EE 679 Computing Assignment 3

Name: Swrangsar Basumatary Roll: 09d07040

Question 1

Finding the narrowband spectrum of four syllables (pre-emphasized):

The functions:

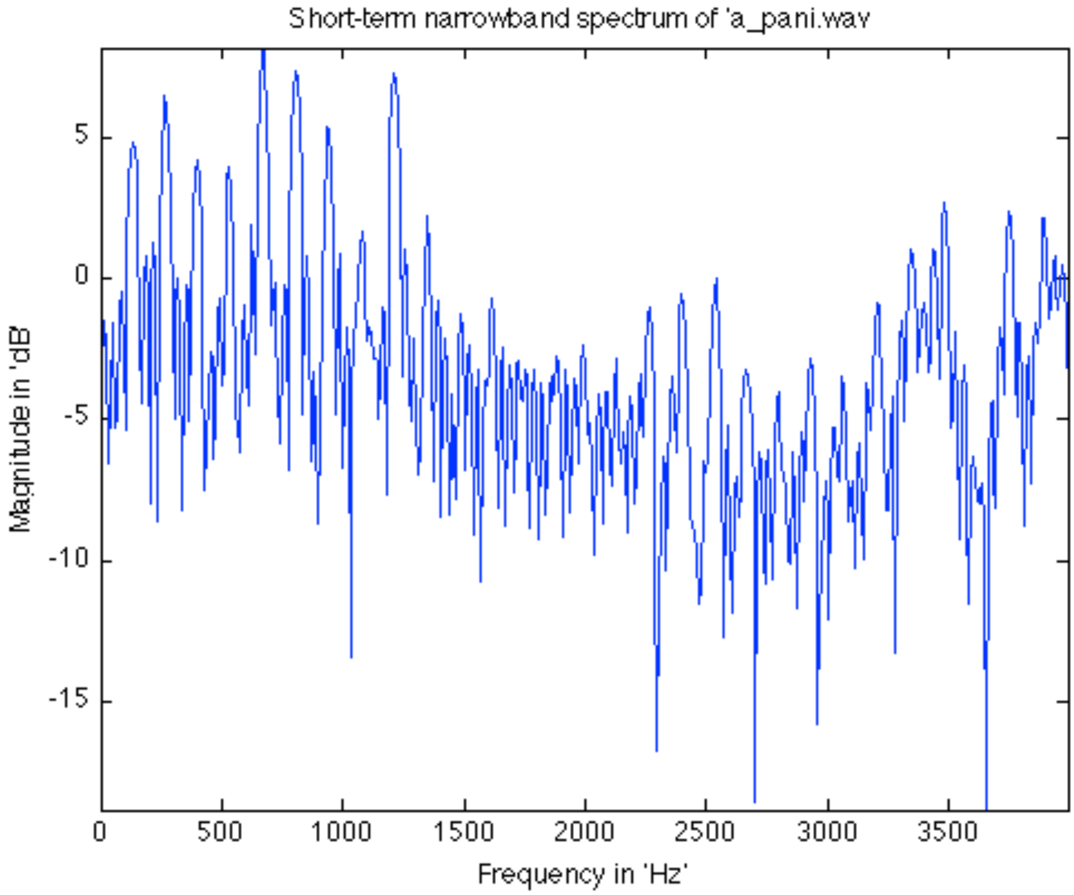
```
function getNarrowbandSpectrum(inputFile)

[windowedSignal, fs] = getWindowedSignal(inputFile);
N = 2 ^ nextpow2(length(windowedSignal) * 4);

magnitudeSpectrum = 10* log10(abs(fft(windowedSignal, N)));
frequencyFactor = fs/length(magnitudeSpectrum);
magnitudeSpectrum =
magnitudeSpectrum(1:round(length(magnitudeSpectrum)/2));
w = (0:length(magnitudeSpectrum)-1) * frequencyFactor;

figure; plot(w, magnitudeSpectrum); axis tight;
title(['Short-term narrowband spectrum of ''', inputFile,
''''], ...
    'interpreter', 'none');
xlabel('Frequency in ''Hz''');
ylabel('Magnitude in ''dB''');

end


%% get hamming windowed central part of a signal

function [windowedSignal, fs] = getWindowedSignal(inputFile)

% inputFile = 'a_pani.wav';
windowDuration = 0.030; % in ms
[y, fs] = preEmphasize(inputFile);
siz = size(y);
length = siz(1);
centralIndex = round(length/2);
M = round(windowDuration * fs);
```

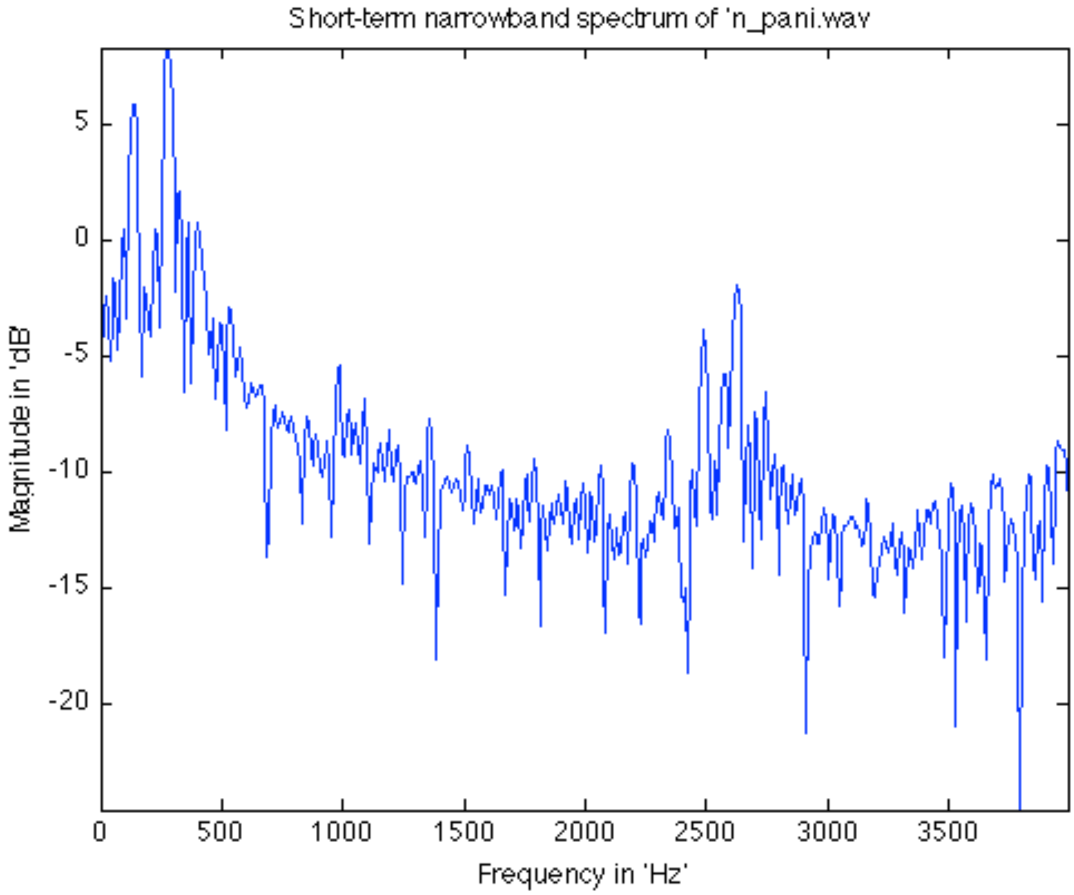
```
startIndex = round(centralIndex - M/2);
windowedSignal = y(startIndex:startIndex + M-1);
end
```

The script:

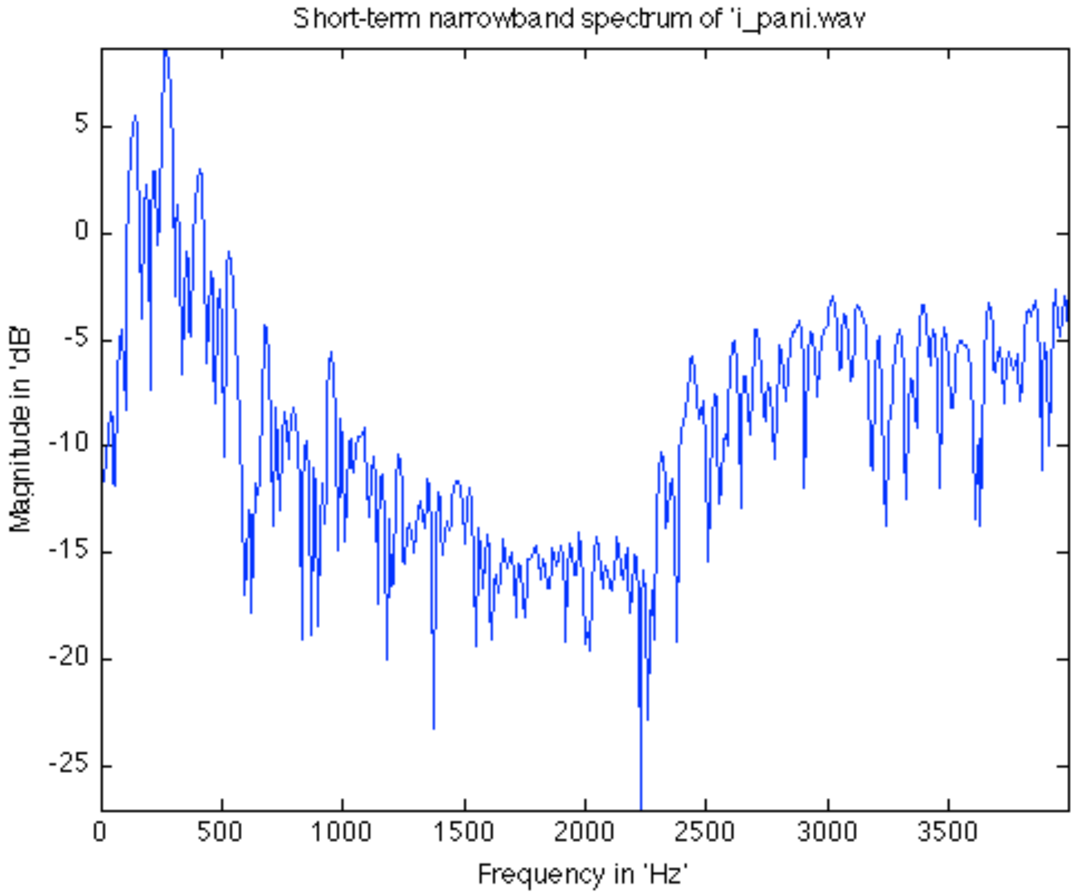
```
close all; clear all;

addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/

getNarrowbandSpectrum('a_pani.wav');
getNarrowbandSpectrum('n_pani.wav');
getNarrowbandSpectrum('i_pani.wav');
getNarrowbandSpectrum('s_uska.wav');

rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/
```

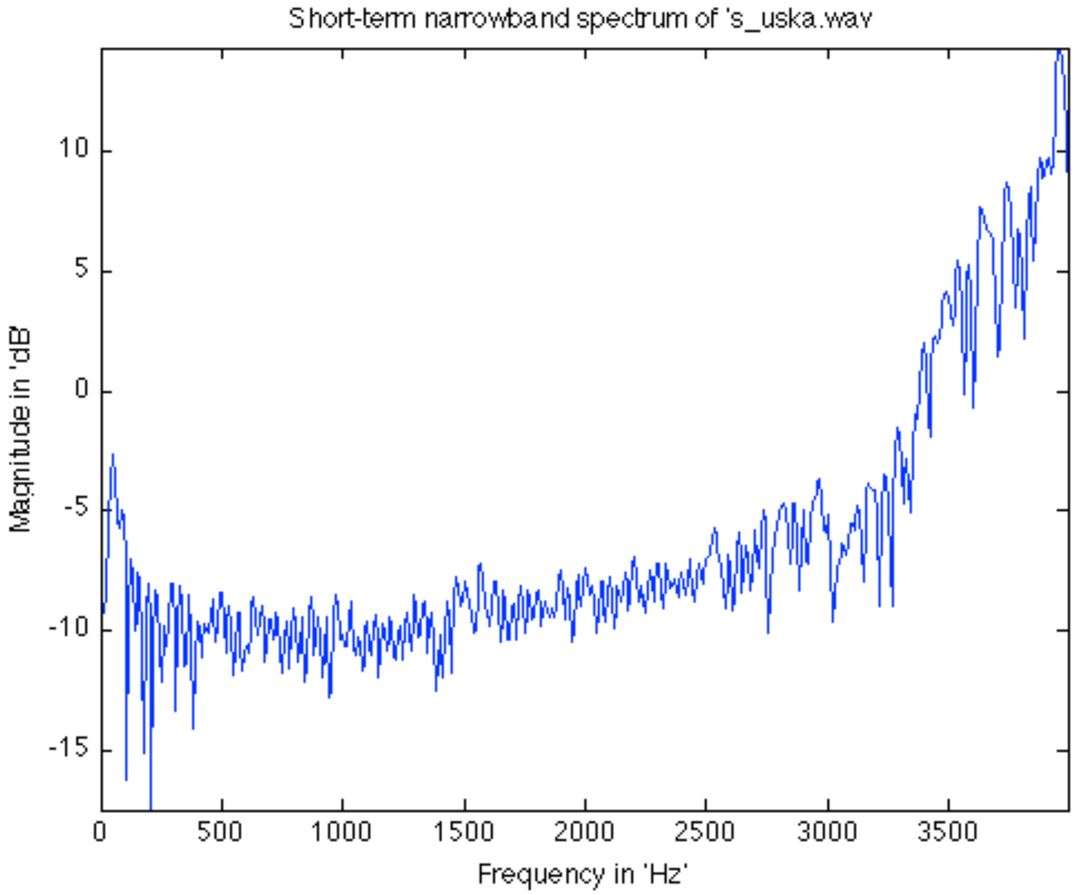
The plots:


Short-term narrowband spectrum of 'a_pani.wav
Magnitude in 'dB'
Frequency in 'Hz'
5
0
-5
-10
-15
0
500
1000
1500
2000
2500
3000
3500



Short-term narrowband spectrum of 'n_pani.wav
Magnitude in 'dB'
Frequency in 'Hz'
5
0
-5
-10
-15
-20
0
500
1000
1500
2000
2500
3000
3500


Short-term narrowband spectrum of 'i_pani.wav
Magnitude in 'dB'
Frequency in 'Hz'
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500

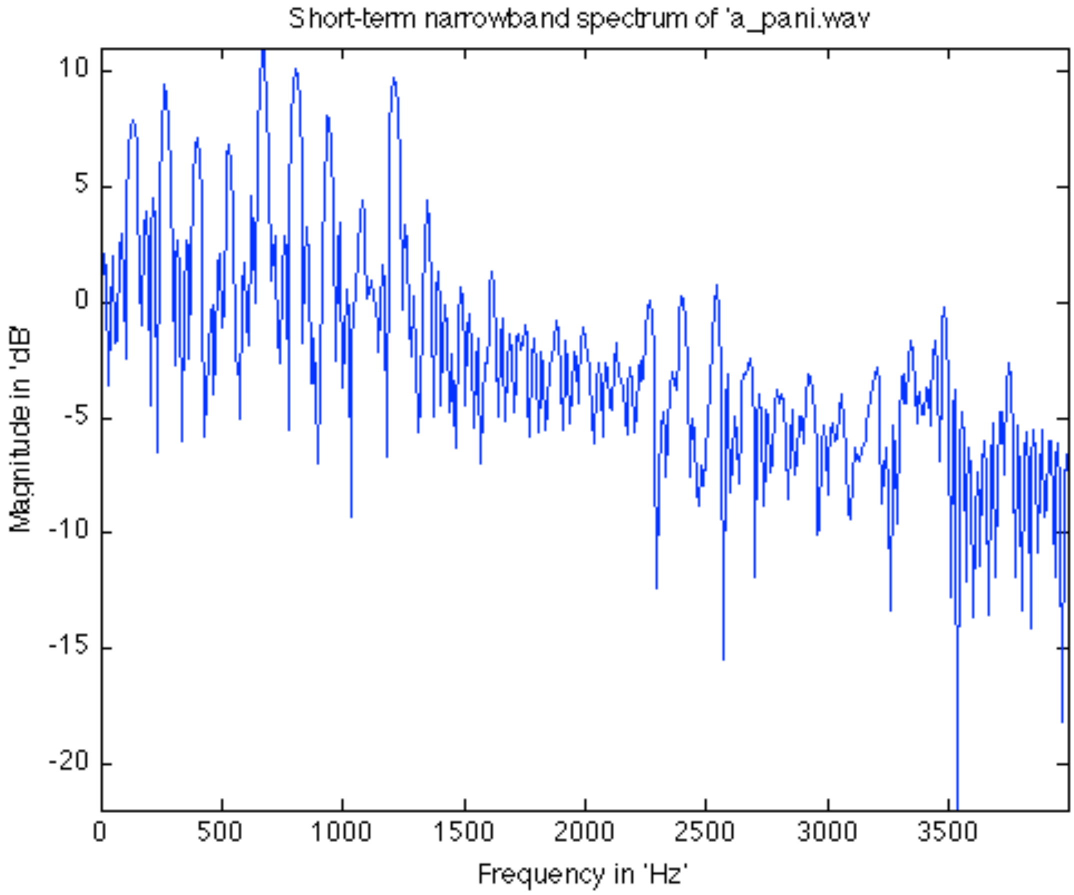


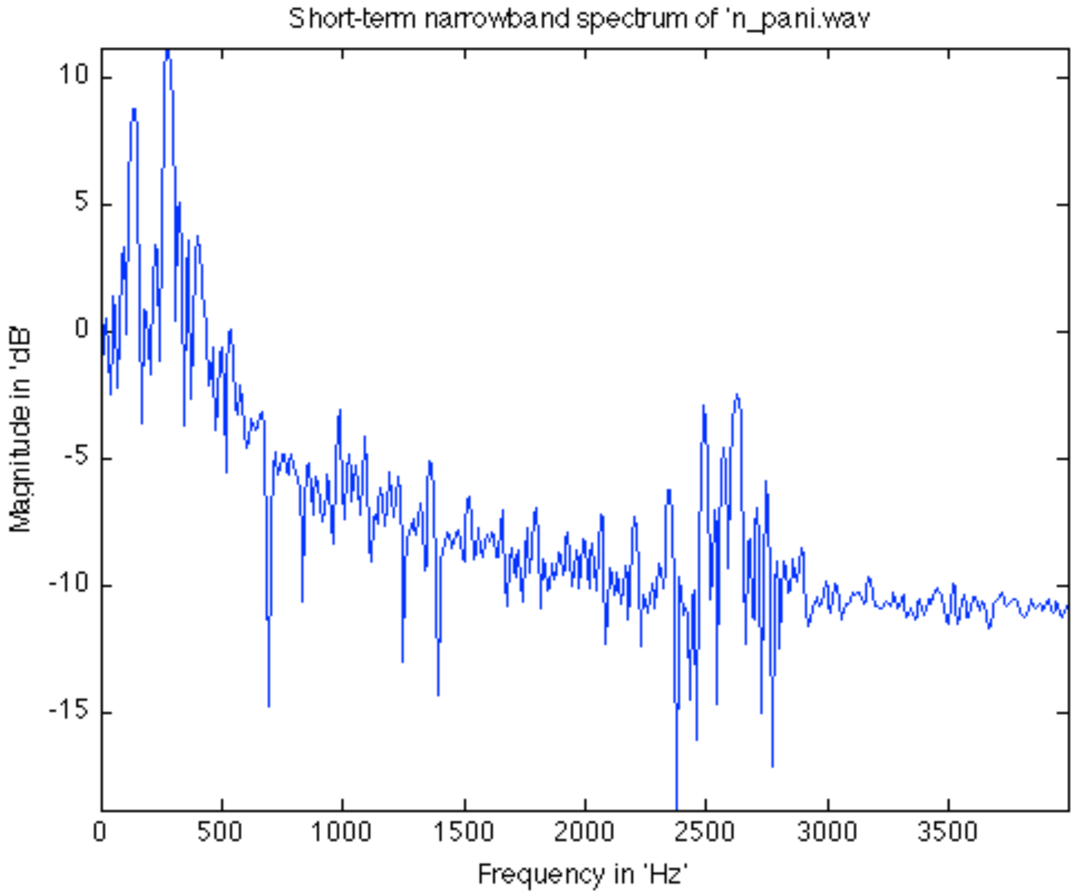
Finding the narrowband spectrum of four syllables (without pre-emphasis):

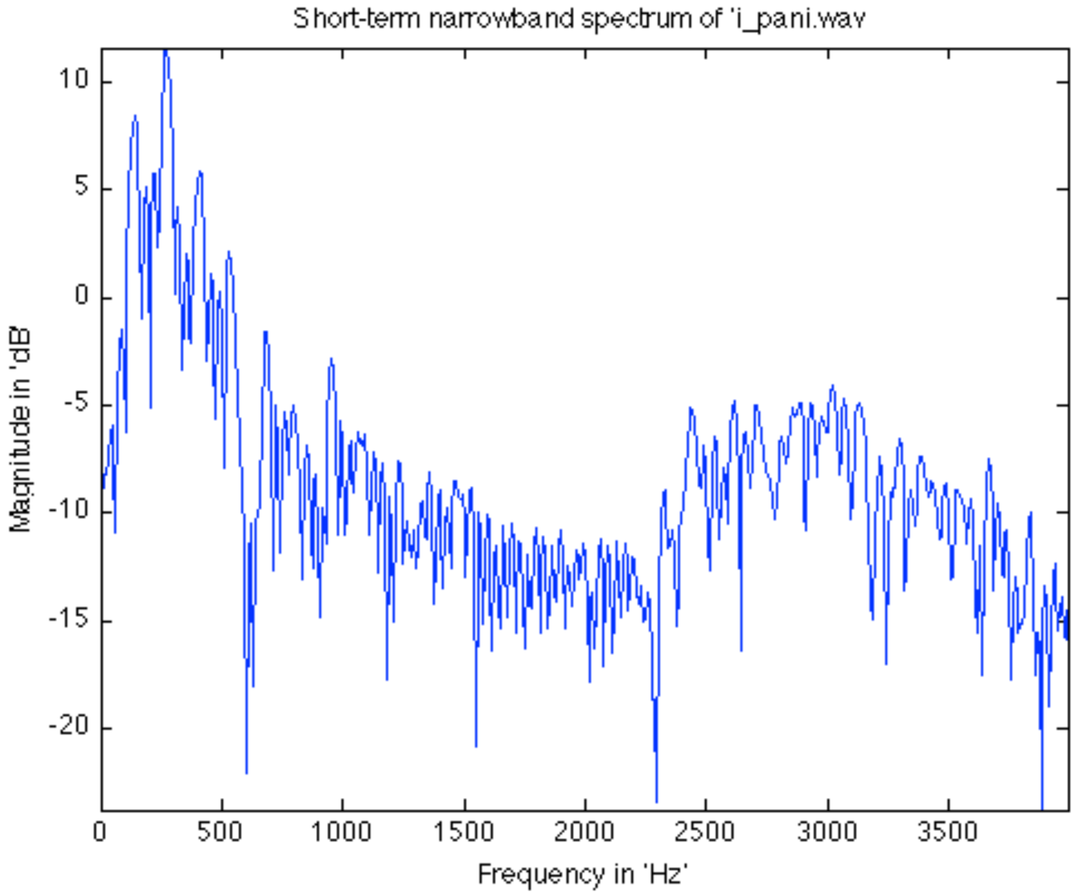
The functions (with changes from above are):

```
function [windowedSignal, fs] = getWindowedSignal(inputFile)

% inputFile = 'a_pani.wav';
windowDuration = 0.030; % in ms
%[y, fs] = preEmphasize(inputFile);
[y, fs] = wavread(inputFile);
siz = size(y);
length = siz(1);
centralIndex = round(length/2);
M = round(windowDuration * fs);
startIndex = round(centralIndex - M/2);
windowedSignal = y(startIndex:startIndex + M-1);
end
```

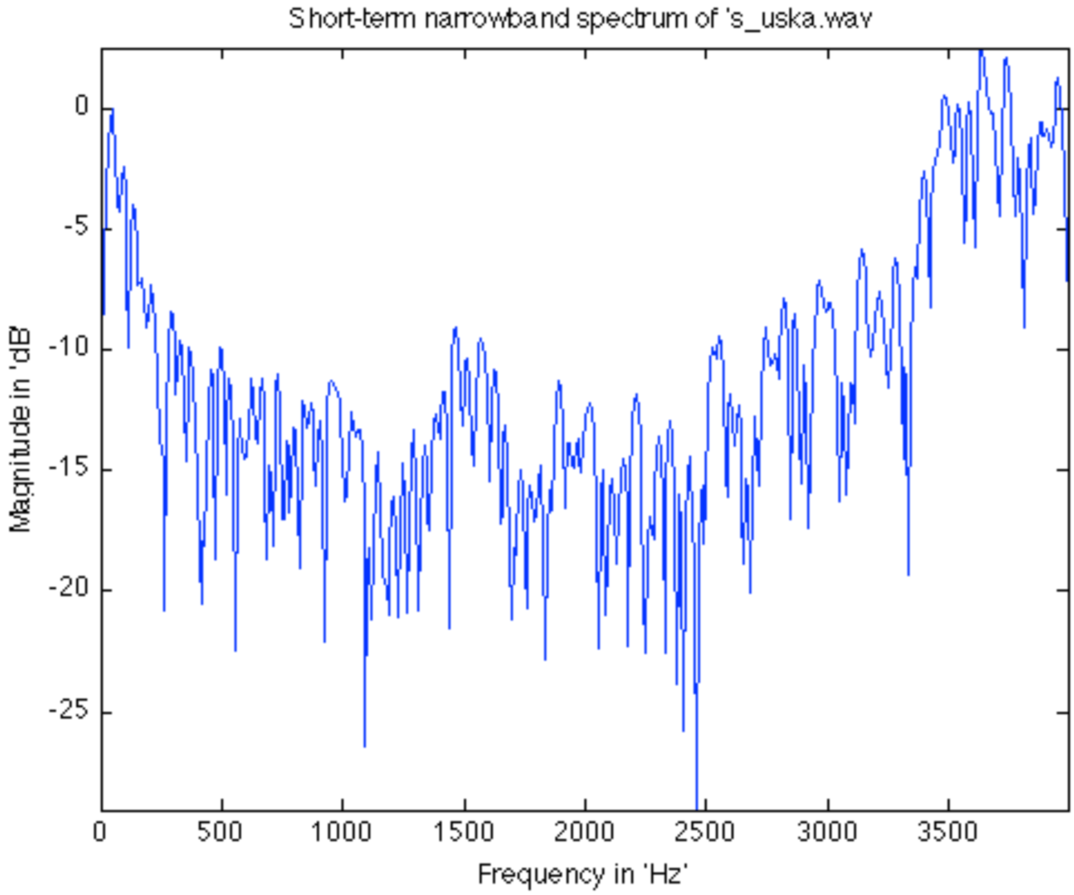
The plots:


Short-term narrowband spectrum of 'a_pani.wav
Magnitude in 'dB'
Frequency in 'Hz'
10
5
0
-5
-10
-15
-20
0
500
1000
1500
2000
2500
3000
3500


Short-term narrowband spectrum of 'n_pani.wav
Magnitude in 'dB'
Frequency in 'Hz'
10
5
0
-5
-10
-15
0
500
1000
1500
2000
2500
3000
3500

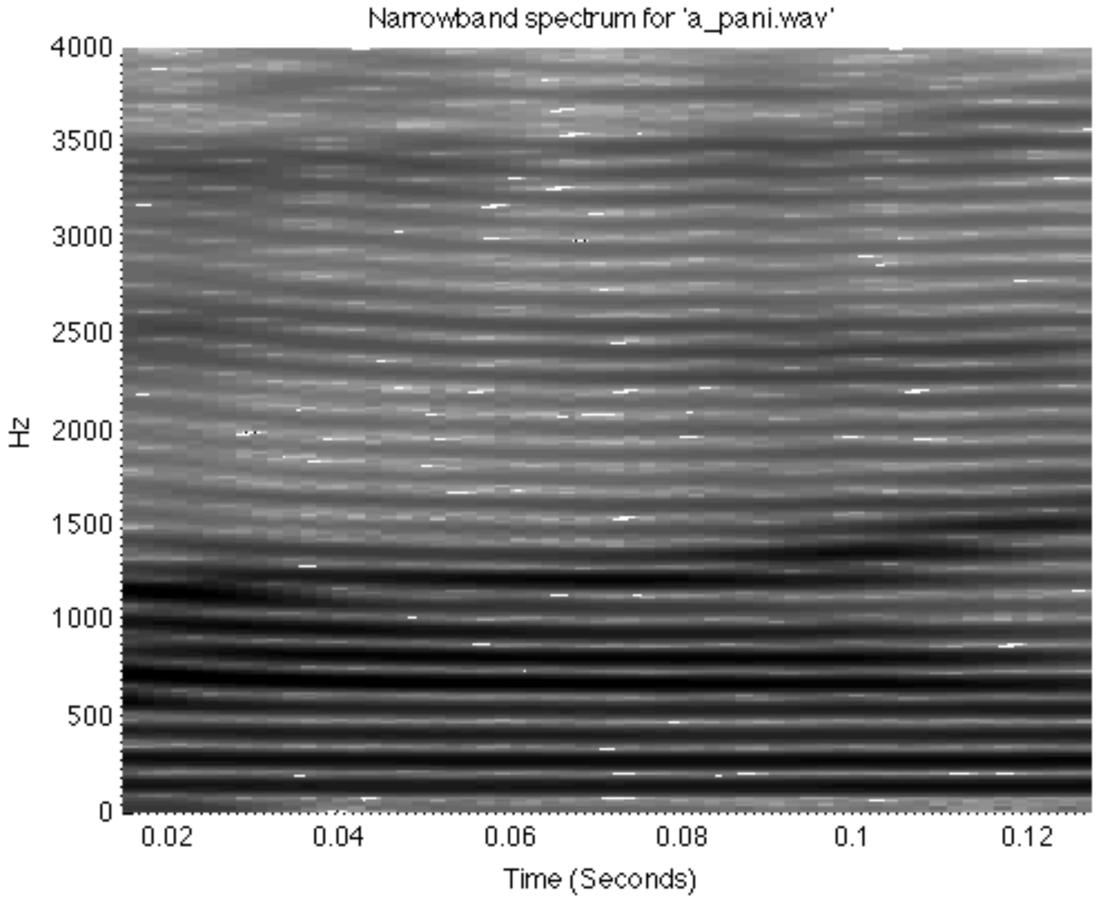


Short-term narrowband spectrum of 'i_pani.wav
Magnitude in 'dB'
Frequency in 'Hz'
10
5
0
-5
-10
-15
-20
0
500
1000
1500
2000
2500
3000
3500

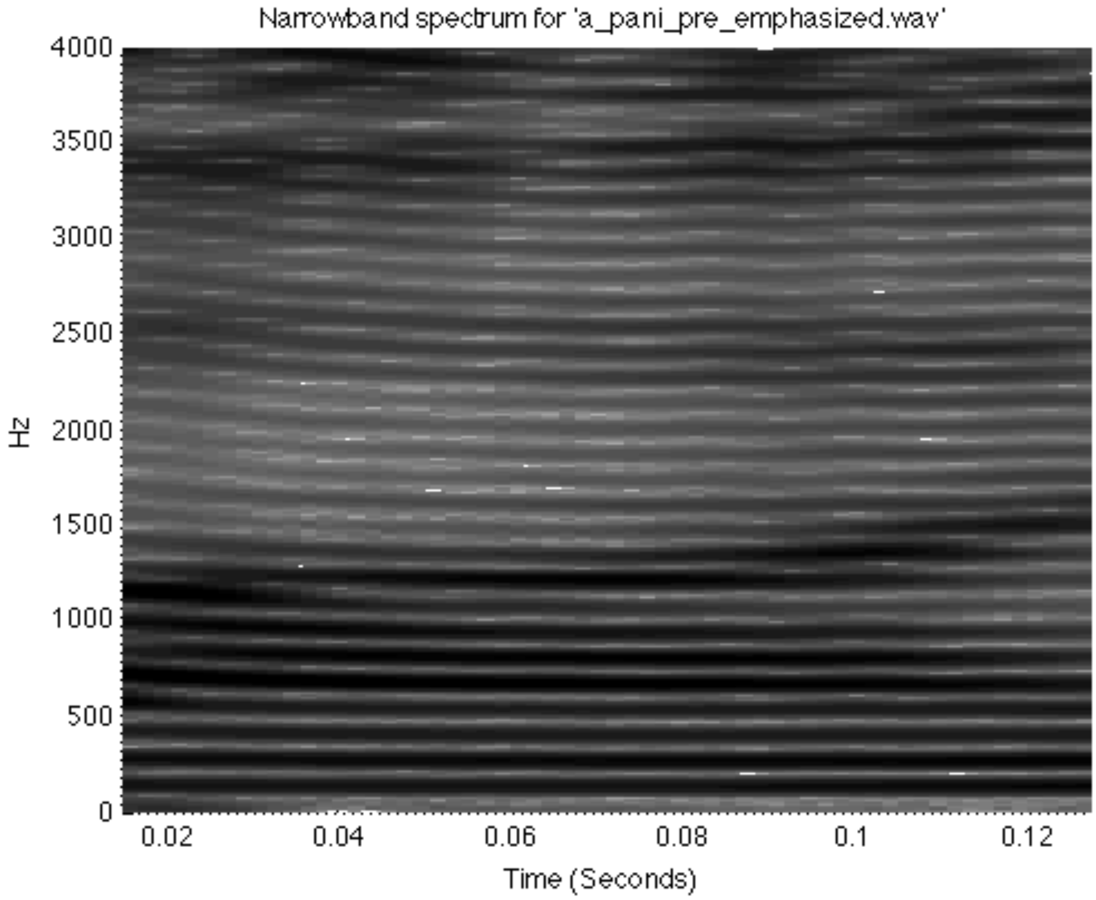


Finding the narrowband spectrogram of four syllables:

a) /a/ in ‘pani’

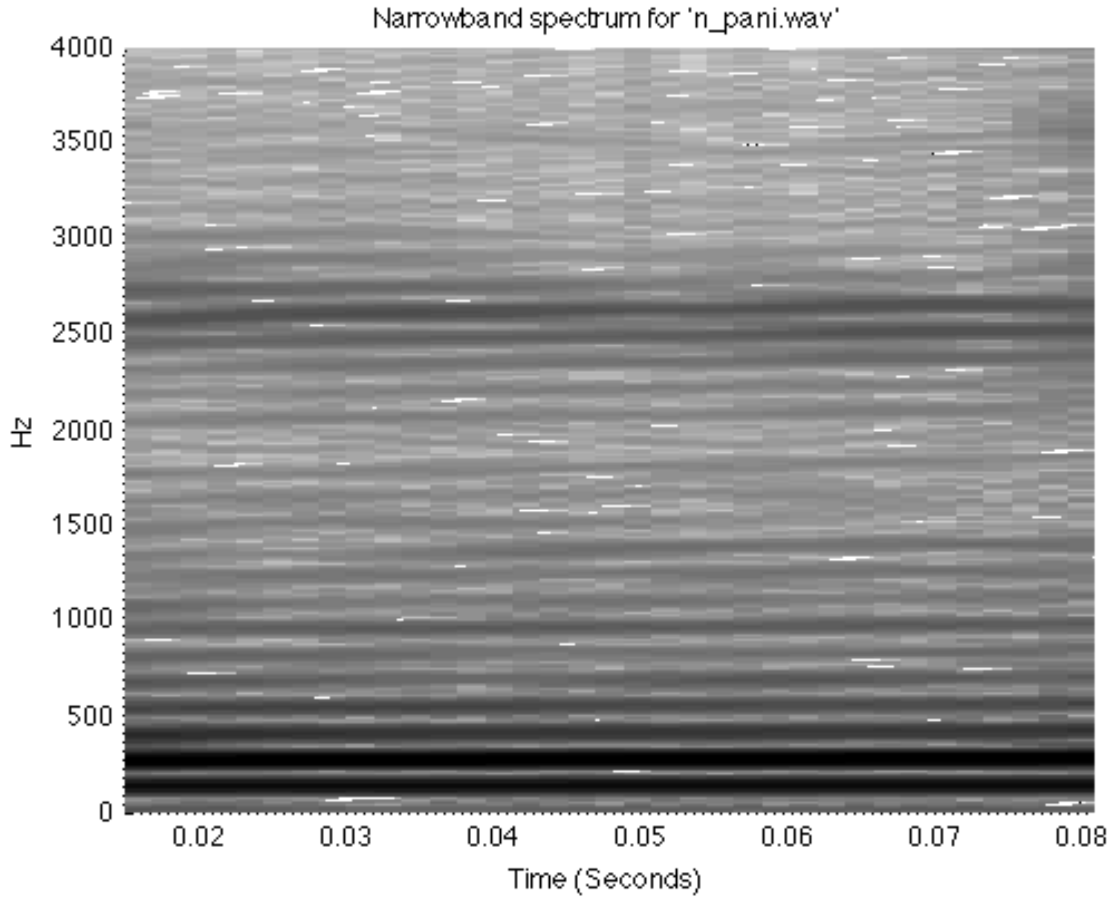


b) /a/ in ‘pani’ pre-emphasized

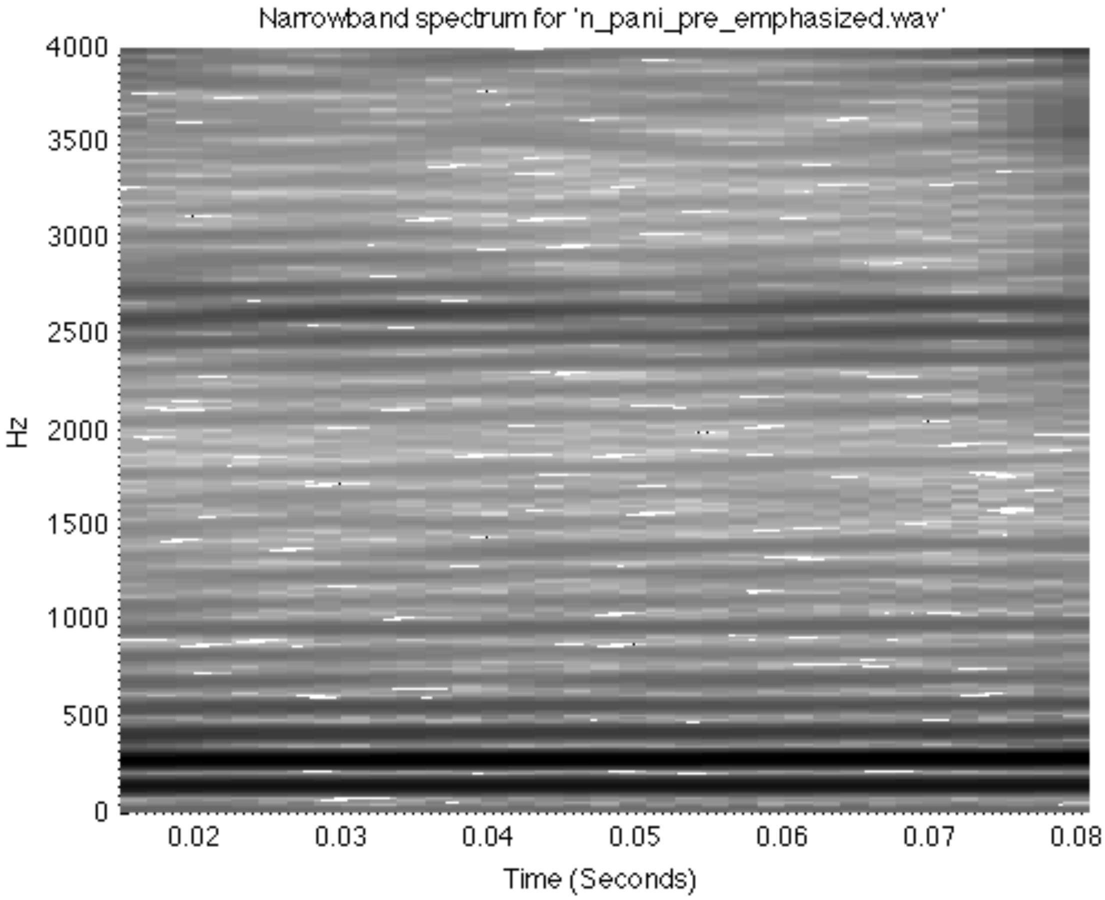


Narrowband spectrum for 'a_pani_pre_emphasized.wav'
Hz
4000
3500
3000
2500
2000
1500
1000
500
0
0.02
0.04
0.06
0.08
0.1
0.12
Time (Seconds)

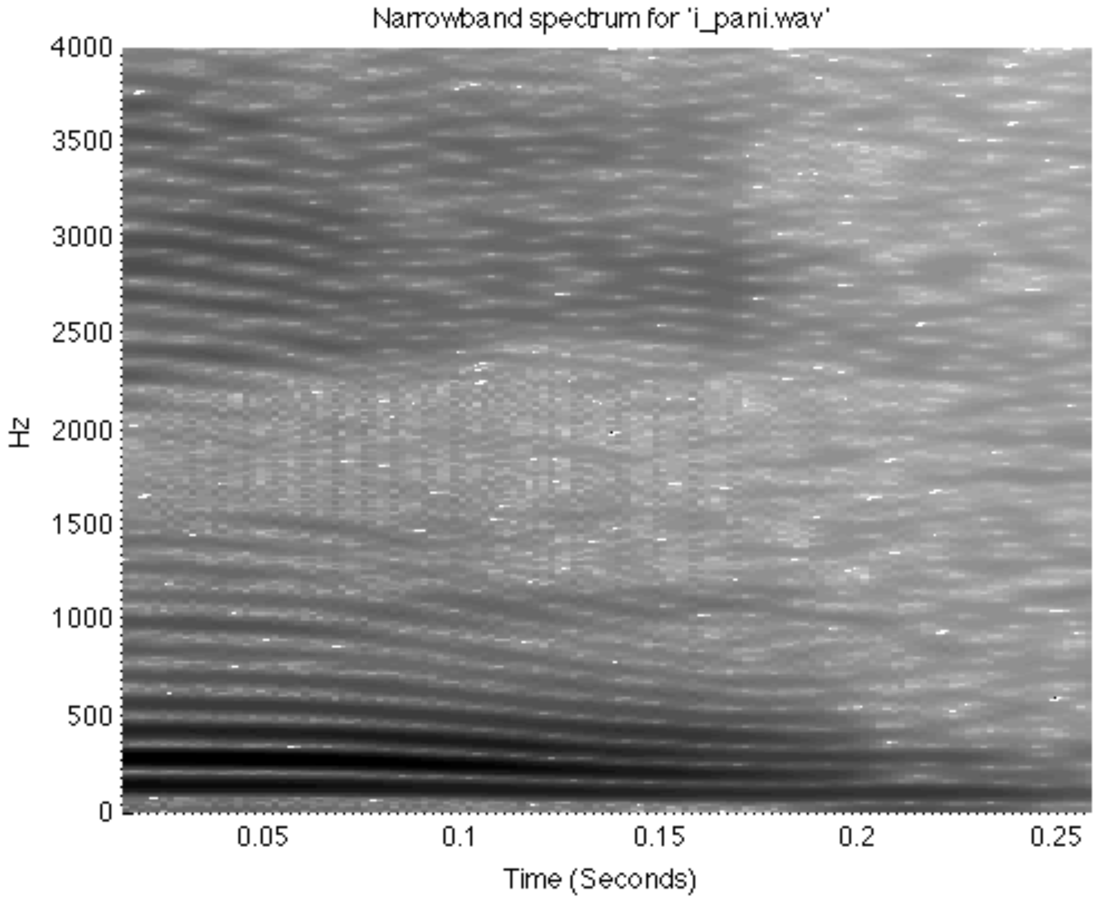
c) /n/ in ‘pani’



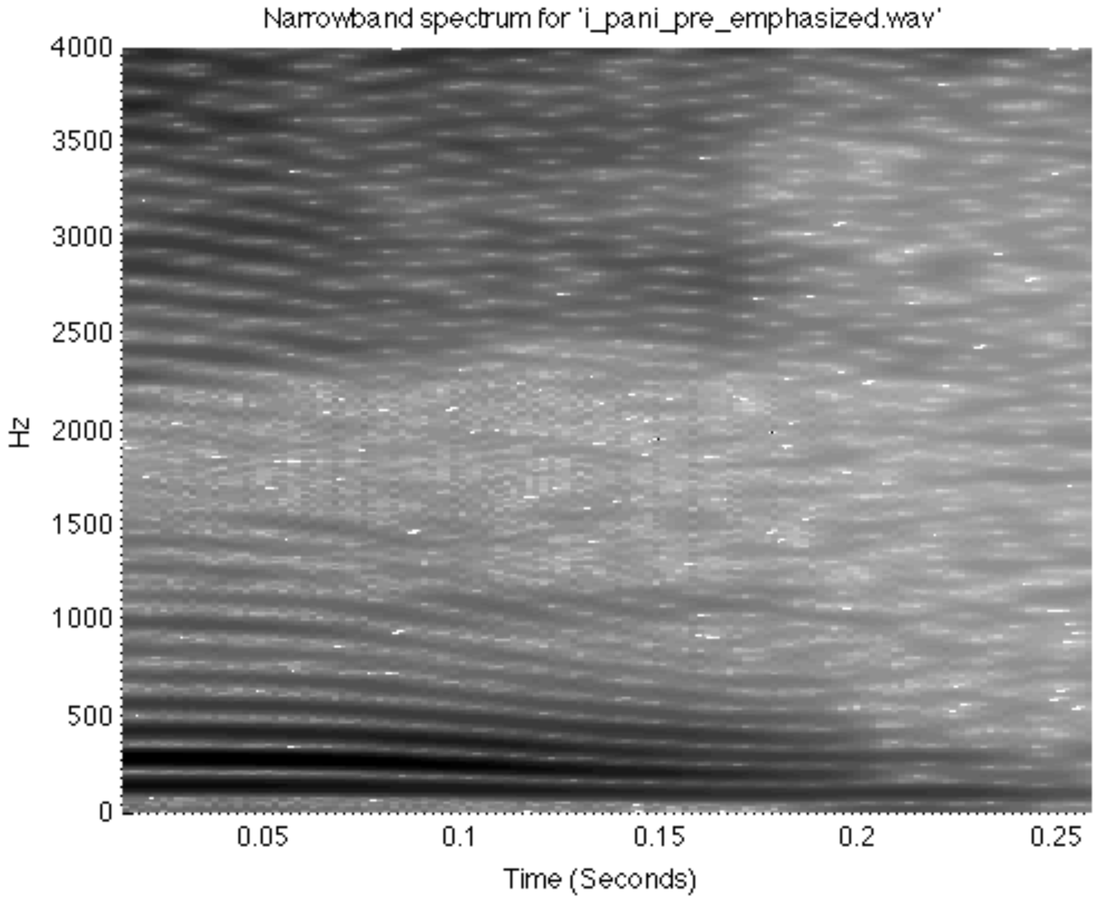
d) /n/ in ‘pani’ pre-emphasized


Narrowband spectrum for 'n_pani_pre_emphasized.wav'
4000
3500
3000
2500
2000
1500
1000
500
0
Hz
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Time (Seconds)

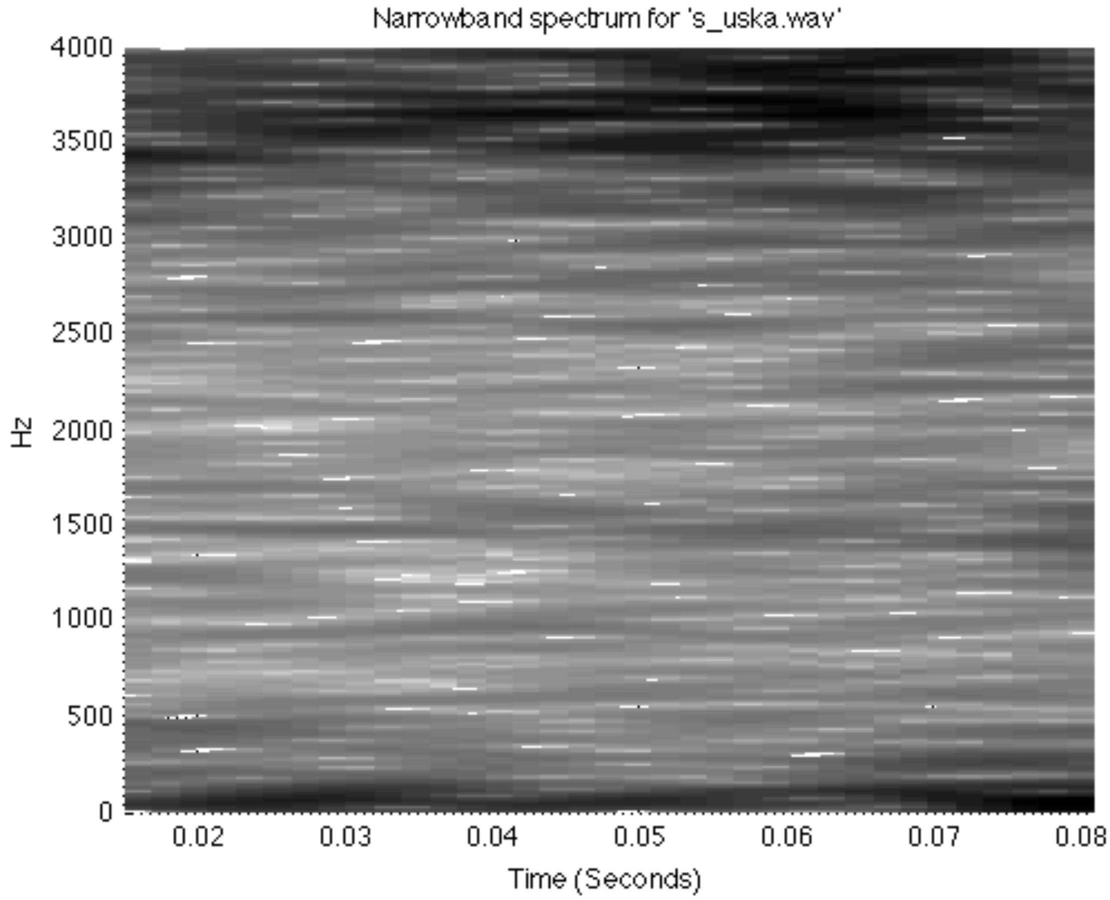
e) /l/ in ‘pani’



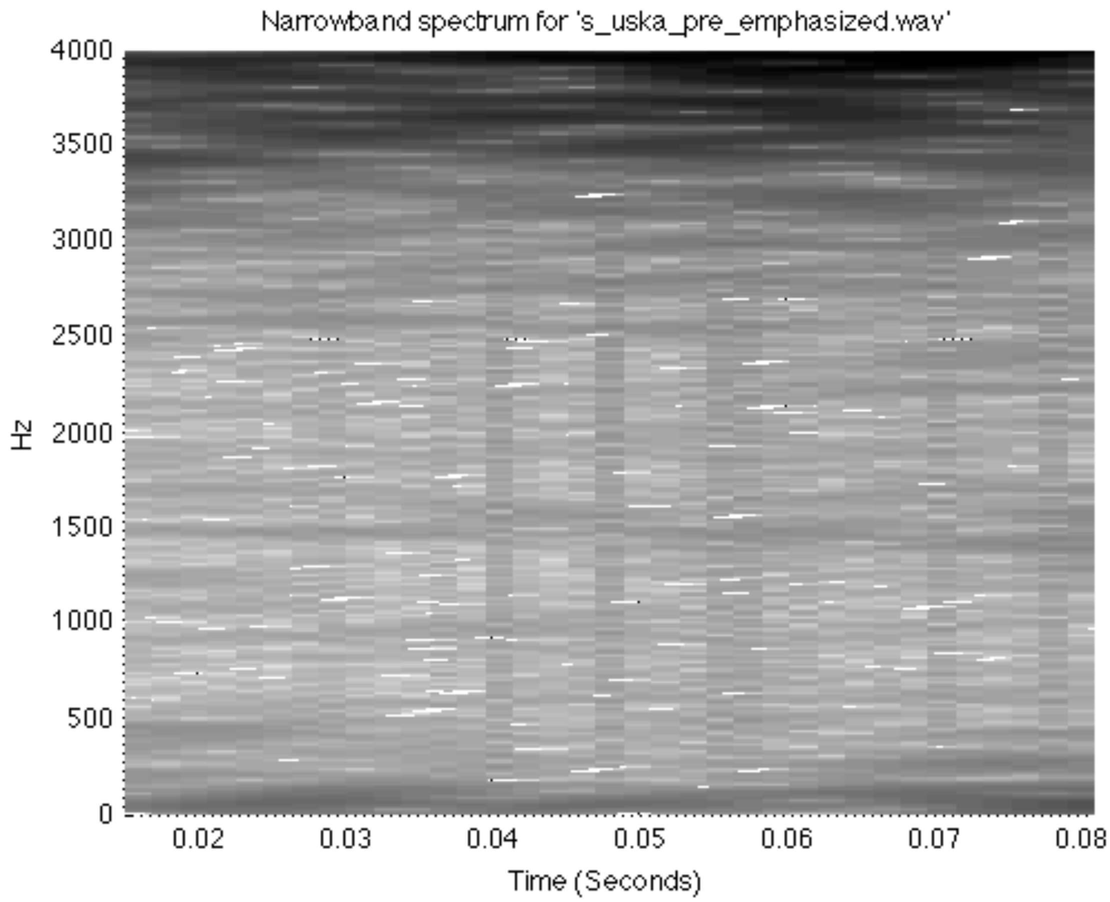
f) /I/ in ‘pani’ pre-emphasized


Narrowband spectrum for 'i_pani_pre_emphasized.wav'
4000
3500
3000
2500
2000
1500
1000
500
0
Hz
0.05
0.1
0.15
0.2
0.25
Time (Seconds)

g) /s/ in ‘uska’



h) /s/ in ‘uska’ pre-emphasized



<u>Script for finding the narrowband spectrum</u>

```
close all; clear all;


figure(100); clf;
printNarrowbandSpectrum('a_pani.wav');
[y, fs] = preEmphasize('a_pani.wav');
wavwrite(y, fs, 32, 'a_pani_pre_emphasized.wav');
figure(200); clf;
printNarrowbandSpectrum('a_pani_pre_emphasized.wav');

figure(300); clf;
printNarrowbandSpectrum('n_pani.wav');
[y, fs] = preEmphasize('n_pani.wav');
wavwrite(y, fs, 32, 'n_pani_pre_emphasized.wav');
figure(400); clf;
printNarrowbandSpectrum('n_pani_pre_emphasized.wav');
```

```
figure(500); clf;
printNarrowbandSpectrum('i_pani.wav');
[y, fs] = preEmphasize('i_pani.wav');
wavwrite(y, fs, 32, 'i_pani_pre_emphasized.wav');
figure(600); clf;
printNarrowbandSpectrum('i_pani_pre_emphasized.wav');

figure(700); clf;
printNarrowbandSpectrum('s_uska.wav');
[y, fs] = preEmphasize('s_uska.wav');
wavwrite(y, fs, 32, 's_uska_pre_emphasized.wav');
figure(800); clf;
printNarrowbandSpectrum('s_uska_pre_emphasized.wav');
```

Code for the functions used in finding the narrowband spectrum

```
function [signal, fs] = preEmphasize(inputFile)

[y, fs] = wavread(inputFile);
siz = size(y);
length = siz(1);

for k = 1:length
    if k > 1
        y(k) = y(k) - (0.97*y(k-1));
    end
end

signal = y;
end


function printNarrowbandSpectrum(fileInput)

[y, fs] = wavread(fileInput);

colormap('gray');
map = colormap;
imap = flipud(map);
M = round(0.030*fs);  % 30 ms window
N = 2^nextpow2(4*M); % with zero padding
w = 0.54 - 0.46 * cos(2*pi*[0:M-1]/(M-1)); % w = hamming(M);
[~,F,T,P] = spectrogram(y, w, (M*(15/16)), N, fs);
```

```
surf(T,F,10*log10(P),'edgecolor','none');
title(['Narrowband spectrum for ''', fileInput, ''''],
'interpreter', 'none');
axis tight;
colormap(imap);
view(0,90);
xlabel('Time (Seconds)'); ylabel('Hz');

end
```

Answer to Question 2(a)

The script:

```
close all; clear all;

autocorrCoeffs1 = getAutoCorrCoefficients('a_pani', 6);
autocorrCoeffs2 = getAutoCorrCoefficients('a_pani', 8);
autocorrCoeffs3 = getAutoCorrCoefficients('a_pani', 10);
autocorrCoeffs4 = getAutoCorrCoefficients('a_pani', 12);
autocorrCoeffs5 = getAutoCorrCoefficients('a_pani', 16);

figure(100); clf;
subplot(3,2,1); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''a_pani'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''a_pani'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''a_pani'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''a_pani'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''a_pani'' for p = ''16''',
'interpreter', 'none');

LPCoeffs1 = getLPCoefficients('a_pani', 6);
LPCoeffs2 = getLPCoefficients('a_pani', 8);
LPCoeffs3 = getLPCoefficients('a_pani', 10);
```

```
LPCoeffs4 = getLPCoefficients('a_pani', 12);
LPCoeffs5 = getLPCoefficients('a_pani', 16);

figure(200); clf;
subplot(3,2,1); stem(LPCoeffs1); axis tight;
title('LP coefficients of ''a_pani'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(LPCoeffs2); axis tight;
title('LP coefficients of ''a_pani'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(LPCoeffs3); axis tight;
title('LP coefficients of ''a_pani'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(LPCoeffs4); axis tight;
title('LP coefficients of ''a_pani'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(LPCoeffs5); axis tight;
title('LP coefficients of ''a_pani'' for p = ''16''',
'interpreter', 'none');


%% Now for n_pani

autocorrCoeffs1 = getAutoCorrCoefficients('n_pani', 6);
autocorrCoeffs2 = getAutoCorrCoefficients('n_pani', 8);
autocorrCoeffs3 = getAutoCorrCoefficients('n_pani', 10);
autocorrCoeffs4 = getAutoCorrCoefficients('n_pani', 12);
autocorrCoeffs5 = getAutoCorrCoefficients('n_pani', 16);

figure(300); clf;
subplot(3,2,1); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''n_pani'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''n_pani'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''n_pani'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''n_pani'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''n_pani'' for p = ''16''',
'interpreter', 'none');
```

```
LPCoeffs1 = getLPCoefficients('n_pani', 6);
LPCoeffs2 = getLPCoefficients('n_pani', 8);
LPCoeffs3 = getLPCoefficients('n_pani', 10);
LPCoeffs4 = getLPCoefficients('n_pani', 12);
LPCoeffs5 = getLPCoefficients('n_pani', 16);

figure(400); clf;
subplot(3,2,1); stem(LPCoeffs1); axis tight;
title('LP coefficients of ''n_pani'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(LPCoeffs2); axis tight;
title('LP coefficients of ''n_pani'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(LPCoeffs3); axis tight;
title('LP coefficients of ''n_pani'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(LPCoeffs4); axis tight;
title('LP coefficients of ''n_pani'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(LPCoeffs5); axis tight;
title('LP coefficients of ''n_pani'' for p = ''16''',
'interpreter', 'none');


%% Now for i_pani

autocorrCoeffs1 = getAutoCorrCoefficients('i_pani', 6);
autocorrCoeffs2 = getAutoCorrCoefficients('i_pani', 8);
autocorrCoeffs3 = getAutoCorrCoefficients('i_pani', 10);
autocorrCoeffs4 = getAutoCorrCoefficients('i_pani', 12);
autocorrCoeffs5 = getAutoCorrCoefficients('i_pani', 16);

figure(500); clf;
subplot(3,2,1); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''i_pani'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''i_pani'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''i_pani'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''i_pani'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(autocorrCoeffs1); axis tight;
```

```
title('Autocorr coefficients of ''i_pani'' for p = ''16''',
'interpreter', 'none');

LPCoeffs1 = getLPCoefficients('i_pani', 6);
LPCoeffs2 = getLPCoefficients('i_pani', 8);
LPCoeffs3 = getLPCoefficients('i_pani', 10);
LPCoeffs4 = getLPCoefficients('i_pani', 12);
LPCoeffs5 = getLPCoefficients('i_pani', 16);

figure(600); clf;
subplot(3,2,1); stem(LPCoeffs1); axis tight;
title('LP coefficients of ''i_pani'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(LPCoeffs2); axis tight;
title('LP coefficients of ''i_pani'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(LPCoeffs3); axis tight;
title('LP coefficients of ''i_pani'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(LPCoeffs4); axis tight;
title('LP coefficients of ''i_pani'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(LPCoeffs5); axis tight;
title('LP coefficients of ''i_pani'' for p = ''16''',
'interpreter', 'none');


%% Now for s_uska

autocorrCoeffs1 = getAutoCorrCoefficients('s_uska', 6);
autocorrCoeffs2 = getAutoCorrCoefficients('s_uska', 8);
autocorrCoeffs3 = getAutoCorrCoefficients('s_uska', 10);
autocorrCoeffs4 = getAutoCorrCoefficients('s_uska', 12);
autocorrCoeffs5 = getAutoCorrCoefficients('s_uska', 16);

figure(700); clf;
subplot(3,2,1); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''s_uska'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''s_uska'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''s_uska'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(autocorrCoeffs1); axis tight;
```

```
title('Autocorr coefficients of ''s_uska'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(autocorrCoeffs1); axis tight;
title('Autocorr coefficients of ''s_uska'' for p = ''16''',
'interpreter', 'none');

LPCoeffs1 = getLPCoefficients('s_uska', 6);
LPCoeffs2 = getLPCoefficients('s_uska', 8);
LPCoeffs3 = getLPCoefficients('s_uska', 10);
LPCoeffs4 = getLPCoefficients('s_uska', 12);
LPCoeffs5 = getLPCoefficients('s_uska', 16);

figure(800); clf;
subplot(3,2,1); stem(LPCoeffs1); axis tight;
title('LP coefficients of ''s_uska'' for p = ''6''',
'interpreter', 'none');
subplot(3,2,2); stem(LPCoeffs2); axis tight;
title('LP coefficients of ''s_uska'' for p = ''8''',
'interpreter', 'none');
subplot(3,2,3); stem(LPCoeffs3); axis tight;
title('LP coefficients of ''s_uska'' for p = ''10''',
'interpreter', 'none');
subplot(3,2,4); stem(LPCoeffs4); axis tight;
title('LP coefficients of ''s_uska'' for p = ''12''',
'interpreter', 'none');
subplot(3,2,5); stem(LPCoeffs5); axis tight;
title('LP coefficients of ''s_uska'' for p = ''16''',
'interpreter', 'none');
```

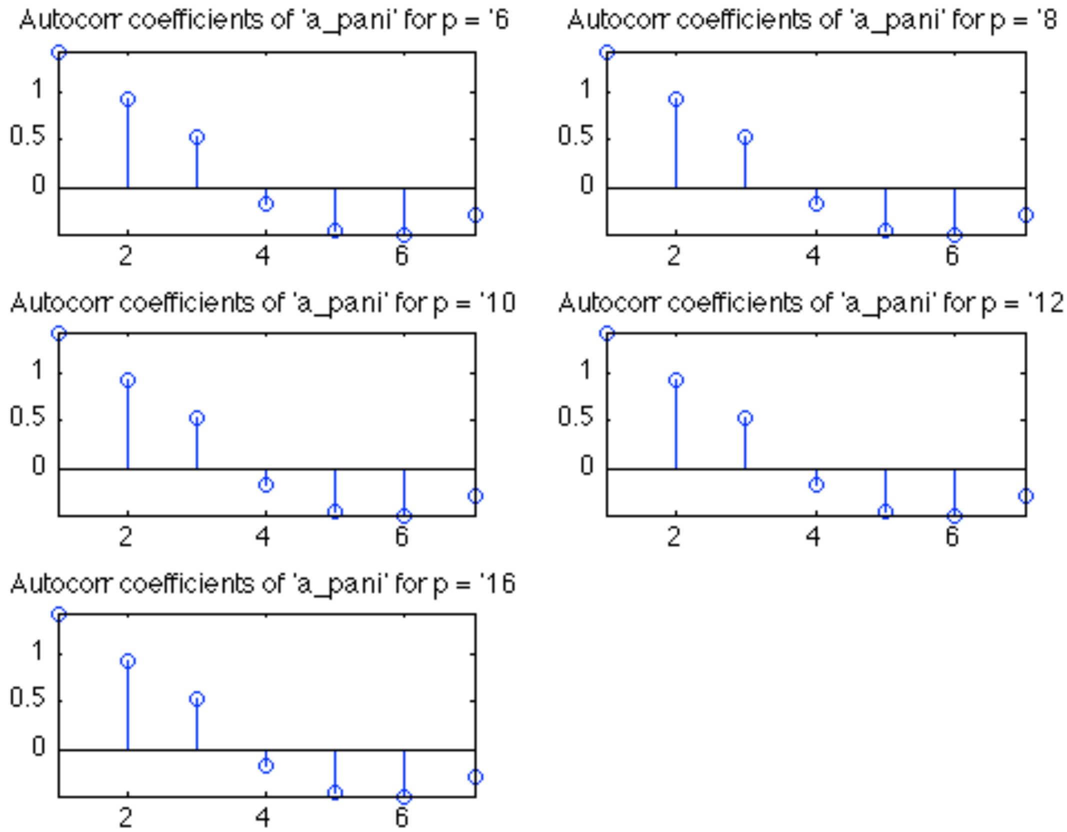
The functions:

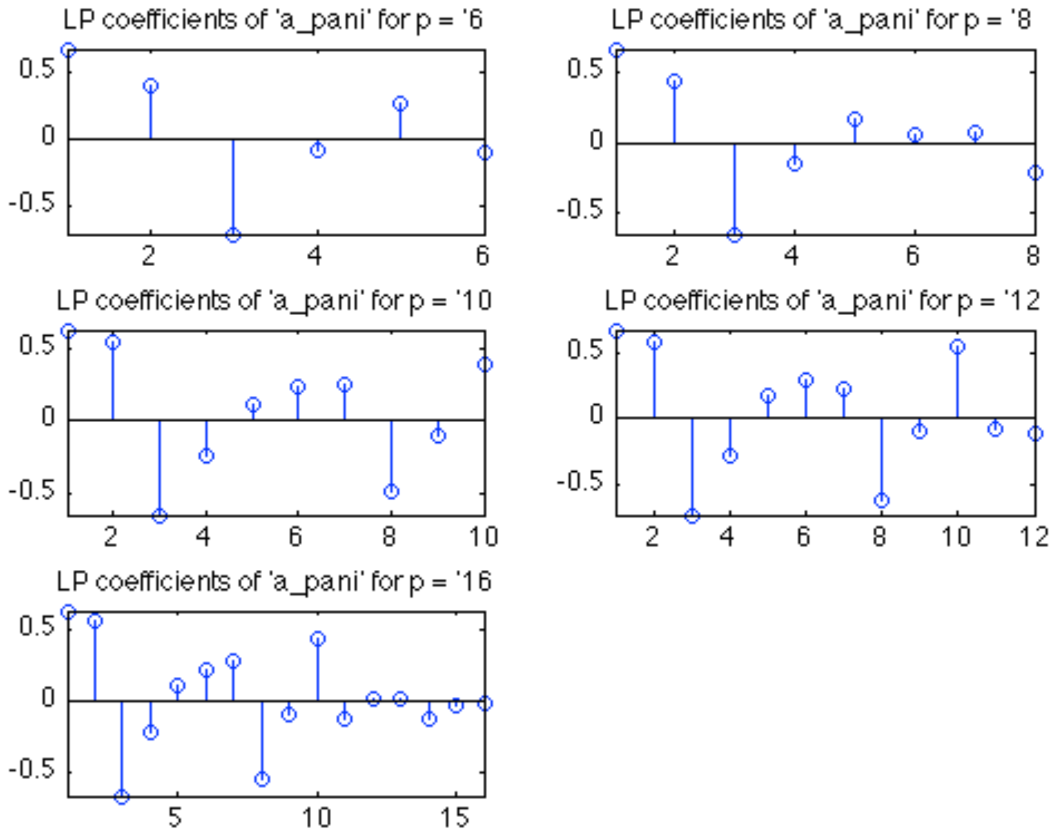
```
function LPCoeffs = getLPCoefficients(inputFile, poleOrder)


autocorrCoeffs = getAutoCorrCoefficients(inputFile, poleOrder);
LPCoeffs = levinsonDurbin(autocorrCoeffs);
end

function autocorrelationCoefficients =
getAutoCorrCoefficients(inputFile, poleOrder)
% poleOrder = 6 ;
% inputFile = 'a_pani.wav';
windowDuration = 0.030; % in ms

[y, fs] = preEmphasize(inputFile);
```

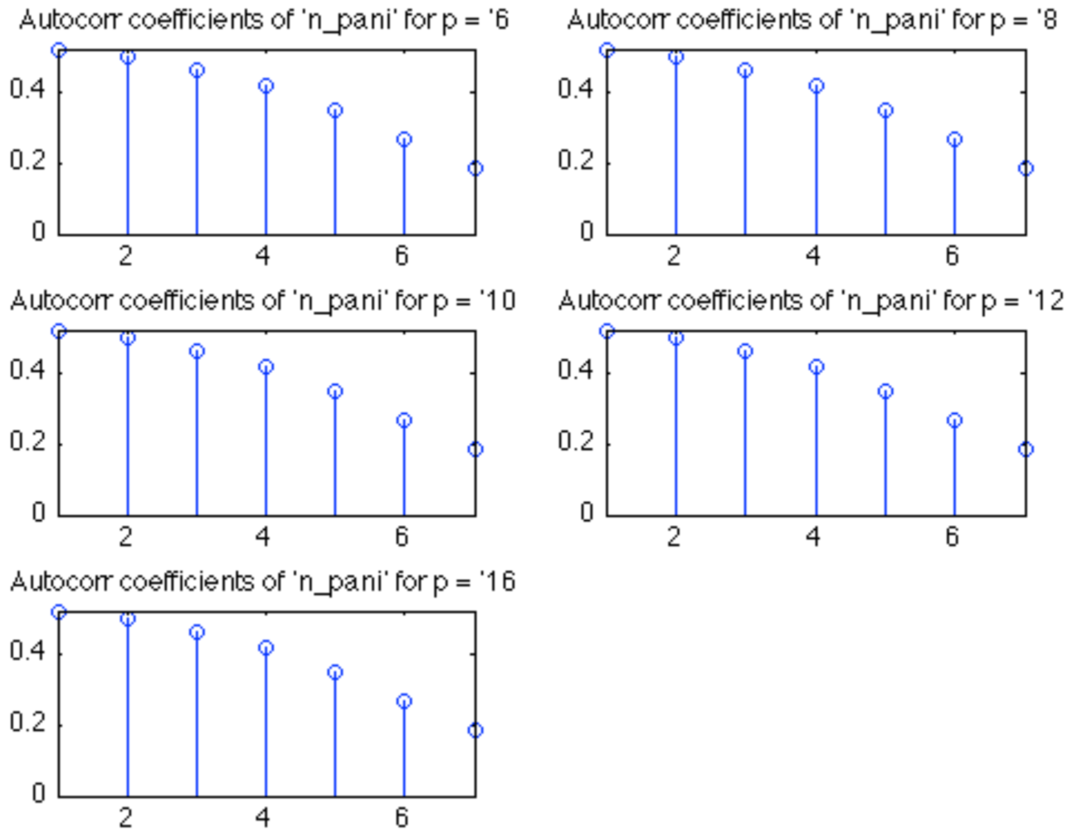
```
siz = size(y);
length = siz(1);
centralIndex = round(length/2);
M = round(windowDuration * fs);
startIndex = round(centralIndex - M/2);

windowedSignal = y(startIndex:startIndex + M-1);
ACCoeff = zeros(poleOrder+1, 1);

for p = 0:poleOrder
    for k = 0:M-1
        valueToBeAdded = 0;
        if k-p >= 0
            valueToBeAdded = windowedSignal(k+1) .*
windowedSignal(k+1-p);
        end
        ACCoeff(p+1) = ACCoeff(p+1) + valueToBeAdded;
    end
end

autocorrelationCoefficients = ACCoeff;
% figure, stem(ACCoeff);
% title(['Autocorrelation coefficients of ''', inputFile, '''
for p = ''', num2str(poleOrder), ''''], 'interpreter', 'none');

end


function [ coeff, b0 ] = levinsonDurbin(autocorrCoeffs)

siz = size(autocorrCoeffs(:));
poleOrder = siz(1) - 1;
p = poleOrder;
rx = autocorrCoeffs(:); % autocorr coefficients
a = zeros(p+1);
e = zeros(1, p+1);
G = zeros(1, p+1);
reflected = zeros(1, p+1);
a(1, 1) = 1; e(1) = rx(1);

for j = 0:p-1;
    G(j+1) = rx(j+2);
    sum = 0;
    if j > 0
        for i = 1:j;
```

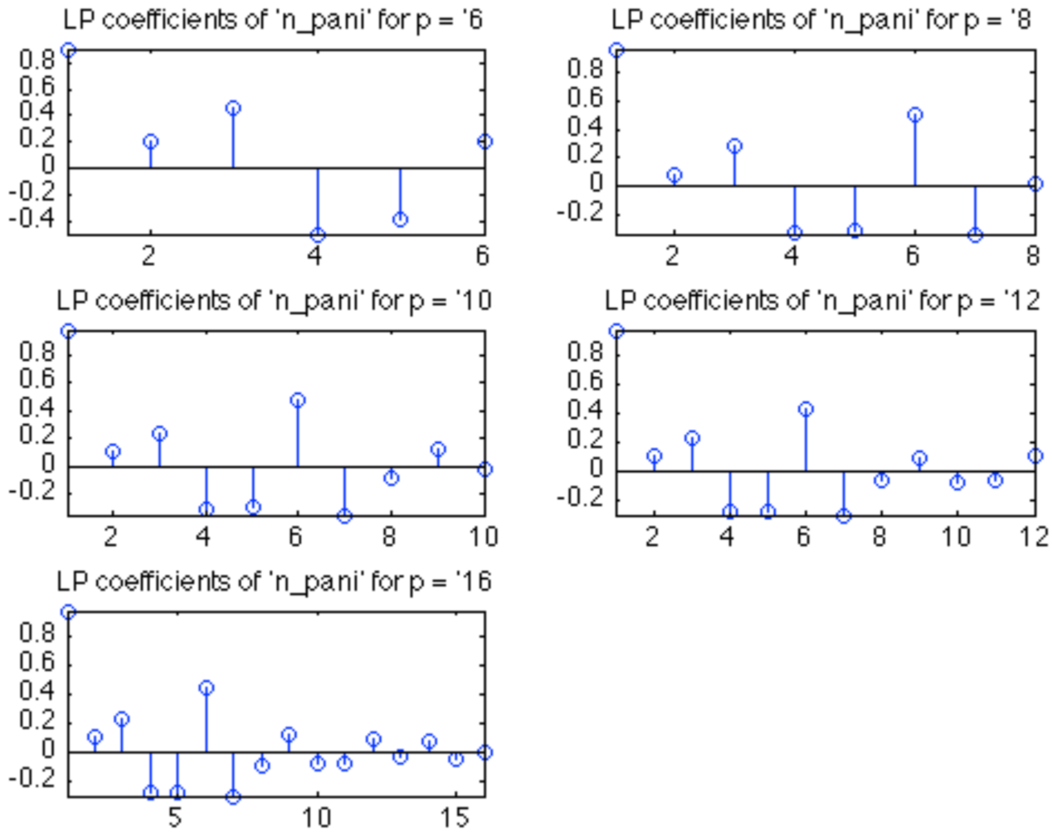
```
            sum = sum + (a(j+1, i+1)*rx(j+2-i));
        end
    end
    G(j+1) = G(j+1) + sum;
    reflected(j + 2) = -G(j+1)/e(j+1);

    for i  = 1:j;
            a(j+2, i+1) = a(j+1, i+1) + (reflected(j+2)*a(j+1,
j-i+2));
    end

    a(j+2, j+2) = reflected(j+2);
    e(j+2) = e(j + 1) * (1-(abs(reflected(j+2))^2));

end
b0 = sqrt(e(p+1)); % the b(0) that we require in the numerator
of the transfer function
coeff = a(p+1, (2:end)); % coeff is the array of a(p)
coefficients
coeff = -1 * coeff;

end
```

<u>The plots:</u>

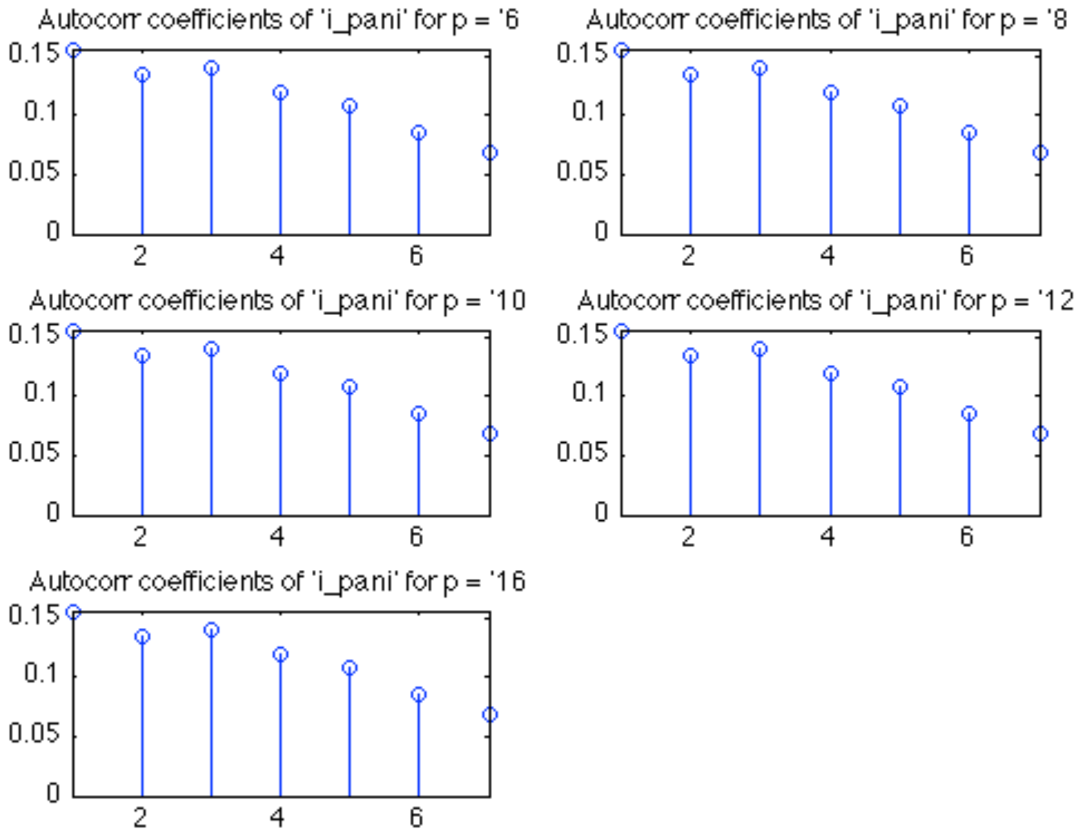

Autocorr coefficients of 'a_pani' for p = '6
Autocorr coefficients of 'a_pani' for p = '8
Autocorr coefficients of 'a_pani' for p = '10
Autocorr coefficients of 'a_pani' for p = '12
Autocorr coefficients of 'a_pani' for p = '16
1
0.5
0
2
4
6

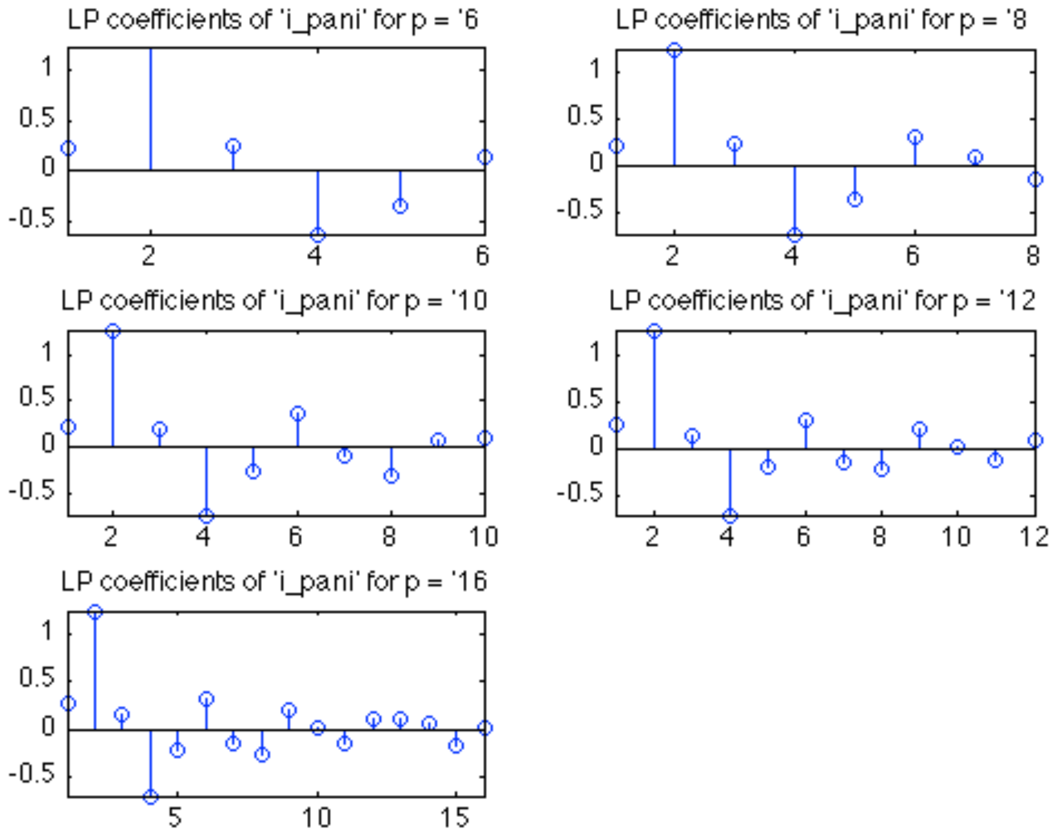

LP coefficients of 'a_pani' for p = '6
LP coefficients of 'a_pani' for p = '8
LP coefficients of 'a_pani' for p = '10
LP coefficients of 'a_pani' for p = '12
LP coefficients of 'a_pani' for p = '16
0.5
0
-0.5

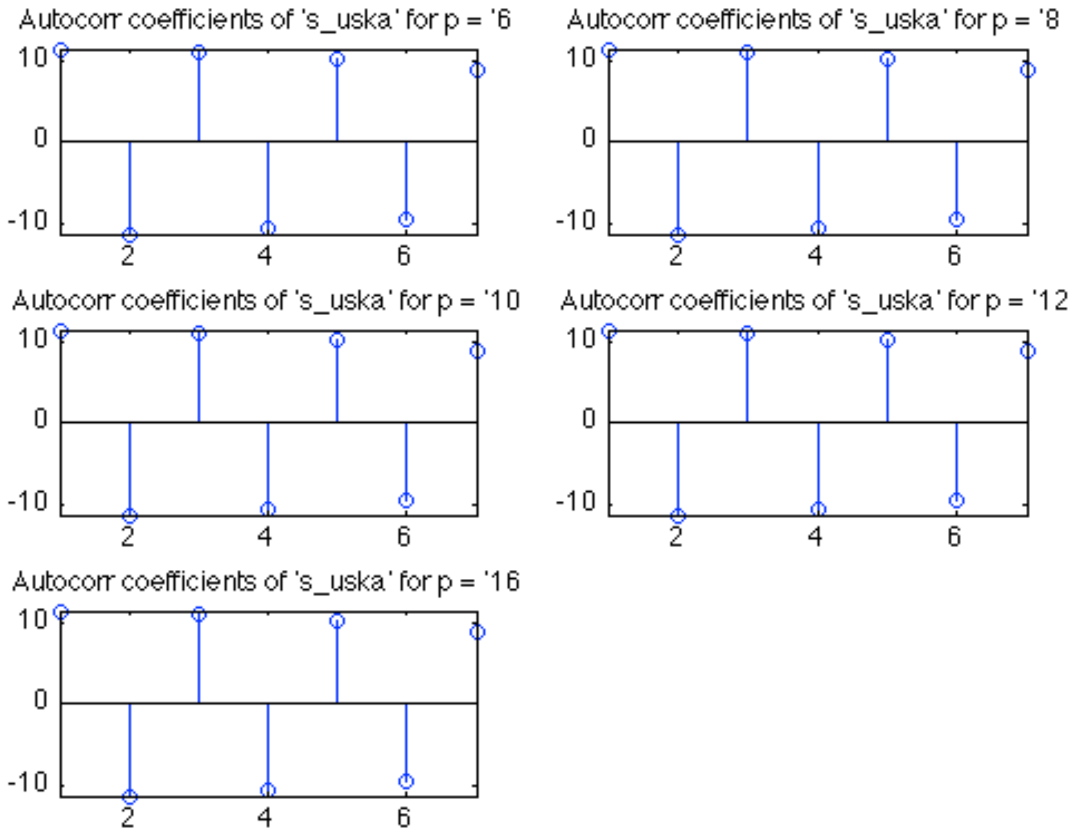

Autocorr coefficients of 'n_pani' for p = '6
Autocorr coefficients of 'n_pani' for p = '8
Autocorr coefficients of 'n_pani' for p = '10
Autocorr coefficients of 'n_pani' for p = '12
Autocorr coefficients of 'n_pani' for p = '16
0
0.2
0.4
2
4
6

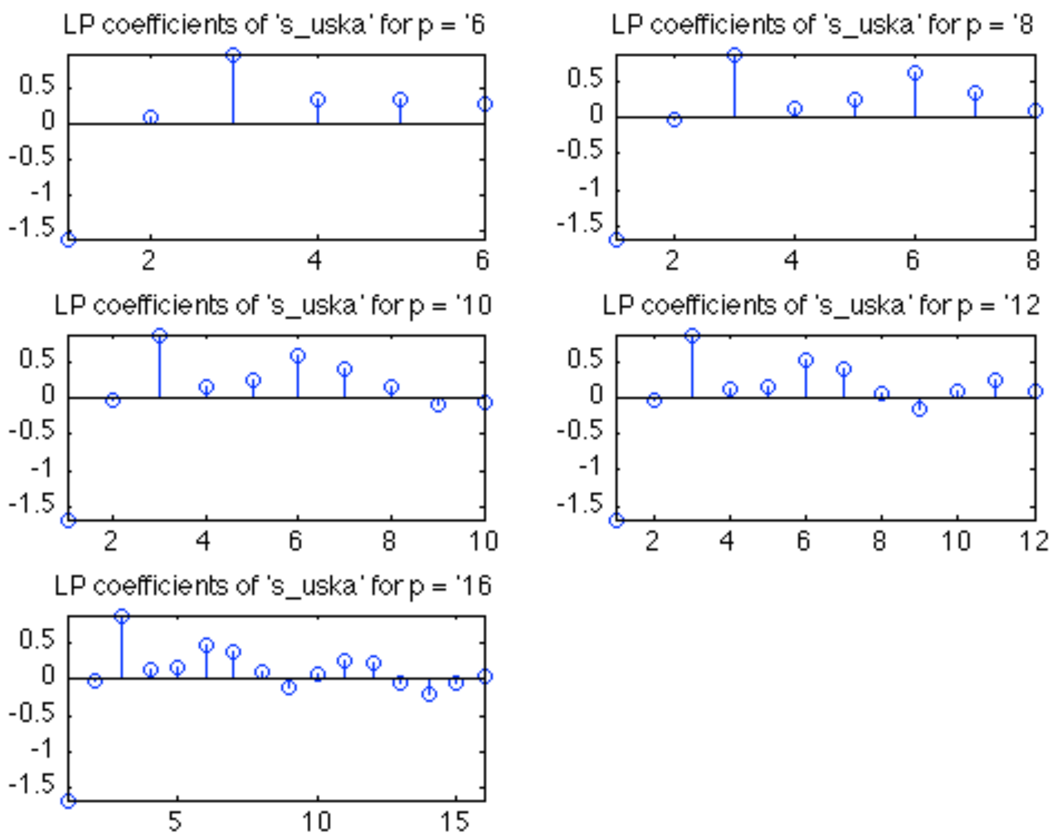


LP coefficients of 'n_pani' for p = '6
0.8
0.6
0.4
0.2
0
-0.2
-0.4
2
4
6
LP coefficients of 'n_pani' for p = '8
0.8
0.6
0.4
0.2
0
-0.2
2
4
6
8
LP coefficients of 'n_pani' for p = '10
0.8
0.6
0.4
0.2
0
-0.2
2
4
6
8
10
LP coefficients of 'n_pani' for p = '12
0.8
0.6
0.4
0.2
0
-0.2
2
4
6
8
10
12
LP coefficients of 'n_pani' for p = '16
0.8
0.6
0.4
0.2
0
-0.2
5
10
15


Autocorr coefficients of 'i_pani' for p = '6
Autocorr coefficients of 'i_pani' for p = '8
Autocorr coefficients of 'i_pani' for p = '10
Autocorr coefficients of 'i_pani' for p = '12
Autocorr coefficients of 'i_pani' for p = '16
0.15
0.1
0.05
0
2
4
6


LP coefficients of 'i_pani' for p = '6
LP coefficients of 'i_pani' for p = '8
LP coefficients of 'i_pani' for p = '10
LP coefficients of 'i_pani' for p = '12
LP coefficients of 'i_pani' for p = '16


Autocorr coefficients of 's_uska' for p = '6
Autocorr coefficients of 's_uska' for p = '8
Autocorr coefficients of 's_uska' for p = '10
Autocorr coefficients of 's_uska' for p = '12
Autocorr coefficients of 's_uska' for p = '16
10
0
-10
2
4
6



<u>The pole zero plots</u>

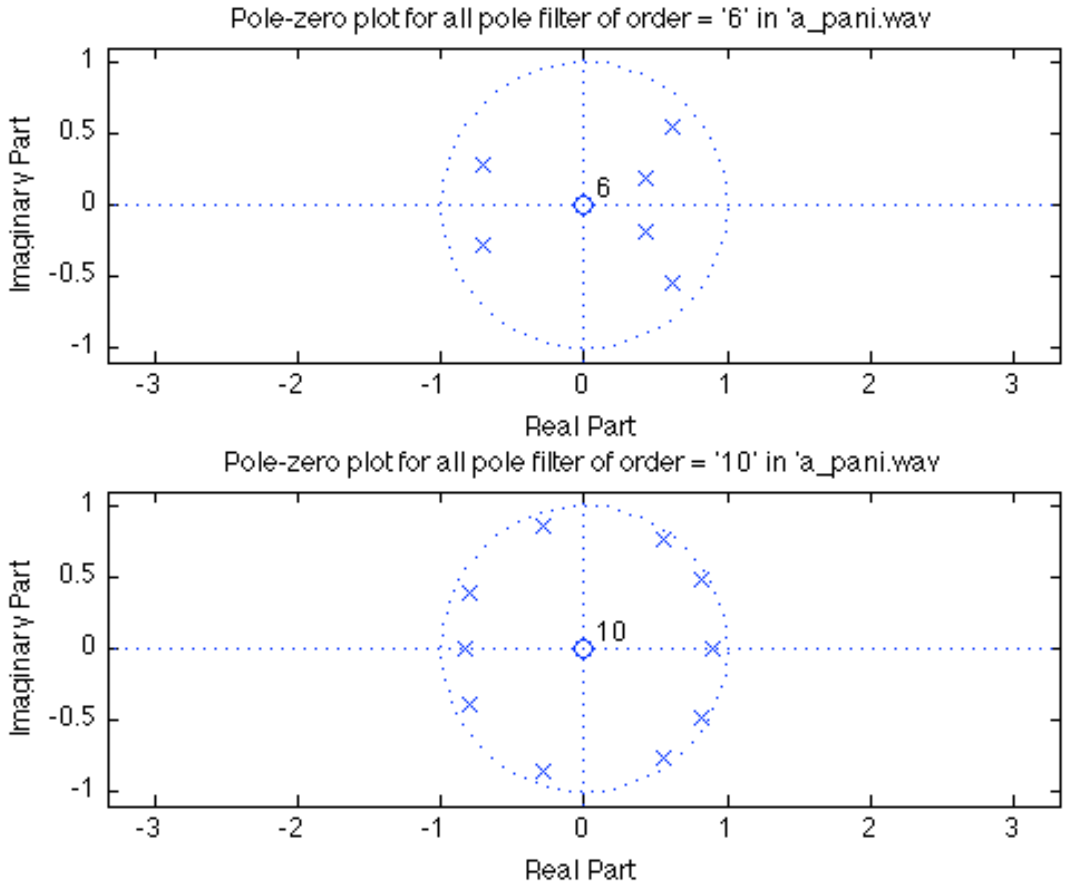
<u>The script:</u>

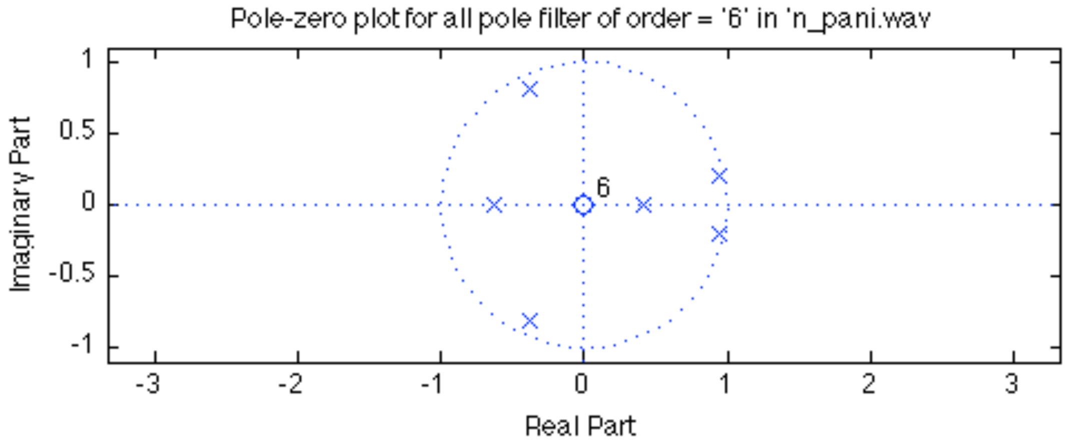
```
close all; clear all;
%% for /a/ in 'pani'

inputFile = 'a_pani.wav';
poleOrder1 = 6;
poleOrder2 = 10;
num = (1);

LPCoeffs1 = -getLPCoefficients(inputFile, poleOrder1);
LPCoeffsFull1 = (cat(1, (1), LPCoeffs1(:)))'
LPCoeffs2 = -getLPCoefficients(inputFile, poleOrder2);
LPCoeffsFull2 = (cat(1, (1), LPCoeffs2(:)))';

figure(100); clf;
```

```
subplot(2, 1, 1), zplane(num, LPCoeffsFull1);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder1), ''' in ''', inputFile, ''''],
'interpreter', 'none');
subplot(2, 1, 2), zplane(num, LPCoeffsFull2);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder2), ''' in ''', inputFile, ''''],
'interpreter', 'none');


%% for /n/ in 'pani'

inputFile = 'n_pani.wav';
poleOrder1 = 6;
poleOrder2 = 10;
num = (1);

LPCoeffs1 = -getLPCoefficients(inputFile, poleOrder1);
LPCoeffsFull1 = (cat(1, (1), LPCoeffs1(:)))'
LPCoeffs2 = -getLPCoefficients(inputFile, poleOrder2);
LPCoeffsFull2 = (cat(1, (1), LPCoeffs2(:)))';

figure(200); clf;
subplot(2, 1, 1), zplane(num, LPCoeffsFull1);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder1), ''' in ''', inputFile, ''''],
'interpreter', 'none');
subplot(2, 1, 2), zplane(num, LPCoeffsFull2);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder2), ''' in ''', inputFile, ''''],
'interpreter', 'none');

%% for /I/ in 'pani'

inputFile = 'i_pani.wav';
poleOrder1 = 6;
poleOrder2 = 10;
num = (1);

LPCoeffs1 = -getLPCoefficients(inputFile, poleOrder1);
LPCoeffsFull1 = (cat(1, (1), LPCoeffs1(:)))'
LPCoeffs2 = -getLPCoefficients(inputFile, poleOrder2);
LPCoeffsFull2 = (cat(1, (1), LPCoeffs2(:)))';

figure(300); clf;
subplot(2, 1, 1), zplane(num, LPCoeffsFull1);
```

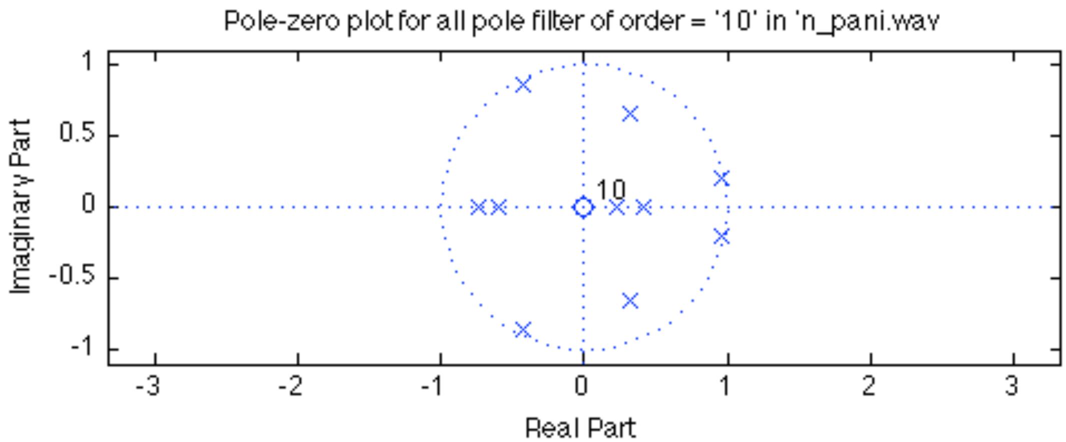
```
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder1), ''' in ''', inputFile, ''''],
'interpreter', 'none');
subplot(2, 1, 2), zplane(num, LPCoeffsFull2);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder2), ''' in ''', inputFile, ''''],
'interpreter', 'none');

%% for /s/ in 'uska'

inputFile = 's_uska.wav';
poleOrder1 = 6;
poleOrder2 = 10;
num = (1);

LPCoeffs1 = -getLPCoefficients(inputFile, poleOrder1);
LPCoeffsFull1 = (cat(1, (1), LPCoeffs1(:)))'
LPCoeffs2 = -getLPCoefficients(inputFile, poleOrder2);
LPCoeffsFull2 = (cat(1, (1), LPCoeffs2(:)))';

figure(400); clf;
subplot(2, 1, 1), zplane(num, LPCoeffsFull1);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder1), ''' in ''', inputFile, ''''],
'interpreter', 'none');
subplot(2, 1, 2), zplane(num, LPCoeffsFull2);
title(['Pole-zero plot for all pole filter of order = ''',
num2str(poleOrder2), ''' in ''', inputFile, ''''],
'interpreter', 'none');
```

The plots:

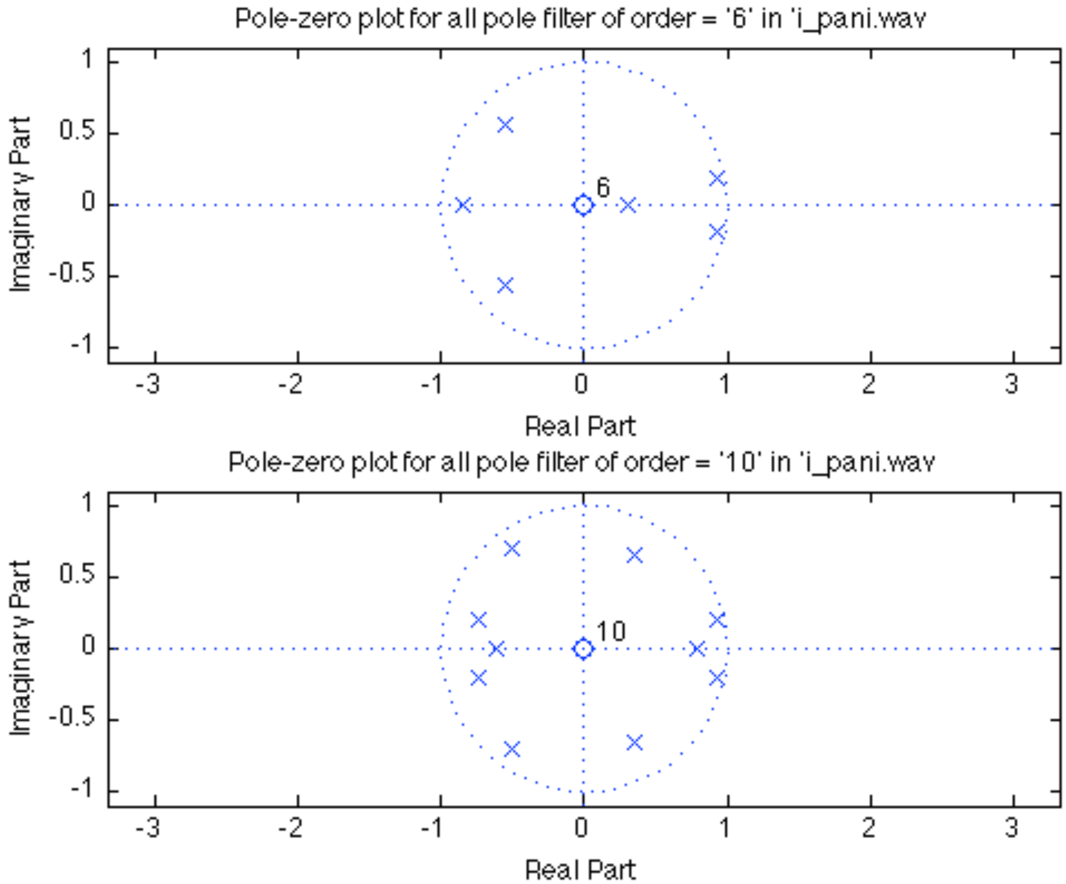

Pole-zero plot for all pole filter of order = '6' in 'a_pani.wav
Imaginary Part
Real Part
6
Pole-zero plot for all pole filter of order = '10' in 'a_pani.wav
Imaginary Part
Real Part
10



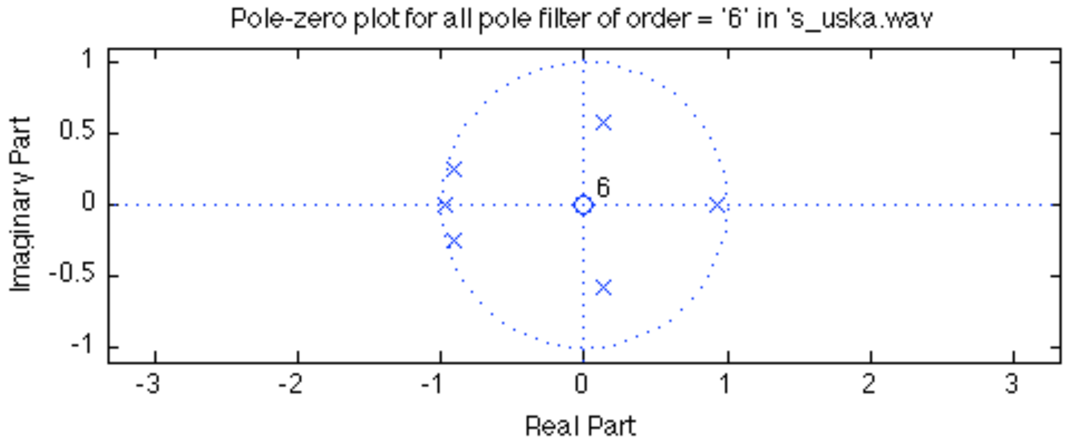
Pole-zero plot for all pole filter of order = '6' in 'n_pani.wav
Imaginary Part
Real Part
6
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3

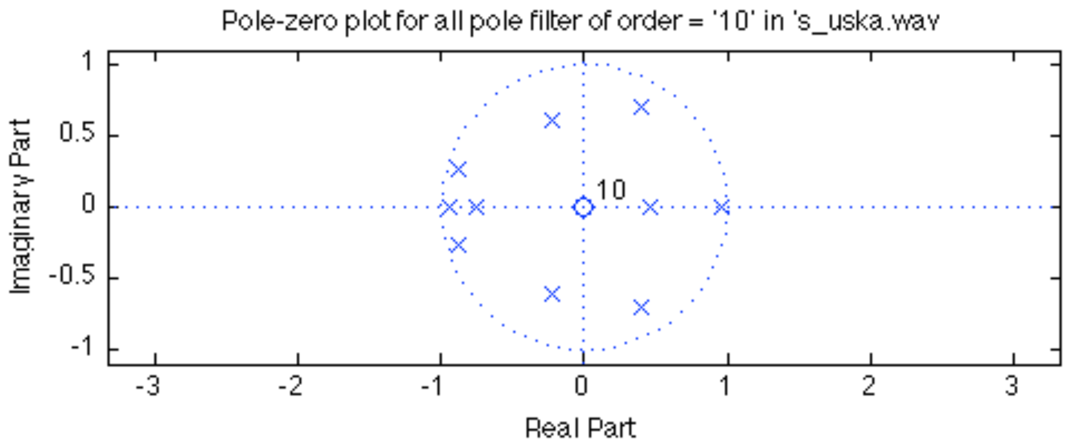


Pole-zero plot for all pole filter of order = '10' in 'n_pani.wav
Imaginary Part
Real Part
10
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3



Pole-zero plot for all pole filter of order = '6' in 'i_pani.wav
Imaginary Part
Real Part
6
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3
Pole-zero plot for all pole filter of order = '10' in 'i_pani.wav
Imaginary Part
Real Part
10





Answer to Q2(b)

The script:

```
close all; clear all;
poleOrder = [6 8 10 12 16];
siz = size(poleOrder(:));
length = siz(1);

for k = 1:length;
    getHammingNarrowbandSpectrum('a_pani.wav', poleOrder(k));
end

for k = 1:length;
    getHammingNarrowbandSpectrum('n_pani.wav', poleOrder(k));
end
```

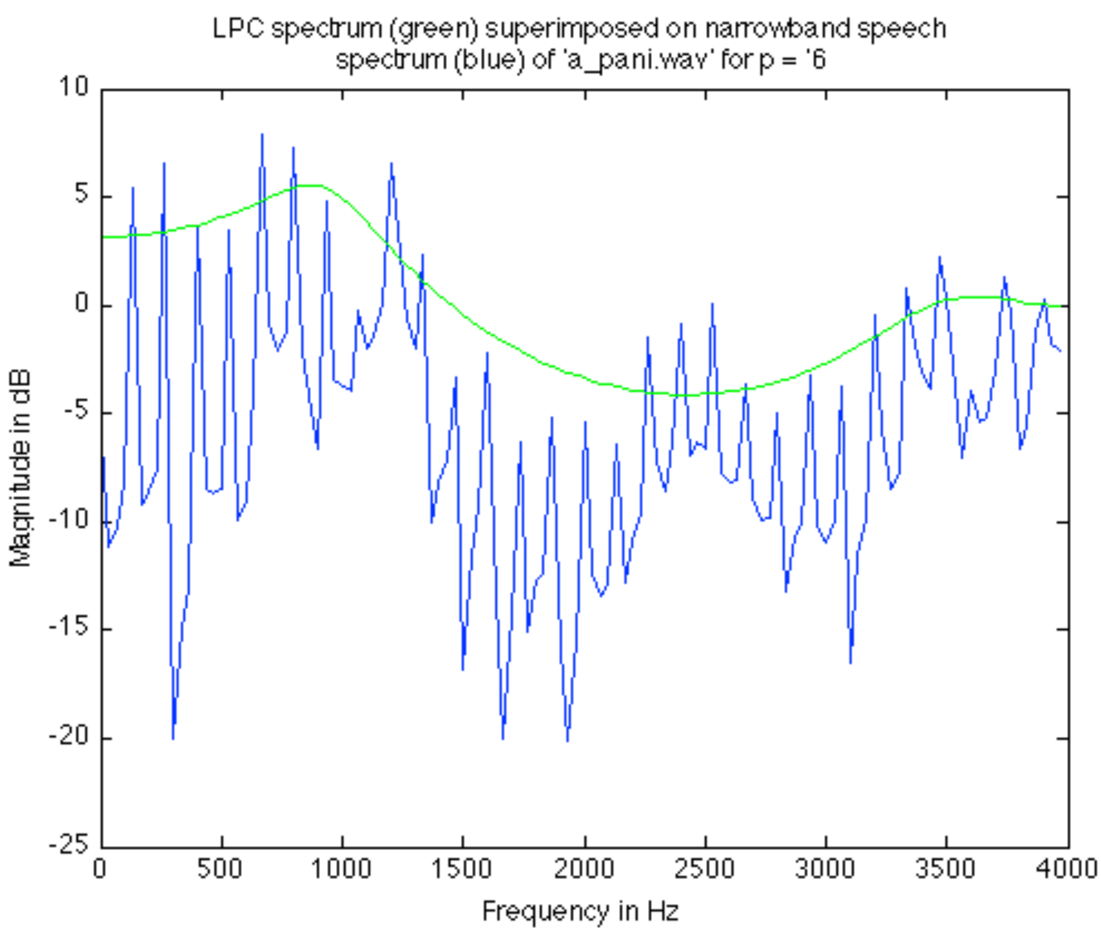
```
for k = 1:length;
    getHammingNarrowbandSpectrum('i_pani.wav', poleOrder(k));
end

for k = 1:length;
    getHammingNarrowbandSpectrum('s_uska.wav', poleOrder(k));
end
```

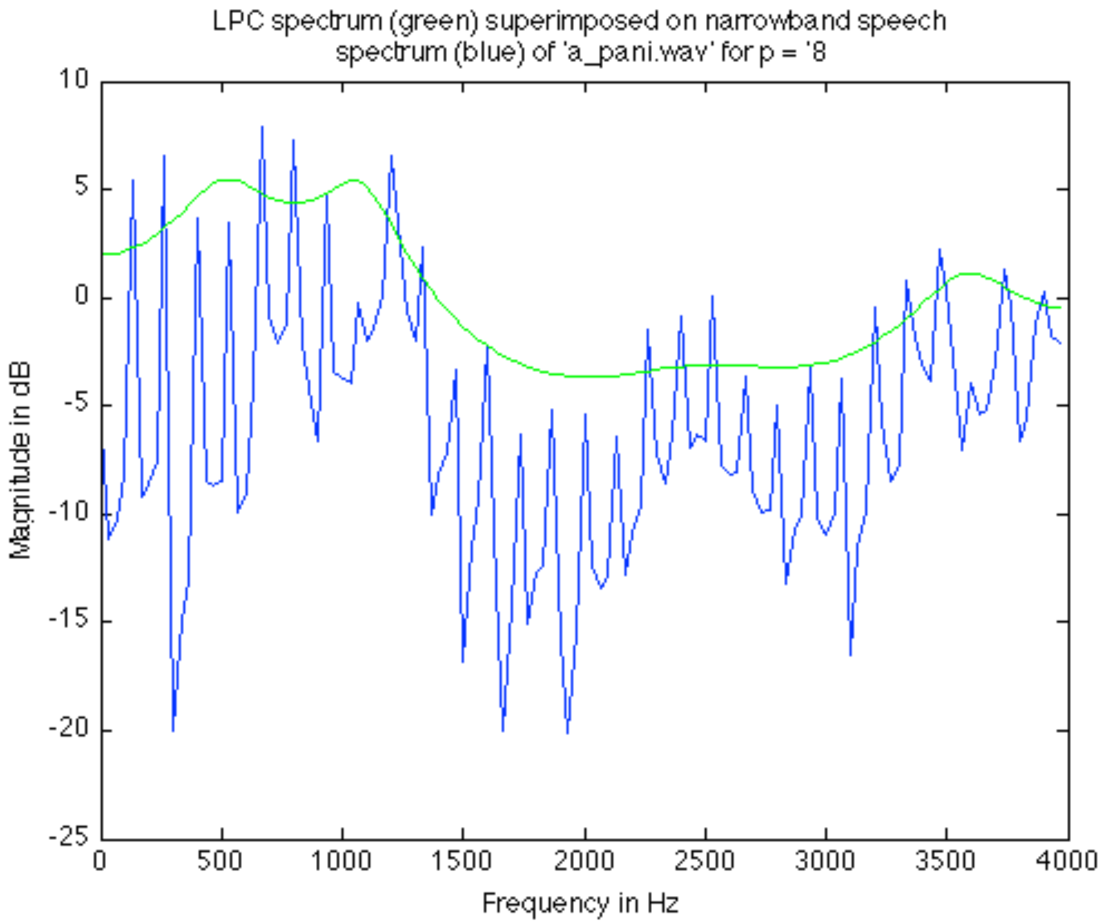
The function:

```
function [narrowbandSpectrum, LPCSpectrum, w ]=
getHammingNarrowbandSpectrum(inputFile, poleOrder)

% poleOrder = 6;
% inputFile = 'a_pani.wav';

windowDuration = 0.030; % in ms

[y, fs] = preEmphasize(inputFile);

siz = size(y(:));
length = siz(1);
centralIndex = round(length/2);
M = round(windowDuration * fs);
startIndex = round(centralIndex - M/2);

windowedSignal = y(startIndex:startIndex + M-1);

narrowbandSpectrum = abs(fft(windowedSignal));
narrowbandSpectrum = narrowbandSpectrum(1:round(M/2));
narrowbandSpectrum = 10 * log10(abs(narrowbandSpectrum));

% getting the LPC spectrum

LPCoeffs = getLPCoefficients(inputFile, poleOrder);
frequencies = (fs/M) * (0:M-1);

denominator = 1;
numerator = 1;

for k = 1:poleOrder
    denominator = denominator - (LPCoeffs(k) * (exp(-1i * 2 * pi
* frequencies ./ fs) .^ k));
end
```

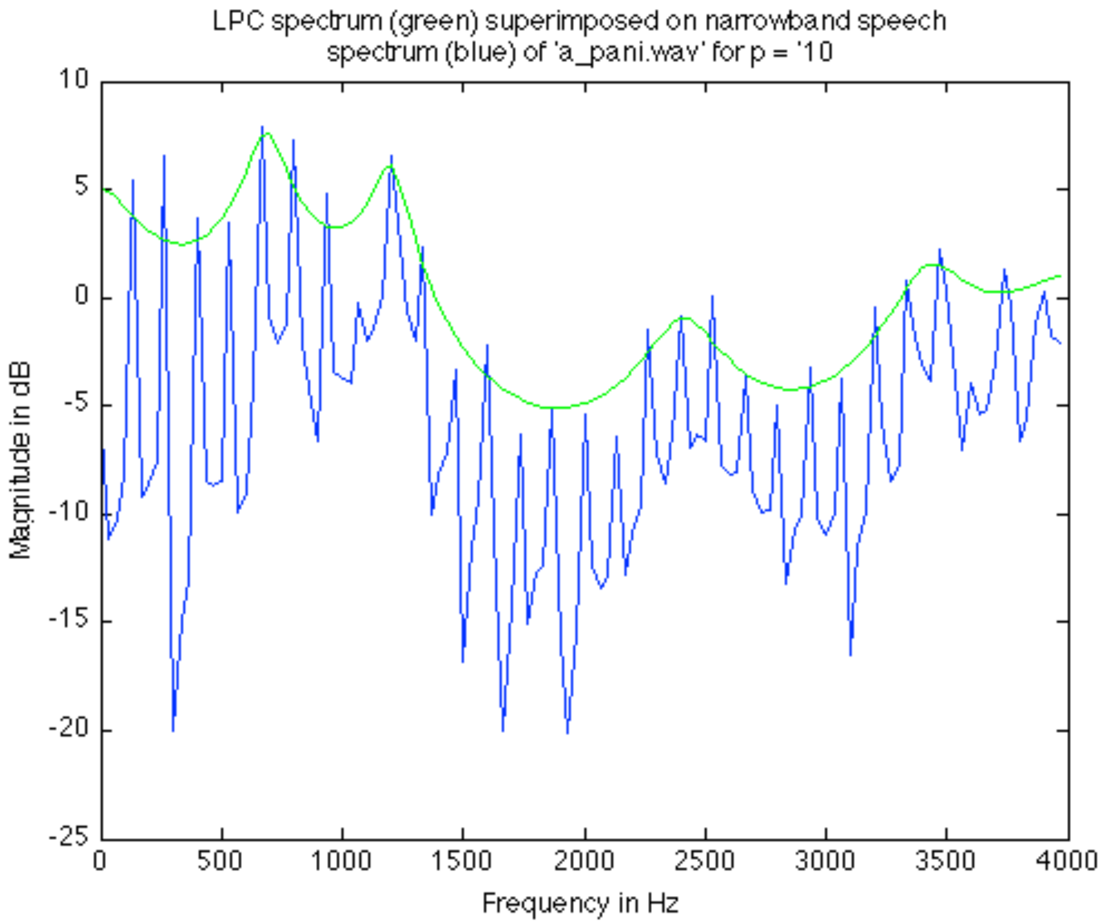
```
H = numerator ./ denominator;
LPCSpectrum = 10 * log10(abs(H));
LPCSpectrum = LPCSpectrum(1:round(M/2));
w = frequencies(1:round(M/2));

figure, plot(w, narrowbandSpectrum);
hold on;
linePlot = plot(w, LPCSpectrum);
set(linePlot,'Color','green');
hold off;
title({'LPC spectrum (green) superimposed on narrowband
speech'; ...
    ['spectrum (blue) of ''', inputFile, ''' for p = ''', ...
    num2str(poleOrder), '''']}, 'interpreter', 'none');
xlabel('Frequency in Hz');
ylabel('Magnitude in dB');

end
```

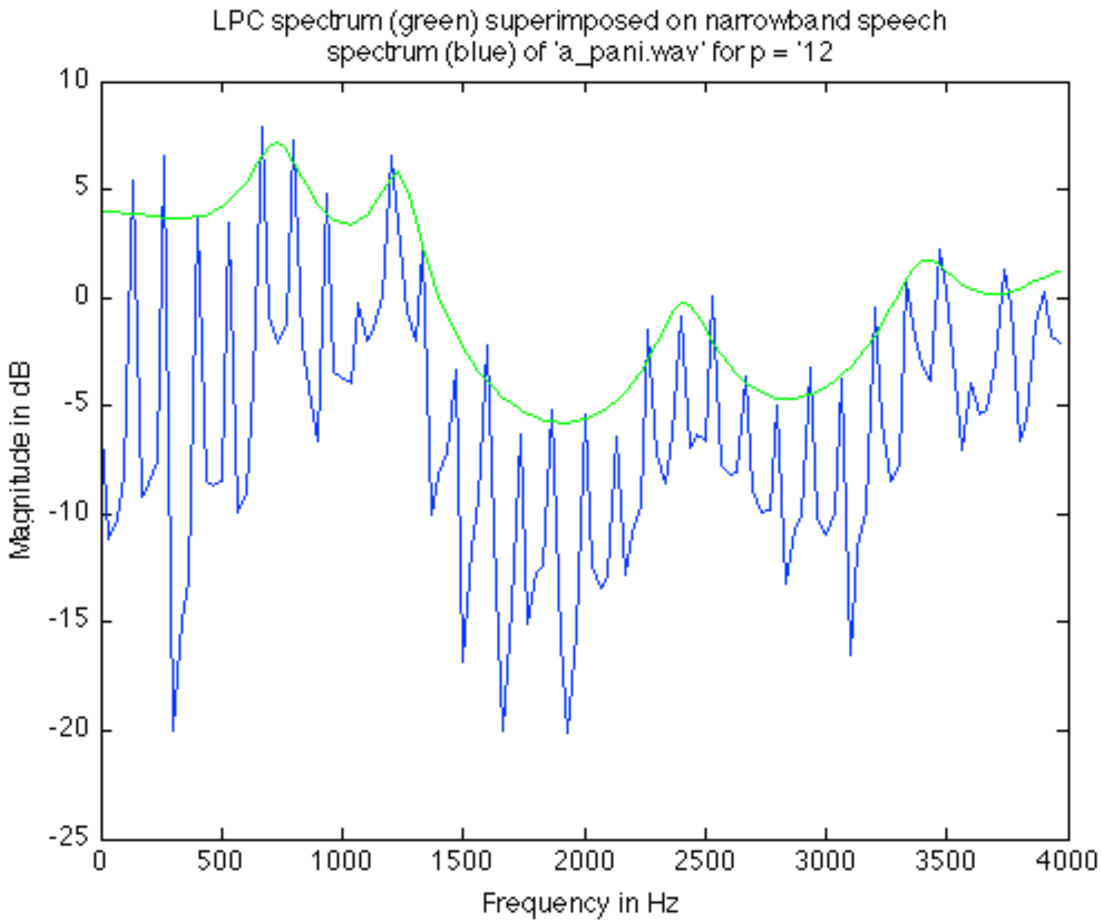
<u>The plots:</u>

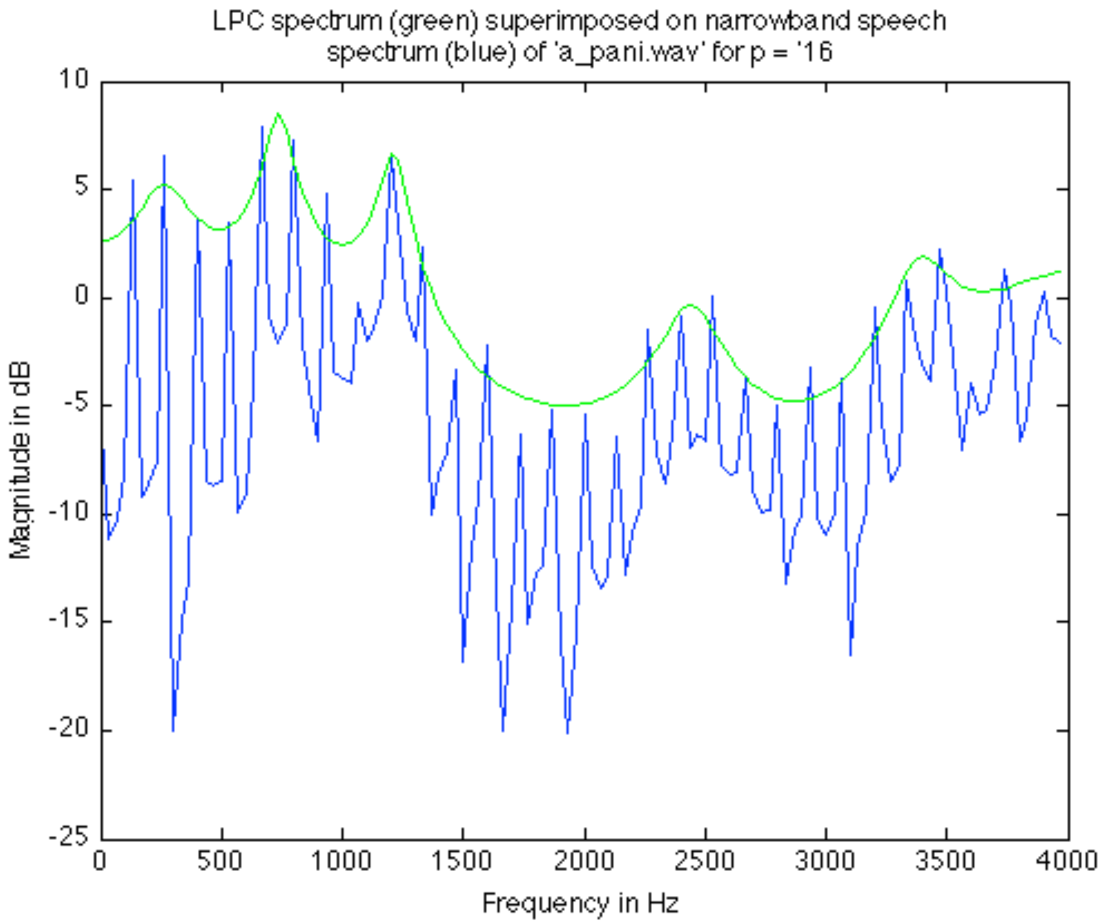


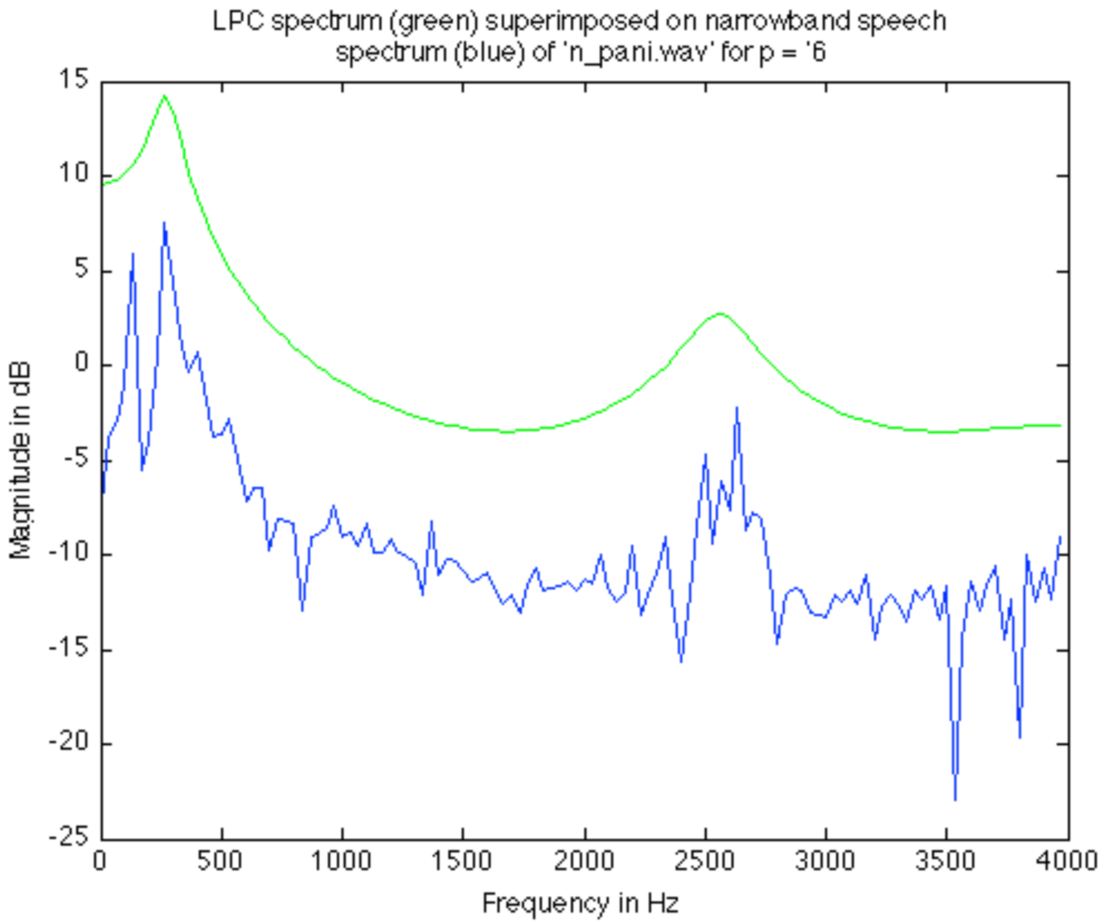


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'a_pani.wav' for p = '8
Magnitude in dB
Frequency in Hz
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000

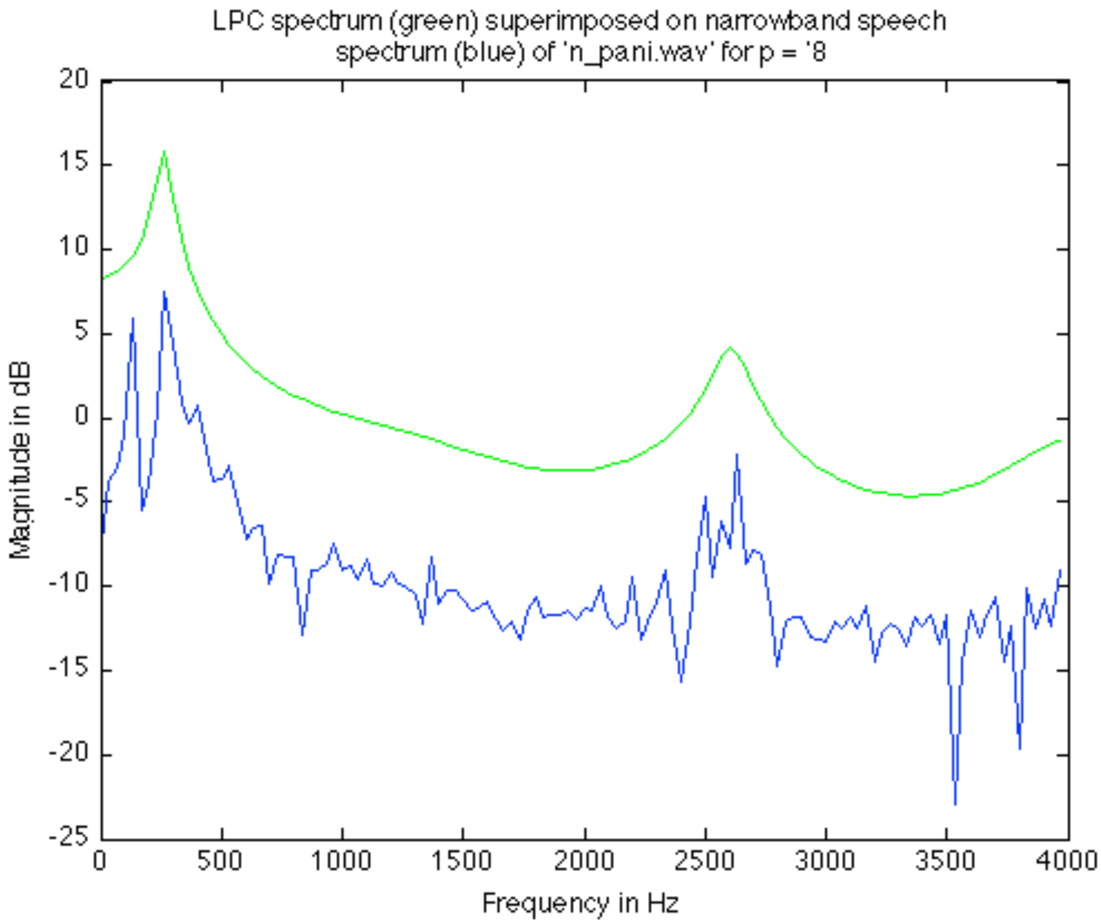

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'a_pani.wav' for p = '10
Magnitude in dB
Frequency in Hz
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'a_pani.wav' for p = '12
Magnitude in dB
Frequency in Hz
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000

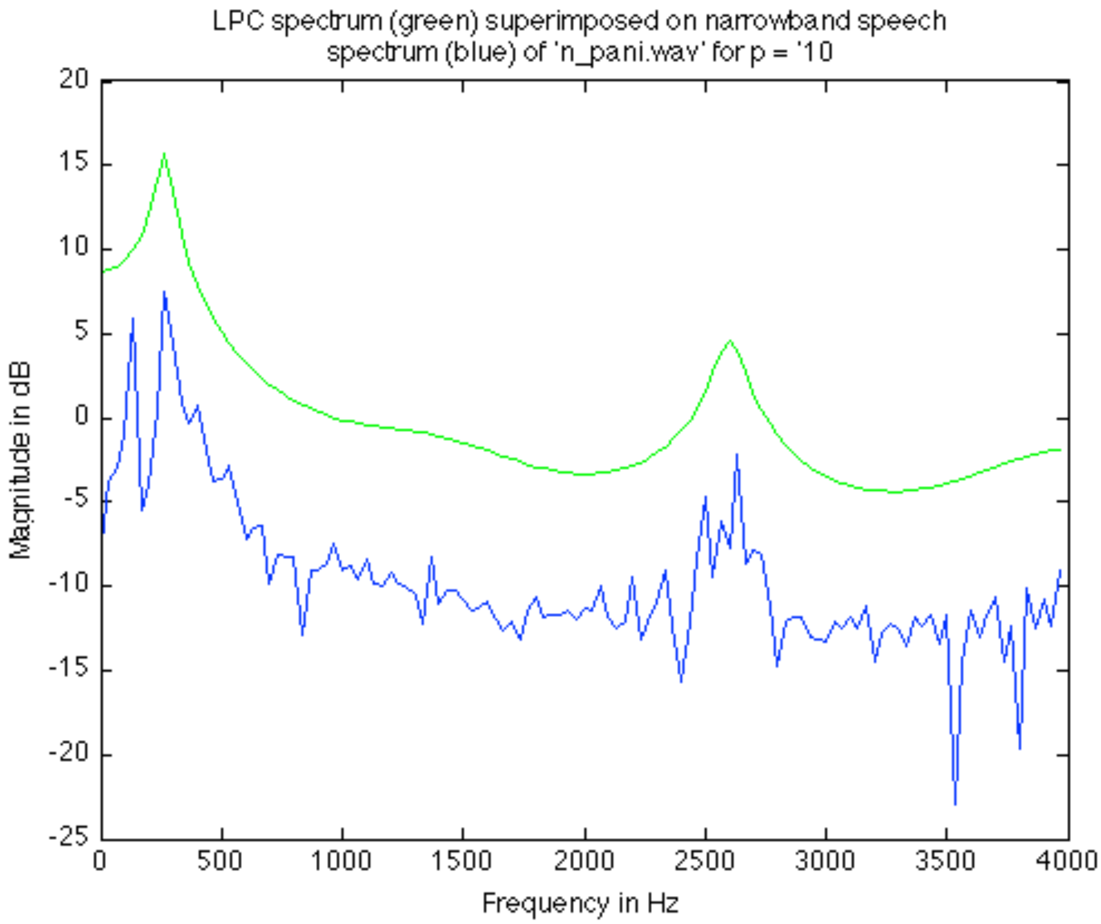

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'a_pani.wav' for p = '16
Magnitude in dB
Frequency in Hz
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000

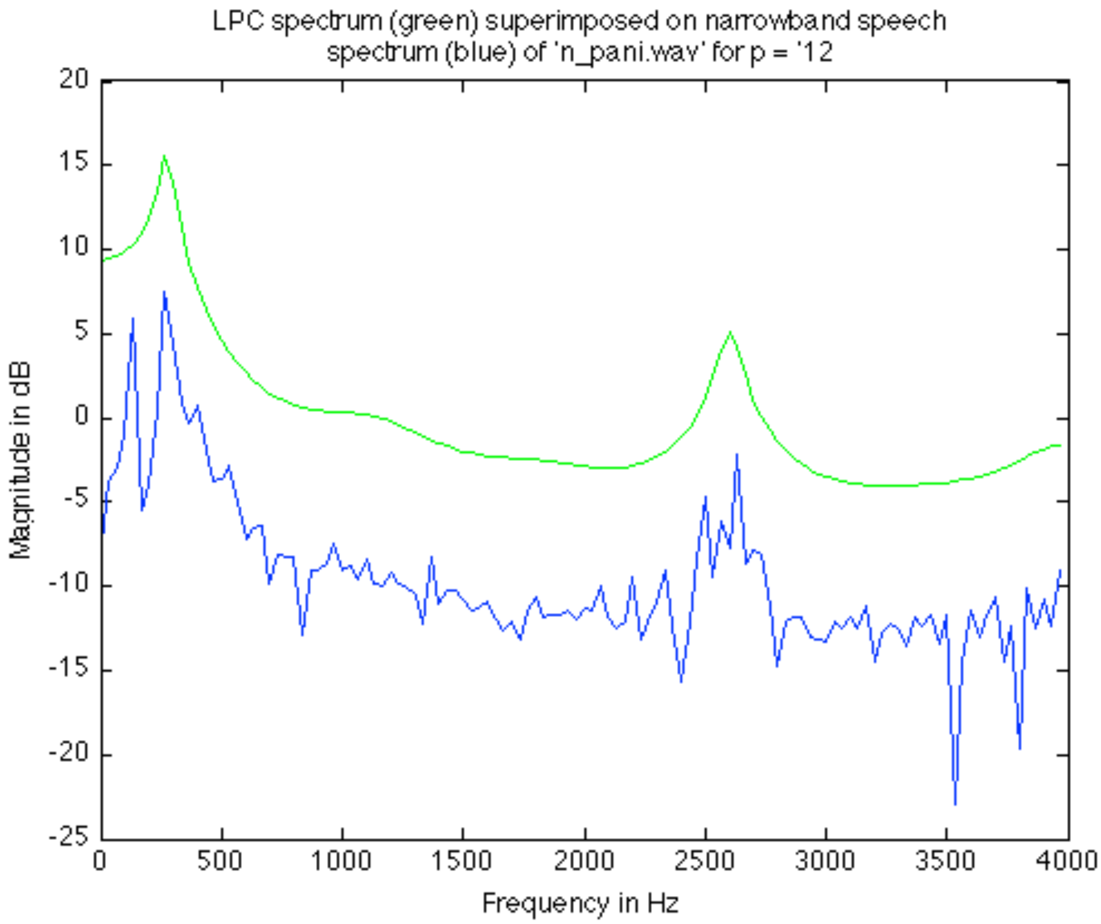


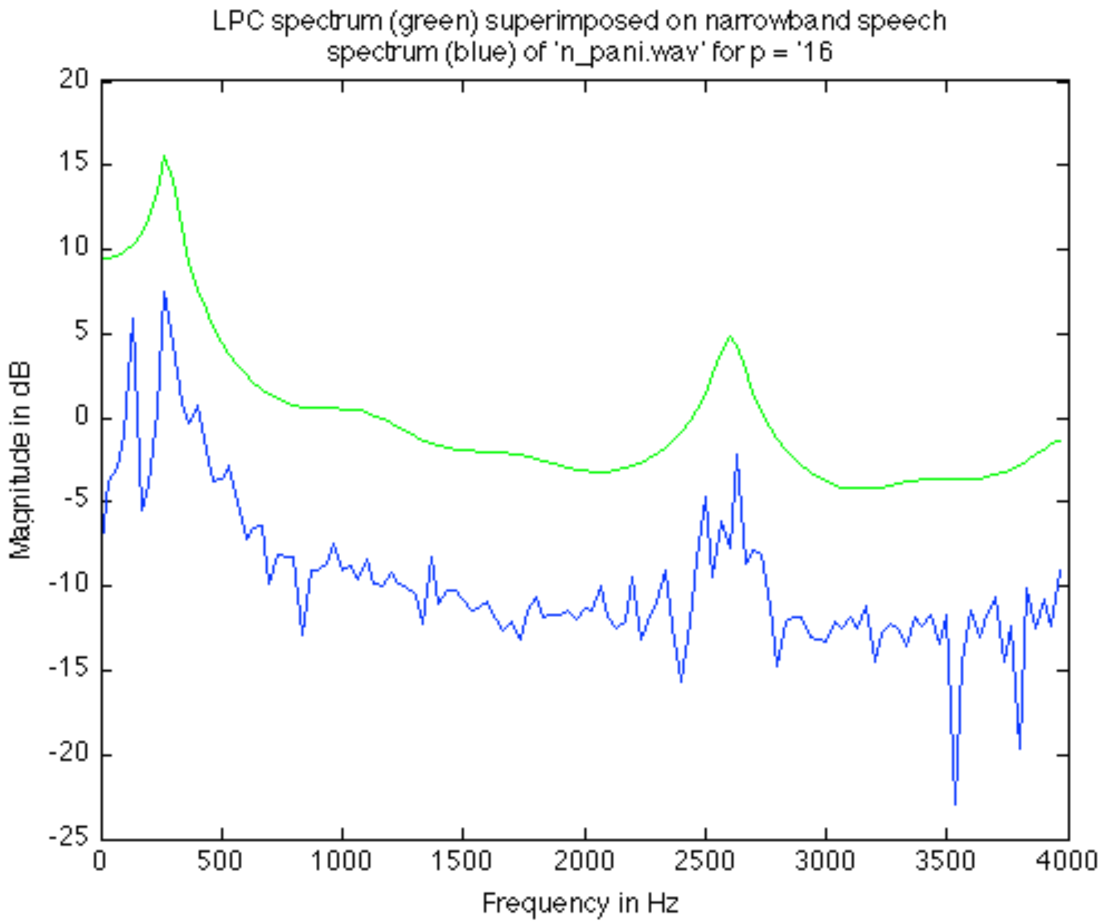
LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'n_pani.wav' for p = '6
Magnitude in dB
Frequency in Hz
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000



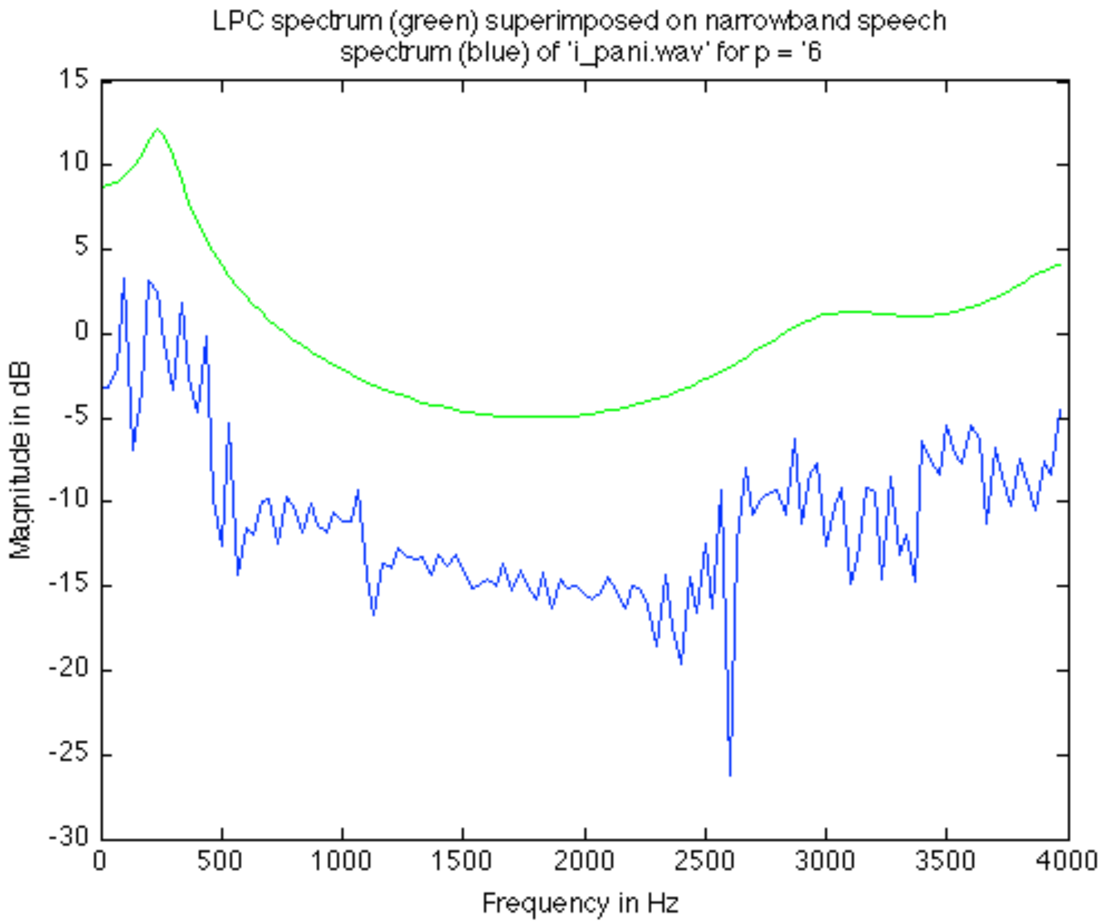
LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'n_pani.wav' for p = '8
Magnitude in dB
Frequency in Hz
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'n_pani.wav' for p = '10
Magnitude in dB
Frequency in Hz
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000

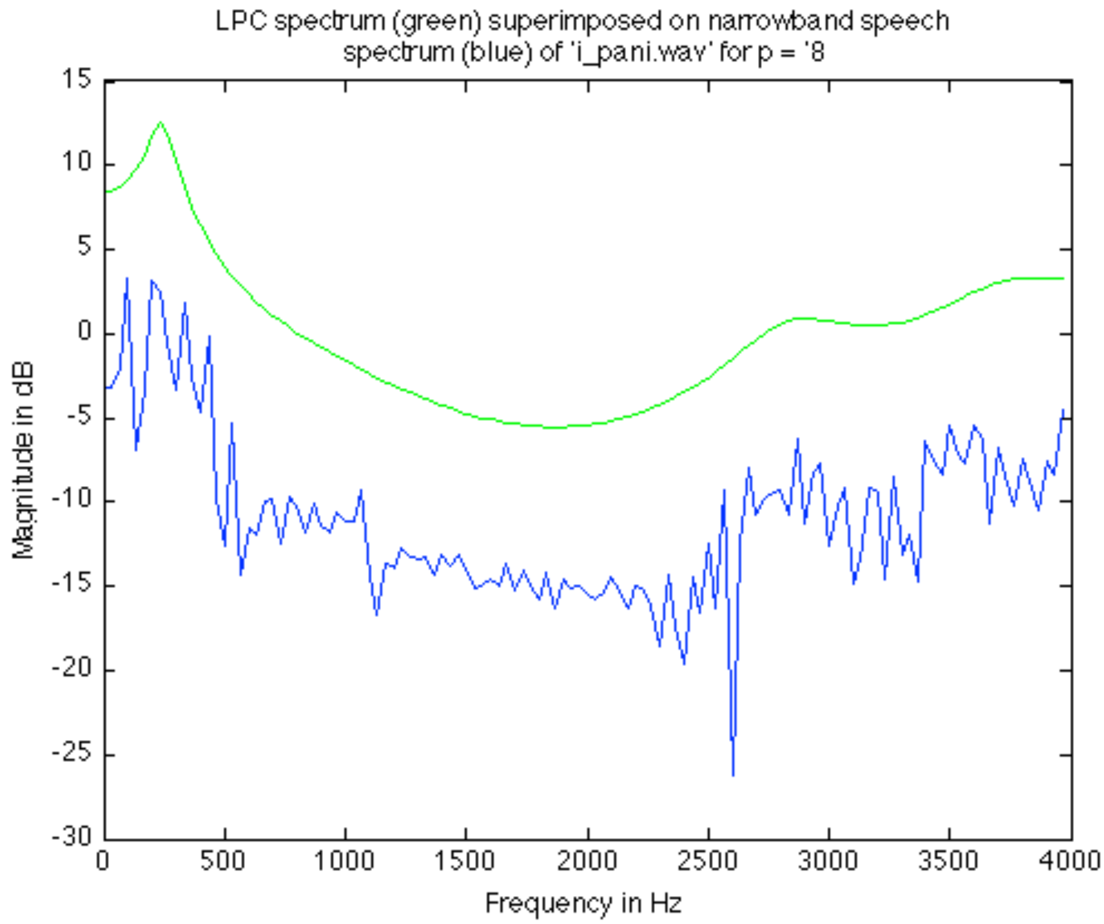

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'n_pani.wav' for p = '12
Magnitude in dB
Frequency in Hz
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000

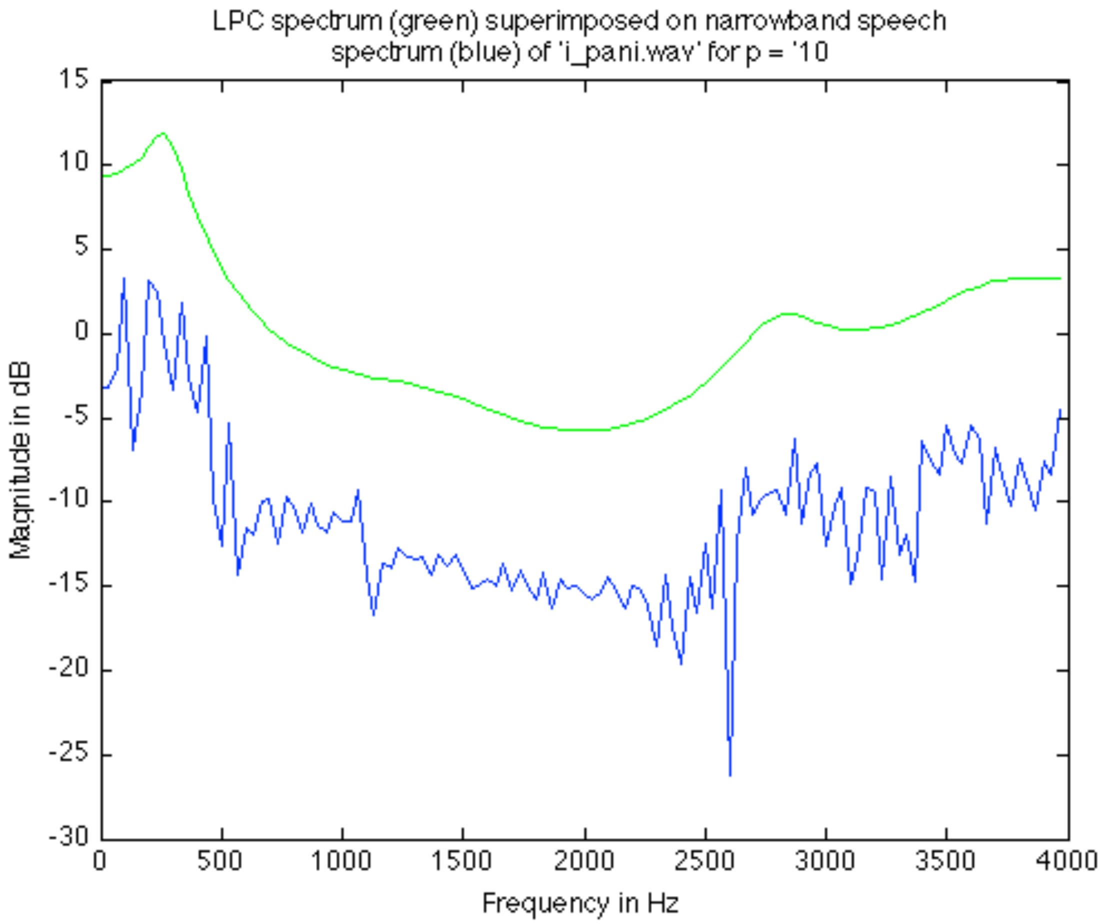


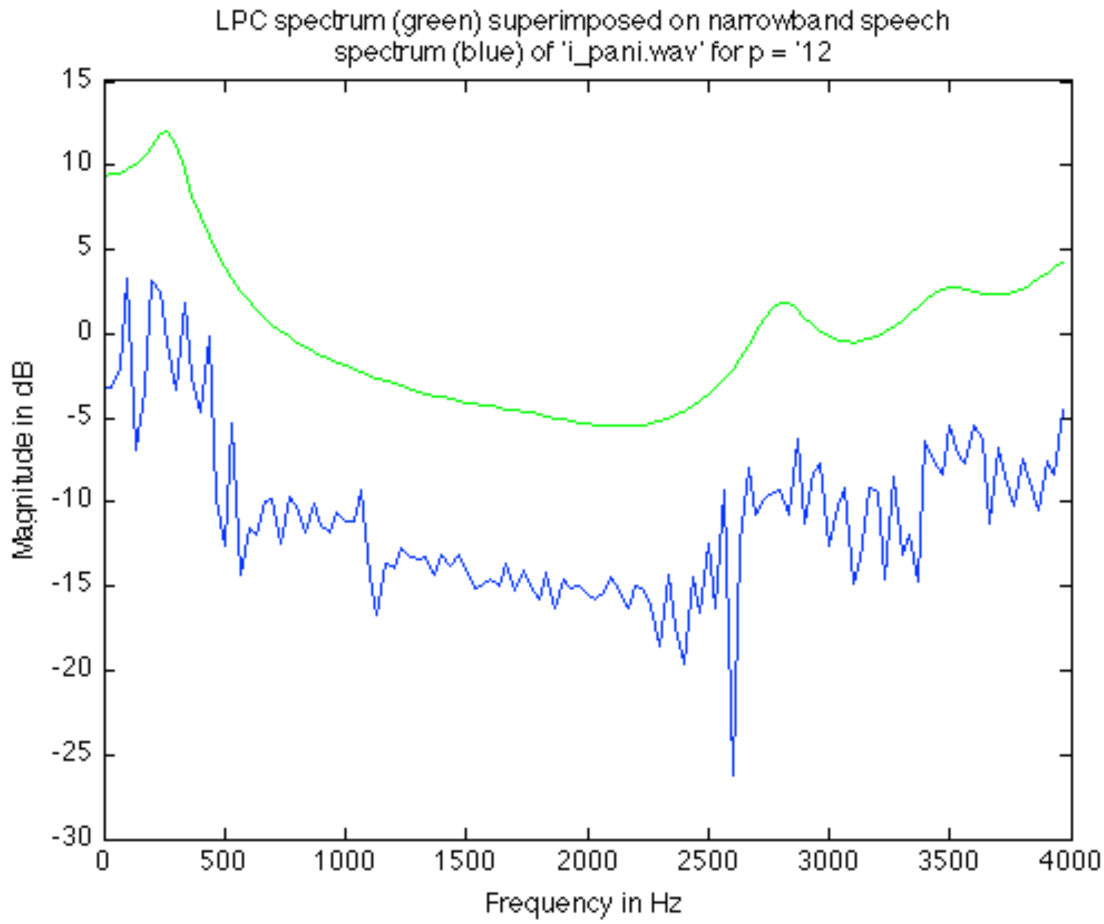
LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'n_pani.wav' for p = '16
Magnitude in dB
Frequency in Hz
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000

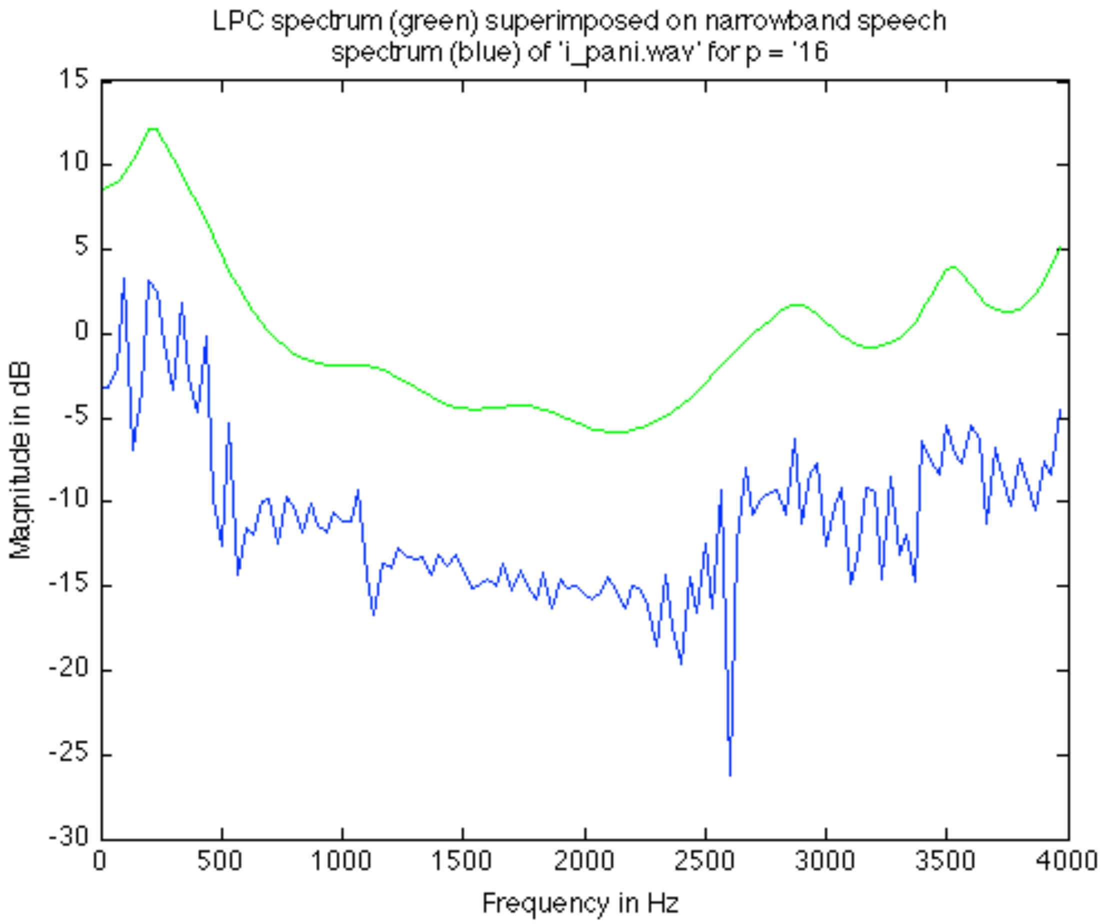


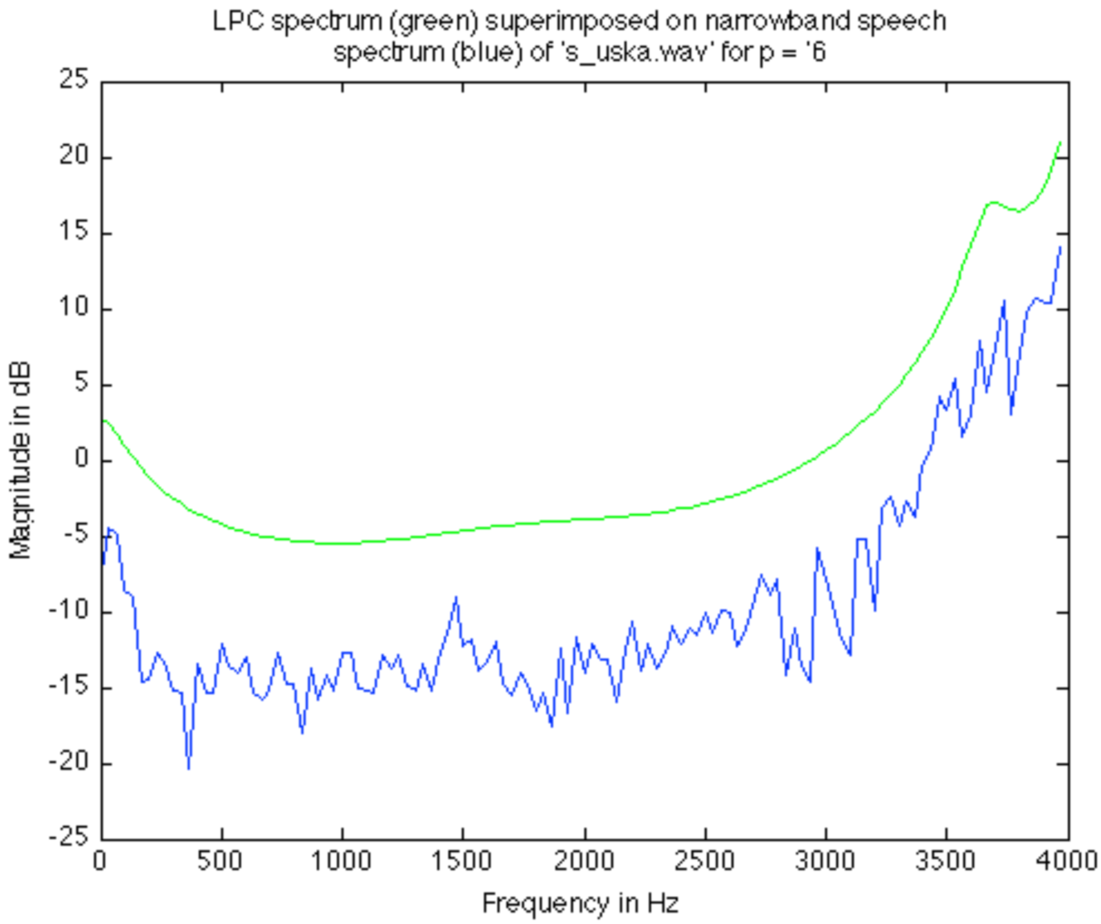
LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'i_pani.wav' for p = '6
15
10
5
0
-5
-10
-15
-20
-25
-30
Magnitude in dB
0
500
1000
1500
2000
2500
3000
3500
4000
Frequency in Hz

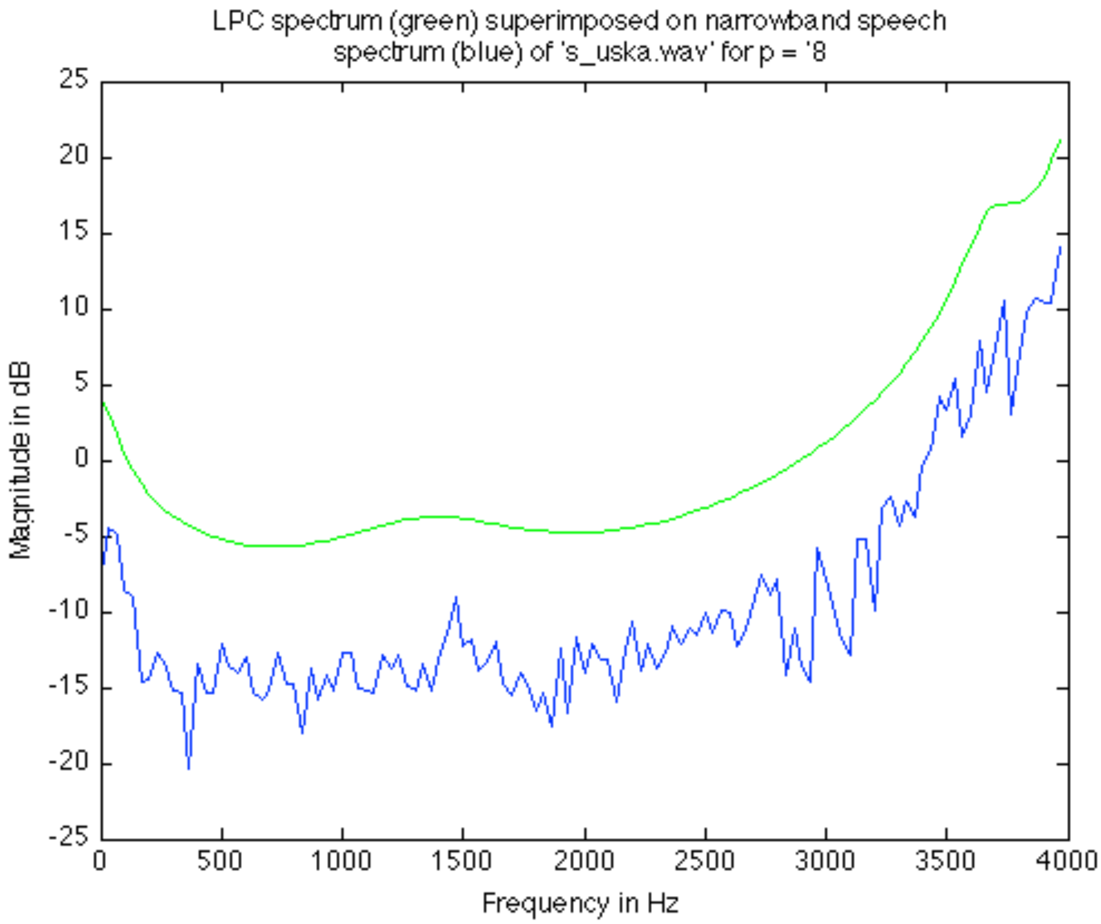

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'i_pani.wav' for p = '8
Magnitude in dB
Frequency in Hz
15
10
5
0
-5
-10
-15
-20
-25
-30
0
500
1000
1500
2000
2500
3000
3500
4000

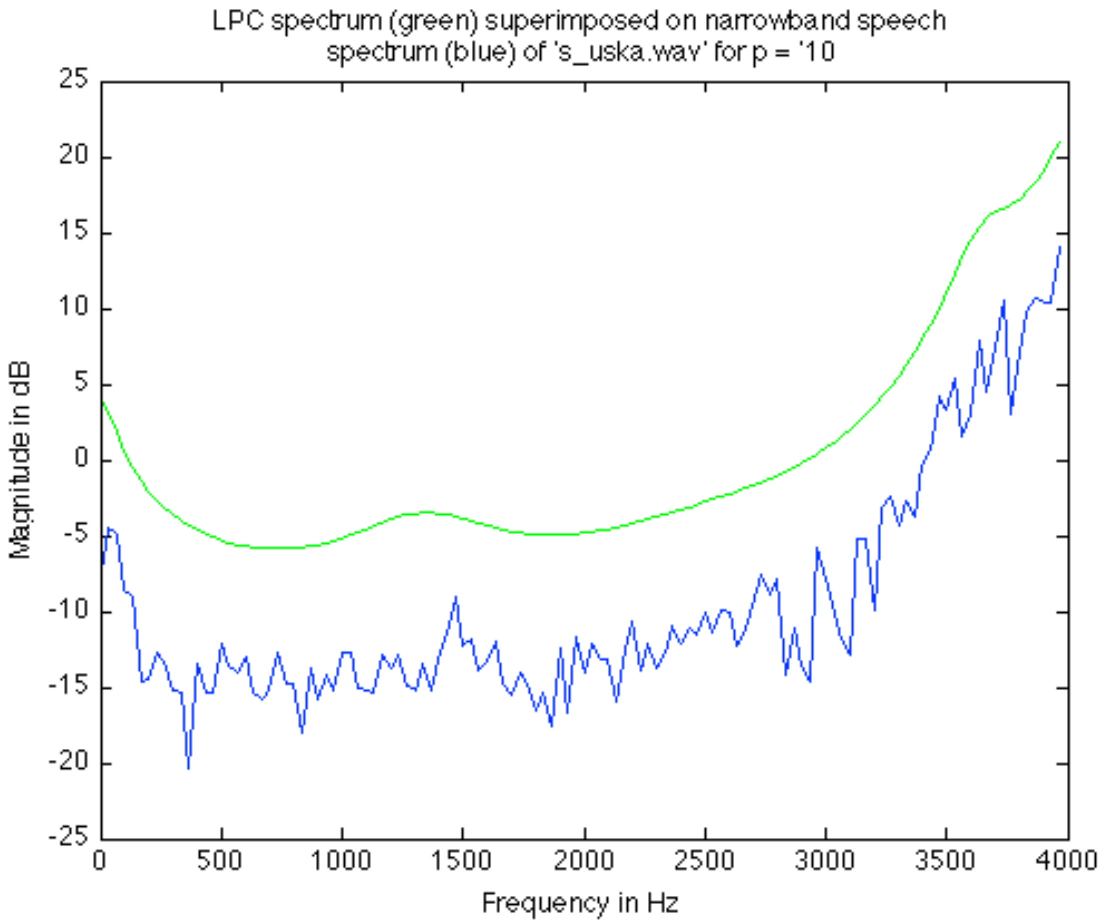

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'i_pani.wav' for p = '10
Magnitude in dB
Frequency in Hz
15
10
5
0
-5
-10
-15
-20
-25
-30
0
500
1000
1500
2000
2500
3000
3500
4000

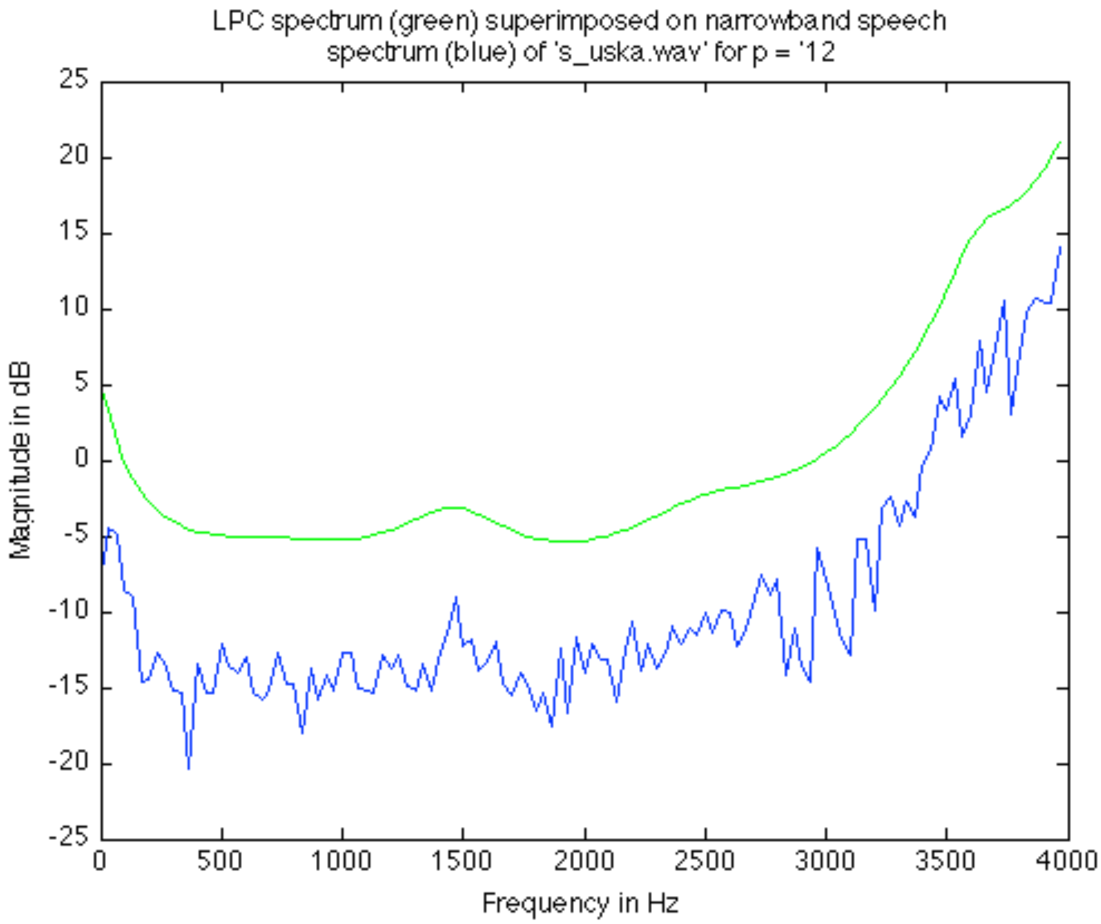

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'i_pani.wav' for p = '12
Magnitude in dB
Frequency in Hz
15
10
5
0
-5
-10
-15
-20
-25
-30
0
500
1000
1500
2000
2500
3000
3500
4000

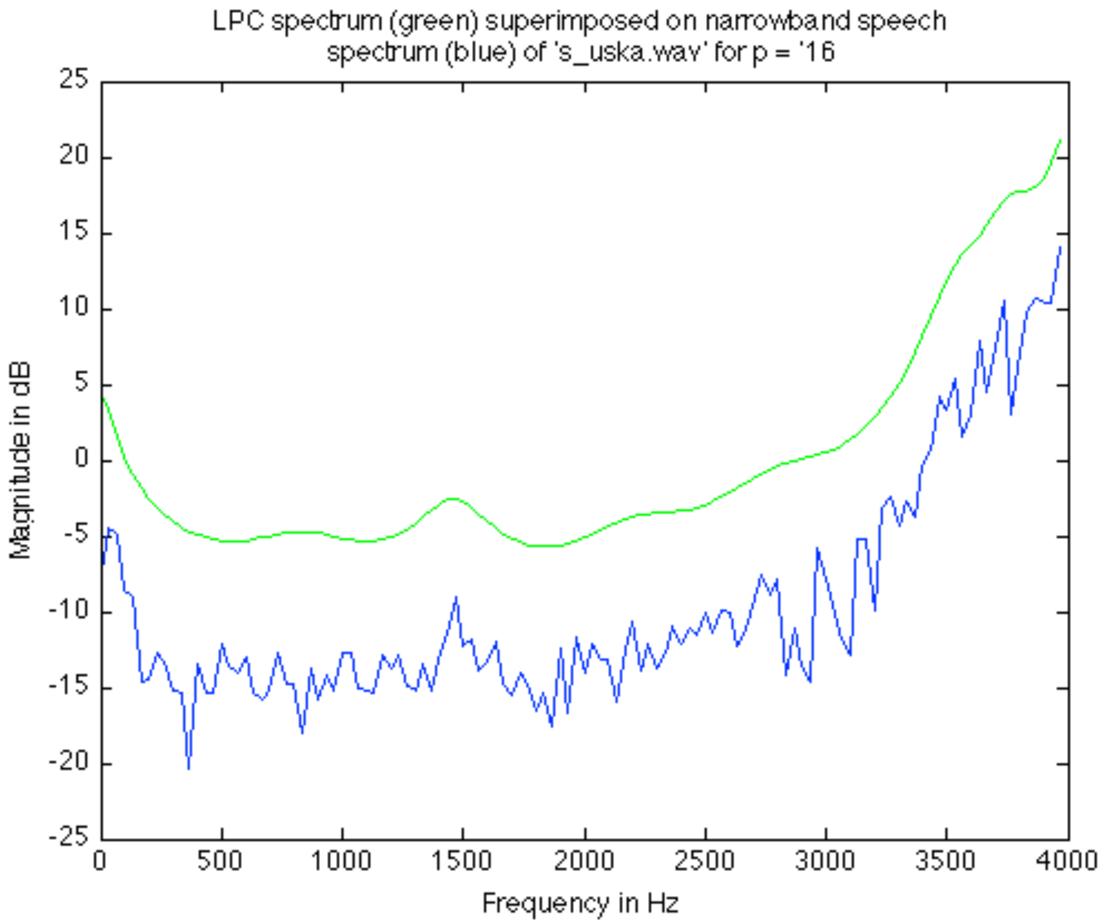

LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 'i_pani.wav' for p = '16
Magnitude in dB
Frequency in Hz
15
10
5
0
-5
-10
-15
-20
-25
-30
0
500
1000
1500
2000
2500
3000
3500
4000


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 's_uska.wav' for p = '6
Magnitude in dB
25
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000
Frequency in Hz


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 's_uska.wav' for p = '8
Magnitude in dB
Frequency in Hz
25
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 's_uska.wav' for p = '10
Magnitude in dB
Frequency in Hz
25
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000


LPC spectrum (green) superimposed on narrowband speech
spectrum (blue) of 's_uska.wav' for p = '12
Magnitude in dB
Frequency in Hz
25
20
15
10
5
0
-5
-10
-15
-20
-25
0
500
1000
1500
2000
2500
3000
3500
4000



The speech spectrum (in blue) follows the LPC spectrum (in green) closely. This means the ratio of their magnitudes lie close to 1 reducing the overall error to a minimum. The parts of the speech spectrum above the LPC spectrum contribute more to the error than the portions below (the valleys). For example |2-1| = 1 is greater than |1-(1/2)| = 1/2.

Answer to Q2(c)

The script:

```
close all; clear all;
addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/

poleOrders = [6 8 10 12 16];

plotResidualEnergyVersusP('a_pani.wav', poleOrders);
plotResidualEnergyVersusP('n_pani.wav', poleOrders);
```

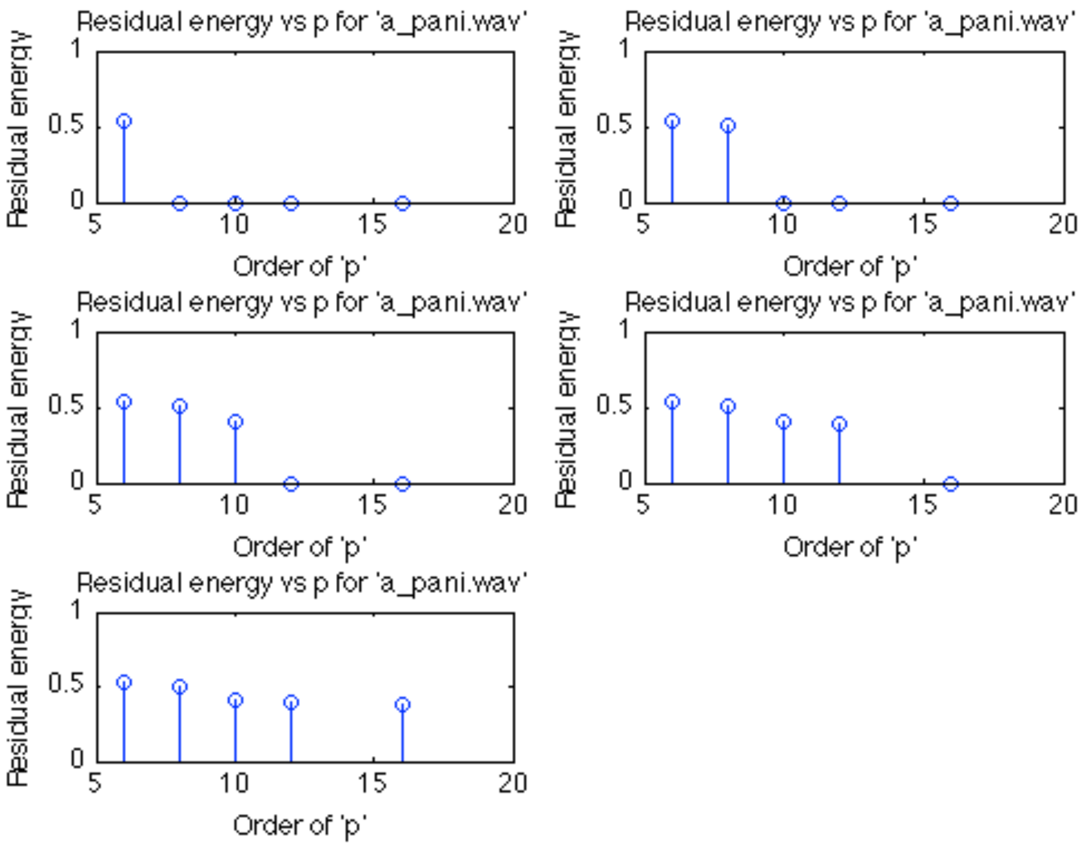
```
plotResidualEnergyVersusP('i_pani.wav', poleOrders);
plotResidualEnergyVersusP('s_uska.wav', poleOrders);


rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/
```

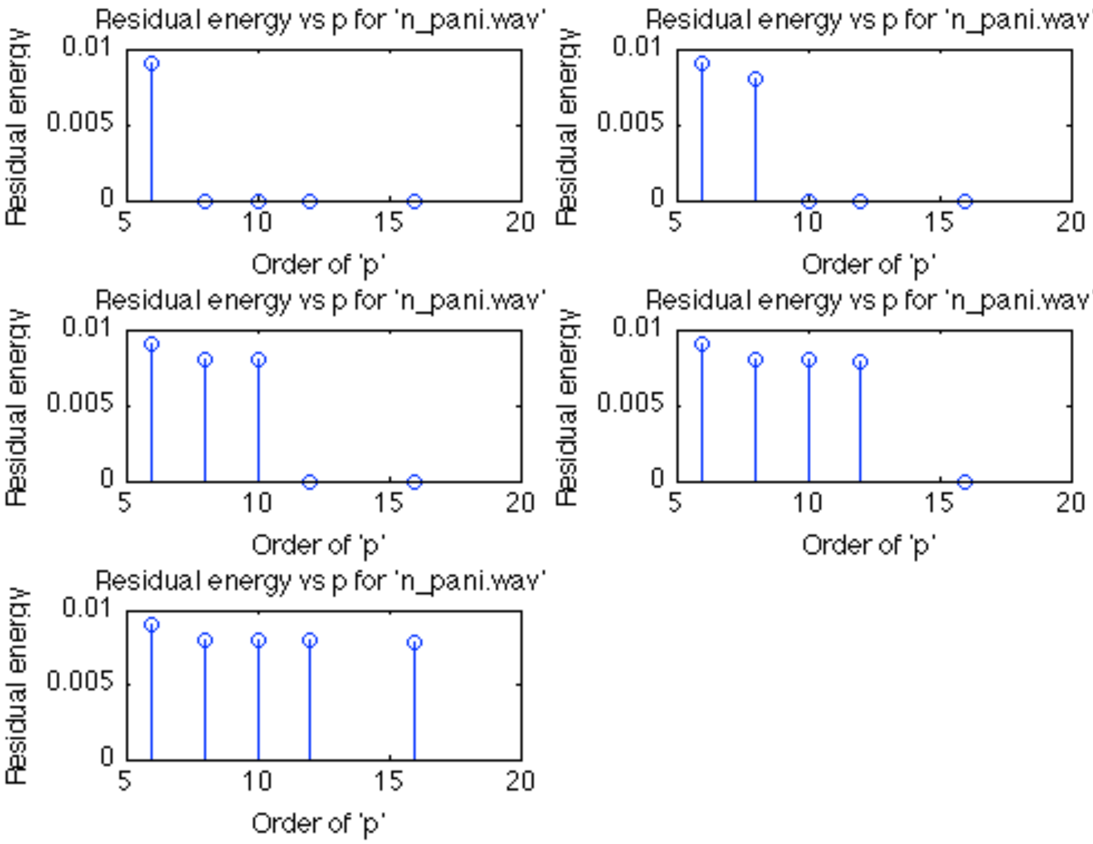
<u>The functions:</u>

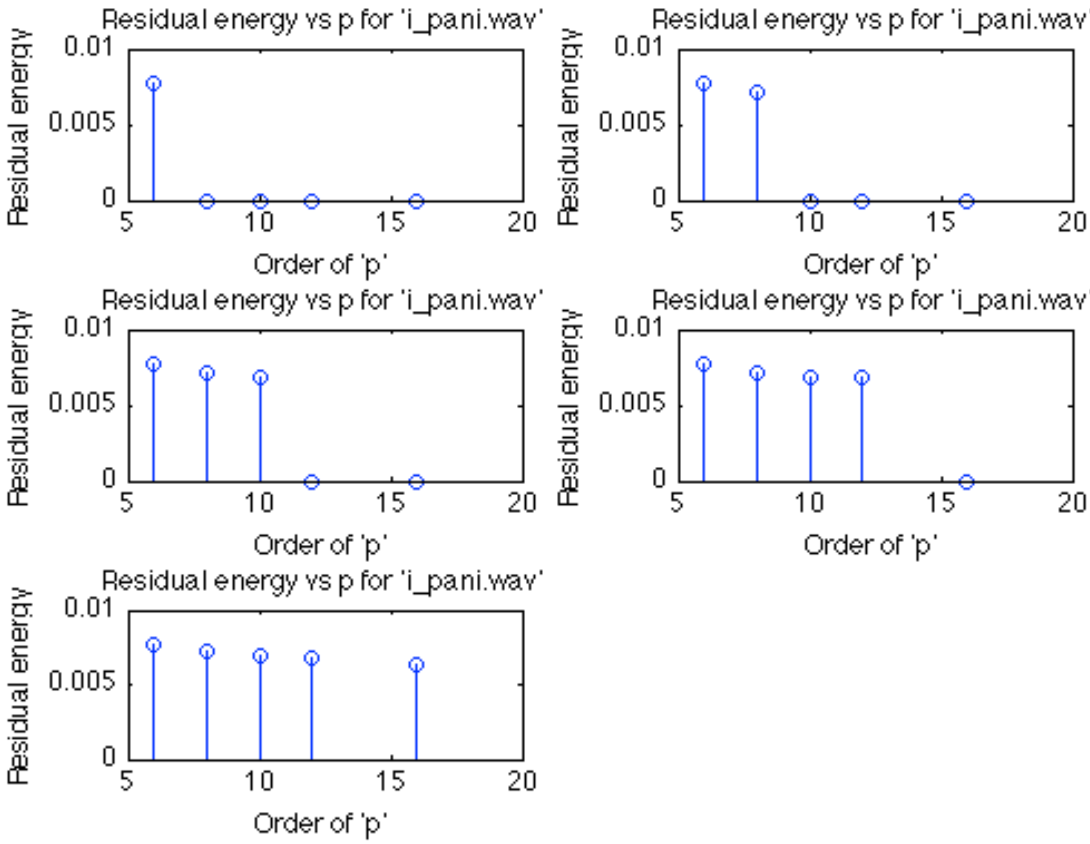
```
function plotResidualEnergyVersusP(inputFile, poleOrders)

residualEnergies = zeros(length(poleOrders), 1);
cols = 2;
rows = ceil(length(poleOrders)/cols);
figure;
for k = 1:length(poleOrders)
    residualEnergies(k) = getResidualEnergy(inputFile,
poleOrders(k));
    subplot(rows, cols, k); stem(poleOrders, residualEnergies);
    title(['Residual energy vs p for ''', inputFile, ''''],
'interpreter', 'none');
    xlabel('Order of ''p''');
    ylabel('Residual energy');
end
end



%% get residual energy

function residualEnergy = getResidualEnergy(inputFile,
poleOrder)

residualEnergy = 0;
residualSignal = getResidualSignal(inputFile, poleOrder);
residualSignal2 = residualSignal .^ 2;
residualEnergy = residualEnergy + sum(residualSignal2(:));
end


%% get residual signal

function residualSignal = getResidualSignal(inputFile,
poleOrder)
```

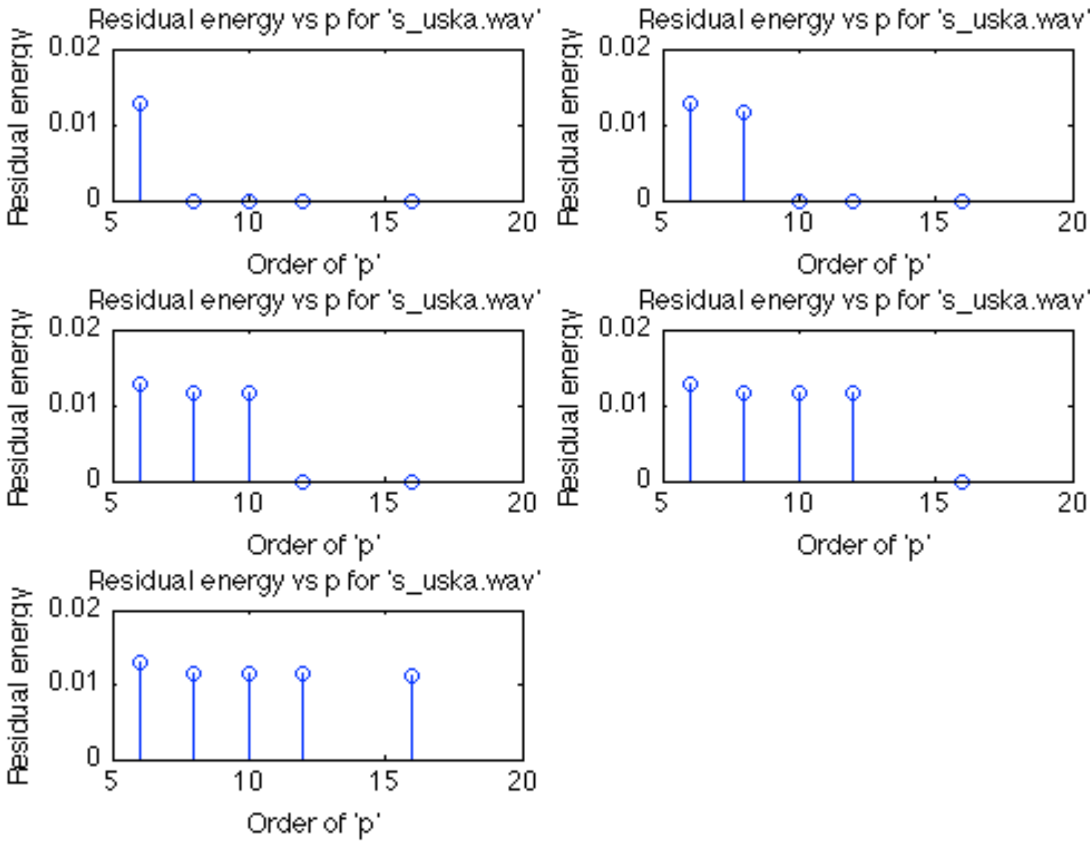
```
LPCoeffs = getLPCoefficients(inputFile, poleOrder);
windowedSignal = getWindowedSignal(inputFile);
siz = size(windowedSignal(:));
M = siz(1);
errorSignal = zeros(M, 1);

for k = 1:M
    errorSignal(k) = windowedSignal(k);
    for j = 1:poleOrder
        if k > j
            errorSignal(k) = errorSignal(k) - LPCoeffs(j) *
windowedSignal(k-j);
        else
            break;
        end
    end
end
residualSignal = errorSignal;


end


%% get hamming windowed central part of a signal

function windowedSignal = getWindowedSignal(inputFile)

% inputFile = 'a_pani.wav';
windowDuration = 0.030; % in ms
[y, fs] = preEmphasize(inputFile);
siz = size(y);
length = siz(1);
centralIndex = round(length/2);
M = round(windowDuration * fs);
startIndex = round(centralIndex - M/2);
windowedSignal = y(startIndex:startIndex + M-1);
end
```

<u>The plots:</u>


Residual energy vs p for 'a_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'a_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'a_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'a_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'a_pani.wav'
Residual energy
Order of 'p'


Residual energy vs p for 'n_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'n_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'n_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'n_pani.wav'
Residual energy
Order of 'p'
Residual energy vs p for 'n_pani.wav'
Residual energy
Order of 'p'


Residual energy vs p for 'i_pani.wav'
Residual energy
Order of 'p'
0.01
0.005
0
5
10
15
20



Answer to question 3

The functions:

```
function [pitch, gain, LPCCoeffs] =
estimateSpeechParameters(inputFile)

poleOrder = 10;
[residualSignal, fs] = getResidualSignal(inputFile, poleOrder);

N = 2 ^ nextpow2(length(residualSignal) * 4);
magnitudeSpectrum = abs(fft(residualSignal, N));
w = 0:length(magnitudeSpectrum)-1;
w = w .* 2 * pi * (1/length(magnitudeSpectrum));

figure; plot(w, magnitudeSpectrum); axis tight;
```

```
title(['Magnitude spectrum of residual signal for ''', 
inputFile, ...
    ''''], 'interpreter', 'none');
xlabel('''w'' in radians');

pitch = getPitch(inputFile, fs);
LPCCoeffs = getLPCCoefficients(inputFile, poleOrder);
gain = getGain(inputFile, poleOrder);

end


%% get residual signal of full signal

function [residualSignal, fs] = getResidualSignal(inputFile, 
poleOrder)

LPCoeffs = getLPCoefficients(inputFile, poleOrder);
[fullSignal, fs] = getFullSignal(inputFile);
siz = size(fullSignal(:));
M = siz(1);
errorSignal = zeros(M, 1);

for k = 1:M
    errorSignal(k) = fullSignal(k);
    for j = 1:poleOrder
        if k > j
            errorSignal(k) = errorSignal(k) - LPCoeffs(j) * 
fullSignal(k-j);
        else
            break;
        end
    end
end
residualSignal = errorSignal;

end

%% get hamming windowed central part of a signal

function [fullSignal, fs] = getFullSignal(inputFile)

[y, fs] = preEmphasize(inputFile);
fullSignal = y;

end
```

```
%% get autocorrelation coefficients of full signal

function autocorrelationCoefficients =
getAutoCorrelationCoefficients(inputFile, poleOrder)

fullSignal = getFullSignal(inputFile);
M = length(fullSignal);
ACCoeff = zeros(poleOrder+1, 1);

for p = 0:poleOrder
    for k = 0:M-1
        valueToBeAdded = 0;
        if k-p >= 0
            valueToBeAdded = fullSignal(k+1) .* fullSignal(k+1-
p);
        end
        ACCoeff(p+1) = ACCoeff(p+1) + valueToBeAdded;
    end
end

autocorrelationCoefficients = ACCoeff;

end

%% get LPC coefficients of full signal

function LPCCoeffs = getLPCCoefficients(inputFile, poleOrder)

autocorrCoeffs = getAutoCorrelationCoefficients(inputFile,
poleOrder);
LPCCoeffs = levinsonDurbin(autocorrCoeffs);

end

%% get gain of the full signal

function gain = getGain(inputFile, poleOrder)

ACCoeffs = getAutoCorrelationCoefficients(inputFile, poleOrder);
LPCCoeffs = getLPCCoefficients(inputFile, poleOrder);

predictionError = ACCoeffs(1);
for k = 1:poleOrder
    predictionError = predictionError - (LPCCoeffs(k) *
ACCoeffs(k+1));
```

```
end
gain = sqrt(predictionError);

end

%% get hamming windowed central part of a signal

function windowedSignal = getWindowedSignal(inputFile)

% inputFile = 'a_pani.wav';
windowDuration = 0.030; % in ms
[y, fs] = preEmphasize(inputFile);
siz = size(y);
length = siz(1);
centralIndex = round(length/2);
M = round(windowDuration * fs);
startIndex = round(centralIndex - M/2);
windowedSignal = y(startIndex:startIndex + M-1);
end

%% get the downsampled signal

function downsampledSignal = getDownsampledSignal(signal,
factor)

downsampledSignal = zeros(size(signal));
l = 1;
k = 1;
while k < (length(signal) + 1)
    downsampledSignal(l) = signal(k);
    l = l + 1; k = k + factor;
end

end

%% highlight the peaks of the signal retaining their algebraic
sign
function highlightedSignal = highlightSignal(signal)

highlightedSignal = signal .^ 3;

end

%% get pitch of the windowed signal using harmonic product
spectrum
```

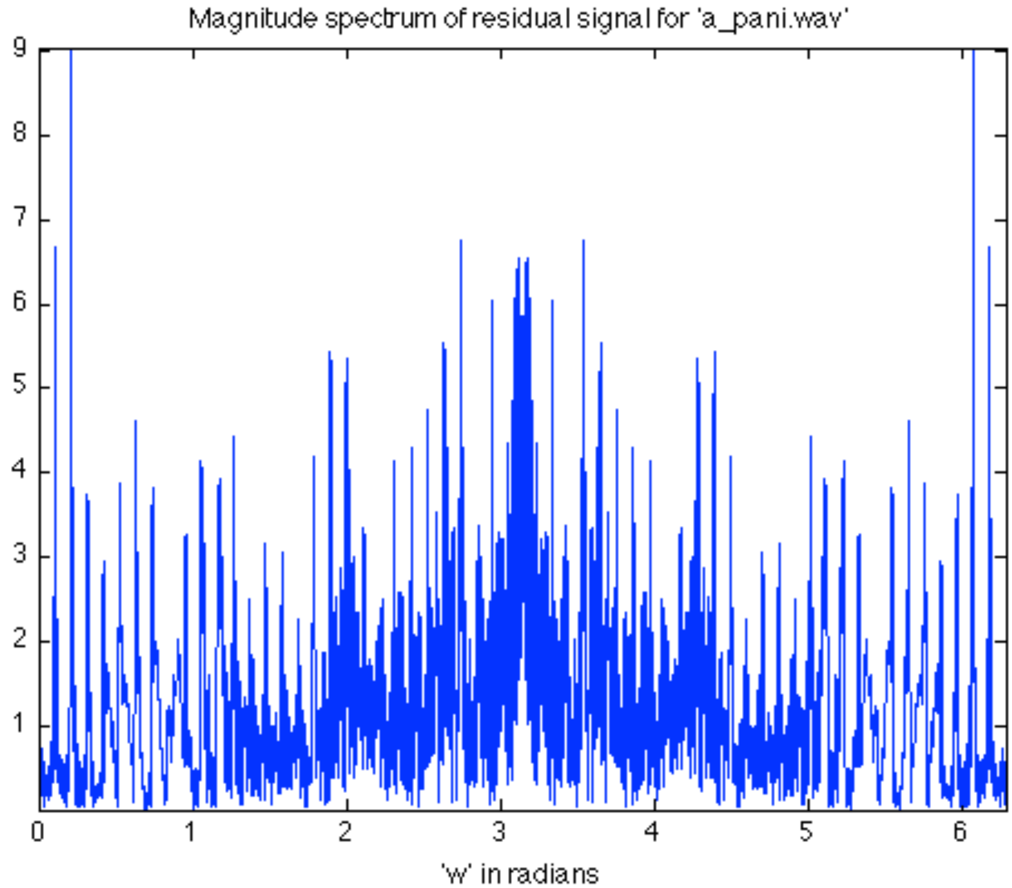
```
function pitch = getPitch(inputFile, fs)
% We calculate the pitch by downsampling the narrowband spectrum
by integer
% values and add them to the original spectrum. At F0, they give
a peak
% because the peaks of other harmonics add up at F0. We take the
second
% peak of the final spectrum because there could be other peaks
in the
% spectrum including one at f = 0 Hz. This way of getting the
pitch is not
% perfect because the pitch values change with the N-point of
FFT
% considered.


windowedSignal = getWindowedSignal(inputFile);
windowedSignal = highlightSignal(windowedSignal);

N = 2 ^ (nextpow2(length(windowedSignal) * 4));
magnitudeSpectrum = abs(fft(windowedSignal, N));
frequencyFactor = fs ./ length(magnitudeSpectrum);
magnitudeSpectrum =
magnitudeSpectrum(1:round(length(magnitudeSpectrum)/2));
spectralPeaks = magnitudeSpectrum;

for k = 2:length(magnitudeSpectrum)
    spectralPeaks = spectralPeaks +
getDownsampledSignal(magnitudeSpectrum, k);
end

% spectralPeaks(1) = 0;
peaks = findpeaks(spectralPeaks);
maxIndex = find(spectralPeaks == peaks(2));
maxIndex = maxIndex(1);

% figure, stem(spectralPeaks); axis tight;

pitch = (maxIndex-1) * frequencyFactor;

end
```

The script:

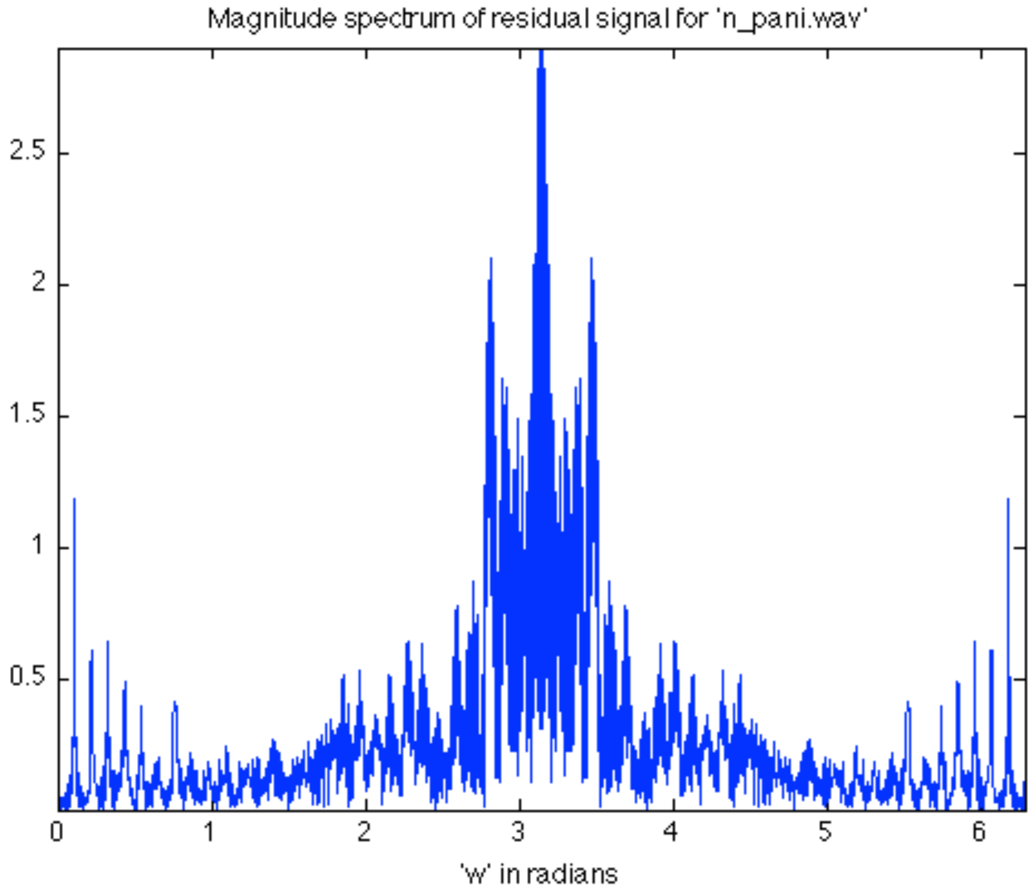
```
close all; clear all;
```

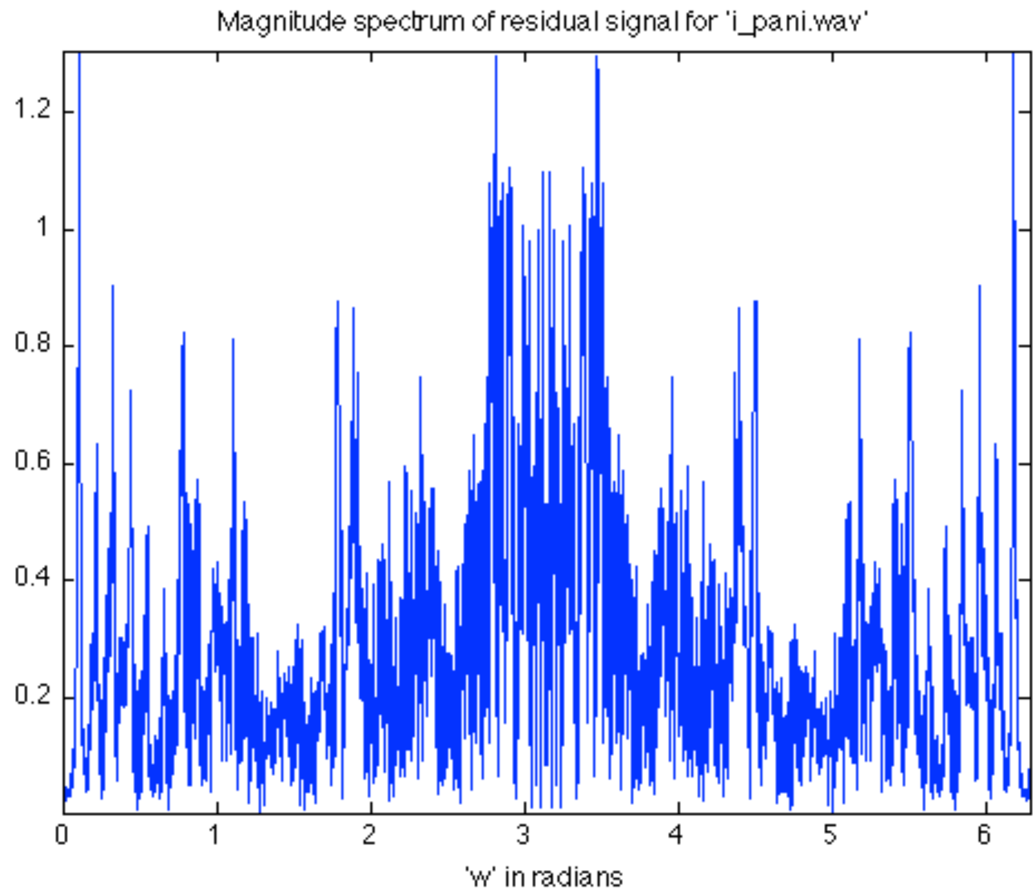
```
addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/q2ab/
addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/

[pitch1, gain1, LPCCoeffs1] =
estimateSpeechParameters('a_pani.wav')
[pitch2, gain2, LPCCoeffs2] =
estimateSpeechParameters('n_pani.wav');
[pitch3, gain3, LPCCoeffs3] =
estimateSpeechParameters('i_pani.wav');
[pitch4, gain4, LPCCoeffs4] =
estimateSpeechParameters('s_uska.wav');


rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/
rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/q2ab/
```

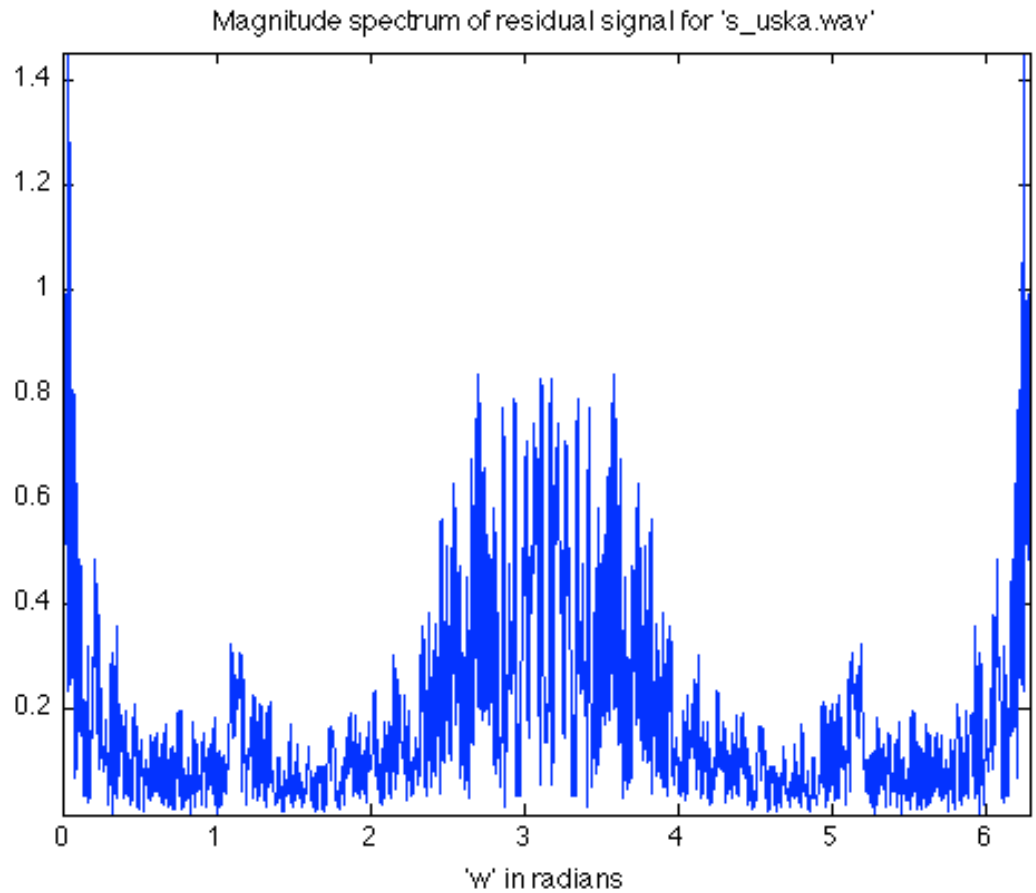
The plots:



Magnitude spectrum of residual signal for 'a_pani.wav'
9
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
'w' in radians


Magnitude spectrum of residual signal for 'n_pani.wav'
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
'w' in radians


Magnitude spectrum of residual signal for 'i_pani.wav'
1.2
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
'w' in radians



<u>The estimated parameters for ‘a_pani.wav’:</u>

pitch1 = 78.1250
gain1 =1.6950
LPCCoeffs1 =
Columns 1 through 7
0.4349 0.6849 -0.6386 -0.2396 0.0940 0.1864 0.2259
Columns 8 through 10
-0.3754 -0.1771 0.4051

<u>The estimated parameters for ‘n_pani.wav’:</u>

pitch2 = 93.7500
gain2 = 0.2697
LPCCoeffs2 =
Columns 1 through 7
0.2994 1.1015 0.2446 -0.3503 -0.7408 0.2767 0.2361
Columns 8 through 10
-0.3347 0.0418 0.0806

The estimated parameters for 'i_pani.wav':

pitch3 = 132.8125
gain3 = 0.3535
LPCCoeffs3 =
 Columns 1 through 7
 -0.1176 1.5257 0.9040 -0.9068 -1.1765 0.1195 0.5747
 Columns 8 through 10
 0.1062 -0.1349 -0.0794

The estimated parameters for 's_uska.wav':

pitch4 = 78.1250
gain4 = 0.1872
LPCCoeffs4 =
 Columns 1 through 7
 -1.6509 0.2766 1.5198 0.5105 0.2722 0.5890 0.0787
 Columns 8 through 10
 -0.3818 -0.3086 -0.0884

Answer to question 4

The functions:

```
function [sampledSignal, samplingFrequency] =
resynthesizeVoicedPhoneUsingLP(inputFile)

duration = 0.300; % in seconds
samplingFrequency = 8000; % in hertz

[pitch, gain, LPCCoeffs] = estimateSpeechParameters(inputFile);
timePeriod = 1/pitch;
signalLength = round(duration/timePeriod);


impulseTrain = getImpulseTrainOfLength(signalLength);
filteredSignal = getFilteredSignal(impulseTrain, gain,
LPCCoeffs);
sampledSignal = getSampledSignal(filteredSignal,
samplingFrequency, timePeriod, duration);
```

```
end


%% get impulse train of specified length

function impulseTrain = getImpulseTrainOfLength(signalLength)

impulseTrain = ones(signalLength, 1);

end


%% get LPC filtered signal of input

function filteredSignal = getFilteredSignal(signal, gain,
LPCCoeffs)

filteredSignal = zeros(size(signal));

for k = 1:length(filteredSignal)
    filteredSignal(k) = gain * signal(k);
    for m = 1:length(LPCCoeffs)
        if (k - m) < 1
            break;
        else
            filteredSignal(k) = filteredSignal(k) +
(LPCCoeffs(m) * filteredSignal(k - m));
        end
    end
end

end

%% get sampled version of the filtered signal

function sampledSignal = getSampledSignal(filteredSignal,
samplingFrequency, timePeriod, duration)

samplingPeriod = 1/samplingFrequency;
sampledSignalLength = duration/samplingPeriod;

sampledSignal = zeros(sampledSignalLength, 1);

for k = 1:length(filteredSignal)
    m = round((k * timePeriod)/samplingPeriod);
    if m > length(sampledSignal)
```

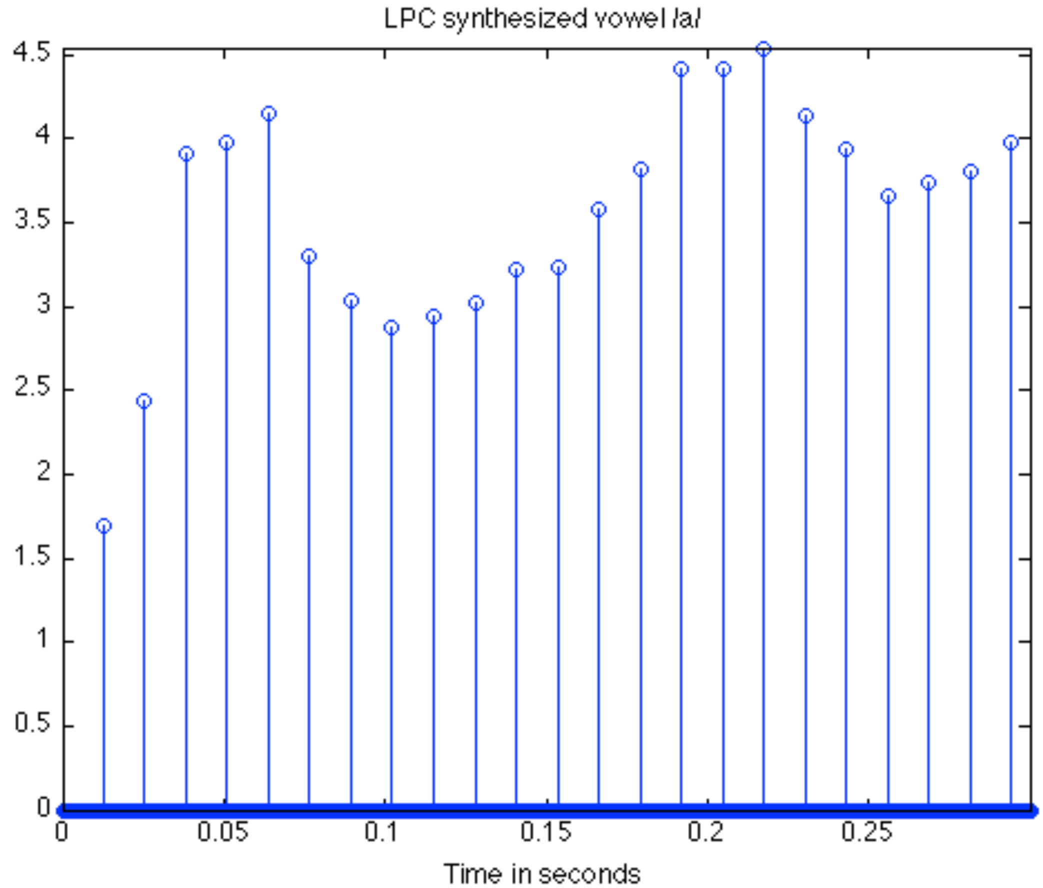
```
        break;
    end
    sampledSignal(m) = filteredSignal(k);
end

end


function [sampledSignal, samplingFrequency] =
resynthesizeUnvoicedPhoneUsingLP(inputFile)

duration = 0.300; % in seconds
samplingFrequency = 8000; % in hertz

[~, gain, LPCCoeffs] = estimateSpeechParameters(inputFile);
timePeriod = 1/samplingFrequency;
signalLength = round(duration/timePeriod);


whiteNoise = wgn(signalLength, 1, 0);

filteredSignal = getFilteredSignal(whiteNoise, gain, LPCCoeffs);
sampledSignal = filteredSignal;
end


%% get LPC filtered signal of input

function filteredSignal = getFilteredSignal(signal, gain,
LPCCoeffs)

filteredSignal = zeros(size(signal));

for k = 1:length(filteredSignal)
    filteredSignal(k) = gain * signal(k);
    for m = 1:length(LPCCoeffs)
        if (k - m) < 1
            break;
        else
            filteredSignal(k) = filteredSignal(k) +
(LPCCoeffs(m) * filteredSignal(k - m));
        end
    end
end

end
```

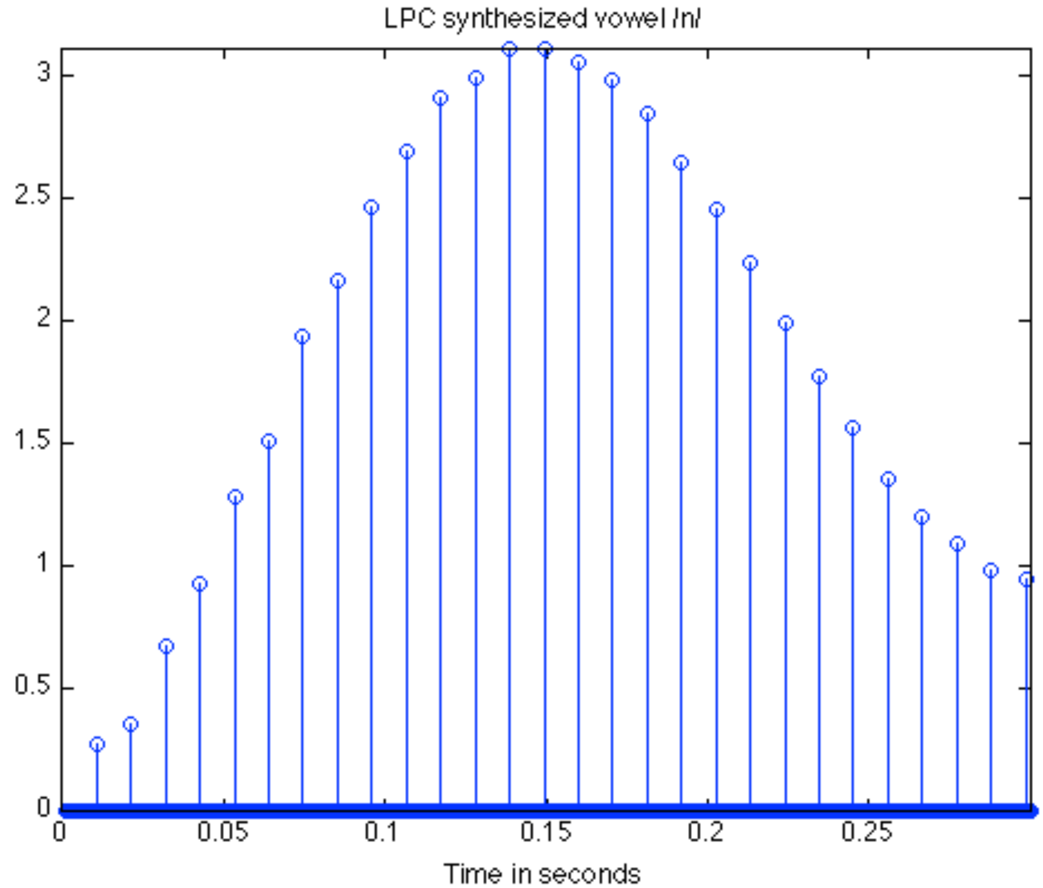
The script:

```
close all; clear all;

addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/
addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/q2ab/
addpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/q3/

[y1, fs1] = resynthesizeVoicedPhoneUsingLP('a_pani.wav');
[y2, fs2] = resynthesizeVoicedPhoneUsingLP('n_pani.wav');
[y3, fs3] = resynthesizeVoicedPhoneUsingLP('i_pani.wav');
[y4, fs4] = resynthesizeUnvoicedPhoneUsingLP('s_uska.wav');


%% viewing the sound

samplingPeriod = 1/fs1; % since all of them are the same using
just one
w1 = (0:length(y1) - 1) * samplingPeriod;
w2 = (0:length(y2) - 1) * samplingPeriod;
w3 = (0:length(y3) - 1) * samplingPeriod;
w4 = (0:length(y4) - 1) * samplingPeriod;

figure(100); stem(w1, y1); axis tight;
title('LPC synthesized vowel /a/');
xlabel('Time in seconds');

figure(200); stem(w2, y2); axis tight;
title('LPC synthesized vowel /n/');
xlabel('Time in seconds');


figure(300); stem(w3, y3); axis tight;
title('LPC synthesized vowel /i/');
xlabel('Time in seconds');

figure(400); stem(w4, y4); axis tight;
title('LPC synthesized phone /s/');
xlabel('Time in seconds');
```

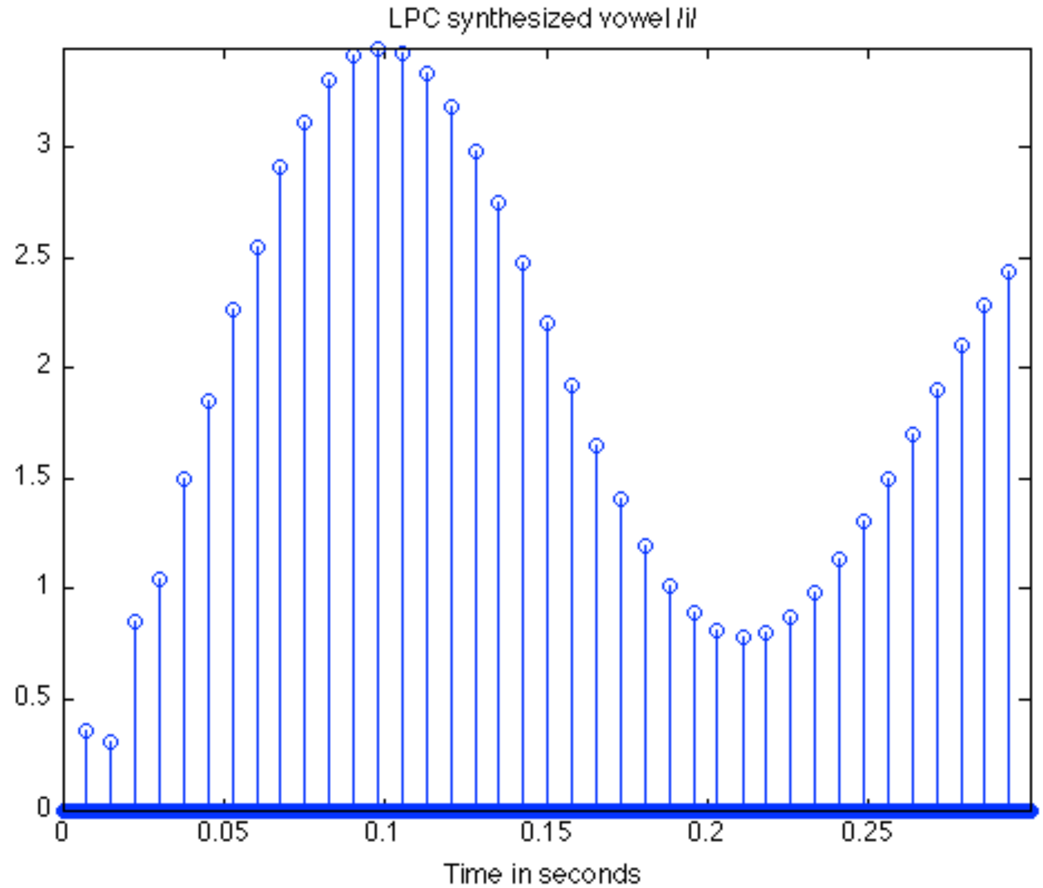
```
%% writing the sound

N = 32; % N-bit sound

wavwrite(y1, fs1, N, 'a_pani_synth.wav');
wavwrite(y2, fs2, N, 'n_pani_synth.wav');
wavwrite(y3, fs3, N, 'i_pani_synth.wav');
wavwrite(y4, fs4, N, 's_uska_synth.wav');


%% playing the sound

bits = 16;

% sound(y1, fs1, bits);
% sound(y2, fs2, bits);
% sound(y3, fs3, bits);
% sound(y4, fs4, bits);


rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/q3/
rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/q2ab/
rmpath /Users/swrangsarbasumatary/Desktop/
speechProcessingProject/
```

<u>The plots:</u>

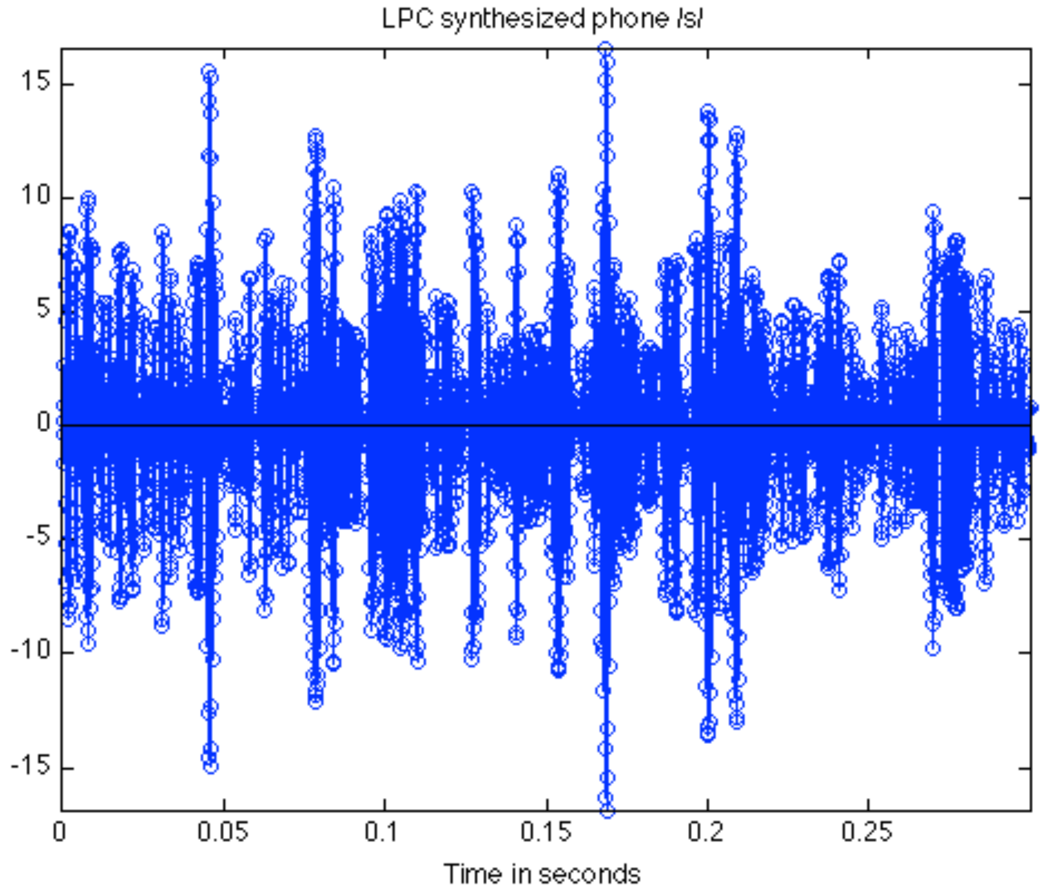

LPC synthesized vowel /a/
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
Time in seconds



LPC synthesized vowel /n/
3
2.5
2
1.5
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
Time in seconds



LPC synthesized vowel /i/
3
2.5
2
1.5
1
0.5
0
0
0.05
0.1
0.15
0.2
0.25
Time in seconds


LPC synthesized phone /s/
15
10
5
0
-5
-10
-15
0
0.05
0.1
0.15
0.2
0.25
Time in seconds