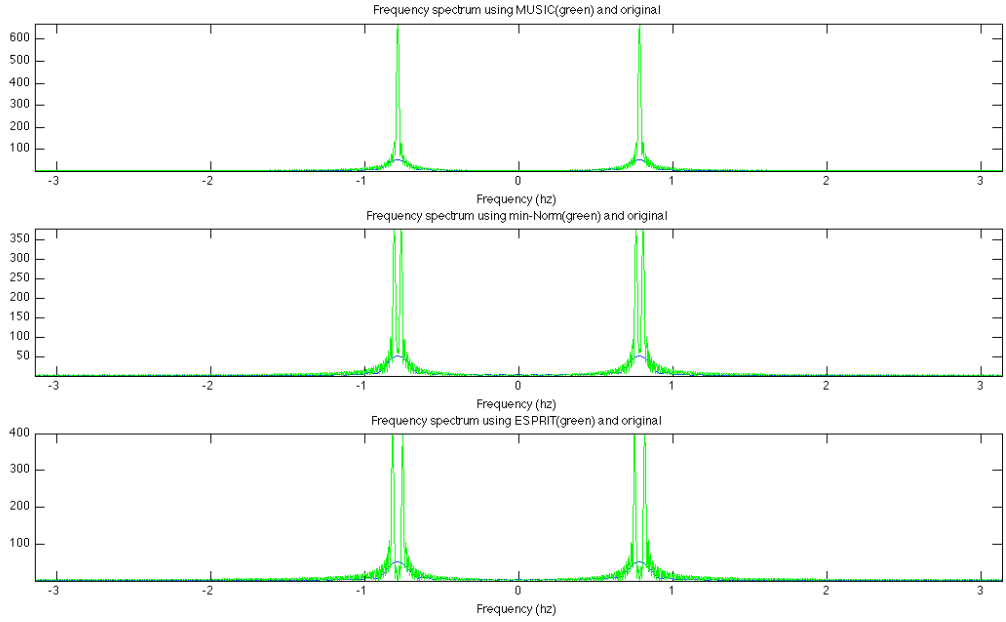
At m= 5 almost all the three methods give resolvable peaks. The MUSIC case is most weakly resolvable at this value of ‘m. At m= 8 the MUSIC case is no longer resolvable. At m = 12 even the min-norm peaks come very close but they still remain resolvable.

This happens because the MUSIC method is statistically less reliable than the min-norm and the ESPRIT method. As m increases and gets closer to ‘N’ - the number of input data samples, the errors become more pronounced. Relatively low power components get neglected.

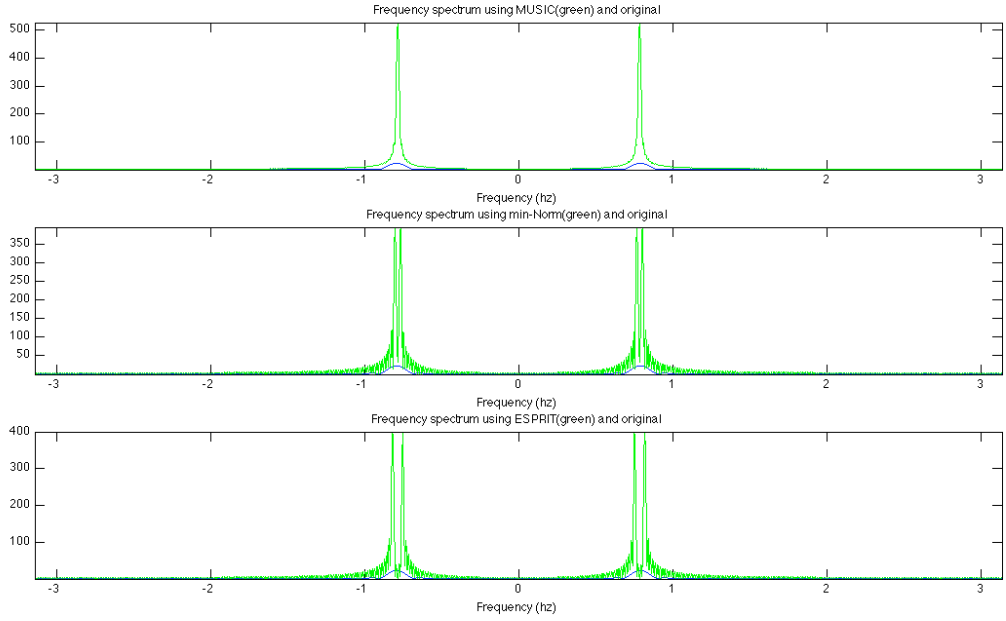
Plot for m = 5



Plot for m = 8



Plot for m = 12

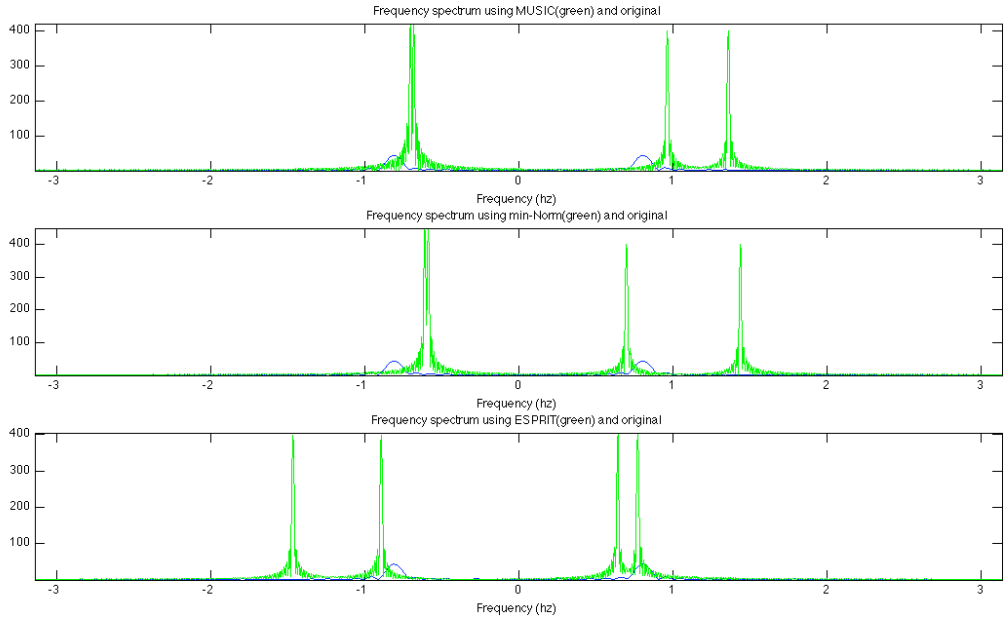


c)

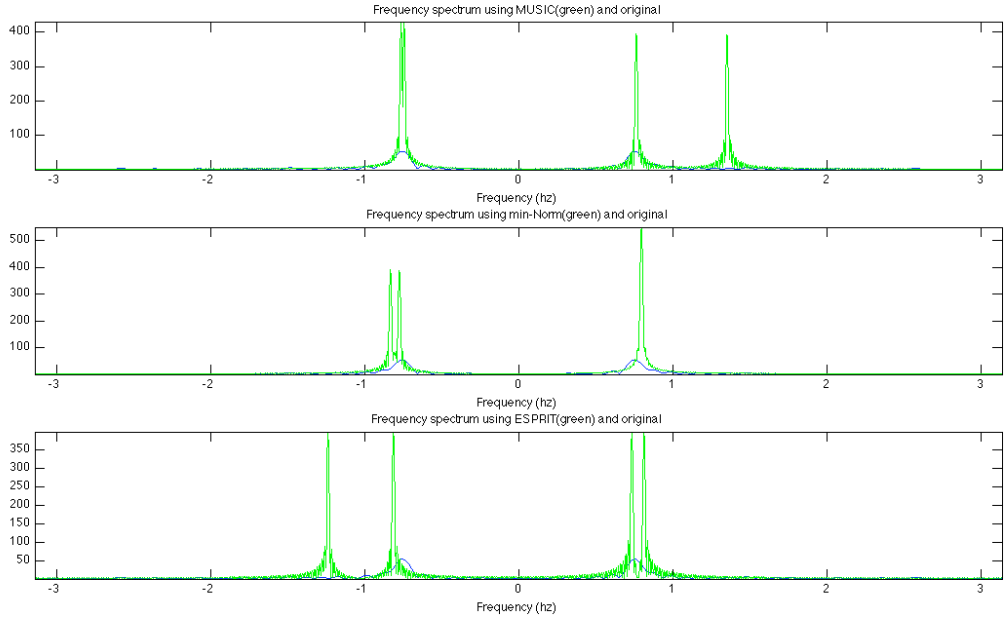
For m = 5 and 8 most of the peaks are not aligned with the original ones. And for m = 12, the peaks of the MUSIC method are not symmetrical about the 0 Hz frequency. The estimates are also not that accurate. In the min-Norm case the peaks are cleanly resolvable but they also do not align with the original completely. The peaks of the ESPRIT case align quite well, are symmetrical about '0' frequency and resolvable.

The methods in decreasing order of reliability is therefore:
ESPRIT > min-Norm > MUSIC

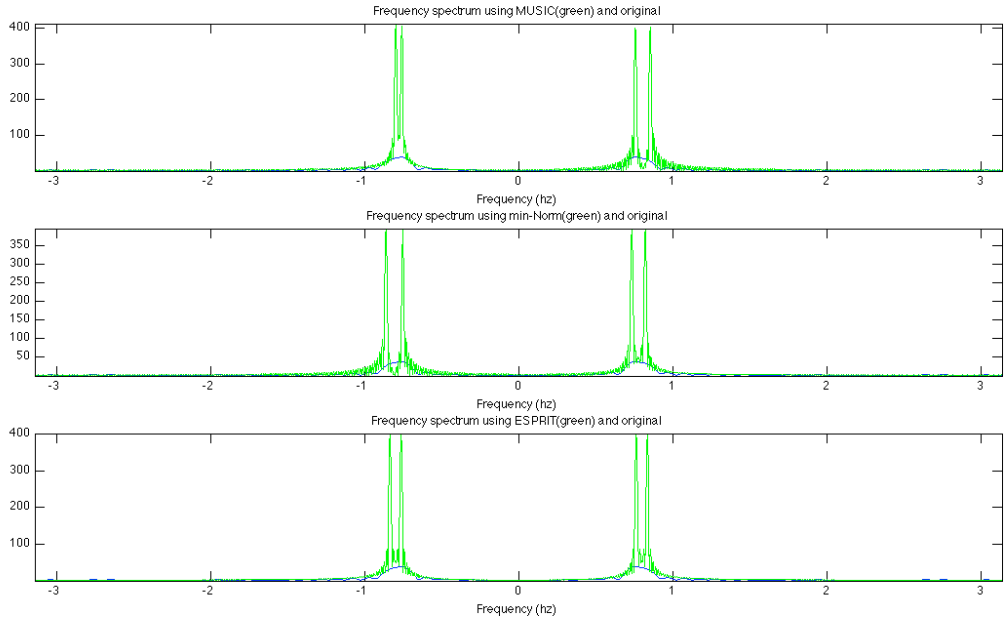
Plot for m = 5 and var = 1



Plot for m = 8 and var = 1



Plot for m = 12 and var = 1



d) The major difference between exercise C3.18 and this problem is that in here we have got impulses at two distinct frequencies whereas in C3.18 though we have got peaks but they are smooth peaks. For this problem it is better to use linear spectral methods such as MUSIC, min-Norm, ESPRIT, etc. rather than AR, ARMA, etc. because linear spectral methods are good at capturing peaks that are impulse-like in the frequency spectrum.

The codes used in this problem are:

Functions:

```
function [dataSpectrum, musicSpectrum, minnormSpectrum,
espritSpectrum] = frequencyEstimateOfLineSpectra(modelOrder,
totalOrder, variance)

% get all spectra of the freq estimates

numberOfRealizations = 50;
realizations = getMonteCarloRealizations(variance,
numberOfRealizations);

musicEstimates = getMusicFrequencyEstimates(realizations,
modelOrder, totalOrder);
musicSpectrum = getSpectrum(musicEstimates);
minnormEstimates = getMinNormFrequencyEstimates(realizations,
modelOrder, totalOrder);
```

```
minnormSpectrum = getSpectrum(minnormEstimates);
espritEstimates = getESPRITFrequencyEstimates(realizations,
modelOrder, totalOrder);
espritSpectrum = getSpectrum(espritEstimates);

dataSpectrum = getDataSpectrum(realizations);

end


%% generate random number in [-pi, pi] with uniform distribution

function phaseMatrix = getPhaseMatrix(numberOfRealizations)

phaseMatrix = -pi + ((2 * pi) .* rand(numberOfRealizations, 2));

end


%% generate a single realization of the sinusoidal signal

function data = generateSingleRealization(phase1, phase2,
variance)

N = 64; % as given in question
t = 1:N;
y = (10 * sin((0.24 * pi * t) + phase1)) + (5 * sin((0.26 * pi *
t) + phase2));
whiteNoise = wgn(1, N, variance, 'linear');
y = y + whiteNoise;
data = y(:);

end

%% generate 50 monte carlo realizations

function realizations = getMonteCarloRealizations(variance,
numberOfRealizations)

realizations = cell(numberOfRealizations, 1);
phaseMatrix = getPhaseMatrix(numberOfRealizations);

for k = 1:numberOfRealizations
    realizations{k} = generateSingleRealization(phaseMatrix(k,
1), phaseMatrix(k, 2), variance);
end
```

```
end

%% get the average frequency estimates of the monte carlo
realizations using Music

function frequencyEstimatesMusic =
getMusicFrequencyEstimates(realizations, modelOrder, totalOrder)

frequencyEstimates = cell(length(realizations), 1);

for k = 1:length(realizations)
    frequencyEstimates{k} = music(realizations{k}, modelOrder,
totalOrder);
end

frequencyEstimatesMusic = zeros(size(frequencyEstimates{1}));
for k = 1:length(frequencyEstimates)
    frequencyEstimatesMusic = frequencyEstimatesMusic +
frequencyEstimates{k};
end
frequencyEstimatesMusic = frequencyEstimatesMusic ./
length(frequencyEstimates);

end


%% get the average frequency estimates of the monte carlo
realizations using min-Norm

function frequencyEstimatesMinNorm =
getMinNormFrequencyEstimates(realizations, modelOrder,
totalOrder)

frequencyEstimates = cell(length(realizations), 1);

for k = 1:length(realizations)
    frequencyEstimates{k} = minnorm(realizations{k}, modelOrder,
totalOrder);
end

frequencyEstimatesMinNorm = zeros(size(frequencyEstimates{1}));
for k = 1:length(frequencyEstimates)
    frequencyEstimatesMinNorm = frequencyEstimatesMinNorm +
frequencyEstimates{k};
end
```

```
frequencyEstimatesMinNorm = frequencyEstimatesMinNorm ./
length(frequencyEstimates);

end


%% get the average frequency estimates of the monte carlo
realizations using ESPRIT

function frequencyEstimatesESPRIT =
getESPRITFrequencyEstimates(realizations, modelOrder,
totalOrder)

frequencyEstimates = cell(length(realizations), 1);

for k = 1:length(realizations)
    frequencyEstimates{k} = esprit(realizations{k}, modelOrder,
totalOrder);
end

frequencyEstimatesESPRIT = zeros(size(frequencyEstimates{1}));
for k = 1:length(frequencyEstimates)
    frequencyEstimatesESPRIT = frequencyEstimatesESPRIT +
frequencyEstimates{k};
end
frequencyEstimatesESPRIT = frequencyEstimatesESPRIT ./
length(frequencyEstimates);

end

%% get the spectrum of given frequencies

function estimatedSpectrum = getSpectrum(frequencyEstimates)

N = 400; % just take a large enough value of samples
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 = abs(fftshift(fft(x, M)));
```

```
end

%% get the spectrum of the generated input data

function dataSpectrum = getDataSpectrum(realizations)

avgRealization = zeros(size(realizations{1}));
for k = 1:length(realizations)
    avgRealization = avgRealization + realizations{k};
end
avgRealization = avgRealization ./ length(realizations);

M = 2 ^ nextpow2(4 * 400); % to match the estimated spectra's
length. here 400 is a magic number.
dataSpectrum = abs(fftshift(fft(avgRealization, M)));

end
```

The script:

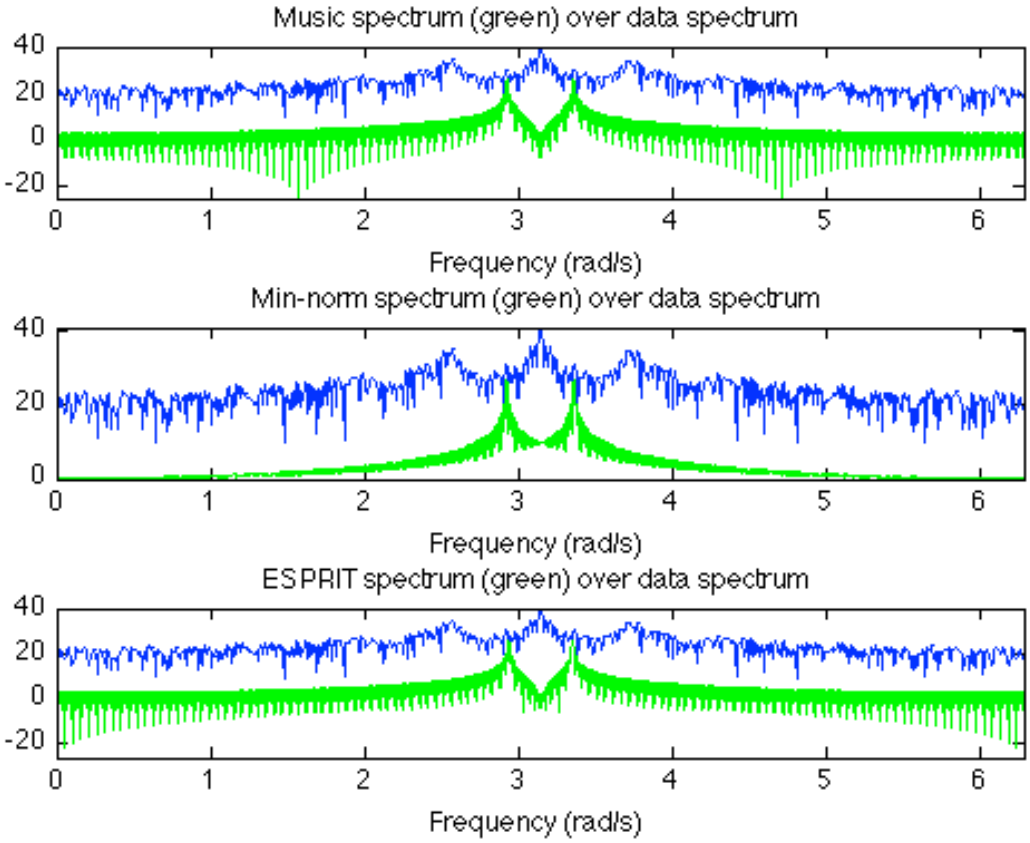
```
close all; clear all;

addpath /Users/swrangsarbasumatary/Desktop/
advancedSignalProcessingAssignment3/ch4/

[dataSpectrum1, musicSpectrum1, minnormSpectrum1,
espritSpectrum1] = frequencyEstimateOfLineSpectra(4, 12, 1);

w = -(length(musicSpectrum1)/2):(length(musicSpectrum1)/2-1);
w = 2 * pi * (w/length(musicSpectrum1));

figure(100); clf;
subplot(3, 1, 1);
plot(w, dataSpectrum1);
hold on;
musicplot = plot(w, musicSpectrum1);
hold off;
axis tight;
title('Frequency spectrum using MUSIC(green) and original');
xlabel('Frequency (hz)');
set(musicplot, 'Color', 'green');

subplot(3, 1, 2);
plot(w, dataSpectrum1);
```

```
hold on;
minnormplot = plot(w, minnormSpectrum1);
hold off;
axis tight;
title('Frequency spectrum using min-Norm(green) and original');
xlabel('Frequency (hz)');
set(minnormplot, 'Color', 'green');

subplot(3, 1, 3);
plot(w, dataSpectrum1);
hold on;
espritplot = plot(w, espritSpectrum1);
hold off;
axis tight;
title('Frequency spectrum using ESPRIT(green) and original');
xlabel('Frequency (hz)');
set(espritplot, 'Color', 'green');


rmpath /Users/swrangsarbasumatary/Desktop/
advancedSignalProcessingAssignment3/ch4/
```

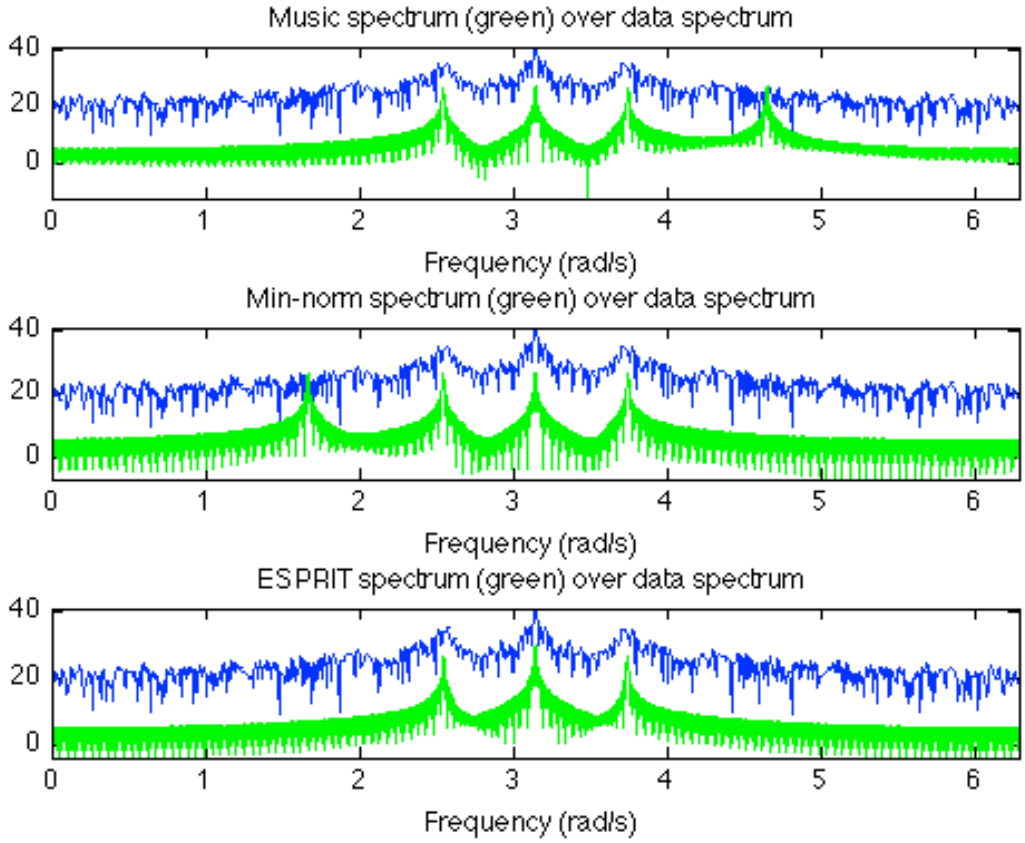
Answer to C4.14

Here we plot the spectrum of the original data (blue) and the estimated spectrum generated from the frequency estimates (green) and compare. For p = 4 (model order) most of the spectral shape coincide. In case of the loglynx data the spectra are very close even in magnitude, means there is very little difference in scale.

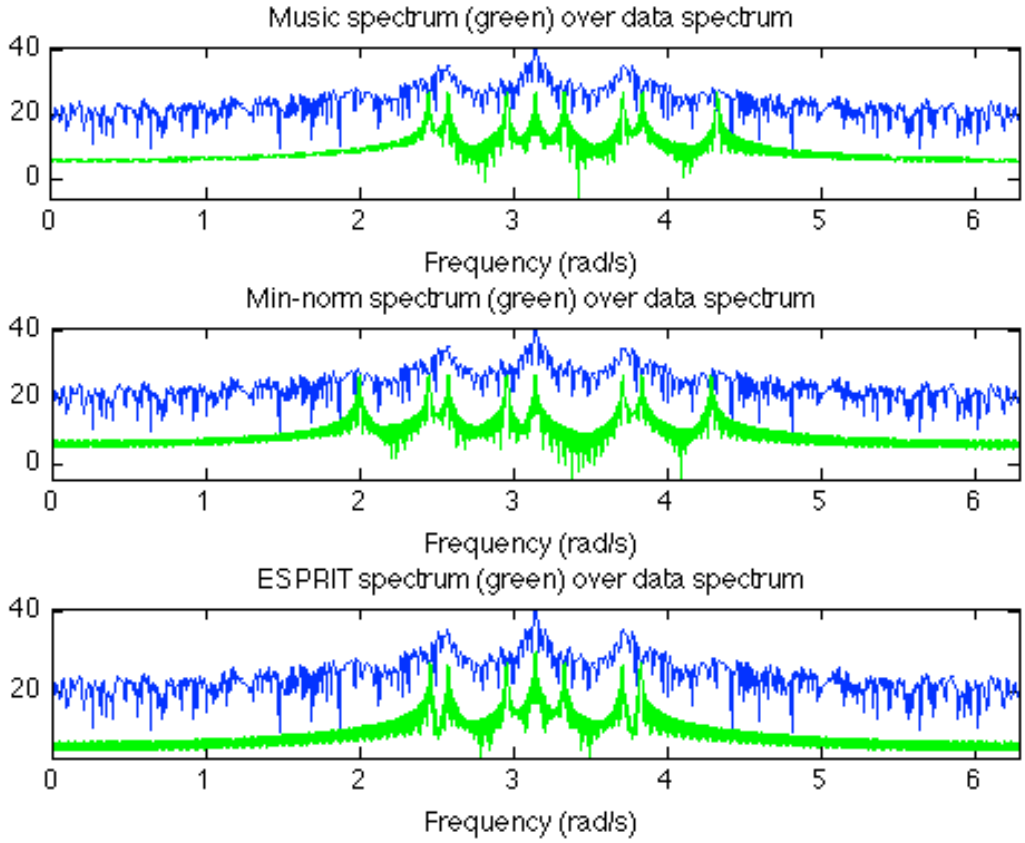
The sunspot spectrum and data spectrum for p = 2:



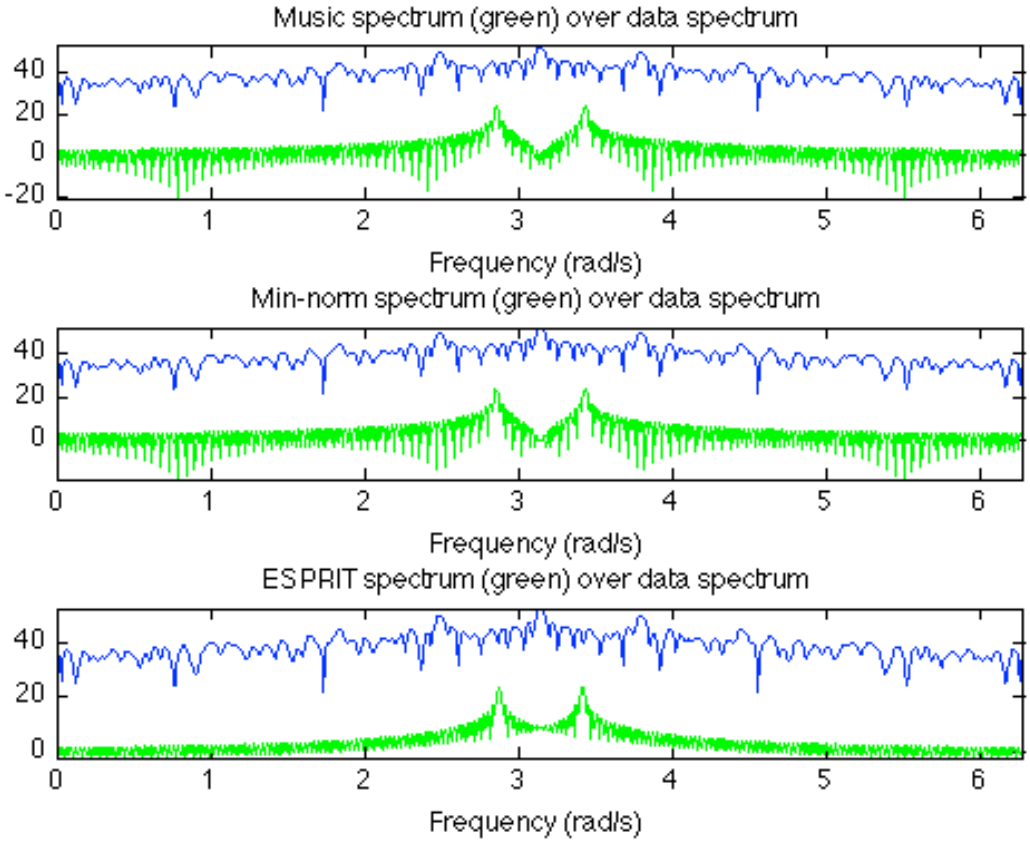
The sunspot spectrum and data spectrum for p = 4:



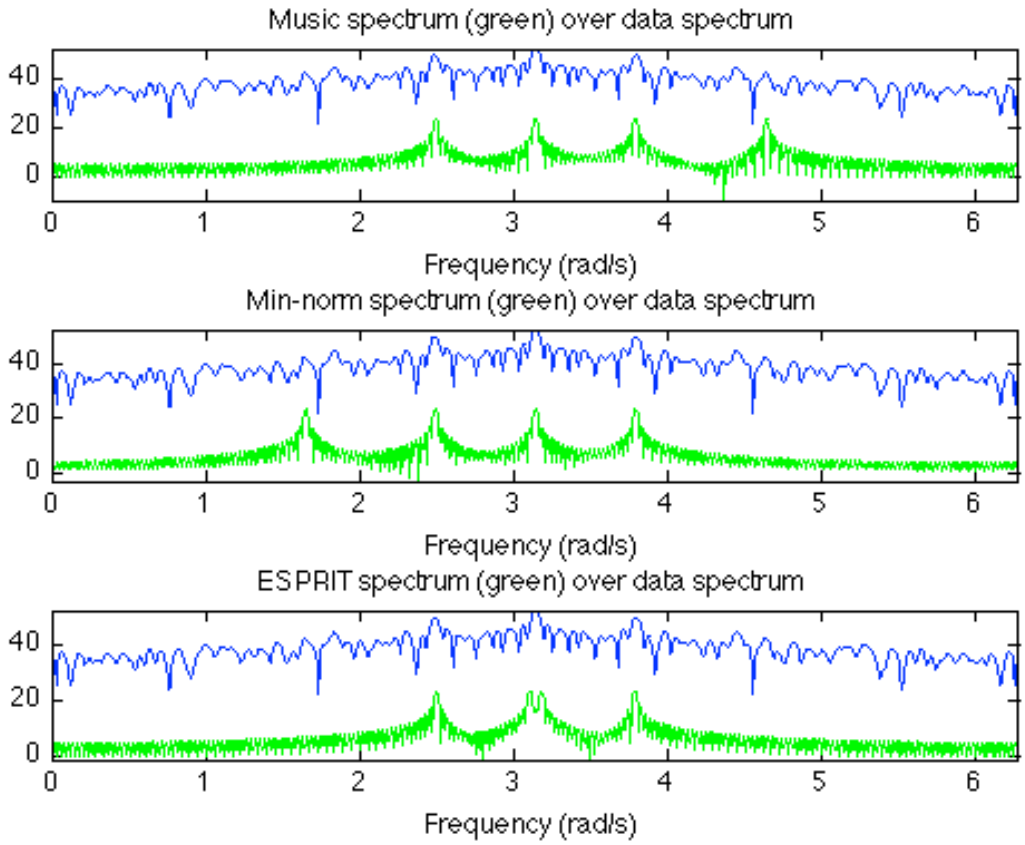
The sunspot spectrum and data spectrum for p = 8:



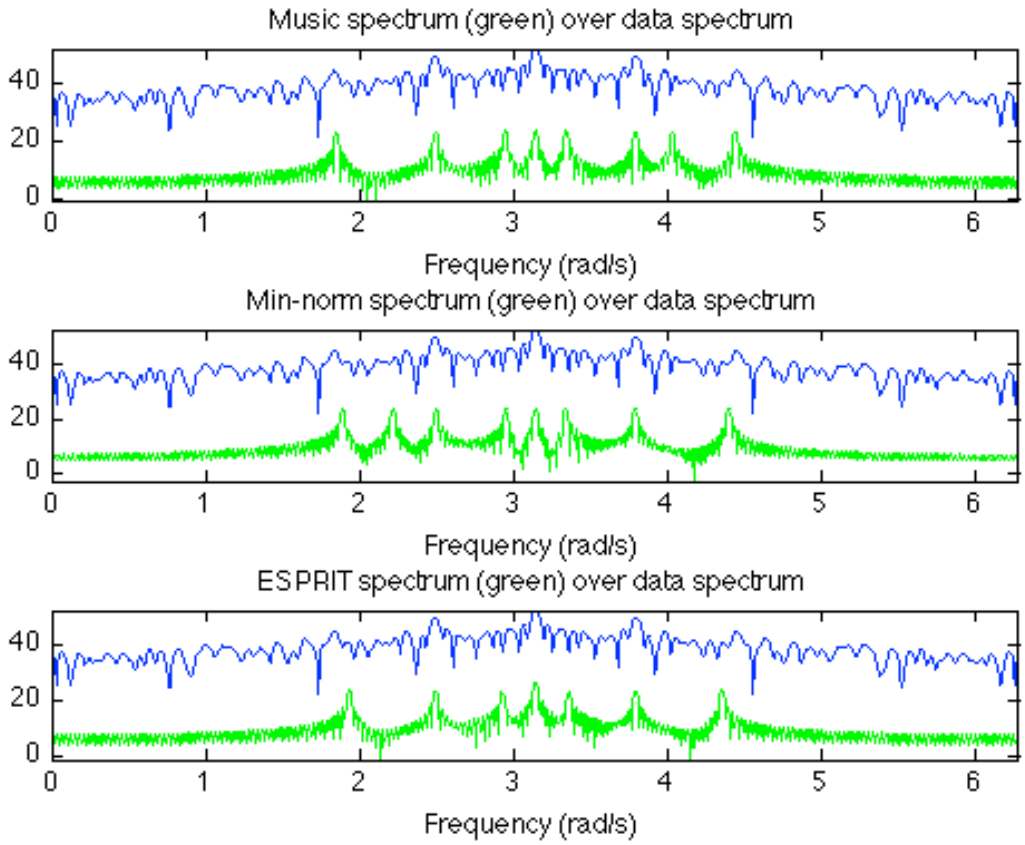
The lynx spectrum and data spectrum for p = 2:



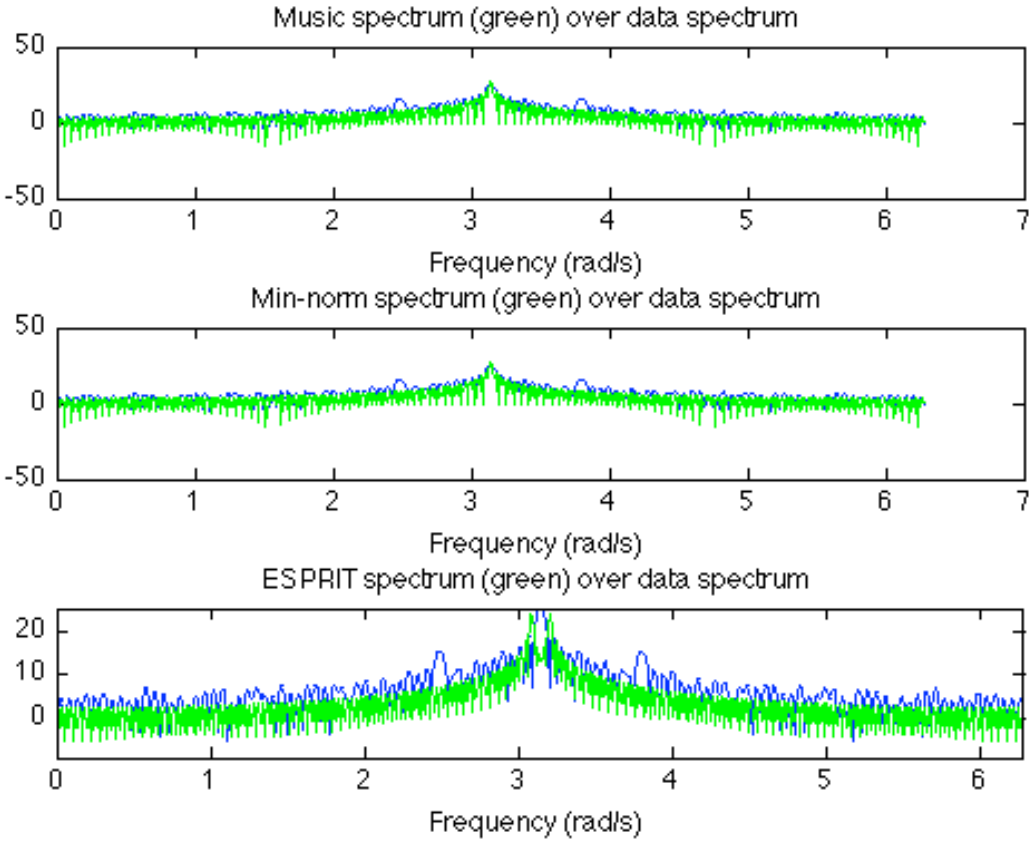
The lynx spectrum and data spectrum for p = 4:



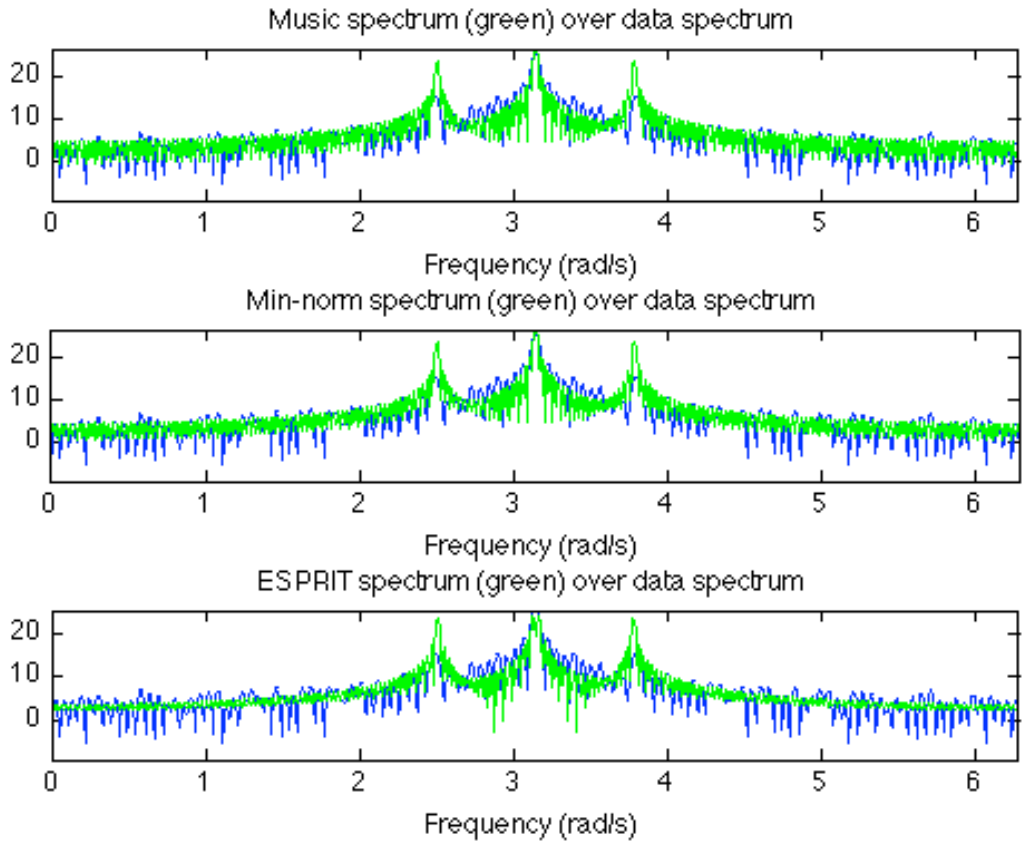
The lynx spectrum and data spectrum for p = 8:



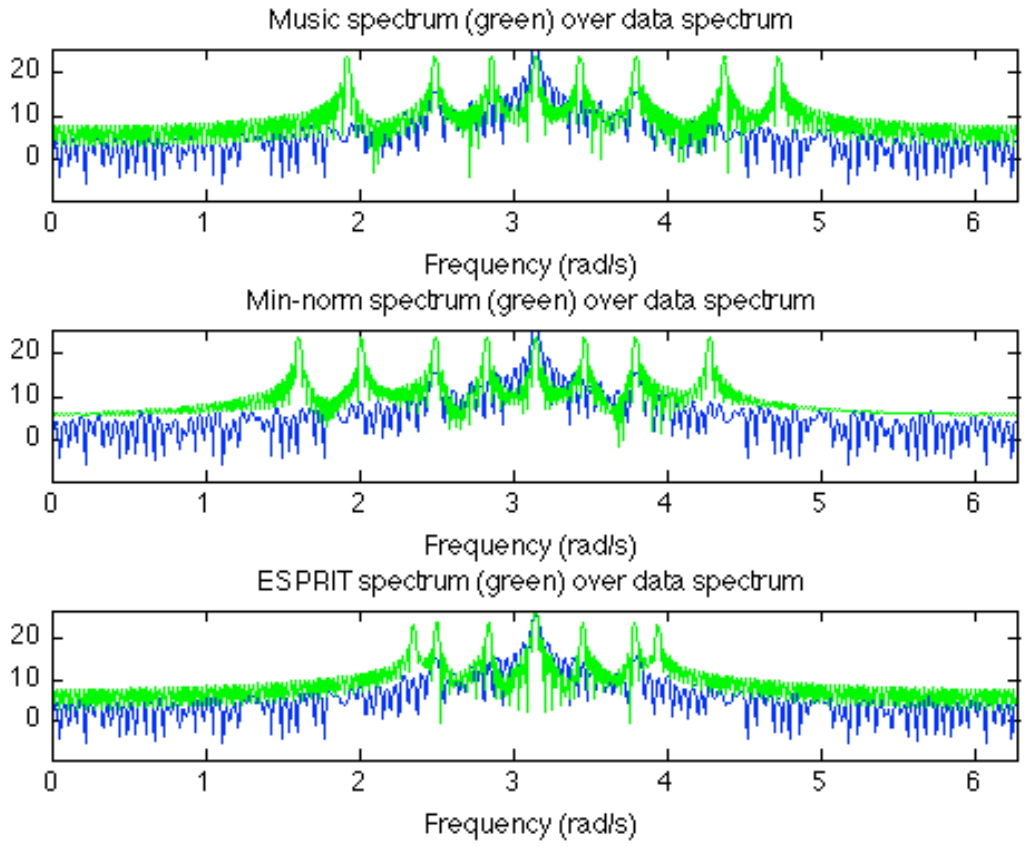
The loglynx spectrum and data spectrum for p = 2:



The loglynx spectrum and data spectrum for p = 4:



The loglynx spectrum and data spectrum for p = 8:



The functions used are:

```
function lineSpectralFrequencyEstimator(data, modelOrder)

totalOrder = 3 * modelOrder;
count = 2 * length(data);
[musicSpectrum, minnormSpectrum, espritSpectrum] = ...
    getFrequencySpectra(data, modelOrder, totalOrder, count);
dataSpectrum = getDataSpectrum(data, count);
subplotDouble(dataSpectrum, musicSpectrum, minnormSpectrum,
espritSpectrum);


end

%% get the frequency spectra using MUSIC, min-Norm and ESPRIT
methods

function [musicSpectrum, minnormSpectrum, espritSpectrum] ...
    = getFrequencySpectra(data, modelOrder, totalOrder, count)
```

```
frequenciesMusic = music(data, modelOrder, totalOrder);
frequenciesMinNorm = minnorm(data, modelOrder, totalOrder);
frequenciesEsprit = esprit(data, modelOrder, totalOrder);

musicSpectrum = getSpectrum(frequenciesMusic, count);
minnormSpectrum = getSpectrum(frequenciesMinNorm, count);
espritSpectrum = getSpectrum(frequenciesEsprit, count);

end

%% get the spectrum of given frequencies

function estimatedSpectrum = getSpectrum(frequencyEstimates, N)

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))));

end

%% get the spectrum of the input data

function dataSpectrum = getDataSpectrum(data, N)

M = 2 ^ nextpow2(4 * N);
dataSpectrum = 10 * log10(abs(fftshift(fft(data, M))));

end

%% superpose two subplots

function subplotDouble(dataSpectrum, musicSpectrum,
minnormSpectrum, espritSpectrum)

w = 0:length(dataSpectrum)-1;
w = 2 * pi * w ./ length(dataSpectrum);
```

```
figure;
subplot(3, 1, 1);
plot(w, dataSpectrum);
hold on;
p = plot(w, musicSpectrum);
hold off;
axis tight;
title('Music spectrum (green) over data spectrum');
xlabel('Frequency (rad/s)');
set(p, 'Color', 'green');

subplot(3, 1, 2);
plot(w, dataSpectrum);
hold on;
p = plot(w, minnormSpectrum);
hold off;
axis tight;
title('Min-norm spectrum (green) over data spectrum');
xlabel('Frequency (rad/s)');
set(p, 'Color', 'green');

subplot(3, 1, 3);
plot(w, dataSpectrum);
hold on;
p = plot(w, espritSpectrum);
hold off;
axis tight;
title('ESPRIT spectrum (green) over data spectrum');
xlabel('Frequency (rad/s)');
set(p, 'Color', 'green');

end
```

The script:

```
close all; clear all;

addpath /Users/swrangsarbasumatary/Desktop/
advancedSignalProcessingAssignment3/ch4/
addpath /Users/swrangsarbasumatary/Desktop/
advancedSignalProcessingAssignment3/data/

load sunspotdata
```

```
lineSpectralFrequencyEstimator(sunspot, 2);
lineSpectralFrequencyEstimator(sunspot, 4);
lineSpectralFrequencyEstimator(sunspot, 8);

clear all;


%% plot the loglynx data

load lynxdata

lineSpectralFrequencyEstimator(lynx, 2);
lineSpectralFrequencyEstimator(lynx, 4);
lineSpectralFrequencyEstimator(lynx, 8);

lineSpectralFrequencyEstimator(loglynx, 2);
lineSpectralFrequencyEstimator(loglynx, 4);
lineSpectralFrequencyEstimator(loglynx, 8);

rmpath /Users/swrangsarbasumatary/Desktop/
advancedSignalProcessingAssignment3/data/
rmpath /Users/swrangsarbasumatary/Desktop/
advancedSignalProcessingAssignment3/ch4/
```