# Homework 7

# High Resolution Spectral Analysis

**Name:** Jinhan Zhang

**Class:** ECE 251A  Digital Signal Processing I

**Date:** 03/07/2017

# High Resolution Spectral Analysis

## • Objective

Use Matlab to do high resolution spectral analysis. Instead of using conventional power spectral analysis, we use autocorrelation method of linear prediction to do spectral analysis and analyze the results.

## • 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. In this assignment, we will use autocorrelation method of linear prediction to do high resolution power spectral analysis.

## • Approach

First of all, we will implement the generation of the process. We are using an all-pole filter H(z), and we have $\mathrm{H(z)} = \frac{1}{A(z)}$, where A(z) is given by

$$\mathrm{A(z)} = \prod_{i=1}^{4} (1 - z_i z^{-1})\,(1 - z_i^* z^{-1})$$

Where $z_i = r_i e^{j\theta_i}$ and parameters are given as table 1 below.

Table. 1 parameters for A(z)

| i | $\theta_i$ | $r_i$ |
|---|---|---|
| 1 | $\pi/8$ | 0.9896 |
| 2 | $2\pi/8$ | 0.9843 |
| 3 | $3\pi/8$ | 0.9780 |
| 4 | $4\pi/8$ | 0.9686 |

Then we make zero/pole plot for H(z), then do 256-point FFT to $\{a_0,\ldots,\ a_8\}$ with trailing zeros and plot $10\log(\frac{1}{|A(k)|^2})$. We will also plot 256-point impulse response h(n), compute 256-point FFT and plot $10\log(|H(k)|^2)$.

Secondly, we pass the Gaussian noise with zero-mean and unit-variance through H(z) to be our observed data x(n). We then plot the power spectral estimation for a 256-point segment of x(n) using single periodogram and Welch's method of averaging modified periodogram respectively.

Finally, we use the retained time series x(n), 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 finally we repeat the process using a 32-point segment of x(n).

In all the places need to window, I choose to use Hanning window.

- **Results**

In Fig. 1, we can see that H(z) has 8 poles as we expected. The poles which have smaller absolute value of $\theta$ have larger distance from the origin. Fig. 2 shows the plot of $10\log(\frac{1}{|A(k)|^2})$, where A(k) is the 256-point FFT of $\{a_0,\ldots,\ a_8\}$ with trailing zeros. Fig. 3 is the 256-point impulse response of the all-pole filter we build, and Fig.4 is the corresponding $10\log(|\mathrm{H(k)}|^2)$, where H(k) is the 256-point FFT of h(n). We can see that Fig. 4 is pretty similar to Fig. 2 due to the fact that $\mathrm{H(z)} = \frac{1}{A(z)}$.

Fig. 5 shows the power spectral estimation of the retained x(n) using single periodogram method with normalization, while Fig.6 is the power spectral estimation of the retained x(n) using Welch's method of averaging modified periodogram. We can easily see that Fig. 6 has lower variance compared to Fig.5, and that is at the expense of horizontal resolution. This is always the trade-off in conventional spectral analysis.

Fig. 7 is 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 $p = 2$, we can see the curve has one formant. Fig. 8 is the corresponding zero/pole plot of the inverse filter. Fig. 9 is 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 $p = 8$, we can see the curve has four formants and is almost identical with what we have in Fig. 2. Fig. 10 is the corresponding zero/pole plot of the inverse filter, we can see that the locations of zeros are the same as the location of poles in Fig. 1. Fig. 11 is 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 that the curve still has 4 formants and seems to have no difference compared with Fig. 9. Fig. 12 is the corresponding zero/pole plot of the inverse filter, we can see that besides the zeros shown in Fig. 10, there are 6 more other zeros in the unit circle, this is due to the increase of $p$ in our linear prediction. Then Fig. 13 shows how the prediction error power changes with the increase of p value. 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 near 0.

Fig. 14 shows the power spectral estimation of the 32-point retained x(n) using single periodogram method with normalization, in which we zero fill out the retained time series to 256-length and use 256-point FFT. Fig. 15-20 are the same pattern with Fig. 7-12. We can see that the only pair of zeros in Fig. 16 get inward slightly, in Fig. 18 the locations of zeros are somehow moved away from the corresponding locations of poles in Fig. 1 and in Fig. 20 the locations of zeros are more inaccurate when compared with Fig. 12. Thus the spectral estimations in Fig. 15/17/19 are more twisted when compared with Fig. 2. Then Fig. 21 shows how the prediction error power changes with the increase of p value in 32-point retained time series case. We can see that the trend of error power is similar to Fig. 13 (i.e. 256-point case), but the 32-point case gives a higher error power for every corresponding p value in 256-point case. That is reasonable since we have less points (i.e. less information) to do the estimation.

- **Summary**

In this assignment, we do high resolution spectral analysis and compare it with our conventional power spectral analysis method. The former one has broken the fundamental trade-off between variance reduction and horizontal resolution in conventional spectral analysis. But also this is completed at the expense of introducing pre-knowledge. And we use autocorrelation method of linear prediction in this assignment, 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**

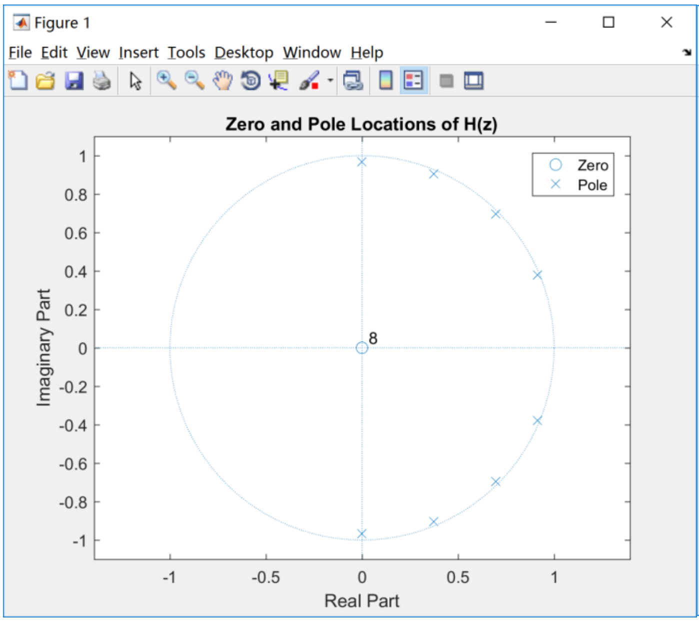

Fig. 1

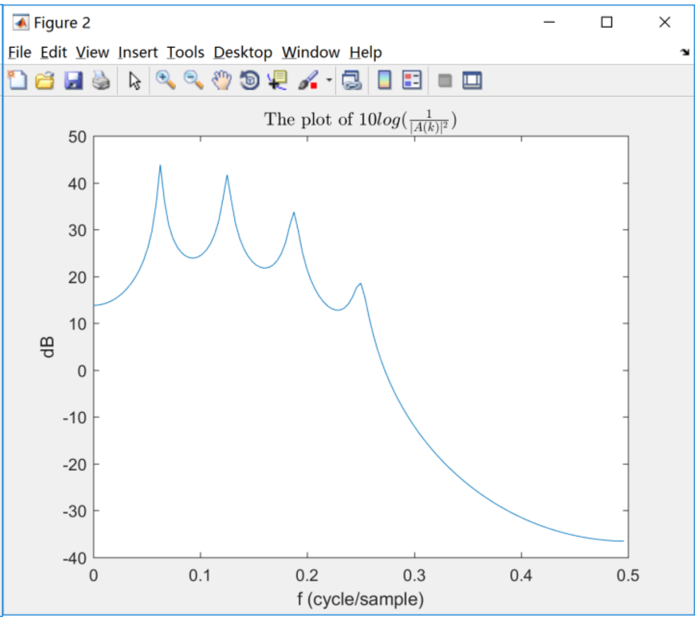

Fig. 2

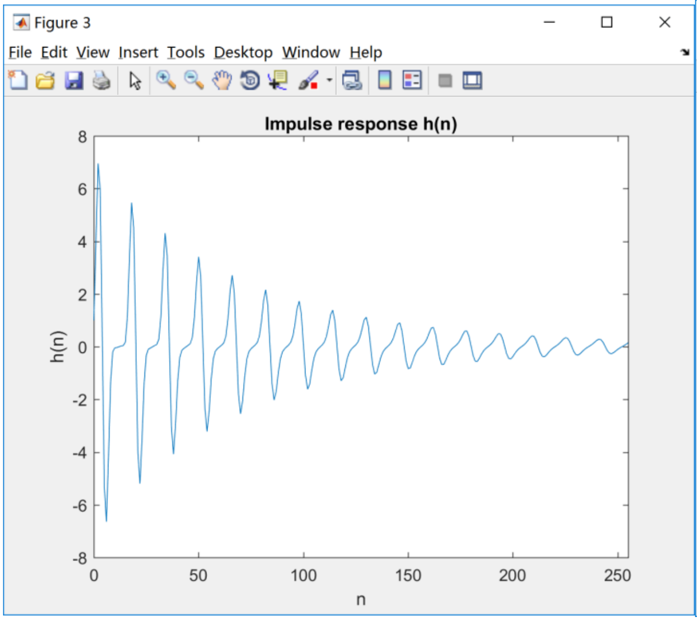

Fig. 3

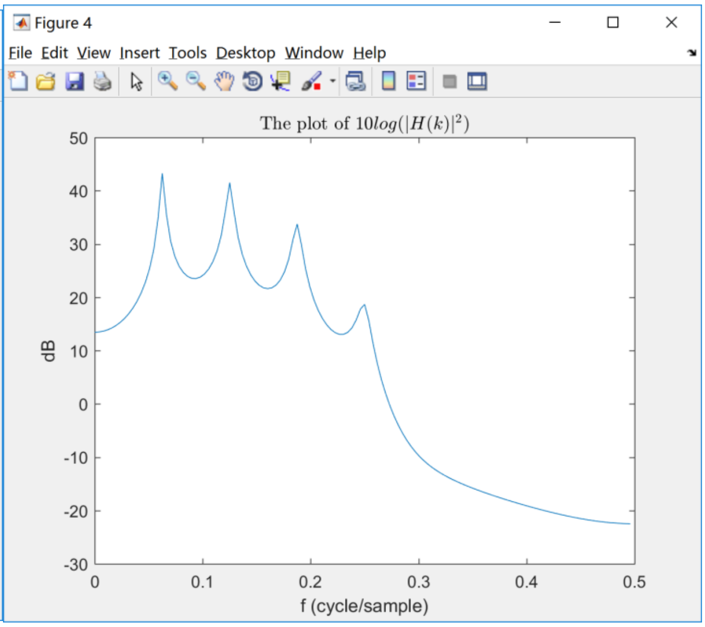

Fig. 4

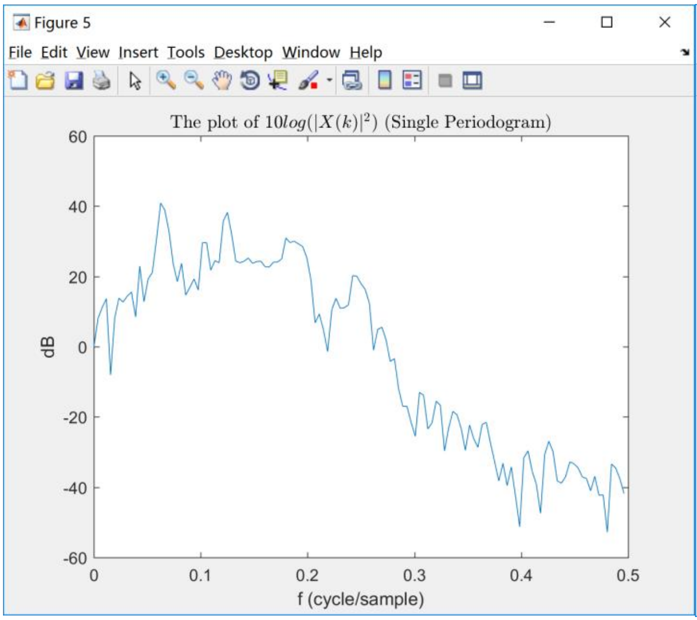

Fig. 5

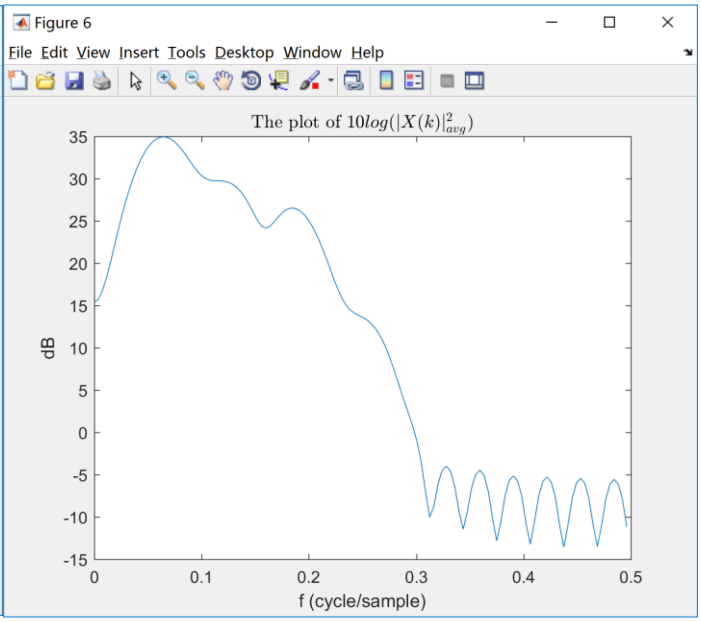

Fig. 6

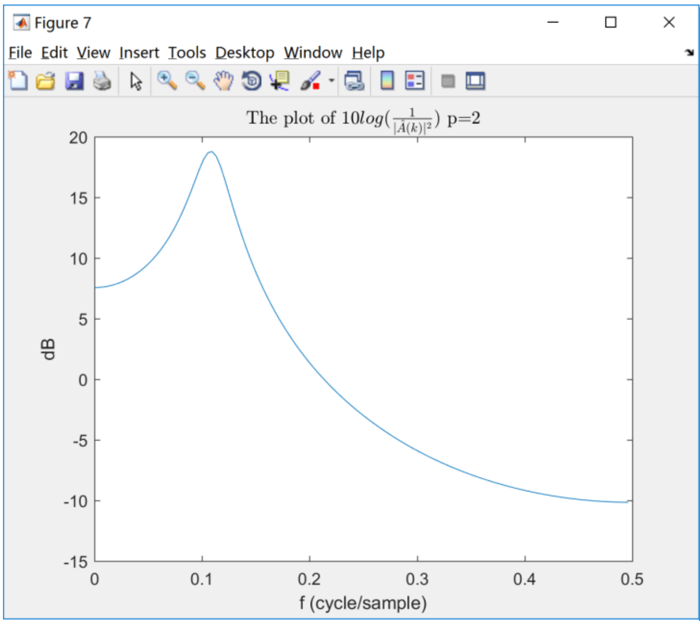

Fig. 7

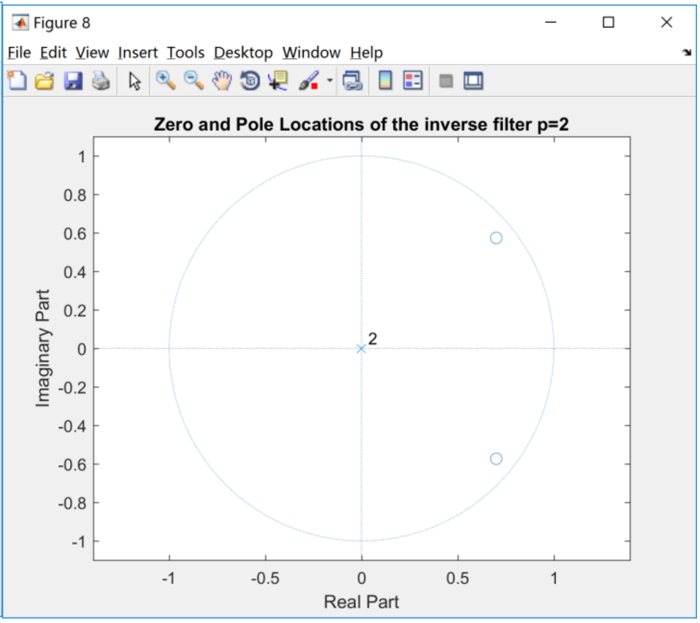

Fig. 8

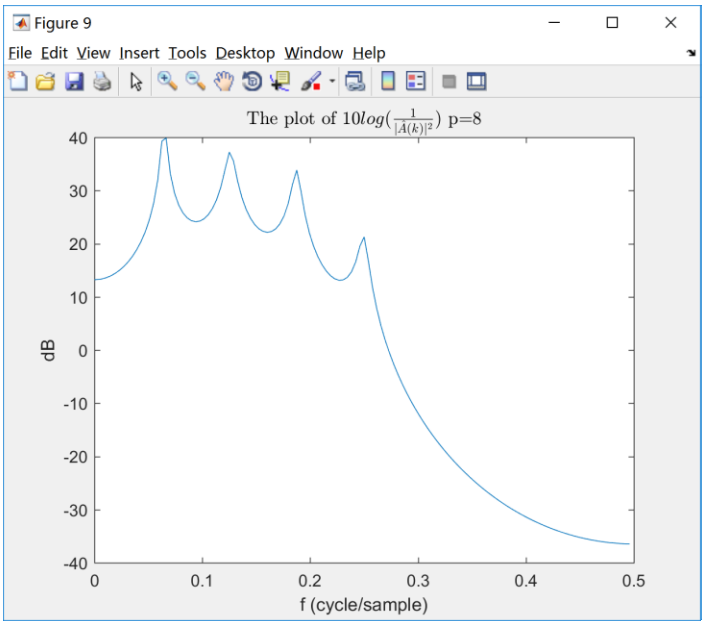

Fig. 9

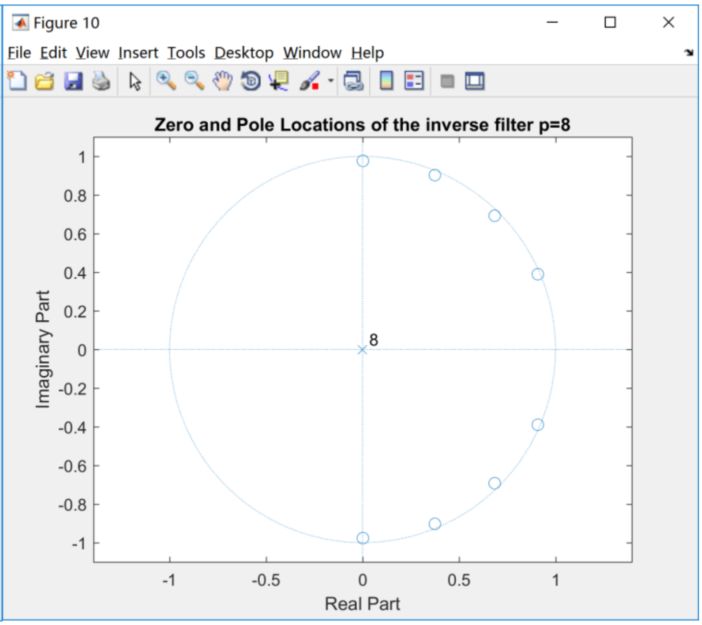

Fig. 10

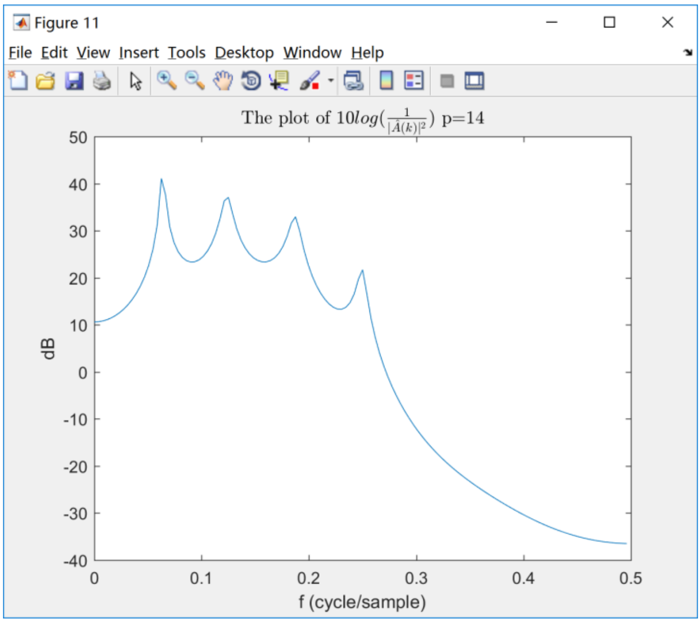

Fig. 11

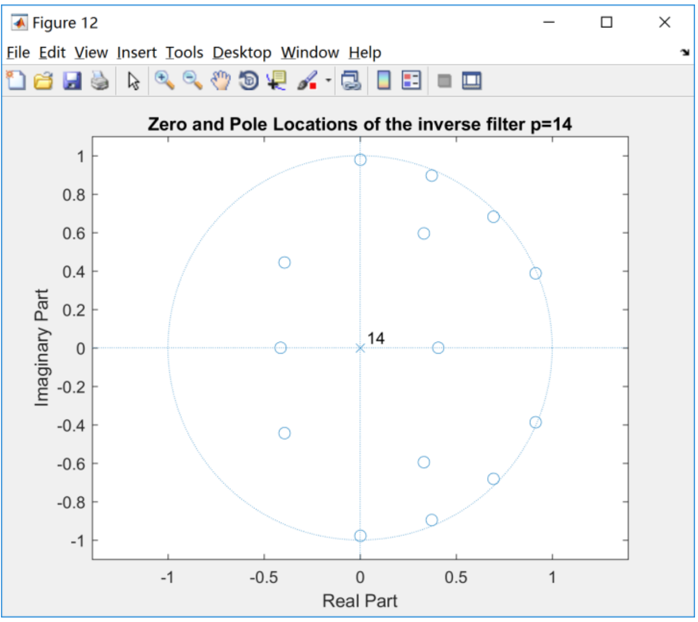

Fig. 12

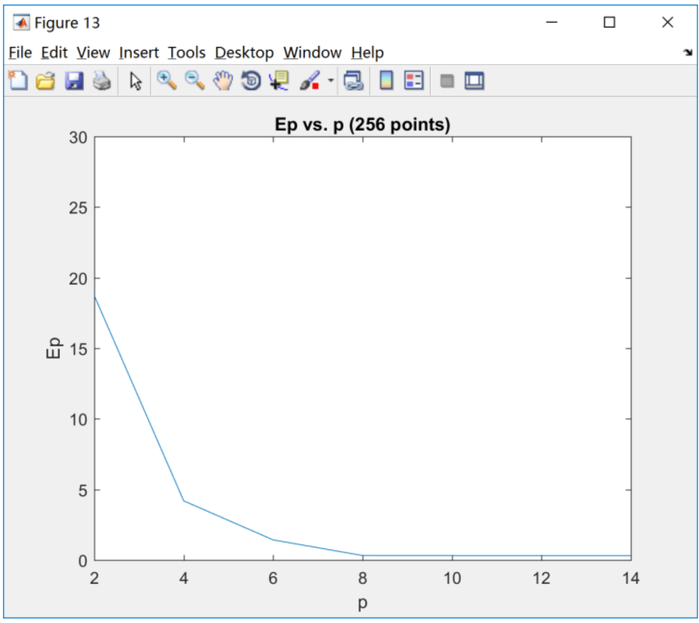
Ep vs. p (256 points)

Fig. 13

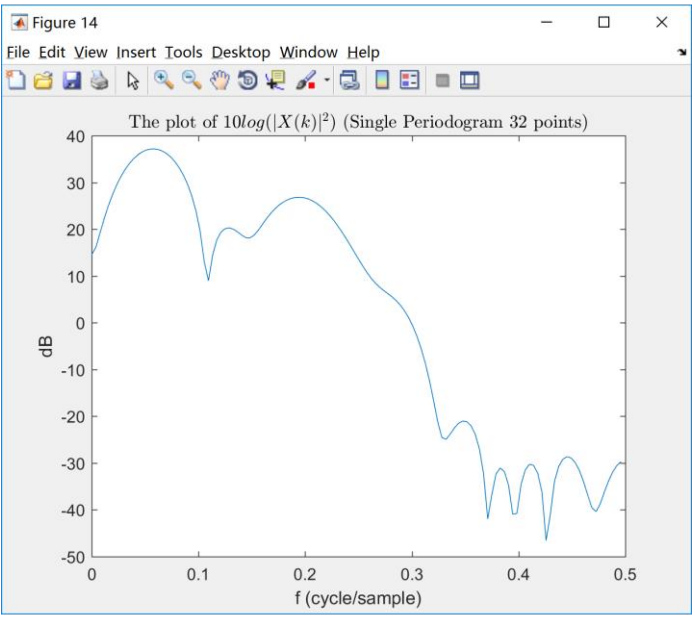
The plot of $10log(|X(k)|^2)$ (Single Periodogram 32 points)

Fig. 14

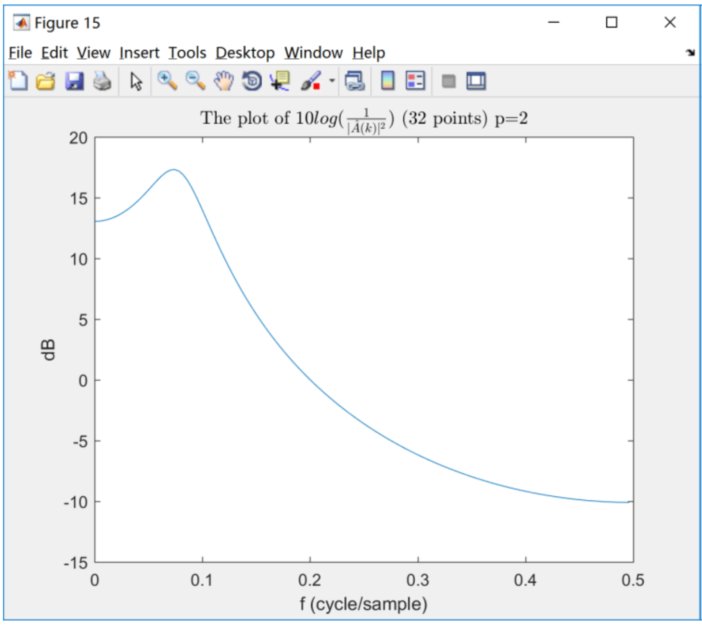
The plot of $10log(\frac{1}{|A(k)|^2})$ (32 points) p=2

Fig. 15

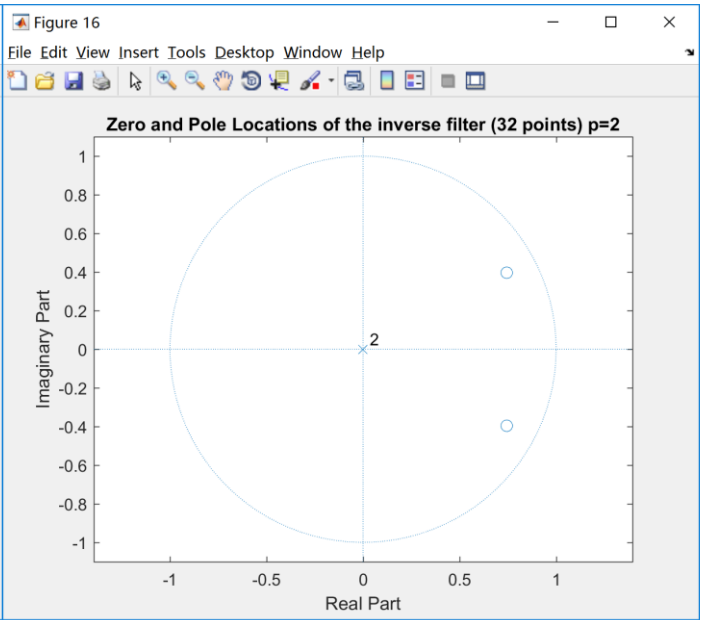
Zero and Pole Locations of the inverse filter (32 points) p=2

Fig. 16

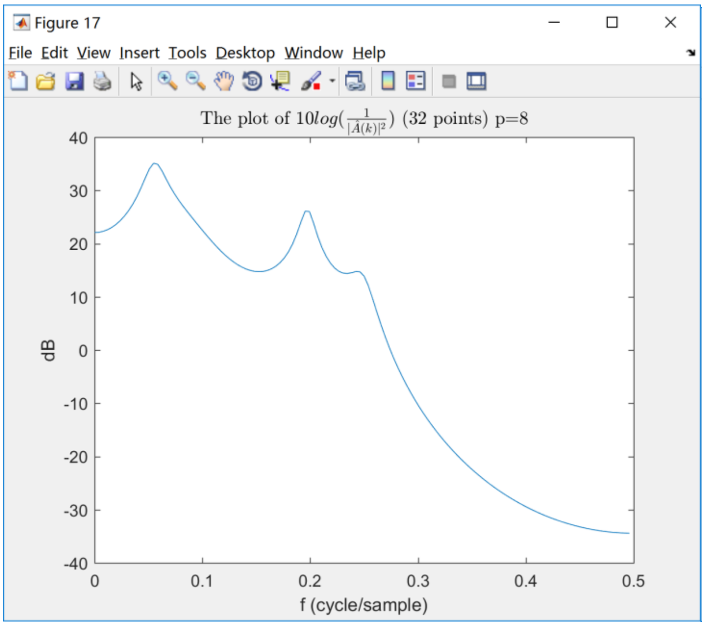
The plot of $10log(\frac{1}{|A(k)|^2})$ (32 points) p=8

Fig. 17

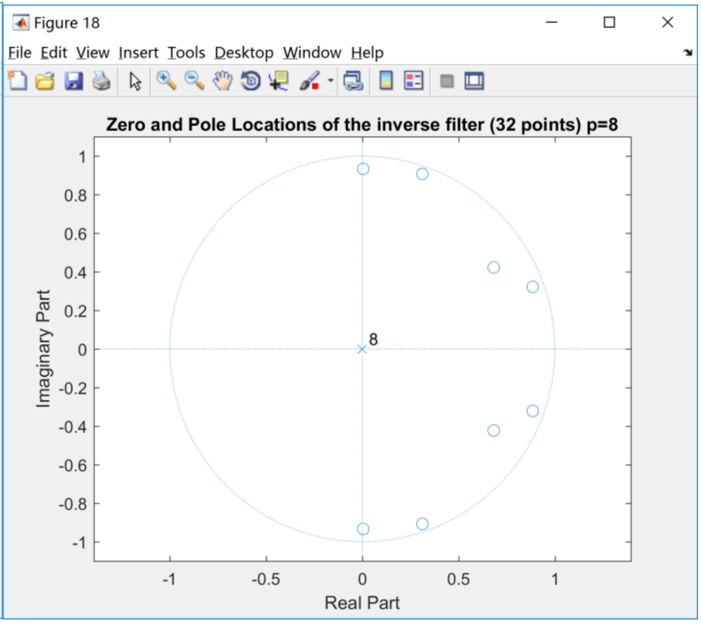
Zero and Pole Locations of the inverse filter (32 points) p=8

Fig. 18

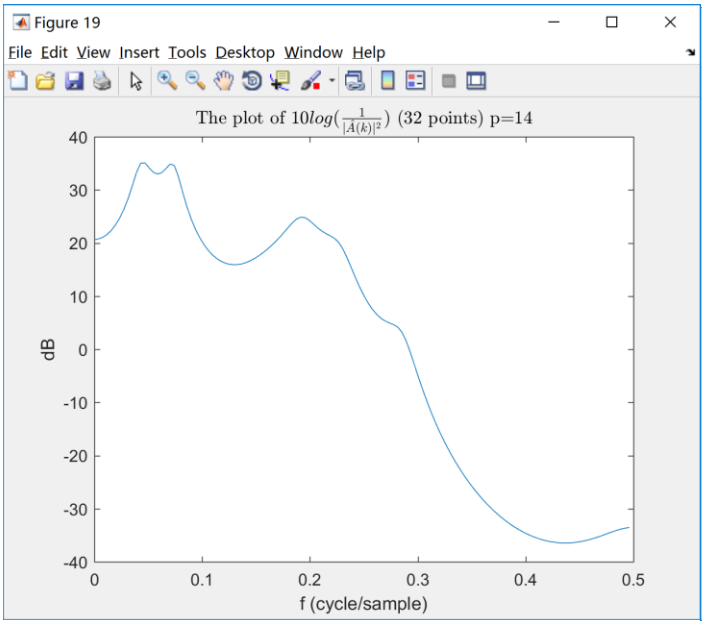

Fig. 19

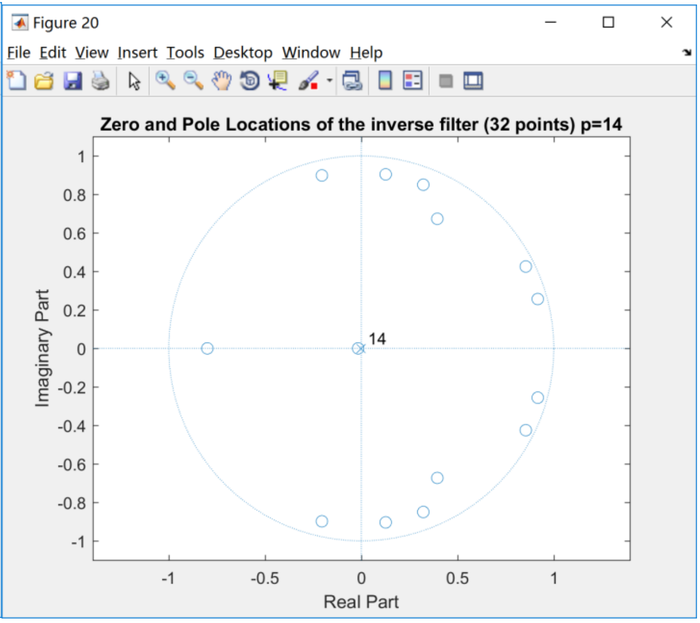

Fig. 20

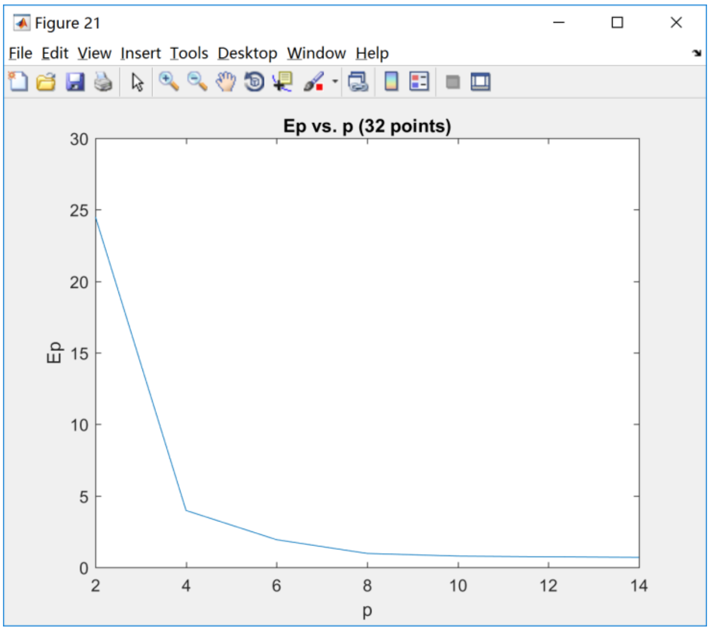

Fig. 21

- Appendix

Script:

```
clear;clc;close all;
%% Part I
rng(8);
z=[0.9896*exp(1i*pi/8) 0.9896*exp(-1i*pi/8) 0.9843*exp(1i*pi/4)
0.9843*exp(-1i*pi/4) ...
    0.9780*exp(1i*3*pi/8) 0.9780*exp(-1i*3*pi/8) 0.9686*exp(1i*pi/2)
0.9686*exp(-1i*pi/2)];
den=poly(z);
num=[1 zeros(1,8)];
figure(1);
zplane(num,den);
```

```
legend('Zero','Pole');
title('Zero and Pole Locations of H(z)');

figure(2);
N=256;
f=[0:0.5/128:0.5-0.5/128];
A=fftshift(fft(den,N));
A=A(129:256);
plot(f,(10*log10(1./(abs(A).^2))));
xlabel('f (cycle/sample)');ylabel('dB');
title('The plot of $10log(\frac{1}{|A(k)|^2})$','Interpreter','latex');

figure(3);
x=[1 zeros(1,255)];
h=filter(num,den,x);
plot(0:255,h);
xlim([0,255]);title('Impulse response h(n)');
xlabel('n');ylabel('h(n)');

figure(4);
H=fftshift(fft(h,N));
H=H(129:256);
plot(f,(log10((abs(H).^2)))*10);
xlabel('f (cycle/sample)');ylabel('dB');
title('The plot of $10log(|H(k)|^2)$','Interpreter','latex');

%% Part II
wn=randn([1,1280]);
figure(5);
xn=filter(num,den,wn);
xn=xn(1024:1279);
window=hanning(N)';     %%window
U=sum(window.^2);
X=fftshift(fft(xn.*window,N));
X=X(129:256);
plot(f,(log10((abs(X).^2)/U))*10);
xlabel('f (cycle/sample)');ylabel('dB');
title('The plot of $10log(|X(k)|^2)$ (Single
Periodogram)','Interpreter','latex');

X_avg = fftshift(pwelch(xn,32,16,256,1,'twosided','psd'));
X_avg=X_avg(129:256);
figure(6);
plot(f,10*log10(X_avg));
xlabel('f (cycle/sample)');ylabel('dB');
title('The plot of $10log(|X(k)|_{avg}^2)$','Interpreter','latex');
%% III
pbin=[2 8 14];
xn1=xn.*window;
```

```
for i=1:size(pbin,2)
   p=pbin(i);
   [a,g] = lpc(xn1,p);
   A2=fftshift(fft(a,N));
   A2=A2(129:256);
   figure(5+2*i);
   plot(f,(10*log10(1./(abs(A2).^2))));
   xlabel('f (cycle/sample)');ylabel('dB');
   title(['The plot of
$10log(\frac{1}{|\hat{A}(k)|^2})$ p=',num2str(pbin(i))],'Interpreter','
latex');
   figure(6+2*i);
   zplane(a);
   title(['Zero and Pole Locations of the inverse filter
p=',num2str(pbin(i))]);
end;

pbin=[2 4 6 8 10 12 14];
Ep=[];
for i=1:size(pbin,2)
   p=pbin(i);
   [a,g] = lpc(xn1,p);
   Ep=[Ep g];
end;
figure(13);
plot(pbin,Ep);
xlabel('p');ylabel('Ep');
title('Ep vs. p (256 points)');
axis([2,14,0,30]);
num=32;
%% IIIC
xn=xn(1:32).*hanning(num)';
X=fftshift(fft(xn,N));
X=X(129:256);
figure(14);
U=sum(hanning(num).^2);
plot(f,(log10((abs(X).^2)/U))*10);
xlabel('f (cycle/sample)');ylabel('dB');
title('The plot of $10log(|X(k)|^2)$ (Single Periodogram 32
points)','Interpreter','latex');

pbin=[2 8 14];
for i=1:size(pbin,2)
   p=pbin(i);
   [a,g] = lpc(xn,p);
   A2=fftshift(fft(a,N));
   A2=A2(129:256);
   figure(13+2*i);
   plot(f,(10*log10(1./(abs(A2).^2))));
```

```
    xlabel('f (cycle/sample)');ylabel('dB');
    title(['The plot of $10log(\frac{1}{|\hat{A}(k)|^2})$ (32 points)
p=',num2str(pbin(i))],'Interpreter','latex');
    figure(14+2*i);
    zplane(a);
    title(['Zero and Pole Locations of the inverse filter (32 points)
p=',num2str(pbin(i))]);
end;

pbin=[2 4 6 8 10 12 14];
Ep=[];
for i=1:size(pbin,2)
    p=pbin(i);
    [a,g] = lpc(xn,p);
    Ep=[Ep g];
end;
figure(21);
plot(pbin,Ep);
xlabel('p');ylabel('Ep');
title('Ep vs. p (32 points)');
axis([2,14,0,30]);
```