

```

% % -----
% Name : Manan Vyas
% USCID: 7483-8632-00
% Email: mvyas@usc.edu
% EE519 : Speech Recognition : Final Exam : P4
% % -----

% Setup
clc;
clear all;
close all;
% Read the *wav file
load('final2014_p4.mat');
Fs = 10000; %Hz
n = length(speech1_10k); % len

% Part (A) Autocorrelation of the input windowed speech sequence
% Apply a 25ms Hamming window ...
% 10k samples/sec => 25msec = 250 samples => Length of Hamming window
wLen = 250;
window = hamming(wLen);
wSpeech = speech1_10k.*window'; % windowed speech
% Autocorrelation.
speechAutoCorr = xcorr(wSpeech,250);
% speechAutoCorr = speechAutoCorr./(abs(max(speechAutoCorr)));
figure
subplot(3,1,1);
plot(speech1_10k); title('Speech Segment');xlabel('samples n ->');ylabel('Amplitude');
subplot(3,1,2);
plot(wSpeech); title('windowed Speech Segment');xlabel('samples n ->');ylabel('Amplitude');
subplot(3,1,3);
plot(speechAutoCorr); title('Autocorrelation');xlabel('k ->');ylabel('Rn[k]');
axis([0 length(speechAutoCorr) min(speechAutoCorr) max(speechAutoCorr)]);
speechAutoCorr = speechAutoCorr(251:length(speechAutoCorr));

% Part (B) Setting up Toeplitz matrix
P = 4;
a = (speechAutoCorr(1:P));
r = (speechAutoCorr(2:(P+1)))';
a = toeplitz(a); % toeplitz => get the 4x4 autocorr matrix as reqd

% Part (C) : Solve for LP co-effs using matrix inversion
L = r'*inv(a);
% L = (-inv(a)*r)';
LPCoeffs (1,1:length([1,L])) = [1,L]

% part (D) : Frequency Response of the LP Filter vs. Log |FFT(Windowed Signal)|
% rn0 = speechAutoCorr(1);
% rnK = speechAutoCorr(2:(P+1));
A = speechAutoCorr(1) - sum(LPCoeffs(2:P+1).*(speechAutoCorr(2:P+1)));
Hw = filt(A,[LPCoeffs(1) -LPCoeffs(2) -LPCoeffs(3) -LPCoeffs(4) -LPCoeffs(5)]);
figure
% plot(abs(A/[LPCoeffs(1) -LPCoeffs(2) -LPCoeffs(3) -LPCoeffs(4) -LPCoeffs(5)]));
[M P] = freqz(A,[LPCoeffs(1) -LPCoeffs(2) -LPCoeffs(3) -LPCoeffs(4) -LPCoeffs(5)]);
%Log|| of FFT of windowed signal
scale = 0:(1024/2-1); %-(1024/2):(1024/2-1);

```

```

figure
hold on
yy = (20*log(abs(fft(wSpeech,1024))));
yy = yy(1:512);
% [M P] = freqz(A,[LPCoeffs(1) -LPCoeffs(2) -LPCoeffs(3) -LPCoeffs(4) -LPCoeffs(5)]);
plot(20*log(abs(M)));
plot(yy,'r-');
axis tight;
xlabel('FFT Points(0->512)'); ylabel('Magnitude (dB)'); title('Log Magnitude plot of FFT of
windowed signal overlayed with the Freq Response of LP filter');
hold off
% freqz(fft(wSpeech,256));

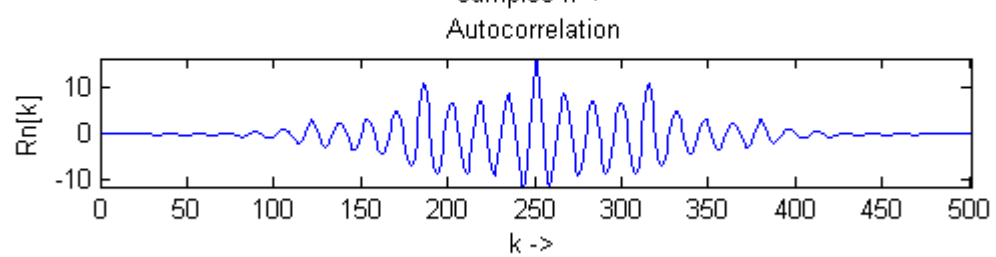
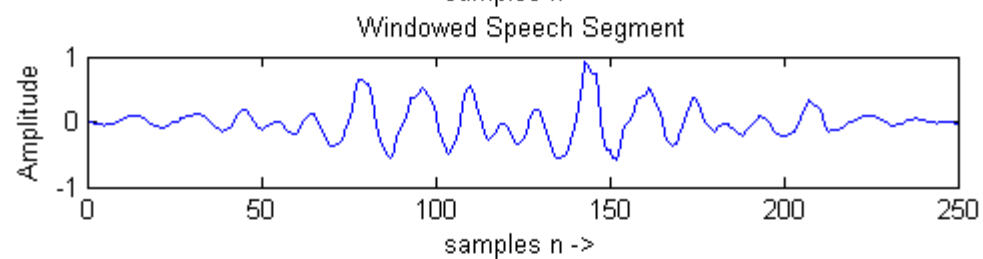
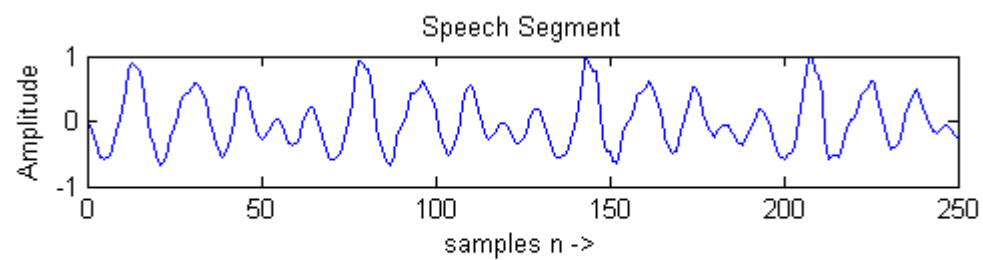
% -- Use LPC to verify
% [pC G] = lpc(wSpeech,4);
% figure
% freqz(G,pC);

% Part (E) : Compute Error Sequence and Compare with the Original Speech
for i=5:1:length(speech1_10k)
    errorSeq(i) = speech1_10k(i)-(speech1_10k(i-4)*LPCoeffs(5) + speech1_10k(i-3)*LPCoeffs(4)
+ speech1_10k(i-2)*LPCoeffs(3) + speech1_10k(i-1)*LPCoeffs(2));
end
figure
hold on
plot(errorSeq,'r-');
plot(speech1_10k);
axis tight;
title('Speech signal and the Error Sequence'); xlabel('n ->'); ylabel('Amplitude');
legend('Error sequence','Original Speech signal');
hold off

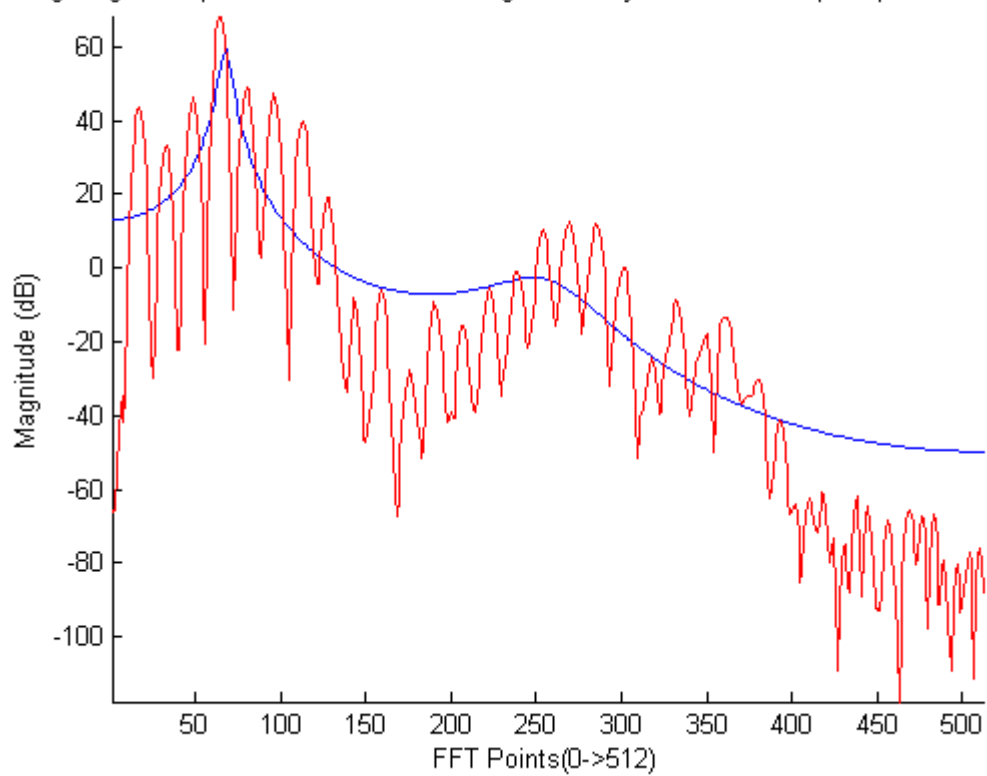
```

LPCoeffs =

1.0000      1.7975      -1.6233      1.1894      -0.6320



Log Magnitude plot of FFT of windowed signal overlayed with the Freq Response of LP filter



Speech Signal and the Error Sequence

