Task 3.1

Let a random process.
x(ζ, t) = sin(2πft) + sin(3πft) + α · n(ζ, t)
The frequency f is 300 Hz, α is 0.1, and n(ζ, t) is normally distributed random noise. An A/D converter takes samples of a pattern function of the process with a sampling frequency of 3 kHz. The length of the buffer of the A/D converter is 2048.

a) Write a Matlab program that calculates and plots the PSD (power spectral density) of the sampled pattern function random process x(ζ, t) using the Wiener-Khintchine theorem. Don't use the Matlab function for direct PSD calculation. Use the Matlab function "randn" for the noise. Plot the sampled time signal in the timeframe from +0.00s to +0.02s, and plot the PSD (positive frequencies only). Don't forget the axis labels.

b) Increase α to 0.3 and run your program again. What do you observe? Submit the plots.

c) Increase α to 2.0 and run your program again. What do you observe? Submit the plots.

d) Run your program with α set to 0.3 and the sampling frequency to 900 Hz. Submit the plots.

e) Run your program with α set to 0.3 and the sampling frequency to 450 Hz. Submit the plots. Explain the results.

f) Run your program with α set to 0.3 and a sampling frequency of 3 kHz. In contrast to b) the buffer of the A/D converter now should have a length of 8192. Submit the plots.

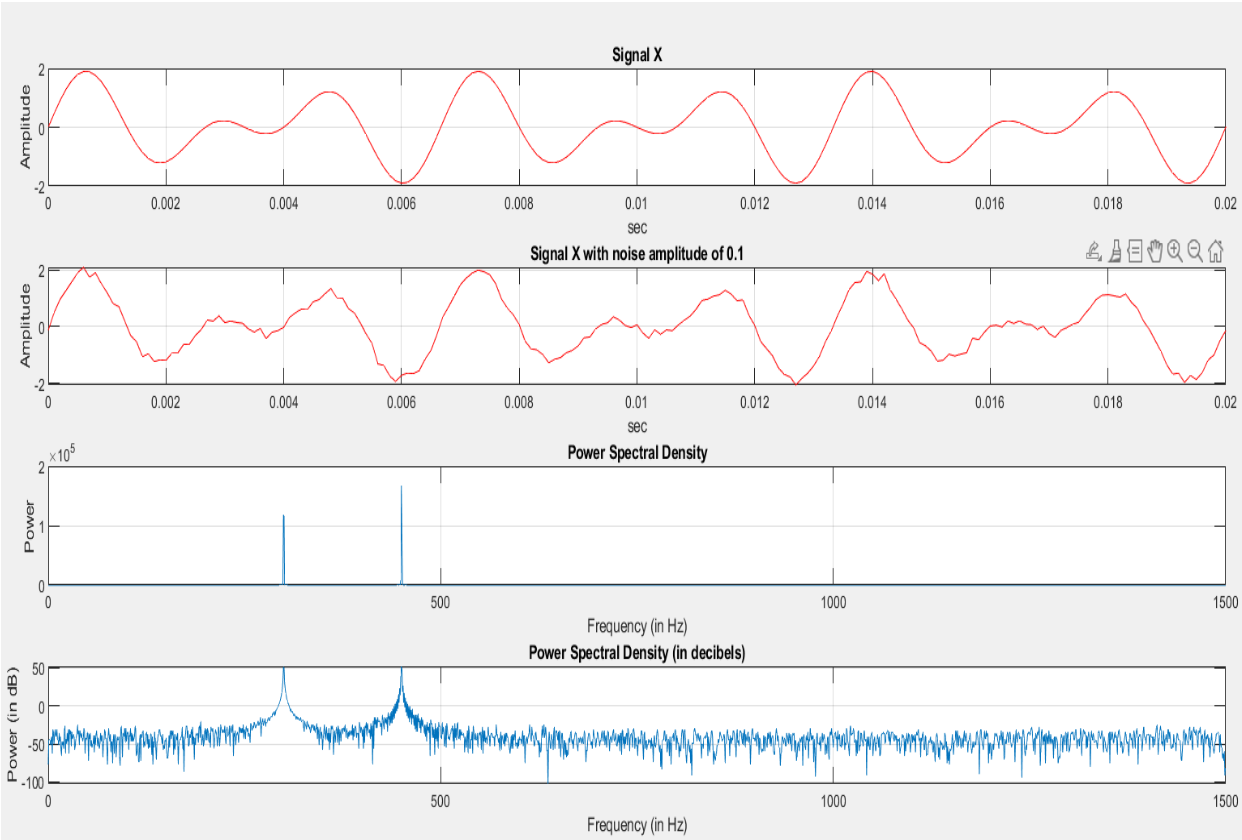
g) Take the settings of f) but instead of normally distributed random noise add uniformly distributed noise.

Answer:

a)

```
% Task 3.1a
close all;
clear all;
clc;

% Number of samples
N = 2048; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 3000;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;

% Given weight of normally distributed random noise
a = 0.1;

x = sin(2*pi*f*t1)+sin(3*pi*f*t1)
```

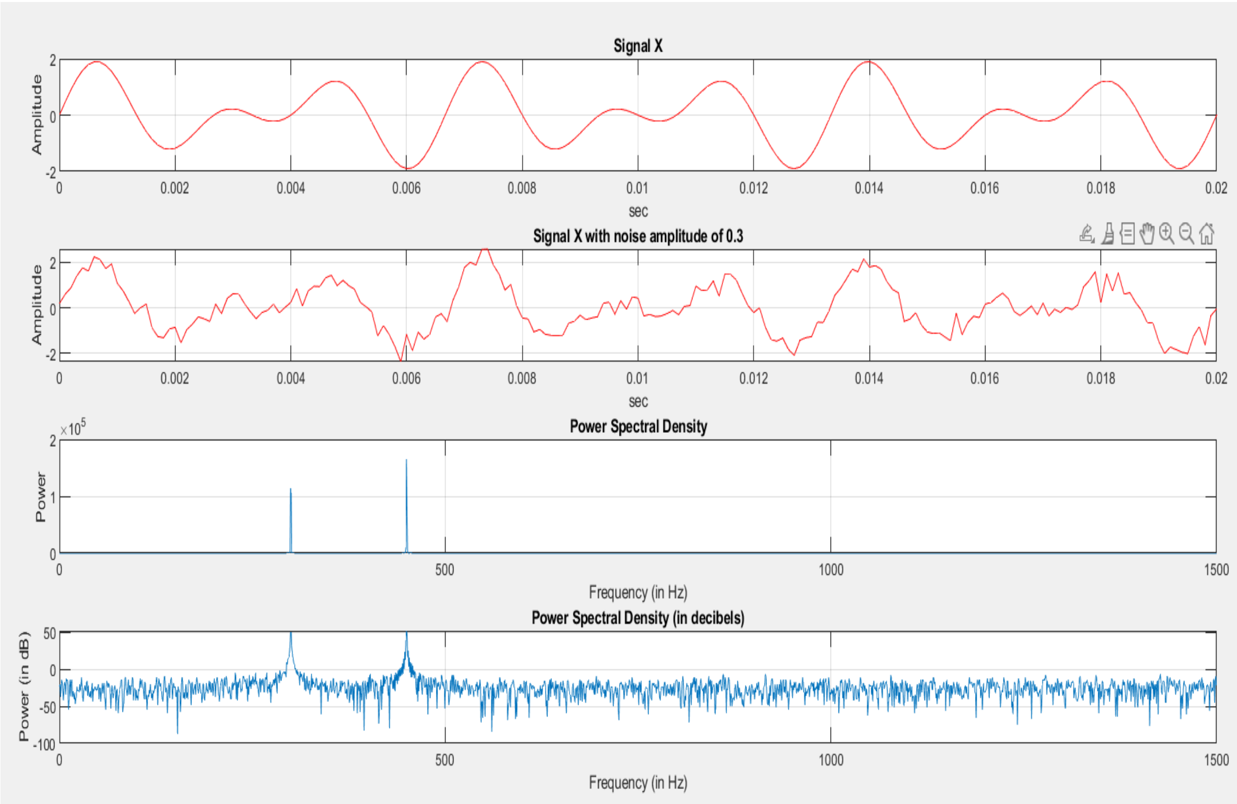
```matlab
x1 = x + a*randn(1,length(t1));

figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 0.1
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 0.1');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of dicrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1); % Size of 4095
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1); % Size of 4095
x1dft = x1dft(1:size_of_corr_signal/2+1); % Size of 2048
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD (Size of 2048)
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector (Size of 2048)
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```



b) Below is the code with increased α to 0.3. The increased value of alpha represents a higher amplitude of the random noise added to the signal. So the signal X is little bit more distorted in addition to noise having weight parameter α =0.3

```
% Task 3.1b

close all;
clear all;
clc;

% Number of samples
N = 2048; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 3000;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;
```

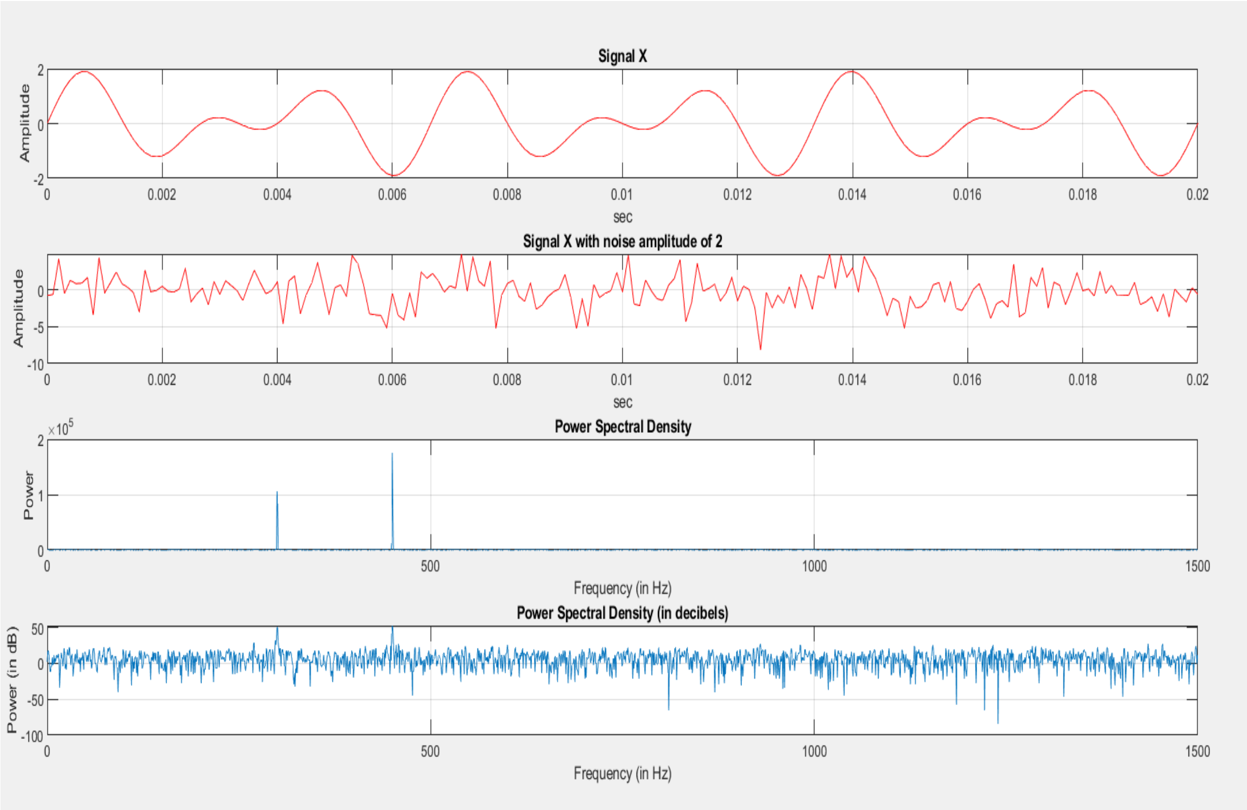
```matlab
% Given weight of normally distributed random noise
a = 0.3;

x = sin(2*pi*f*t1)+sin(3*pi*f*t1)
x1 = x + a*randn(1,length(t1));

figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 0.3
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 0.3');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of dicrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1);
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1);
x1dft = x1dft(1:size_of_corr_signal/2+1);
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```



c) Below is the code with increased α to 2. The increased value of alpha represents a higher amplitude of the random noise added to the signal. So, the signal X is much more distorted in addition to noise having weight parameter α =2.

```
% Task 3.1c
close all;
clear all;
clc;

% Number of samples
N = 2048; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 3000;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;
```

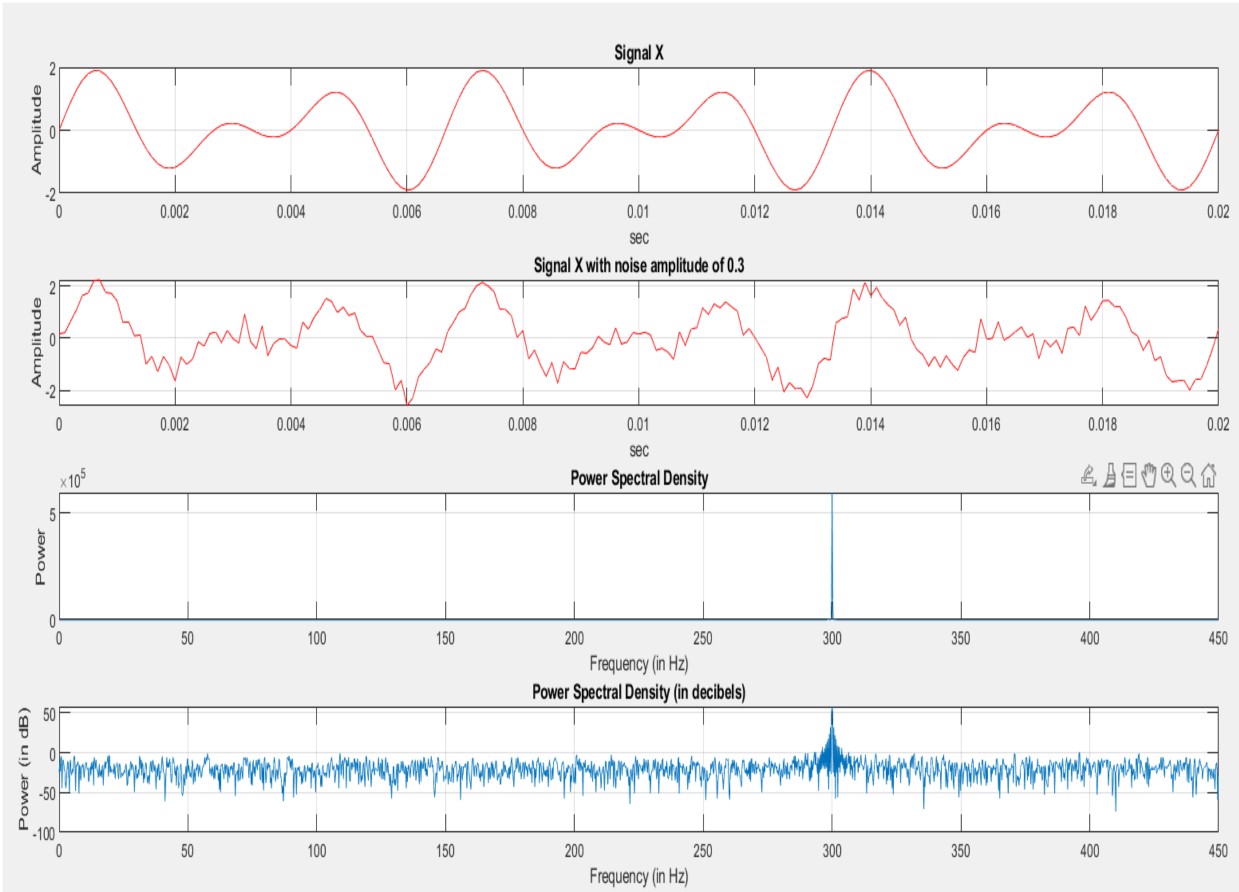
```
% Given weight of normally distributed random noise
a = 2;

x = sin(2*pi*f*t1)+sin(3*pi*f*t1)
x1 = x + a*randn(1,length(t1));

figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 2
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 2');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of dicrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1);
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1);
x1dft = x1dft(1:size_of_corr_signal/2+1);
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```



d)

```
% Task 3.1d
close all;
clear all;
clc;

% Number of samples
N = 2048; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 900;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;

% Given weight of normally distributed random noise
a = 0.3;
```

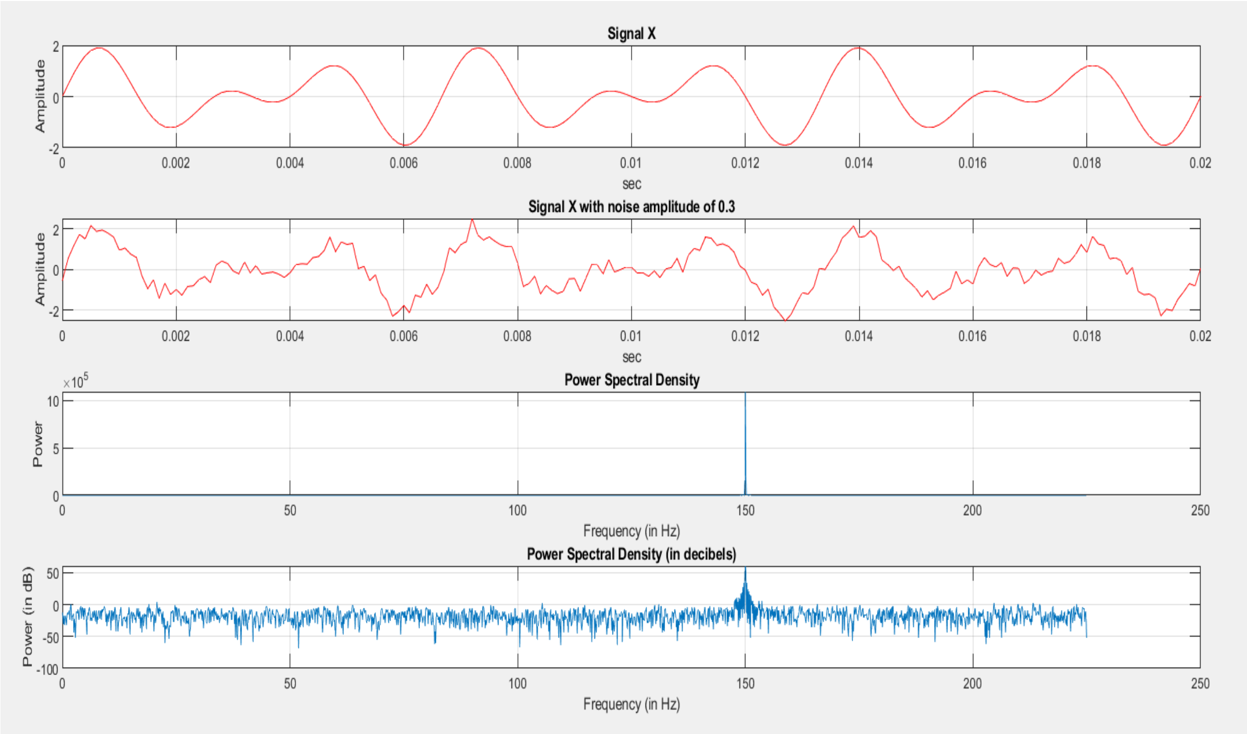
```
x = sin(2*pi*f*t1)+sin(3*pi*f*t1)
x1 = x + a*randn(1,length(t1));

figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 0.3
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 0.3');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of dicrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1);
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1);
x1dft = x1dft(1:size_of_corr_signal/2+1);
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```



e)

```
% Task 3.1e
close all;
clear all;
clc;

% Number of samples
N = 2048; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 450;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;
```

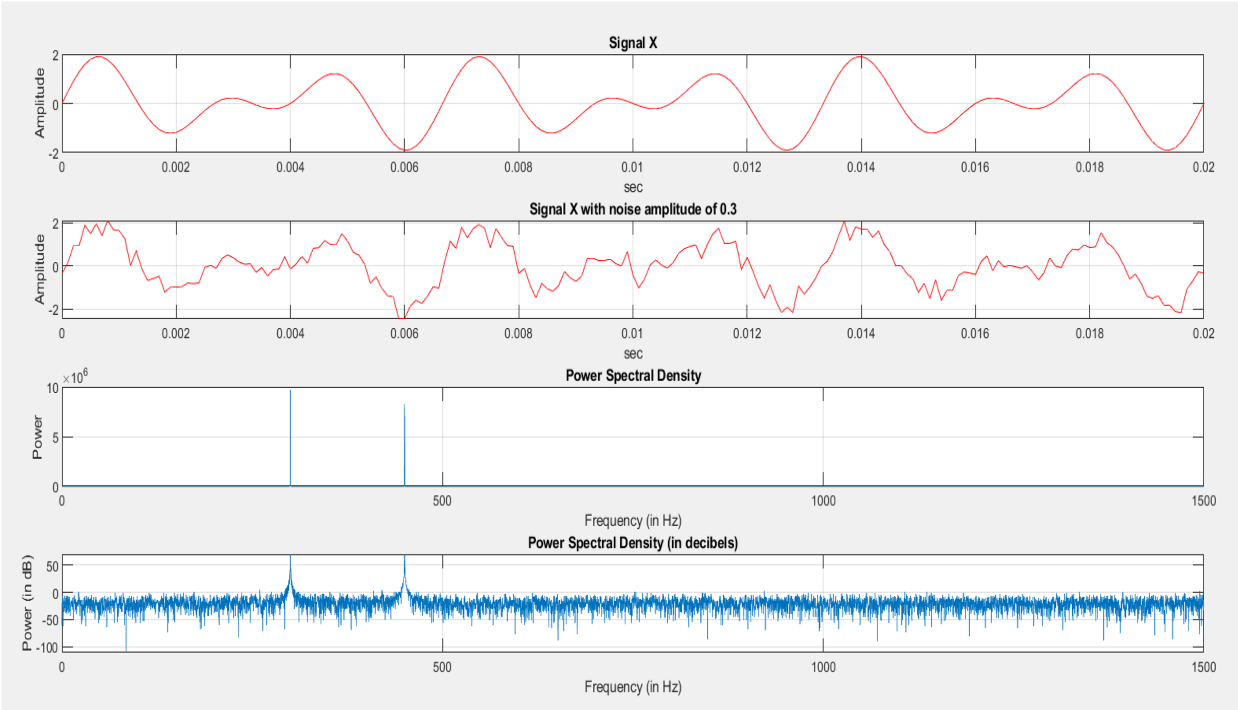
```
% Given weight of normally distributed random noise
a = 0.3;

x = sin(2*pi*f*t1)+sin(3*pi*f*t1)
x1 = x + a*randn(1,length(t1));

figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 0.3
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 0.3');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of discrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1);
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1);
x1dft = x1dft(1:size_of_corr_signal/2+1);
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```



f)

```
% Task 3.1f
close all;
clear all;
clc;

% Number of samples
N = 8192; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 3000;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;

% Given weight of normally distributed random noise
```

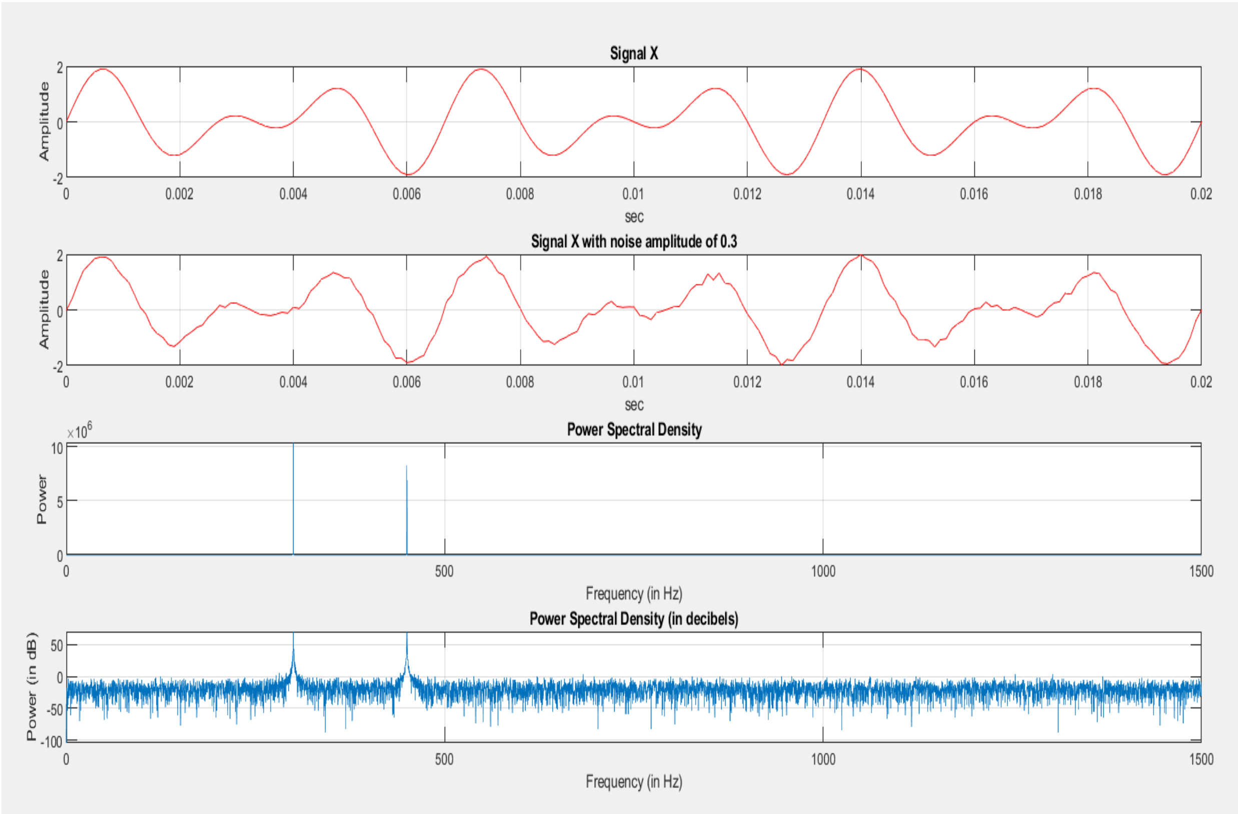
```
a = 0.3;

x = sin(2*pi*f*t1)+sin(3*pi*f*t1)
x1 = x + a*randn(1,length(t1));

figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 0.3
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 0.3');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of dicrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1);
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1);
x1dft = x1dft(1:size_of_corr_signal/2+1);
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```



g)

```
% Task 3.1g
close all;
clear all;
clc;

% Number of samples
N = 8192; % ADC buffer
% Frequency of Sine wave in Hz
f = 300;

% Sampling Frequency
Fs = 3000;
Ts = 1/Fs; % sampling period

% Time axis
t = (0:1:N-1)*1/Fs;
t1 = 0:1/10000:0.02;

% Given weight of random noise
a = 0.3;

x = sin(2*pi*f*t1)+sin(3*pi*f*t1)

% Generates signal x with uniformly distributed noise with a noise amplitude range of
[-0.5, 0.5]
x1 = x + a*(rand(1, length(t1)) - 0.5);
```

```
figure1 = figure('Position', [30, 100 ,1500, 600]);

% Original Signal Wave
subplot(4,1,1);
plot(t1, x, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X');
grid on

% Signal with noise amplitude of 0.3
subplot(4,1,2);
plot(t1, x1, 'r');
xlabel('sec');
ylabel('Amplitude');
title('Signal X with noise amplitude of 0.3');
grid on

x1 = sin(2*pi*f*t)+sin(3*pi*f*t) + a*randn(1,length(t));
% Use xcorrfunction to find PSD
N1 = 2*N; % Number of dicrete points in FFT = 2 * ADC buffer
[auto_cor1, lag1] = xcorr(x1);
size_of_corr_signal = length(auto_cor1);
x1dft = fft(auto_cor1);
x1dft = x1dft(1:size_of_corr_signal/2+1);
x1psd = (1/(Fs*size_of_corr_signal))*(abs(x1dft).^2); % Calculate PSD
x1psd(2:end-1) = 2*x1psd(2:end-1); % Multiply the amplitude by factor of 2
freq1 = (0:Fs/length(auto_cor1):Fs/2); % Frequency vector
Avg_power1 = sum(x1psd)*Fs/(N1); % Calculate Average power using PSD

subplot(4,1,3);
plot(freq1, x1psd);
title('Power Spectral Density');
xlabel('Frequency (in Hz)');
ylabel('Power');
grid on

subplot(4,1,4);
plot(freq1, 10*log10(x1psd));
title('Power Spectral Density (in decibels)');
xlabel('Frequency (in Hz)');
ylabel('Power (in dB)');
grid on
```


Signal X
Amplitude
sec
Signal X with noise amplitude of 0.3
Amplitude
sec
Power Spectral Density
Power
Frequency (in Hz)
Power Spectral Density (in decibels)
Power (in dB)
Frequency (in Hz)

Task 3.2

$x_1 = -1$, $x_2 = 0$, $x_3 = 1$

$$P(\{x(\tau, t+\tau) = x_i\} \mid \{x(\tau, t) = x_j\}) = \begin{cases} \frac{1}{3}(1+2e^{-|\tau|}) & \text{for } i=j \\ \frac{1}{3}(1-e^{-|\tau|}) & \text{for } i \neq j \end{cases}$$

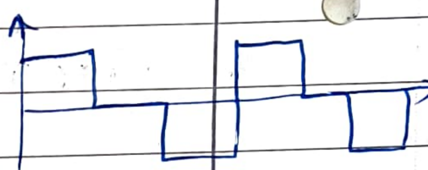
$i, j = 1, 2, 3$

a) $P(\{x(\tau, t) = x_i\}) = ?$ for $i = 1, 2, 3$

$p = \frac{1}{3}$

$$\frac{1}{3}(1+2e^{-|\tau|}) = a$$

$$\frac{1}{3}(1-e^{-|\tau|}) = b$$



$$P(\{x(\tau, t) = x_1\}) = \frac{1}{3}a + \frac{1}{3}b + \frac{1}{3}b$$

$$= \frac{a}{3} + \frac{2}{3}b$$

$$P(\{x(\tau, t) = x_2)) = \frac{1}{3}b + \frac{1}{3}a + \frac{1}{3}b$$

$$= \frac{a}{3} + \frac{2}{3}b$$

$$P(\{x(\tau, t)) = \frac{1}{3}b + \frac{1}{3}b + \frac{1}{3}a$$

$$= \frac{a}{3} + \frac{2}{3}b$$

b)

$$S_{XX}(\tau) = \sum_{i=1}^{3}\sum_{j=1}^{3} x_i x_j P(\{x(\tau, t+\tau) = x_i\} | \{x(\tau,t) = x_j\}) \, P\{x(\tau, t = x_j)\}$$

$$= \cancel{(1)(1)\frac{1}{3}(1+2e)}$$

$$= \frac{1}{3}\Big[(-1)(-1)\,a + (-1)(0)\,b + (-1)(1)\,b + (0).(-1)\,b + (0)(1)(b) + (0)(0)\,a + (1)(1)\,a + (1)(-1)\,b + (1)(0)\,b\Big]$$

$$= \frac{1}{3}\left[a - b + a - b\right]$$

$$= \frac{2}{3}(a-b) = \frac{2}{3} \times \frac{1}{3}\left[\cancel{1} + 2e^{-|\tau|} - \cancel{1} + e^{-|\tau|}\right]$$

$$= \frac{2}{3 \times 3}\left[3e^{-|\tau|}\right]$$

$$= \frac{2e^{-|\tau|}}{3}$$