

Interface Digital Para Amputados Dos Membros Superiores

António Figueira, *ist1102448*, João Freitas, *ist1103218*, e Guilherme Costa-Ferreira, *ist1103394*

I. INTRODUÇÃO

EM 2022, nos Estados Unidos da América (EUA) cerca de 2 milhões de americanos sofreram de algum tipo de perda de um dos membros, quer fosse por doenças (diabetes mellitus, por exemplo), como por acidente ou guerra. Prevê-se que estes números aumentem para 3.6 milhões em 2050.

A interseção entre tecnologia de assistência e a melhoria da acessibilidade digital para pessoas com amputações de parte de membros superiores tem sido um campo de pesquisa em crescimento. Embora as próteses sejam frequentemente consideradas a solução mais completa, não resultam necessariamente em maiores níveis de satisfação entre os indivíduos com membros amputados. [7]. Por essa razão, tecnologias de assistência continuam a ser desenvolvidas.

Em 2020, um estudo utilizou um giroscópio e um eletromiograma (EMG) para desenvolver um rato de computador. O giroscópio, capaz de medir a orientação e a velocidade angular em três eixos, foi empregue para medir a orientação da mão e codificar o controlo do cursor no ecrã, enquanto o EMG nos músculos bíceps e tríceps codificou as funcionalidades do rato. Além disso, a performance do dispositivo foi aprimorada com o uso de algoritmos de *auto-thresholding* e deteção de contração muscular [5].

Em 2022, uma interface chamada EMKEY foi concebida para simular funções do rato de computador e do teclado por meio do reconhecimento facial e da voz, em substituição ao uso das mãos. Esta interface é dividida em dois módulos: um responsável por traduzir os movimentos faciais em movimento do cursor, e o segundo módulo dedicado ao reconhecimento de comandos de voz e transcrição para texto [6].

As tecnologias de assistência mencionadas fazem parte do campo da interface pessoa-máquina, que se torna cada vez mais acessível para pessoas com amputações [2].

No entanto, é importante notar que essas tecnologias ainda não estão suficientemente avançadas para serem usadas de forma confiável no quotidiano. Portanto, o desenvolvimento de dispositivos com funcionalidades mais especializadas torna-se fundamental.

II. CONCEITO INICIAL

Originalmente, propusemos a conceção de uma interface digital recorrendo a um sensor EMG e um acelerómetro, aproveitando o movimento do membro residual nos diferentes eixos (e respectivas direções) que o compõem e a contração muscular para se traduzir em ações efetivas do controlo do rato. Consoante a ação, seria possível manipular as diferentes

funcionalidades na interface digital (movimento, clique e criação de menus, por exemplo). A escolha do membro superior por parte do utilizador para manobrar o dispositivo levaria à utilização de uma meia que incorporaria o acelerómetro (garantindo a sua estabilização na zona distal do braço) e ao longo do membro residual estariam implantados os sensores EMG.

III. PROOF OF CONCEPT

Antes de procedermos à explicação da abordagem tomada, é de referir que, ao longo do tempo, foi necessária a adaptação do plano de trabalho a seguir, de modo a tornar o projeto concretizável consoante os recursos disponíveis. De qualquer forma, todas as alterações são descritas neste relatório e continuam a focar-se no objetivo inicial: restituir o *workflow* e a importância no local de trabalho dos amputados dos membros superiores.

Assim, desenvolveu-se como protótipo final o representado na figura 1.

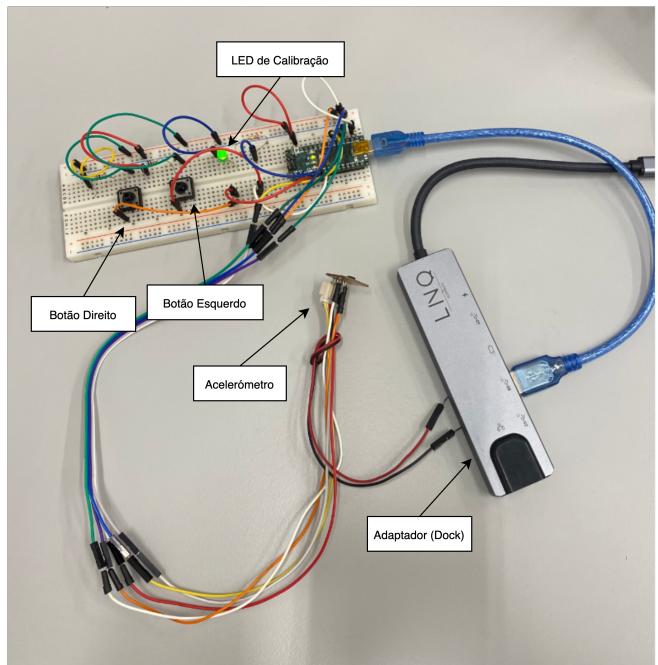


Fig. 1. Imagem legendada do protótipo

Entendemos que não seria necessário utilizarmos a abordagem do EMG. As funções essenciais num rato comum de computador estão asseguradas apenas pelo uso do

acelerómetro (movimentos) e botões (cliques). Outra alteração substancial à proposta inicial é a utilização da meia, algo que não concretizámos pois, contextualizando no ambiente de trabalho da *breadboard* e das componentes acrescentadas (fios e botões, por exemplo), não seria prático e, de certa forma concretizável, integrar todo este dispositivo na extremidade do membro residual. Logo, criamos um *proof of concept* relativo à solução apresentada.

Para indicar ao utilizador que o protótipo encontra-se em funcionamento, a LED verde encontra-se ligada e, aquando momentos de calibração, transita para um estado intermitente.

Para concluir esta secção, é importante reconhecer a formulação do problema deste caso específico. Ao recorrermos a um acelerómetro, os dados obtidos deste - após a conversão da voltagem a níveis discretos (entre 0 a 1023) no Analog to Digital Converter (ADC) do Arduino Nano - podem ser expressos por um vetor a cada instante n :

$$\mathbf{x}(n) = [x_0, x_1, x_2]^T \quad (1)$$

em que cada componente do vetor corresponde às acelerações que se registam nos 3 eixos principais (x , y e z), respetivamente.

IV. CIRCUITO

De modo a tornar visual a estrutura do circuito, apresentamos uma imagem do ThinkerCAD que simula a versão física da *breadboard* juntamente com os vários componentes. Tal representação encontra-se na figura 2.

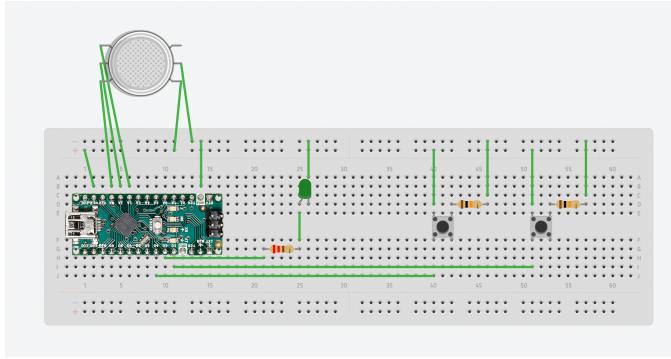


Fig. 2. Esquema do Circuito

Devido às limitações do ThinkerCAD, utilizamos um sensor de fumo como sensor análogo (possui o mesmo número de entradas/saídas).

V. PROCESSO

É importante reportar que, de forma a existir um fio condutor que guiasse o raciocínio do grupo, fomos baseando os nossos passos nas apresentações das aulas disponibilizadas pelo corpo docente, quer teóricas, quer laboratoriais, evitando esquecer os métodos fulcrais para uma correta aquisição, processamento e análise de informação.

Com efeito, ao investigarmos o modo de funcionamento do acelerómetro, percebemos que a forma mais indicada de aproveitarmos este dispositivo tendo em conta as funcionalidades do *mouse*, baseia-se na variação das acelerações

associadas ao movimento nas suas diferentes componentes, ou seja, nas direções dos 3 eixos principais x , y e z - a outra abordagem de utilização do acelerómetro consiste na aquisição exclusiva de variações relativas ao eixo z (abordagem *default*). Deste modo, o mecanismo descrito assemelha-se ao de um *joystick*, sendo por isso uma inclinação do acelerómetro para a frente ou para trás (eixo x) associados a um movimento ascendente ou descendente na interface digital, respetivamente. De um modo semelhante, inclinações para a esquerda ou para a direita (eixo y) correspondem à mesma deslocação no ecrã para os mesmos sentidos.

O próximo ponto a abordar debruça-se na conceção do código associado ao Arduino IDE e Python.

Relativamente ao código do Arduino IDE, estão presentes diversas funções que, de um modo resumido, são responsáveis pela recolha: dos valores do sinal obtido pelo acelerómetro e respetiva marcação temporal mas também de informação sobre os botões e respetiva atividade. Estes dados são enviados para o Python. No *loop* também há transmissão de *data* no sentido inverso (Python - Arduino IDE) em que, caso o acelerómetro esteja no momento de calibração, o Arduino recebe informação para alterar o estado do LED presente na *breadboard*.

A extração de dados do acelerómetro e consequentes processamento e estudo do sinal foram desenvolvidos em Python, tal como a interação com o *mouse*. Assim, podemos dividir esta secção em dois grupos: execução da ação e estudo do sinal - sendo que este permitiu melhorar o primeiro no decorrer do projeto através de diferentes parâmetros e métodos.

A. Processamento de Sinal

Através do estudo do sinal foi possível adequar a nossa abordagem ao problema. Para tal, agrupámos todas as suas funções numa função *loadData* que, desde o início da sua criação, permitiu-nos retirar conclusões sobre o nosso sinal através da elaboração de um gráfico de densidade espectral (PSD: Power Spectral Density), onde se representa a proporção do contributo de cada frequência para a potência total do sinal. Neste caso de estudo, os movimentos realizados são caracterizados por frequências baixas (inferiores a 1Hz), incentivando assim ao uso de um filtro passa-baixo (com frequência de corte igual a um valor compreendido entre 1 e 5Hz). Como é possível observar na seguinte figura, a filtragem contribui para a suavização do sinal e a obtenção de um resultado livre de ruído e artefactos indesejáveis (figura 3).

Tratando-se de um sinal obtido pelas variações de aceleração, as unidades comumente utilizadas são m/s^2 ou g's. Como o sinal enviado pelo Arduino é ADC, realizando um exercício de calibração (rotação lenta sobre o eixo z), podemos converter as unidades.

$$\text{ACC(g)} = \frac{\text{ADC} - C_{\min}}{C_{\max} - C_{\min}} \times 2 - 1 \quad (2)$$

Utilizando a equação 2 e substituindo o ADC pelos valores do sinal, o C_{\max} e o C_{\min} pelos valores dos extremos, obtivemos os resultados na figura 4.

Através desta abordagem, percebemos que a melhor forma de discretizar o sinal seria através de um movimento *joystick*

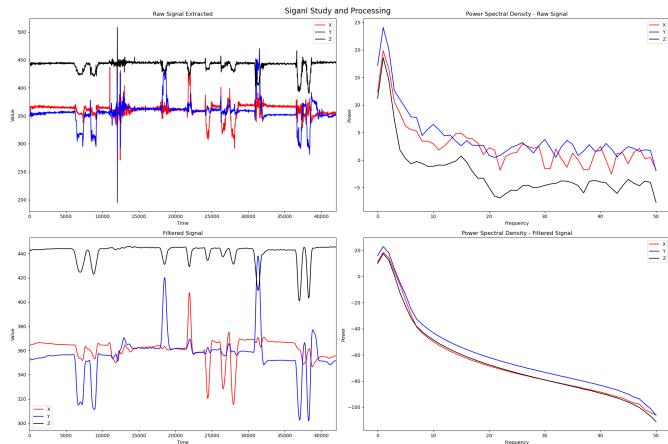


Fig. 3. Resultados com passa-baixo com uma frequência de corte de 1Hz



Fig. 4. Exercício de Calibração

onde cada ação teria uma leitura clara e distinta. Tal ficou definido após analisar os dados presentes na figura 5. Assim, seria necessário acrescentar *thresholds* e, consequentemente *baselines* (obtidos através de um período de calibração), ao nosso sistema.

Nas primeiras iterações de aquisição de sinal, verificou-se *delay* acumulativo na propagação do mesmo, visível no movimento prolongado do *mouse*. Este fenómeno deve-se ao tempo que os sistemas a operar necessitam para receber e ler as informações que são comunicadas entre eles que, mesmo sendo muito reduzido, vai acumulando face às inúmeras iterações deste processo [9]. Para a resolução holística de tal contratempo, foi diminuída a frequência de amostragem (de 100Hz para 10Hz) e, consequentemente, registou-se a redução da quantidade de informação que chega ao computador num determinado intervalo de tempo, reduzindo a magnitude deste problema. Esta alteração não comprometeu o nosso sinal, respeitando a lei de Nyquist - a frequência de amostragem deve ser no mínimo o dobro do valor da frequência máxima do sinal analógico (como referido anteriormente, inferior a 1Hz) [8]. De tal forma obtemos apenas resultados onde o gráfico apresenta-se menos "denso". Como evidenciado na figura 6.

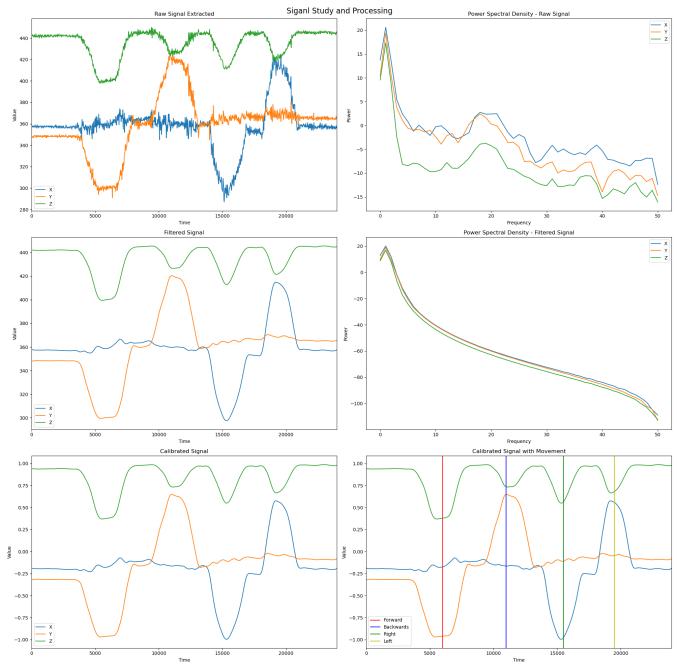
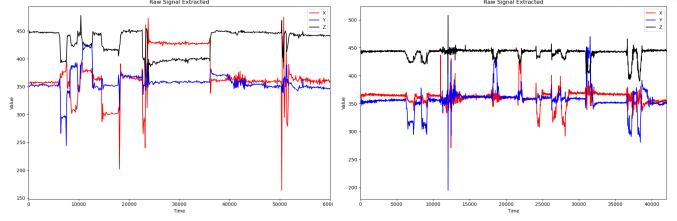


Fig. 5. Análise do teste de movimentos

Fig. 6. Comparação entre uma *test run* com 10Hz (esquerda) com outra de 100Hz (direita)

B. Execução do Sinal

A execução da ação baseia-se exclusivamente na função *extractData*. Como o próprio nome indica, é responsável pela obtenção de dados a partir do Arduino e na tradução em movimento e/ou clique do *mouse*.

Como concluído na fase anterior, necessitamos de utilizar *thresholds* (para garantir que só ocorre movimento quando desejado) e, como tal será necessário uma *baseline* para os definir. Assim, no decorrer deste processo de tradução, há uma etapa inicial de calibração que, durante os primeiros 5 segundos de ação do protótipo, são calculadas as respetivas médias de valores (por outras palavras, as *baselines*) para cada eixo (x, y e z).

É de notar que a direção dos movimentos pode ser inferida pelas oscilações dos valores x e y e que, independentemente da direção, uma diminuição dos valores z acontece.

As *baselines* permitirão o cálculo dos *thresholds* que serão essenciais no controlo do movimento digital consoante o movimento físico. Uma *baseline* resultará em dois *thresholds* que, aquando do registo do sinal, incitarão a ação do *mouse* caso exista a passagem destes limites em dois dos eixos (num dos eixos direcionais e em z). Garante-se assim que movimentos acidentais como o reposicionamento do braço

para uma posição mais adequada/confortável para o utilizador (situação em que apenas um dos eixos regista alteração de valores) são considerados pelo sistema como redundantes e não traduzíveis em efeito digital.

Como novas calibrações podem ser necessárias, o *threshold* superior de z (até agora não utilizado) é utilizado como input para ocorrer uma nova calibração.

Como referido anteriormente, no desenvolvimento do nosso projeto, tivemos em consideração as abordagens apresentadas nas aulas da UC de Fundamentos de Bioinstrumentação. Ao definir a nossa filtragem com um filtro passa-baixo, tivemos de implementar para funcionar em tempo real. Assim, este foi adaptado para um filtro denominado *Moving Average*. Caracteriza-se por ser um filtro temporal passa-baixo [8] e, como o próprio nome indica, é realizada a média de um certo conjunto de valores enquanto o sistema recebe o sinal, tratando-se assim de um filtro não causal - depende apenas dos inputs presente e passados. Esta explicação traduz-se na expressão 3 [1]:

$$y_i[n] = \frac{1}{N} \sum_{i=0}^{N-1} (x_i[n-i]) \quad (3)$$

Onde N é a dimensão da janela de amostras utilizado. Notemos que o peso associado (1/N) a cada input é igual para todos os elementos de uma determinada amostra. Adicionalmente, é percutível que quanto menor for a janela, mais suave se torna o resultado final ao desacentuar-se as grandes variações do sinal original. Por outro lado, quanto mais se aumentar a sua dimensão, obter-se-á a tendência do sinal, pois são incorporados mais pontos a cada média efetuada [1]. Esta equação irá ser aplicada aos sucessivos vetores descritos na equação 1.

Para converter o tamanho da amostra num valor de frequência de corte utilizamos a equação 4 [3], que, recorrendo ao código presente em [3], nos permitiu concluir que um tamanho de 3 aproxima o *Moving Average* de um sinal filtrado com um passa-baixo de 1Hz.

$$\sin^2\left(\frac{f_c N}{2}\right) - \frac{N^2}{2} \sin^2\left(\frac{f_c}{2}\right) = 0 \quad (4)$$

Assim, o nosso sinal fica pronto a utilizar, possuindo tipicamente uma *test run* como representada nos gráficos presentes em 7.

VI. MELHORIAS E DESAFIOS FUTUROS

No seguimento da adaptação enunciada no início do presente relatório, um melhoramento futuro consistiria em incorporar o protótipo na dita luva que, ao fixar-se na extremidade do membro residual, permitiria um manuseamento do *mouse* fidedigno do movimento efetuado pelo utilizador. A nível de software, apesar de não ser essencial para se alcançar o objetivo principal, poderia ser implementado código que permitisse mover o *mouse* e clicar simultaneamente, desbloqueando a função de arrastar. Outra funcionalidade que poderia ser introduzida consiste na ação de "scroll" que é possível de executar num rato usual de computador. Por fim, descrevemos aquele que seria um desafio relacionado com o

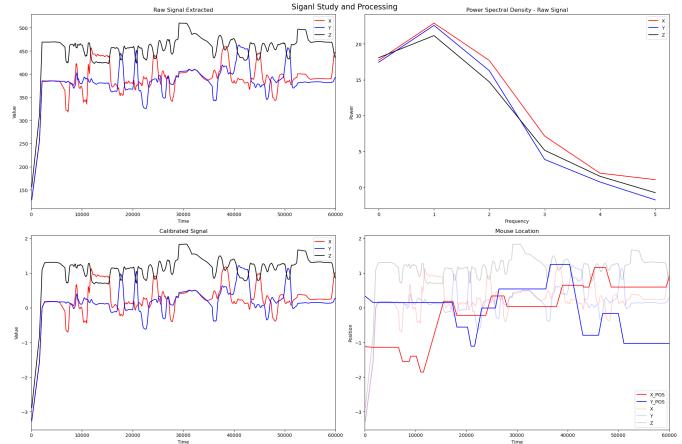


Fig. 7. *Test Run* de 1 minuto com os novos parâmetros

próprio princípio de aceleração. Caso o protótipo fosse testado em movimento (num comboio, por exemplo), as acelerações externas à utilização do acelerómetro iriam influenciar os valores registados.

VII. CONCLUSÃO

For people without disabilities, technology makes things easier. For people with disabilities, technology makes things possible [4].

Foi com esta citação que começámos este percurso e que também nos ajuda a concluir um protótipo que assegura o objetivo proposto: restituir o *workflow* de amputados de parte dos membros superiores, possibilitando a movimentação do rato e ação de clique numa interface digital.

REFERENCES

- [1] Yan Chen, Dan Li, Yanhai Li, Xiaoyuan Ma, and Jianming Wei. Use moving average filter to reduce noises in wearable ppg during continuous monitoring. In *Use Moving Average Filter to Reduce Noises in Wearable PPG During Continuous Monitoring*, volume 181, pages 193–203, 06 2016.
- [2] Daniele Esposito, Jessica Centracchio, Emilio Andreozzi, Gaetano D. Gargiulo, Ganesh R. Naik, and Paolo Bifulco. Biosignal-based human-machine interfaces for assistance and rehabilitation: A survey. *Sensors*, 21(20), 2021.
- [3] Geoffrey Hunter.
- [4] Xueliang Huo. Introduction and preliminary evaluation of the tongue drive system: Wireless tongue-operated assistive technology for people with little or no upper-limb function. *The Journal of Rehabilitation Research and Development*, 45(6):921–930, Dec 2008.
- [5] Md. Rokib Raihan, Abdullah Bin Shams, and Mohiuddin Ahmad. Wearable multifunctional computer mouse based on emg and gyro for amputees. In *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, pages 129–134, 2020.
- [6] Pablo Ramos, Mireya Zapata, Kevin Valencia, Vanessa Vargas, and Carlos Ramos-Galarza. Low-cost human-machine interface for computer control with facial landmark detection and voice commands. *Sensors*, 22(23), 2022.
- [7] Julie Rekant, Lee E. Fisher, Michael L. Boninger, Robert A. Gaunt, and Jennifer L. Collinger. Amputee, clinician, and regulator perspectives on current and prospective upper extremity prosthetic technologies. *Assistive Technology*, page 1–13, Feb 2022.
- [8] Andrew G Webb and Cambridge University Press. *Principles of Biomedical Instrumentation*. Cambridge University Press, Cambridge, UK ; New York ; Melbourne ; Delhi ; Singapore, 2019.
- [9] Jinlong Xing, Gongliu Yang, and Tijing Cai. Modeling and calibration for dithering of mdrlg and time-delay of accelerometer in sins. *Sensors*, 22(1), 2022.