EE-414 Speech Processing Lab

Lab-10

Estimation Of Pitch From Speech Signals

180020002

Aim:

● Develop a method for the estimation of pitch by the autocorrelation of speech signal.

● Develop a cepstrum pitch estimation method.

● Develop a simple inverse filtering technique(SIFT) pitch estimation method.

● Comparison of all these three methods.

Theory:

Speech signal can be classified into voiced, unvoiced and silence regions. The near periodic vibration of vocal folds is excitation for the production of voiced speech. The random like excitation is present for unvoiced speech. There is no excitation during silence region. Majority of speech regions are voiced in nature that includevowels, semivowels and other voiced components. The voiced regions looks like a near periodic signal in the time domain representation. In a short term , we may treat the voiced speech segments to be periodic for all practical analysis and processing. The periodicity associated with such segmentsis defined is 'pitch period To' in the time domain and 'Pitch frequency or Fundamental Frequency Fo' in the frequency domain. Unless specified, the term 'pitch' refers to the fundamental frequency ' Fo'. Pitch is an important attribute of voiced speech. It contains speaker-specific information. It is also needed for speech coding task. Thus estimation of pitch is one of the important issue in speech processing.

There are a large set of methods that have been developed in the speech processing area for the estimation of pitch. Among them the three mostly used methods include, autocorrelation of speech, cepstrum pitch determination and single inverse filtering technique (SIFT) pitch estimation. One success of these methods is due to the involvment of simple steps for theestimation of pitch. Even though autocorrelation method is of theoritical interest, it produce a frame work for SIFT methods.

Pitch estimation by Autocorrelation method

The information about pitch period 'To' is more pronounced in the autocorrelation sequence of voiced speech compared to the speech segment itsely. The 'To' information is more pronounced in the autocorrelation sequence compared to speech. By that we mean, the second largest peak is the autocorrelation sequence, represents To and can be picked up easily by a simple peak picking algorithm compared to finding 'To' from the speech segment itself. Hence autocorrelation method is preferred over other direct methods of pitch estimation from speech.

There is no prominent peak as in the case of voiced speech. This is the fundamental distinction between voiced and unvoiced speech.

Cepstrum Pitch Determination

The main limitation of pitch estimation by the auto correlation of speech is that there may be peaks larger than the peak corresponding to the pitch period T0 due to that of the vocal tract, As a result there may be picking of peaks and hence wrong estimation of pitch. The approach to minimize such errors is to separate the vocal tract and excitation source related information in the speech signal and there use the source information for pitch estimation. The ceptral analysis of speech provides such an approach.

The ceptrum of speech is defined as the inverse Fourier transform of the log magnitude spectrum. The cepstrum projects all the slowly varying components in log magnitude spectrum to the low frequency region and fast varying components to the high frequency regions. In the log magnitude spectrum, the slowly varying components represent the envolope corresponds to the vocal tract and the fast varying components to the excitation source. As a result the vocal tract and excitation source components get represented naturally in the spectrum of speech.

The initial few values in the cepstrum typically 13-15 cepstral values represent the vocal tract information. The large peak present after these initial values represent the excitation information. In particular, the pitch period T0 starting from the zeroth value in number of samples. As a result the peaks that may be occuring in case of autocorrelation analysis get naturally eliminatedin cepstrum pitch determination. Hence the merit of the cepstrum method.

The initial 13-15 values represent the vocal tract information and later about the excitation source. By comparing the plots it may be observed that there is no prominent peak in case of ceptrum of unvoiced speech after the 13-15 initial cepstral values. This is the main distinction between cepstrum of voiced and unvoiced speech. This observation also a method for cepstrum pitch determination.

The largest peak location in samples gives T0 and thus pitch is computed as F0= Fs / T0. By computing plots, it can be further observe that, the cepstral approach does not have large peaks as in the autocorrelation case that may interfere with the estimation of pitch

Pitch estimation by SIFT method:

The SIFT method is yet another mostly used pitched estimation method. This is based on the linear prediction (LP) analysis of speech. The SIFT in turn employs the autocorrelation method for the estimation of pitch. However, the main discussion is, it performs autocorrelation of the LP residual than speech directly. For the optimal LP , more of the vocal tract information is modeled by the LP coefficients and hence the LP residual mostly contains the excitation source information. The autocorrelation of LP residual will therefore have unambiguous peaks representing the pitch period 'T0' information.

In case of LP analysis, the vocal tract information is modelled in terms of LP coefficient (LPC) as a process of the prediction of current sample of a combination of 'p' samples. The LPCs represents the coefficients of the LP filter . The LP filter is the representation of vocal tract. The corresponding inverse filter can be constructed using the inverse property. By passing the speech signal through the inverse filter results in the LP residual as the output. We plot the LP residual and the autocorrelation of the LP residual. As can be observed, there is a prominent peak at the pitch period 'T0' and there are no other peaks that interfere with its peak. This is the merit of SIFT over autocorrelation of speech.

From 30 msec segment of unvoiced speech, its LP residual and the autocorrelation of the LP residual. There is no prominent peak as in the case of voiced speech. Thus, a single peak picking can be employed for the estimation of pitch period 'T0' . Once 'T0' is known, then F0 = Fs/ T0 , where both Fs and T0 in samples.

Comparision of Pitch estimation methods:

All the methods show a peak at the pitch period 'T0'. Cepstral and the SIFT method sequences have less ambiguity compared to autocorrelation of speech. Hence either cepstrum or SIFT method is equally preferable for the estimation of pitch. In case of speech coding, the mostly used are in LP analysis and hence SIFT method is prefered over Cepstrum pitch estimation. Alternatively, in tasks like speaker recognition, since cepstral analysis is used for the feature extraction tasks, ceptrum pitch estimation may be employed.

Apart from the avarage pitch value, the other interse is in plotting time varying pitch values resulting in pitch contour information.] The random values in the pitch contours compared to the unvoiced and silence regions. The random values in the unvoiced and silence regions. The continuous segments correspond to the voiced regions. Most of the random values can be eliminated by the voiced/ unvoiced classification. In name of the voiced regions, the autocorrelation of speech has same random values due to the peak picking.

Procedure:

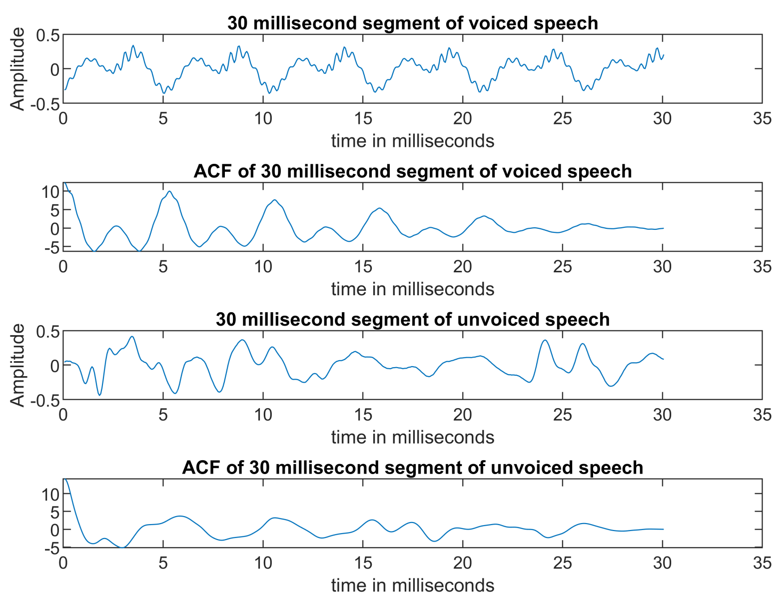
Record (16kHz, 16bit) the word “speech signal”; truncate long silence regions.

A. Pitch estimation by autocorrelation method:

a. Divide the given speech signal into 30-40ms blocks of speech frames. Find and plot the auto-correlation sequence of a voiced frame and an unvoiced frame.

```
clc;clear all;close all;
warning('off');
%%Load the .wav file with 'speech signal'
[y,fs]=audioread('l7_speech_signal.wav');

%normalising the signal amplitudes to be in -1 to 1
y=y./(1.01*abs(max(y)));

% 30ms long Voiced segment
y_v = y(ceil(0.217*fs) : floor(0.247*fs));

% 30ms long Voiced segment
y_uv = y(ceil(0.475*fs) : floor(0.505*fs));

% ACF of voiced segment
y_va=autocorr(y_v);

% ACF of unvoiced speech
y_uva=autocorr(y_uv);
```

```
t_v=(1/fs:1/fs:(length(y_v)/fs))*1000;
t_uv=(1/fs:1/fs:(length(y_uv)/fs))*1000;

t_va=(1/fs:1/fs:(length(y_va)/fs))*1000;
t_uva=(1/fs:1/fs:(length(y_uva)/fs))*1000;

figure;
subplot(411);
plot(t_v,y_v);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('30 millisecond segment of voiced speech');

subplot(412);
plot(t_va,y_va);
xlabel('time in milliseconds');
title('ACF of 30 millisecond segment of voiced speech');


subplot(413);
plot(t_uv,y_uv);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('30 millisecond segment of unvoiced speech');

subplot(414);
plot(t_uva,y_uva);
xlabel('time in milliseconds');
title('ACF of 30 millisecond segment of unvoiced speech');
```



b. Estimate the pitch frequency using this computed auto-correlation for the above voiced frame and an unvoiced frame. You may set a threshold for a significant peak, and assign zero to pitch frequency if there is no significant peak.

```
y_va=y_va./(abs(max(y_va))); %Normalisation

% Searching the peak
min_pitch=20; %minimum possible pitch period
max_pitch=30*16; %maximum possible pitch period
y_peaks=y_va(min_pitch:max_pitch); %searching the peaks in autocorrelation sequence
[y_val,y_loc]=max(y_peaks); %finding the location of the second largest peak in autocorrelatio
% computing the pitch period as the distance of the central peak to second
% largest peak in autocorrelation sequence
pitch_period=min_pitch+y_loc % samples
```

```
pitch_period = 86
```

```
pitch_freq=(1./pitch_period)*fs % Computing the pitch frequency
```

```
pitch_freq = 186.0465
```

```
y_uva=y_uva./(abs(max(y_uva))); %Normalisation

% Searching the peak
min_pitch=4; %minimum possible pitch period
max_pitch=30*16; %maximum possible pitch period
y_peaks=y_uva(min_pitch:max_pitch); %searching the peaks in autocorrelation sequence
[y_val,y_loc]=max(y_peaks); %finding the location of the second largest peak in autocorrelatio
% computing the pitch period as the distance of the central peak to second
% largest peak in autocorrelation sequence
pitch_period=min_pitch+y_loc % samples
```

```
pitch_period = 5
```

```
pitch_freq=(1./pitch_period)*fs % Computing the pitch frequency
```

```
pitch_freq = 3200
```

Observation:

Pitch Frequency(voiced): 186.0465 Hz

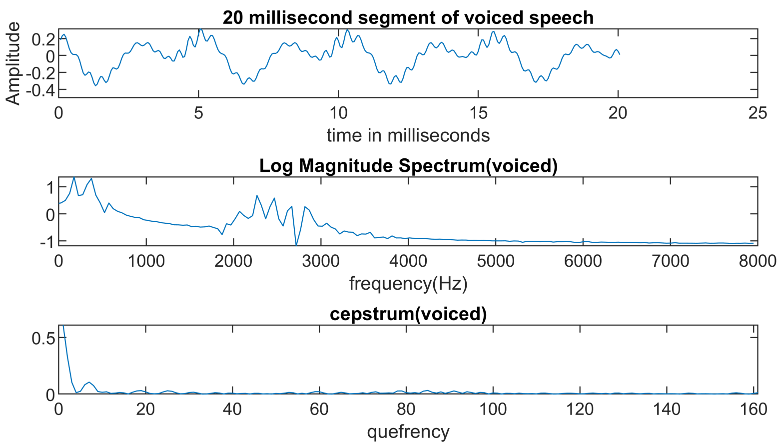
Pitch Frequency(unvoiced): 3200 Hz

B. Cepstrum based pitch estimation:

a. Divide the speech into short segments of 15-20ms frame size. Compute the cepstrum of the speech segment in the quefrency domain for each of these frames and plot for one voiced frame.

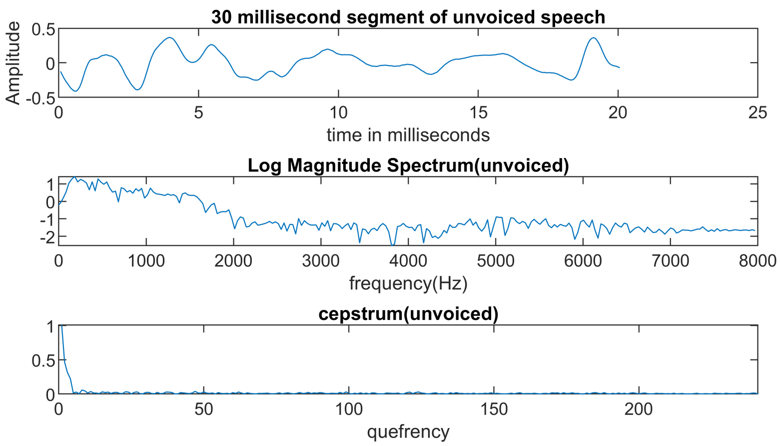
```
% Voiced speech segment
y_v1=y(ceil(0.226*fs) : floor(0.246*fs));
t_v1=(1/fs:1/fs:(length(y_v1)/fs))*1000;
Y_v = fftshift(fft(y_v1));
```

```
F_v = -fs/2 : fs/length(Y_v) : fs/2 - fs/length(Y_v);

figure;
subplot(411);
plot(t_v1,y_v1);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('20 millisecond segment of voiced speech');

%plotting Log Magnitude Spectrum of the speech signal
subplot(412);
plot(F_v, log10(abs(Y_v)));
xlim([0, fs/2]);
xlabel('frequency(Hz)');
title('Log Magnitude Spectrum(voiced)');

%plotting IDFT of the speech signal
c_v = ifft(log10(abs(Y_v))); % ifft
% PLOT
subplot(413);
plot(abs(c_v));
xlim([0,ceil(length(c_v)/2)]);
xlabel('quefrency');
title('cepstrum(voiced)');
```




```
% Unvoiced speech segment
y_mid=round(length(y_uv)/2);
y_uv1=y_uv(round(y_mid-0.01*fs): round(y_mid+0.01*fs));
t_uv1=(1/fs:1/fs:(length(y_uv1)/fs))*1000;
Y_uv = fftshift(fft(y_uv));
F_uv = -fs/2 : fs/length(Y_uv) : fs/2 - fs/length(Y_uv);

figure;
```

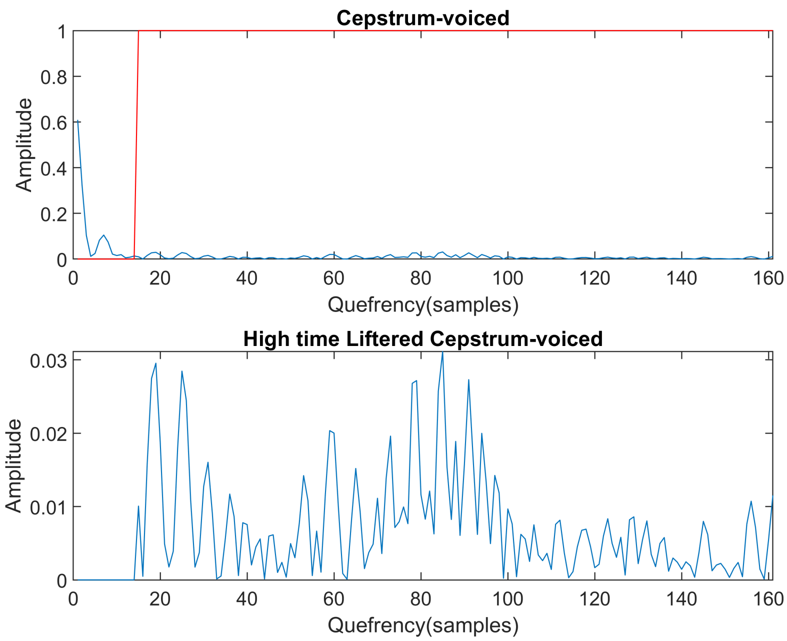
```
subplot(411);
plot(t_uv1,y_uv1);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('30 millisecond segment of unvoiced speech');

%plotting Log Magnitude Spectrum of the speech signal
subplot(412);
plot(F_uv, log10(abs(Y_uv)));
xlim([0, fs/2]);
xlabel('frequency(Hz)');
title('Log Magnitude Spectrum(unvoiced)');

%plotting IDFT of the speech signal
c_uv = ifft(log10(abs(Y_uv))); % ifft
% PLOT
subplot(413);
plot(abs(c_uv));
xlim([0,ceil(length(c_uv)/2)]);
xlabel('quefrency');
title('cepstrum(unvoiced)');
```



b. Estimate the pitch period by the high time liftering of the cepstrum of the voiced speech.

```
%High time liftering
c_s_1= c_v(1:ceil(length(c_v)/2)); % As the cepstrum is symmetric, half the cepstral coefficien
L=zeros(1,length(c_s_1)); %For defining liftering window
L(15:length(L))=1; %Liftering window
c_v_ht=c_s_1.*L; %High time liftered cepstrum

figure;
subplot(211);
plot(abs(c_s_1));
```

```
hold on
plot(L,'r');
xlim([0,length(c_s_1)]);
title('Cepstrum-voiced');
xlabel('Quefrency(samples)');
ylabel('Amplitude');
hold off
subplot(212);
plot(abs(c_v_ht));
xlim([0,length(c_s_1)]);
title('High time Liftered Cepstrum-voiced');
xlabel('Quefrency(samples)');
ylabel('Amplitude');
```




```
[y_val,y_loc]=max(c_v_ht); % Finding the peak in the high time liftered cepstrum
% Location of the peak gives pitch period in quefrency samples
pitch_period=y_loc
```

```
pitch_period = 85
```

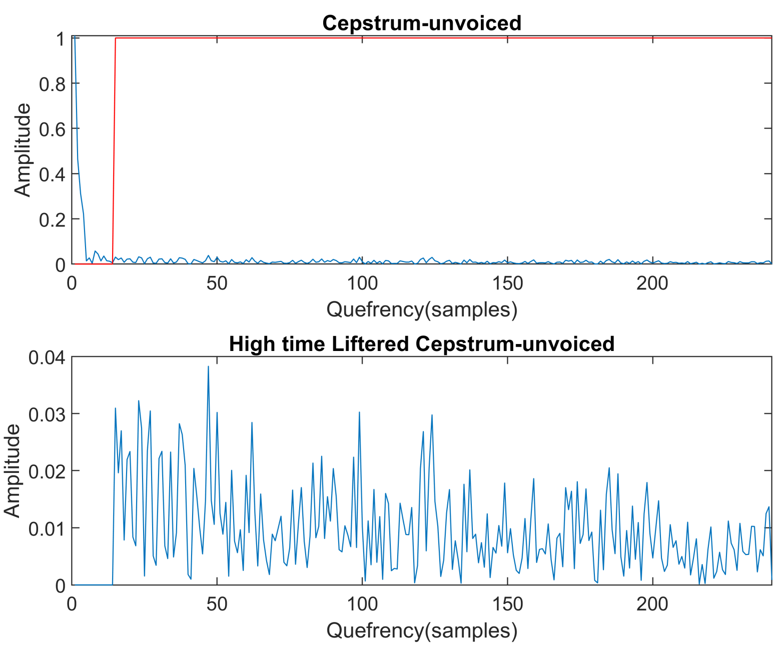
```
% converting pitch period in samples into frequency
pitch_frequency=(1/pitch_period)*fs
```

```
pitch_frequency = 188.2353
```

c. Estimate the pitch period by the high time liftering of the cepstrum of the unvoiced speech.

```
%High time liftering
c_uv_1= c_uv(1:ceil(length(c_uv)/2)); % As the cepstrum is symmetric, half the cepstral coeffic
L=zeros(1,length(c_uv_1)); %For defining liftering window
L(15:length(L))=1; %Liftering window
c_uv_ht=c_uv_1.*L; %High time liftered cepstrum
```

```
figure;
subplot(211);
plot(abs(c_uv_1));
hold on
plot(L,'r');
xlim([0,length(c_uv_1)]);
title('Cepstrum-unvoiced');
xlabel('Quefrency(samples)');
ylabel('Amplitude');
hold off
subplot(212);
plot(abs(c_uv_ht));
xlim([0,length(c_uv_1)]);
title('High time Liftered Cepstrum-unvoiced');
xlabel('Quefrency(samples)');
ylabel('Amplitude');
```




```
[y_val,y_loc]=max(c_uv_ht); % Finding the peak in the high time liftered cepstrum
% Location of the peak gives pitch period in quefrency samples
pitch_period=y_loc
```

```
pitch_period = 47
```

```
% converting pitch period in samples into frequency
pitch_frequency=(1/pitch_period)*fs
```

```
pitch_frequency = 340.4255
```

Observations:

Pitch period(voiced): 85 samples

Pitch period(unvoiced): 47 samples

C. Pitch estimation by Simple Inverse Filtering Technique (SIFT):

a. Take a 30 ms voiced speech segment and compute the Linear Prediction (LP) residual by LP analysis. Perform autocorrelation on the LP residual. Estimate of pitch period from the autocorrelation sequence of the LP residual.

```
win= dsp.Window('Hamming'); % Applying hamming window
y_hv = win(y_v);

%LPC coeficients using autocorrelation method
p=10;
[a,g] = lpc(y_hv,p);
LP_v_coeff=a;

%% LP residual signal
y_v_lp=conv(y_hv,LP_v_coeff); %Inverse filtering to get residual

%Removing additional p samples from the begin and end portion of the
%residual due to convolution
y_v_lp=y_v_lp(round(p/2):length(y_v_lp)-round(p/2)-1);

%Autocorrelation of LP residual
y_v_lp_a=autocorr(y_v_lp);
y_v_lp_a=y_v_lp_a./(abs(max(y_v_lp_a))); %Normalisation

% Searching the peak
min_pitch=20; %minimum possible pitch period
max_pitch=30*16; %maximum possible pitch period
y_peaks=y_v_lp_a(min_pitch:max_pitch); %searching the peaks in autocorrelation sequence
[y_val,y_loc]=max(y_peaks); %finding the location of the second largest peak in autocorrelatio
% computing the pitch period as the distance of the central peak to second
% largest peak in autocorrelation sequence
pitch_period=min_pitch+y_loc
```

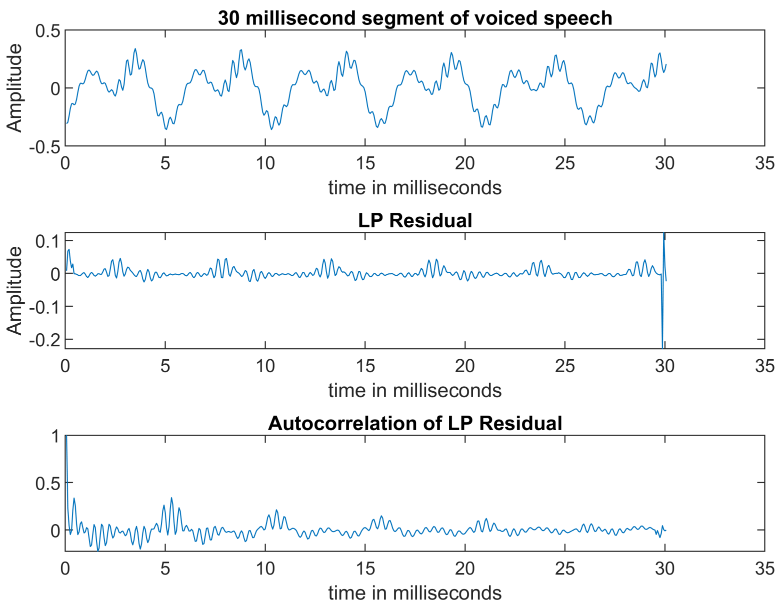
```
pitch_period = 86
```

```
pitch_freq=(1./pitch_period)*fs % Computing the pitch frequency
```

```
pitch_freq = 186.0465
```

```
t_v=(1/fs:1/fs:(length(y_hv)/fs))*1000;
t_v_lp=(1/fs:1/fs:(length(y_v_lp)/fs))*1000;
t_v_lp_a=(1/fs:1/fs:(length(y_v_lp_a)/fs))*1000;

figure;
subplot(311);
plot(t_v,y_hv);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('30 millisecond segment of voiced speech');
```

```
subplot(312);
plot(t_v_lp,y_v_lp);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('LP Residual');

subplot(313);
plot(t_v_lp_a,y_v_lp_a);
xlabel('time in milliseconds');
title('Autocorrelation of LP Residual');
```



b. Perform the same for the unvoiced speech segment also.

```
win= dsp.Window('Hamming'); % Applying hamming window
y_huv = win(y_uv);

%LPC coeficients using autocorrelation method
p=10;
[a,g] = lpc(y_huv,p);
LP_uv_coeff=a;

%% LP residual signal
y_uv_lp=conv(y_huv,LP_uv_coeff); %Inverse filtering to get residual

%Removing additional p samples from the begin and end portion of the
%residual due to convolution
y_uv_lp=y_uv_lp(round(p/2):length(y_uv_lp)-round(p/2)-1);

%Autocorrelation of LP residual
y_uv_lp_a=autocorr(y_uv_lp);
y_uv_lp_a=y_uv_lp_a./(abs(max(y_uv_lp_a))); %Normalisation
```

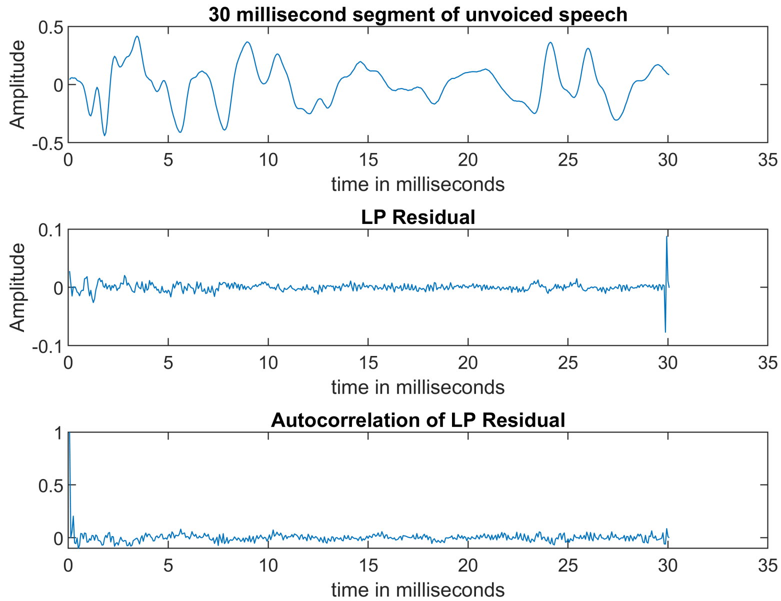
```
% Searching the peak
min_pitch=4; %minimum possible pitch period
max_pitch=30*16; %maximum possible pitch period
y_peaks=y_uv_lp_a(min_pitch:max_pitch); %searching the peaks in autocorrelation sequence
[y_val,y_loc]=max(y_peaks); %finding the location of the second largest peak in autocorrelation
% computing the pitch period as the distance of the central peak to second
% largest peak in autocorrelation sequence
pitch_period=min_pitch+y_loc
```

```
pitch_period = 5
```

```
pitch_freq=(1./pitch_period)*fs % Computing the pitch frequency
```

```
pitch_freq = 3200
```

```

t_uv=(1/fs:1/fs:(length(y_huv)/fs))*1000;
t_uv_lp=(1/fs:1/fs:(length(y_uv_lp)/fs))*1000;
t_uv_lp_a=(1/fs:1/fs:(length(y_uv_lp_a)/fs))*1000;

figure;
subplot(311);
plot(t_uv,y_huv);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('30 millisecond segment of unvoiced speech');

subplot(312);
plot(t_uv_lp,y_uv_lp);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('LP Residual');

subplot(313);
plot(t_uv_lp_a,y_uv_lp_a);
xlabel('time in milliseconds');
title('Autocorrelation of LP Residual');
```



Observations:

Pitch period(voiced): 86

Pitch period(Unvoiced): 5

D. Comparison of pitch estimation methods:

a. Plot the entire input speech signal and it's pitch contours estimated using autocorrelation, cepstrum and SIFT based pitch estimation methods.

```
% Autocorrelation

Frame_size = 30; %Input: Frame-size in millisecond
Frame_shift = 10; %Input: Frame-shift in millisecond

t=(1/fs:1/fs:(length(y)/fs))*1000;

window_period=Frame_size/1000;
shift_period=Frame_shift/1000;
pitch_freq=0;

figure;
subplot(411);
plot(t,y);
xlim([0, 1000]);
xlabel('time in milliseconds');
ylabel('Amplitude');
title('Speech Signal waveform');

window_length = window_period*fs;
sample_shift = shift_period*fs;
sum_a=0; energy_a=0; autocorrelation=0;
```

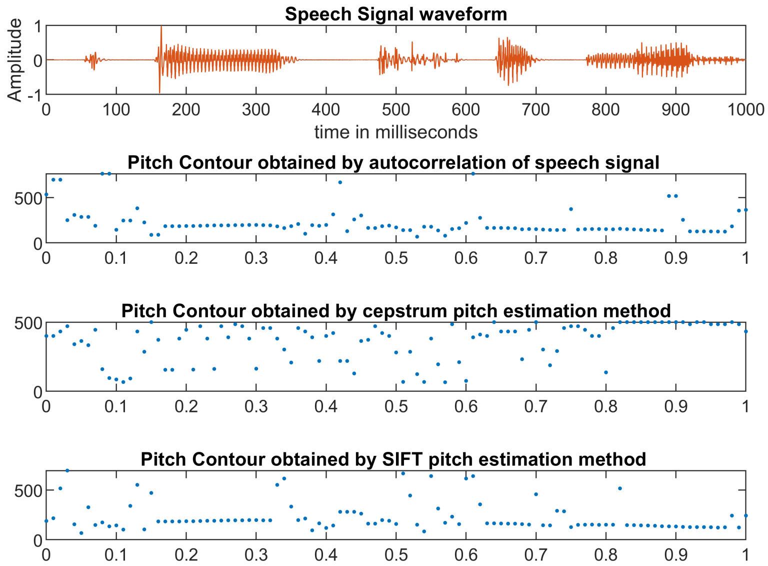
```
autocor=[];
for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
  k=1;yy=0;
  for j=(((i-1)*sample_shift)+1):(((i-1)*sample_shift)+window_length)
    yy(k)=y(j);
    k=k+1;
  end
  for l=0:(length(yy)-1)
    sum_a=0;
    for u=1:(length(yy)-l)
      s=yy(u)*yy(u+l);
      sum_a=sum_a+s;
    end
     autocor(l+1)=sum_a;
     autocorrelation(l+1,i)= autocor(l+1);
  end
  auto=autocor(21:240);
  max_a=0;
  for uu=1:220
    if(auto(uu)>max_a)
      max_a=auto(uu);
      sample_no=uu;
    end
  end
  pitch_freq(i)=1/((20+sample_no)*(1/fs));
end
[rows,cols]=size(autocorrelation);
kkk=1/fs:shift_period:(cols*shift_period);
subplot(412);
plot(kkk,pitch_freq,'.');
xlim([0, 1]);
title('Pitch Contour obtained by autocorrelation of speech signal ');
```

```
%Cepstral Analysis
o=1;
Frame_size=30; pitch_freq = 0;
Frame_shift=10;

window_period=Frame_size/1000;
shift_period=Frame_shift/1000;

window_length = window_period*fs;
sample_shift = shift_period*fs;
sum_c=0;energy_c=0;
for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
  k=1;yy=0;
  for j=(((i-1)*sample_shift)+1):(((i-1)*sample_shift)+window_length)
    yy(k)=y(j);
    k=k+1;
  end

     t=1/fs:1/fs:(length(yy)/fs);
```

```
    t=(t(1:length(t)/2))*1000;
    Y = abs(fftshift(fft(yy)));
for i=1:length(Y)
  if (Y(i)==0)
    Y(i)=1D-16;
  end
end
Y_log=log10(Y);

c_s=abs(ifft(Y_log));
c_s_1= c_s(1:ceil(length(c_s)/2)); % As the cepstrum is symmetric, half the cepstral coefficien
L=zeros(1,length(c_s_1)); %For defining liftering window
L(16:length(L))=1; %Liftering window
c_s_ht=c_s_1.*L; %High time liftered cepstrum

max1=max(c_s_ht);
for uu=1:length(c_s_ht)
  c_s_ht(uu);
    if(c_s_ht(uu)==max1)
      max1=c_s_ht(uu);
      sample_no=uu;
    end
  end
  pitch_freq1=1/((16+sample_no)*(1/fs));
  pitch_freq(o)= pitch_freq1;
  o=o+1;
end
kk=1/fs:shift_period:(length(pitch_freq)*shift_period);
subplot(413);
plot(kk,pitch_freq,'.');
xlim([0, 1]);
title('Pitch Contour obtained by cepstrum pitch estimation method');

% [y_val,y_loc]=max(c_s_ht); % Finding the peak in the high time liftered cepstrum
% % Location of the peak gives pitch period in quefrency samples
% pitch_period=y_loc;
%
% % converting pitch period in samples into frequency
% pitch_frequency=(1/pitch_period)*fs;
%   pitch_freq(o)= pitch_frequency;
%   o=o+1;
% end
% kk=1/fs:shift_period:(length(pitch_freq)*shift_period);
% subplot(413);plot(kk,pitch_freq,'.');
% xlim([0, 1]);
% title('Pitch Contour obtained by cepstrum pitch estimation method');
```

```
% LP Analysis

Frame_size = 30;  %Input: Frame-size in millisecond
Frame_shift = 10; %Input: Frame-shift in millisecond
```

```
window_period=Frame_size/1000;
shift_period=Frame_shift/1000;
pitch_freq=0;

window_length = window_period*fs;
sample_shift = shift_period*fs;
sum_lp=0;energy=0;autocorrelation=0;
for i=1:(floor((length(y))/sample_shift)-ceil(window_length/sample_shift))
  k=1;yy=0;
  for j=(((i-1)*sample_shift)+1):(((i-1)*sample_shift)+window_length)
    yy(k)=y(j);
    k=k+1;
  end

  yy=yy/(1.01*abs(max(yy)));
t=(1/fs:1/fs:(length(yy)/fs))*1000;
P=10;

[a,g] = lpc(yy,P); % LP coefficients computed
LPCoeffs=a;
y5=conv(yy,LPCoeffs);
y5=y5(round(P/2):length(y5)-round(P/2)-1);

  for l=0:(length(y5)-1)
    sum_a=0;
    for u=1:(length(y5)-l)
      s=y5(u)*y5(u+l);
      sum_a=sum_a+s;
    end
    autocor(l+1)=sum_a;
    autocorrelation(l+1,i)= autocor(l+1);
  end
  auto=autocor(21:240);
  max_a=0;
  for uu=1:220
    if(auto(uu)>max_a)
      max_a=auto(uu);
      sample_no=uu;
    end
  end
  pitch_freq(i)=1/((20+sample_no)*(1/fs));
end
[rows,cols]=size( autocorrelation);
kkk=1/fs:shift_period:(cols*shift_period);
subplot(414);
plot(kkk,pitch_freq,'.');
xlim([0, 1]);
title('Pitch Contour obtained by SIFT pitch estimation method ');
```



Observations:

We observe smooth contour for Voiced speech segment and non- smooth contour(random values) for Unvoiced speech segment.

From the plots, it is observed that SIFT method is preferrable as they have less ambiguity compared to other methods.

```
function [a]= autocorr(y)
sum=0;
for i=0:(length(y)-1)
    sum=0;
    for u=1:(length(y)-i)
        s=y(u)*y(u+i);
        sum=sum+s;
    end
    autocorr(i+1)=sum;
end
a=autocorr;
return;
end
```