Marim Elhanafy 201803468

Reham Eshbair 201701979

Asma Alnaimi 201678057

In Lab 2

**Design a system that can detect falls:**

1. Acquire motion data from mobile phone where it shows a simulation of the different motion states.

Done using Reham's phone.

1. Walking (no fall)
2. Putting phone down (no fall)
3. Sitting (no fall)
4. Fall

2. Analyze the signals in time and frequency domain and write your observations.

close all; clear; clc;

% Read Files:

walking = csvread('Walking.csv',1,1);

putting\_down = csvread('Putting Down.csv',1,1);

sitting = csvread('Sitting.csv',1,1);

falling = csvread('Falling.csv',1,1);

Time Domain:

% 1. Walking

t\_walking = walking(:,1);

z\_walking = walking(:,2);

y\_walking = walking(:,3);

x\_walking = walking(:,4);

% Plot

figure, plot(t\_walking,z\_walking,t\_walking,y\_walking,t\_walking,x\_walking);

xlabel('Time - Second Elapsed (sec)'); ylabel('Acceleration');

title('Walking (no fall) - Time Domain'); grid on;

legend('z','y','x');

% 2. Putting Down

t\_putting\_down = putting\_down(:,1);

z\_putting\_down = putting\_down(:,2);

y\_putting\_down = putting\_down(:,3);

x\_putting\_down = putting\_down(:,4);

% Plot

figure, plot(t\_putting\_down,z\_putting\_down,t\_putting\_down,y\_putting\_down,t\_putting\_down,x\_putting\_down);

xlabel('Time - Second Elapsed (sec)'); ylabel('Acceleration');

title('Putting Down (no fall) - Time Domain'); grid on;

legend('z','y','x');

% 3. Sitting

t\_sitting = sitting(:,1);

z\_sitting = sitting(:,2);

y\_sitting = sitting(:,3);

x\_sitting = sitting(:,4);

% Plot

figure, plot(t\_sitting,z\_sitting,t\_sitting,y\_sitting,t\_sitting,x\_sitting);

xlabel('Time - Second Elapsed (sec)'); ylabel('Acceleration');

title('Sitting (no fall) - Time Domain'); grid on;

legend('z','y','x');

% 4. Falling

t\_falling = falling(:,1);

z\_falling = falling(:,2);

y\_falling = falling(:,3);

x\_falling = falling(:,4);

% Plot

figure, plot(t\_falling,z\_falling,t\_falling,y\_falling,t\_falling,x\_falling);

xlabel('Time - Second Elapsed (sec)'); ylabel('Acceleration');

title('Falling - Time Domain'); grid on;

legend('z','y','x');

Observation for time domain:

In the first three (no fall) situations, the acceleration maximum range was small between -1.5 and 1.5 for walking, -4.5 to 4.5 for putting down, and -4 to 2.5 for sitting. While the acceleration maximum range for the falling was large which is from -66 to 84. Our second observation is that for the walking, the signal was almost uniform, and this is logic because the walking is a uniform movement.

Frequency Domain:

% 1. Walking:

walking\_concat = [z\_walking y\_walking x\_walking];

N\_w = length(walking\_concat);

Y\_w = fftshift(fft(walking\_concat,N\_w));

Y\_mag\_w = abs(Y\_w/N\_w);

Y\_ph\_w = angle(Y\_w);

Y\_power\_w = Y\_mag\_w.^2;

% f-axis

T\_w = max(walking(:,1));

Fs\_w = N\_w/T\_w;

df\_w = Fs\_w/N\_w; % Resolution in the frequency domain

f\_w = -Fs\_w/2:df\_w:Fs\_w/2-df\_w;

% Plot

figure, plot(f\_w,Y\_mag\_w);

xlabel('Frequency (Hz)'); ylabel('Magnitude');

title('Walking Magnitude Spectrum');

figure, plot(f\_w,Y\_ph\_w);

xlabel('Frequency (Hz)'); ylabel('Phase (rad)');

title('Walking Phase Spectrum');

figure, plot(f\_w,Y\_power\_w); xlim([0 Fs\_w/2]);

xlabel('Frequency (Hz)'); ylabel('Power');

title('Walking Power Spectrum');

% 2. Putting Down:

putting\_concat = [z\_putting\_down y\_putting\_down x\_putting\_down];

N\_p = length(putting\_concat);

Y\_p = fftshift(fft(putting\_concat,N\_p));

Y\_mag\_p = abs(Y\_p/N\_p);

Y\_ph\_p = angle(Y\_p);

Y\_power\_p = Y\_mag\_p.^2;

% f-axis

T\_p = max(putting\_down(:,1));

Fs\_p = N\_p/T\_p;

df\_p = Fs\_p/N\_p; % Resolution in the frequency domain

f\_p = -Fs\_p/2:df\_p:Fs\_p/2-df\_p;

% Plot

figure, plot(f\_p,Y\_mag\_p);

xlabel('Frequency (Hz)'); ylabel('Magnitude');

title('Putting Down Magnitude Spectrum');

figure, plot(f\_p,Y\_ph\_p);

xlabel('Frequency (Hz)'); ylabel('Phase (rad)');

title('Putting Down Phase Spectrum');

figure, plot(f\_p,Y\_power\_p); xlim([0 Fs\_p/2]);

xlabel('Frequency (Hz)'); ylabel('Power');

title('Putting Down Power Spectrum');

% 3. Sitting:

sitting\_concat = [z\_sitting y\_sitting x\_sitting];

N\_s = length(sitting\_concat);

Y\_s = fftshift(fft(sitting\_concat,N\_s));

Y\_mag\_s = abs(Y\_s/N\_s);

Y\_ph\_s = angle(Y\_s);

Y\_power\_s = Y\_mag\_s.^2;

% f-axis

T\_s = max(sitting(:,1));

Fs\_s = N\_s/T\_s;

df\_s = Fs\_s/N\_s; % Resolution in the frequency domain

f\_s = -Fs\_s/2:df\_s:Fs\_s/2-df\_s;

% Plot

figure, plot(f\_s,Y\_mag\_s);

xlabel('Frequency (Hz)'); ylabel('Magnitude');

title('Sitting Magnitude Spectrum');

figure, plot(f\_s,Y\_ph\_s);

xlabel('Frequency (Hz)'); ylabel('Phase (rad)');

title('Sitting Phase Spectrum');

figure, plot(f\_s,Y\_power\_s); xlim([0 Fs\_s/2]);

xlabel('Frequency (Hz)'); ylabel('Power');

title('Sitting Power Spectrum');

% 4. Falling:

falling\_concat = [z\_falling y\_falling x\_falling];

N\_f = length(falling\_concat);

Y\_f = fftshift(fft(falling\_concat,N\_f));

Y\_mag\_f = abs(Y\_f/N\_f);

Y\_ph\_f = angle(Y\_f);

Y\_power\_f = Y\_mag\_f.^2;

% f-axis

T\_f = max(falling(:,1));

Fs\_f = N\_f/T\_f;

df\_f = Fs\_f/N\_f; % Resolution in the frequency domain

f\_f = -Fs\_f/2:df\_f:Fs\_f/2-df\_f;

% Plot

figure, plot(f\_f,Y\_mag\_f);

xlabel('Frequency (Hz)'); ylabel('Magnitude');

title('Falling Magnitude Spectrum');

figure, plot(f\_f,Y\_ph\_f);

xlabel('Frequency (Hz)'); ylabel('Phase (rad)');

title('Falling Phase Spectrum');

figure, plot(f\_f,Y\_power\_f); xlim([0 Fs\_f/2]);

xlabel('Frequency (Hz)'); ylabel('Power');

title('Falling Power Spectrum');

Observation for frequency domain:

First, we observed that both phase and power spectrum will not be efficient to observe the difference between the situations, because the yellow (x) is not appearing clearly in both falling and putting down. Second observation, the magnitude spectrum was very clear to observe the differnces. First, the orange (y) in walking, putting, and sitting situations were very close to each other around 0.2, while the falling case is higher magnitude, witch is 1.4. Second, the difference also appears in the blue (z), the magnitude of all (no fall) situations was almost zero, while in the fallling situation it was almost 2.

3. Based on observations, implement a DSP algorithm that can process the motion data and detect the fall and the time of the fall (if any) to inform the user in MATLAB.

We will implement the DSP algorithm on the frequency domain. First, in the no fall situations the maximum magnitude was 0.3, while for falling was 2. So, to select our threshold, we calculated it as following: (2 - 0.3) / 2 = 0.85, and for safety we will choose our threshold to be 0.9. So, if the magnitude is higher than 0.9 we will assume it as a falling situation.

detect\_motion(Y\_mag\_w,t\_walking); % No output (no detected motion)

detect\_motion(Y\_mag\_p,t\_putting\_down);% No output (no detected motion)

detect\_motion(Y\_mag\_s,t\_sitting);% No output (no detected motion)

detect\_motion(Y\_mag\_f,t\_falling);