

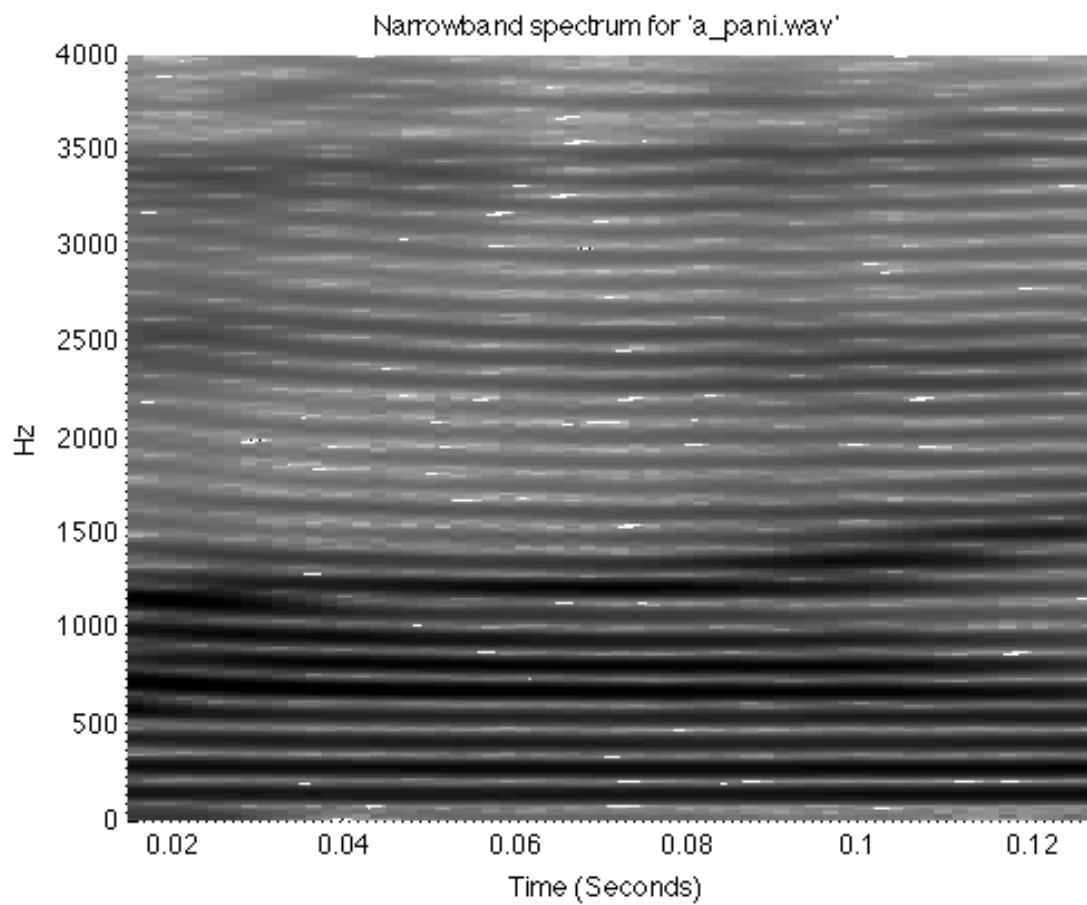
# EE 679 Computing Assignment 3

Name: Swrangsar Basumatary Roll: 09d07040

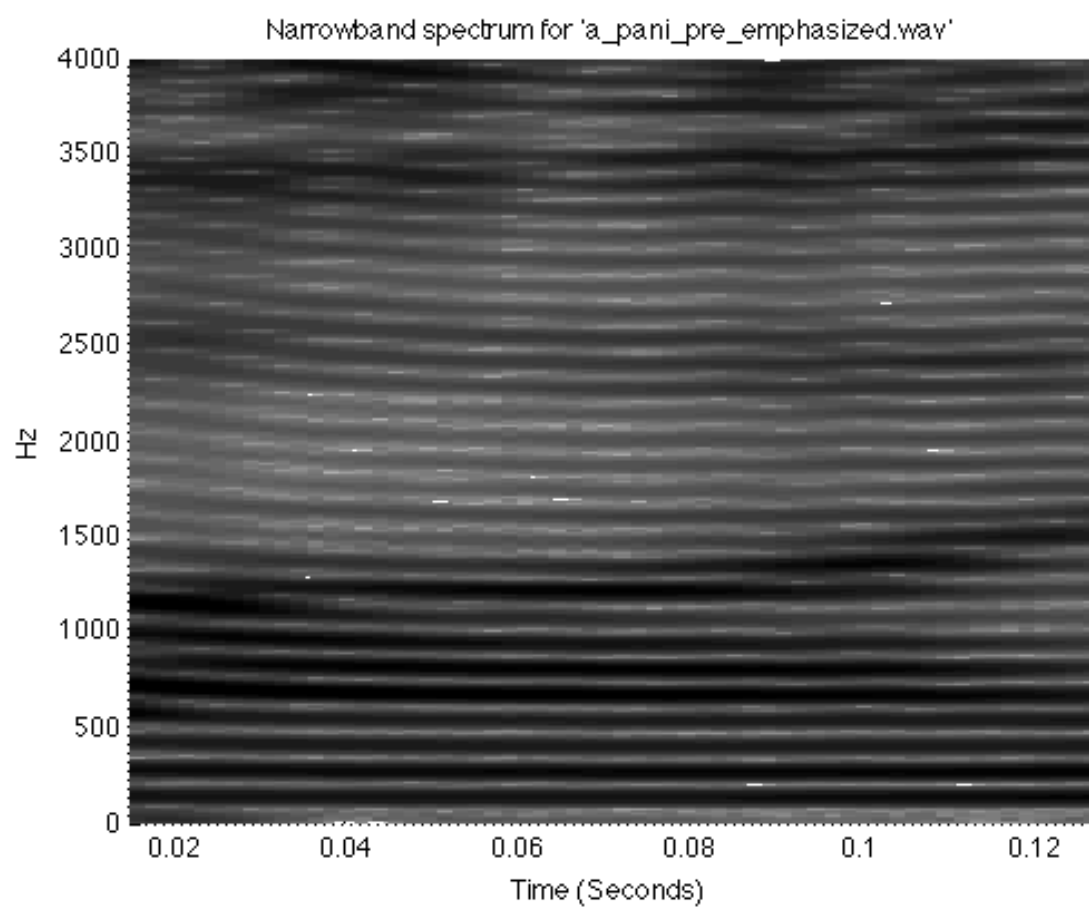
## Question 1

Finding the narrowband spectrum of four syllables:

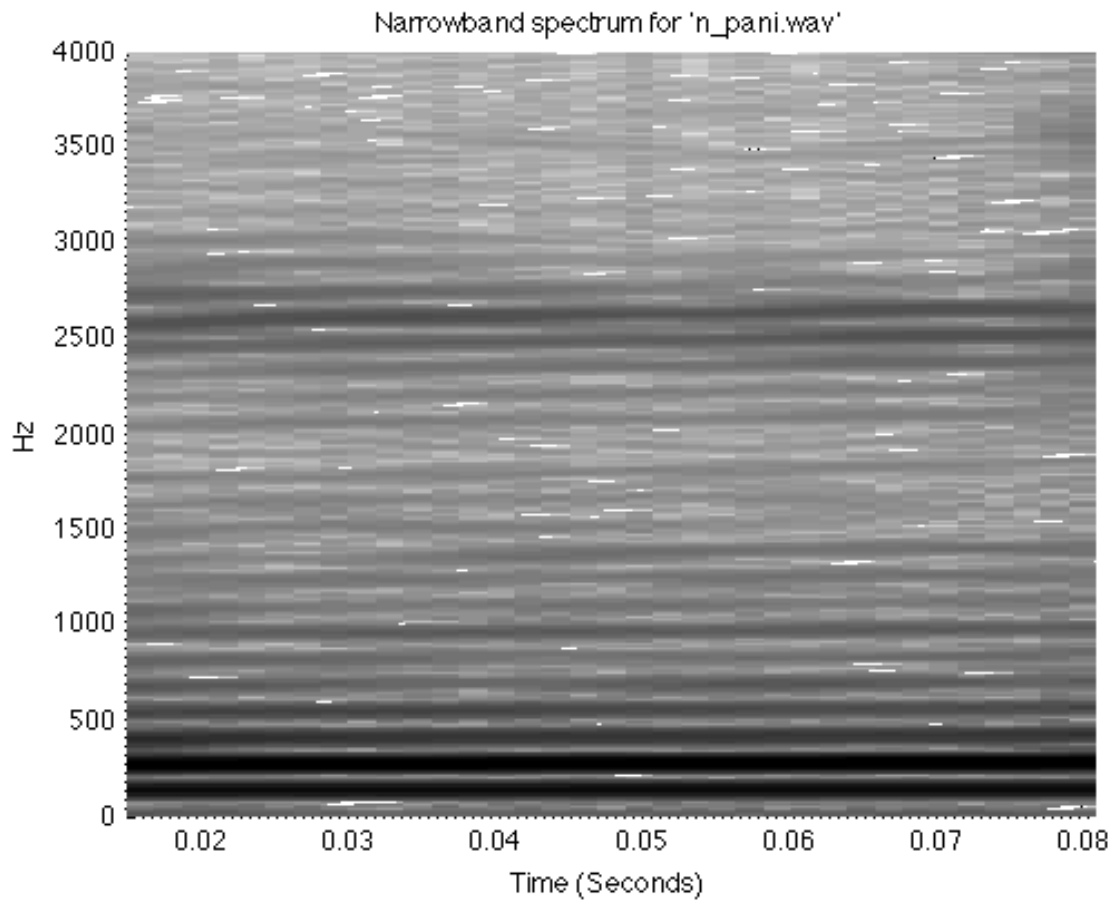
a) /a/ in 'pani'



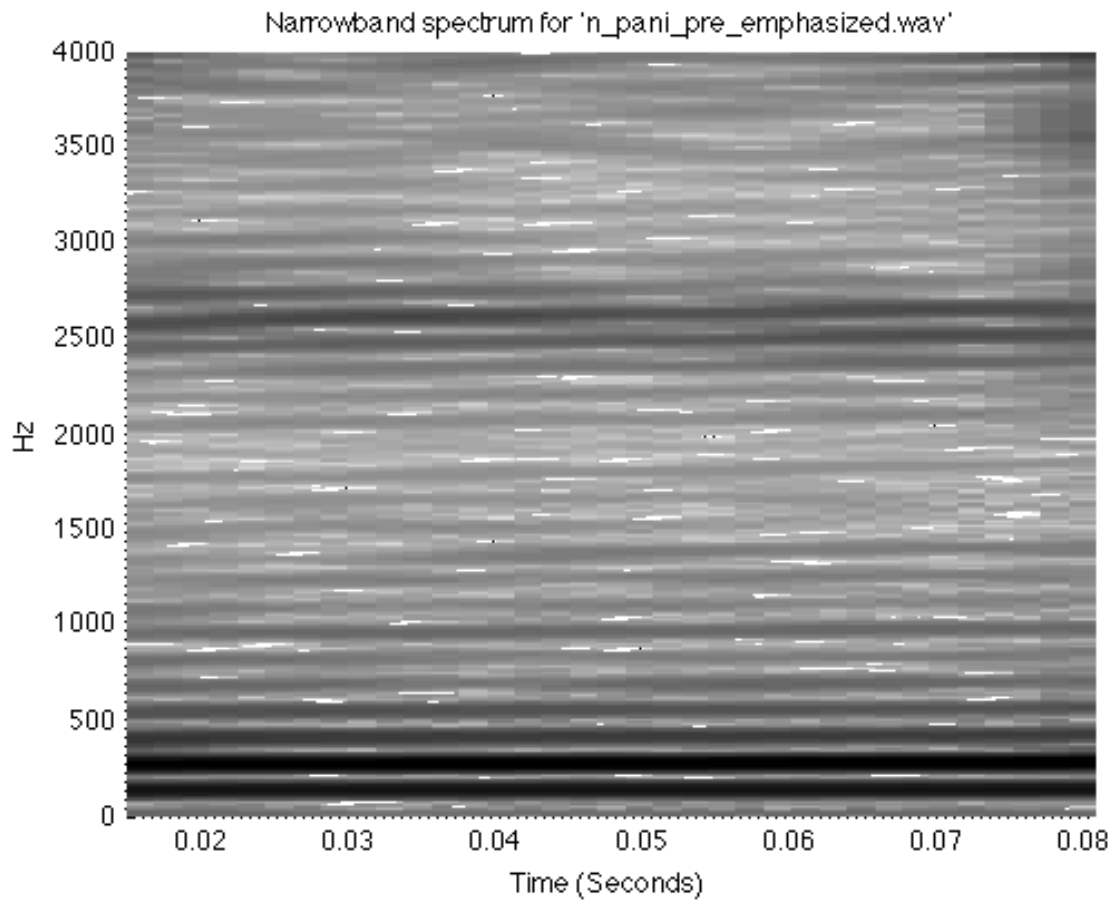
b) /a/ in 'pani' pre-emphasized



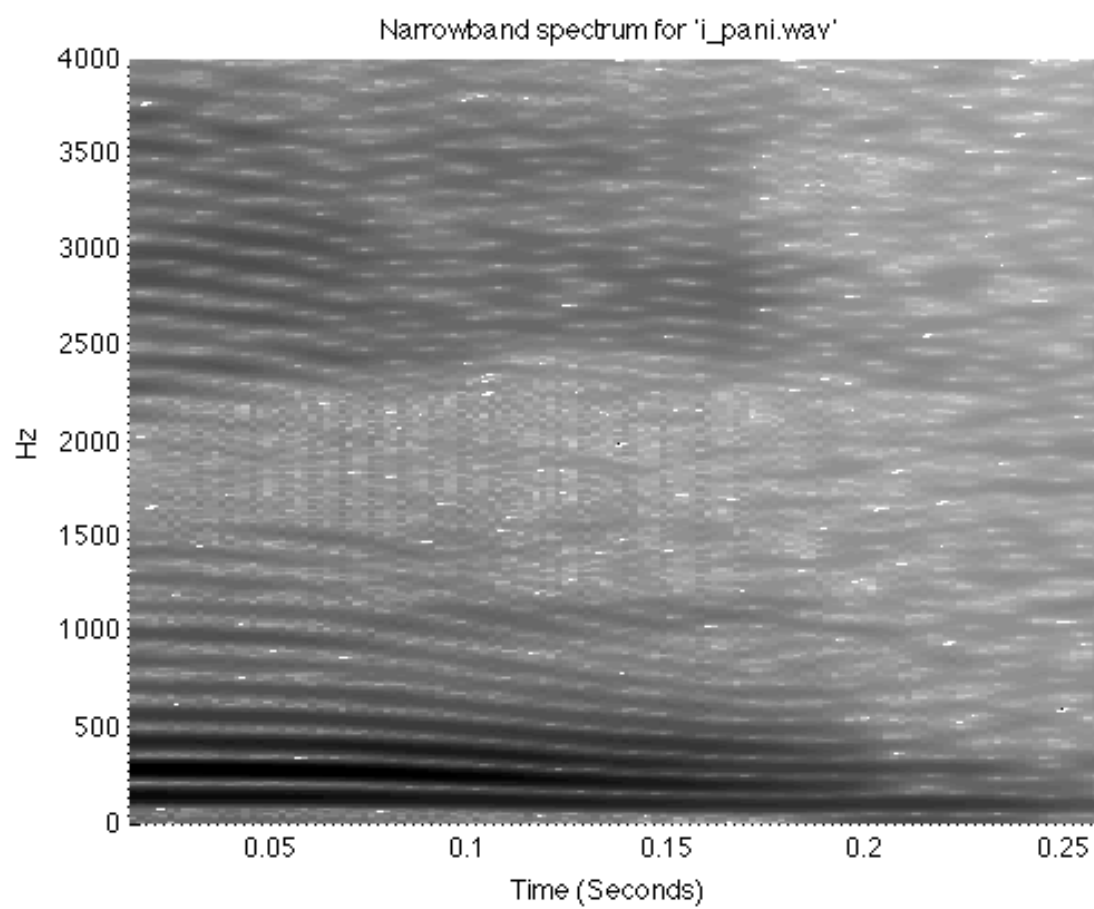
c) /n/ in 'pani'



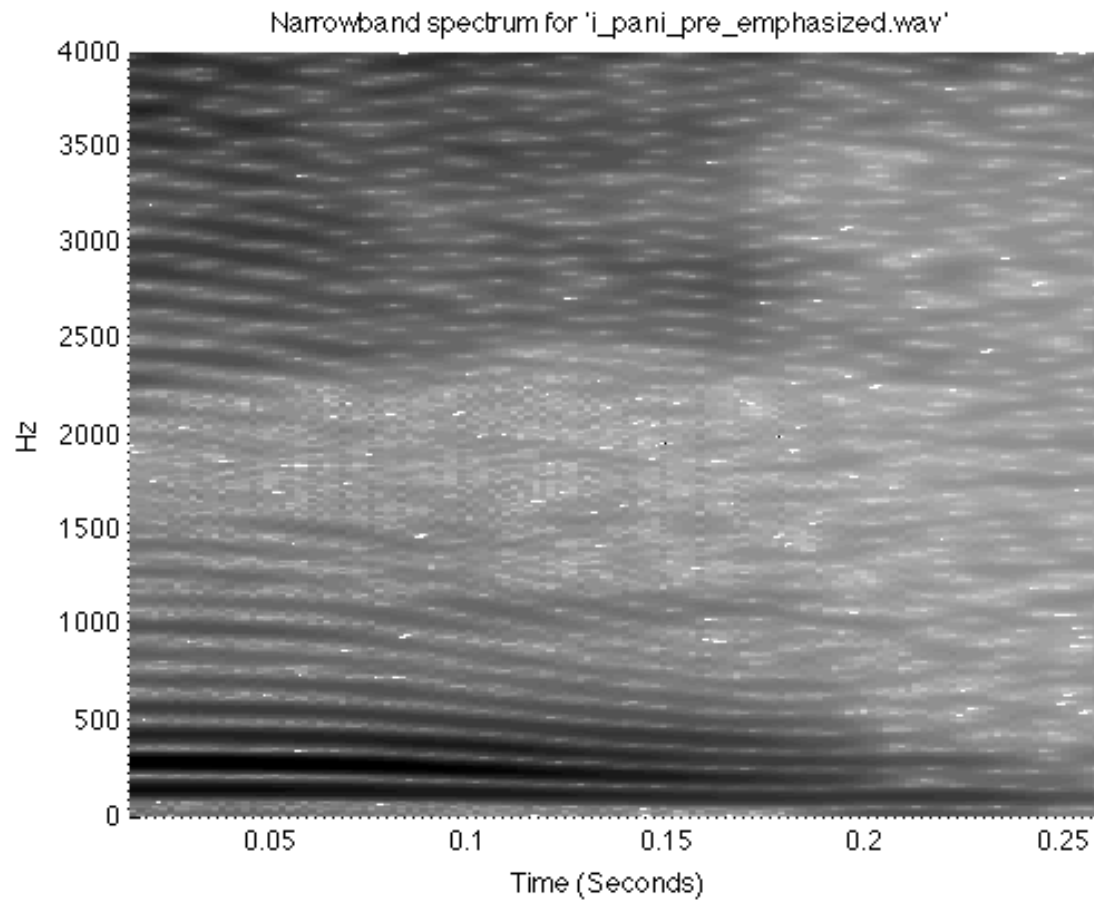
d) /n/ in 'pani' pre-emphasized



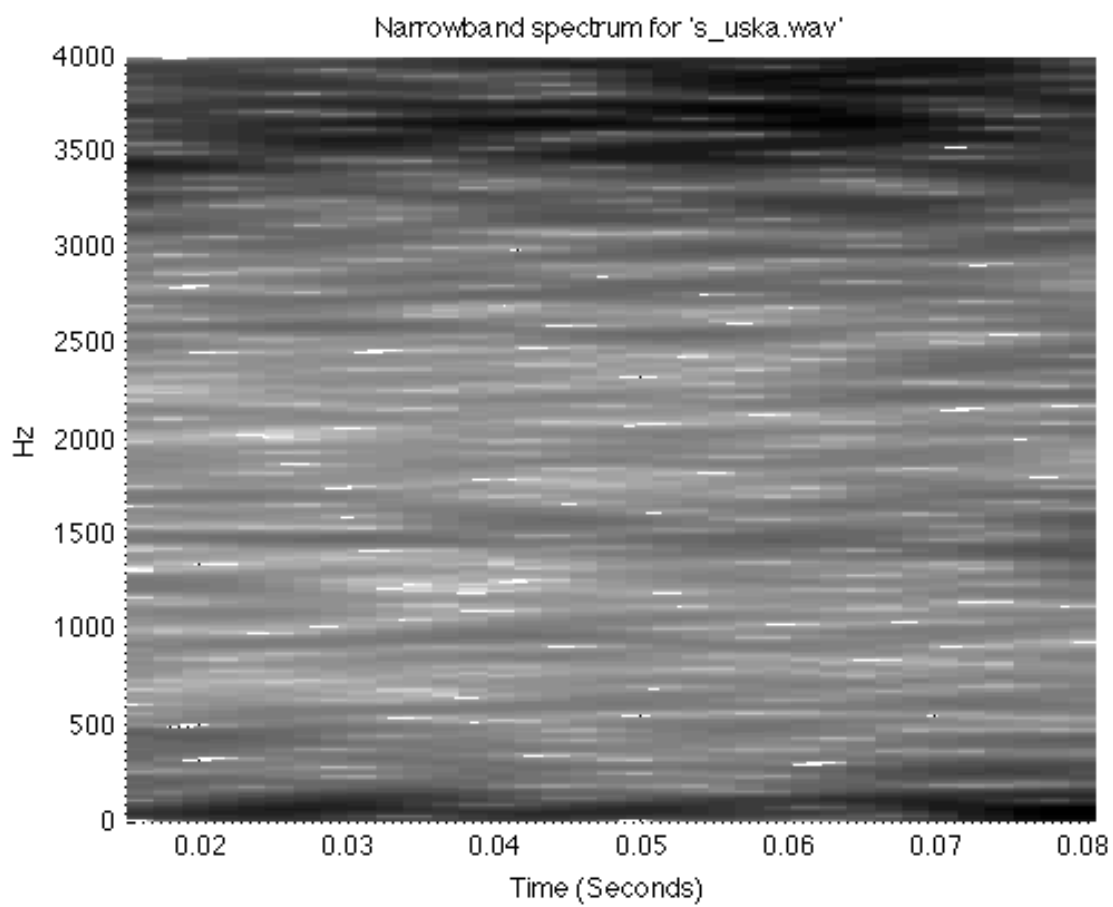
e) // in 'pani'



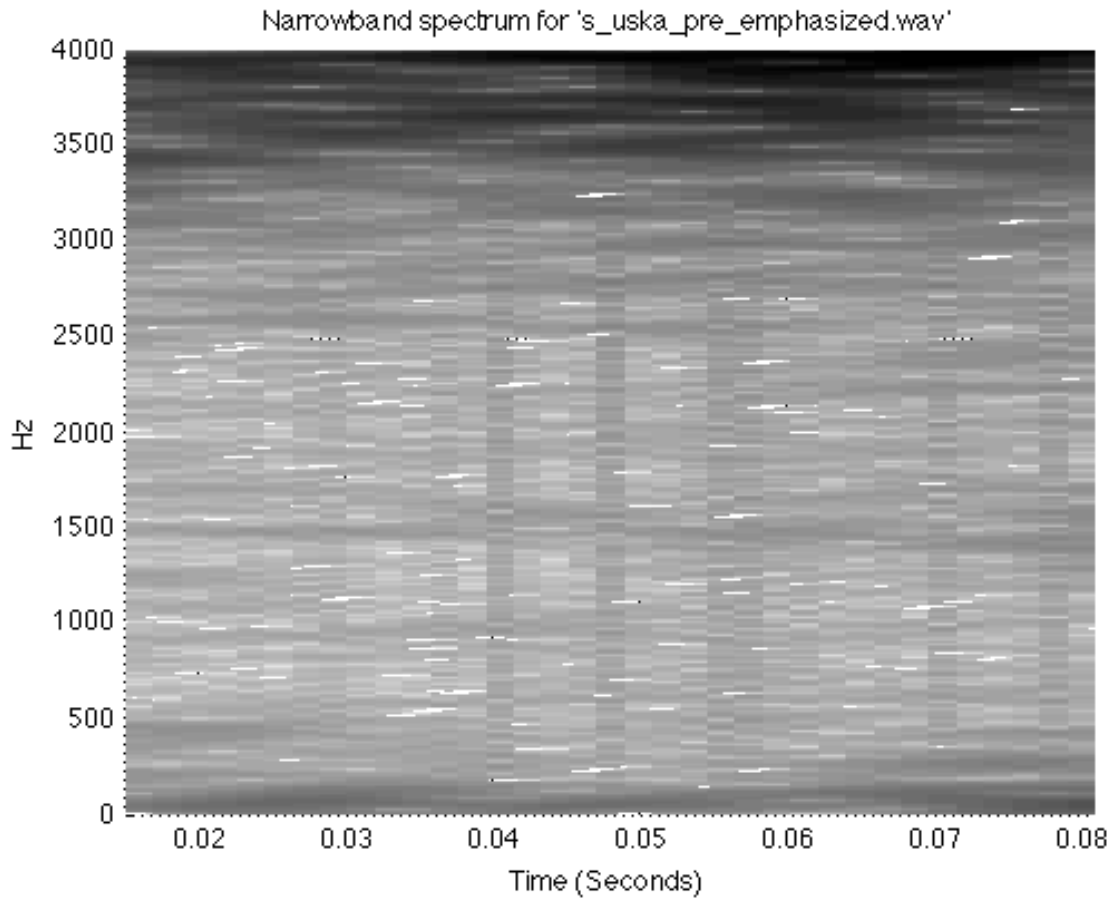
f) // in 'pani' pre-emphasized



g) /s/ in 'uska'



h) /s/ in 'uska' pre-emphasized



### Script for finding the narrowband spectrum

```
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);

```

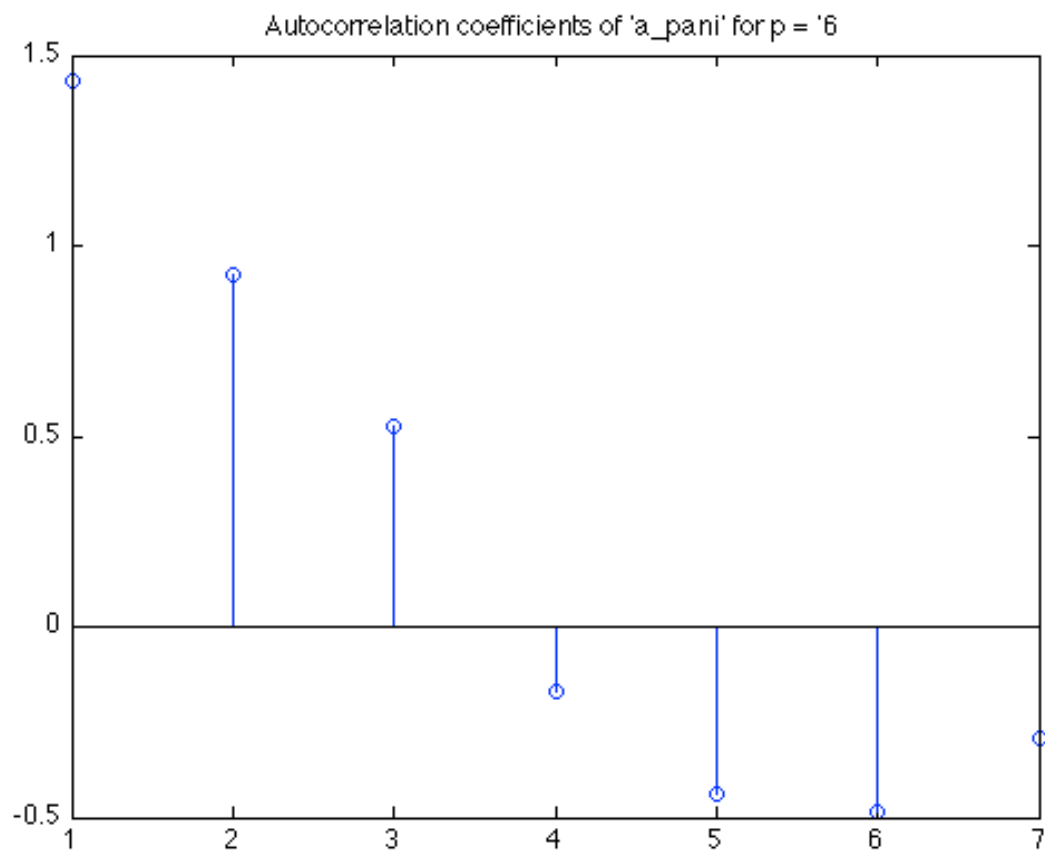
```

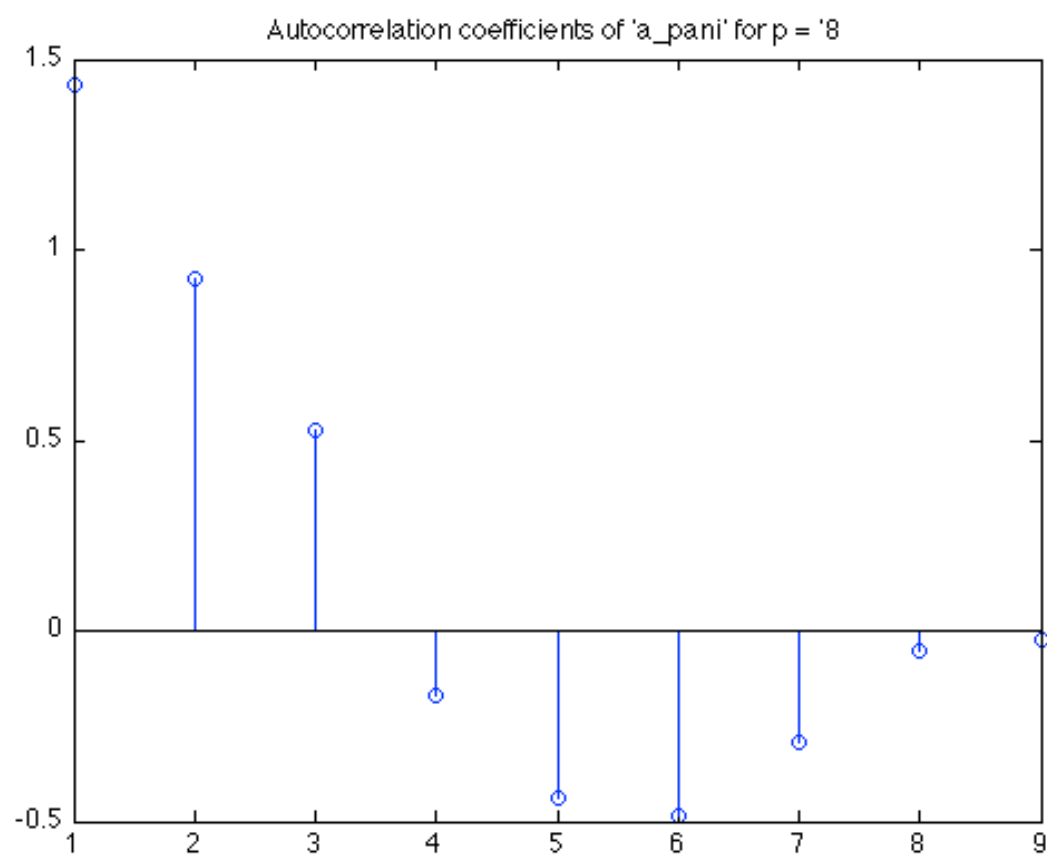
surf(T,F,10*log10(P),'edgecolor','none');
title(['Narrowband spectrum for ', fileInput, ''],
'interpreter', 'none');
axis tight;
colormap(imap);
view(0,90);
xlabel('Time (Seconds)'); ylabel('Hz');

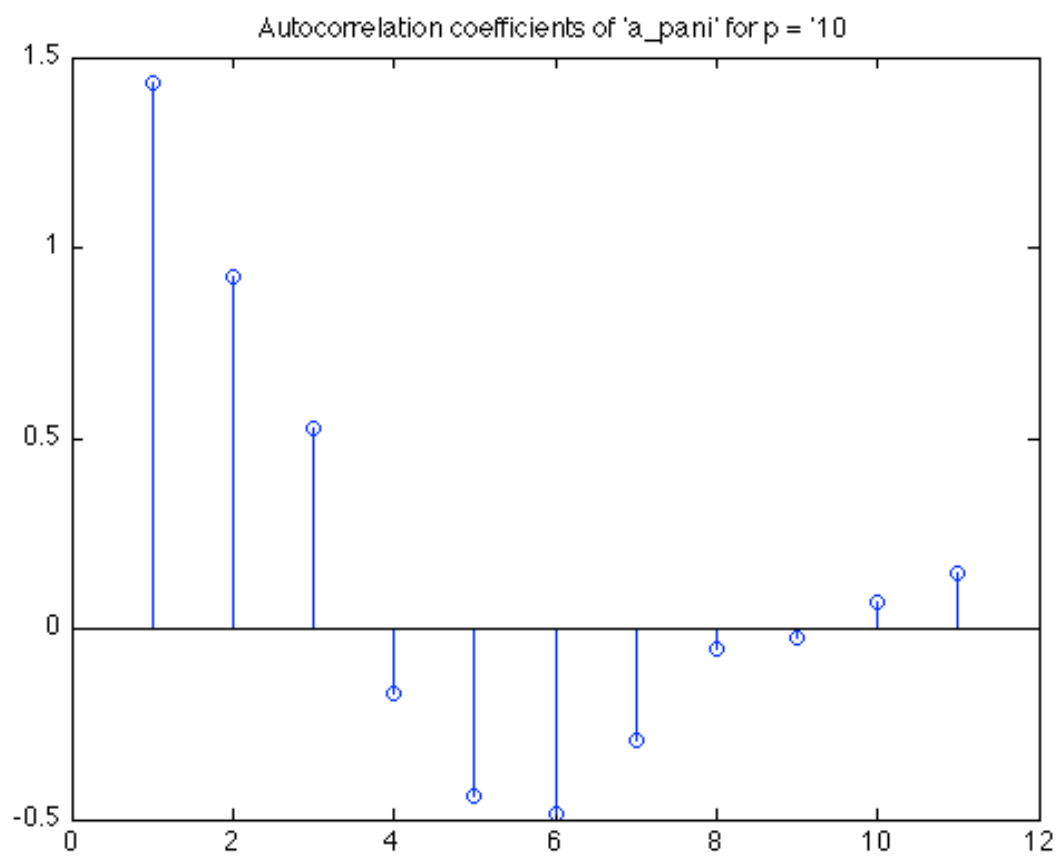
end

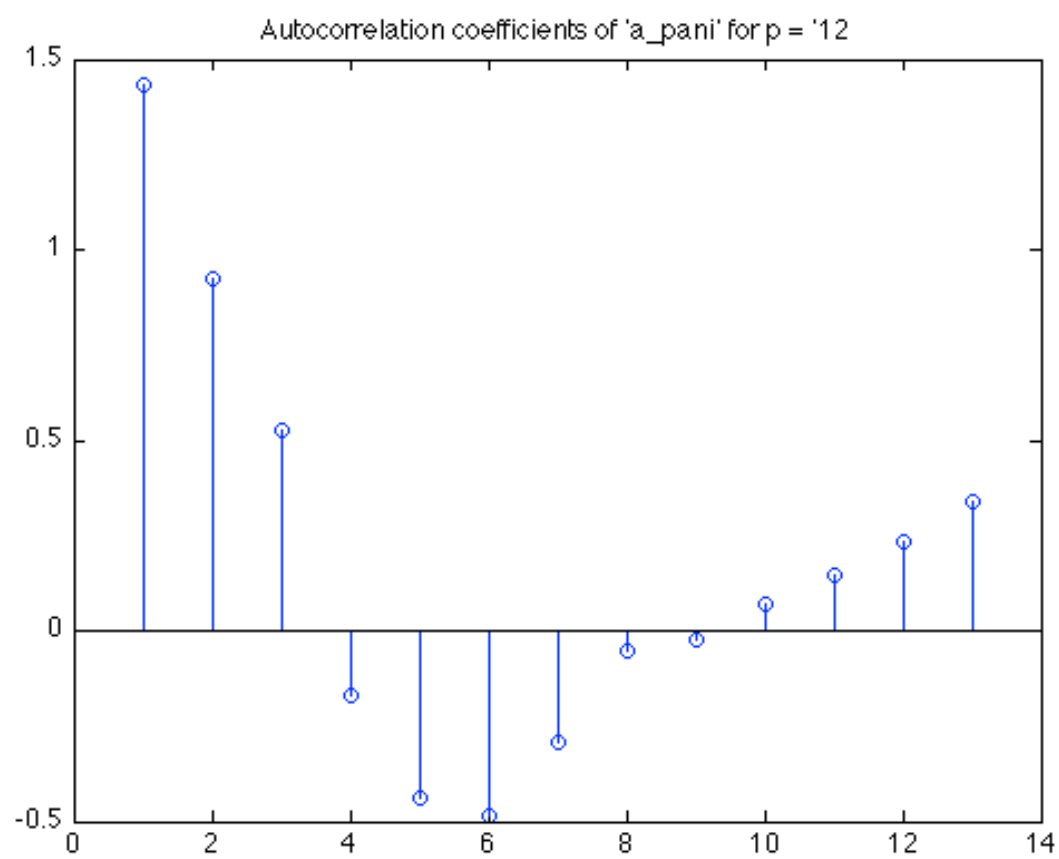
```

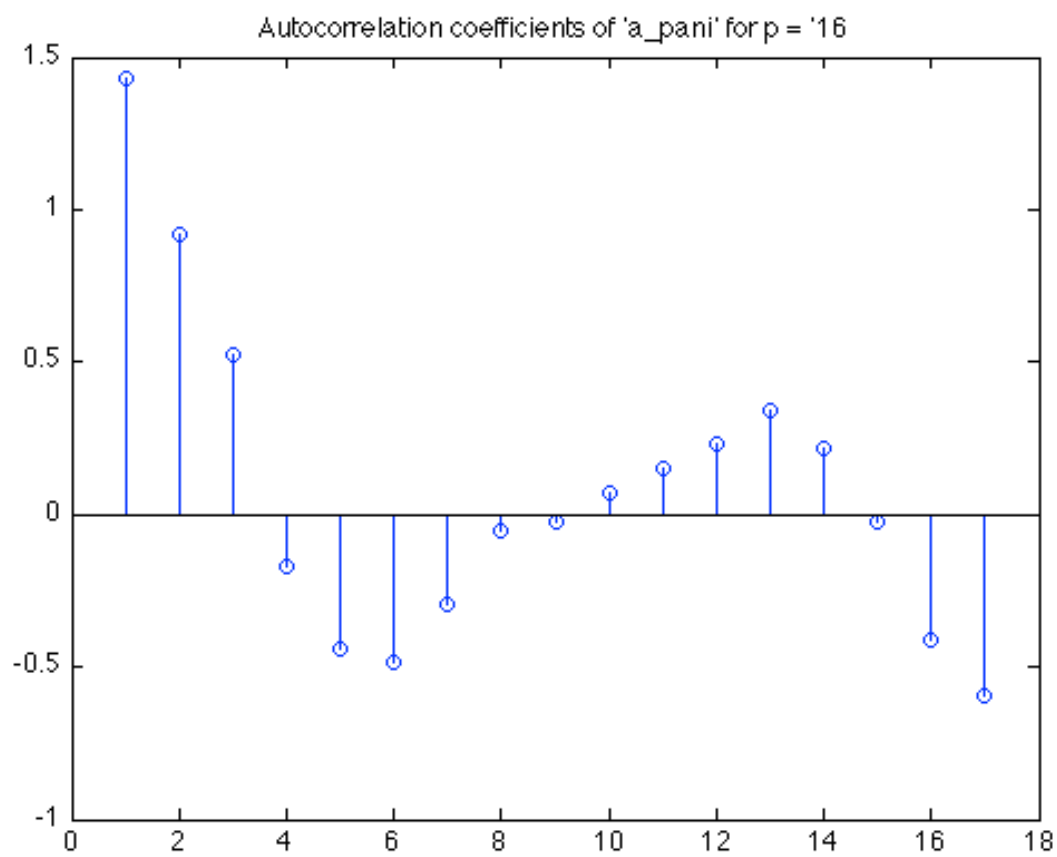
### Answer to Question 2(a)

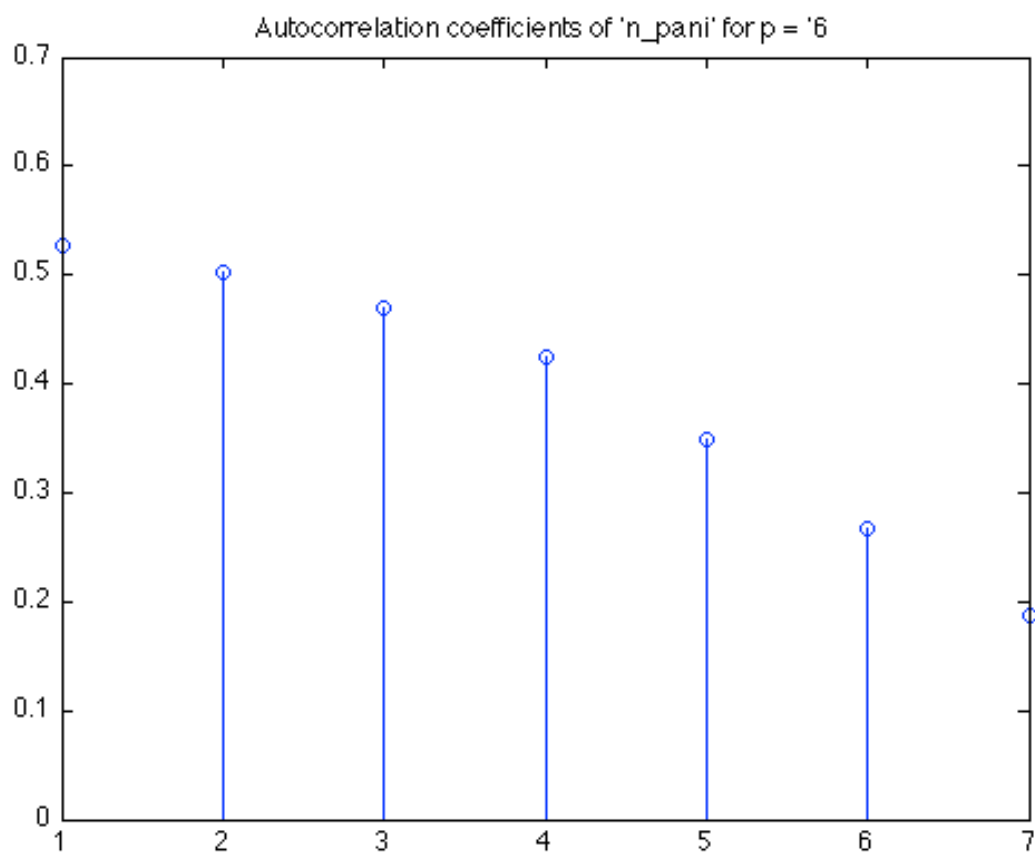


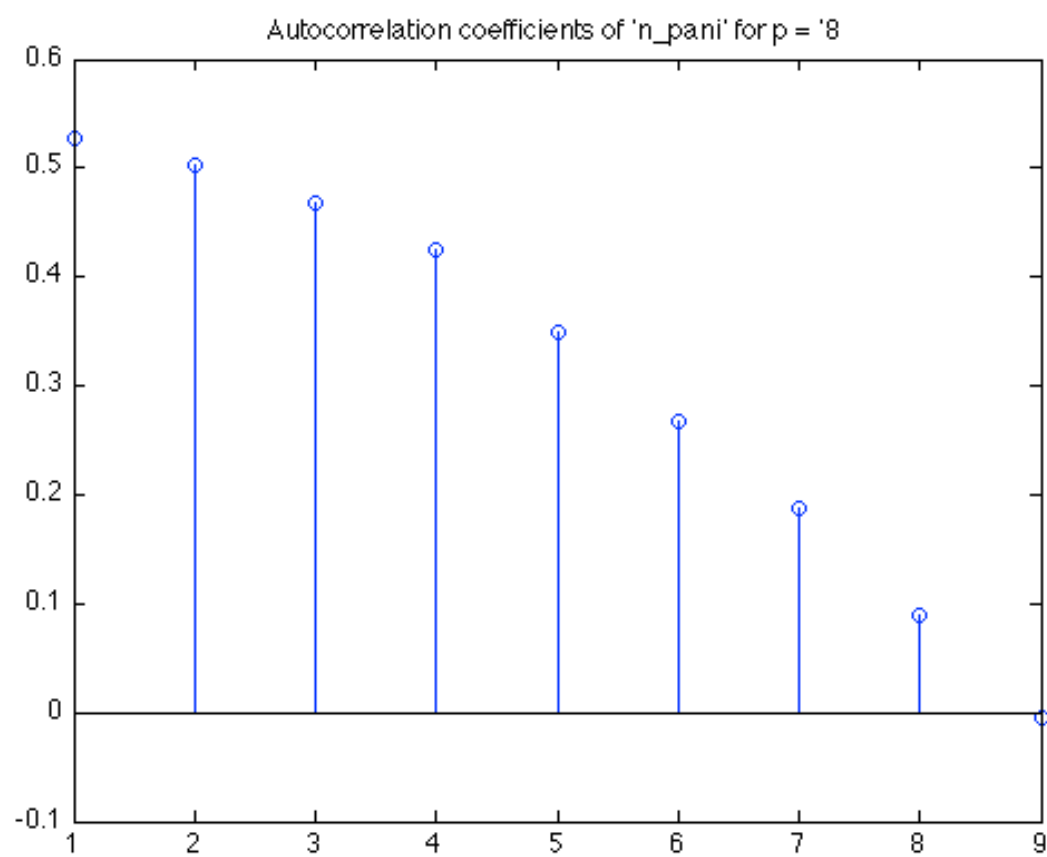




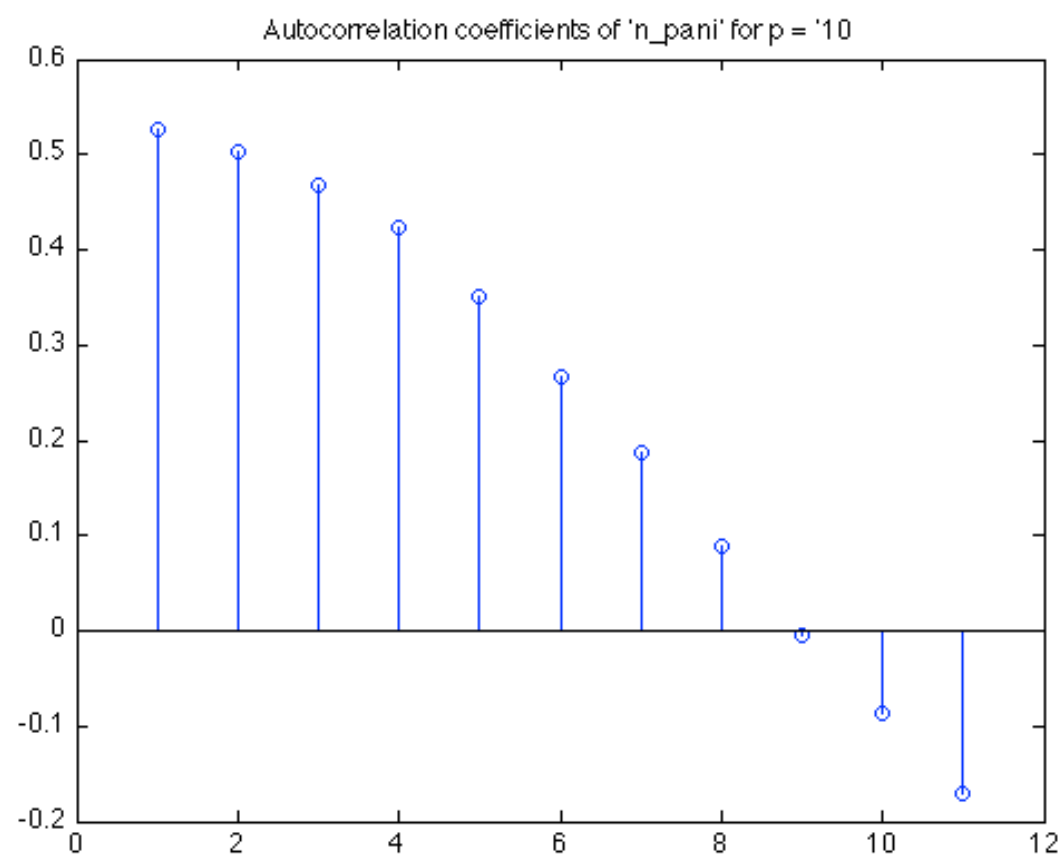


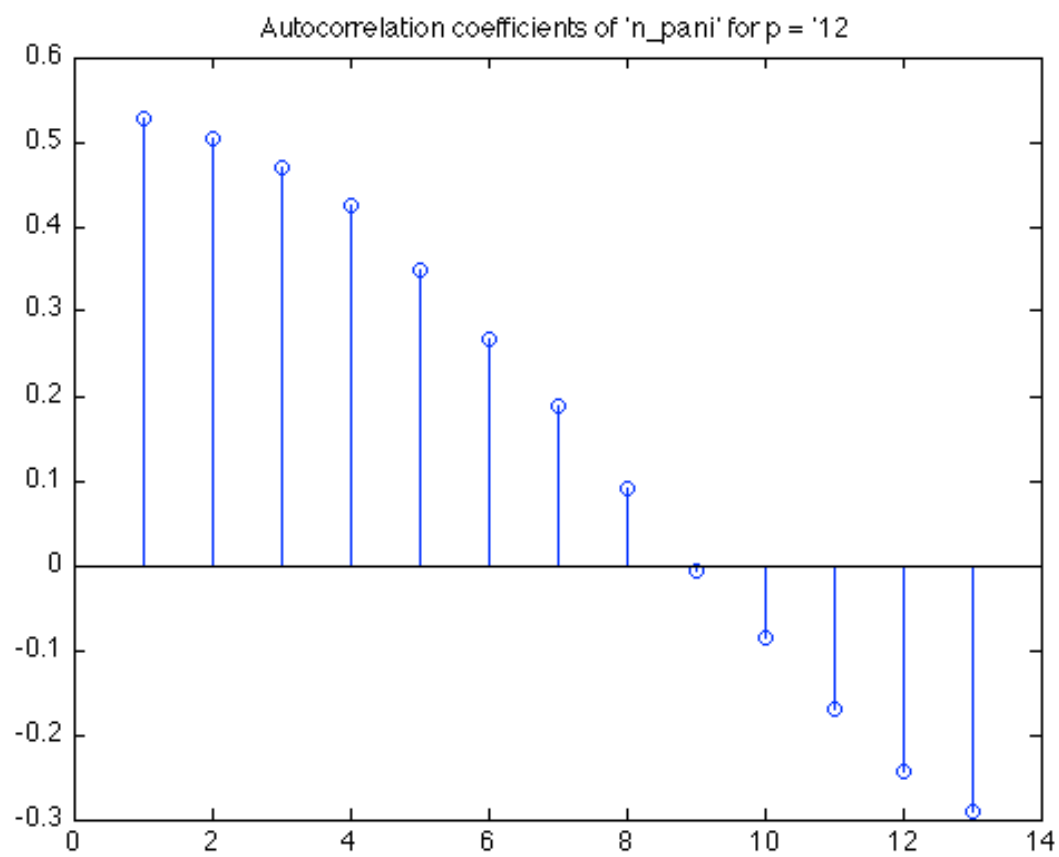


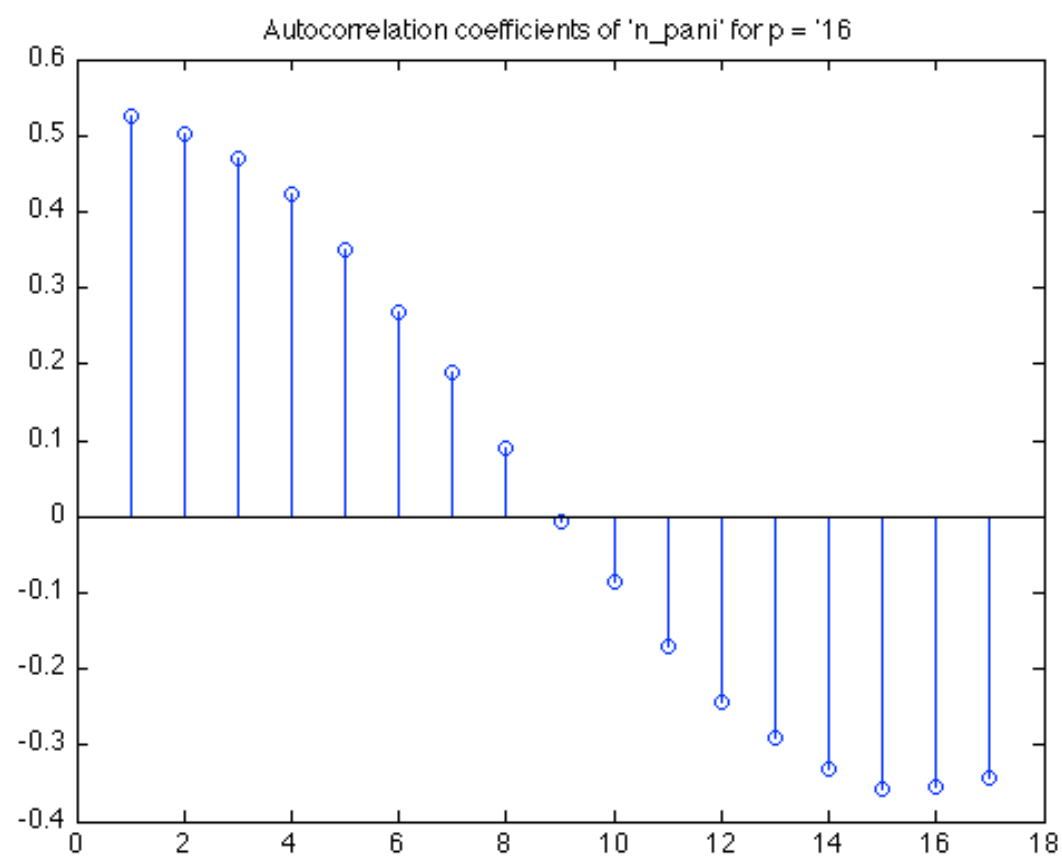


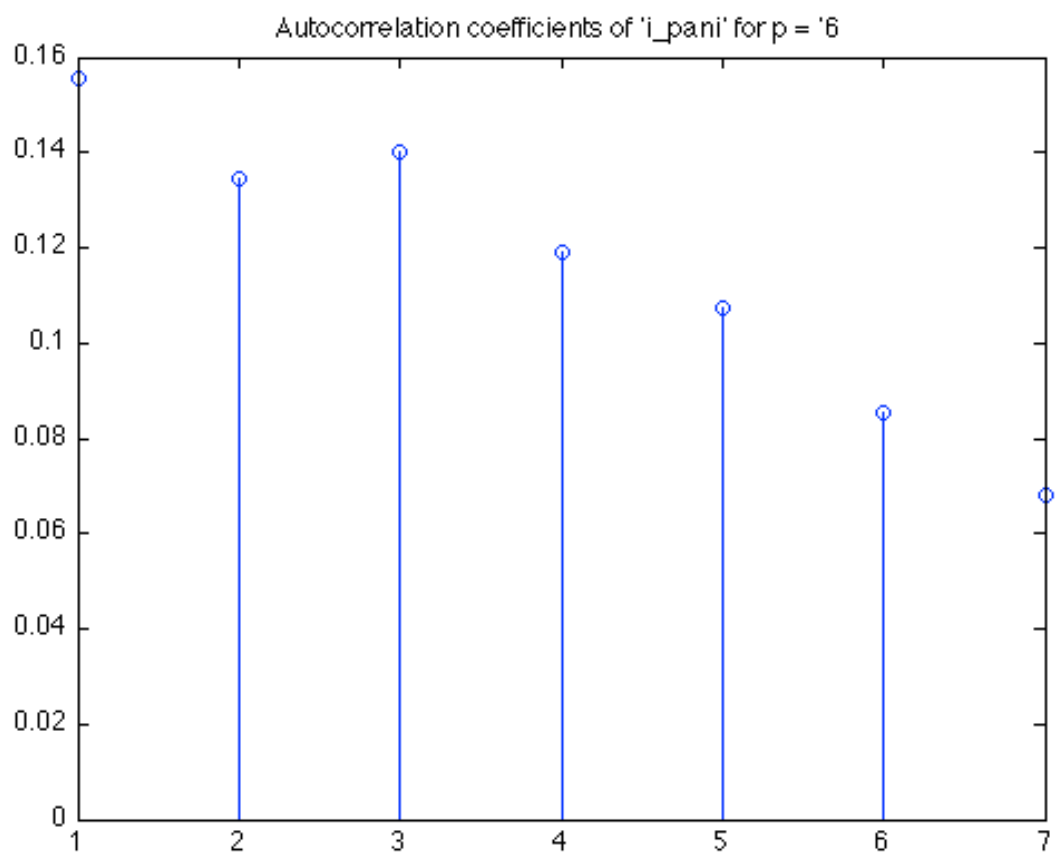


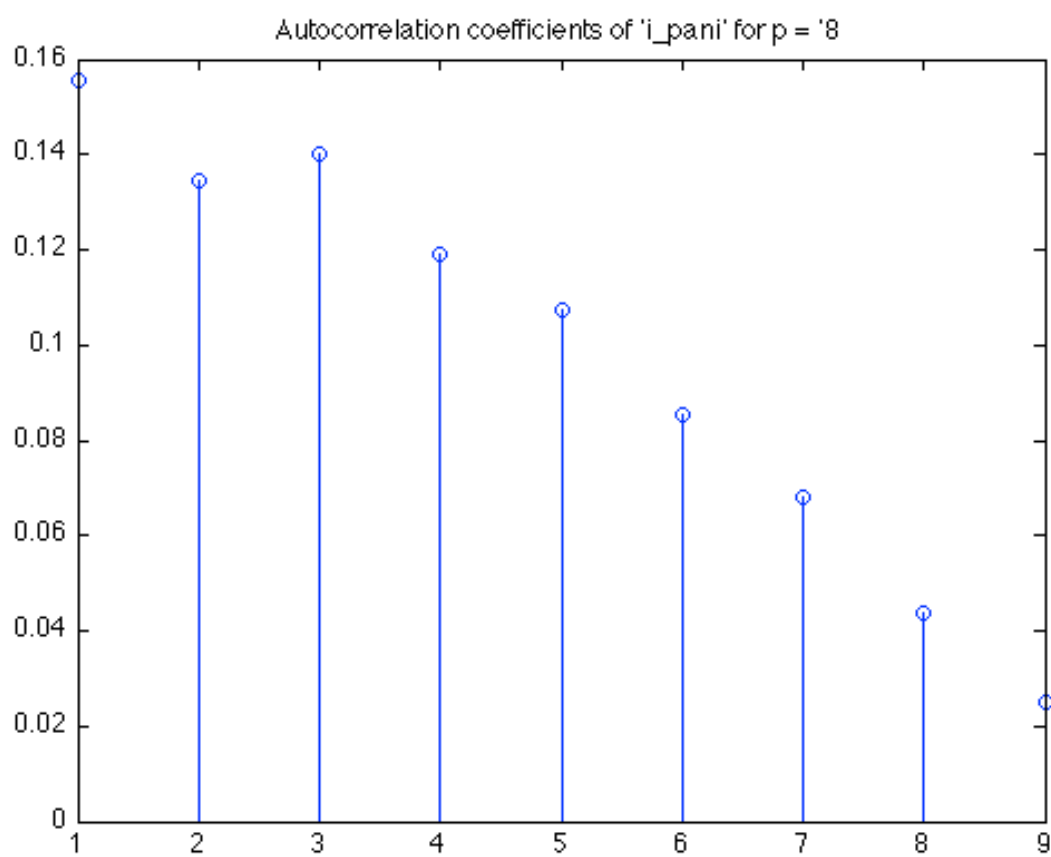


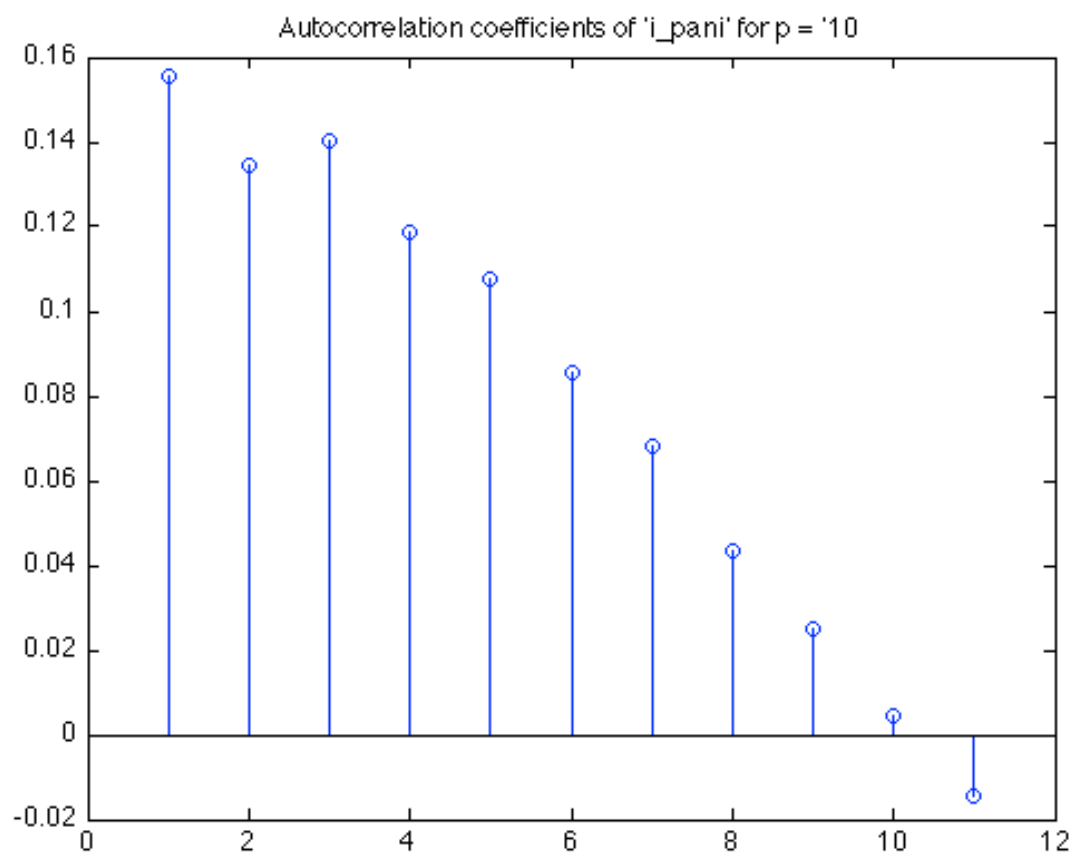


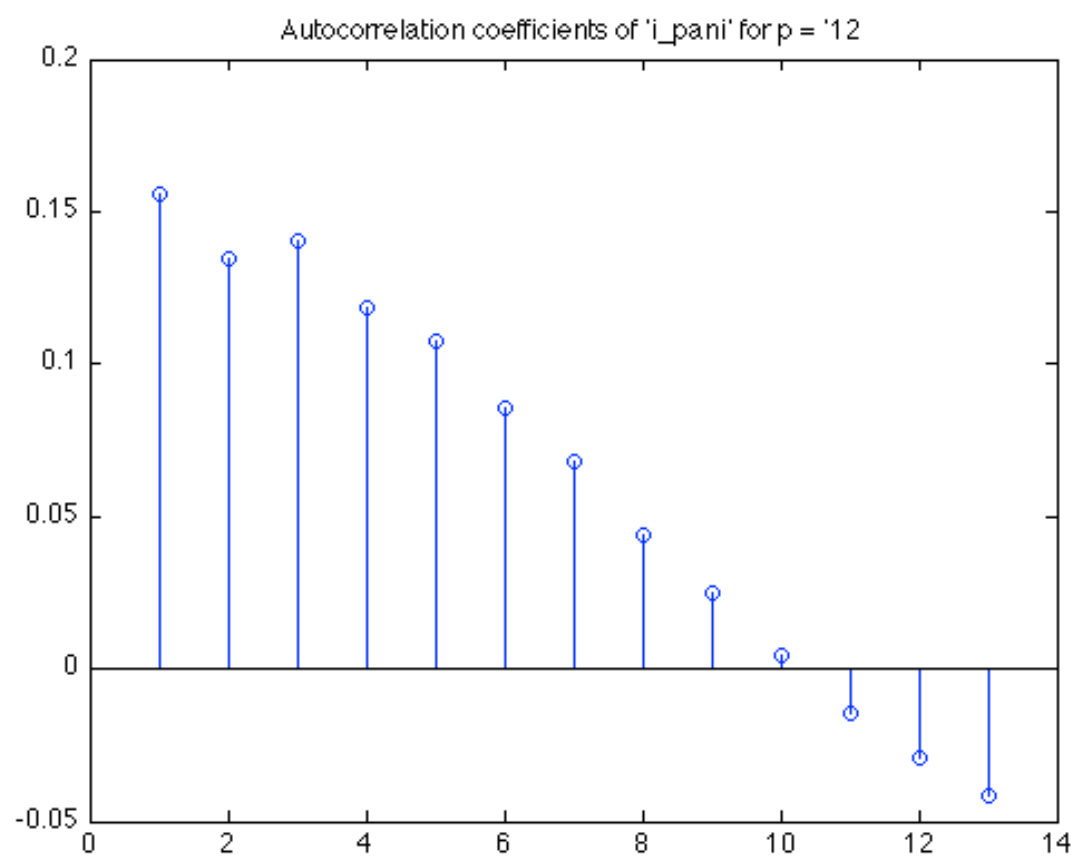


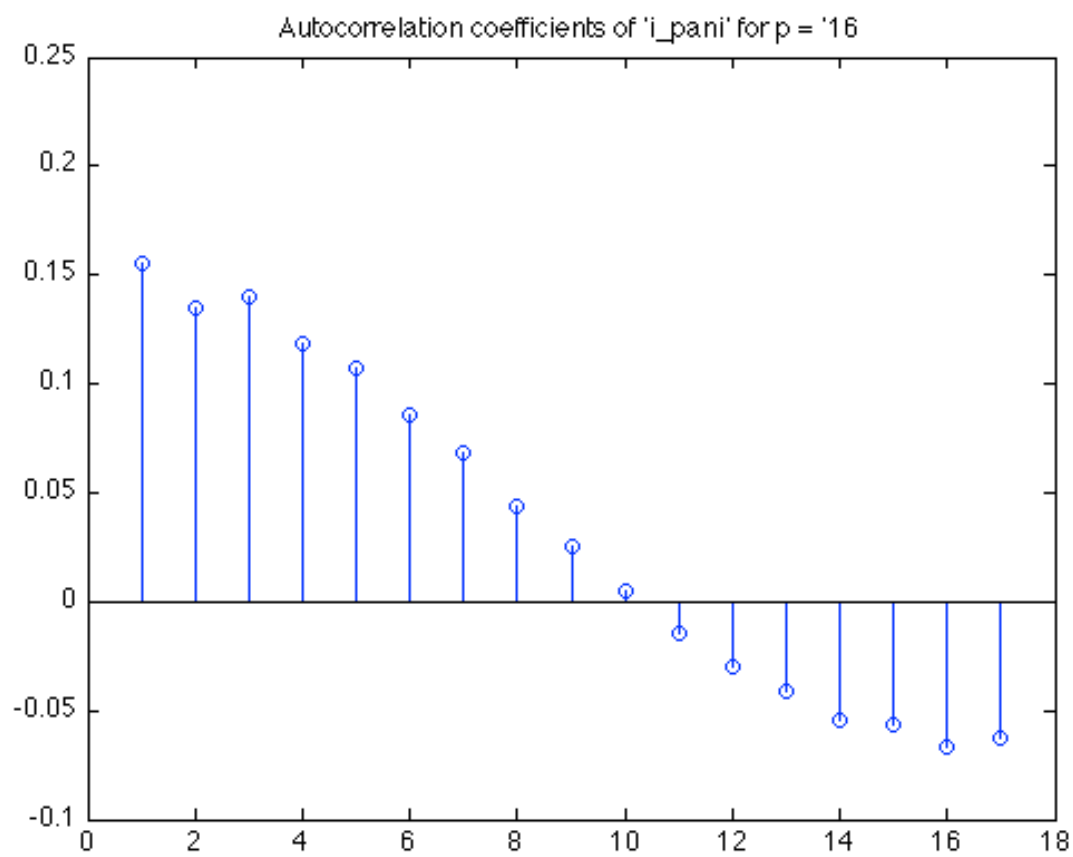




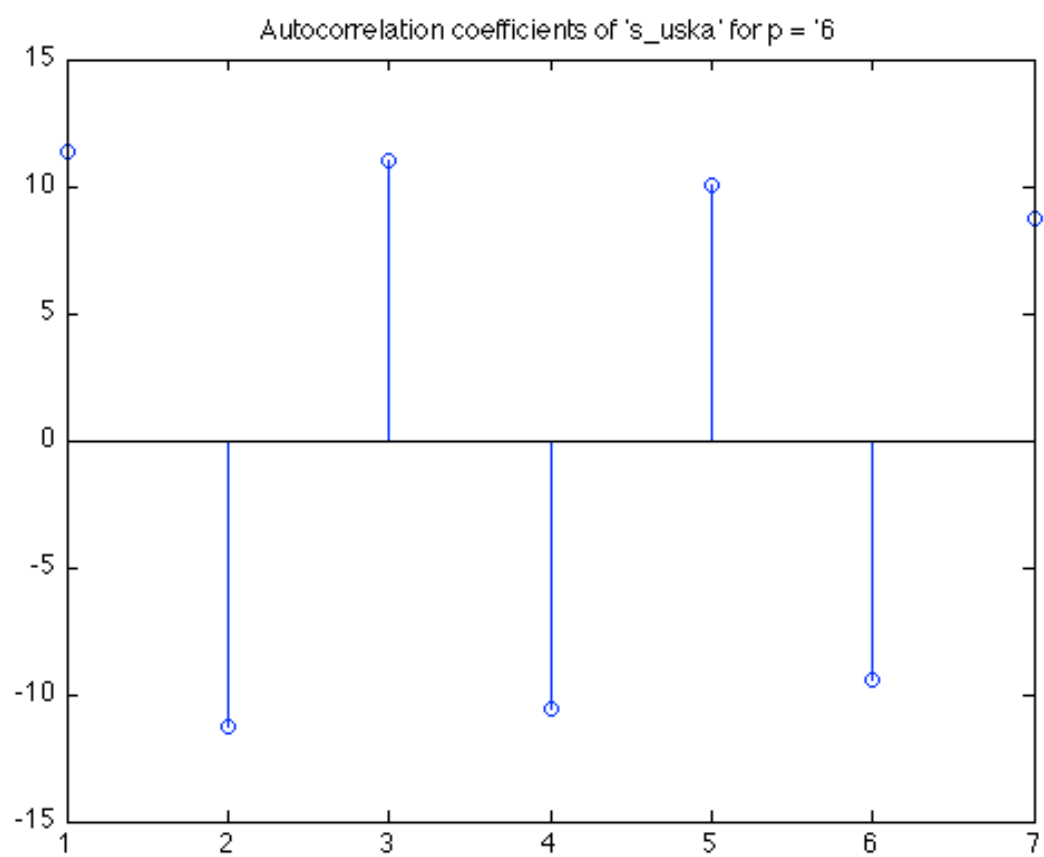


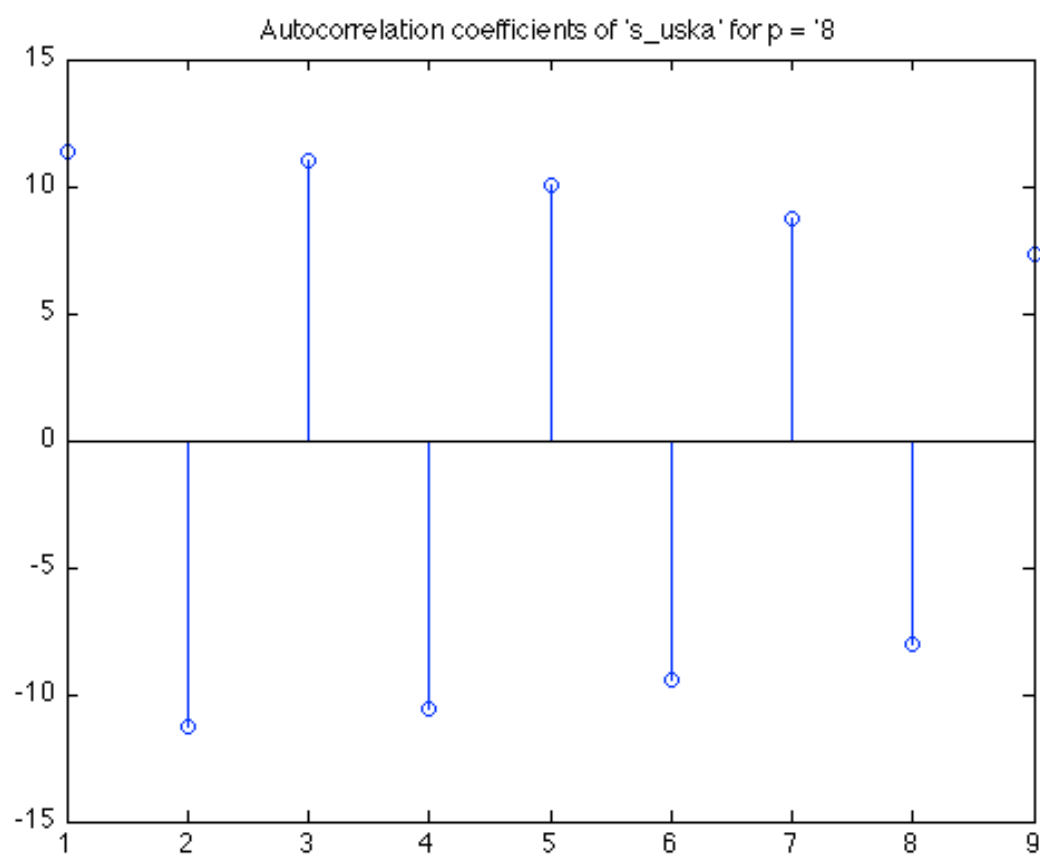


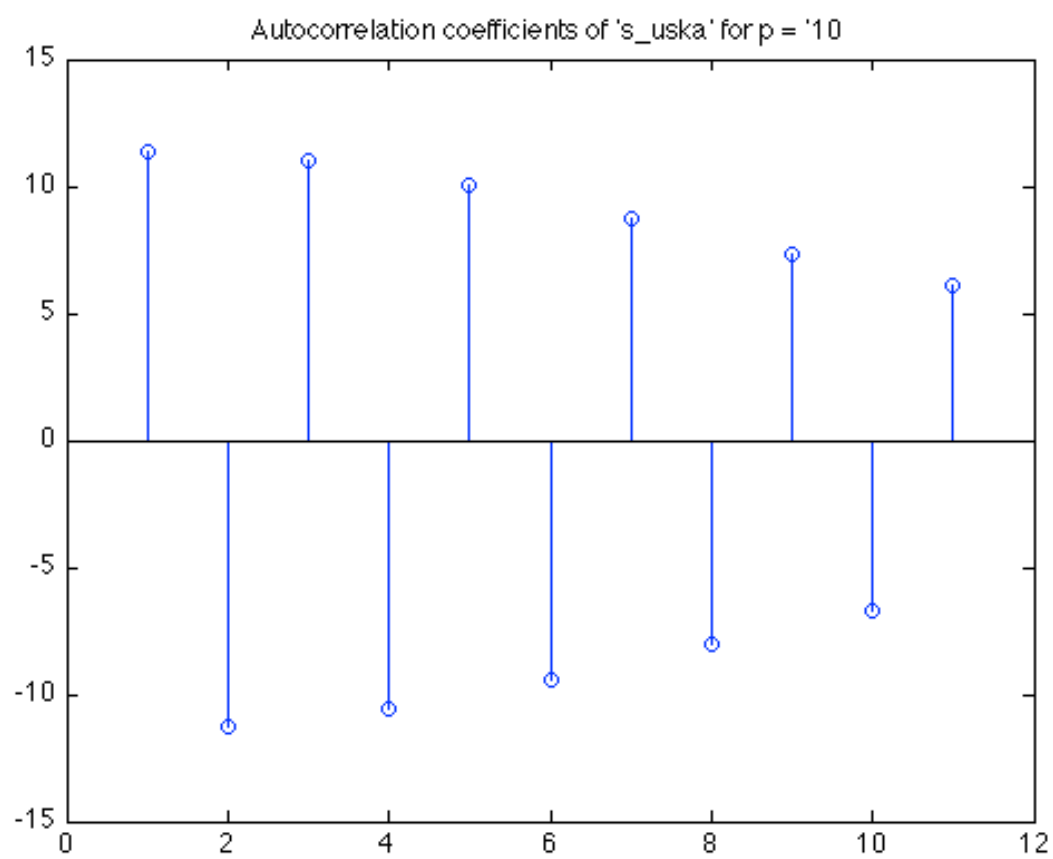


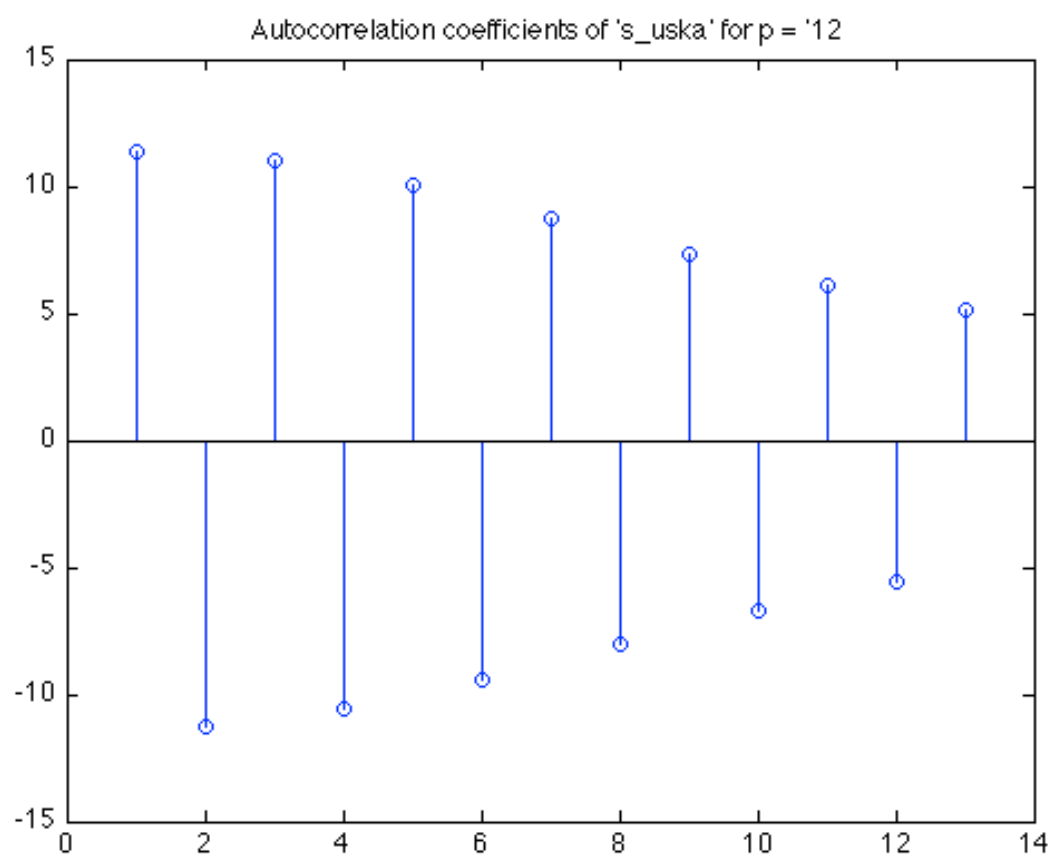


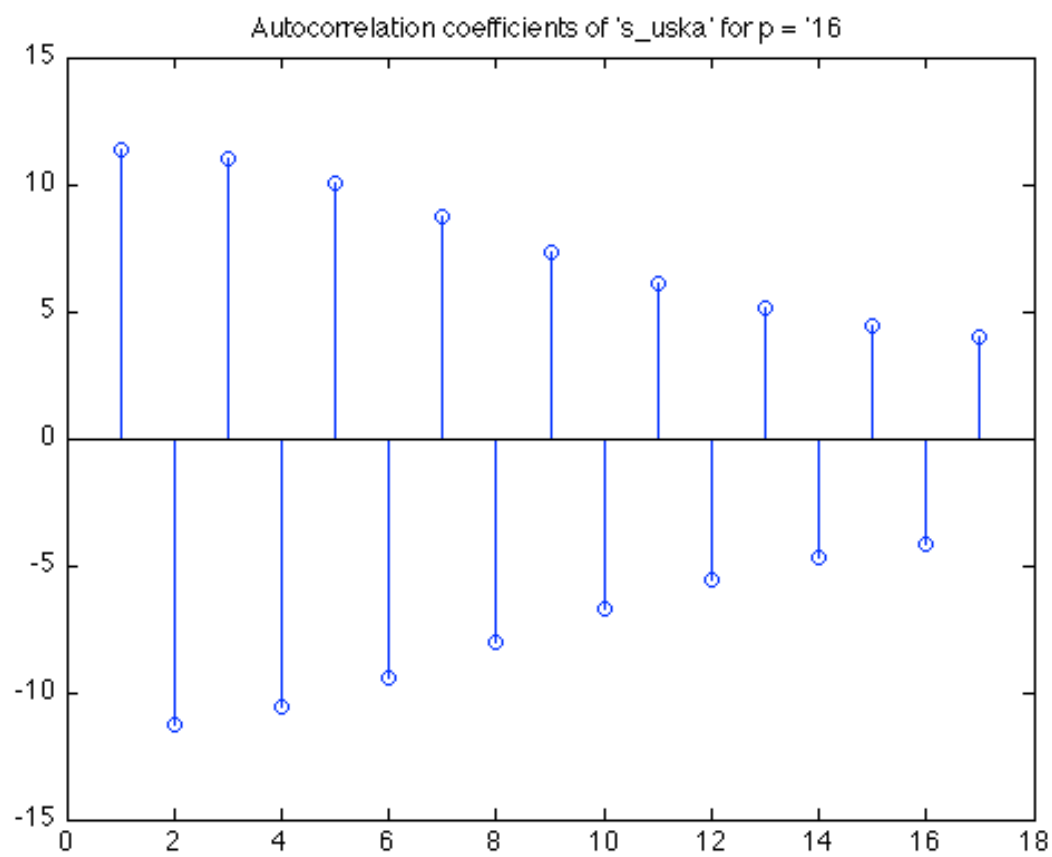




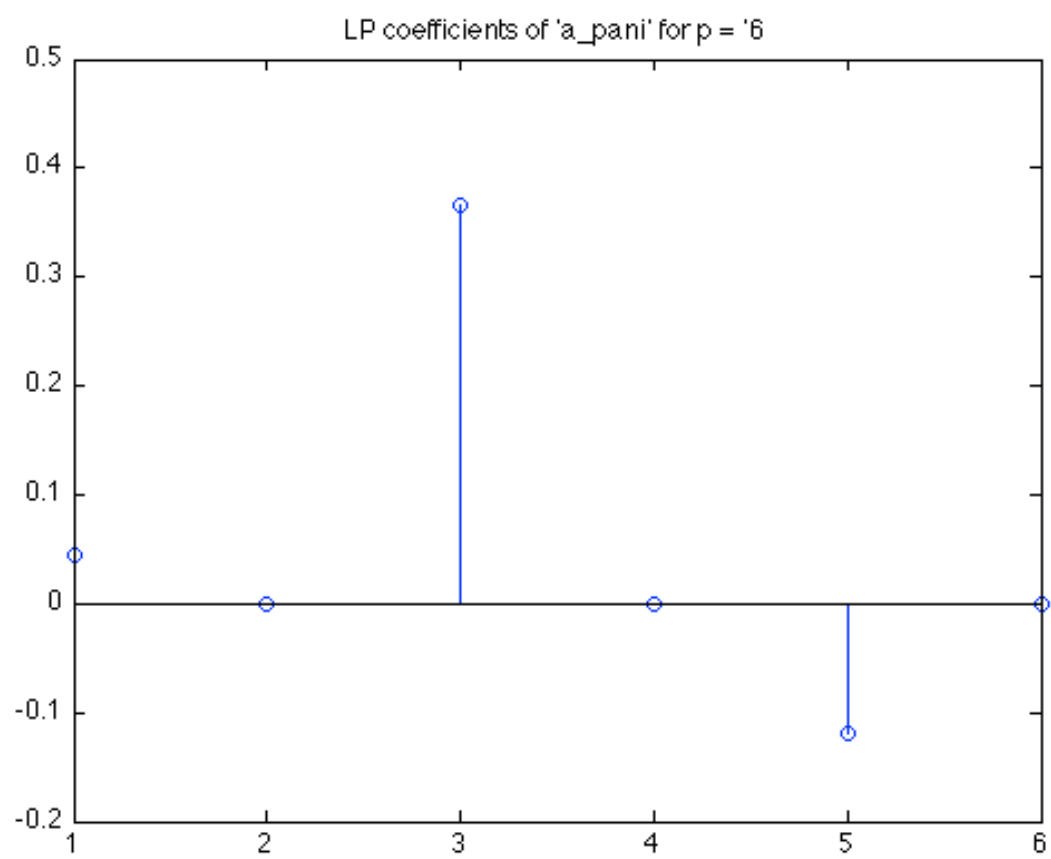


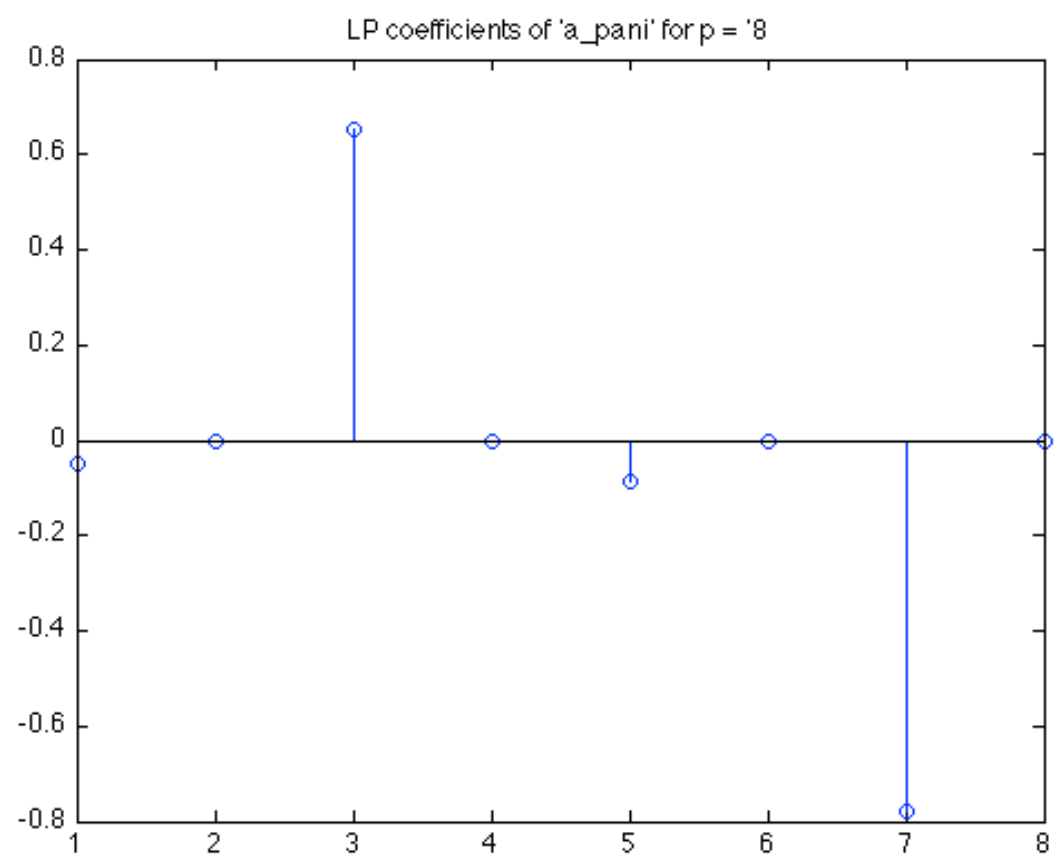


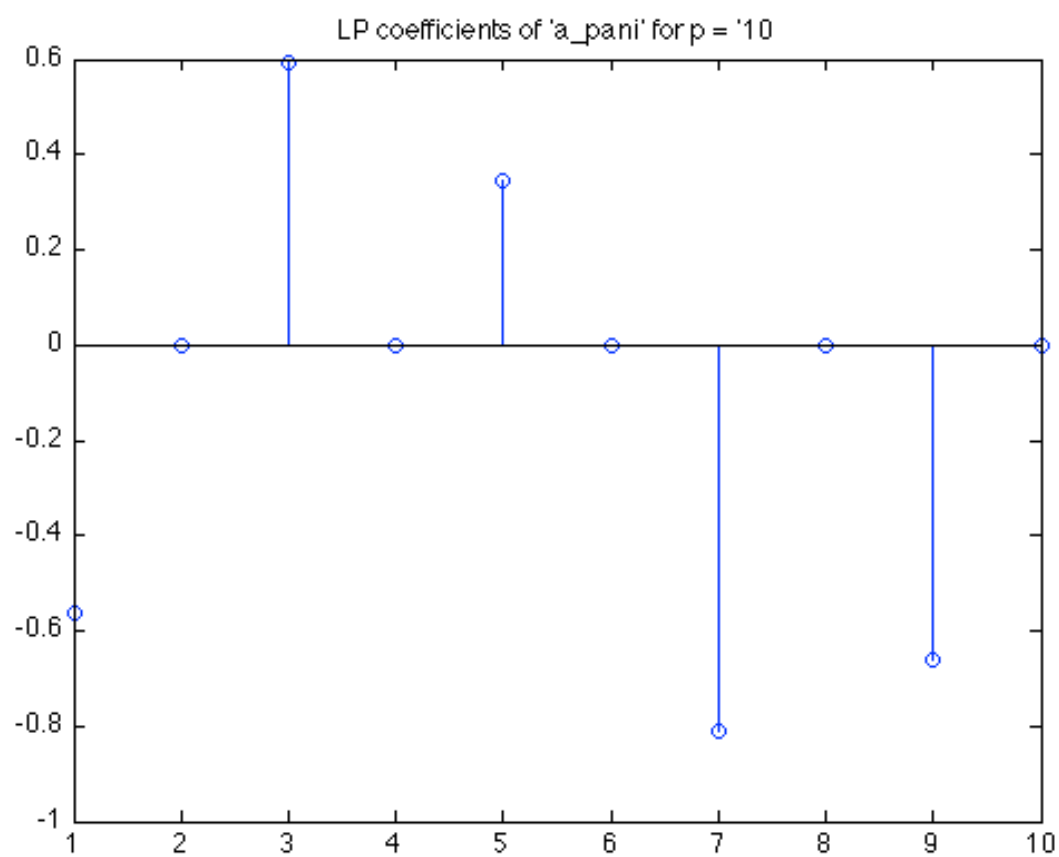




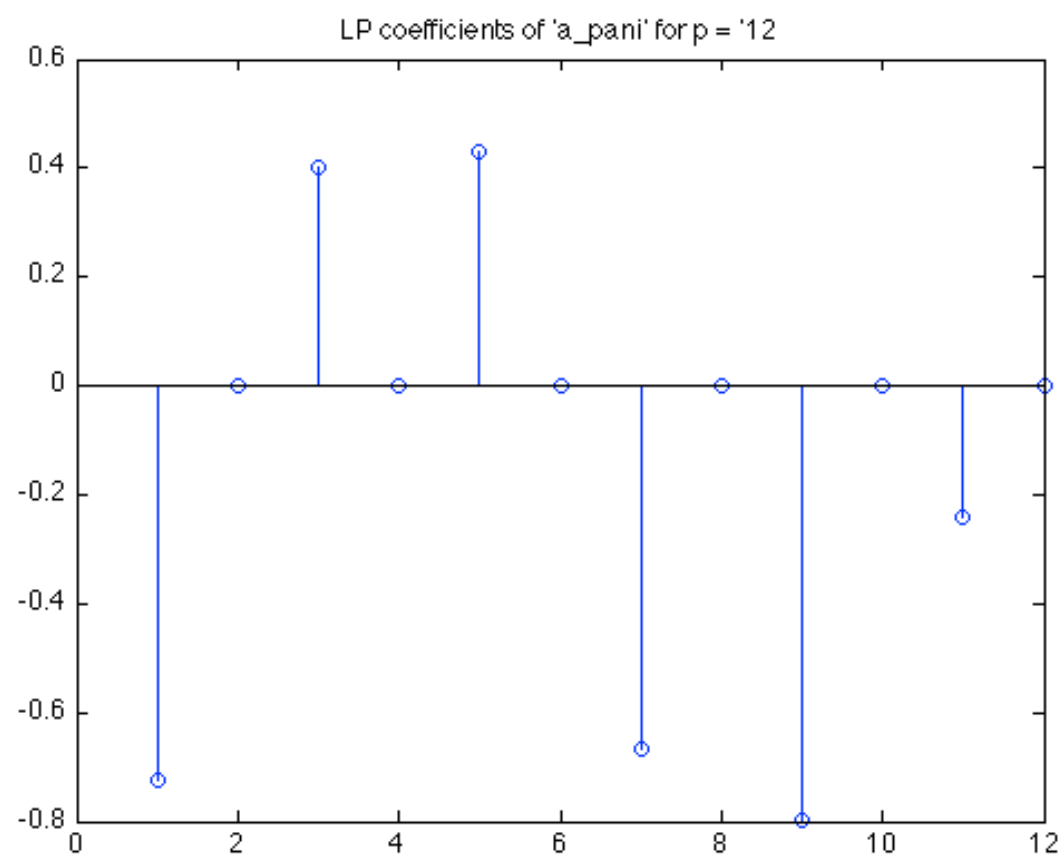
Now the LP coefficient plots:

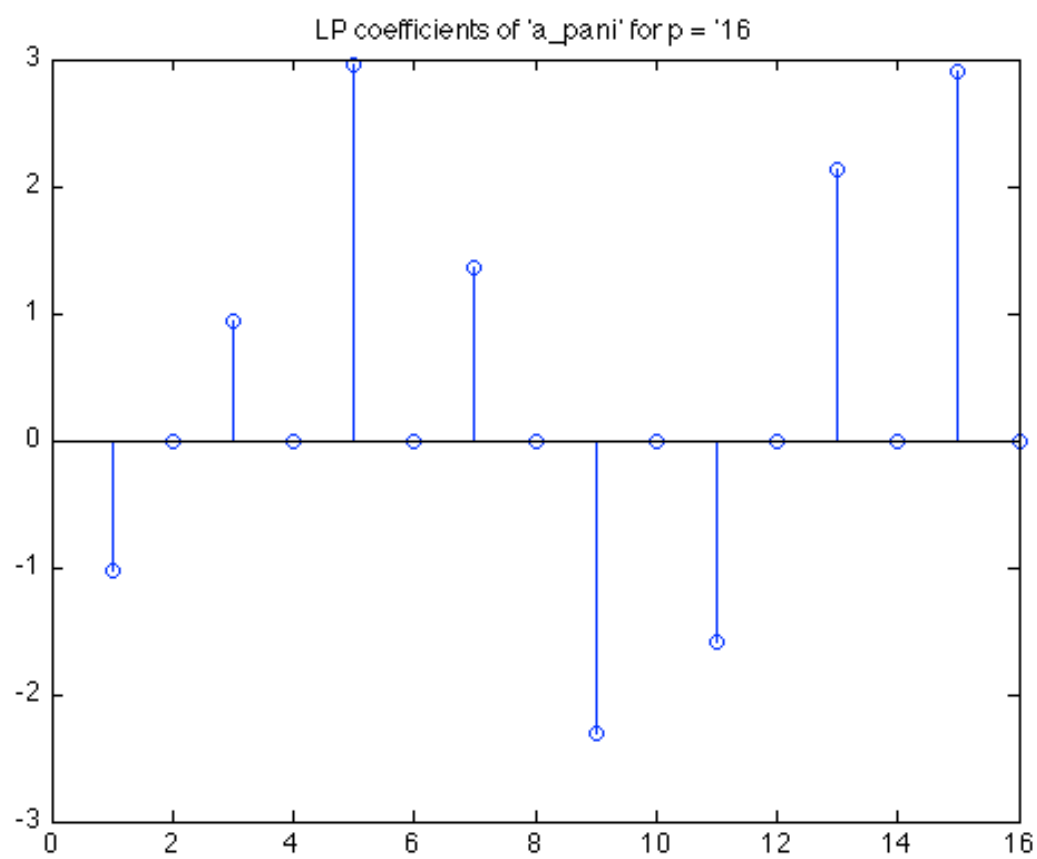


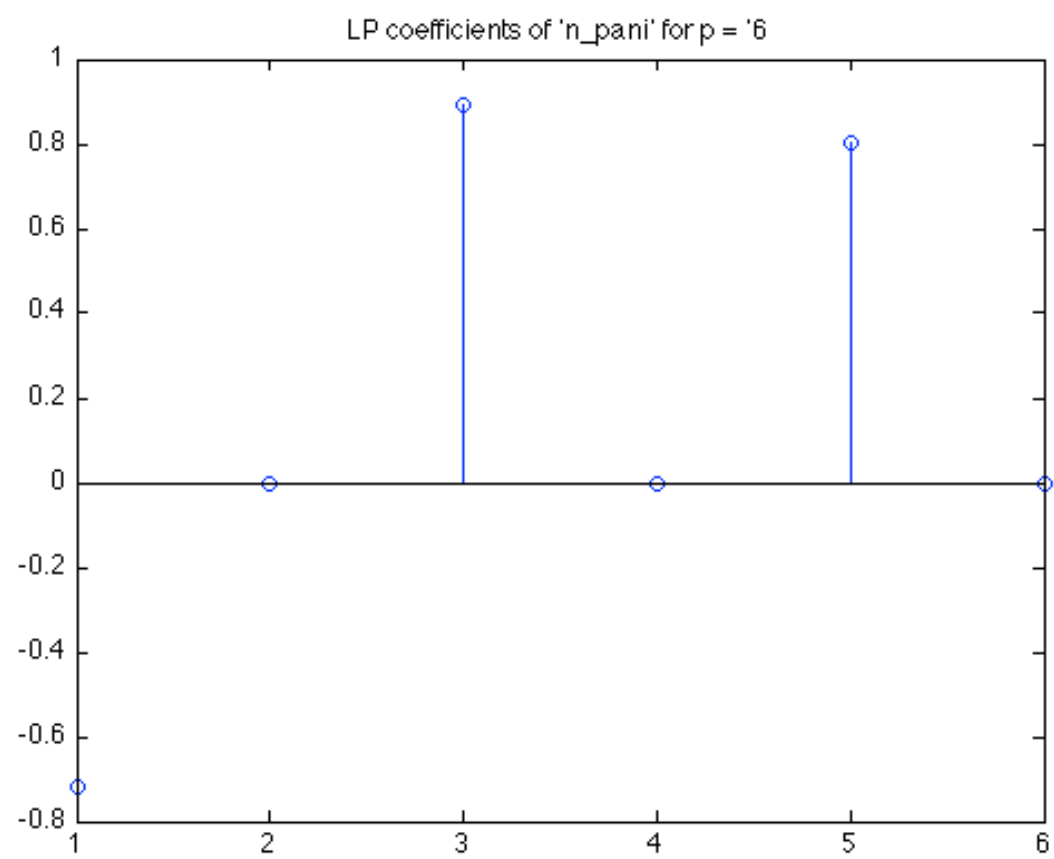


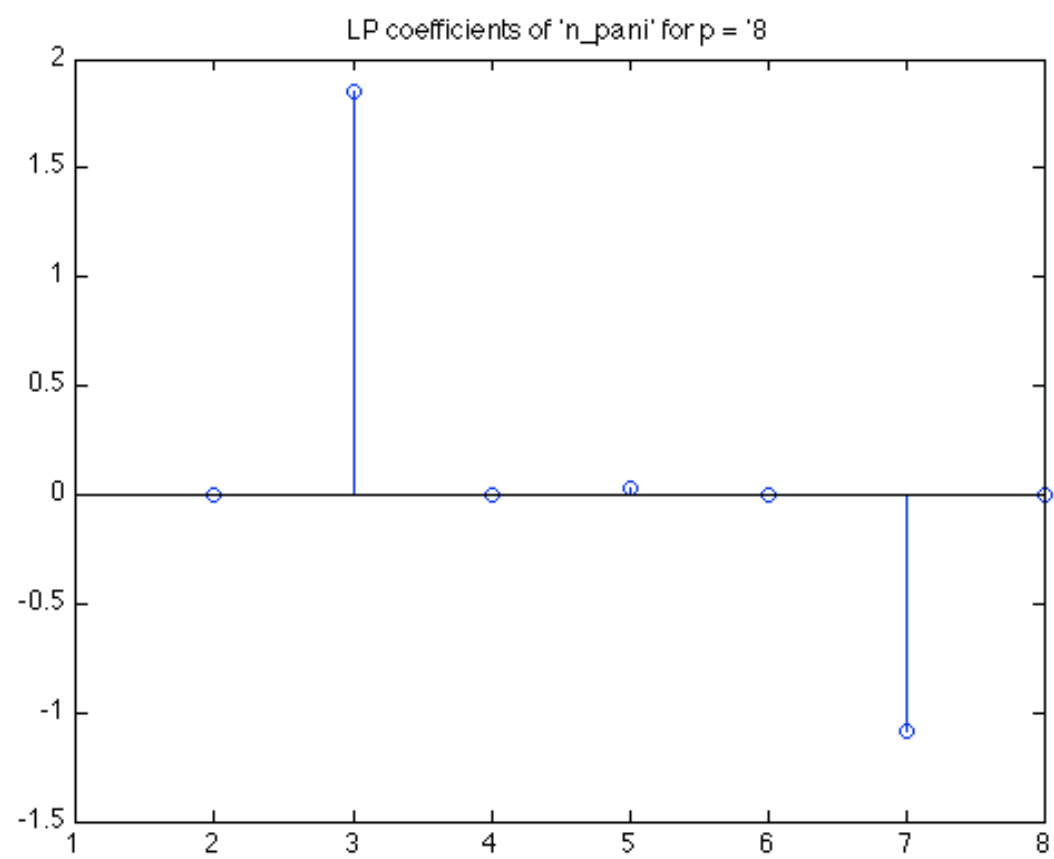


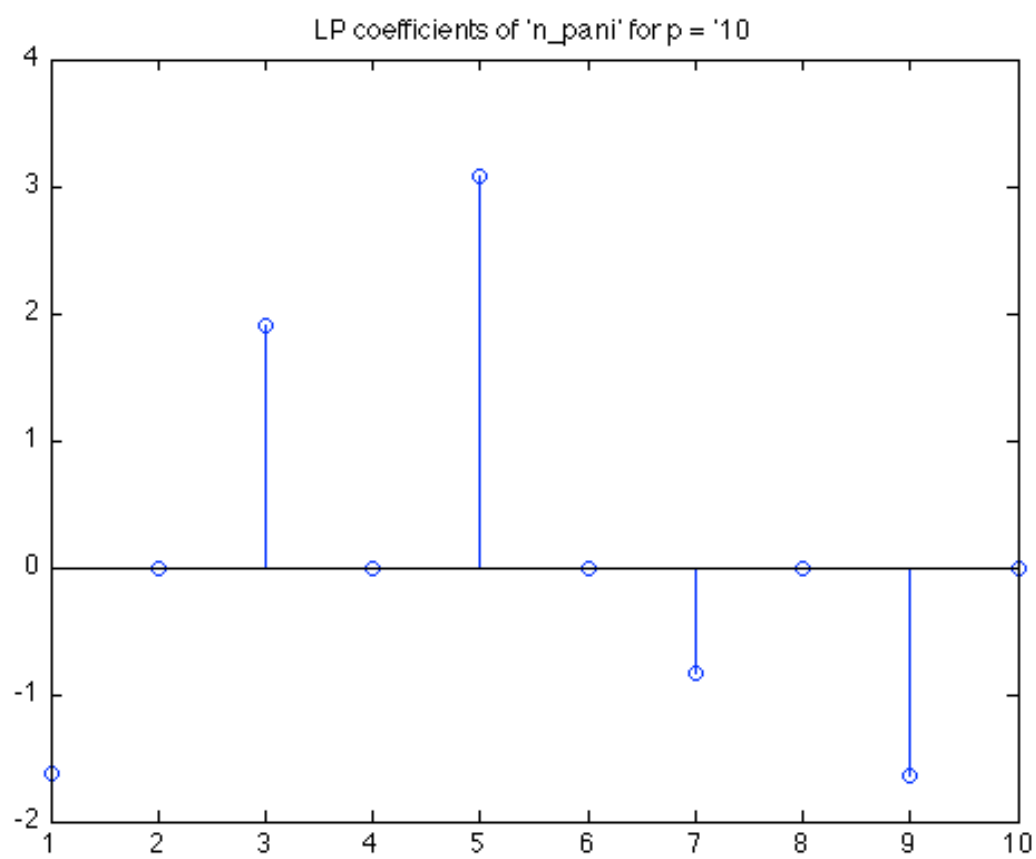


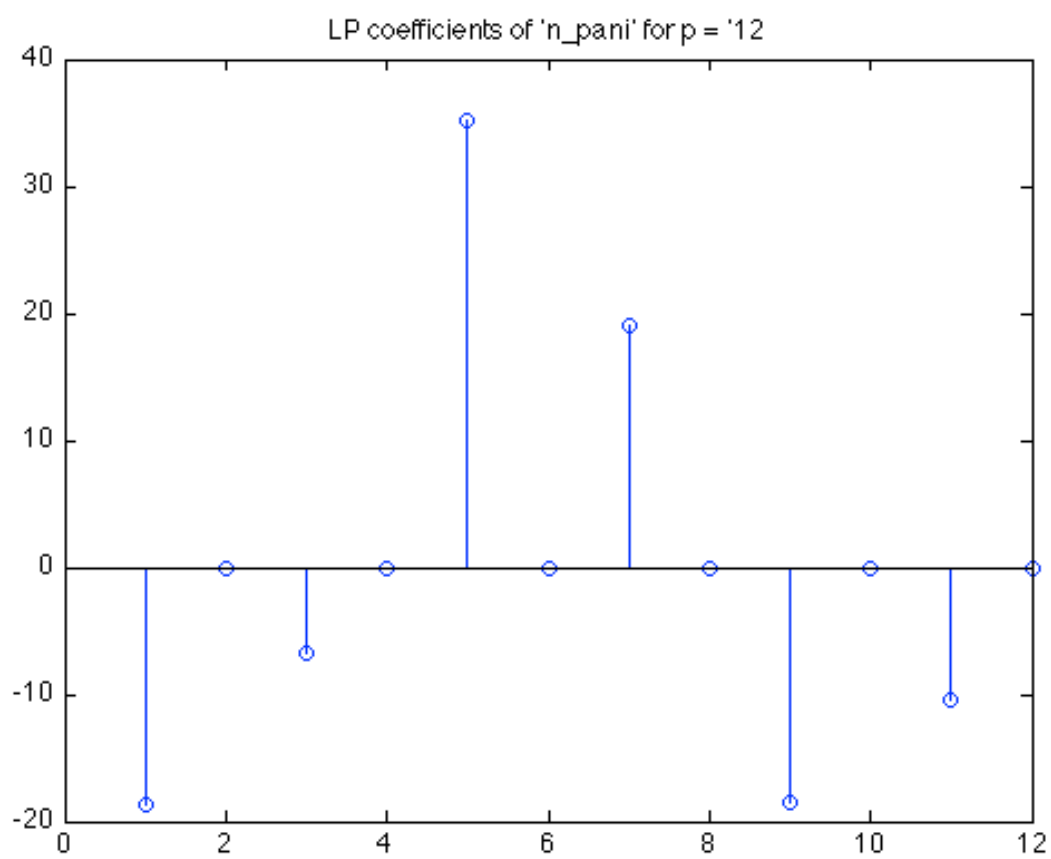


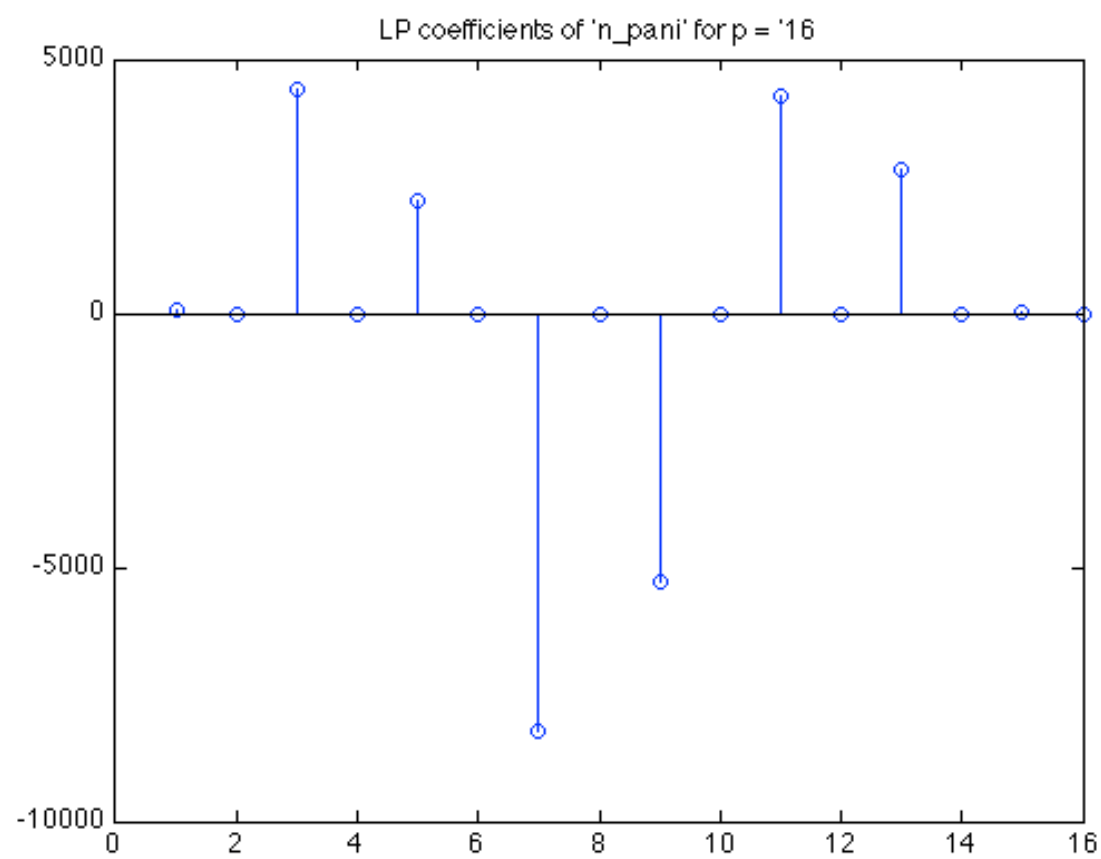


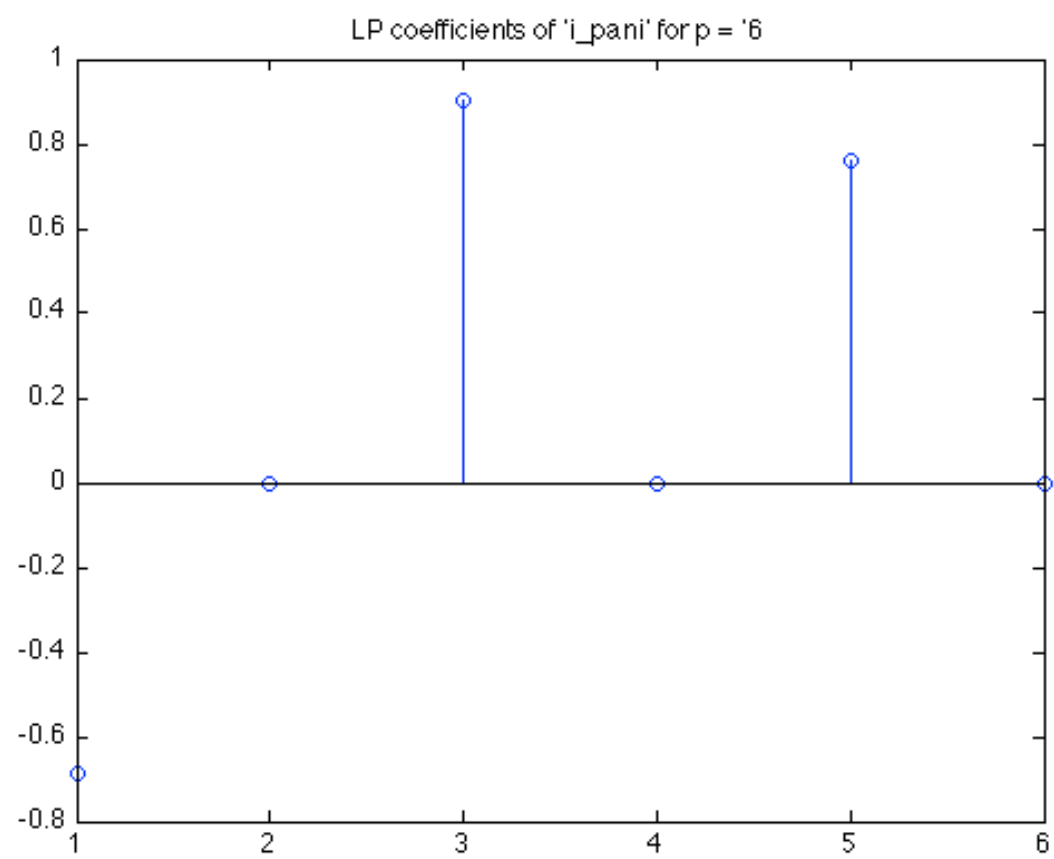




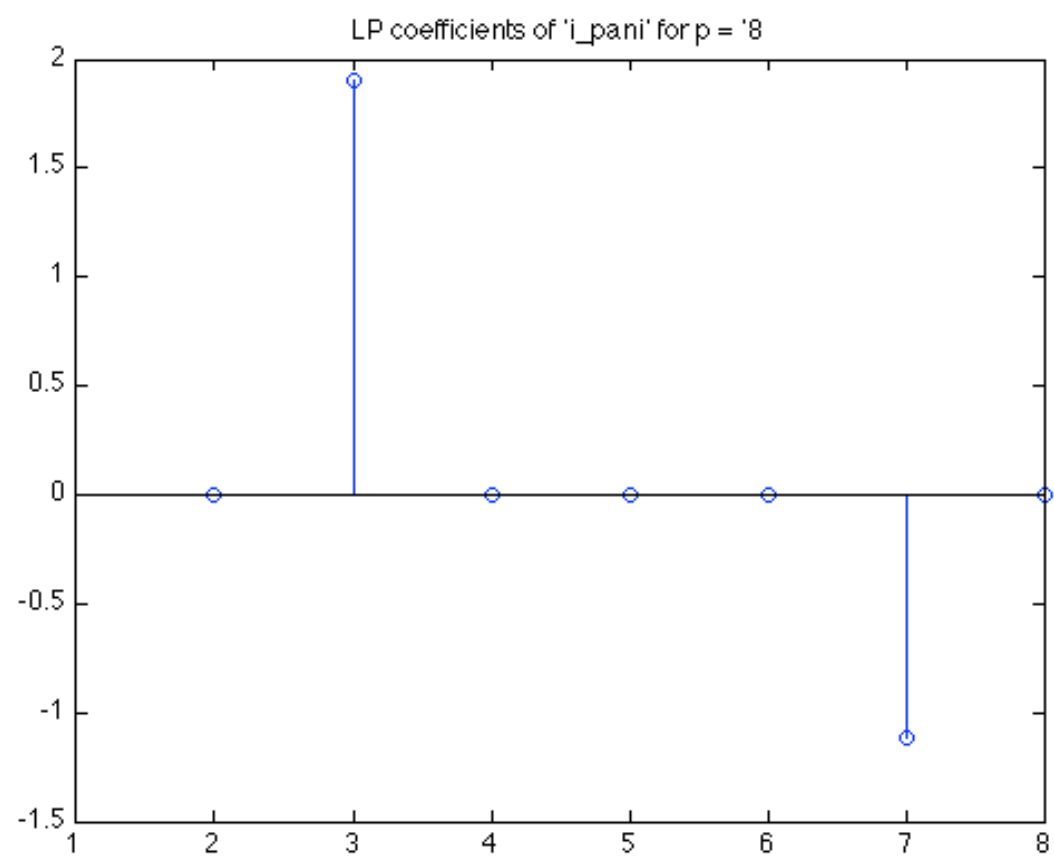


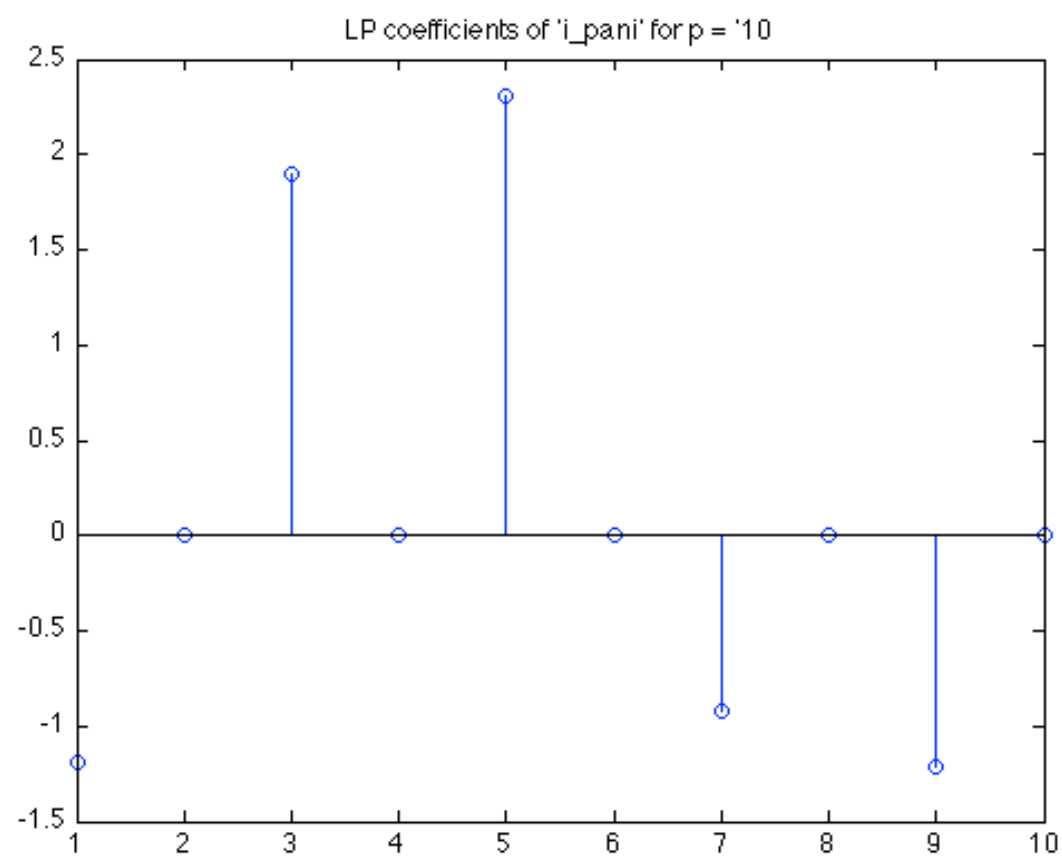


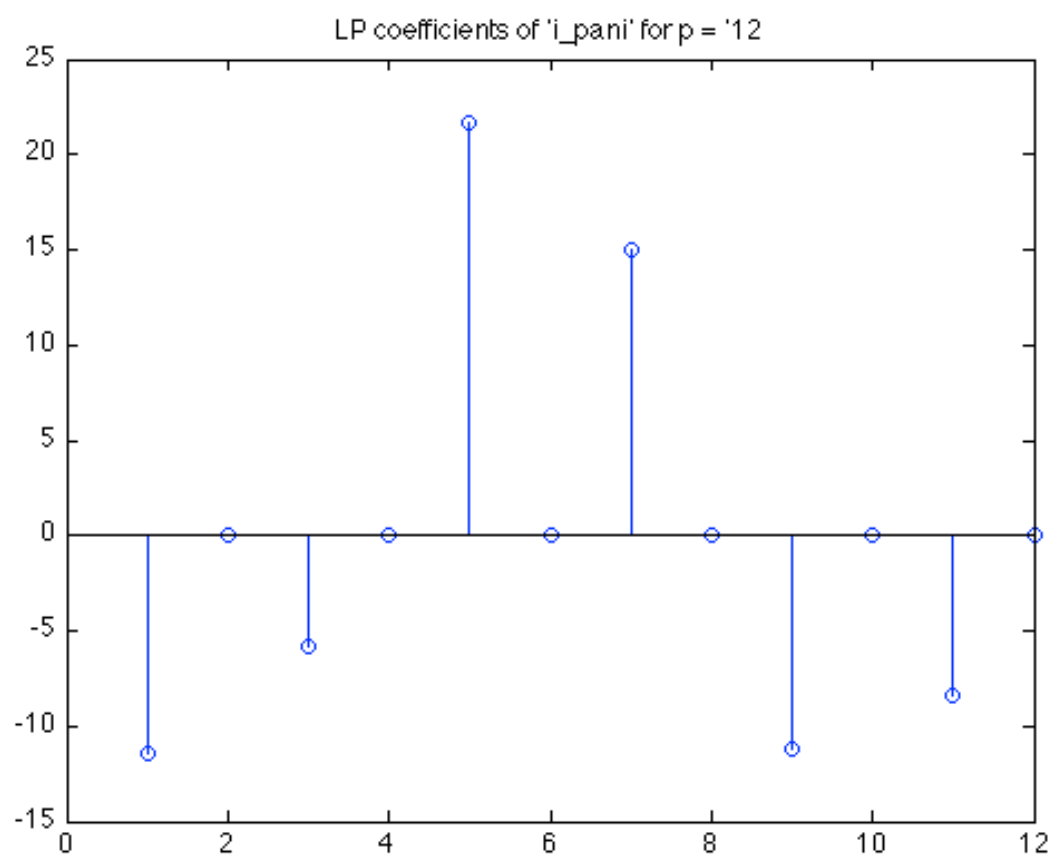


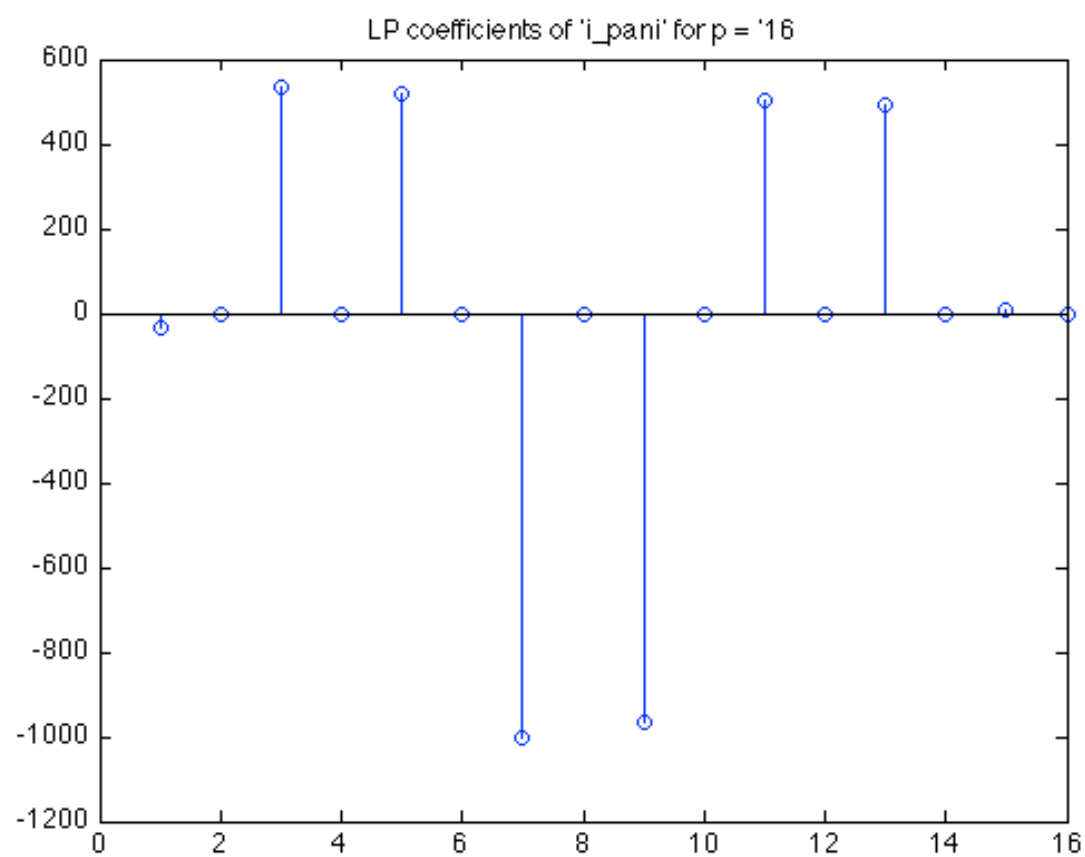


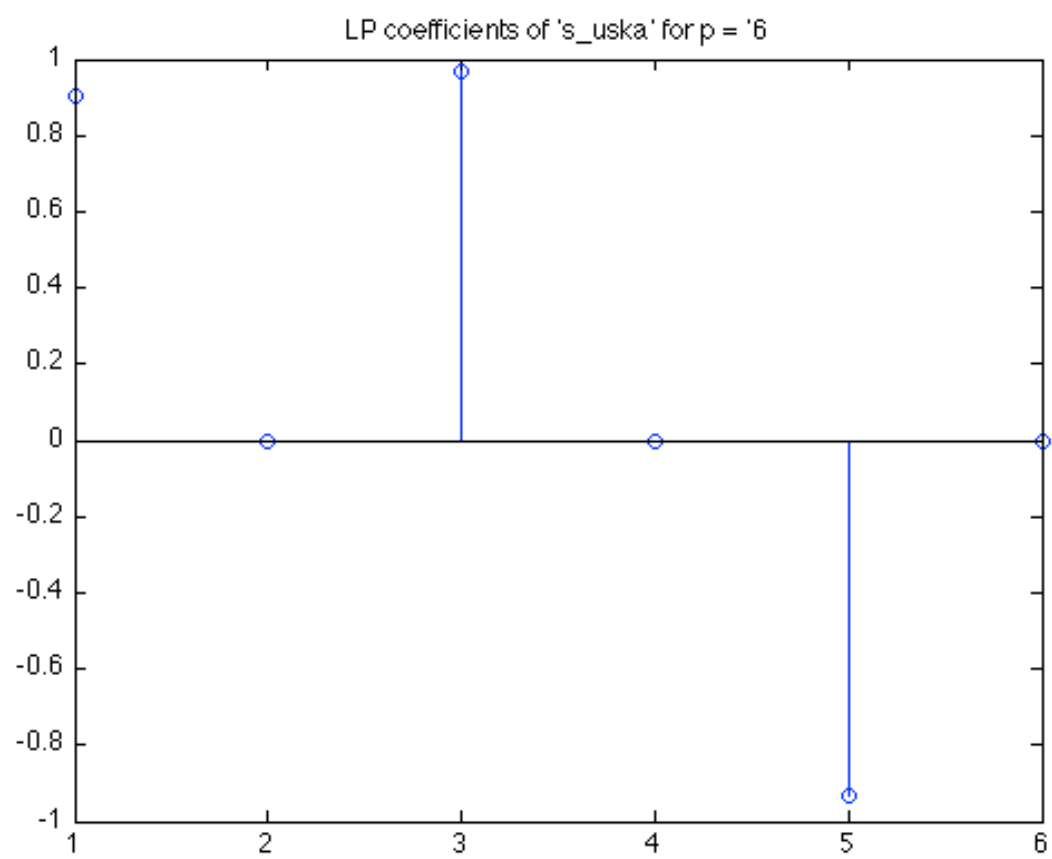


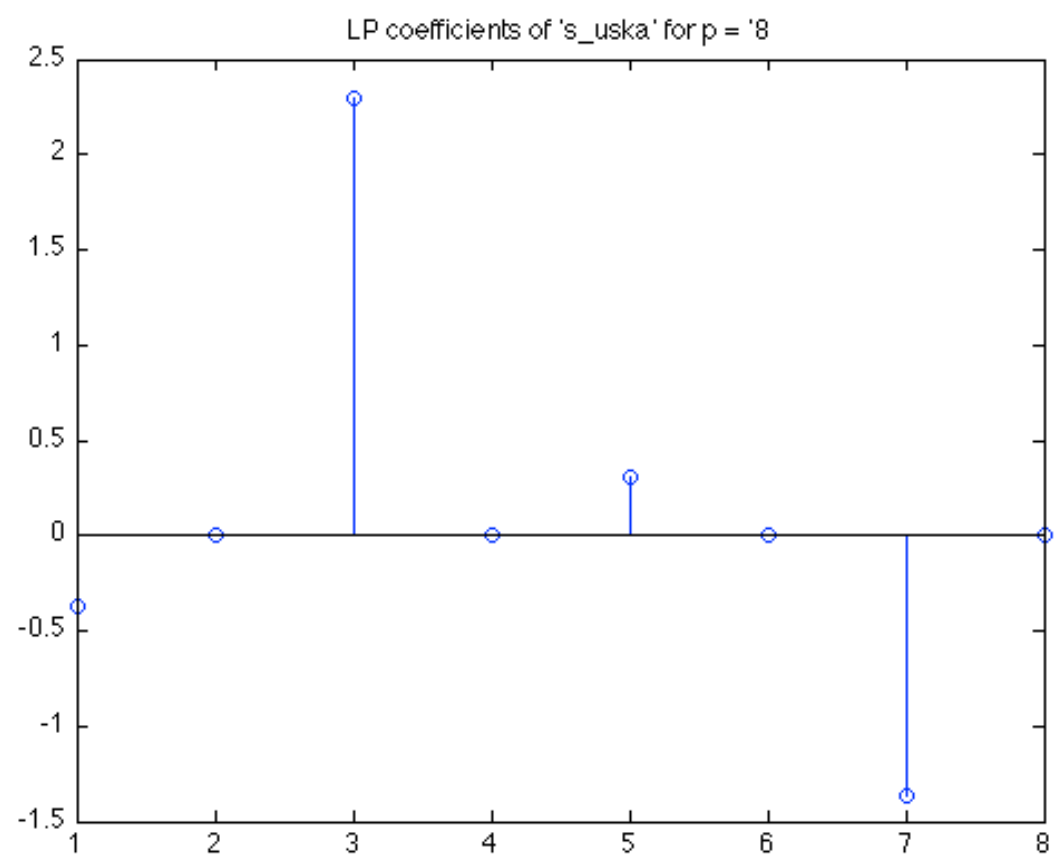


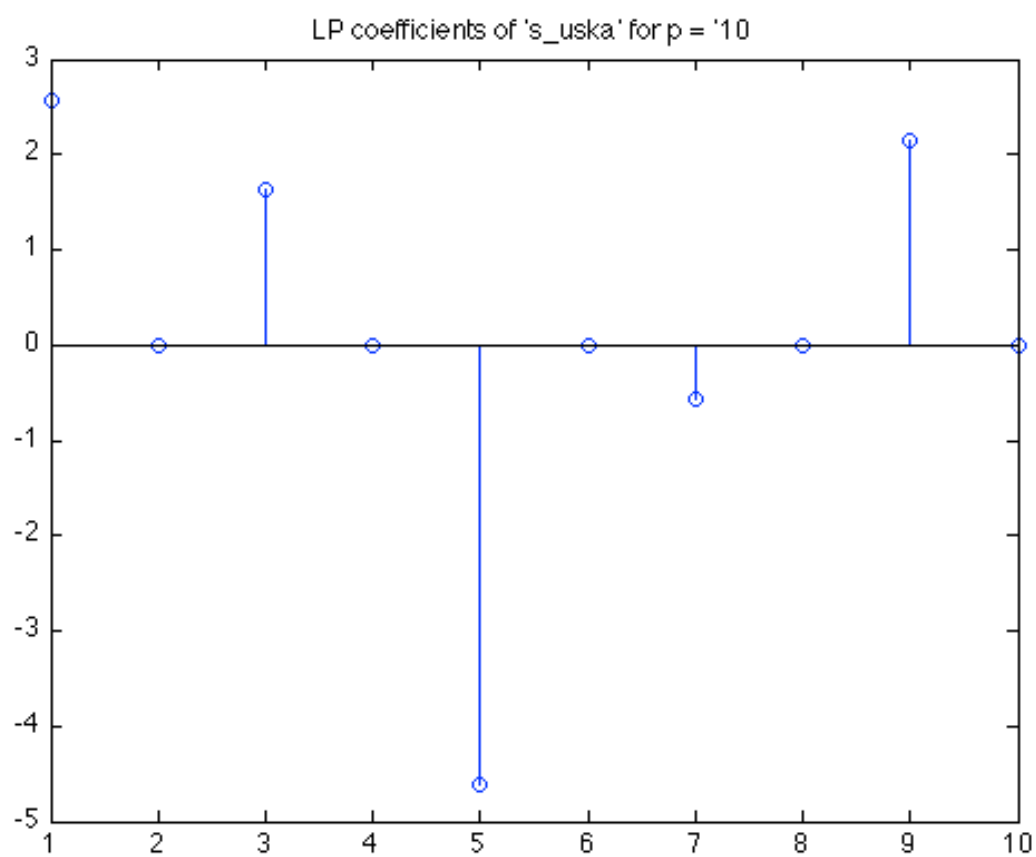


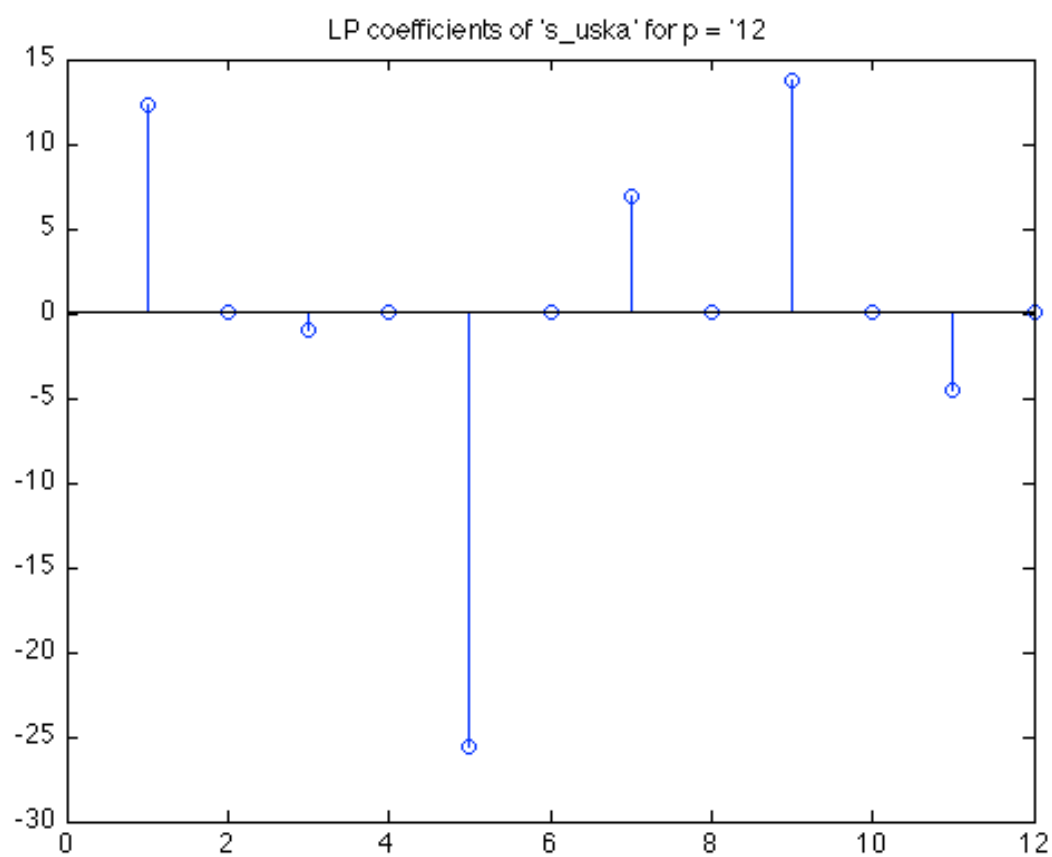




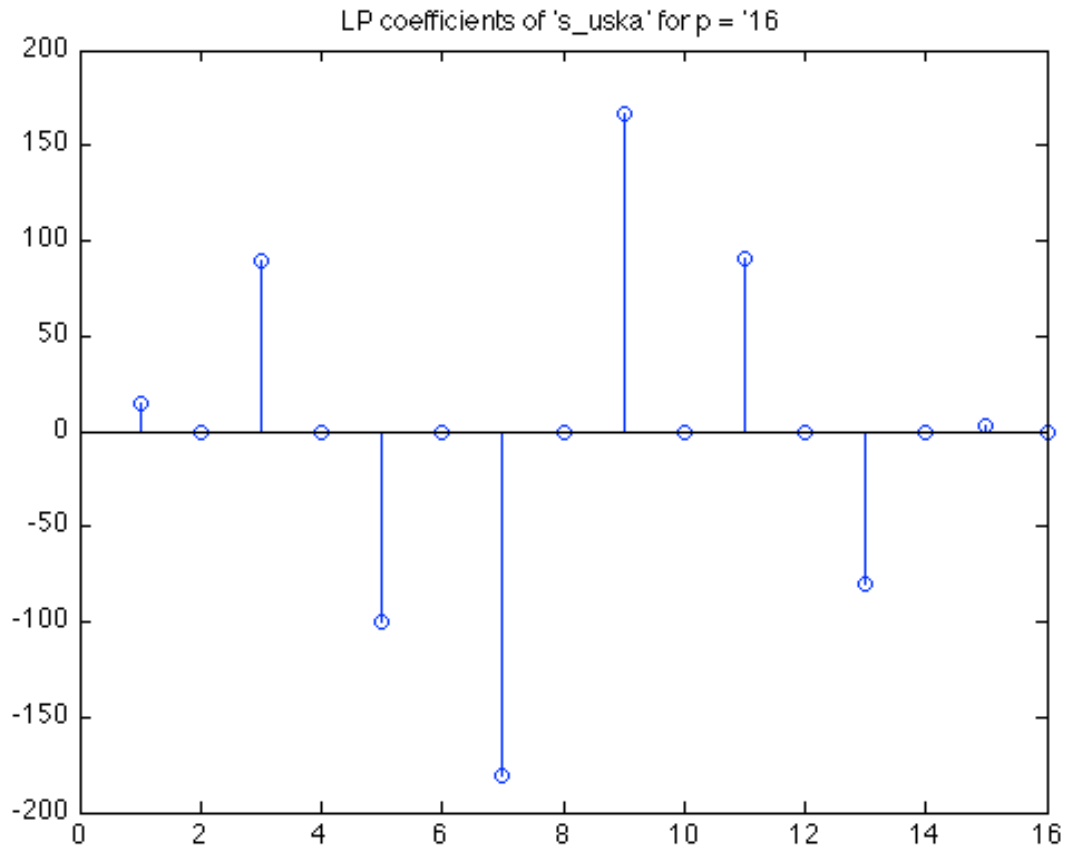












**The script:**

```
close all; clear all;

getLPCoefficients('a_pani', 6);
getLPCoefficients('a_pani', 8);
getLPCoefficients('a_pani', 10);
getLPCoefficients('a_pani', 12);
getLPCoefficients('a_pani', 16);

getLPCoefficients('n_pani', 6);
getLPCoefficients('n_pani', 8);
getLPCoefficients('n_pani', 10);
getLPCoefficients('n_pani', 12);
getLPCoefficients('n_pani', 16);

getLPCoefficients('i_pani', 6);
getLPCoefficients('i_pani', 8);
getLPCoefficients('i_pani', 10);
getLPCoefficients('i_pani', 12);
getLPCoefficients('i_pani', 16);
```

```

getLPCoefficients('s_uska', 6);
getLPCoefficients('s_uska', 8);
getLPCoefficients('s_uska', 10);
getLPCoefficients('s_uska', 12);
getLPCoefficients('s_uska', 16);

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

figure, stem(LPCoeffs);
title(['LP coefficients of ', inputFile, ' for p = ', num2str(poleOrder), ''], 'interpreter', 'none');

```

### **The functions:**

```

function getLPCoefficients(inputFile, poleOrder)

autocorrCoeffs = getAutoCorrCoefficients(inputFile, poleOrder);
LPCoeffs = levinsonDurbin(autocorrCoeffs, poleOrder);
figure, stem(LPCoeffs);
title(['LP coefficients of ', inputFile, ' for p = ', num2str(poleOrder), ''], 'interpreter', 'none');

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( data , p)

```

```

rx = data; % 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 = 1:p;
    G(j) = rx(j+1);
    sum = 0;
    for i = 2:j;
        sum = sum + (a(j, i)*rx(j-i+1));
    end
    G(j) = G(j) + sum;
    reflected(j+1) = -G(j)/e(j);

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

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