Homework 4

Spectral Averaging

Name: Jinhan Zhang

Class: ECE 251A Digital Signal Processing I

Date: 02/06/2017

Spectral Averaging

• Objective

In this assignment, we do estimation of the power spectrum using different ways, and investigate the effect of segmenting a time series when doing power spectrum estimations.

• Background

Power spectrum estimation plays a very important role in digital signal processing, and has plenty of fields of applications. In general, we can use single periodogram, Bartlett's Procedure of Averaging Periodograms and Welch's Method of Averaging Modified Periodograms to do power spectrum estimation. In this assignment, we study on the power spectrum estimation of classical sinusoid plus Gaussian noise signal.

• Approach

First, we generate a 1024-point time series consisting of a bin centered sinusoid for a FFT of length NFFT = 128, and add Gaussian white noise to it.

Because we have:

$$\mathrm{SNR} = 10\log\frac{A}{2\sigma^2} = 0dB$$

And $\sigma^2 = 1$, so we can get $\mathrm{A} = \sqrt{2}$.

Secondly, we use single segments of length 128, 256, 512, and 1024 points to do periodogram power spectrum estimation respectively. I choose Hamming window for all 4 cases and use $(f_s MU)^{-1}$ to normalize the power spectrum.

Then we investigate on power spectrum estimate of averaging methods. We split the original 1024-point time into $\mathrm{K} = 8,15,29$ segments, with $0\%, 50\%, 75\%$ overlap respectively. Then we average the K spectra together to get a power spectrum estimate.

Finally, we use coherent average to do power spectrum estimate which is rarely used in practice.

• Results

Part A: Fig. 1 shows the 128-point segment I extracted from the original 1024-point signal, which is a sinusoid ($\mathrm{A} = \sqrt{2}\ \ f_s = 0.125\ \ cycle/sample$) plus Gaussian white noise.

Part B: Fig. 2-5 show the single segment power spectrum estimation with length N=128, 256, 512, 1024. We can see from the plots that the SNR gains are around 18dB, 21 dB, 24 dB ,27 dB for four N values respectively. But the variances of periodograms remain almost the same as N increases. That is due to:

$$\mathrm{Var}[I_N(\omega)] = P_{xx}^2(\omega)\{1 + (\frac{sin\omega N}{Nsin\omega})^2\}$$

So when N increases, the “1” term always remains, and the change of the latter term becomes tiny as N is large. In this part and following parts, I use hamming window for all segments, and the power spectrum estimate results are all normalized by $(f_s MU)^{-1}$.

Part C: In Part C, we use averaging periodograms to do power spectrum estimation. We segment the original 1024-point time series into K 128-points segments. We have $\mathrm{K} = 8, 15, 29$ which accordingly means $0\%, 50\%, 75\%$ overlap in each segment. Fig. 6 shows the averaging periodograms power spectrum estimation for all 3 cases above, we can see that the peak values are the same in all 3 cases. The Fig.6 K=8 case has a great reduction on variance compared with Fig.2 no averaging case, which is mainly because the variance after averaging $\mathrm{Var}[B_{xx}(\omega)] \approx \frac{1}{K} P_{xx}^2(\omega)$.

Fig. 7 shows the part of noise alone regions of the power spectrum for different K values. We can see from the plots that the variances of 50% and 75% overlap cases are greatly smaller than 0% overlap case. But the variance reduction of 75% overlap case compared with 50% overlap case is much less. This is because $\mathrm{Var}[B_{xx}(\omega)] \approx \frac{1}{K} P_{xx}^2(\omega)$ relies on the assumption that each segment is independent. The independence of the segments will decrease with the increment of overlap, so the effect of variance reduction will become less and less.

Part D: Fig. 8 shows averaged time series using the 8 segments in Part C(1) and Fig. 9 shows its corresponding power spectrum estimate. Fig. 10 shows the power spectrum estimate of the coherently averaged FFT's.

Fig. 9 and Fig. 10 have exactly the same shape because they are essentially equal. Since we are using a bin centered sinusoid, the additions are along the same direction, so we still have a good estimation of power spectrum in this case. But when we deal with a non bin centered sinusoid, this may be not the case. In that case, the "add" may be not the real add, the difference of phase may lead to offset, thus the sinusoid in power spectrum estimation may be weakened or even disappeared. So this kind of method has very limited range of use in practice.

- **Summary**

In this assignment, we have used classical sinusoid buried in "white" noise model to investigate different ways to do power spectrum estimation. We have tested the performances of single periodogram, averaging periodograms and overlapping averaging periodograms, and the 50% overlapping averaging periodograms has the best widespread use. We also test the coherent averaging way, which is rarely used in practice.

• **Plots**

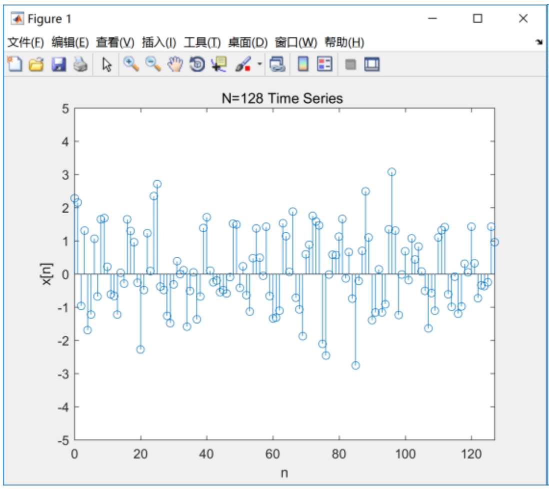

Fig. 1

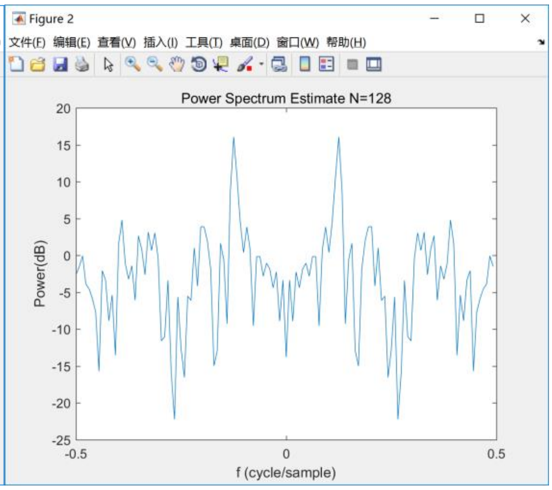

Fig. 2

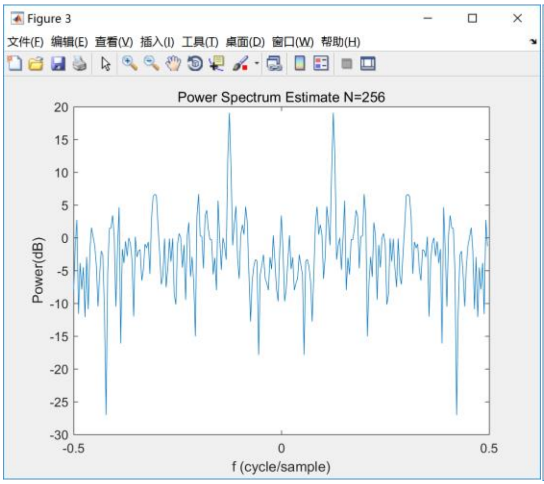

Fig. 3

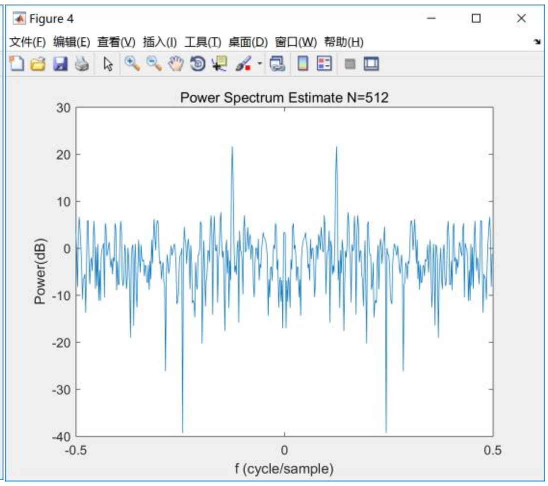

Fig. 4

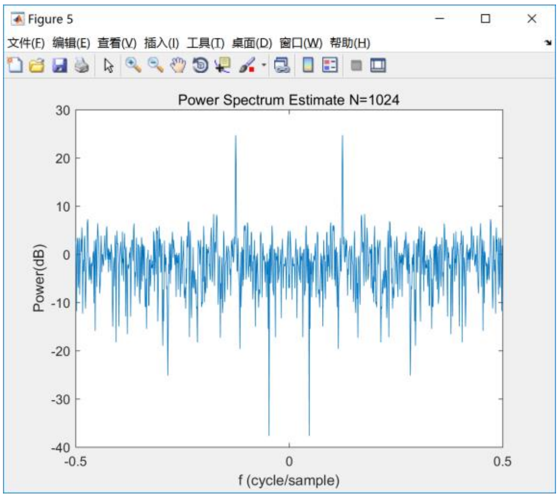

Fig. 5

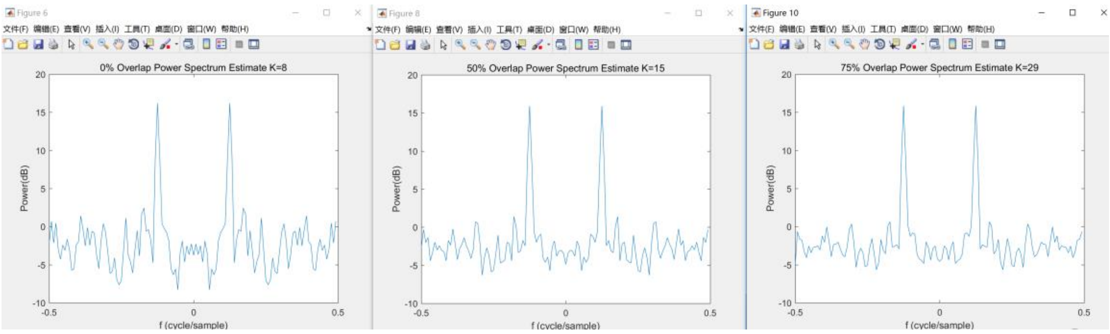

Fig. 6

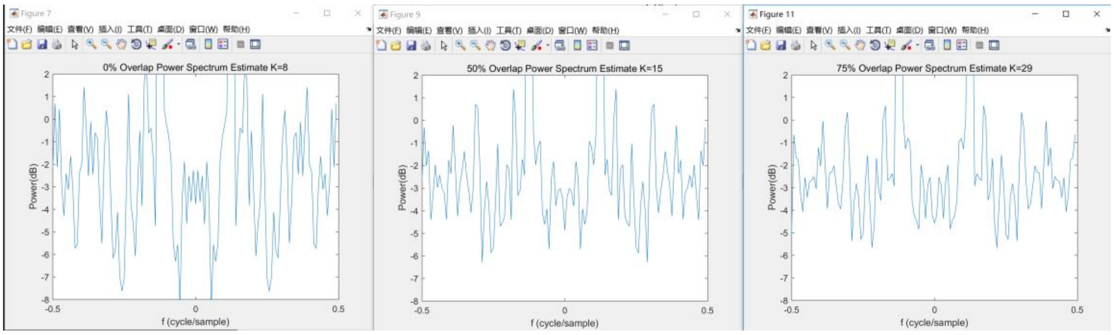

Fig. 7

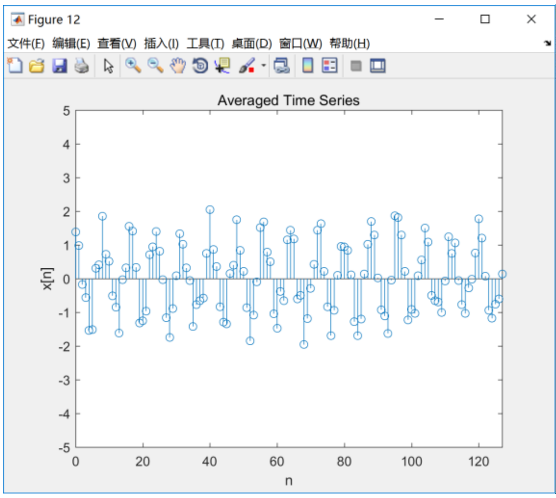

Fig. 8

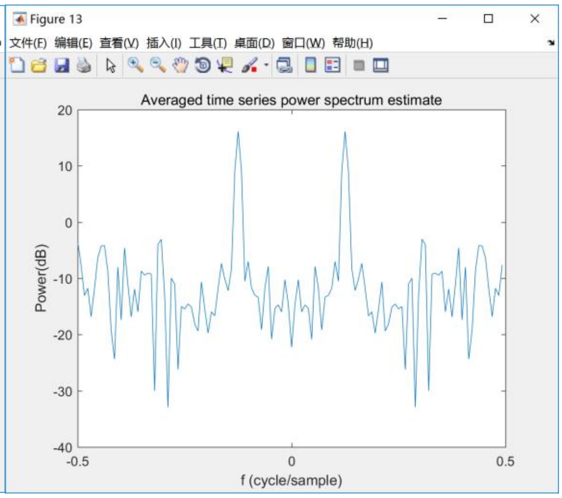

Fig. 9

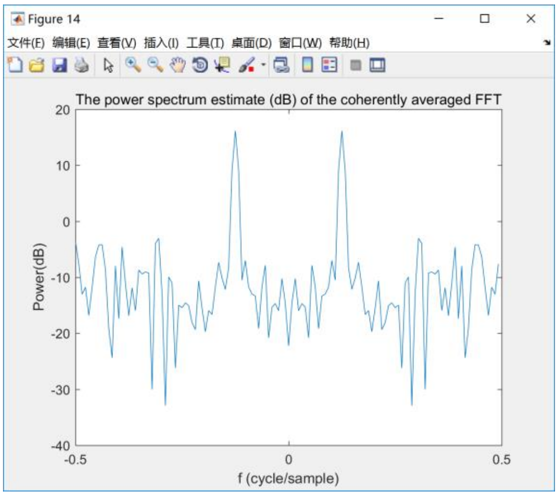

Fig. 10

- Appendix

Script:

```
clear;close all;clc;

f=0.125;
N=1024;
n=0:N-1;
y=sqrt(2)*cos(2*pi*f*n);
rd=randn([1,N]);
sig_o=y+rd;

%% Prob.A
figure(1);
NN=128;
stem(0:NN-1,sig_o(1:NN));
axis([0,NN-1,-5,5]);title('N=128 Time Series ');
xlabel('n');ylabel('x[n]');
%% Prob. B
NFFT_bin=[128,256,512,1024];
for k=1:4
    NFFT=NFFT_bin(k);
    window=hamming(NFFT)';        %% window should be
designed by number NFFT
    wn=sig_o(1:NFFT).*window;
    U=sum(window.^2);
    Pxx=10*log10(abs(fftshift(fft(wn)).^2)/U);
    Noise_mean=mean(Pxx);
```

```
    figure(k+1);
    f=[-0.5:1/NFFT:0.5-1/NFFT];
    plot(f,Pxx);
    xlabel('f
(cycle/sample)');ylabel('Power(dB)');title(['Power
Spectrum Estimate N='  num2str(NFFT)]);
end;
%% Prob. C
    M=128;
    step=[128,64,32];
    num=[8,15,29];
    str=[0,50,75];
    window=hamming(M)';
    U=sum(window.^2);
    for i=1:3
        sum_1=zeros(1,128);
        start=1;
        for j=1:num(i)
            wn=sig_o(start:start+M-1).*window;
            Pxx=10*log10(abs(fftshift(fft(wn)).^2)/U);
            sum_1 = sum_1 + Pxx;
            start = start + step(i);
        end;
        avg = sum_1 / num(i);
        figure(2*i+4);
        f=[-0.5:1/M:0.5-1/M];
        plot(f,avg);
        xlabel('f
(cycle/sample)');ylabel('Power(dB)');title([num2str(str(
i)) '% Overlap Power Spectrum Estimate K='
num2str(num(i))]);

        figure(2*i+5);
        plot(f,avg);
        xlabel('f
(cycle/sample)');ylabel('Power(dB)');title([num2str(str(
i)) '% Overlap Power Spectrum Estimate K='
num2str(num(i))]);
        axis([-0.5 0.5 -8 2]);
    end;
 %% Prob. D
    M=128;
    num=8;
    window=hamming(M)';
```

```
    U=sum(window.^2);
    sum_2=zeros(1,128);
    start=1;
    for j=1:8
        sum_2 = sum_2 + sig_o(start : start + M - 1);
        start = start + M;
    end;
    avg = sum_2 / num;
    figure(12);
    stem(0:M-1,avg);
    axis([0,M-1,-5,5]);title('Averaged Time Series');
xlabel('n');ylabel('x[n]');

    figure(13);
    wn=avg.*window;
    Pxx=10*log10(abs(fftshift(fft(wn)).^2)/U);
    f=[-0.5:1/M:0.5-1/M];
    plot(f,Pxx);
    xlabel('f
(cycle/sample)');ylabel('Power(dB)');title('Averaged
time series power spectrum estimate');

    figure(14);
    M=128;
    num=8;
    window=hamming(M)';
    U=sum(window.^2);
    sum_2=zeros(1,128);
    start=1;
    for j=1:8
        wn=sig_o(start:start+M-1).*window;
        sum_2 = sum_2 + fftshift(fft(wn));
        start = start + M;
    end;
    avg=sum_2/num;
    Pxx=10*log10((abs(avg).^2)/U);
    f=[-0.5:1/M:0.5-1/M];
    plot(f,Pxx);
    xlabel('f
(cycle/sample)');ylabel('Power(dB)');title('The power
spectrum estimate (dB) of the coherently averaged FFT');
```