**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)

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

A screenshot of a graph

Description automatically generated with low confidence

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;

% 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

A screenshot of a graph

Description automatically generated with low confidence

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

A screenshot of a graph

Description automatically generated with low confidence

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

A screenshot of a graph

Description automatically generated with low confidence

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

A screenshot of a graph

Description automatically generated with low confidence

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

A screenshot of a graph

Description automatically generated with low confidence

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

A screenshot of a graph

Description automatically generated with low confidence