

Manuel Fideles 2018282990  
Pedro Zenário  
Alexandre Santos

## TOC:

Introdução	_____
Tarefas 1,2,3 - Preparação	_____
Tarefa 4 - Análise	_____
Tarefa 4.1 - Calcular a DFT e comparar funções de janelamento	_____
Tarefa 4.2 - Estatísticas de passos para as atividades dinâmicas	_____
Tarefa 4.3 - Distinção de atividades dinâmicas e não-dinâmicas	_____
Tarefa 4.4 - Distinção de atividades não-dinâmicas	_____
Tarefa 4.5 - Distinção entre atividades dinâmicas	_____
Tarefa 5 - STFT	_____
Bibliografia	_____

## Introdução

No âmbito da cadeira de Análise e Transformação de Dados, foi-nos proposto analisar uma gama de 12 atividades através de sinais detetados pelo giroscópio e acelerómetros de um telemóvel colocado na cintura. Estas atividades podem-se dividir em 3 grupos distintos, dinâmicos (Walking (**W**), Subindo Escadas (**WU**), Descendo Escadas (**WD**)), estáticos (Sentado (**SIT**), De pé (**STAND**), Deitado (**LAY**)) e por fim de transição (De pé - Sentado (**STAND2SIT**), Sentado - De pé (**SIT2STAND**), Sentado - Deitado (**SIT2LIE**), Deitado - Sentado (**LIE2SIT**), De pé - Deitado (**STAND2LIE**), Deitado - De pé (**LIE2STAND**)).

Estas atividades foram todas captadas com um Dataset SBHAR com 2 acelerómetros e 1 giroscópio num smartphone Samsung S2, recolhendo dados dos 3 eixos espaciais com uma frequência de amostragem, **fs**, de 50 Hz.

Todos os gráficos aqui exibidos são referentes à experiência 1 (à exceção das tarefas onde são precisos dados de todas as tarefas).

## Tarefas 1, 2 e 3 - Preparação

Para recolher os dados fornecidos no *dataset*, foi desenvolvido o método *openExperimentFile()*, que abre o ficheiro da experiência e retorna as medições que se encontram em cada uma das 3 colunas de dados. É também recolhida a informação acerca das labels e dos intervalos a que pertence cada uma das atividades.

Os dados de cada experiência são depois enviados para o método *plotExperiment()*, que, como o nome indica, gera os gráficos das experiências com o devido labeling e identificação de cada atividade.

## Tarefa 4 - Análise

### Tarefa 4.1 - Calcular a DFT e comparar funções de janelamento

Antes de fazer qualquer tipo de análise, procedemos à separação do sinal inteiro nas atividades que o compõem, de maneira a analisar cada atividade isoladamente. Retiramos também a componente da tendência a cada atividade, usando o método *detrend()*. Estas duas operações são feitas no método *getActivityVectors()*, que devolve um *cell array* que contém as medições para cada ocorrência e para cada eixo (por ex: W tem dimensão 1x3, em que cada elemento é um eixo (x, y, z) e dentro de cada eixo tem a informação referente a cada ocorrência de W no sinal *raw*). Os sinais são depois agrupados nas 3 categorias que os distinguem : *DINAMIC*, *STATIC*, *TRANSITION*.

As DFTs de cada sinal são obtidas com o método *calcActivityDFT()*, que recebe as categorias de atividade.

No método *viewThroughWindow()* podemos aplicar 2 tipos diferentes de janela às atividades - Hamming e Hann (para além da retangular, que é *default*).

Para decidir qual das duas seria a mais apropriada, aplicámos cada janela aos sinais de 3 atividades representativas da categoria onde se inserem (fig. 1 até 3) e às suas respetivas DFTs (fig. 3 até 6) - **W (dinâmica)**, **SIT (estática)**, **STAND2SIT (transitória)**.

As imagens que se seguem são referentes ao eixo dos xx.

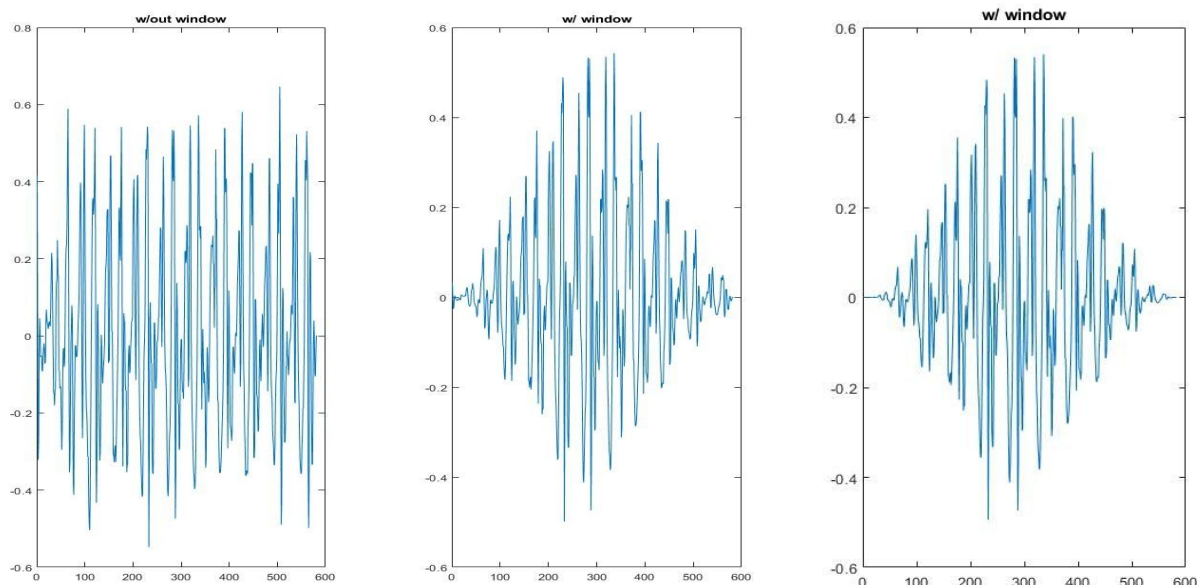


Figura 1 - (da esquerda para a direita) W s/ janelamento, c/ janela de Hamming, c/ janela de Hann

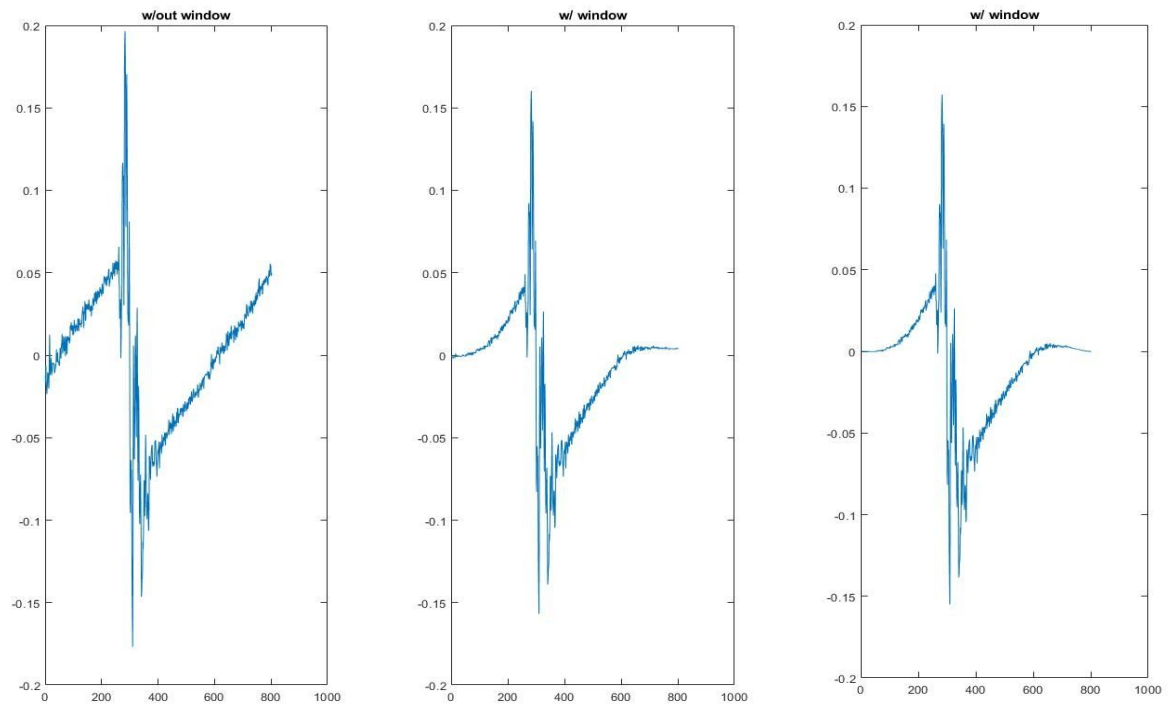


Figura 2 - (da esquerda para a direita) **SIT** s/ janelamento, c/ janela de Hamming, c/janela de Hann

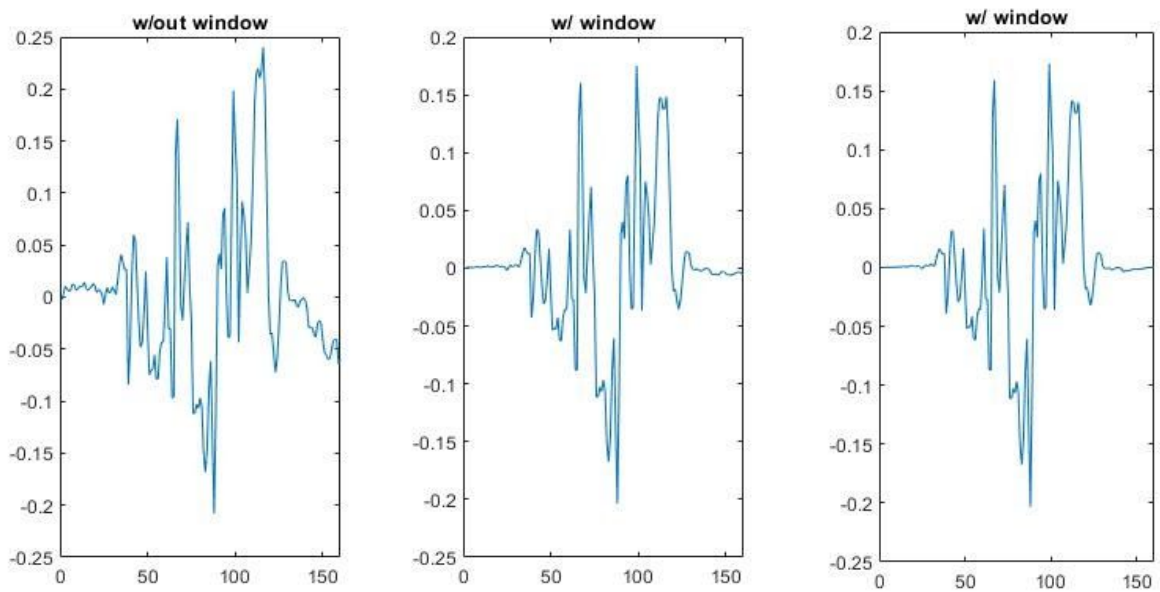


Figura 3 - (da esquerda para a direita) **STAND2SIT** s/ janelamento, c/ janela de Hamming, c/janela de Hann

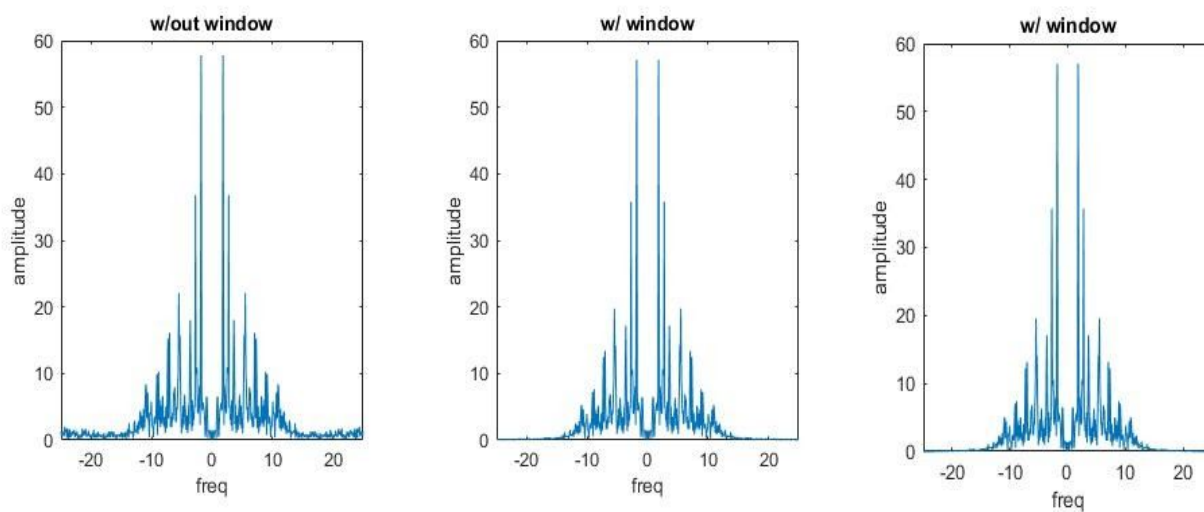


Figura 4 - (da esquerda para a direita)  $DFT$  de  $W$  s/ janelamento, c/ janela de Hamming, c/ janela de Hann

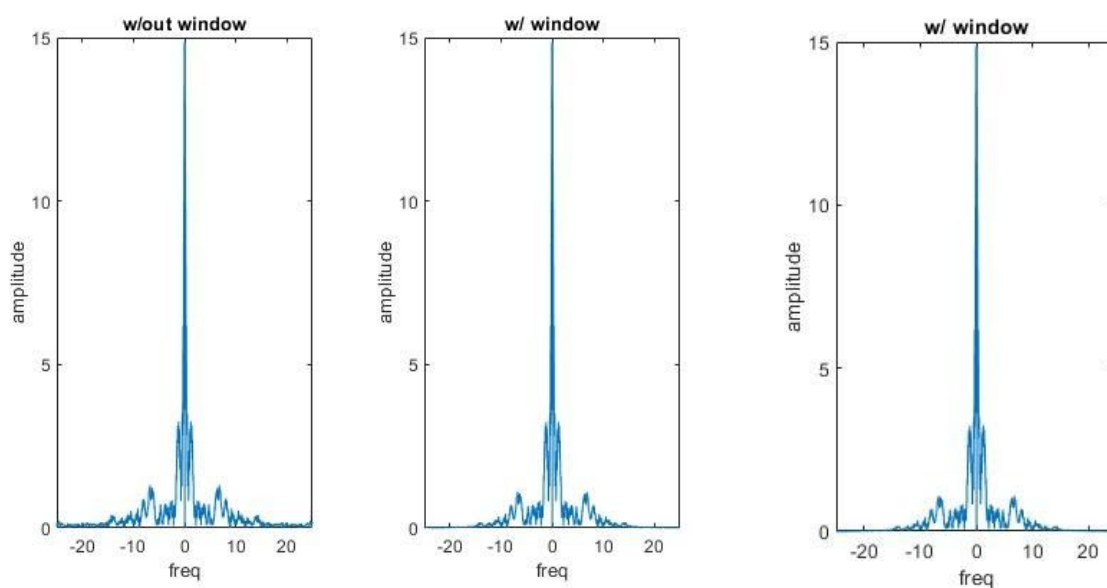


Figura 5 - (da esquerda para a direita)  $DFT$  de  $SIT$  s/ janelamento, c/ janela de Hamming, c/ janela de Hann

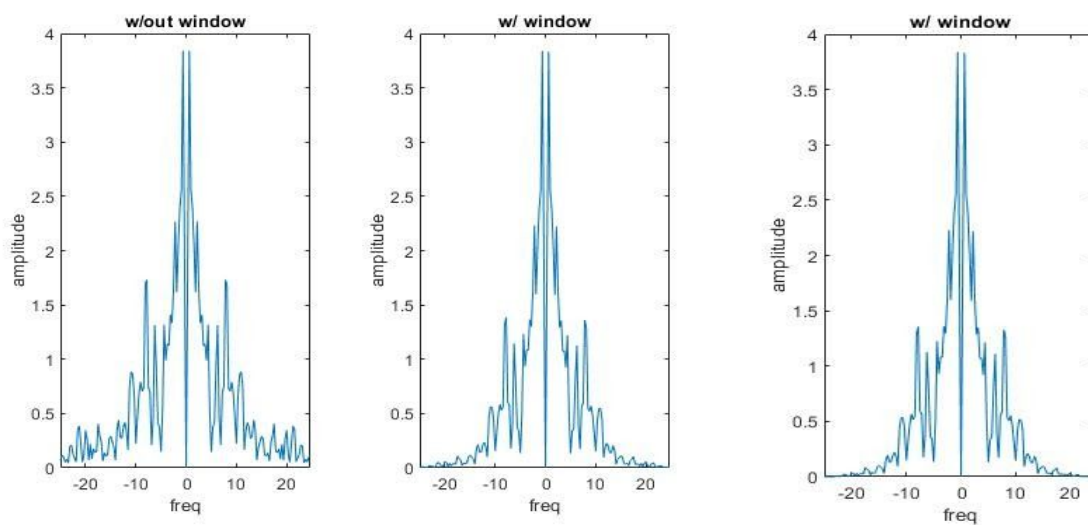


Figura 6 - (da esquerda para a direita)  $DFT$  de  $STAND2SIT$  s/ janelamento, c/ janela de Hamming, c/ janela de Hann

Após uma análise macroscópica, estávamos insatisfeitos com os resultados pouco esclarecedores obtidos nos gráficos acerca de qual janela escolher, e depois de alguma pesquisa, observámos e aprendemos o seguinte:

Há dois fatores de grande importância que se tem que ter em conta ao escolher uma função de janelamento: a *main lobe width* - que afeta diretamente a resolução do espectro quando a janela é aplicada às DFTs (em termos práticos, menor é melhor) - e o *peak side lobe level* - que impacta a capacidade de reconhecer frequências adjacentes quando estamos na presença de *spectral leakage* (quando a energia de um sinal é “derramada” para outras frequências - quanto menor, melhor). A grande diferença entre as duas janelas que são aqui comparadas é o facto de que a de Hann elimina qualquer tipo de descontinuidade da entrada, pois toca em 0 nas suas extremidades (fig 8), enquanto que na janela de Hamming o mesmo não se verifica (fig 7). Assim sendo, decidimos optar por usar a janela de Hamming pois é que a nos oferece o equilíbrio mais satisfatório entre *leakage factor* e *main lobe width* (tabela 1). Em boa verdade, qualquer uma das opções seria apropriada para o contexto.

Para os gráficos e tabela que se seguem, utilizámos o método *wvtool()*.

Janela	Leakage Factor (%)	Main Lobe Width
Hamming	0.036	0.0057372
Hann	0.05	0.0073243

Tabela 1 - Leakage factor e main lobe width médios para as 3 atividades

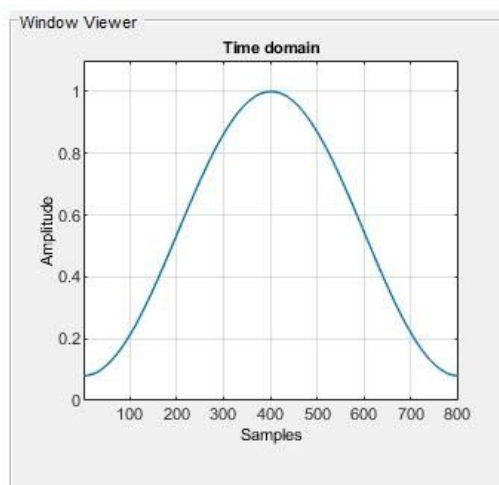


Figura 7 - Janela de Hamming

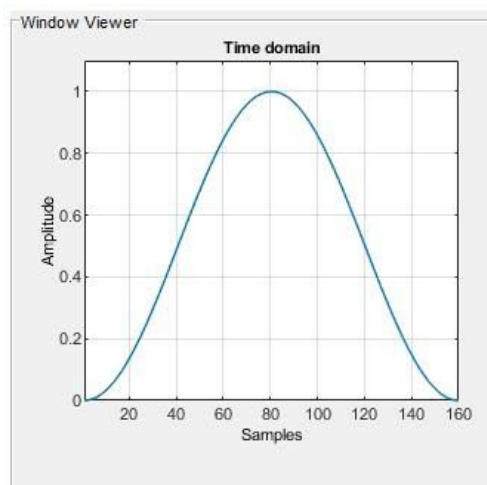


Figura 8 - Janela de Hann

## Tarefa 4.2 - Estatísticas de passos para as atividades dinâmicas

Para calcular as estatísticas de passos relativamente às atividades dinâmicas, desenvolvemos o método `getSPM()`, que recebe a DFT de uma atividade dinâmica e retorna a frequência com maior amplitude apenas no eixo dos xx, em passos por minutos. O valor calculado é adicionado a um *array*, onde estão guardadas os *steps-per-minute* de cada experiência. No fim de todas as experiências, são corrigidos os *outliers* dos dados em questão com recurso a *'pchip'*, utilizando o método `filloutliers()`. É depois calculada a média e desvio padrão de cada atividade (com os métodos `mean()` e `std()`, respetivamente).

Decidimos usar apenas os valores do deslocamento no eixo xx pois é nesse eixo que grande parte do movimento se processa. Como é explicado nos vídeos (falta link) que acompanham o artigo, os sujeitos andam em linha reta, sem deslocamento lateral ou vertical (excetuando WU e WD, que apresentam deslocamento vertical) e, como tal, as medições nos eixos yy e zz descriminam com menos fidelidade a frequência da passada. Na tabela 2 estão representados os valores obtidos.

	mean	std
W	104.8592	5.6278
WU	96.5820	14.3549
WD	101.1371	4.8450

Tabela 2 - Frequência média e desvio padrão (em SPM) para cada atividade dinâmica, no eixo dos xx

Citando o artigo de onde é oriundo o *dataset*,

*“The cadence range of an average person walking is [90, 130] steps/min [BenAbdelkader et al., 2002] which denotes a minimum speed of 1.5 steps/sec”*

e, assim, concluímos que os valores obtidos estão dentro do esperado.

## Tarefa 4.3 - Distinção de atividades dinâmicas e não-dinâmicas

Para esta tarefa, foi desenvolvido um método - `plotMaxFrequency()` -, que, utilizando a mesma abordagem da tarefa anterior, calcula a amplitude máxima das DFTs de cada tipo de atividade, para os 3 eixos. Os valores retornados por essa função são depois adicionados a arrays próprios para cada eixo e tipo de atividade (*maxFreqDinamicX/Y/Z* e *maxFreqNonDinamicX/Y/Z*). No fim de todas as experiências serem processadas, o programa executa o método `plotAllMaxFrequencies()`, que recebe os arrays preenchidos anteriormente e representa graficamente os valores neles contidos (fig. 9).

Para além desta abordagem, desenvolvemos também outro método, chamado `plotMaxAmplitudes()` e, analogamente, o método `plotAllMaxAmplitudes()`. Os resultados desta abordagem estão representados em baixo (fig. 10), mas como se pode ver, os resultados por si são inconclusivos. Pensámos que talvez pudesse ser usado como método complementar ao que foi descrito na primeira abordagem, quiçá para cruzar a informação obtida nos dois métodos e poder avaliar melhor uma futura entrada no *dataset*.

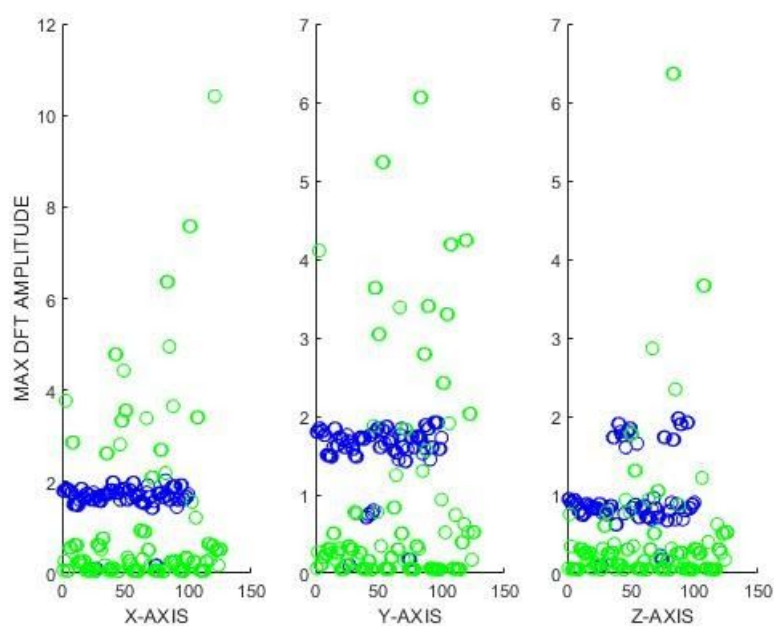


Figura 9 - Amplitude máxima das DFTs de cada atividade, nos eixos xx, yy, e zz. A verde, as não-dinâmicas e a azul, as dinâmicas.

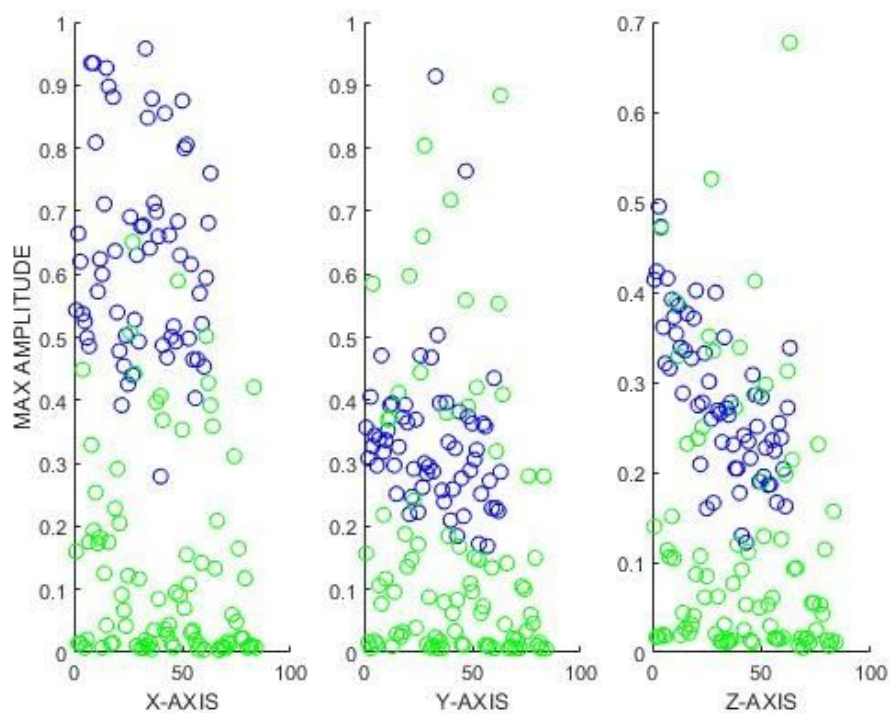


Figura 10 - Amplitude máxima do sinal original de cada atividade, nos eixos xx, yy e zz. A verde, as não-dinâmicas e a azul, as dinâmicas.



A nível de sensibilidade, podemos afirmar que uma nova entrada que se encontre dentro das zonas assinaladas pode ser considerada uma atividade dinâmica.

#### Tarefa 4.4 - Distinção entre atividades estáticas e transicionais

Olhando para o sinal original, podemos imediatamente observar que as atividades de transição são muito mais curtas que as atividades estáticas (fig. 11). Assim, usámos a sua duração como métrica de comparação. Desenvolvemos o método *plotDuration()*, que recebe os sinais *raw* das atividades estáticas e transicionais para obter a duração de cada atividade. Os valores calculados nesse método são depois inseridos num *cell array* que guarda todas as durações para os dois tipos de atividade (*staticDuration* e *transitionDuration*).

Depois de processadas todas as experiências, as durações são representadas graficamente, com o auxílio do método *plotAllDurations()*. Os resultados desta abordagem estão representados na figura 12.

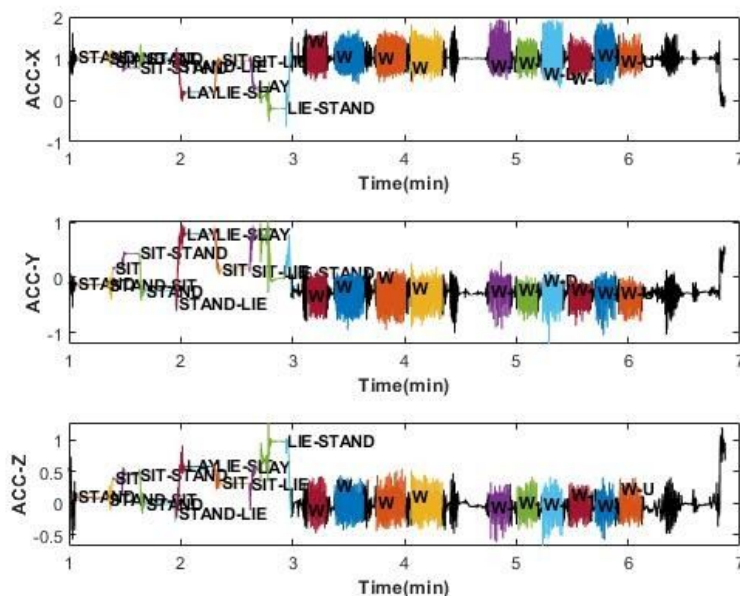


Figura 11 - Sinal original. Note-se a diferença de duração entre STAND e LIE-STAND, por exemplo.

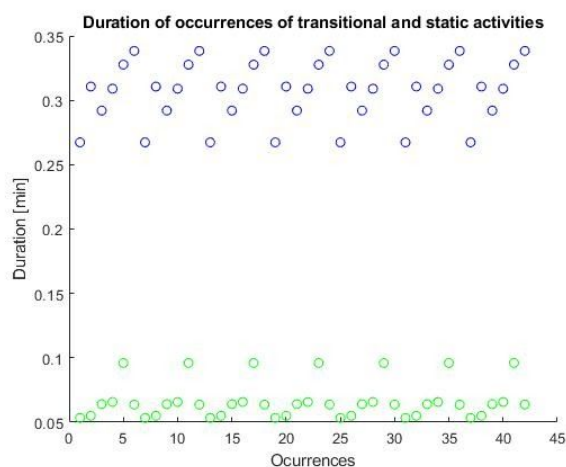


Figura 12 - Durações dos dois tipos de atividades objeto de comparação

#### Tarefa 4.5 - Distinção entre atividades dinâmicas

Ao representarmos graficamente as DFTs de cada atividade nos três eixos, verificamos que havia uma diferença significativa entre eixos nas amplitudes máximas das DFTs (fig 13 até fig 15). Como tal, decidimos usar essa característica como métrica de comparação entre atividades dinâmicas. Abaixo estão representados os resultados obtidos para a experiência 1.

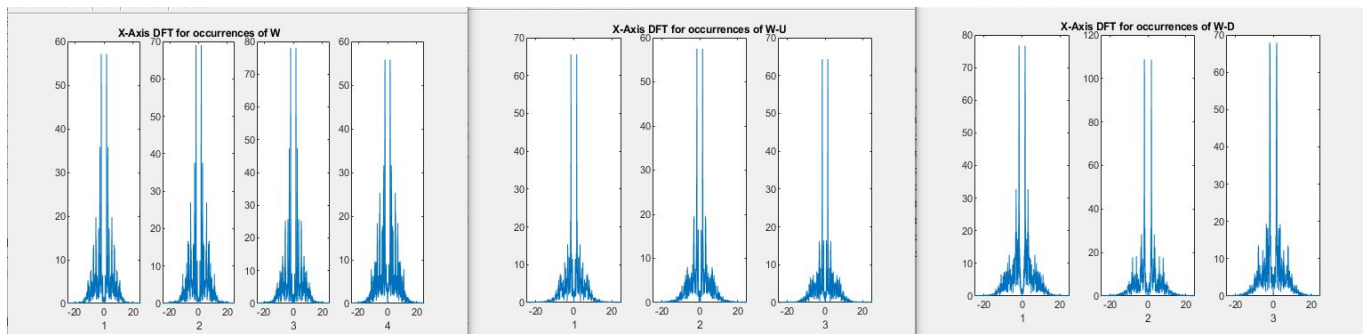


Figura 13 - DFTs das atividades dinâmicas no eixo xx

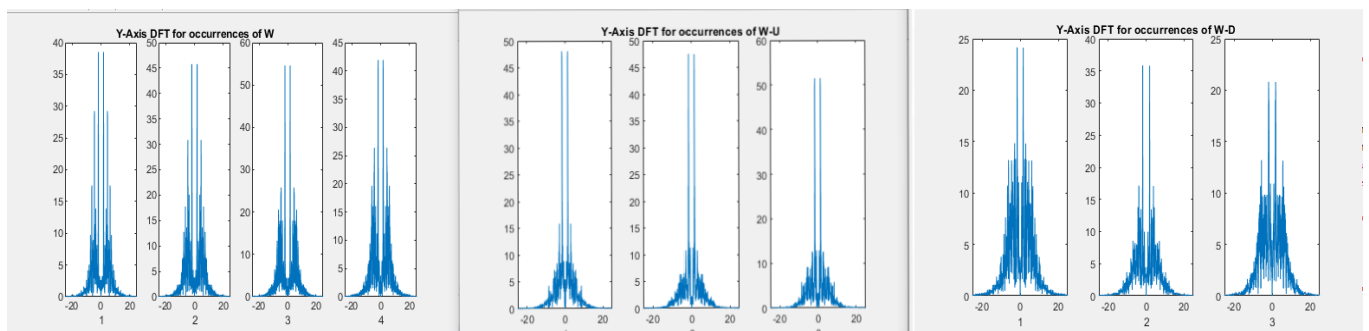


Figura 14 - DFTs das atividades dinâmicas no eixo yy

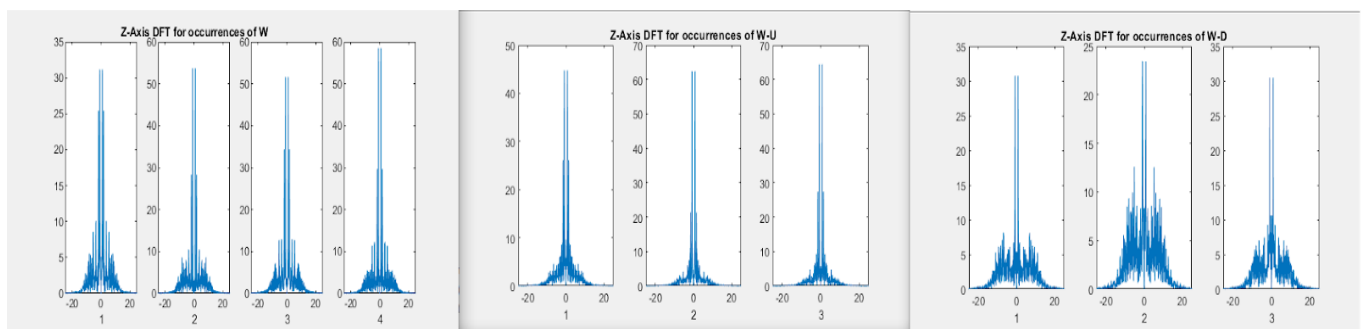


Figura 15 - DFTs das atividades dinâmicas no eixo zz

Assim, utilizámos o método *plotMaxDFTAmplitude()* para obter as maiores amplitudes da DFT nos eixos de cada atividade. Os resultados estão dispostos na tabela 3. Os valores a verde são os menores da sua linha (cada linha corresponde a uma experiência), e os vermelhos são os maiores. Os ficheiro das experiências 8 e 10 não foram considerados por incapacidade de serem abertos pelo programa.

Wx	WUx	WDx		Wy	WUy	WDy		Wz	WUz	WDz
77,9259	65,6059	108,7709		54,6113	51,5301	35,8423		58,5186	64,4061	30,736
59,2498	79,6331	63,5323		51,7206	35,906	53,9715		39,3061	25,605	58,5062
62,741	37,5092	66,3307		44,7406	25,9461	43,9556		35,5201	30,6474	35,3205
56,2915	55,715	49,7914		38,7303	35,0901	35,6547		25,843	30,9578	26,861
46,086	47,6392	49,5359		35,4251	21,9651	32,8304		20,7881	18,1164	29,3327
74,2225	44,6518	59,2845		78,9504	21,6627	38,6858		56,8861	21,5172	23,4692
59,2845	59,0387	74,2269		52,8125	22,4328	33,3288		19,9652	22,1448	30,0198
-	-	-		-	-	-		-	-	-
47,2313	117,2805	85,1341		35,0948	22,212	45,6718		31,056	44,2342	32,3477
-	-	-		-	-	-		-	-	-

Podemos verificar que, no eixo dos yy, se forma uma tendência. Na verdade, usando este método, podemos distinguir as atividades dinâmicas umas das outras em 75% dos casos que se encontram nos dados fornecidos.

## Tarefa 5 - STFT

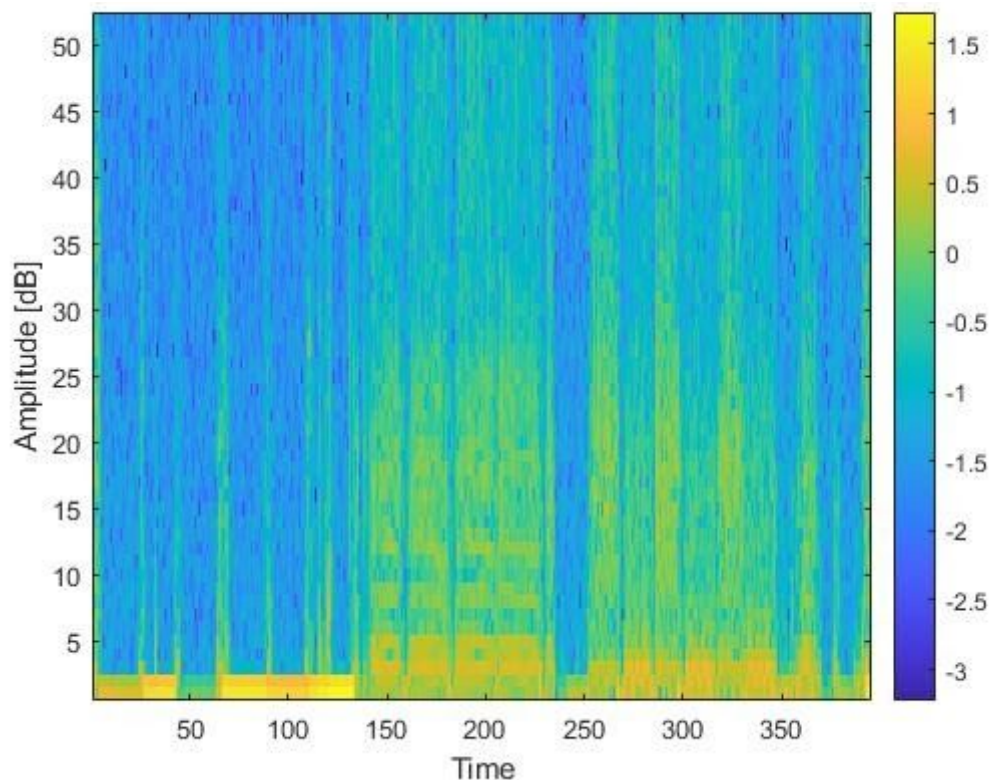


Figura 16 - Espetro do sinal oriundo do acelerómetro z (amplitude em dB)

Para esta tarefa, desenvolvemos o método *getSTFT()* - que recebe o sinal do acelerômetro do eixo zz da experiência 1 -, largamente apoiados na função desenvolvida nas aulas práticas. Utilizamos uma *time frame* de  $0.005 * t$ , sendo  $t$  o tempo de amostragem do sinal, e com *overlap* de 0.5 (figura 16). Importante referir que esta abordagem pode também ser usada para completar a tarefa 4.5 (inclusive com maior eficiência).

## Bibliografia

<https://pdfs.semanticscholar.org/ed2c/d4d99795a83a429d733e5cae5fab2dfb868f.pdf>  
<https://dsp.stackexchange.com/questions/20454/why-are-there-so-many-windowing-functions>  
<https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf>  
[https://en.wikipedia.org/wiki/Window\\_function](https://en.wikipedia.org/wiki/Window_function)  
[https://www.youtube.com/watch?v=XOEN9W05\\_4A](https://www.youtube.com/watch?v=XOEN9W05_4A)