PROJETO DE ANÁLISE E TRANSFORMAÇÃO DE DADOS

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Trabalho realizado por:

- Aníbal Rodrigues 2019224911
- João Silva 2019216753
- Mário Lemos 2019216792

Introdução:

Este projeto foi realizado no âmbito da disciplina de Análise e Transformação de Dados com o objetivo de analisar os dados recolhidos de acelerómetros de smartphones, nos domínios do tempo e da frequência tentando identificar os diferentes tipo de atividades.

Importação de sinais:

Para facilitar no acesso futuro às diferentes atividades (dinâmicas, estáticas, transição) começamos por definir alguns vetores (**dyn_activities**, **sta_activities, trans_activities**) e também um array com os nomes das atividades e outro com as cores respetivas. Foi também definida a frequência de amostragem com o valor presente no enunciado bem como um array com o nome dos ficheiros a acessar.

Na importação dos sinais usamos a função **importdata** que guarda os valores das coordernadas (x,y,z) presentes nos ficheiros de texto numa matriz (chamada **x**). Cada coluna de x corresponde um ficheiro de texto. Já na variável **labels** é guardado os dados presentes no ficheiro "labels.txt".

```
%%
%1)
format long;

path = 'data\';
files = {'acc_exp09_user05' 'acc_exp10_user05' 'acc_exp11_user06' ...
    'acc_exp12_user06' 'acc_exp13_user07' 'acc_exp14_user07' ...
    'acc_exp15_user08' 'acc_exp16_user08' 'labels'};

n_activities = 12;
dyn_activities = 1:3;
sta_activities = 4:6;
trans_activities = 7:12;

activities = {'Walking' 'Walking_Upstairs' 'Walking_Downstairs' ...
                'Sitting' 'Standing' 'Laying' ...
                'Stand_to_sit' 'Sit_to_stand' 'Sit_to_lie' 'Lie_to_sit'...
                'Stand_to_lie' 'Lie_to_stand'};

colors = {'5a79d2' '2356ea' '8597c9' 'f0d725' 'f08425' 'f03025' ...
    '25f076' '37a765' '1f7c46', '83e66e', '4fb938', '165e07'};

fs = 50;
%disp_files_perU = 2;
analyse_file = 2; %Usado no 3.1 para testar as janelas

x=cell(1,8);

for i=1:9
    filename =[path files{i} '.txt'];
    if(i==9)
        labels = importdata(filename,' ');
    else
        x{i} = importdata(filename,' ');
    end
end
```

Representação gráfica dos sinais importados:

Antes de passarmos para a representação, reorganizamos os dados recolhidos numa tabela **data,** onde cada linha representa uma atividade diferente. Já as colunas dessa mesma tabela representam: vetores de tempo e a secções nas dimensões x, y e z (por esta ordem). Esta tabela será bastante útil no futuro na análise das diferentes características das atividades

Para representar cada experiência começamos por ir buscar o seu nº correspondente e utilizador. A partir desta informação construímos um vetor tempo consoante o nº de dados que o ficheiro em análise contém. De seguida construímos com ajuda do comando **figure** e a função **subplot** o espaço onde iriamos representar o nosso sinal (sendo feito um subplot *3x1* onde em cada linha teremos o gráfico correspondente a cada eixo de coordenadas).

Usamos finalmente a função **plot_signal** para finalmente representar graficamente os sinais nos diferentes eixos.

Na figura 1 é mostrado um exemplo desta representação.

```
%%
% 2)

data = cell(n_activities, 4);
data(:) = {{}};
for i = 1:length(x)
    % Obter o nº da experiência e o utilizador correspondente
    file_ids = sscanf(files{i}, 'acc_exp%d_user%d');

    file_labels = labels(labels(:,1)==file_ids(1) & ...
                labels(:,2)==file_ids(2), 3:end);
    % Vetor de tempo
    t=(0:length(x{i})-1/fs/60);

    for j = 1:length(file_labels)
        % Intervalo de ocorrência de determinada experiência
        interval = file_labels(j,2):file_labels(j,3);
        % Guardar o vetor de tempo em função do intervalo obtido
        data{file_labels(j,1),1} = [data{file_labels(j,1),1}; t(interval)];
        % Guardar nas últimas 3 colunas as coordenadas correspondentes
        for k = 1:3
            data{file_labels(j,1),k+1} = ...
           [data{file_labels(j,1),k+1}; x{i}(interval,k)];
        end
    end                 -                                   -
    file_ids = sscanf(files{i}, 'acc_exp%d_user%d');
    file_labels = labels(labels(:,1) == file_ids(1) & ...
        labels(:,2) == file_ids(2),3:end);
    %vetor de tempo
    N = length(x{i});
    t = (0:N-1)/fs/60;
    %criar uma figura
    figure
    %fazer plot das 3 dimensões no mesmo gráfico
    sgtitle(files{i})
    for j = 1:3
        subplot(3, 1, j);
        plot_signal(t, x{i}(:,j), file_labels, j, colors, activities);
    end
end
```

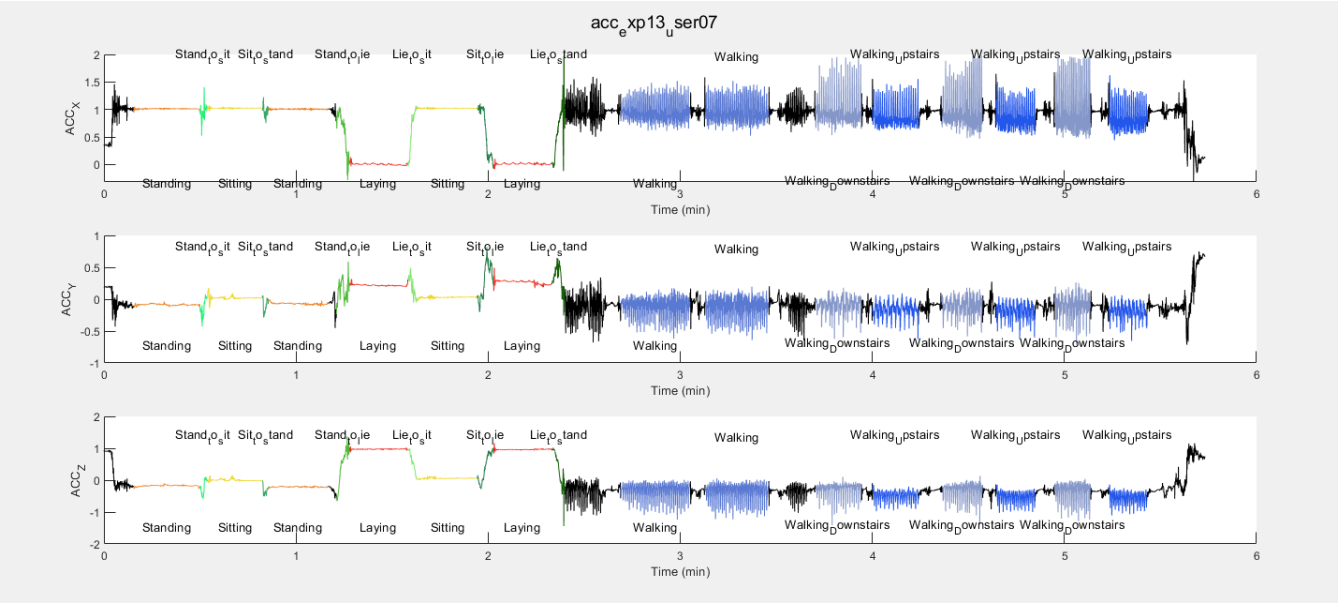

Figura 1 – Representação gráfica do sinal do ficheiro `'acc_exp13_user07.txt'`

Escolha da janela deslizante dentro dos tipos estudados:

Visto que a DFT conta que o sinal seja periódico, uma janela retangular aplicada num sinal leva a que as DFTs (depois de aplicadas as janelas) contenham componentes de frequência que não estão presentes no sinal (quando aplicada a janela). Isto acontece, pois, as suas bordas costumam constituir descontinuidades no presumido sinal periódico. Escolher uma janela onde as bordas tenham valores próximos de zero acaba por atenuar, na sua grande maioria, as descontinuidades, mas tendo a desvantagem de tornar a informação nas bordas do sinal menos relevante.

Ora visto que os sinais a estudar contêm secções onde as frequências diferem bastante, seria vantajoso utilizar uma janela não retangular, já que a atenuação das frequências resultantes das descontinuidades não permite o desaparecimento das frequências mais relevantes de cada secção.

Já no domínio da frequência, é importante que a DFT, segundo a janela, escolhida apresente largura espectral reduzida. Desta forma evitamos sobreposição de frequências e também que os lobos laterais apresentem valores reduzidos, sendo que assim não serão geradas frequências indesejadas na DFT final. De modo que o efeito dos lobos laterais significativos seja maior (e em contrapartida o menor nos lobos menos significativos) seria preciso então uma função onde as bordas apresentassem valores iguais a zero.

Logo, dentro das janelas estudadas no projeto, escolhemos a janela de Hann já que esta apresenta valores nulos nas bordas, eliminando assim todas as descontinuidades. Foi também verificado a janela de Hann apresenta um pico menos largo na DFT e também resultados satisfatórios em 95% dos casos em que é usado.

Na figura 2 é apresentado a aplicação das diferentes janelas numa das experiências. Já na figura 3 é apresentada a aplicação das diferentes janelas na atividade Walking Downstairs ao longo de todos os dados recolhidos e na sua respetiva DFT.

```
%%
% 3.1) AD -> Walking_Downstairs) (+/- percebido, ver frame e overlap)

activity = dyn_activities(3);

% Obter o tamanho do sinal
N = size(x{analyse_file}, 1);
disp(N);
% Definir a propriedades do frame (windows)
frame = fix(N*0.1);
overlap = fix(frame/2);

% Dar plot do sinal com diferentes janelas
plot_signal_framed([0 (N-1)/fs/60], x{analyse_file}(:,2), ...
    {@rectwin, @hamming, @hann}, {'Rectangular', 'Hamming', 'Hann'}, ...
    frame, overlap, 'Different frame functions effects on a signal');
```

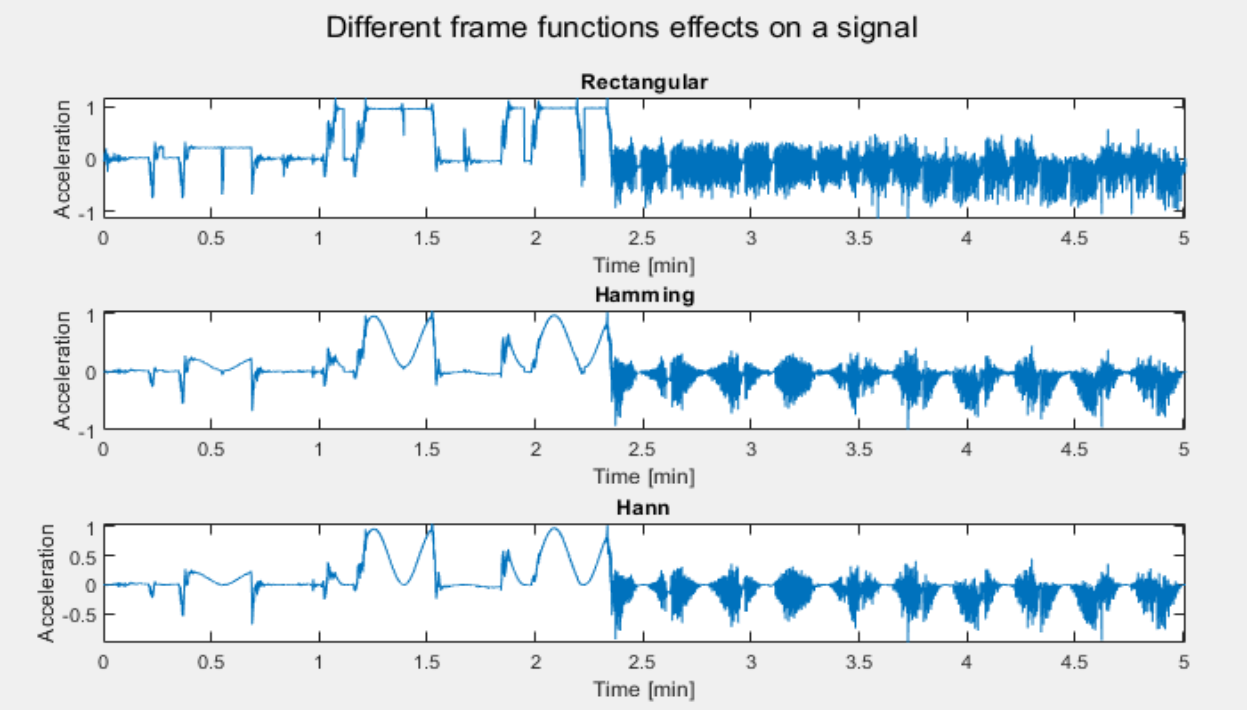

Figura 2 – Representação da aplicação das diferentes janelas no sinal do ficheiro `acc_exp10_user05.txt`

```
% Dar plot da atividade Walking_Downstairs com diferentes janelas
plot_signal_dft_framed(data{activity,1}{activity}, data{activity,3}{activity}, ...
    {@rectwin, @hamming, @hann}, {'Rectangular', 'Hamming', 'Hann'}, ...
    fs, 'Different frame functions effects on the |DFT|', ...
    colors{activity});
```

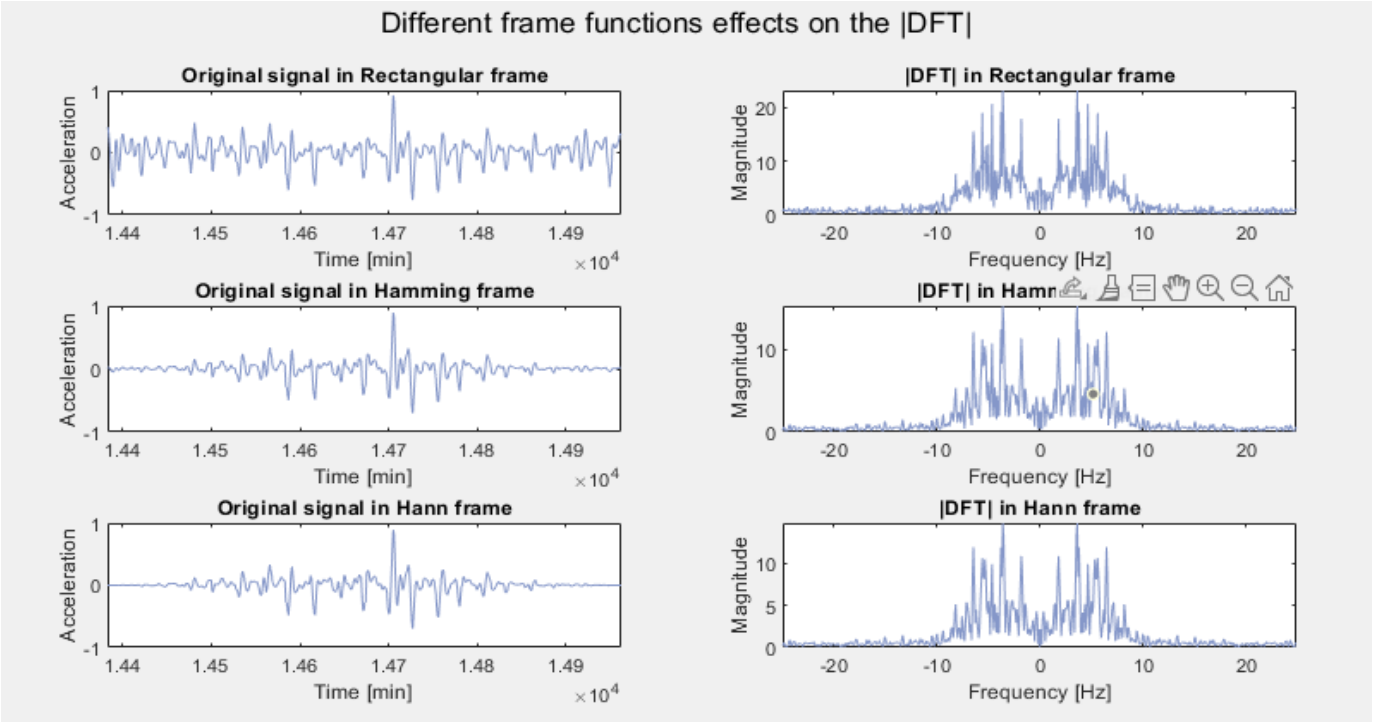

Figura 3 – Representação da aplicação das diferentes janelas na atividade Walking Downstairs ao longo de todos os dados recolhidos e na sua respetiva DFT.

Representação gráfica das DFTs dos sinais importados:

Para representar graficamente cada DFT dos sinais recorremos à utilização da janela escolhida (Hann). O procedimento é praticamente o mesmo usado para representar o sinal normal, com algumas diferenças notórias. Primeiramente em vez de representar o sinal de cada ficheiro todo corrido fomos dividindo em subplots de modo a calcular a DFT de cada atividade que iria ocorrendo. Segundo, em vez de representar o sinal na sua originalidade, foi representada a sua DFT. Ainda assim fizemos este mesmo procedimento também para os três eixos. Por fim usamos a função **plot_acc_dft** para representar as DTFs dos sinais.

Na figura 4 é mostrado um exemplo desta representação.

```
%%
% 3.2)
for i=1:length(x)
    file_ids = sscanf(files{i}, 'acc_exp%d_user%d');
    file_labels = labels(labels(:,1) == file_ids(1) & ...
        labels(:,2) == file_ids(2),3:end);
    % Obter o vetor tempo
    N = length(x{i});
    % Criar uma nova figura
    figure
    % Dar plot nos 3 eixos de todas as atividades da experiência em
    % questão
    plot_acc_dft(x{i}, file_labels, colors, activities, fs, @hann, files{i});
end
```

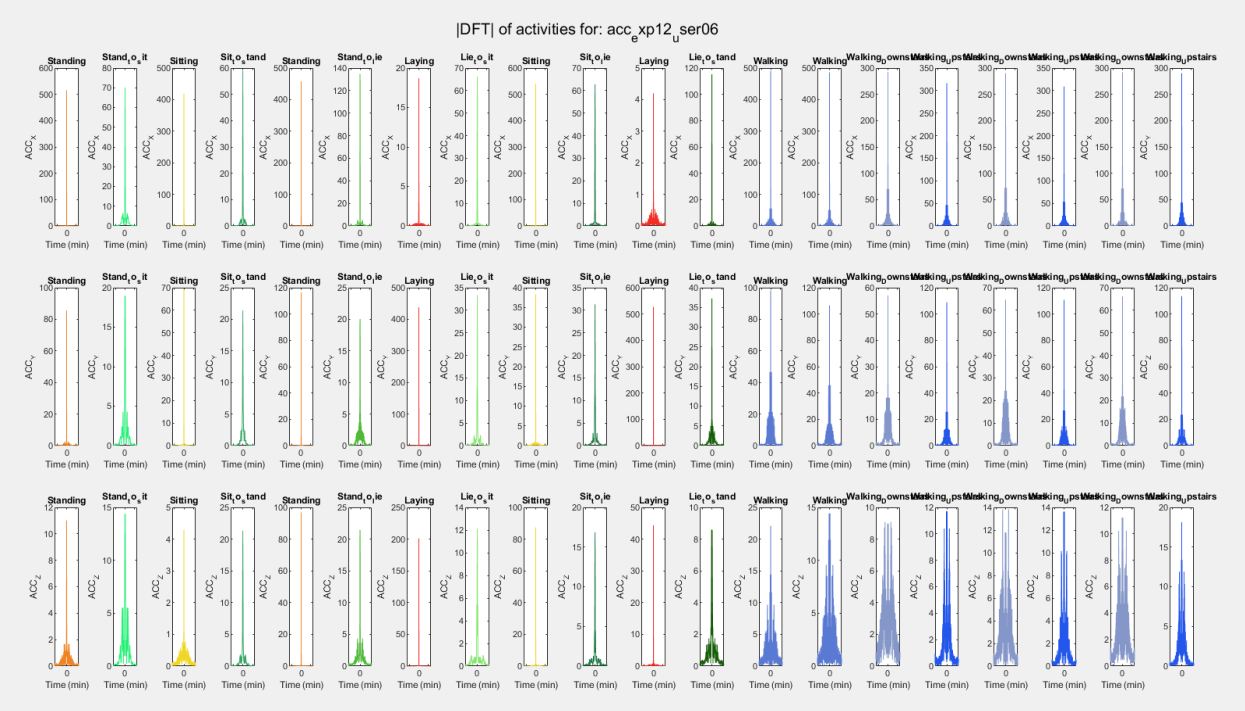

Figura 4 - Representação gráfica da DFT (segundo a janela de Hann) do sinal do ficheiro `'acc_exp12_user06.txt'`

Que tipo de características espectrais nos poderiam ajudar a diferenciar o tipo de atividade?

Para ajudar a diferenciar o tipo de atividade poderíamos usar diferentes tipos de características espectrais, tais como: o **período** do sinal, a **frequência** do sinal, a **amplitude** do sinal, **magnitude da dft** do sinal**, média** do sinal ou até mesmo o **declive obtido pela regressão linear** do sinal original.

Passaremos no ponto seguinte a esta análise.

Análise dos diferentes grupos de atividades segundo as características espectrais:

- <u>Nas atividades dinâmicas</u>:

  A distinção neste tipo de atividades começa por separar as atividades de Walk de Walking Upstairs e Walking Downstairs através da distância média entres os picos da DFT (com treshold de 7% do máximo), onde valores superiores a 25 no eixo x representam a atividade de Walk, enquanto para valores menores a 25 no eixo x teríamos a atividades de Walking Upstairs e Walking Downstairs (Figura 6).

  De seguida tentámos utilizar média dos sinais, onde conseguimos identificar pequenos grupos de pontos onde cada um deles representa uma atividade diferente (Figura 7).

  Analisando o declive obtido pela regressão linear onde sem grandes surpresas verificamos que todas atividades dinâmicas apresentavam declives dentro da mesma ordem (Figura 8).

  Em suma concluímos que provavelmente a melhor característica espectral para este distinguir dentro das atividades dinâmicas seria distância média entres os picos da DFT (com treshold de 7% do máximo)
  As imagens seguintes pretendem clarificar estes resultados.

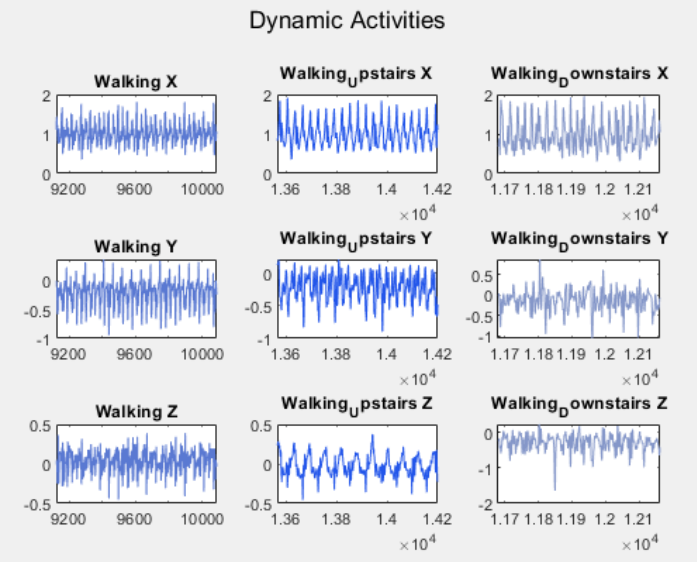

Figura 5 - Representação geral dos dados vindos das atividades dinâmicas

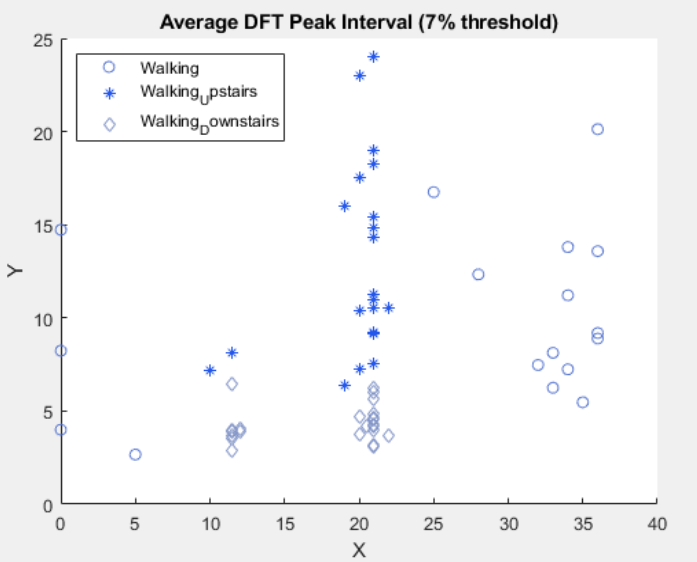

Figura 6 – Análise segundo a distância média entre os picos da DFT (com treshold de 7% do máximo)

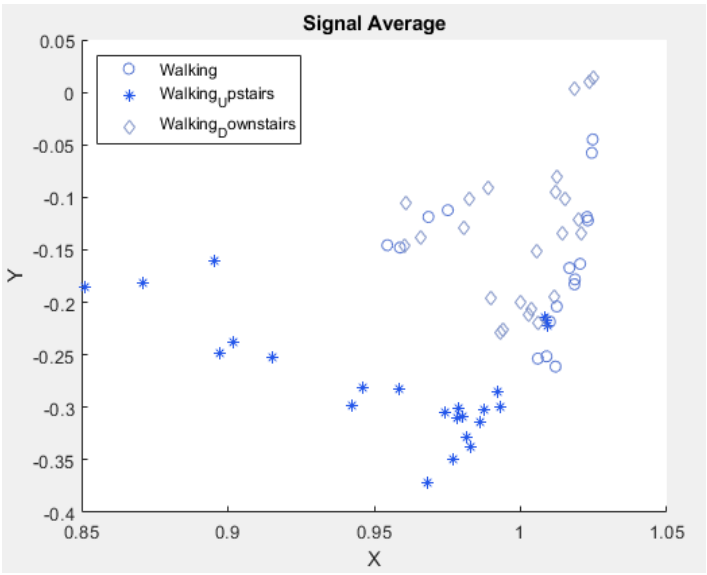

Figura 7 – Análise segundo a média do sinal

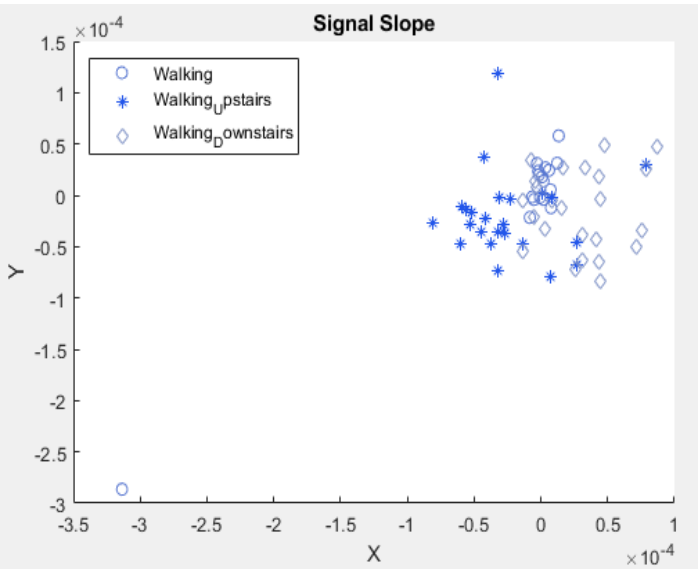

Figura 8 – Análise segundo o declive obtido pela regressão linear

- <u>Nas atividades estáticas:</u>

  Começando novamente pela análise segundo a distância média entre os picos da DFT (com treshold de 7% do máximo). Apesar de haver uma pequena distinção entre as atividades Sitting e Standing da atividade de Laying, onde a atividade Laying apresenta uma componente nula no eixo do Y, mas em contrapartida uma componente sempre positiva no eixo X, sendo que as atividades de Stitting e Standing apresentam um comportamento contrário, ainda assim seria um pouco complicado distinguir a atividade Sitting de Standing (Figura 10).

  Mas prosseguindo para a análise segundo a média do sinal verificamos que as atividades do Laying apresentam valores inferiores a 0.2 no eixo do X enquanto as atividades de Sitting e Standing apresentam valores superior a 0.7 neste mesmo eixo X. Na distinção entre a atividade Sitting e Standing verificamos que as atividades Sitting apresentam valores de Y superiores aos valores de Y na atividade Standing (na sua grande maioria) (Figura 11).

  Já segundo o declive obtido pela regressão linear pouco se conseguiu concluir quanto à distinção de atividades (Figura 12).

  Para finalizar chegámos à conclusão de que a característica espectral de eleição neste caso seria a média do sinal.
  As imagens seguintes pretendem clarificar estes resultados.

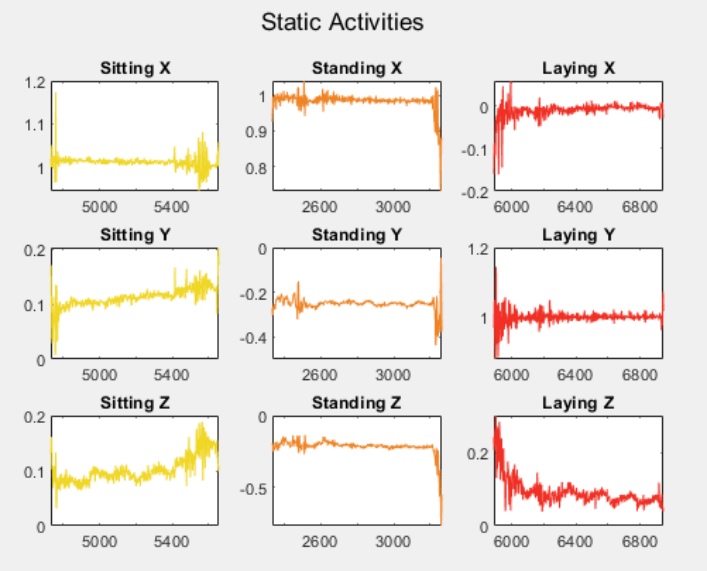

Figura 9 - Representação geral dos dados vindos das atividades estáticas

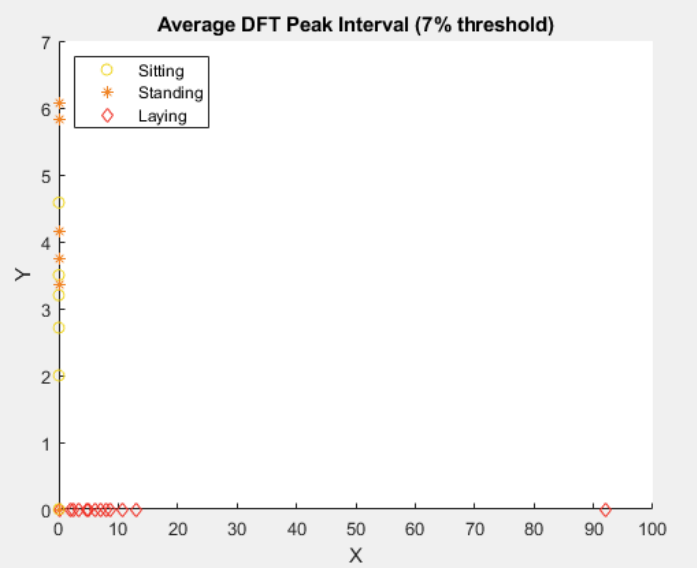

Figura 10 - Análise segundo a distância média entre os picos da DFT (com treshold de 7% do máximo)

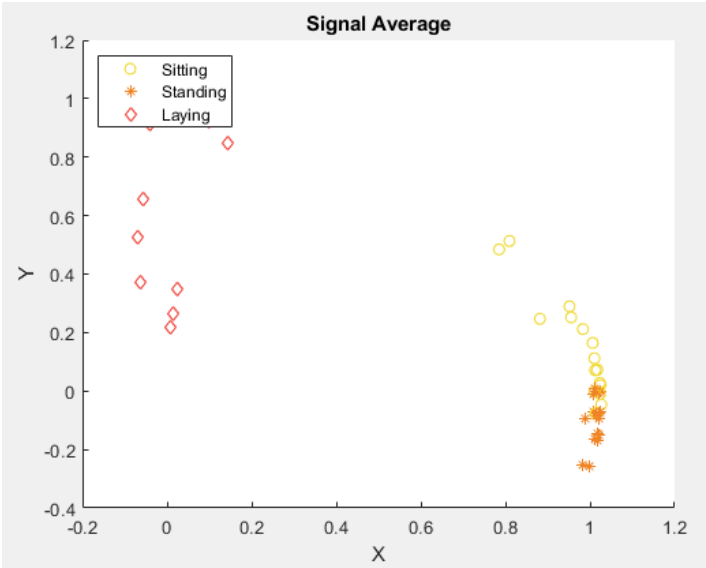

Figura 11 - Análise segundo a média do sinal

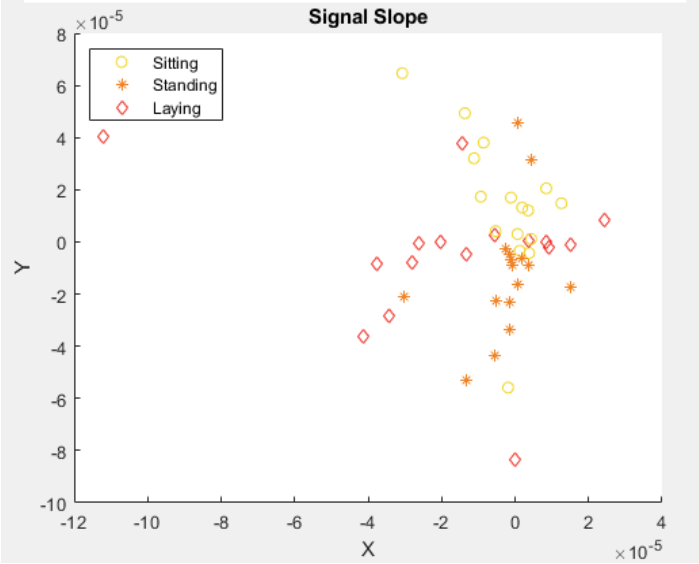

Figura 12 - Análise segundo o declive obtido pela regressão linear

- <u>Nas atividades de transição:</u>

  Iniciando pela análise segundo a distância média entre os picos da DFT (com treshold de 7% do máximo), quase à semelhança do que aconteceu nas atividades estáticas, teremos certos grupos que obedecem a certos "parâmetros". Como analisámos as atividadades Stand to Sit, Sit to Stand, Lie to Sit e Stand to Lie (com exceção de 1 outlier) essas apresentam uma componente nula no eixo X enquanto as restantes atividades, Lie to Stand (com exceção de 2 outliers) e Sit to Lie apresentam essa mesma componente não nula (Figura 14).

  Passando para uma análise segundo a média do sinal nada se conseguiu retirar para a distinção das atividades (Figura 15).

Analisando finalmente segundo o declive obtido pela regressão linear conseguimos reparar que pequenos grupos de pontos se formam, sendo cada grupo correspondente a uma atividade (Figura 16).

E chegámos ao fim de mais uma análise onde verificamos que a melhor característica para distinção de atividades de transição seria o declive obtido pela regressão linear.
As imagens seguintes pretendem clarificar estes resultados.

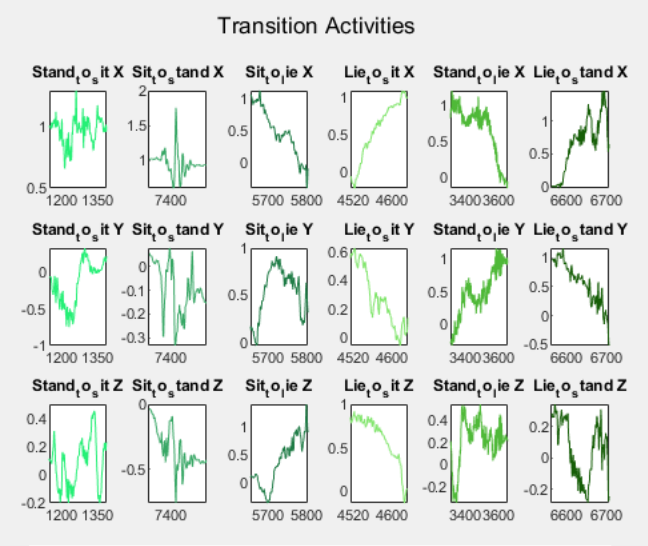

Figura 13 - Representação geral dos dados vindos das atividades de transição

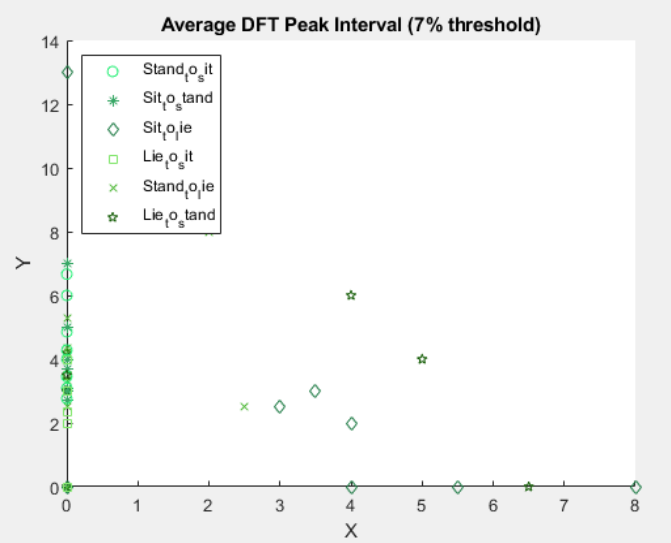

Figura 14 - Análise segundo a distância média entre os picos da DFT (com treshold de 7% do máximo)

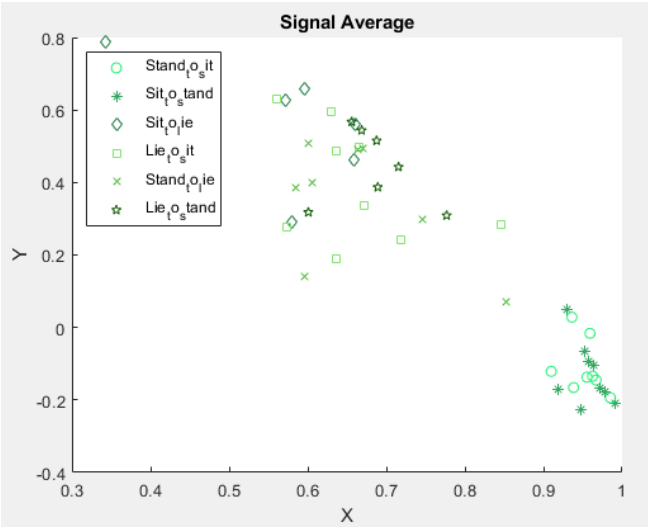

Figura 15 - Análise segundo a média do sinal

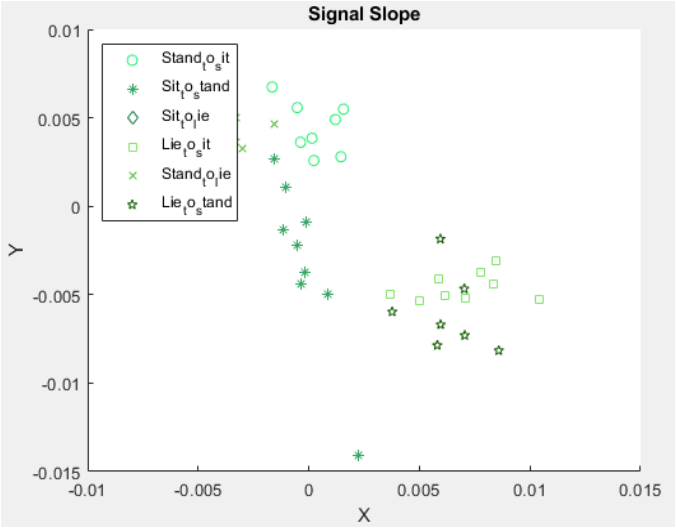

Figura 16 - Análise segundo o declive obtido pela regressão linear

Finalizamos esta análise começando por realçar a importância de ser feito um estudo por diversas características, de modo a termos uma análise mais aprofundada de como distinguir as diferentes atividades.

É ainda de realçar que observando a representação no geral das atividades (Figura 17) cada grupo de atividades apresenta algumas diferenças nas suas características. No grupo atividades dinâmicas, por exemplo, observamos altas amplitudes quase constantes (quando comparados com outros grupos) e também períodos bastante pequenos. No grupo das atividades de transição já observamos amplitudes menores, novamente quase constantes, e períodos um pouco maiores. Finalmente nas atividades de transição observamos provavelmente as maiores diferenças, com amplitudes também elas "altas", mas bastante irregulares e períodos pouco regulares que na sua maioria acabam por ser curtos.

Para terminar realçamos também que nesta análise acabamos por juntar todas as atividades de todos os utilizadores num gráfico apenas de modo a termos uma visão mais geral acerca das características espetrais de cada atividade.

Seguem-se as Figura 17, com a representação geral de todas as atividades, e as Figura 18 e 19 com uma vista geral sobre as amplitudes e o módulo o declive obtido pela regressão linear das atividades.

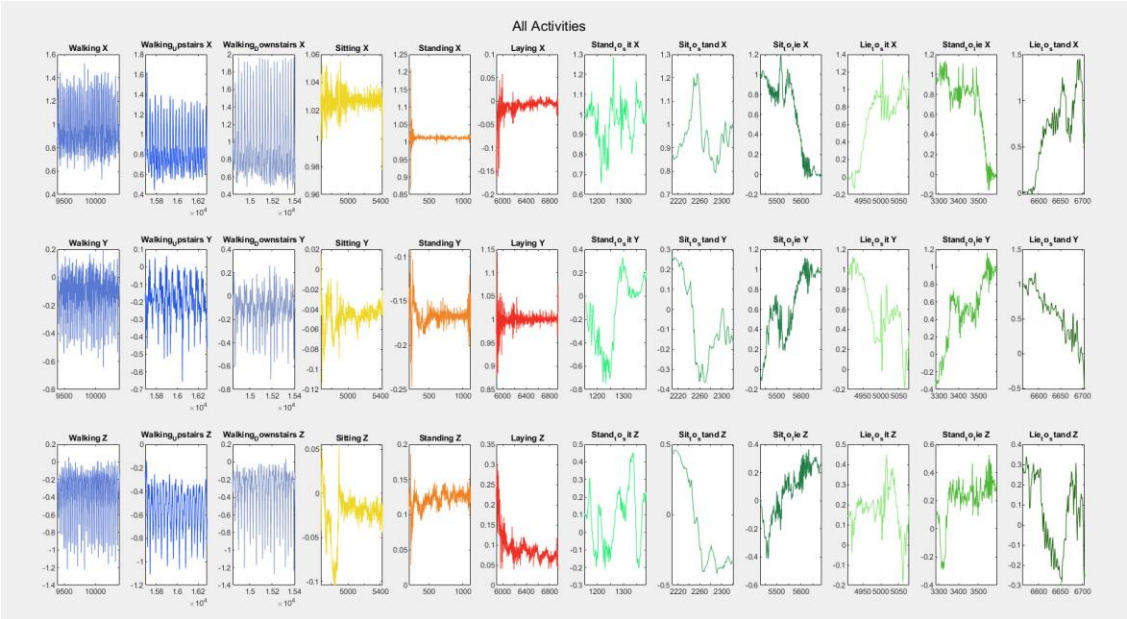

Figura 17 - Representação geral de todas as atividades

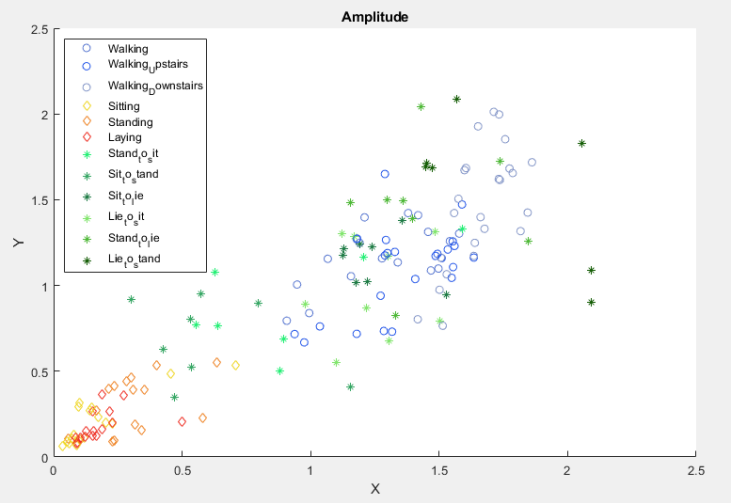

Figura 18 - Representação geral das amplitudes de todas as atividades

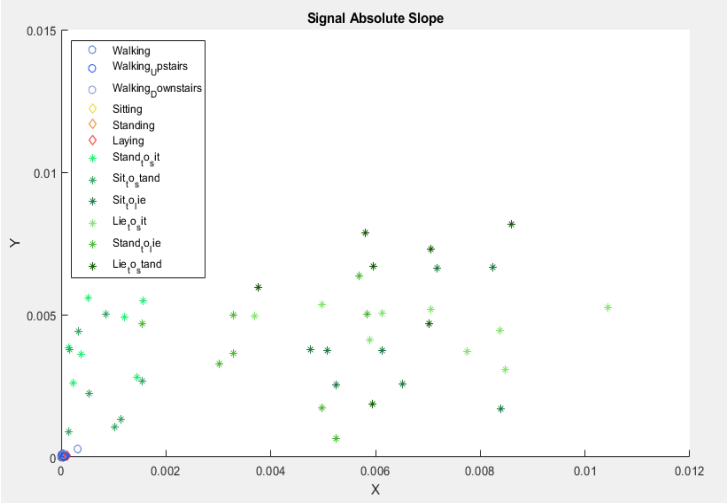

Figura 19 - Representação geral do módulo o declive obtido pela regressão linear de todas as atividades

Contagem dos passos nas atividades dinâmicas:

Para obter o número de passos por minuto médio de cada atividade dinâmica foi desenvolvida a função **steps_counter**, que devolve o número passos por segundo.

Esta função usa as 3 dimensões do sinal. De que forma? Basicamente as 3 dimensões da aceleração estão decompostas nas componentes gravitacionais e do usuário. A separação destas componentes é feita através dum *lowpass filter* de 0.2 Hz, visto que a frequência da aceleração gravítica é quase nula.

Mais tarde, a partir dum produto interno, as componentes do usuário são projetadas nas componentes gravitacionais, como resultado obteremos um único sinal que será a aceleração vertical. Finalmente calculamos a magnitude da DFT da aceleração vertical, onde o primeiro pico relevante terá uma abcissa onde se encontrará a frequência que corresponderá ao número de passos.

Iremos aplicar esta contagem de passos em todas atividades dinâmicas de todos os ficheiros (sendo que cada ficheiro corresponde a uma experiência). Por fim aplicamos a média e o desvio padrão dessas mesmas contagens. A tabela seguinte contém os resultados obtidos.

Actividade	E: 9 | U:5	E: 10 | U:5	E: 11 | U:6	E: 12 | U:6	E: 13 | U:7	E: 14 | U:7	E: 15 | U:8	E: 16 | U:9
Walking	101.68 +- 1.5	112.03 +- 0.8	105.39 +- 3.7	105.37 +- 0.3	107.21 +- 2.2	111.11 +- 0.3	88.35 +- 39.6	115.52 +- 0.2
Walking_U	96.50 +- 4.1	102.09 +- 2.6	97.14 +- 3.2	98.69 +- 1.0	99.58 +- 9.8	102.51 +- 0.4	109.92 +- 1.6	112.54 +- 4.7
Walking_D	102.19 +- 6.1	114.37 +- 5.4	107.45 +- 0.9	106.78 +- 0.3	98.63 +- 12.9	109.17 +- 4.0	109.58 +- 13.0	121.10 +- 7.7

```
%%
% 3.4)

display_names = {'Walking', 'Walking Upstairs', 'Walking Downstairs'};

for i=1:length(x)
    disp(['Média e desvio padrão no ' files{i}])
    file_ids = sscanf(files{i}, 'acc_exp%d_user%d');
    % Criar um array é para uma atividade diferente
    array1=[];
    array2=[];
    array3=[];
    for j=1:length(labels)
        if(labels(j,1)==file_ids(1) && labels(j,2)==file_ids(2))
            if(labels(j,3)==1)
                % Intervalo da atividade em questão
                interval=labels(j,4):labels(j,5);
                % Contar o número de passos
                n_steps=steps_counter([x{1,i}(interval, 1), x{1,i}(interval, 2), x{1,i}(interval, 3)], fs);
                % Multiplica para dar os passos por minuto
                array1=[array1 n_steps*60];
            elseif(labels(j,3)==2)
                interval=labels(j,4):labels(j,5);
                n_steps=steps_counter([x{1,i}(interval, 1), x{1,i}(interval, 2), x{1,i}(interval, 3)], fs);
                array2=[array2 n_steps*60];
            elseif(labels(j,3)==3)
                interval=labels(j,4):labels(j,5);
                n_steps=steps_counter([x{1,i}(interval, 1), x{1,i}(interval, 2), x{1,i}(interval, 3)], fs);
                array3=[array3 n_steps*60];
            end
        end
    end
    disp([display_names{1} ': ' num2str(mean(array1)) ' +- ' num2str(std(array1))])
    disp([display_names{2} ': ' num2str(mean(array2)) ' +- ' num2str(std(array2))])
    disp([display_names{3} ': ' num2str(mean(array3)) ' +- ' num2str(std(array3))])
    disp(" ")
end
```

Aplicação da STFT e sua representação a partir dum espectro:

Nesta parte do trabalho foi criada a função STFT que tem como objetivo aplicar a *Short Time Fourier Transform* num sinal, a partir de uma janela deslizante e os valores de tamanho e de sobreposição relativos ao tipo de janela. Esta função irá retornar a magnitude da DFT de cada aplicação da janela deslizante sobre o sinal.

A partir do espectrograma apresentado é possível visualizar, no domínio do tempo e frequência, as diferenças entre as atividades, fazendo assim em paralelo a comparação com o sinal original. Após alguns testes chegámos à conclusão de que o consenso entre a resolução temporal e de frequência estaria numa janela de 0.5% do tamanho sinal.

Ora, visualizando o espetrograma é de realçar a distinção entre as atividades estáticas e as restantes, onde basicamente as últimas contêm frequências mais altas que as primeiras.

Já nas atividades de transição é complicado distinguir entre elas qual é qual, mas comparado com os outros tipos de atividades terão uma duração bem diferente.

Finalmente nas atividades dinâmicas conseguimos distinguir de forma clara a atividade Walking pela sua duração mais elevada e baixa magnitude comparativamente com as atividades Walking Upstairs e Downstairs.

Na figura 20 é representada a comparação entre o sinal original e a STFT.

```
%%
% 4.1) e 4.2)

    % Obter o nº da experiência e o utilizador correspondente
    file_ids = sscanf(files{analyse_file}, 'acc_exp%d_user%d');
    file_labels = labels(labels(:,1) == file_ids(1) & ...
        labels(:,2) == file_ids(2),3:end);

    % Obter o vetor tempo
    N = length(x{analyse_file});
    t = (0:N-1)/fs/60;
    % Dar plot do eixo Z do sinal
    figure;
    subplot(2, 1, 1);
    plot_signal(t, x{analyse_file}(:,3), file_labels, 3, colors, activities);
    title('Original Signal');

    % Obter a stft e os vetores tempo e de frequência
    frame = fix(N*0.005);
    overlap = fix(frame/2);
    [t,f,stft] = STFT(x{analyse_file}(:,3), @hann, frame, ...
        overlap, fs);
    % Dar plot da stft
    subplot(2, 1, 2);
    imagesc(t, f, 20*log10(stft));
    set(gca, 'YDir', 'normal');
    title('Short Time Fourier Transform (STFT)');
    xlabel('Time [min]');
    ylabel('Frequency [Hz]');
    c = colorbar;
    c.Label.String = 'Power [dB]';
```

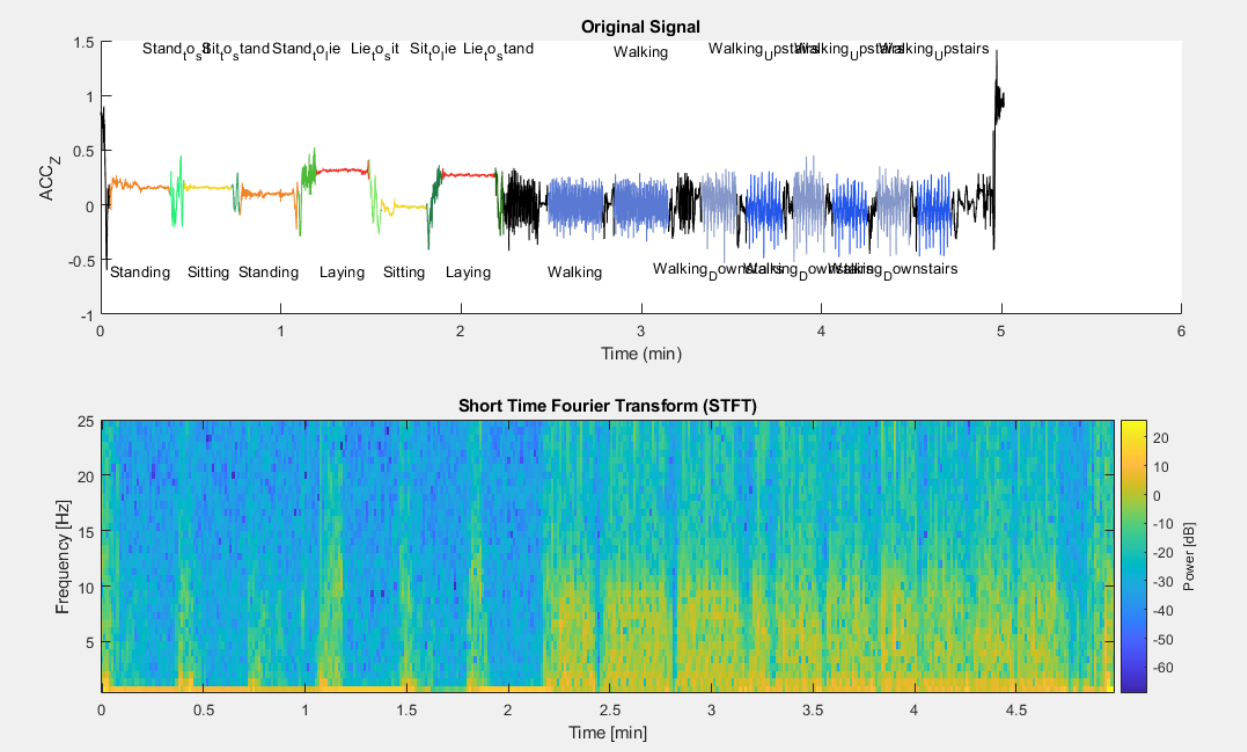

Figura 20 – Representação da comparação entre o sinal original e a STFT do ficheiro acc_exp10_user05.txt