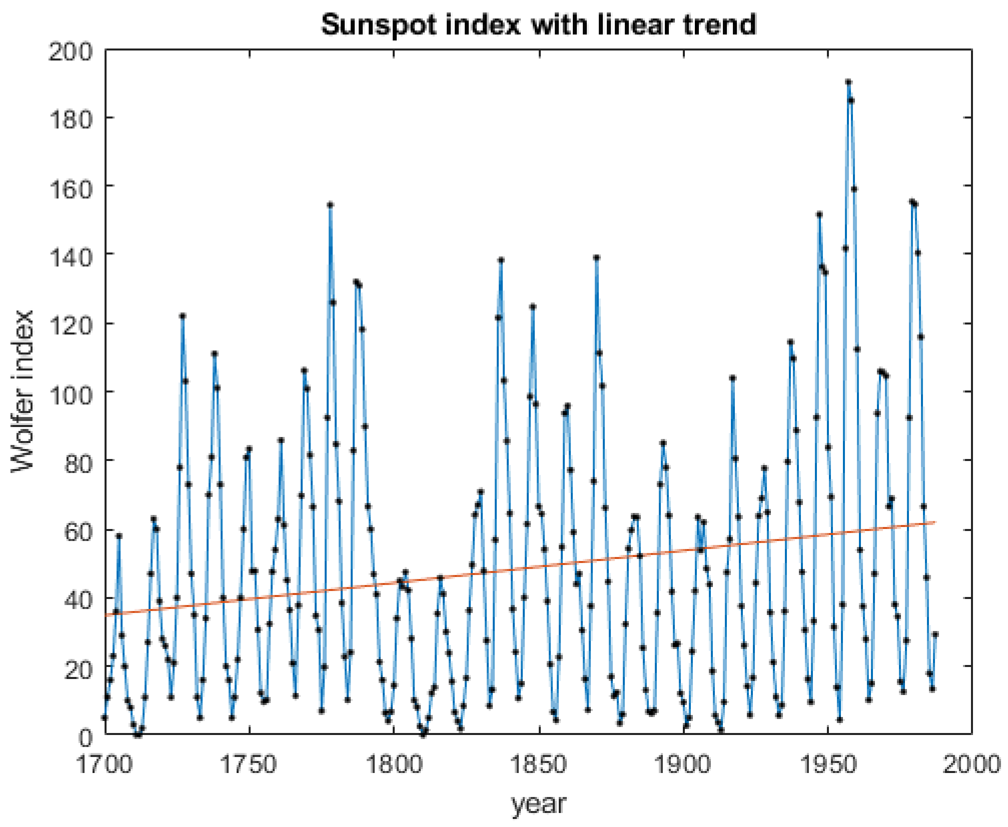
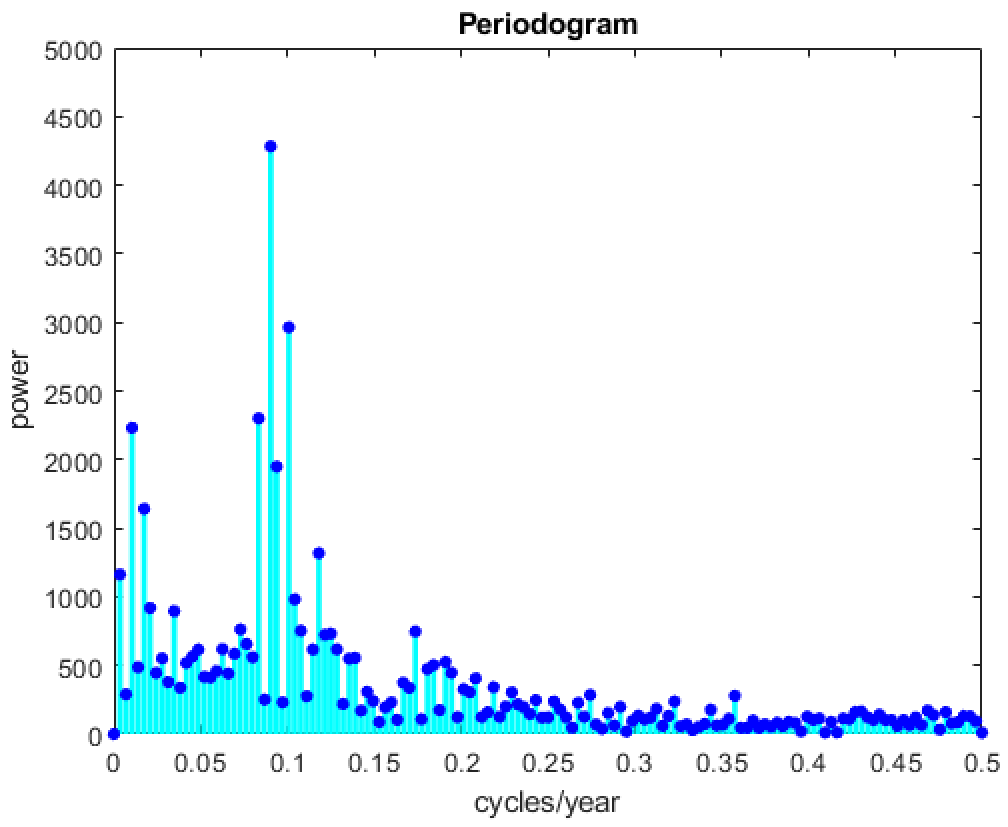
Marwin B. Alejo 2020-20221 EE214_Module5-LabEx1

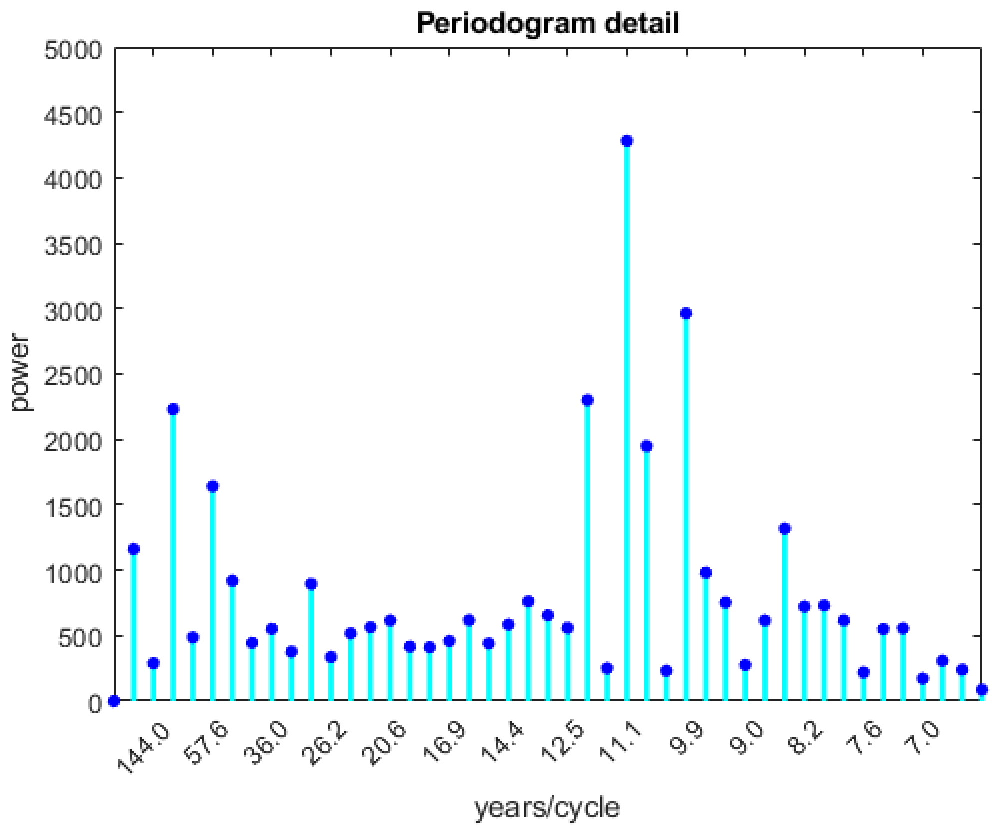
*

I. Sunspot Activity Analysis

```
sunspot_demo
```




Periodogram
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
power
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
cycles/year


Periodogram detail
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
power
144.0
57.6
36.0
26.2
20.6
16.9
14.4
12.5
11.1
9.9
9.0
8.2
7.6
7.0
years/cycle

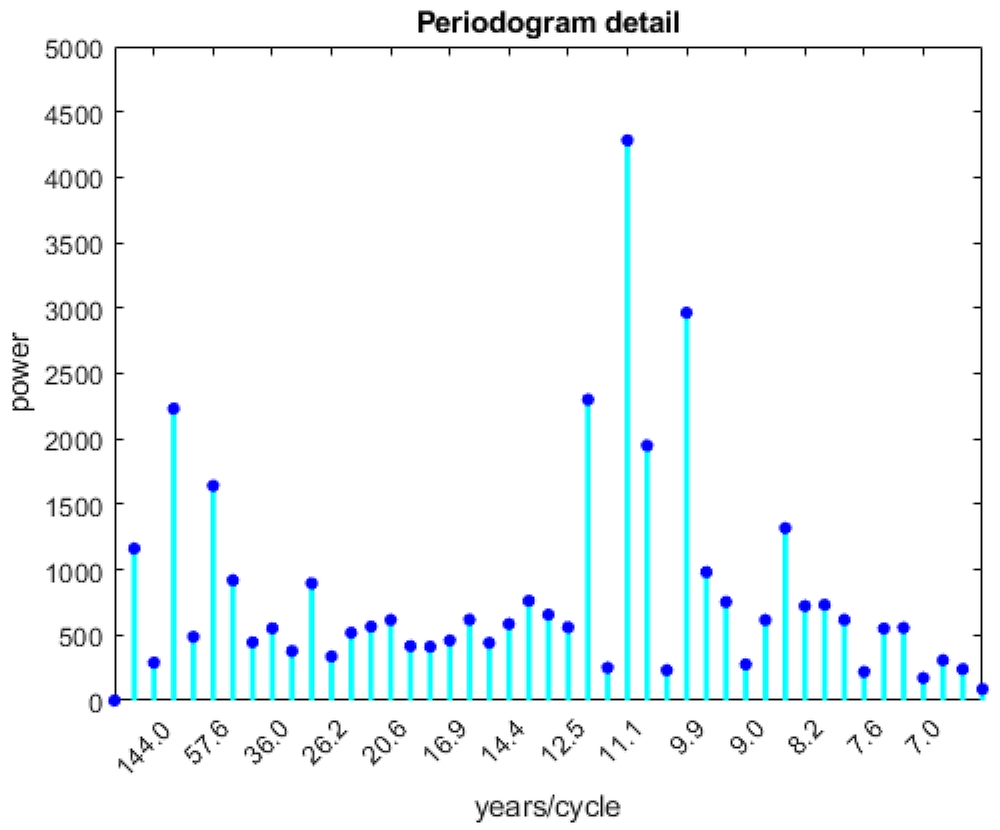
a. Study the code and the comments in the code give your insights on how the spectral analysis was done, how was the cycle period computed?

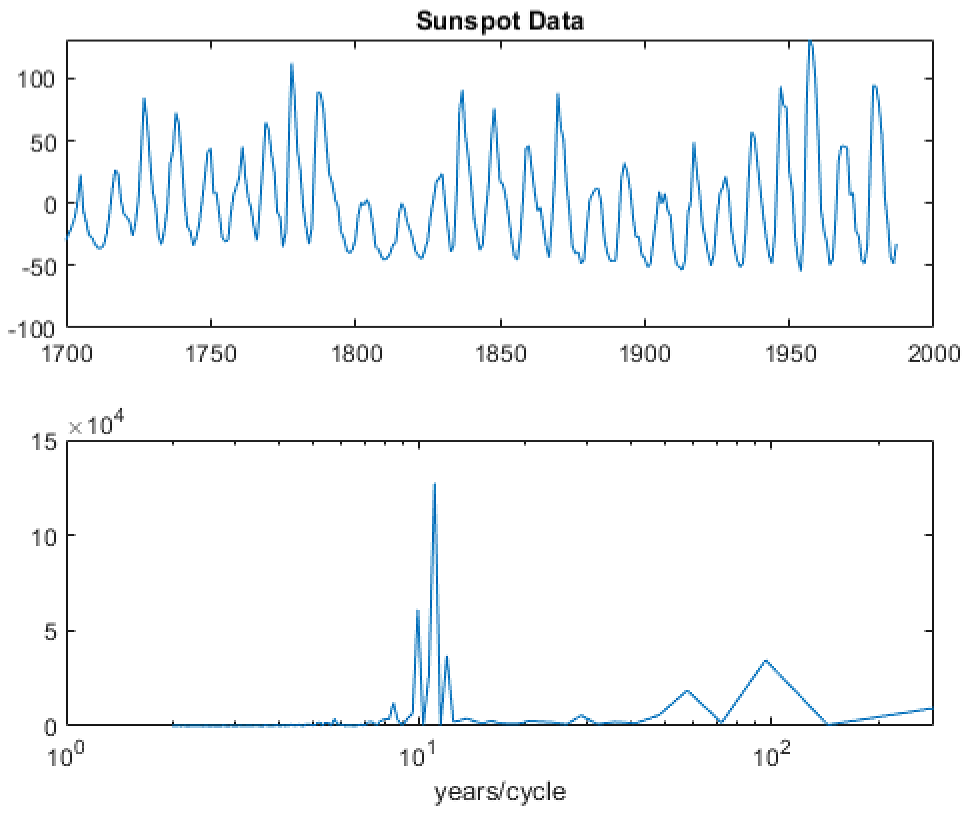
Based on the given MATLAB code, the sunspot cycle period was computed using periodogram through **Fourier transform** or **FFT**. Fourier transform is not only useful in solving partial differential equations but in data analysis of time series data as well. Periodogram on the other hand allows the identification of dominant frequencies or cyclical behavior of a time series data, particularly when cycles happened under an unusual or longer redundant timeframe.

b. Write your own MATLAB code using `periodogram()` to estimate the sunspot activity cycle.

Using **periodogram()**

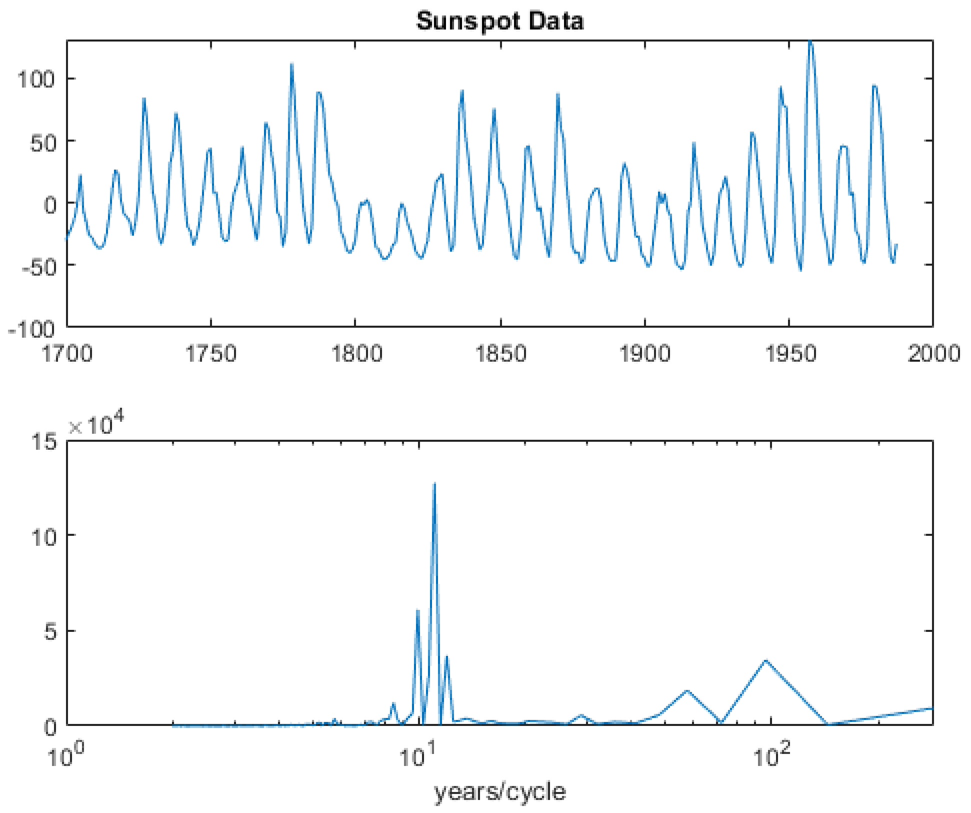
```
load sunspot.dat
year=sunspot(:,1);
data=sunspot(:,2);
data=detrend(data);
figure;
subplot(2,1,1), plot(year,data)
title('Sunspot Data')
nfft=length(data);
[Pxx,f] = periodogram(data, [], nfft, 1);
nn=2:length(f);
subplot(2,1,2), semilogx(1./f(nn), Pxx(nn))
xlabel('years/cycle')
```

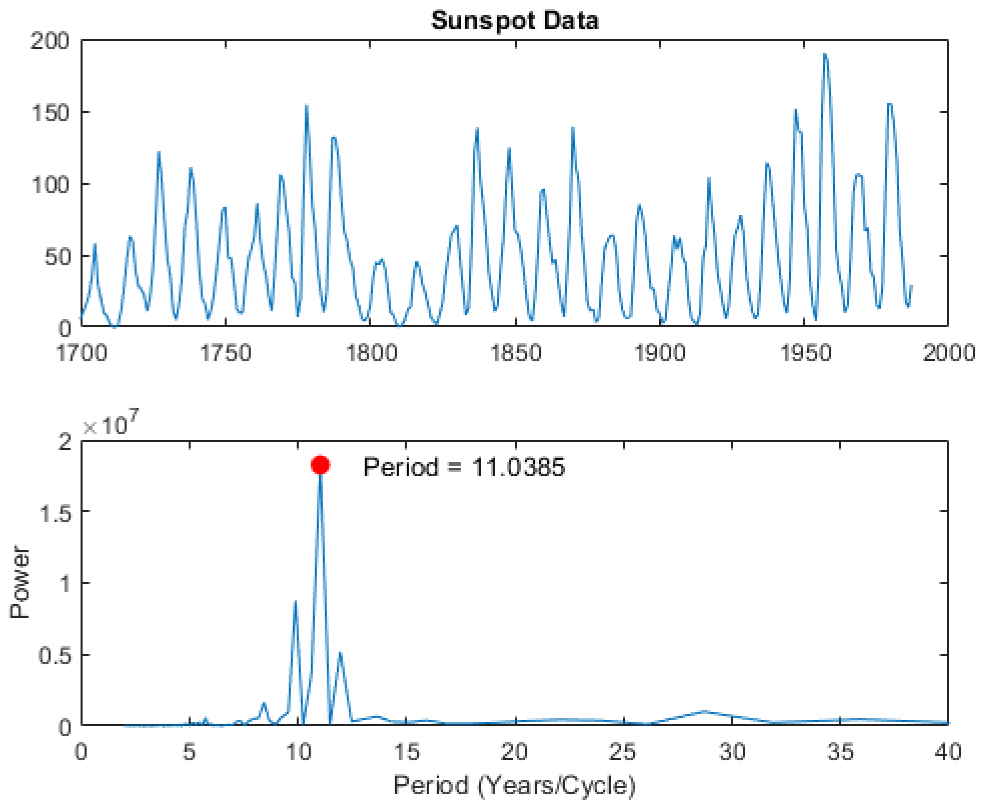




Using **FFT** for checking of the above results.

```
load sunspot.dat
year=sunspot(:,1);
relNums=sunspot(:,2);
figure; subplot(2,1,1); plot(year,relNums)
title('Sunspot Data')
Y = fft(relNums);
Y(1)=[];
n=length(Y);
power = abs(Y(1:floor(n/2))).^2;
nyquist = 1/2;
freq = (1:n/2)/(n/2)*nyquist;
period=1./freq;
subplot(2,1,2); plot(period,power);
axis([0 40 0 2e+7]);
ylabel('Power');
xlabel('Period (Years/Cycle)');
hold on;
index=find(power==max(power));
mainPeriodStr=num2str(period(index));
plot(period(index),power(index),'r.', 'MarkerSize',25);
text(period(index)+2,power(index),['Period = ',mainPeriodStr]);
hold off;
```


Sunspot Data
×10^4
years/cycle


Sunspot Data
×10^7
Period = 11.0385
Power
Period (Years/Cycle)

Observation: As shown above, the first code snippet through the periodogram function shows that the sunspot occurs every 11 years. The second code snippet using FFT function only proves that the generate result of the first snippet are correct. Both of these functions allow to generate the PDS of any time series data except that the later method (`pediogram()`) require only direct vector input of the detrended observed data while `fft()` require the recomputations of the observed data frequency to cyclic period.

Discussion: The power spectral density (PSD) of a WSS random process X(t) is given by the Fourier transform of its autocorrelation function $S_x(f) = \int_{-\infty}^{\infty} R_X(\tau)e^{-j2\pi f\tau}d\tau$

For a discrete-time process X_n, the psd is given by the discrete-time Fourier transform (DTFT) of its autocorrelation sequence $S_x(f) = \sum_{n=-\infty}^{\infty} R_x(n)e^{-j2\pi fn}, \frac{-1}{2} \leq f \leq \frac{1}{2}$

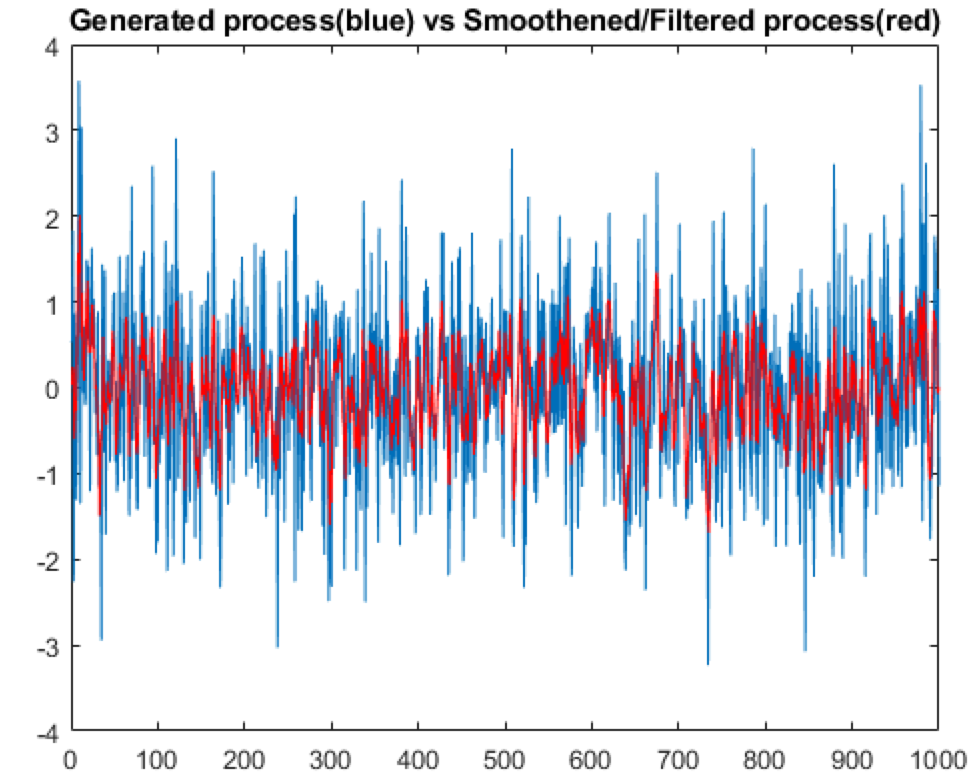
Since the DTFT is periodic in f with period 1, we only need to consider $|f| \leq \frac{1}{2}$

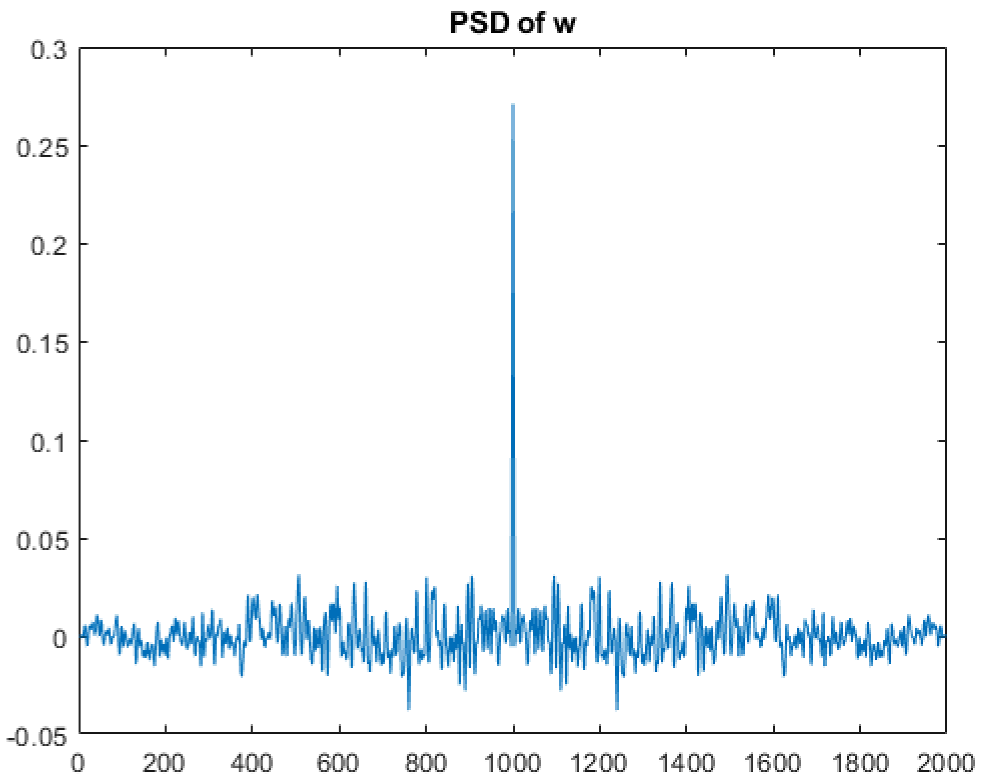
$R_x(\tau)(R_x(n))$ can be recovered from $S_x(f)$ by taking the inverse Fourier Transform $R_X(\tau) = \int_{-\infty}^{\infty} S_X(f)e^{j2\pi f\tau}df, R_X(n) = \int_{\frac{-1}{2}}^{\frac{1}{2}} S_X(f)e^{j2\pi fn}df$

II. Noise and LTI Systems

a. We have shown previously that the process is stationary and ergodic but non-correlated, we will not pass this process to a LTI system, a simple low pass filter.

```
x = randn(1000,1); %generate 1000point gaussian dist
w = conv(x,ones(1,4)/4,'same'); %smoothening filter with mentioned IR in the
 manual
figure; plot(x); hold on; plot(w,'r'); hold off;
title('Generated process(blue) vs Smoothened/Filtered process(red)');
figure; plot(xcorr(w,'biased')); title('PSD of w');
```


Generated process(blue) vs Smoothened/Filtered process(red)
4
3
2
1
0
-1
-2
-3
-4
0
100
200
300
400
500
600
700
800
900
1000


PSD of w
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
200
400
600
800
1000
1200
1400
1600
1800
2000

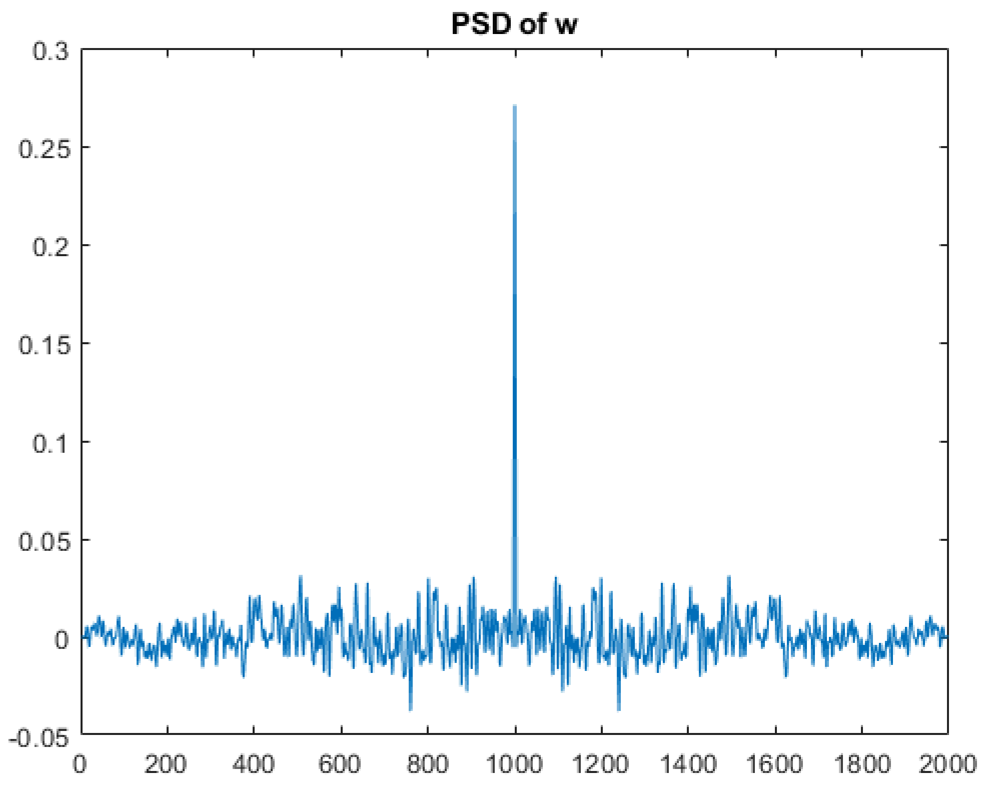
Observation: `randn()` allows the generation of gaussian process of length 1000. The `conv()` function allows the application of a simple LP filter to smoothen the generated gaussian process by [0.25 0.25 0.25 0.25] as given in the manual. As shown in the figure, the filtered signal (red) has smoother time variations when autocorrelation was introduced by the simple LPF.

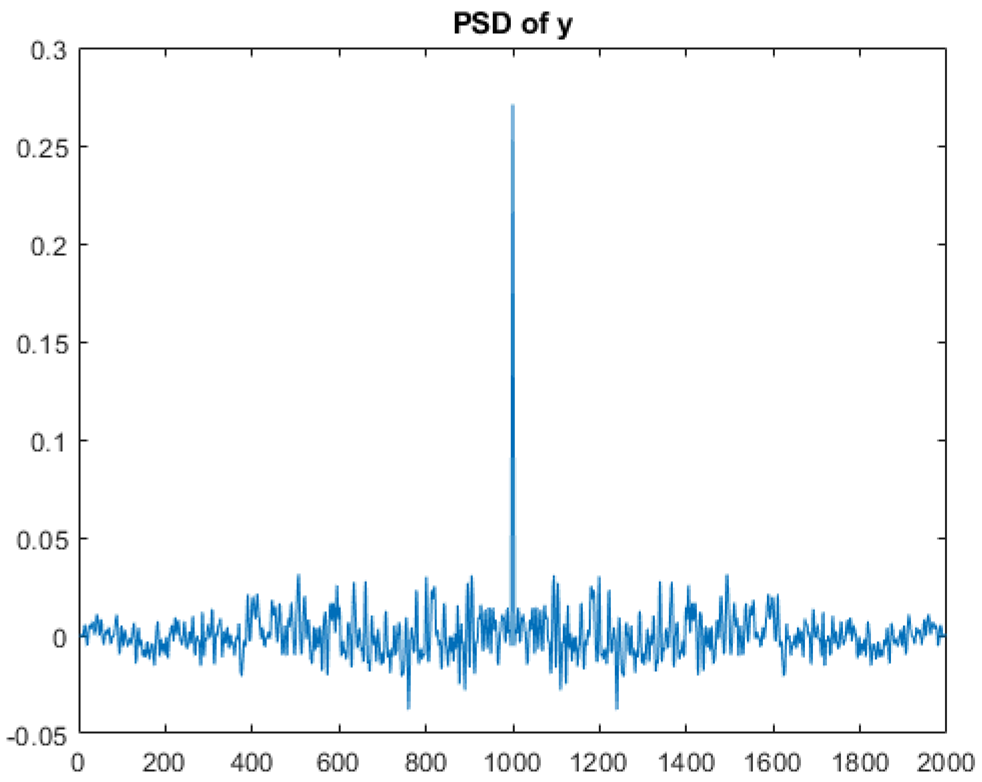
b. Predict the resulting autocorrelation of the output process y(t). Is the output still ergodic? Is it WSS? Explain your answer.

In theory, the filtered gaussian process is **WSS** since its mean is approximately similar to the mean of the original gaussian process; consider `mean(x,1)` and `mean(w,1)`. Moreover, the filtered gaussian process is surely **ergodic** given the fact that there is only one function process exsiting in the time space.'

c. To verify the properties of the output process y(t), implement the resulting filter with x(t) s input using the matlab command: `y = filter([0.25 0.25 0.25 0.25], 1, x);` where x is a vector containing a sample function of the process x(t). Estimate the autocorrelation $Ry(\tau)$ using `xcorr()`, i.e. `ry = xcorr(y, biased')`. Does the output autocorrelation agree with theory?

```
y = filter(ones(1,4)/4, 1, x);
figure; plot(xcorr(w,'biased')); title('PSD of y');
```

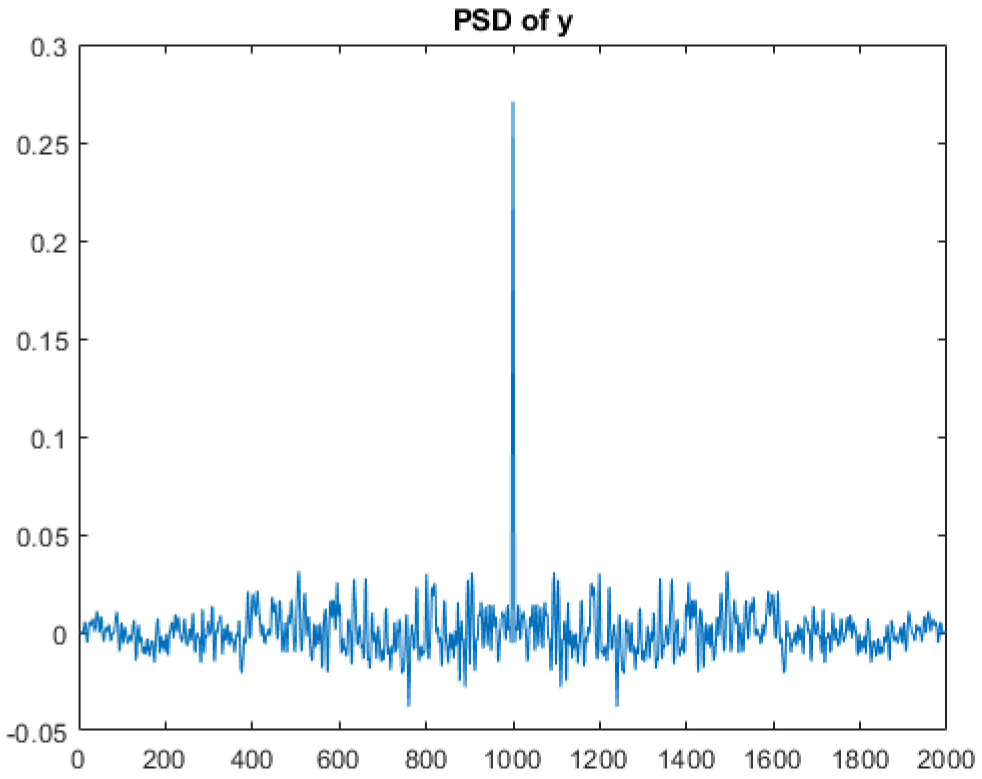


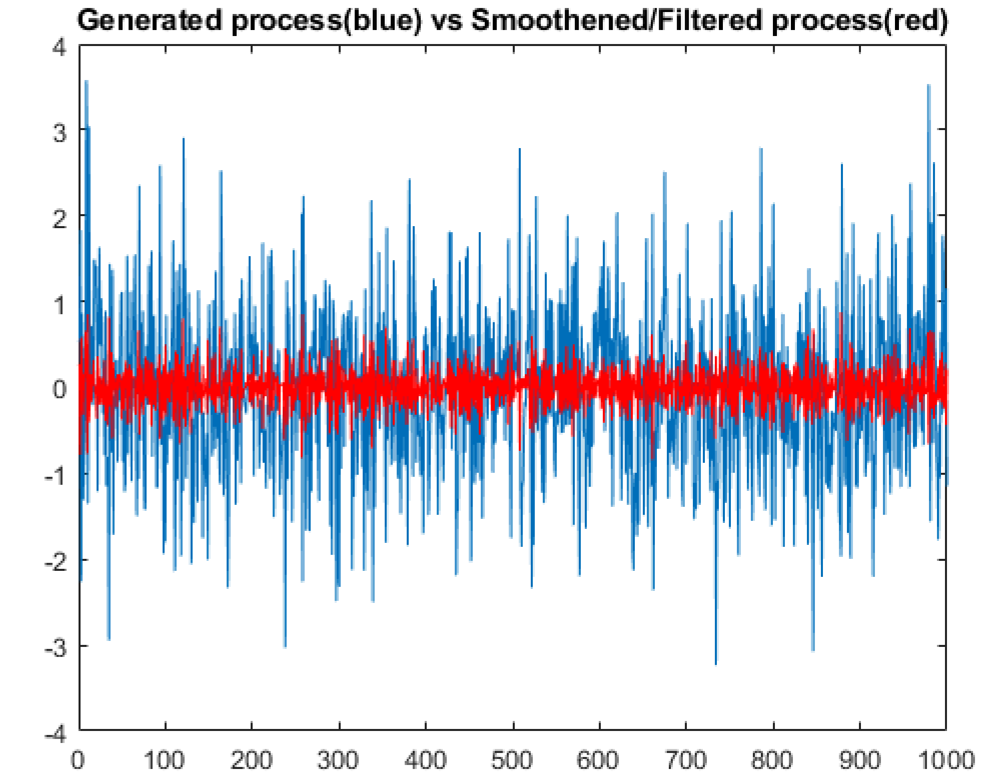


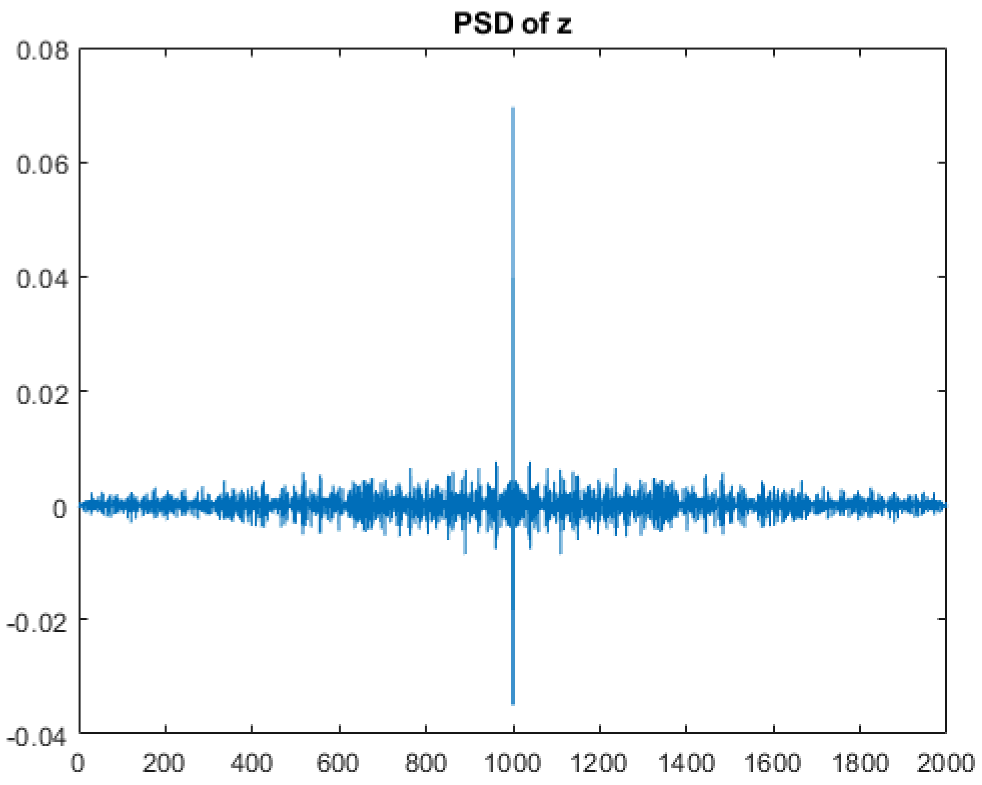
Based on the generated results, the experimental outcome of 1.c agreed with the theoretical outcome of 1.b.

d. Let s use another filter defined by the following equation `y(t) = 0.9y(t-1) + 0.2x(t)` Suppose we x(t) as input to this filter and predict the autocorrelation of the output process. Is the output still ergodic? Is it WSS? Justify your answer.

```
z = conv(0.2.*x,[1 -0.9],'same'); %smoothening filter with mentioned IR in the
 manual
figure; plot(x); hold on; plot(z,'r'); hold off;
title('Generated process(blue) vs Smoothened/Filtered process(red)');
figure; plot(xcorr(z,'biased')); title('PSD of z');
```


PSD of y
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
200
400
600
800
1000
1200
1400
1600
1800
2000

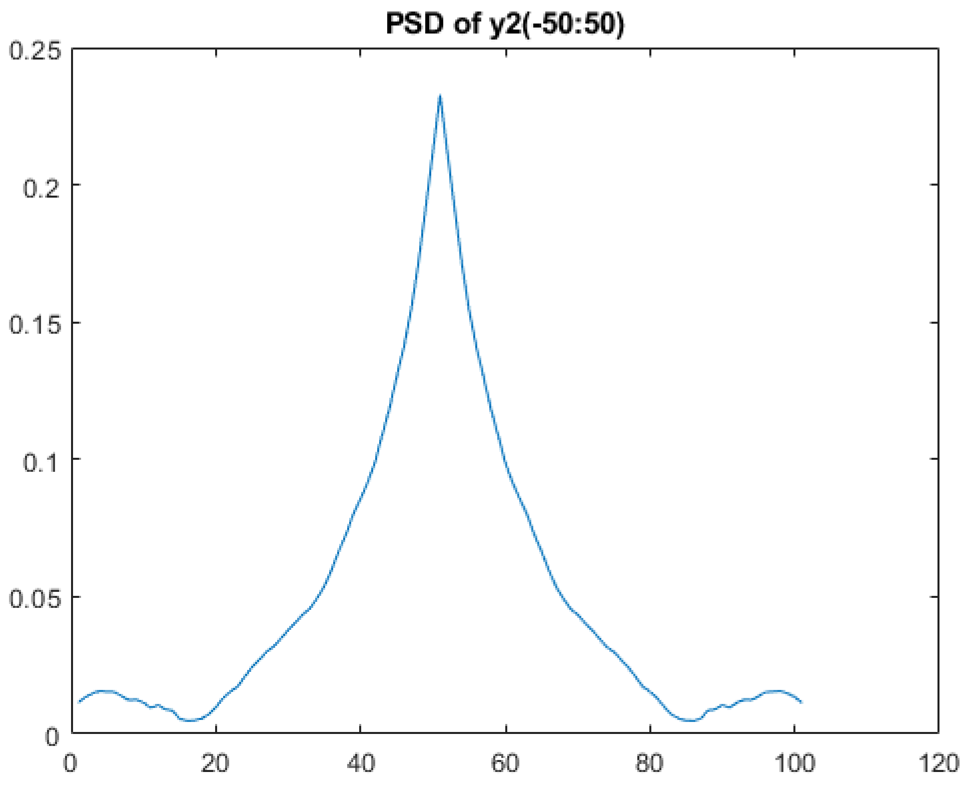

Generated process(blue) vs Smoothened/Filtered process(red)
4
3
2
1
0
-1
-2
-3
-4
0
100
200
300
400
500
600
700
800
900
1000



In theory, given the IR `[1 -0.9]`, the filtered gaussian process 'x' is surely an ergodic process given the fact that there is only one process in the time space but it is not WSS given the fact that the mean of the filtered gaussian process is not the same as with the mean of the original gaussian process.

e. To compute the output of the above filter in Matlab, we have: `y = filter(0.2, [1 -0.9], x);` where x is a vector containing a sample function of the process x(t). Estimate the autocorrelation $Ry(\tau)$ using `xcorr()`, i.e. `ry = xcorr(y, 'biased')`. Plot and examine Ry(\tau)$ for $-50 \leq \tau \leq 50$. Does it agree with the theory?

```
y2 = filter(0.2, [1 -0.9], x);
cor = xcorr(y2,'biased');
figure; plot(cor(950:1050)); title('PSD of y2(-50:50)');
```

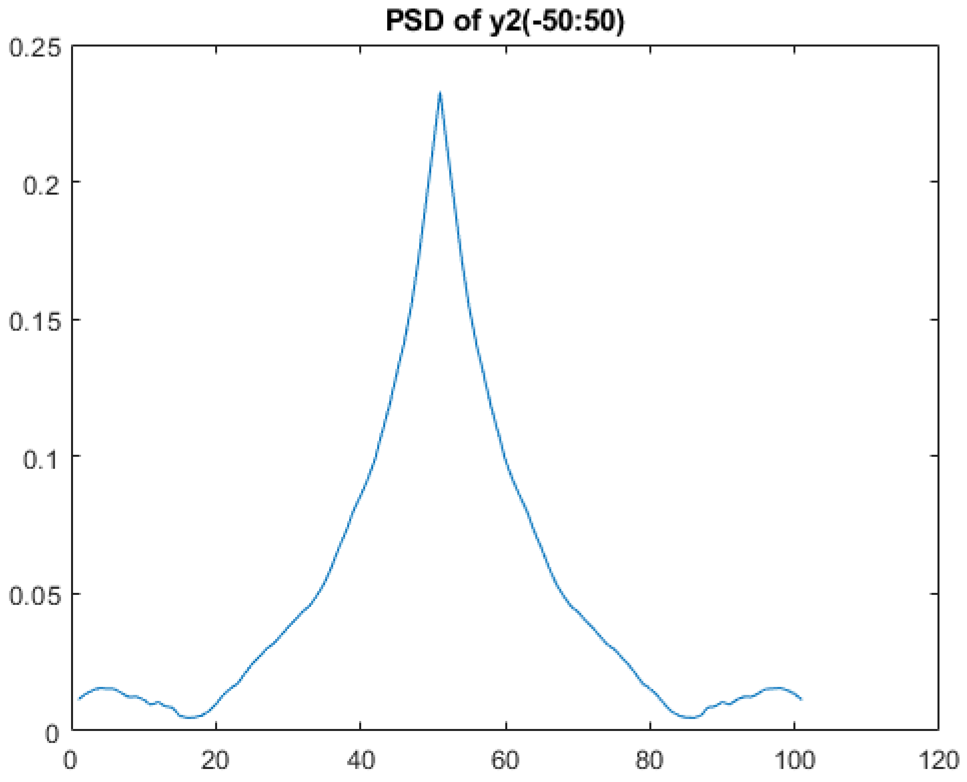


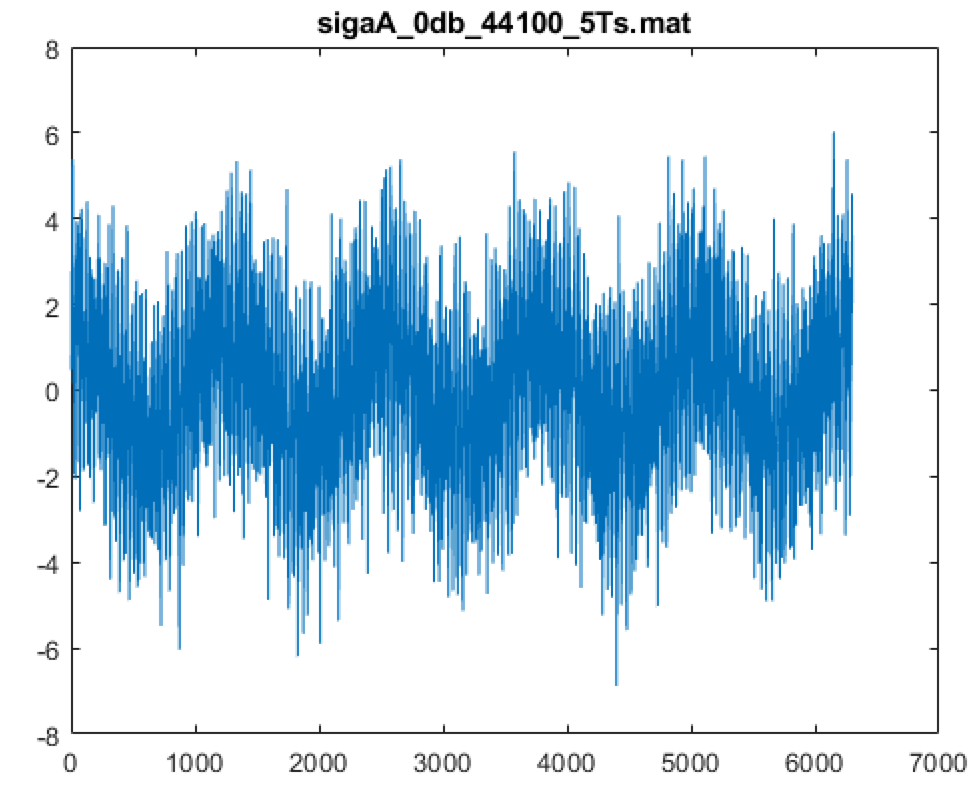
Given the results and the experimental results in 1.e does not suffice or agreed with those of in 1.d.

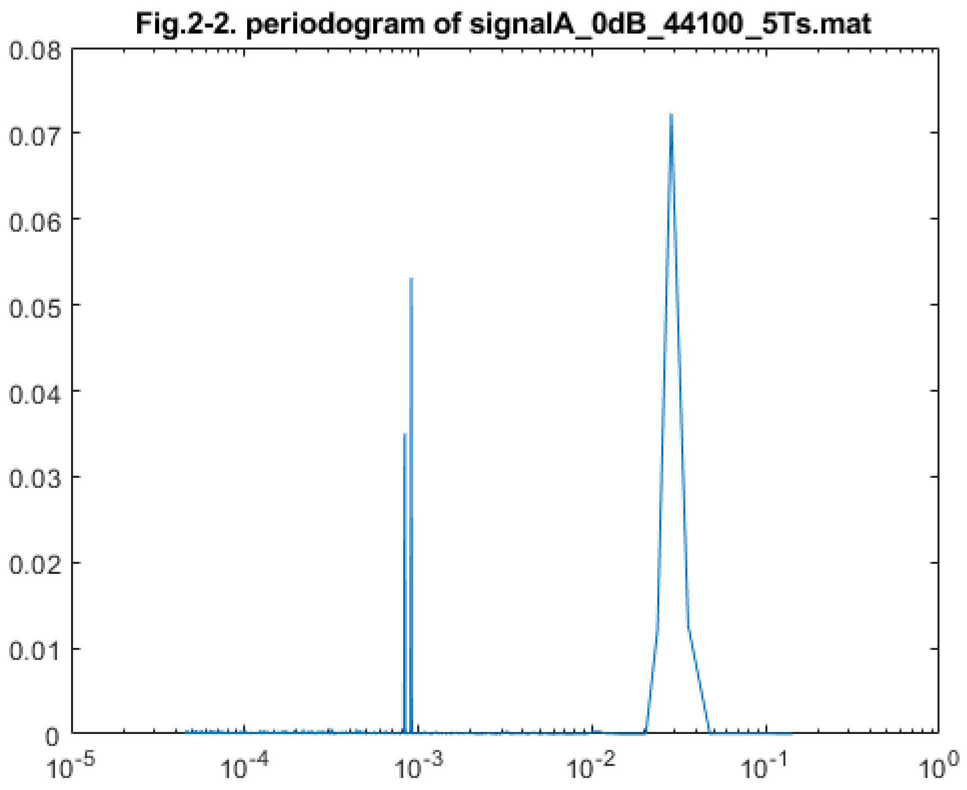
II. Spectral Analysis of Signals with Noise

1. Start with the signal with 0 dB noise level and sampling rate at 44100, perform spectral analysis using periodogram. What parameters of the periodogram did you specify (number of FFT points? averaging? one-sided?, ect)?

```
clear;clc;
load('signalA_0dB_44100_5Ts.mat')
data = ytn;
data=detrend(data);
figure; plot(data); title('Fig.2-1. signalA\_0dB\_22000\_5Ts.mat');
title('sigaA\_0db\_44100\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 44100);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-2. periodogram of signalA
\_0dB\_44100\_5Ts.mat');
```


PSD of y2(-50:50)
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120


sigaA_0db_44100_5Ts.mat
8
6
4
2
0
-2
-4
-6
-8
0
1000
2000
3000
4000
5000
6000
7000



a. Briefly justify the parameters you have chosen for the periodogram.

This experiment uses the `periodogram()` function of MATLAB to easily perform the spectral analysis. By default, this function uses the Hamming windowing function due to its accurate effect of the spectrum of the signal. Assuming that all discrete points of the signal is affected by the noise (although 0db at this sample), this experiment considered using the same length of the signal as the number of FFT points leading to suppressing the noise. Furthermore, detrending was performed (although not everytime) to ensure that the spectral analysis is focused on the hidden signal of the given process. The Fs input is equivalent to the FS values indicated in the table of the manual, which is 44100 in this case.

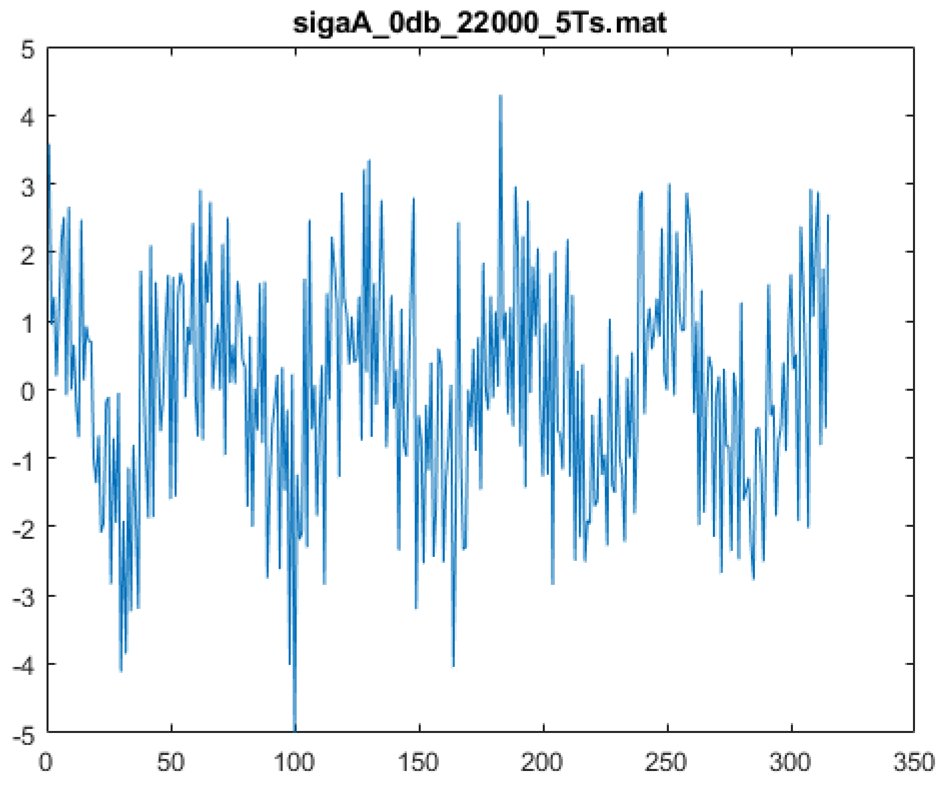
b. Were you able to determine the frequencies of the signals buried in noise? What are those frequencies and relative amplitudes?

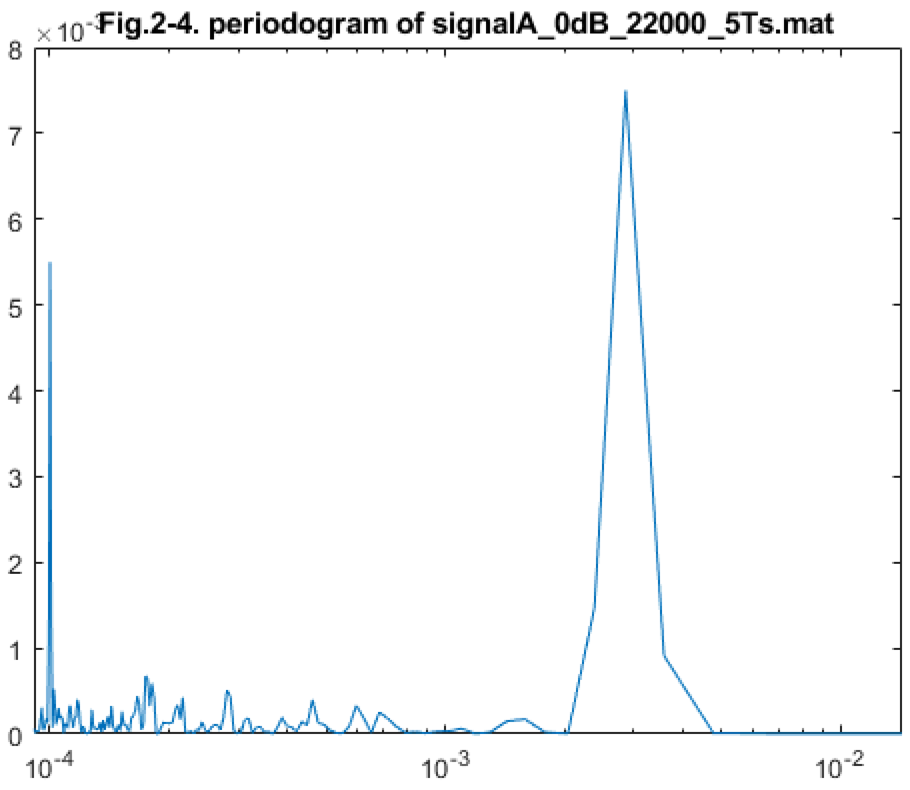
Using the code snippet in 1., it is evident that this experiment was able to determine the frequency and relative information of the signal which are shown in figure 2-2.

2. Repeat your spectral analysis the same signal with 0 dB but with a different sampling rate and perform spectral analysis.

```
clear;clc;
load('signalA_0dB_22000_5Ts.mat')
data = ytn;
data=detrend(data);
figure; plot(data); title('Fig.2-3. signalA\_0dB\_22000\_5Ts.mat');
title('sigaA\_0db\_22000\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 22000);
```

```
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-4. periodogram of signalA
\_0dB\_22000\_5Ts.mat');
```





a. What parameters did you use? Are the signal frequencies still discernible?

Using the same the parameters used in 1.a-b of this section (II) such as the length of the used process and the defined Fs of it in the manual, the smoothened signal is discernable after undergoing detrending and the FFT function of the `periodogram()` function although it is not finer as those in figure 2-2. Note that hamming windowing function is used on the process of estimating the PSD of the used signal to accurately analyze the spectrum of the signal. Figure 2-4 shows the periodogram of the signal with 22000 Fs.

b. What is the effect of a lower sampling rate on the spectral analysis?

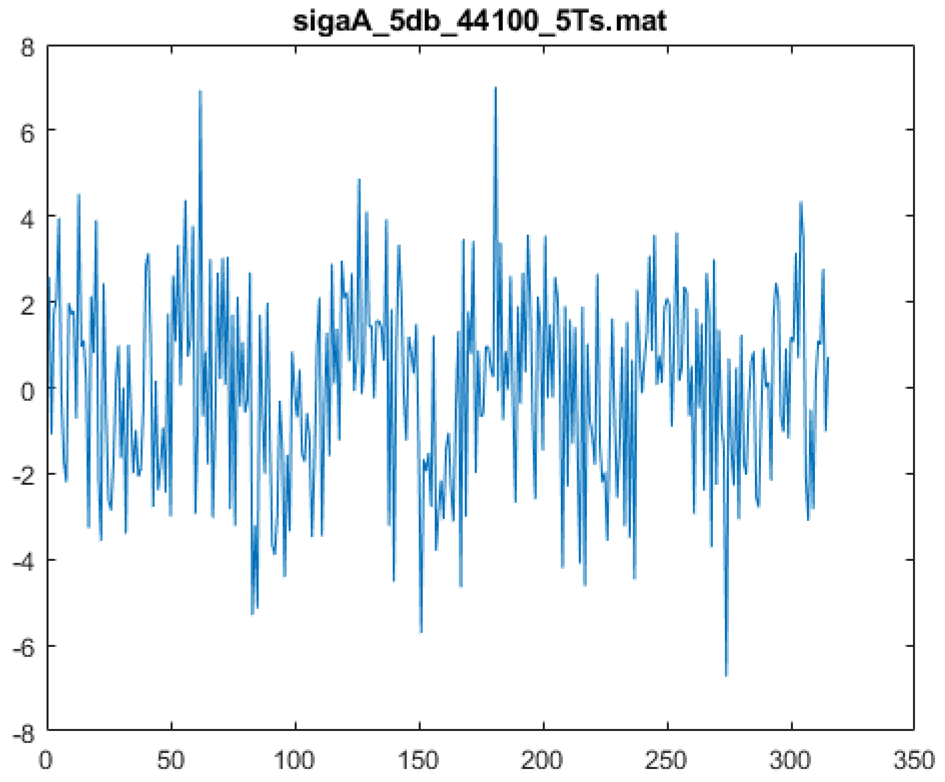
In theory, lower sampling rate might yield or contribute to the probability of having an aliased signal or a signal that was improperly sampled which might affect in estimating the PSD of the signal. Hence, spikes of noise might be present on the PSD estimate of the signal when it's Fs is lower or improper for its spectrum.

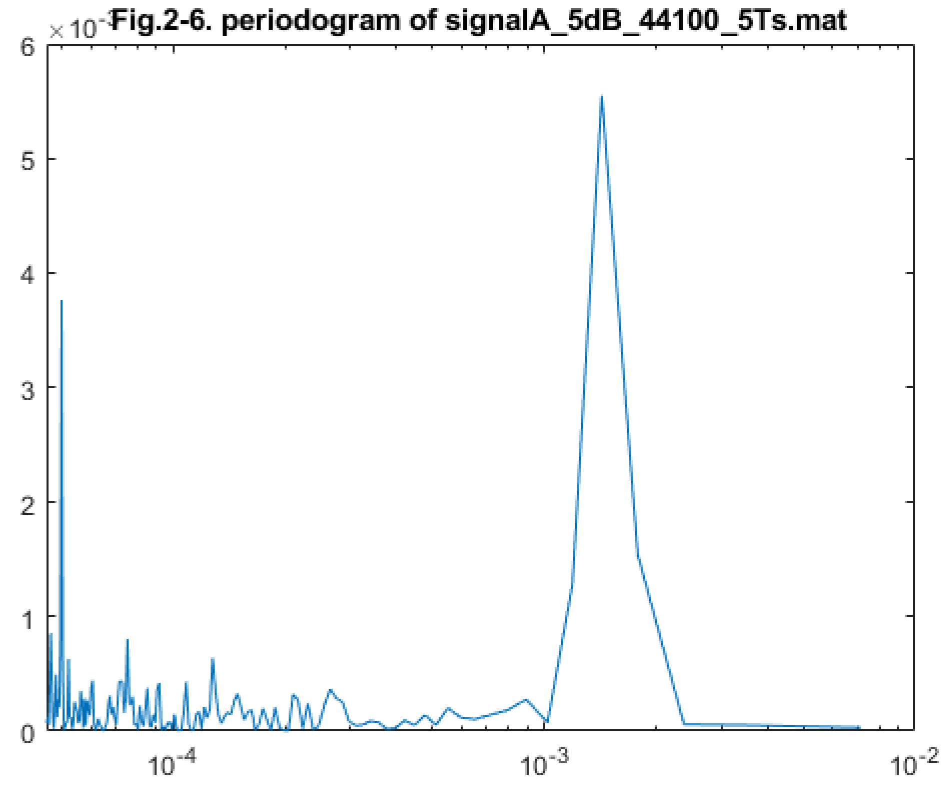
3. Repeat the process of doing the spectral analysis on the other signals with lower (more negative) SNR and at different sampling rates.

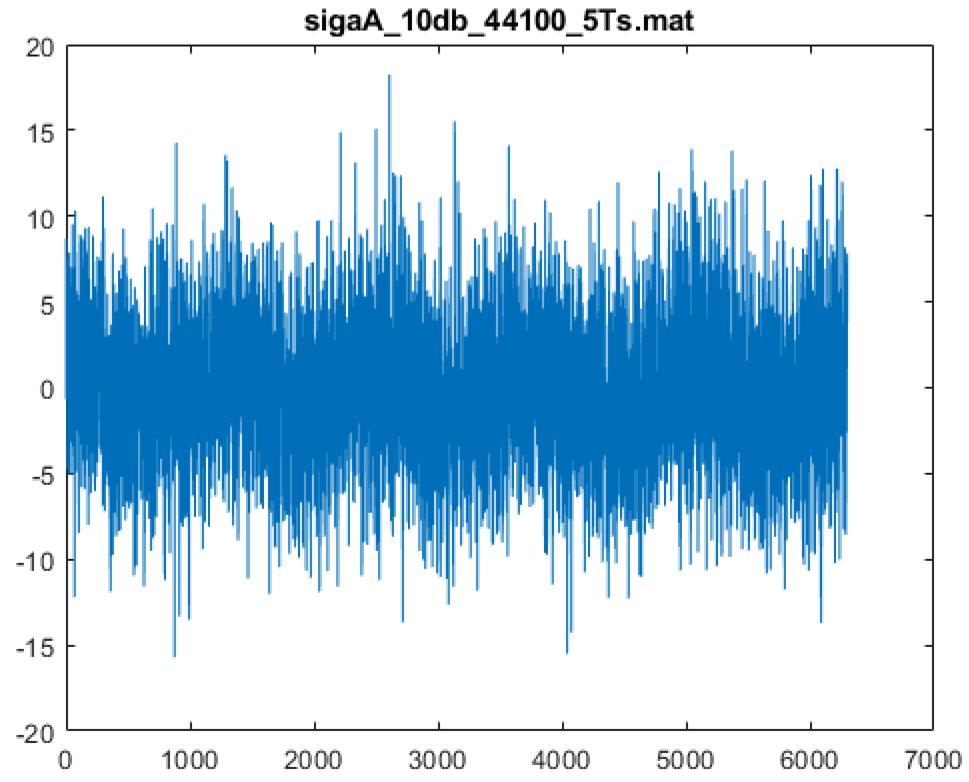
a. Can you still determine the signal frequencies and relative amplitudes at -5 dB? at -10 dB

```
clear;clc;
load('signalA_5dB_44100_5Ts.mat')
data = ytn;
data=detrend(data);
figure; plot(data); title('Fig.2-5. signalA\_5dB\_44100\_5Ts.mat');
title('sigaA\_5db\_44100\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 44100);
nn=2:length(f);
```

```
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-6. periodogram of signalA
\_5dB\_44100\_5Ts.mat');

clear;clc;
load('signalA_10dB_44100_5Ts.mat')
data = ytn;
data=detrend(data);
figure; plot(data); title('Fig.2-7. signalA\_10dB\_44100\_5Ts.mat');
title('sigaA\_10db\_44100\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 44100);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-8. periodogram of signalA
\_10dB\_44100\_5Ts.mat');

clear;clc;
load('signalA_5dB_22000_5Ts.mat')
data = ytn;
data=detrend(data);
figure; plot(data); title('Fig.2-9. signalA\_5dB\_22000\_5Ts.mat');
title('sigaA\_5db\_22000\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 22000);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-10. periodogram of signalA
\_5dB\_22000\_5Ts.mat');

clear;clc;
load('signalA_10dB_22000_5Ts.mat')
data = ytn;
data=detrend(data);
figure; plot(data); title('Fig.2-11. signalA\_10dB\_22000\_5Ts.mat');
title('sigaA\_10db\_22000\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 22000);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-12. periodogram of signalA
\_10dB\_22000\_5Ts.mat');
```


sigaA_5db_44100_5Ts.mat
8
6
4
2
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350


× 10-3
Fig.2-6. periodogram of signalA_5dB_44100_5Ts.mat
6
5
4
3
2
1
0
10-4
10-3
10-2


sigaA_10db_44100_5Ts.mat
20
15
10
5
0
-5
-10
-15
-20
0
1000
2000
3000
4000
5000
6000
7000

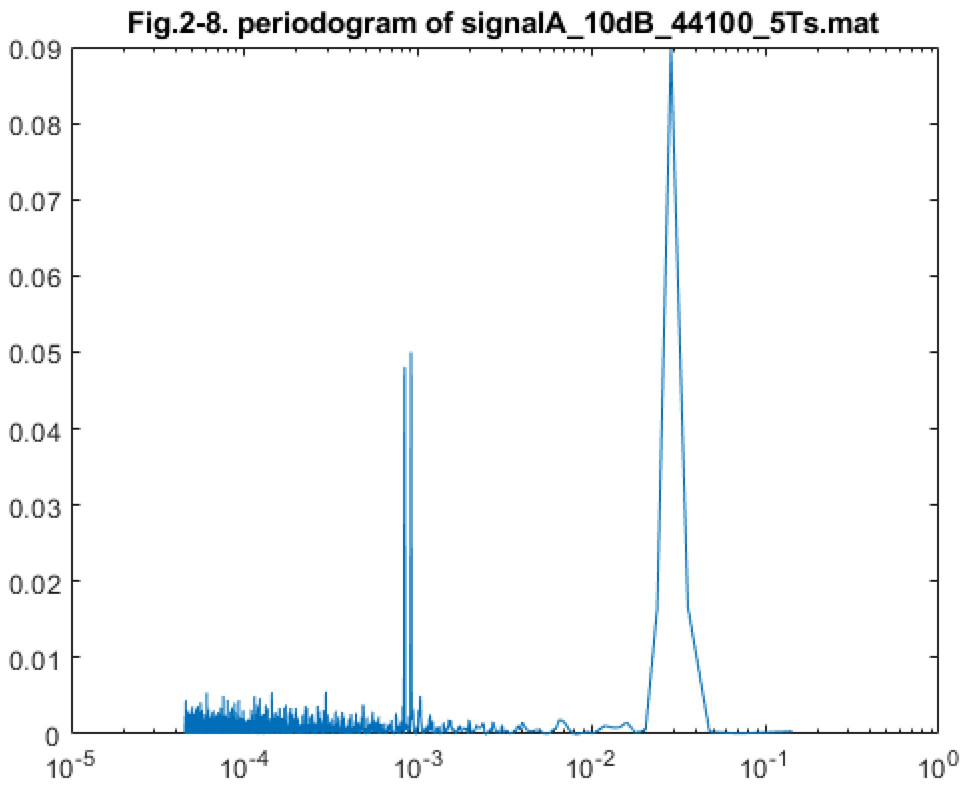
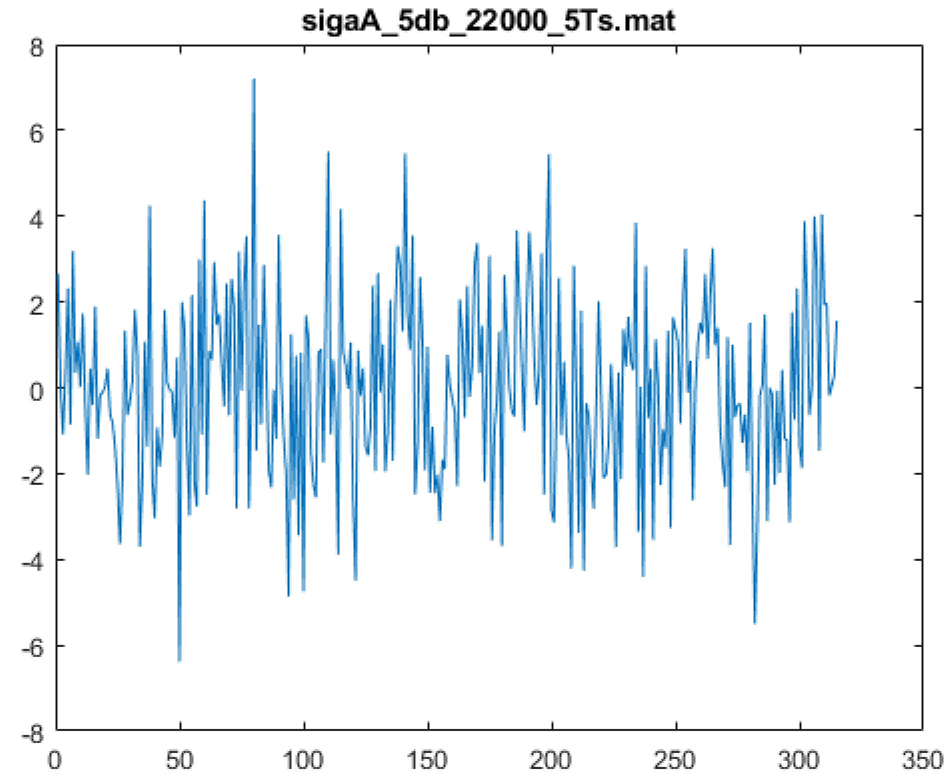


Fig.2-8. periodogram of signalA_10dB_44100_5Ts.mat
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}



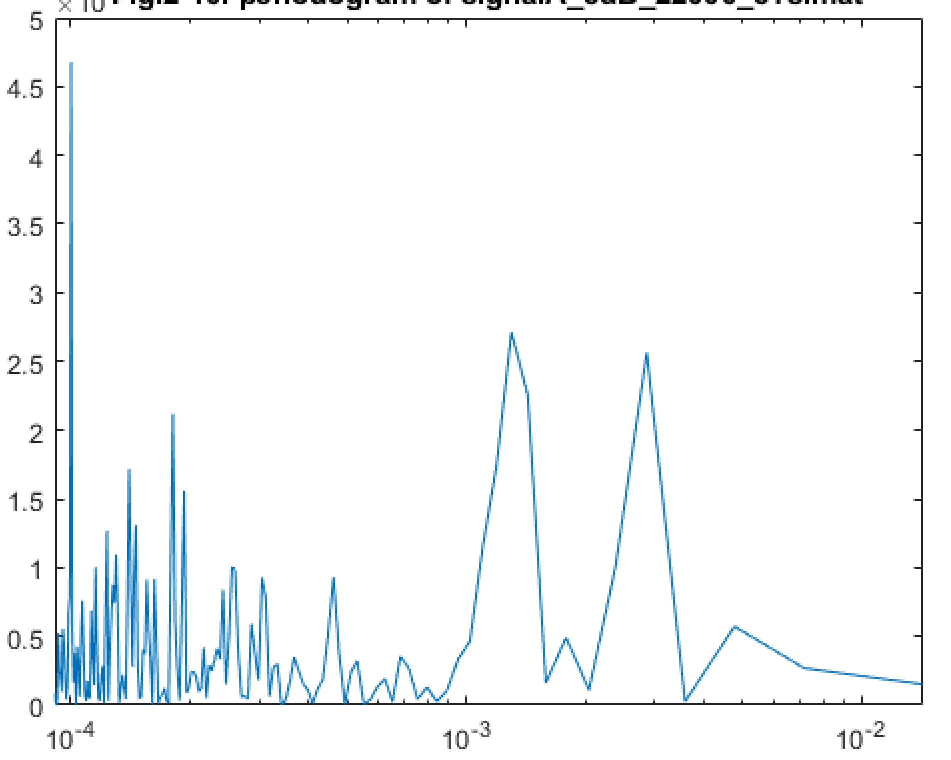
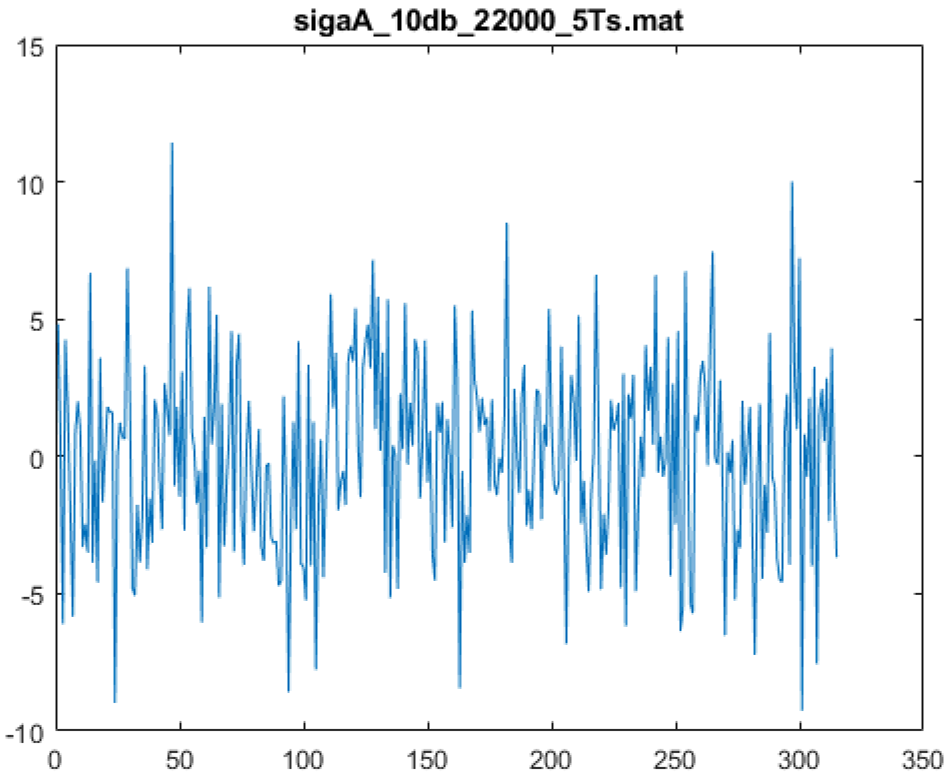


Fig.2-10. periodogram of signalA_5dB_22000_5Ts.mat


sigaA_10db_22000_5Ts.mat
15
10
5
0
-5
-10
0
50
100
150
200
250
300
350

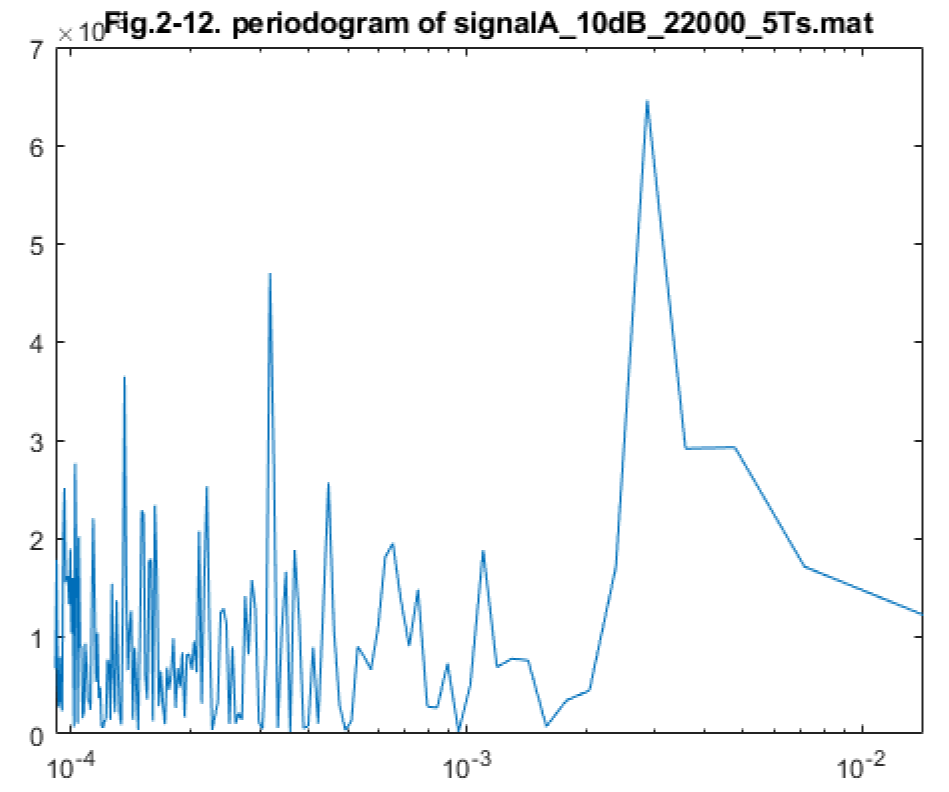


Fig.2-12. periodogram of signalA_10dB_22000_5Ts.mat
×10
7
6
5
4
3
2
1
0
10^{-4}
10^{-3}
10^{-2}

a. Can you still determine the signal frequencies and relative amplitudes at -5 dB? at -10 dB?

Given the results above using the provided signals on selected configurations, the frequencies and other information for signals whose Fs is 44100 regardless of the noise level are still discernible while those signals with 22000 Fs and with -5|-10dB noise are no longer discernible. These results subtlely prove the theory in 2.b of this section (II) that signals immersed with noise but are sampled lower than the requirement of its spectrum affect the estimation of its PSD. Thus, for outputs as shown in figures 2-10 and 2-12, the actual PSD were overridden by noise while the outputs as shown in figures 2-6 and 2-8 although few spikes of noise are tolerable and does not completely affect the actual power of the signal.

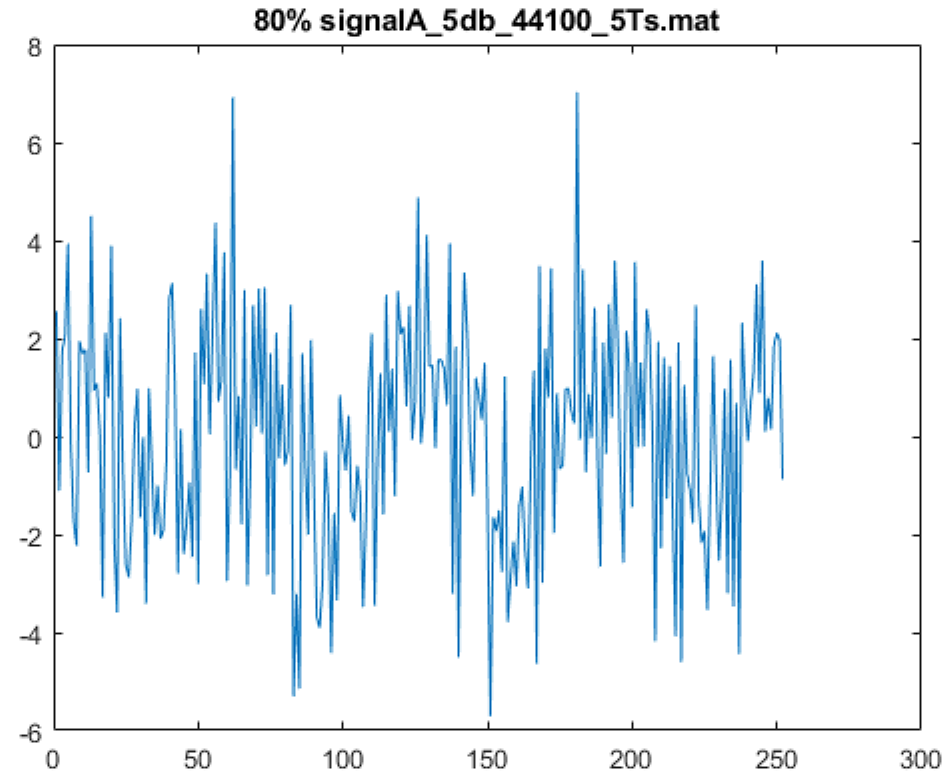
b. What is the effect of increasing noise level and sampling rate on the ability to discern signal frequencies?

As mentioned in 3.a of this section (II), aside from **aliased signal** , it misleads the actual PSD estimation of the signal by overiding the actual information *(see figures 2-10 and 2-12)*.

4. Assuming we do not have enough data record about the signal due to a shorter observation period, we can simulate this by truncating part of the signal and analysing the rest.

a. For the signals at -5 dB noise level, at sampling rate of 44100, truncate the signal and retain only 80% of its original length, or set the values of the signals to zero on the truncated part. Perform spectral analysis on the truncated signal. Can you still discern the signal frequencies and relative amplitudes?

```
clear;clc;
load('signalA_5dB_44100_5Ts.mat')
data = ytn;
data_len = length(data)*0.8; % new length by 80% of orig data
new_data = data(1:data_len); % 20% trimmed data, new data with 80% of orig
data=detrend(new_data);
figure; plot(data); title('Fig.2-13. signalA\_5dB\_44100\_5Ts.mat 80%');
title('80% signalA\_5db\_44100\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 44100);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-14. periodogram of signalA
\_5dB\_44100\_5Ts.mat 80%');
```



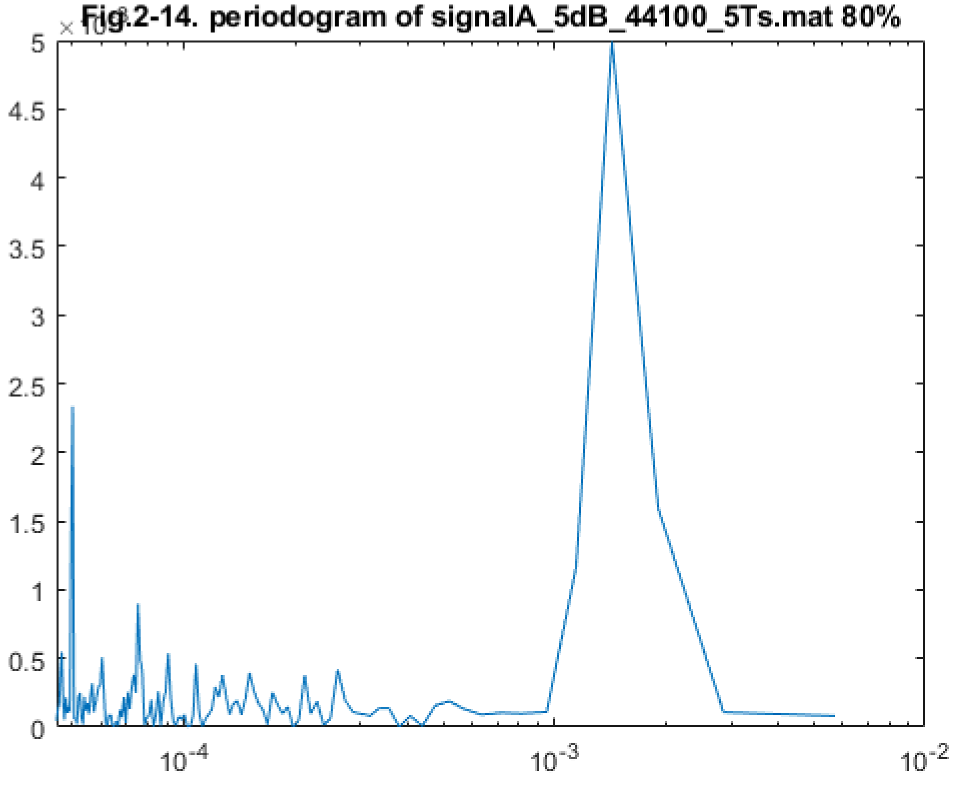
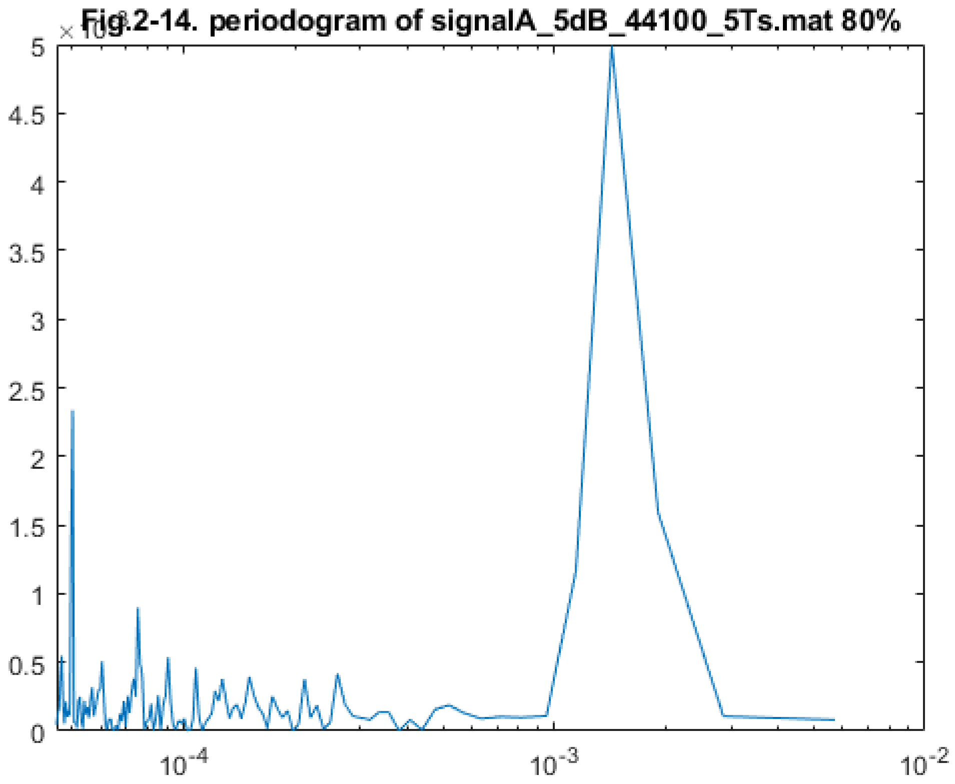


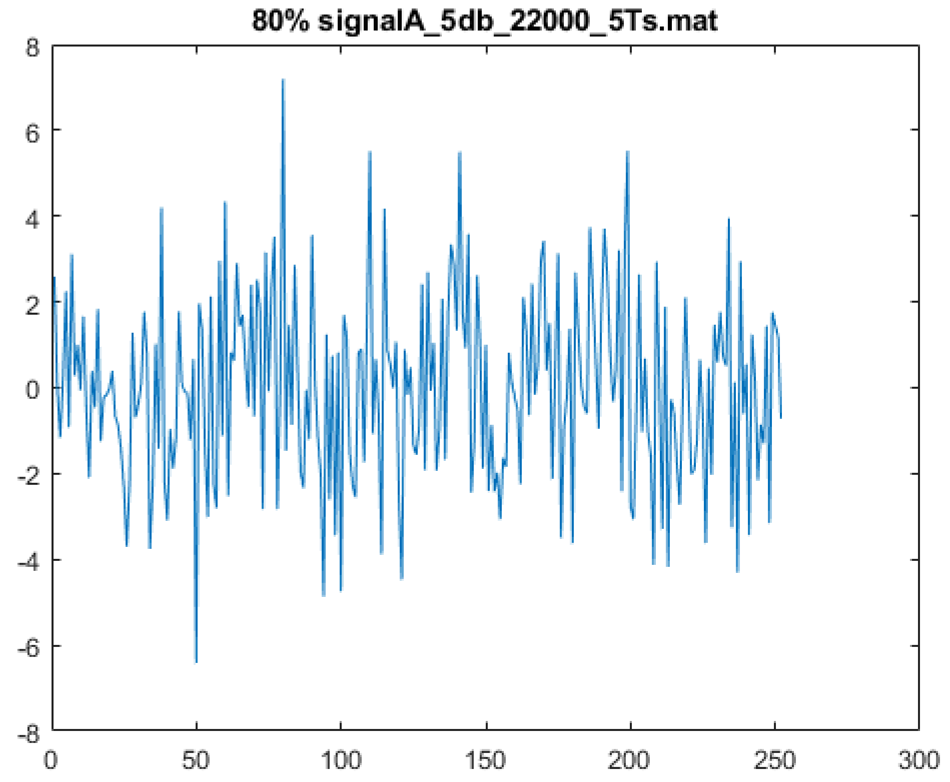
Fig.2-14. periodogram of signalA_5dB_44100_5Ts.mat 80%

Given the results in 4.a. and the required data configuration, the signal frequency may either be discernable or not depending on which part of the signal was truncated. However, the signal frequency when the last 20% of the data was truncated is still discernable.

b. Truncate also the signal at -5 dB noise level and sampling rate of 22000 to 80% of its original length. Perform spectral analysis and see if you can still determine the spectral characteristics of the signals.

```
clear;clc;
load('signalA_5dB_22000_5Ts.mat')
data = ytn;
data_len = length(data)*0.8; % new length by 80% of orig data
new_data = data(1:data_len); % 20% trimmed data, new data with 80% of orig
data=detrend(new_data);
figure; plot(data); title('Fig.2-15. signalA\_5dB\_22000\_5Ts.mat 80%');
title('80% signalA\_5db\_22000\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 22000);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-16. periodogram of signalA
\_5dB\_22000\_5Ts.mat 80%');
```




80% signalA_5db_22000_5Ts.mat
8
6
4
2
0
-2
-4
-6
-8
0
50
100
150
200
250
300

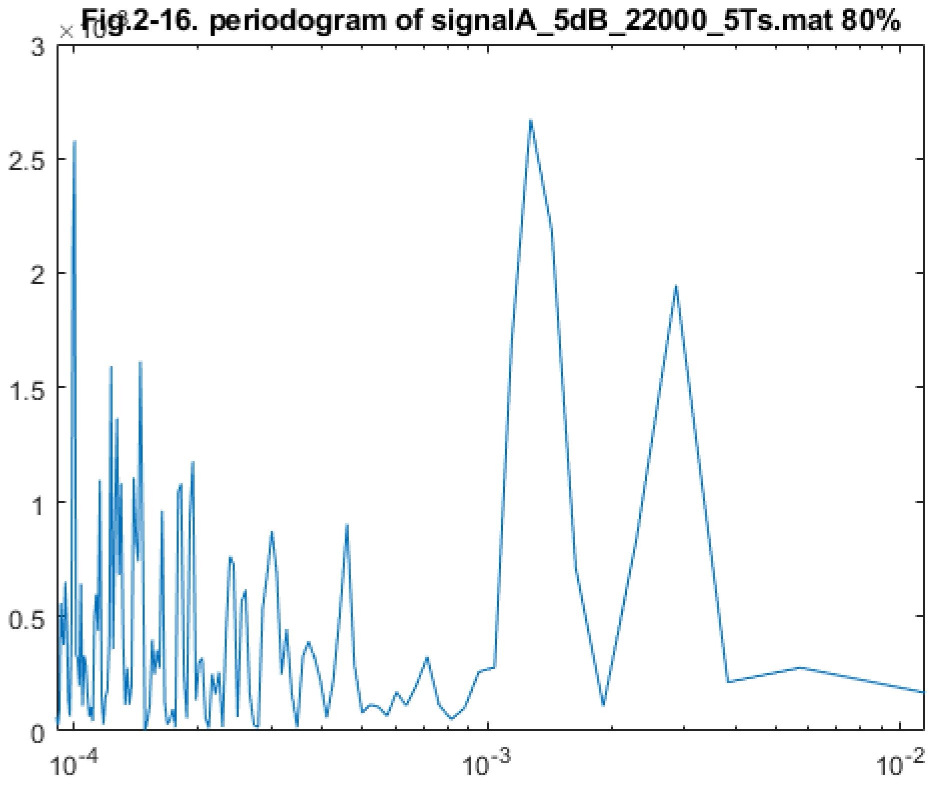


Fig.2-16. periodogram of signalA_5dB_22000_5Ts.mat 80%
× 10⁻³
3
2.5
2
1.5
1
0.5
0
10^{-4}
10^{-3}
10^{-2}

Unlike the output in figure 2-10, the trimmed signal by 20% of the same configuration as in figure 2-10 yield a discernable frequency as shown in figure 2-16 although several noise spikes require to be filtered for the actual power of the signal to be discernable from the effects of noise and lower Fs.

c. Repeat a) and b) above but this time truncate the signal to 60% of the original length. What can you say about having a shorter data record and the ability to determine the spectral characteristics of the signal?

```
clear;clc;
load('signalA_5dB_44100_5Ts.mat')
data = ytn;
data_len = length(data)*0.6; % new length by 60% of orig data
new_data = data(1:data_len); % 40% trimmed data, new data with 60% of orig
data=detrend(new_data);
figure; plot(data); title('Fig.2-17. signalA\_5dB\_44100\_5Ts.mat 60%');
title('60% signalA\_5db\_44100\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 44100);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-18. periodogram of signalA
\_5dB\_44100\_5Ts.mat 60%');

clear;clc;
load('signalA_5dB_22000_5Ts.mat')
data = ytn;
data_len = length(data)*0.6; % new length by 60% of orig data
new_data = data(1:data_len); % 40% trimmed data, new data with 60% of orig
data=detrend(new_data);
figure; plot(data); title('Fig.2-19. signalA\_5dB\_22000\_5Ts.mat 60%');
title('60% signalA\_5db\_22000\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 22000);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-20. periodogram of signalA
\_5dB\_22000\_5Ts.mat 60%');
```

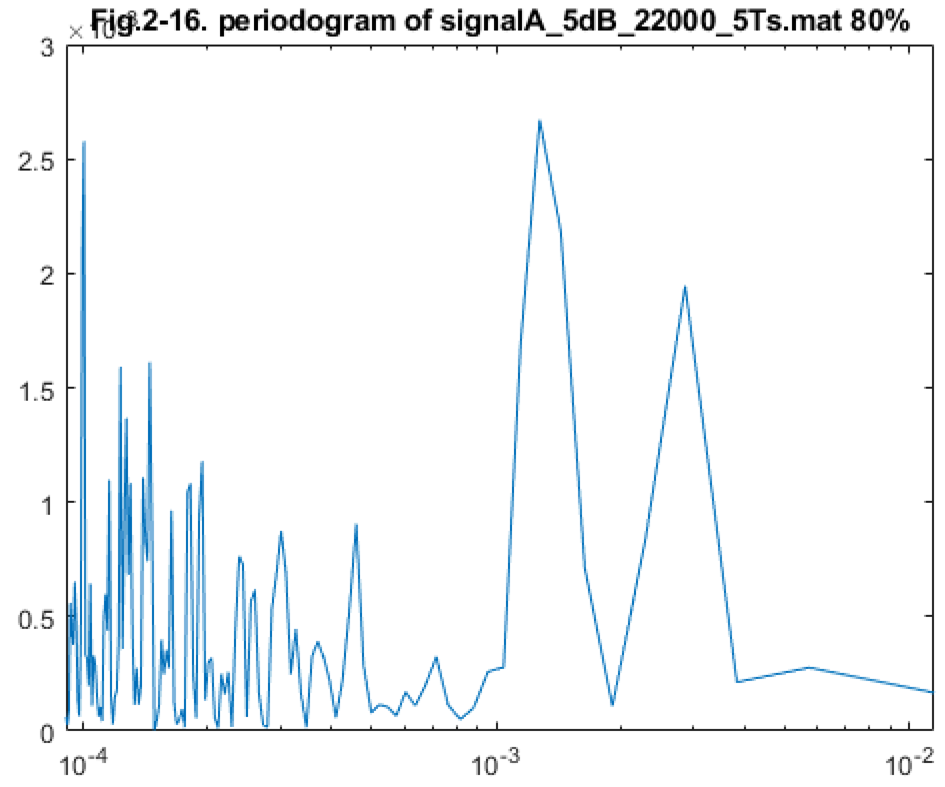
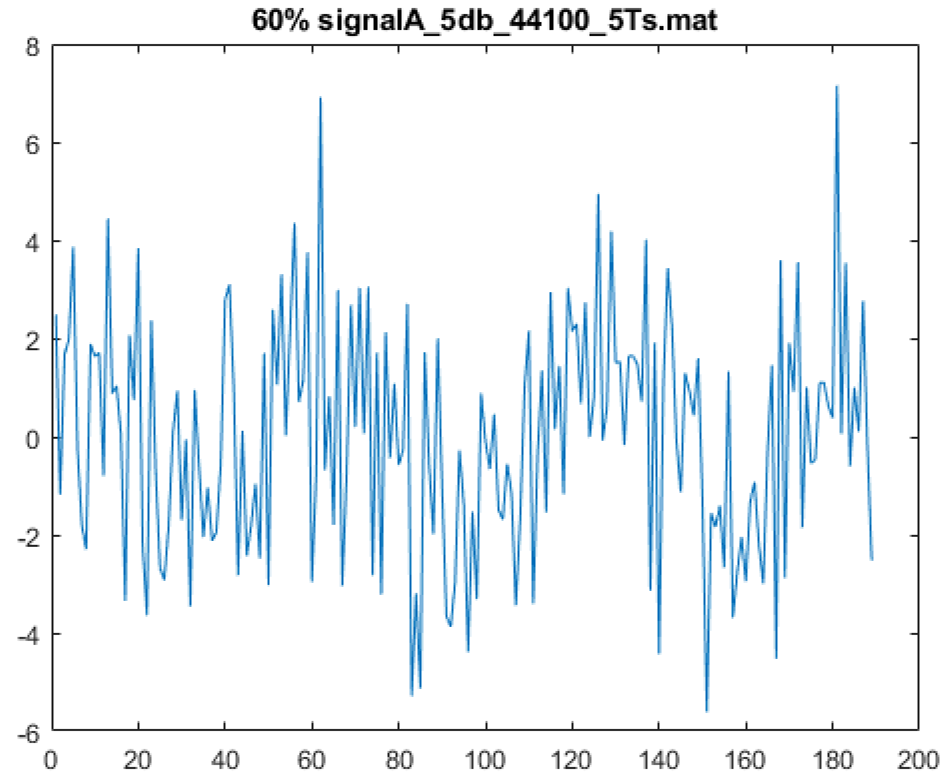


Fig.2-16. periodogram of signalA_5dB_22000_5Ts.mat 80%
× 10^-3
3
2.5
2
1.5
1
0.5
0
10^-4
10^-3
10^-2


60% signalA_5db_44100_5Ts.mat
8
6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160
180
200

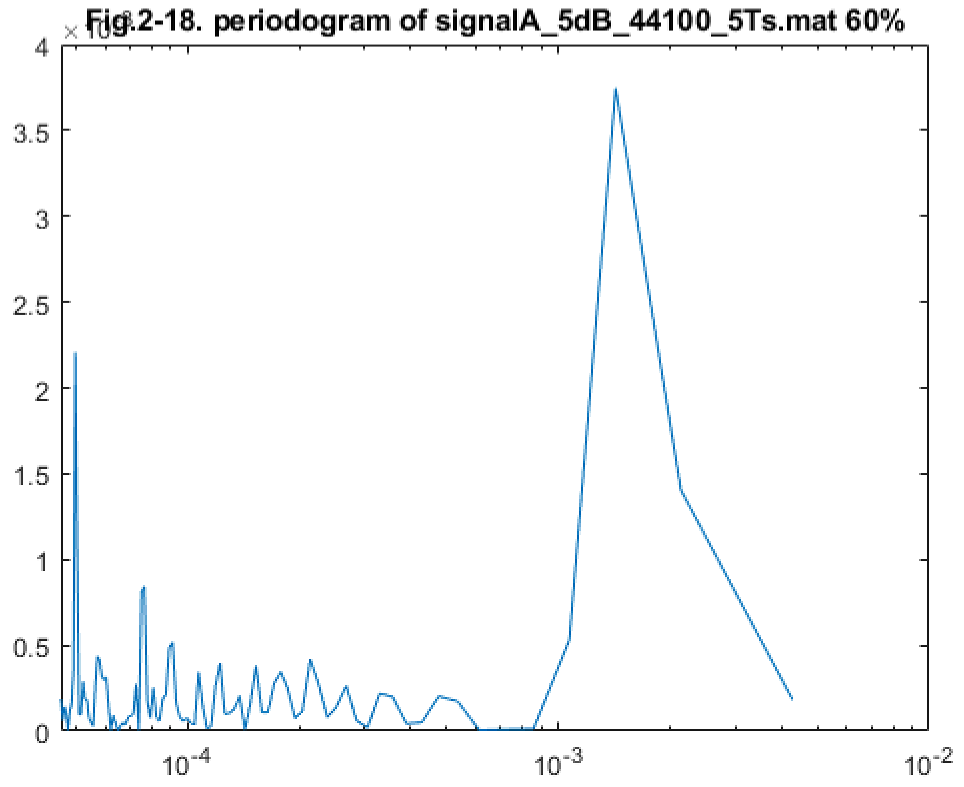
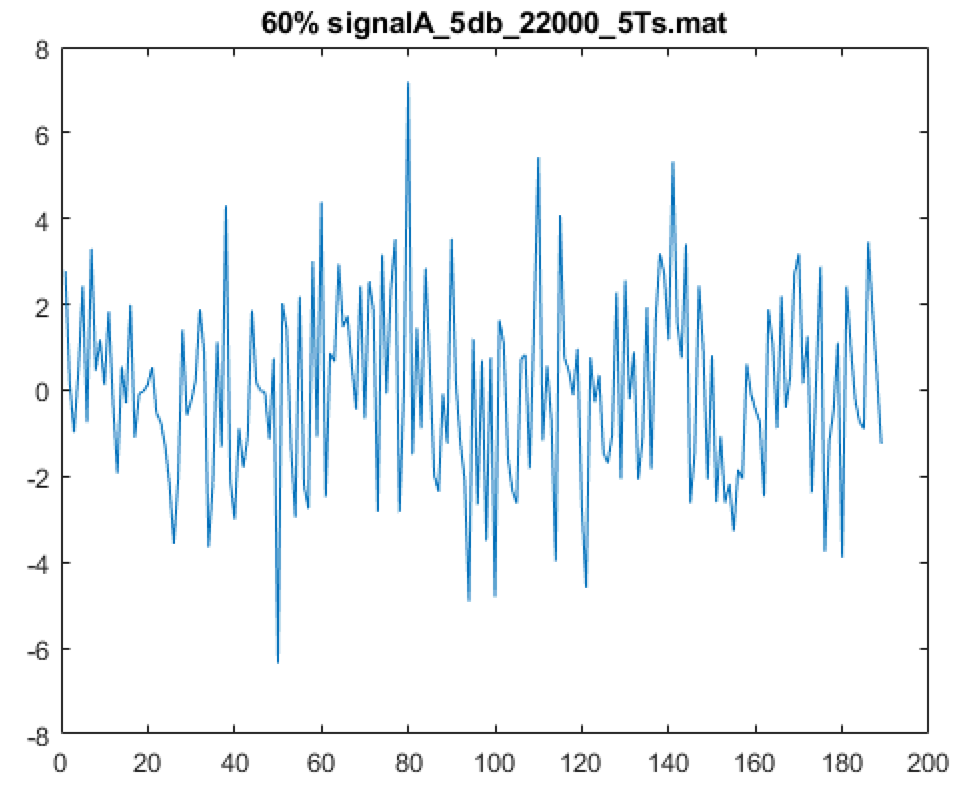
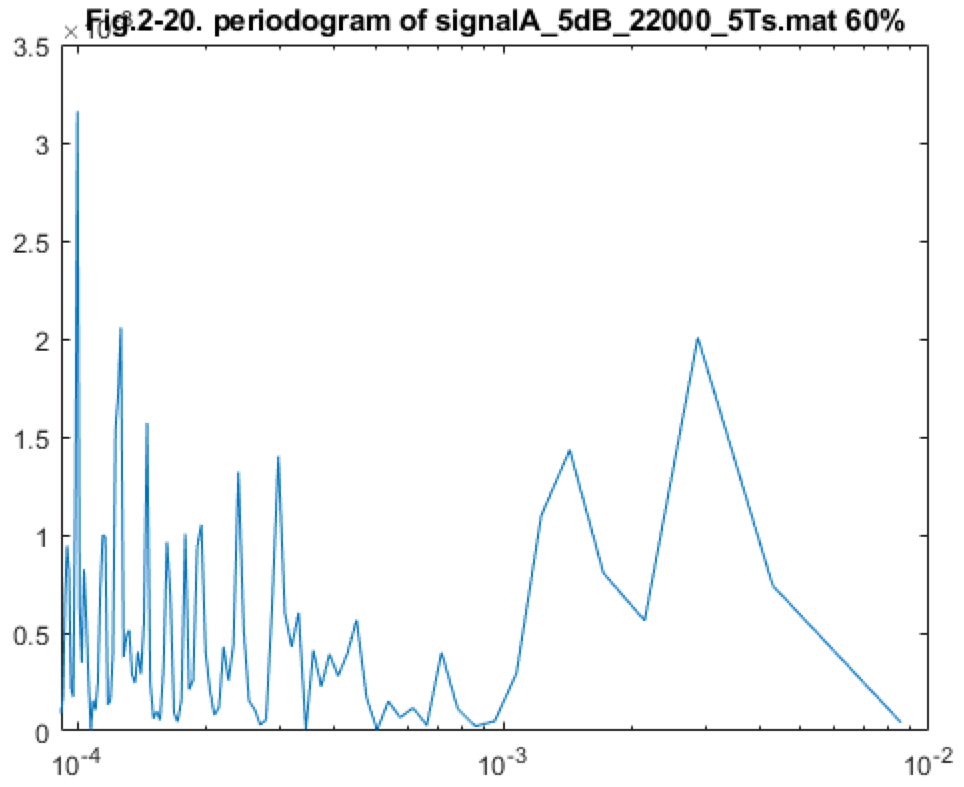


Fig.2-18. periodogram of signalA_5dB_44100_5Ts.mat 60%
× 10⁻³
4
3.5
3
2.5
2
1.5
1
0.5
0
10⁻⁴
10⁻³
10⁻²


60% signalA_5db_22000_5Ts.mat
8
6
4
2
0
-2
-4
-6
-8
0
20
40
60
80
100
120
140
160
180
200



Given the outputs in 4.c and as shown in figures 2-18 and 2-20, the signal with 44100 Fs and trimmed by 40% still yield a discernable frequency (figure 2-18) while the signal with 22000 Fs and trimmed or truncated by 40% of its total length yielded a discernable frequncy and other necessary spectral information although it still require further filtration as noise spikes are also discernable within its PSD estimate.

d. What is the shortest data record that you can afford but still be able to differentiate the signal frequencies and relative amplitudes for a signal with -5dB noise at 44100 sampling rate?

```
clear;clc;
load('signalA_5dB_44100_5Ts.mat')
data = ytn;
data_len = length(data)*0.5; % new length by 50% of orig data
new_data = data(1:data_len); % 50% trimmed data, new data with 50% of orig
data=detrend(new_data);
figure; plot(data); title('Fig.2-21. signalA\_5dB\_44100\_5Ts.mat 50%');
title('50% signalA\_5db\_44100\_5Ts.mat');
nfft=length(data);
[Pxx,f] = periodogram(data, hamming(nfft), nfft, 44100);
nn=2:length(f);
figure; semilogx(1./f(nn), Pxx(nn)); title('Fig.2-22. periodogram of signalA
\_5dB\_44100\_5Ts.mat 50%');
```

```
Warning: Integer operands are required for colon operator when used as index.
```

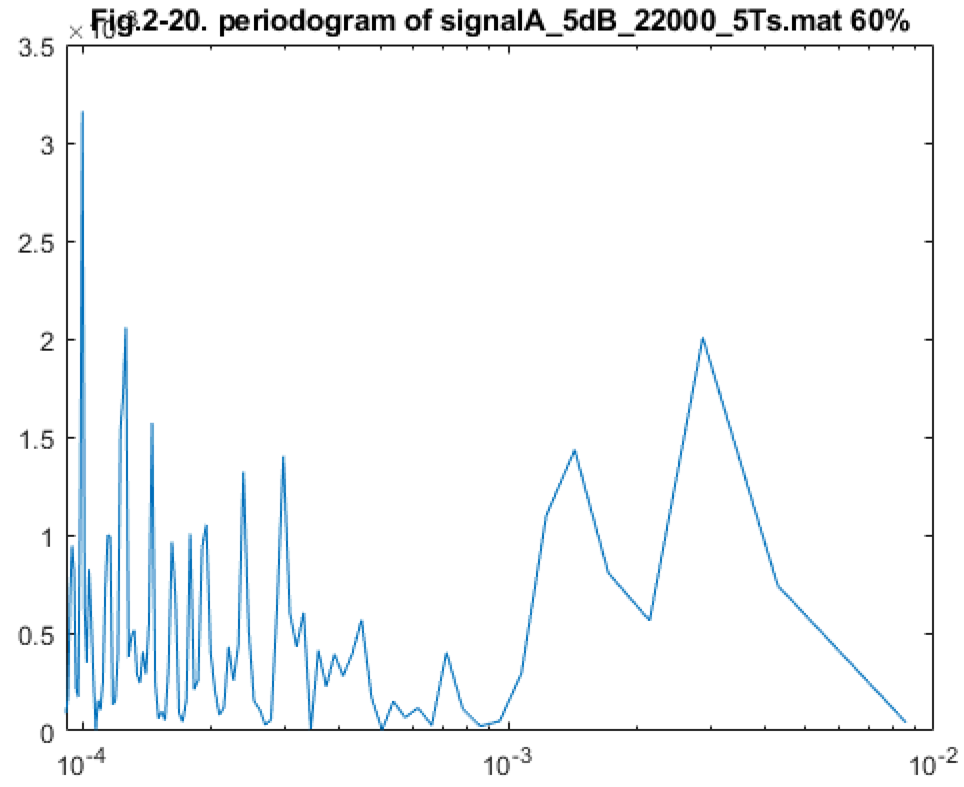
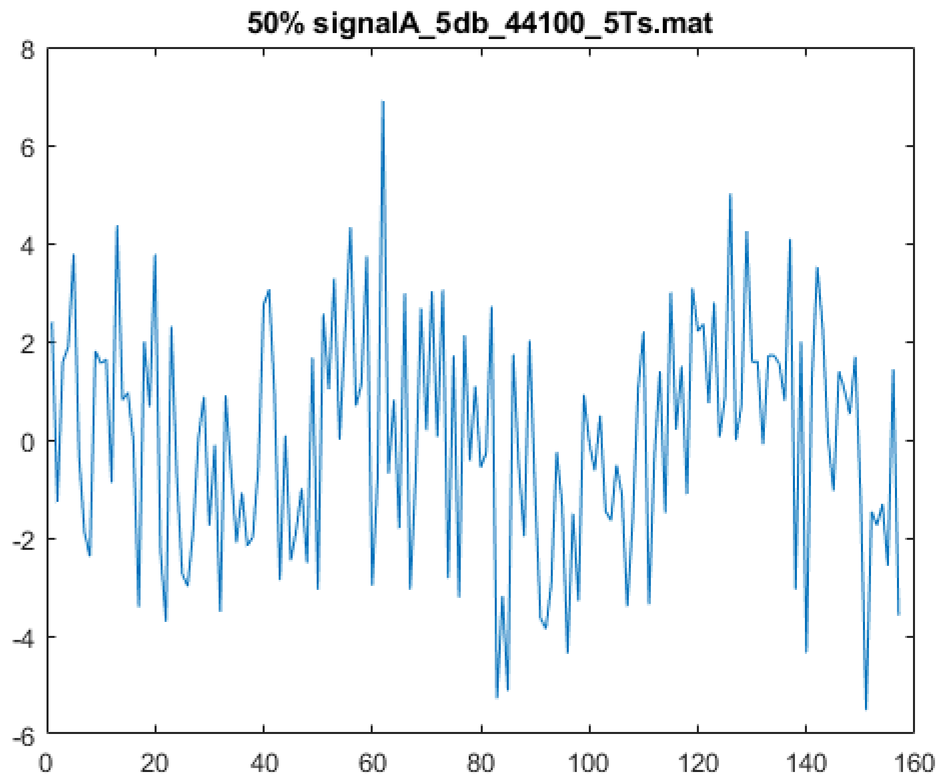
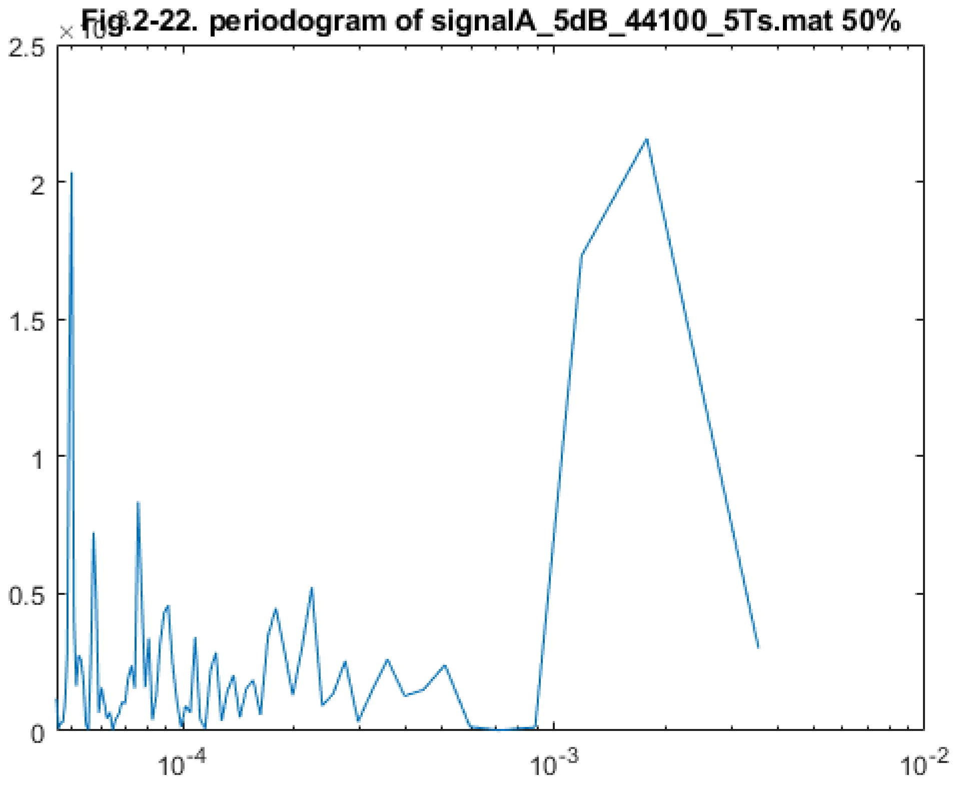


Fig.2-20. periodogram of signalA_5dB_22000_5Ts.mat 60%
× 10^-3
3.5
3
2.5
2
1.5
1
0.5
0
10^-4
10^-3
10^-2


50% signalA_5db_44100_5Ts.mat
8
6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160



Given the output in 4.d, 50% truncation of the signal is the shortest data in order for the given signal's frequency and spectral information to be discernable. Below 50%, its spectral informtation and PSD estimates will be compromised by the noise.

END OF FILE.

```
clear;clc;
```

Published with MATLAB® R2021b