EE-414 Speech Processing Lab

Lab-9 - 180020002

Aim:

- To compute LP coefficients and LP residual of a given speech signal.
- To compute the formant parameters by LP analysis.
- To compute the excitation parameters like pitch by LP analysis.
- To compute the normalized error curves for voiced and unvoiced segments of speech.

Theory:

The cepstral analysis does the deconvolution of speech into source and system components by traversing through frequency domain, the deconvolution task becomes computational intensive process. To reduce such type of computational complexity and finding the source and system components from time domain itself, the *Linear Prediction* analysis is developed.

LP Analysis

The redundancy in the speech signal is exploited in the LP analysis. The prediction of current sample as a linear combination of past *p* samples form the basis of linear prediction analysis where *p* is the order of prediction. The predicted sample s^(n) can be represented as follows:

$$\hat{s}(n) = -\sum_{k=1}^{p} a_k.s(n-k)$$

where aks are the linear prediction coefficients and s(n) is the windowed speech sequence obtained by multiplying short time speech frame with a hamming or similar type of window which is given by,

$$s(n) = x(n).w(n)$$

where ω(n) is the windowing sequence. The prediction error e(n) can be computed by the difference between actual sample s(n)and the predicted sample s^(n) which is given by,

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^{p} a_k.s(n-k)$$

$$e(n) = s(n) - \hat{s}(n)$$

$$e(n) = s(n) + \sum_{k=1}^{p} a_k.s(n-k)$$

The primary objective of LP analysis is to compute the LP coefficients which minimized the prediction error e(n). The popular method for computing the LP coefficients by least squares auto correlation method. This achieved by minimizing the total prediction error. The total prediction error can be represented as follows,

$$E = \sum_{n=-\infty}^{\infty} e^2(n)$$

This can be expanded using the equation (5) as follows,

$$E = \sum_{n=-\infty}^{\infty} [s(n) + \sum_{k=1}^{p} a_k.s(n-k)]^2$$

The values of a_ks which minimize the total prediction error E can be computed by finding

$$\frac{\partial E}{\partial a_k}$$

and equating to zero for k=0,1,2,...p.

$$\frac{\partial E}{\partial a_k} = 0$$

for each a_k give p linear equations with p unknowns. The solution of which gives the LP coefficients. This can be represented as follows,

$$\frac{\partial E}{\partial a_k} = \frac{\partial}{\partial a_k}. \sum_{n=-\infty}^{\infty} [s(n) + \sum_{k=1}^{p} a_k.s(n-k)]^2 = 0$$

The differentiated expression can be written as,

$$\sum_{n=-\infty}^{\infty} s(n-i).s(n) = \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s(n-i).s(n-k),$$

where i=1, 2, 3...p. The equation can be written in terms of autocorrelation sequence R(i) as follows,

$$\sum_{k=1}^{p} a_k R(i-k) = R(i)$$

$$R(i) = \sum_{n=i}^{N-1} s(n)s(n-i)$$

for i= 1,2,3...p and N is the length of the sequence.

This can be represented in the matrix form as follows,

$$R.\,A = -r$$

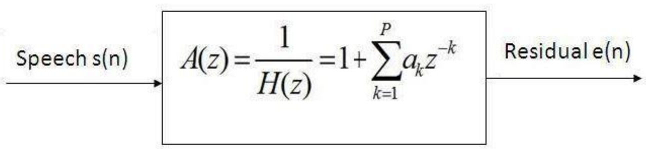
where R is the pXp symmetric matrix of elements R(i, k) = R(|i-k|), (1<=i, k<=p), r is a column vector with elements (R(1),R(2), ...R(P)) and finally A is the column vector of LPC coefficients (a(1), a(2), ...a(p)). It can be shown that R is *toeplitz* matrix which can be represented as,

$$R = \begin{bmatrix} R(1), R(2), R(3), ..., R(P) \\ R(2), R(1), R(2), ..., R(P-1) \\ R(3), R(2), R(1), ..., R(P-2) \\ . \\ . \\ R(P), R(P-1), R(P-2), ..., R(1) \end{bmatrix}$$

The LP coefficients can be computed as shown,

$A = -R^{-1}.\,r$

where R^{-1} is the inverse of the matrix R



Computing LP Residual

LP residual is the prediction error e(n) obtained as the difference between the predicted speech sample s^(n) and the current sample s(n).

$$e(n) = s(n) - \hat{s}(n)$$

$$e(n) = s(n) + \sum_{k=1}^{p} a_k.s(n-k)$$

In the frequency domain, the equation can be represented as,

$$E(z) = S(z) + \sum_{k=1}^{p} a_k.S(z)z^{-k}$$

$$A(z) = \frac{E(z)}{S(z)} = 1 + \sum_{k=1}^{p} a_k.z^{-k}$$

It can be shown that the LP spectrum H(z) as,

$$H(z) = \frac{1}{1 + \sum_{k=1}^{P} a_k z^{-k}} = \frac{1}{A(z)}$$

As A(z) is the reciprocal of H(z), LP residual is obtained by the inverse filtering of speech.

Estimating vocaltract parameters by LP analysis

LP analysis separates the given short term sequence of speech into its slowly varying vocal tract component represented by LP filter (H(z)) and fast varying excitation component given by the LP residual (e(n)). The LP filter (H(z)) induces the desired spectral shape for the shape on the flat spectrum (E(z)) of the noise like excitation sequence. As the LP spectrum provides the vocaltract characteristics, the vocaltract resonances (formants) can be estimated from the LP spectrum. Various formant locations can be obtained by picking the peaks from the magnitude LP spectrum (|H(z)|).

$$S(z) = E(z).H(z)$$

where S(z) is the spectrum of the given short time speech signal.

Estimating pitch from LP residual

As the LP residual is an error signal obtained by the LP analysis, it is noisy in nature. As the pitch marks are characterized by the sharp and periodic discontinuity, it cause a large error in the computed LP residual. So the periodicity of the error gives the pitch period of that segment of speech and this can be computed by the autocorrelation method. It has to be noted that for unvoiced speech signal the residual will be like random noise without any periodicity.

Normalized Error

The normalized error can be used to find the optimal prediction order required for the LP analysis of the given speech segment. The normalized error can be defined as the ratio of the total minimum error to the total energy of the signal. If Ep is the total minimum error obtained for a prediction order p and R(1) is the total energy of the signal, then the normalized error Vp can be represented as,

$$V_p = \frac{E_p}{R(1)}$$

The expression for total minimum error for the given prediction order p can be given by,

$$E_p = \sum_{n=-\infty}^{\infty} s^2(n) + \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s(n).s(n-k)$$

This can be written in terms of autocorrelation sequence,

$$E_p = R(1) + \sum_{k=1}^{p} a_k.R(k)$$

Procedure:

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

A. Estimating Linear Prediction (LP) coefficients from the speech.

a. Select a frame (25 ms long) at the center of a voiced segment. Estimate the LPCs of the segment using the autocorrelation method.

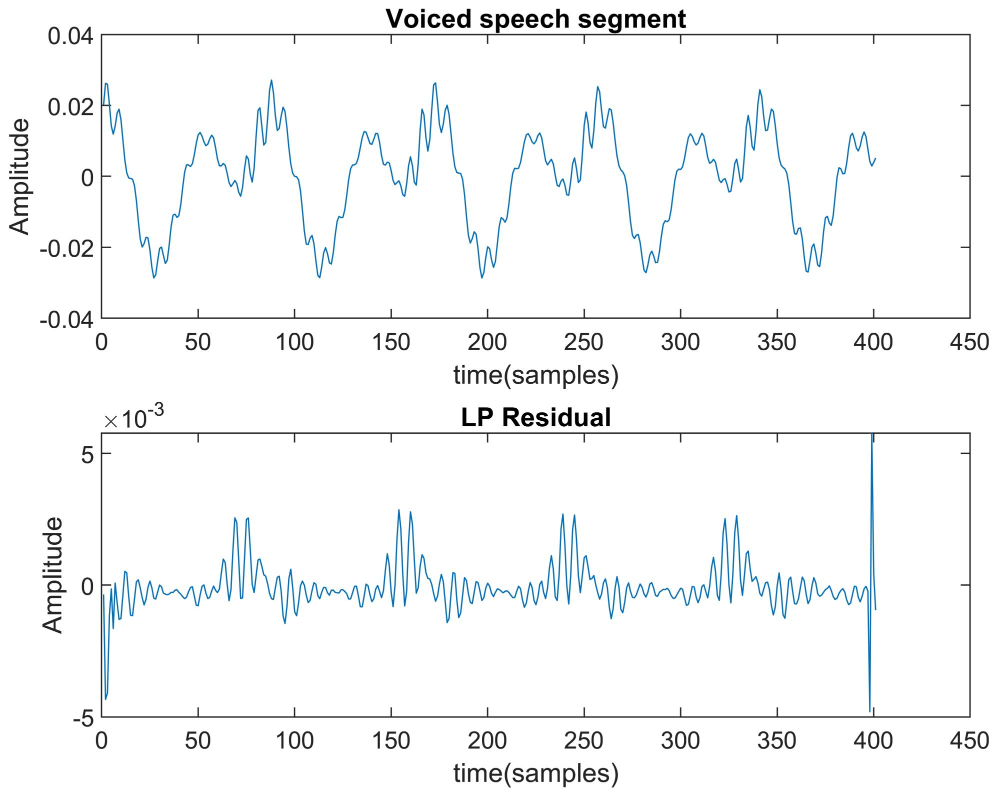
```
clc;clear all;close all;
warning('off');
%%Load the .wav file with 'speech signal'
[y,fs]=audioread('l7_speech_signal.wav');

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

%/ee/- 25ms long Voiced segment
y_v = y(ceil(0.215*fs) : floor(0.240*fs));

%Hamming Window
t = 0 : 1 / fs : (length(y_v) - 1) / fs;
w = 0.54 - 0.46*cos(2*pi*t/(length(y_v)-1));
y_hv = y_v.*w;

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

```
LP_v_coeff = 1×11
    1.0000   -1.6709    0.9365    0.0915   -0.3582   -0.2283    0.0893    0.1546 ⋯
```

B. Computing LP residual

a. Using the computed LPCs, derive the LP residual signal.

```
%% LP residual signal
y_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_lp=y_lp(round(p/2):length(y_lp)-round(p/2)-1);
figure;
subplot(211);
plot(y_hv);
xlabel('time(samples)');
ylabel('Amplitude');
title('Voiced speech segment');
subplot(212);
plot(y_lp);
xlabel('time(samples)')
ylabel('Amplitude')
title('LP Residual');
```



C. Pitch estimation from LP residual:

a. Estimate the pitch from the estimated LP residual using autocorrelation.

```
y_lp_corr=autocorr(y_lp,160);
y_lp_corr=y_lp_corr./(abs(max(y_lp_corr))); %Normalisation

% Searching the peak
min_pitch=20; %minimum possible pitch period
max_pitch=160; %maximum possible pitch period
y_peaks=y_lp_corr(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 autocorrelatior
% computing the pitch period as the distance of the central peak to second
% largest peak in autocorrelation sequence
pitch_period=min_pitch+y_loc
```

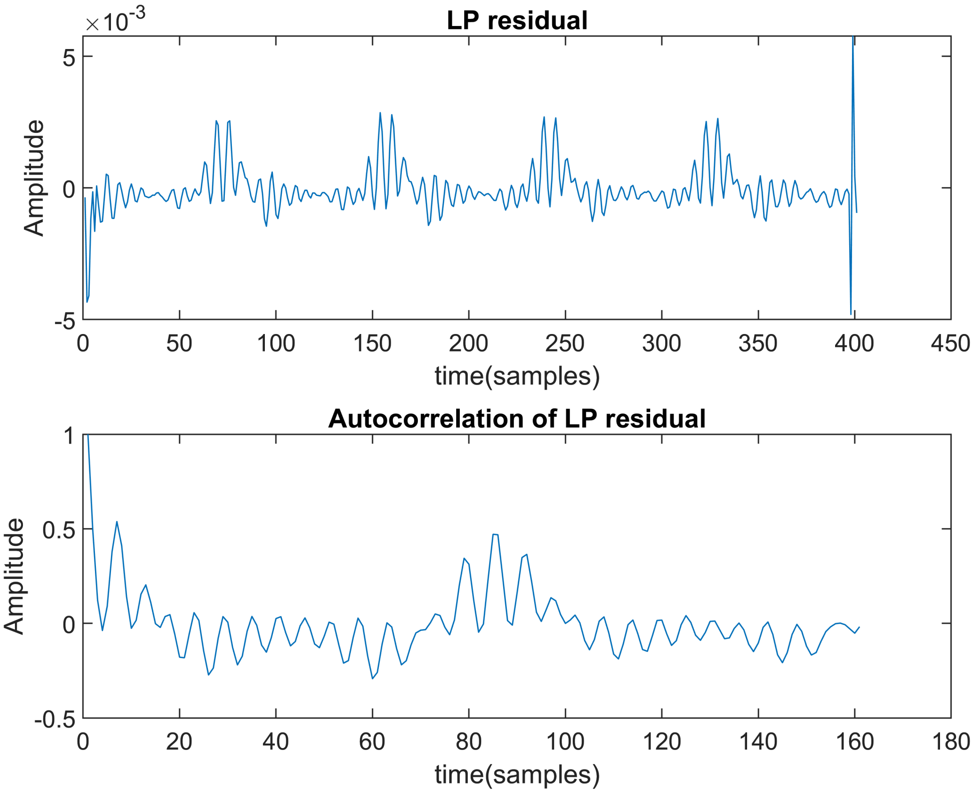
```
pitch_period = 86
```

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

```
pitch_freq = 186.0465
```

```
figure;
subplot(211);
```

```
plot(y_lp);
xlabel('time(samples)');
ylabel('Amplitude');
title('LP residual');
subplot(212);
plot(y_lp_corr);
xlabel('time(samples)')
ylabel('Amplitude')
title('Autocorrelation of LP residual');
```



D. Formant estimation from LP spectrum:

a. Explain, step by step, the procedure of computing the LP spectrum from LPCs.

b. Demonstrate the same on the voiced frame selected above.

Procedure:

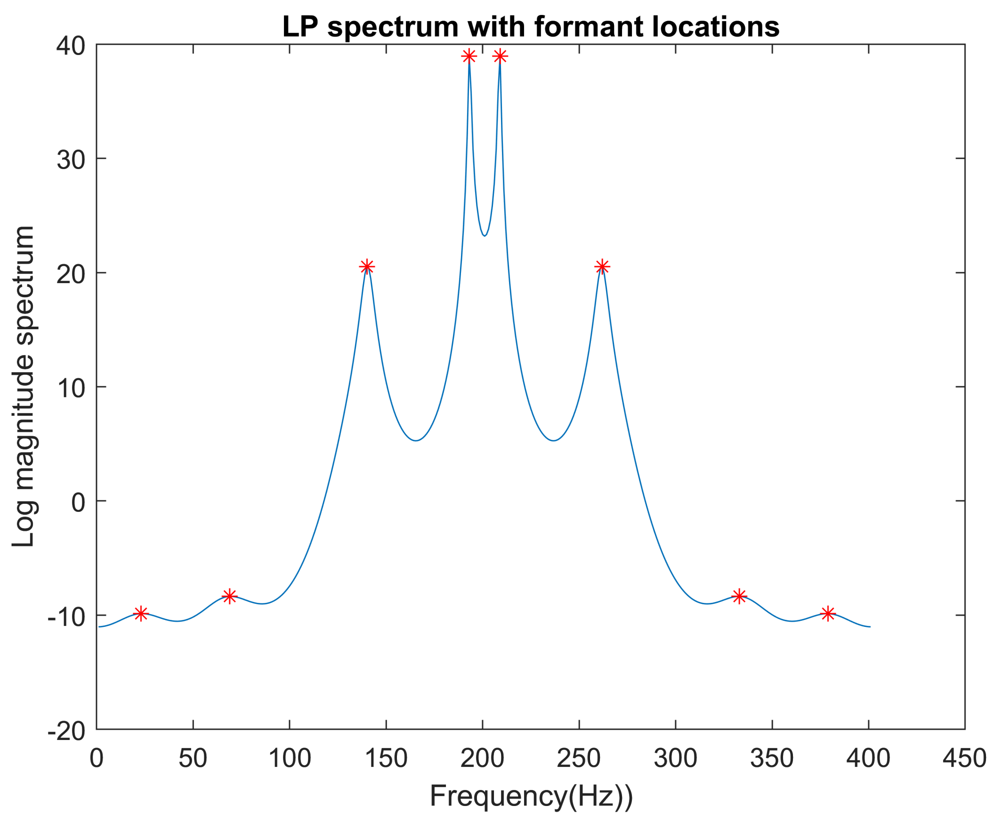
1. We first calculate the LP coefficients using the autocorrelation method.

2. Then we obtain the frequency response of the filter using the LPCs.

3. We can plot this filter H to get the LP spectrum

```
%Computing LP spectrum by taking FFT of LP coefficients
y_az=abs(fftshift(fft(LP_v_coeff,length(y_hv)))); % representing inverse spectrum, A(z)
y_az=y_az.^(-1); %representing LP spectrum as H(z)=1/A(z);
y_az=20*log10(y_az); %converting to db
```

```
%peak picking algorithm
k=1;
for i=2:length(y_az)-1
    if (y_az(i-1)<y_az(i)) && (y_az(i+1)<y_az(i))
        formant_mag_v(k)=y_az(i); %formant magnitude
        formant_v(k)=i; %formant location
        k=k+1;
    else
        continue;
    end
end

figure;
plot(y_az);
hold on;
plot(formant_v,formant_mag_v,'r*');
hold off;
title('LP spectrum with formant locations');
xlabel('Frequency(Hz))');
ylabel('Log magnitude spectrum');
```

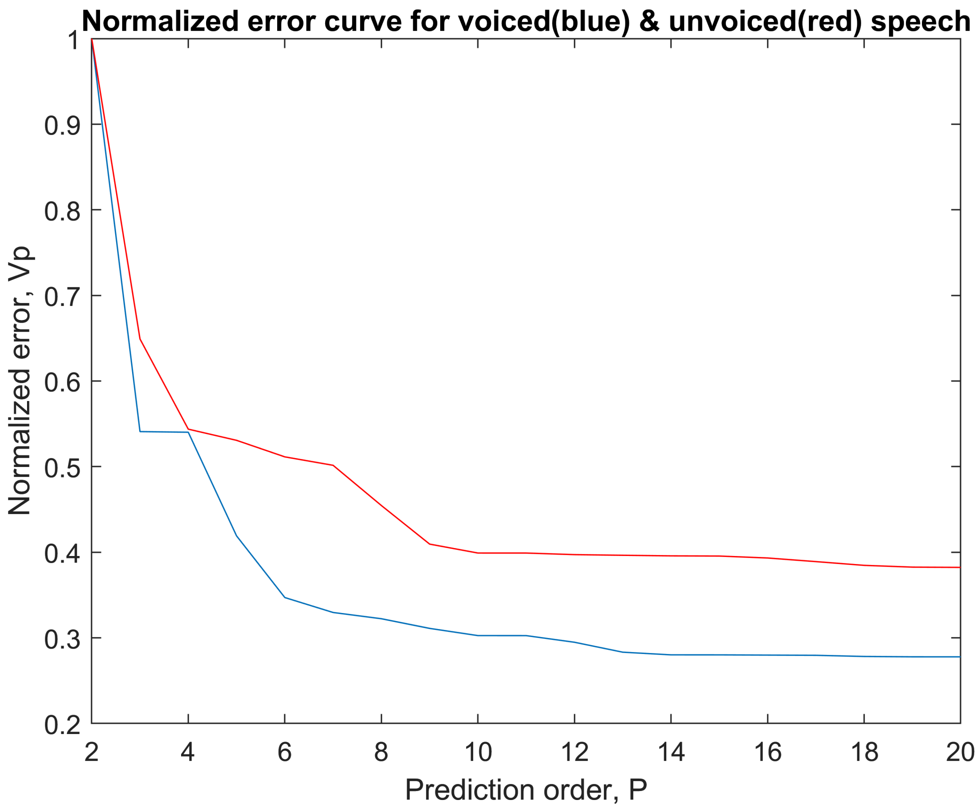


E. Normalized Error

a. Select the 25ms frame at the center of the voiced and unvoiced frame respectively. Compute the normalized LP residual error as a function of the order of LP prediction. Plot normalized error curve against the prediction order for both voiced and unvoiced frames

b. Comment upon the choice of optimal prediction order for the segments.

```
t_v = 0 : 1 / fs : (length(y_v) - 1) / fs;
w_v = 0.54 - 0.46*cos(2*pi*t/(length(y_v)-1));
y_hv = y_v.*w_v;

% /s/- 25ms long unvoiced segment
y_uv= y(ceil(0.054*fs) : floor(0.079*fs));
t_uv = 0 : 1 / fs : (length(y_uv) - 1) / fs;
w_uv = 0.54 - 0.46*cos(2*pi*t/(length(y_uv)-1));
y_huv = y_uv.*w_uv;

y_v_corr= autocorr(y_hv); %autocorrelation of voiced speech
y_uv_corr=autocorr(y_huv); %autocorrelation of unvoiced speech

y_v_corr=y_v_corr./(abs(max(y_v_corr))); %normalisation
y_uv_corr=y_uv_corr./(abs(max(y_uv_corr))); %normalisation
count_v=1; count_uv=1; %counters
for p=2:20
    %computation of normalised error curve for voiced speech
    [a,g] = lpc(y_hv,p); % LP coefficients computed
    LP_v_coeff=a;
    Ep_v(count_v)=y_v_corr(1)+sum(LP_v_coeff(2:end).*y_v_corr(2:p+1)); %computing the normalise
    count_v=count_v+1; %increment the counter
    clear A_v r_v A_v2 L_v LP_v_coeff ; %clear the variables used in the loop

    %computation of normalised error curve for unvoiced speech
    [a,g] = lpc(y_huv,p); % Lp coefficients computed
    LP_uv_coeff=a;
    Ep_uv(count_uv)=y_uv_corr(1)+sum(LP_uv_coeff(2:end).*y_uv_corr(2:p+1)); %computing the norm
    count_uv=count_uv+1; %increment the counter
    clear A_uv r_uv A_uv2 L_uv LP_uv_coeff ; %clear the variables used in the loop
end

figure
plot([2:20],Ep_v/max(Ep_v));
hold on;
plot([2:20],Ep_uv/max(Ep_uv),'r');
hold off;
title('Normalized error curve for voiced(blue) & unvoiced(red) speech');
xlabel('Prediction order, P');
ylabel('Normalized error, Vp');
```




```
[v_val,v_loc]=min(Ep_v/max(Ep_v));
[uv_val,uv_loc]=min(Ep_uv/max(Ep_uv));
v_loc
```

```
v_loc = 19
```

```
uv_loc
```

```
uv_loc = 19
```

Observations:

The prediction order after which the normalized error is not changing can be selected as the optimal prediction order for that segment of speech.

For voiced segment its 19.

For unvoiced segment its 19.