Homework 8

Speech Processing

Name: Jinhan Zhang

Class: ECE 251A   Digital Signal Processing I

Date: 03/14/2017

Speech Processing

• Objective

Use Matlab to process speech signals, use autocorrelation method of linear prediction to do high resolution spectral analysis.

• Background

For conventional averaging power spectral estimation, we will always have the fundamental trade-off between variance reduction and horizontal resolution. We want to beat this by introducing pre-knowledge in the estimation process. In high resolution spectral analysis, we end up with proposing a model that will capture how the data is generated. For speech signals, they are usually stationary in the time span of about 25ms, and speech signals are often sampled at the frequency of 8-10kHz. So there are only about 250 samples available for us to do spectral analysis, thus it would be unlikely to get a great estimation by conventional averaging method. So in this assignment, we will use autocorrelation method of linear prediction to do high resolution power spectral analysis on speech signals and analyze the formant frequencies for vowel sound.

• Approach

Firstly, we need to import the speech signals into Matlab. In this assignment, we mainly analyze 3 kinds of phonemes: /i/ in “eve”, /u/ in “boot” and /η/ in “rung”. Each speech data file consists of ASCII text with one integer sample value per line and a total of 15360 points, and the signal is digitized at the sample rate of 10kHz. We can use ‘load’ function to import these signals.

Secondly, we plot a 256-point segment (n=1024, …, 1279) of three signals respectively, and do spectral estimation using periodogram. And then we also do that using the first 64 points of our 256-point segment.

Thirdly, we use the 256-point data block, window them and make estimation of inverse filter for different numbers of orders to see their performance and plot an error vs. the number of orders figure to see how error changes with the increase of p. And also, we plot the estimate for case p=14. Finally, we repeat the process using the first 64 points of the 256-point segment.

All the process above involving window function is implemented by Hanning window in this assignment.

• Results

Fig 1 is the 256-point time series from phoneme /i/, we can clearly see its periodicity. Fig 2 is the corresponding spectrum estimation, we can find the first formant at 234Hz clearly from this plot, but other formants are hard to recognize. I've also normalized the spectrum by $(f_s\mathrm{MU})^{-1}$, and this also applies for periodogram estimations below. Fig 3 is the corresponding single periodogram estimation using the first 64-point signal, we can see that the curve looks smoother than that of 256-point estimate, but also it's hard to find the second and third formant using this plot. Fig 4 is the 256-point time series from phoneme /u/, we can clearly see its periodicity. Fig 5 is the corresponding spectrum estimation, we can find the first formant at 312Hz from this plot, while it's hard to recognize other formants. Fig 6 is the corresponding single periodogram estimation using the first 64-point signal, we can see that the curve looks smoother than that of 256-point estimate, but also we are unable to find the second and third formant using this plot. Fig 7-9 repeat all the same process to /ŋ/ as what we did for /i/ and /u/.

Fig. 10 shows how the prediction error power changes with the increase of order p in autocorrelation method of linear prediction using 256-point segment of /i/. We can see that the error power drops greatly when p is small, the rate of decline gradually decreases and when p is bigger than 8 the error power almost stays at the same level. Fig. 11 shows the plot of $10\log(\frac{1}{|\hat{A}(k)|^2})$ where $\hat{A}(k)$ is the 256-point FFT of $\{1,\hat{a}_1,\ldots,\ \hat{a}_p\}$ with trailing zeros when $p = 14$. We can see the first formant at 234Hz, second formant at 2148Hz and third one at 3125Hz, which is quite reasonable according to the values reported in literature. Fig. 12 shows how the prediction error power changes with the increase of order p in autocorrelation method of linear prediction using 256-point of /u/. We can see that the error power drops greatly between certain intervals and when p is bigger than 12 the error power almost stays at the same level. Fig. 13 shows the plot of $10\log(\frac{1}{|\hat{A}(k)|^2})$ where $\hat{A}(k)$ is the 256-point FFT of $\{1,\hat{a}_1,\ldots,\ \hat{a}_p\}$ with trailing zeros when $p = 14$. We can see the first formant at 313Hz, second formant at 859Hz and third one at 3203Hz. The first and second formant fit the result in literature well, while the third one does not seem that close. And this may be caused by the difference of everyone's speech. Fig. 14 and 15 show the similar process of using 256-point of /η/, which have similar trend. Estimates of the formant frequencies for the vowel sounds using 256-point segment and the value in literature are shown in Table 1.

Table 1. Comparison of formant frequencies for **vowel** with that in literature (256 points)

Vowel	F_1		F_2		F_3	
	Literature	Empirical	Literature	Empirical	Literature	Empirical
/i/	270Hz	234Hz	2290Hz	2148Hz	3010Hz	3125Hz
/u/	300Hz	313Hz	870Hz	859Hz	2240Hz	3203Hz

Fig. 16 shows how the prediction error power changes with the increase of order p in autocorrelation method of linear prediction using 64-point segment of $/\mathrm{i}/$. We can see that the error power drops greatly between certain intervals. Fig. 17 shows the plot of $10\log(\frac{1}{|\hat{A}(k)|^2})$ where $\hat{A}(k)$ is the 256-point FFT of $\{1,\hat{a}_1,\ldots, \hat{a}_p\}$ with trailing zeros when $p = 14$. We can see the first formant at 313Hz, second formant at 2305Hz and third one at 3203Hz, which is still reasonable according to the values reported in literature. Fig. 18 shows how the prediction error power changes with the increase of order p in autocorrelation method of linear prediction using 64-point of $/\mathrm{u}/$. We can see that the error power drops greatly between certain intervals and when p is bigger than 10 the error power almost stays at the same level. Fig. 19 shows the plot of $10\log(\frac{1}{|\hat{A}(k)|^2})$ where $\hat{A}(k)$ is the 256-point FFT of $\{1,\hat{a}_1,\ldots, \hat{a}_p\}$ with trailing zeros when $p = 14$. We can see the first formant at 352Hz, second formant at 937Hz and third one at 2695Hz. The first and second formant fit the result in literature well, while the result of the third one may be caused by the difference of speech. Fig. 20 and 21 show the similar process of using 64-point of $/\eta/$, which have similar trend. Estimates of the formant frequencies for the vowel sounds using 64-point segment and the value in literature are shown in Table 2.

Table 2. Comparison of formant frequencies for **vowel** with that in literature (64 points)

Vowel	F_1		F_2		F_3	
	Literature	Empirical	Literature	Empirical	Literature	Empirical
$/\mathrm{i}/$	270Hz	313Hz	2290Hz	2305Hz	3010Hz	3203Hz
$/\mathrm{u}/$	300Hz	313Hz	870Hz	938Hz	2240Hz	2695Hz

- **Summary**

In this assignment, we do high resolution spectral analysis on speech signals and compare the result with our conventional single periodogram method. The former one has broken the fundamental trade-off between variance reduction and horizontal resolution in conventional spectral analysis at the expense of introducing pre-knowledge. And we usually just cannot get that much points to do averaging in speech signals process, so we use autocorrelation method of linear prediction in this assignment. The formant frequencies match the result in literature when p=14 to some degree, and the error power would firstly decrease if we increase p and then remain stable and we wouldn’t benefit much to increase the order then.

- **Plots**

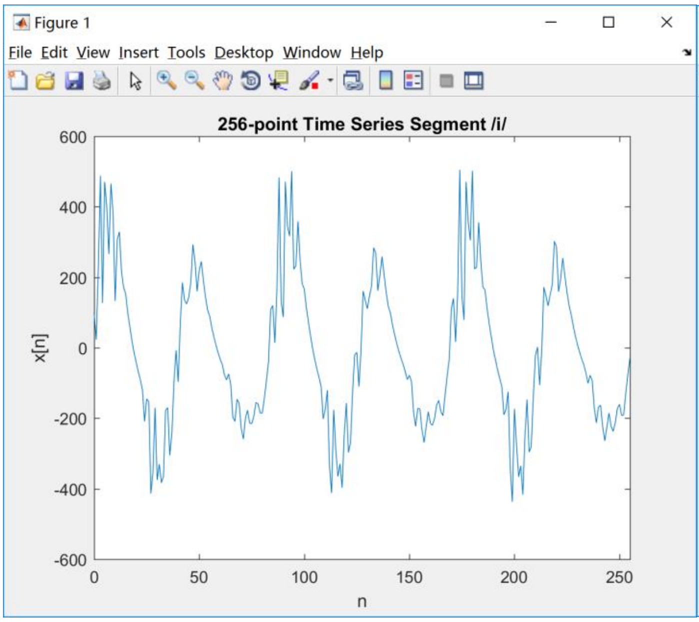

Fig 1

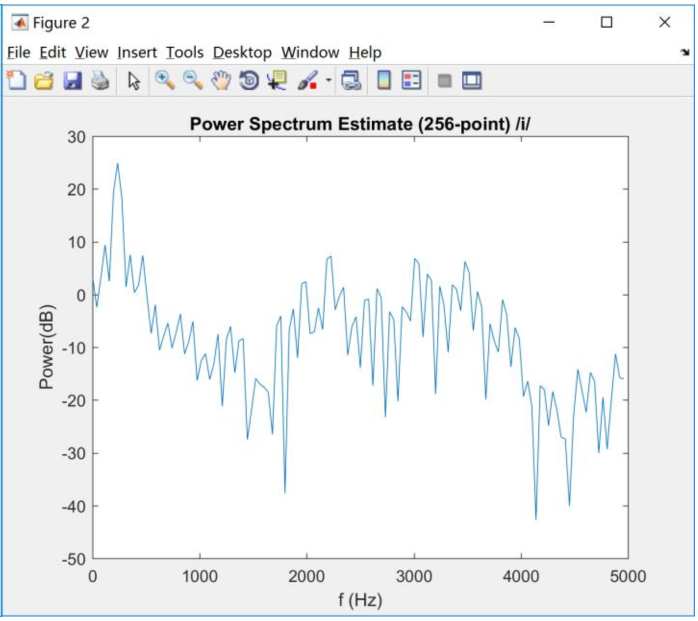

Fig 2

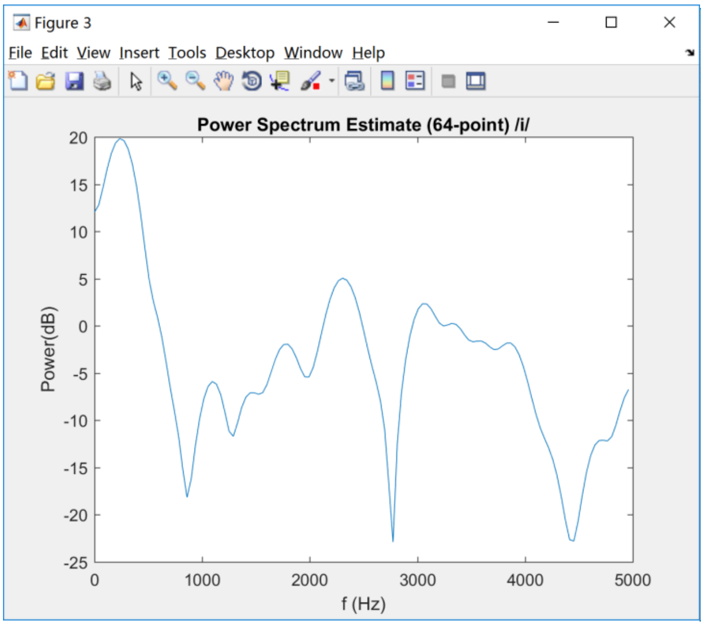

Fig 3

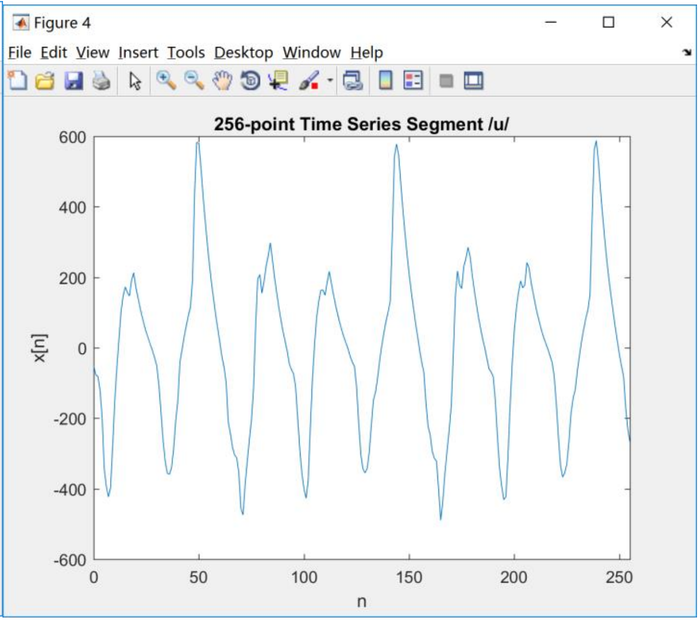

Fig 4

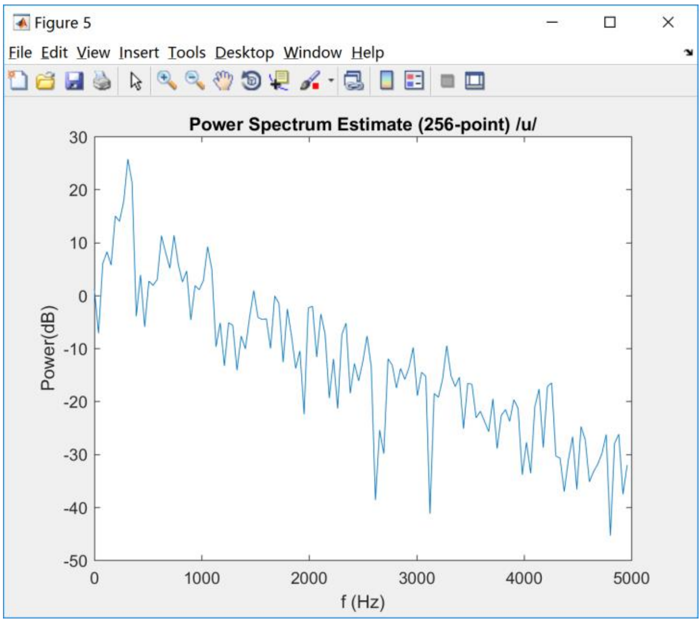

Fig 5

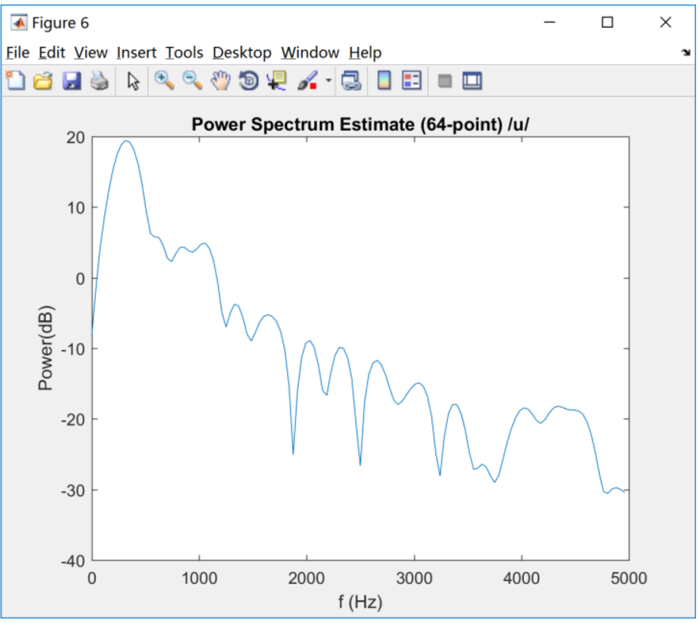

Fig 6

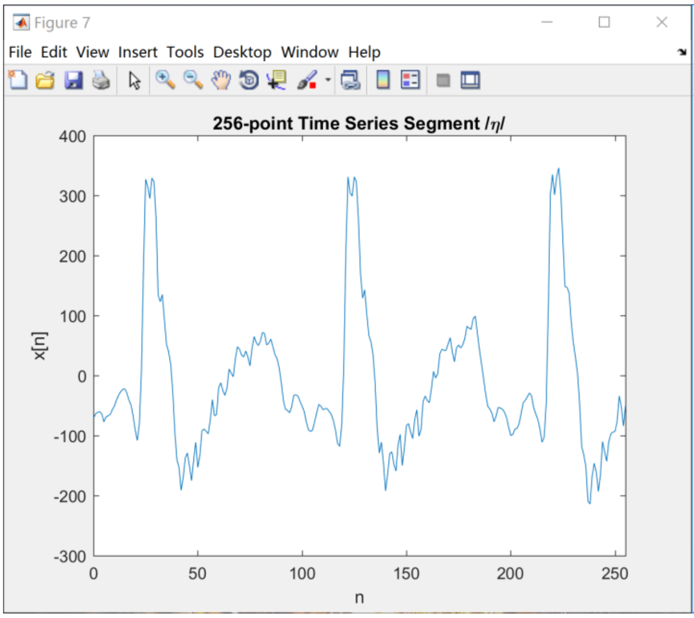

Fig 7

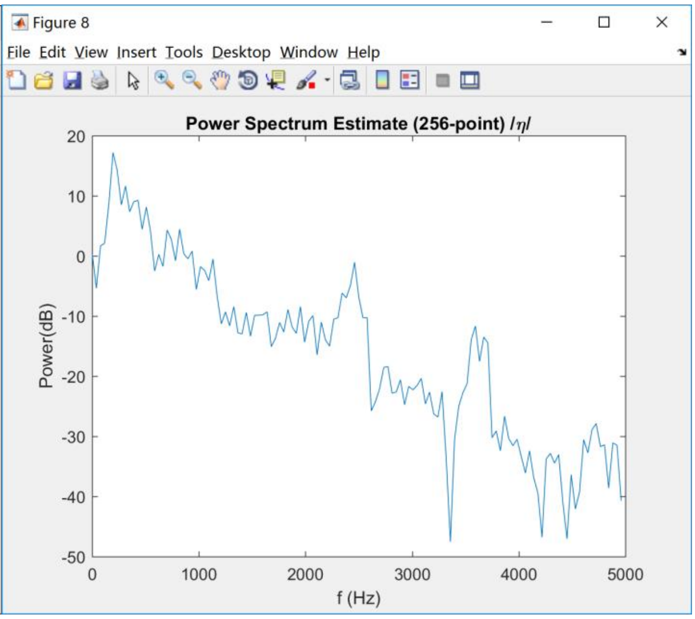

Fig 8

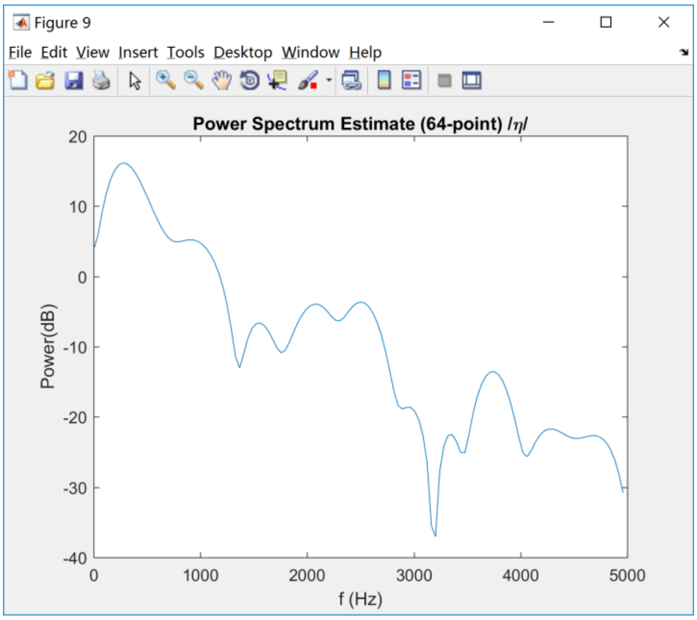

Fig 9

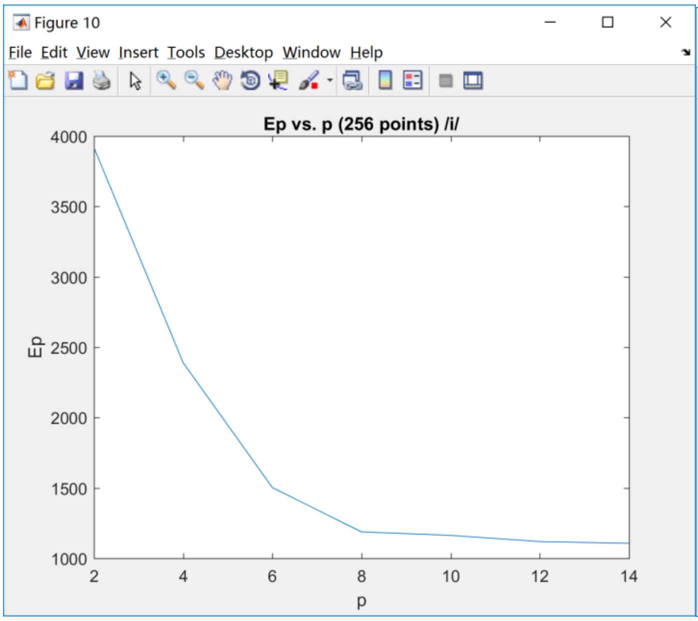

Fig 10

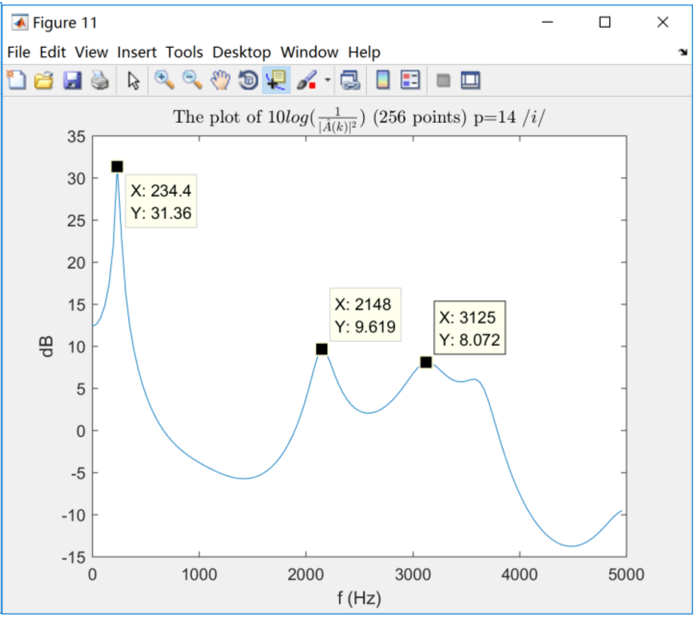

Fig 11

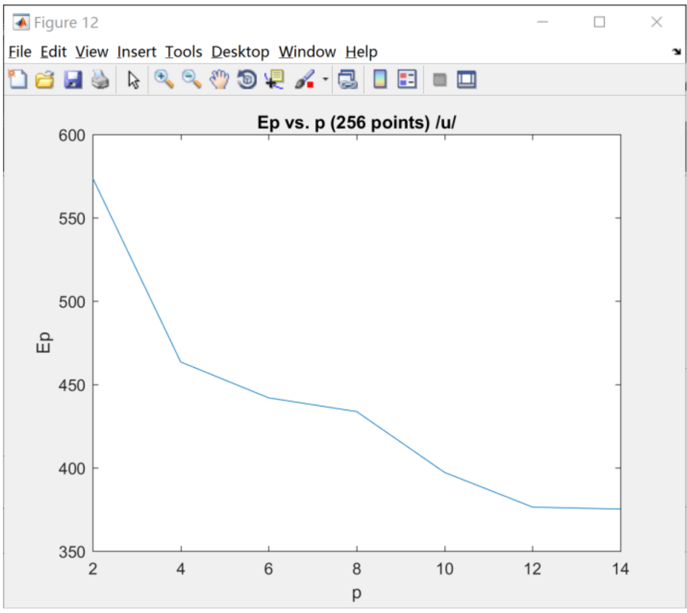

Fig 12

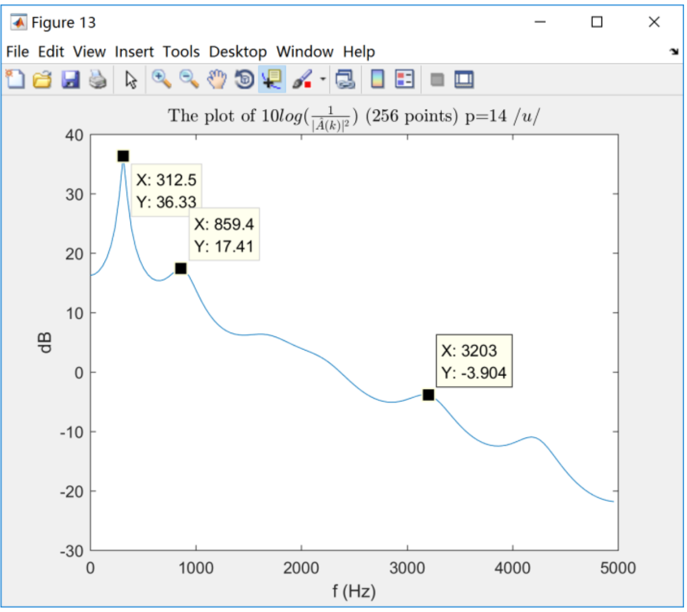

Fig 13

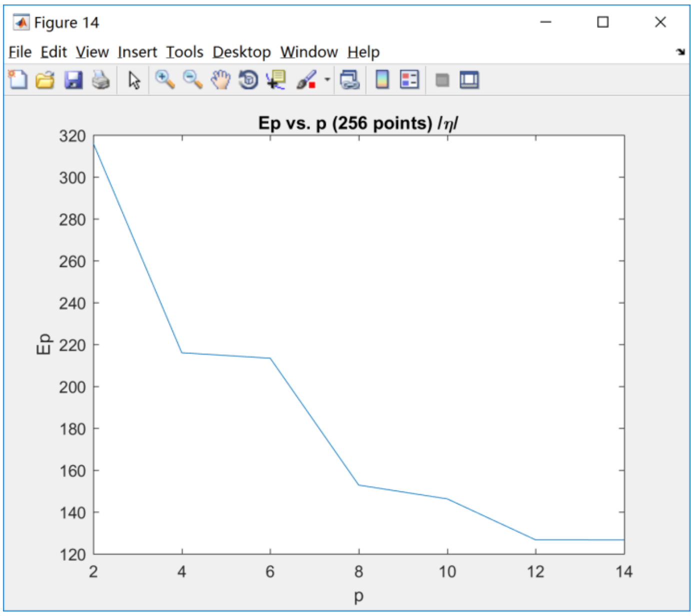

Fig 14

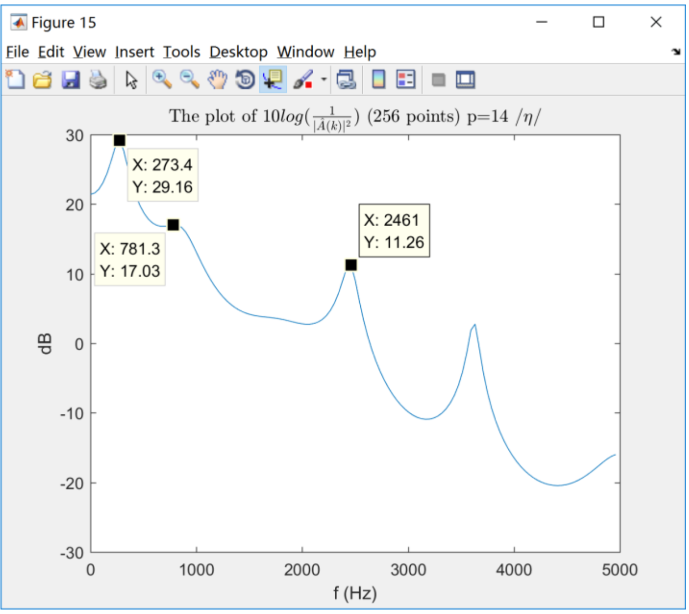

Fig 15

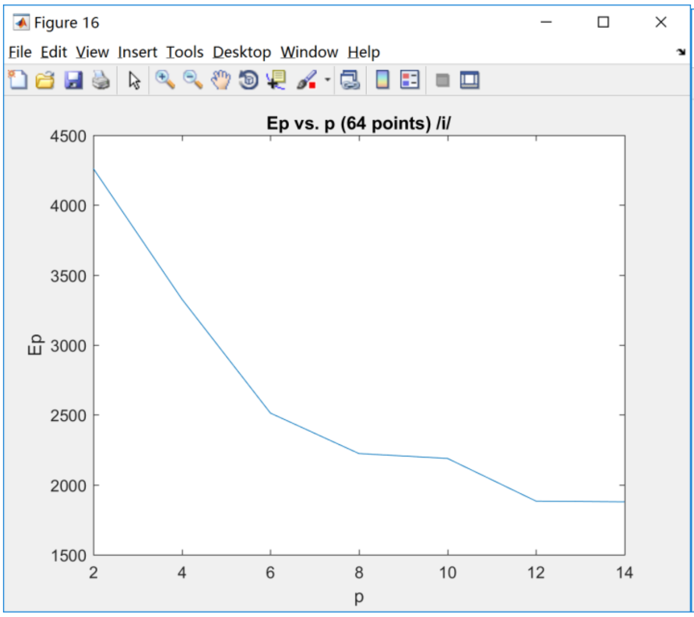

Fig 16

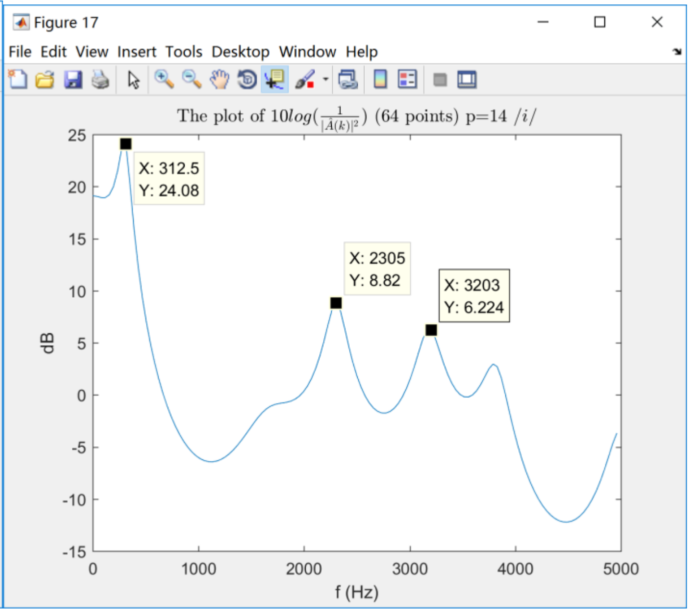

Fig 17

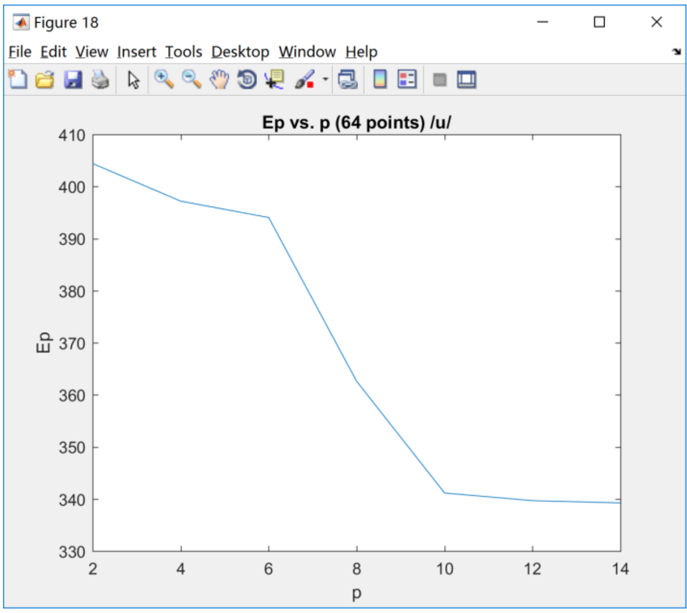

Fig 18

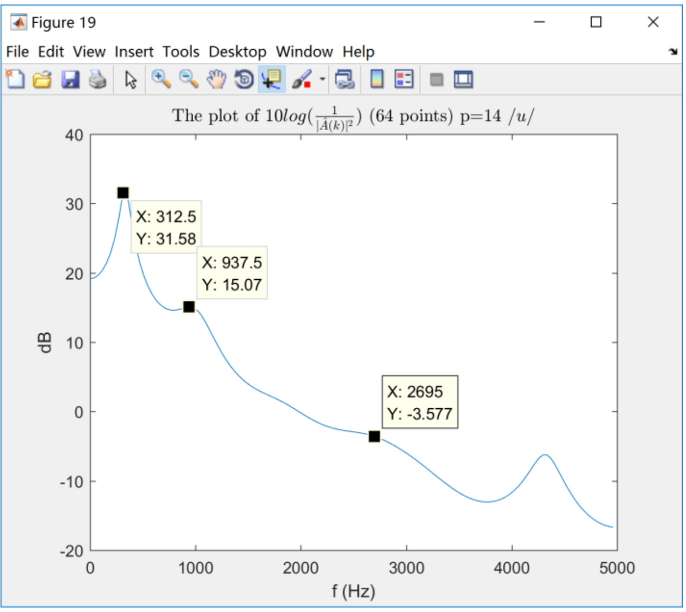

Fig 19

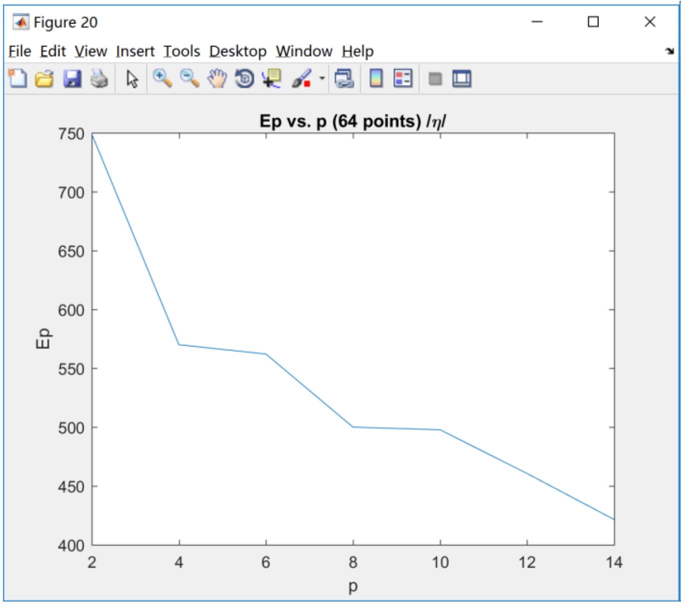

Fig 20

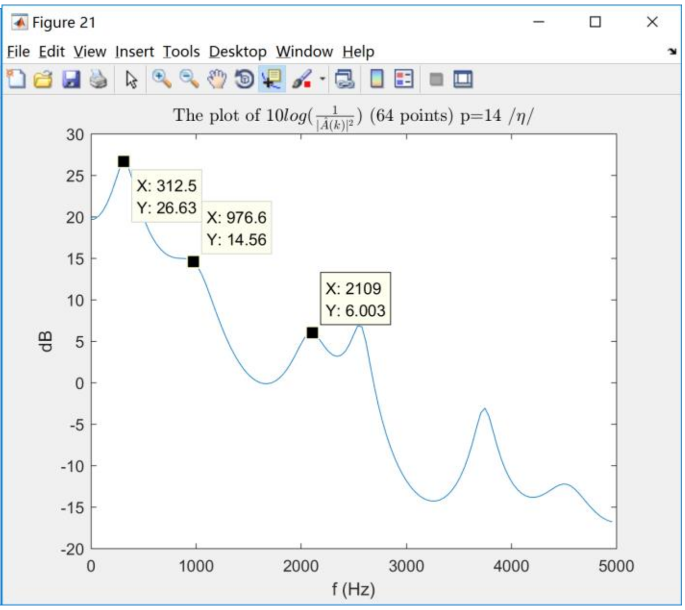

Fig 21

- Appendix

Script:

```
clc;clear;close all
load -ascii bili_single_col.txt
load -ascii bilu_single_col.txt
load -ascii bilng_single_col.txt
load -ascii jcnwwa_single_col.txt
%% Part I
sig_all = [bili_single_col bilu_single_col bilng_single_col];
N = 256;
f=[0:10000/N:5000-10000/N];
str = 'iun';
window=hanning(N);
window2=hanning(64);
U=sum(window.^2);
U2=sum(window2.^2);
for i = 1:3
    sig = sig_all(1024:1279,i);
    figure(3*(i-1)+1);
    plot(0:N-1,sig);
    title(['256-point Time Series Segment /',str(i),'/']);
    xlabel('n');ylabel('x[n]');
    xlim([0 255]);

    figure(3*(i-1)+2);
    X=fftshift(fft(sig.*window));
    X=X(129:256);
    plot(f,10*log10((abs(X).^2)/U/10000));
    xlabel('f (Hz)');ylabel('Power(dB)');title(['Power Spectrum
Estimate (256-point) /',str(i),'/']);

    figure(3*i);
    X=fftshift(fft(sig(1:64).*window2,N));
    X=X(129:256);
    plot(f,10*log10((abs(X).^2)/U2/10000));
    xlabel('f (Hz)');ylabel('Power(dB)');title(['Power Spectrum
Estimate (64-point) /',str(i),'/']);
end;
%% Part II
for i = 1:3
    sig = sig_all(1024:1279,i);
    sig = sig .* window;
    figure(2*(i-1)+10);
    pbin=[2:2:14];
    Ep=[];
    for j=1:size(pbin,2)
        p=pbin(j);
        [a,g] = lpc(sig,p);
        Ep=[Ep g];
    end;
    plot(pbin,Ep);
    xlabel('p');ylabel('Ep');
    title(['Ep vs. p (256 points) /',str(i),'/']);
    xlim([2,14]);

    figure(2*(i-1)+11);
    p=14;
    [a,g] = lpc(sig,p);
    A=fftshift(fft(a,N));
    A=A(129:256);
    plot(f,(10*log10(1./(abs(A).^2))));
    xlabel('f (Hz)');ylabel('dB');
    title(['The plot of $10log(\frac{1}{|\hat{A}(k)|^2})$ (256 points)
p=14 /',str(i),'/'],'Interpreter','latex');
end;

for i = 1:3
    sig = sig_all(1024:1279,i);
    sig = sig(1:64);
    sig = sig .* window2;
    figure(2*(i-1)+16);
    pbin=[2:2:14];
    Ep=[];
    for j=1:size(pbin,2)
        p=pbin(j);
        [a,g] = lpc(sig,p);
        Ep=[Ep g];
    end;
    plot(pbin,Ep);
    xlabel('p');ylabel('Ep');
    title(['Ep vs. p (64 points) /',str(i),'/']);
    xlim([2,14]);

    figure(2*(i-1)+17);
    p=14;
    [a,g] = lpc(sig,p);
    A=fftshift(fft(a,N));
    A=A(129:256);
    plot(f,(10*log10(1./(abs(A).^2))));
    xlabel('f (Hz)');ylabel('dB');
    title(['The plot of $10log(\frac{1}{|\hat{A}(k)|^2})$ (64 points)
p=14 /',str(i),'/'],'Interpreter','latex');
end;
```