

## APLICAÇÃO DE TÉCNICAS DE MACHINE LEARNING PARA CONTROLE DE UMA MÃO ROBÓTICA <sup>1</sup>

**Carlos Eduardo Palmieri TEIXEIRA**<sup>2</sup>  
Graduando em Engenharia Eletrônica  
IFSP/Câmpus São Paulo

**Giovanni Antunes BONIN**<sup>3</sup>  
Graduando em Engenharia Eletrônica  
IFSP/Câmpus São Paulo

**Alexandre Brincalepe CAMPO**<sup>4</sup>  
Doutor em Engenharia Elétrica/USP  
Docente do Departamento de Elétrica  
IFSP/Câmpus São Paulo

### RESUMO

A proposta deste trabalho é desenvolver um sistema que acione uma mão robótica de acordo com os sinais eletromiográficos processados utilizando técnicas de análise e classificação com ferramentas de aprendizado de máquina. A captura dos sinais eletromiográficos de superfície será realizada a partir de um dispositivo que efetua a leitura e a transmissão de sinais eletromiográficos (Myo® Armband). O armazenamento e processamento dos dados serão efetuados em um *hardware* microprocessado (Raspberry Pi 3B). Neste trabalho será utilizado um algoritmo do tipo MLP (*Multilayer Perceptron*) responsável por classificar o sinal amostrado e, com isso, realizar o acionamento de uma mão eletromecânica de acordo com os gestos identificados.

**Palavras-chave:** Myo®; *Machine Learning*; Mão Robótica.

### Introdução

Diversos trabalhos (VUJAKLIJA, 2016; WANG, 2015) descrevem diferentes abordagens para o desenvolvimento de próteses e órteses. As primeiras próteses foram

---

<sup>1</sup> Artigo resultante de Trabalho de Conclusão de Curso. Orientador Prof. Dr. Alexandre Brincalepe Campo.

<sup>2</sup> Endereço eletrônico: edu-palmieri@hotmail.com

<sup>3</sup> Endereço eletrônico: giovanni.bonin@gmail.com

<sup>4</sup> Endereço eletrônico: brinca@ifsp.edu.br

construídas apenas como uma extensão do membro afetado, conferindo uma reposição do mesmo, mas que não possuíam capacidade de movimentação de seu punho e de sua mão. A evolução dessas próteses trouxe um grande avanço, permitindo então que se realizassem movimentos dos dedos, em posições específicas, como: mão fechada, aberta, pinça tridigital etc. No entanto, ainda não conferiam um grau de movimento e ação antropomórfica ao usuário, não permitindo graus de liberdade elevados, bem como demais movimentos. Para que se possa controlar os movimentos em uma prótese, se faz necessária a captura dos sinais denominados Sinais Eletromiográficos de Superfícies (SEMGs) (REAZ, 2006, 11) (Figura 1).

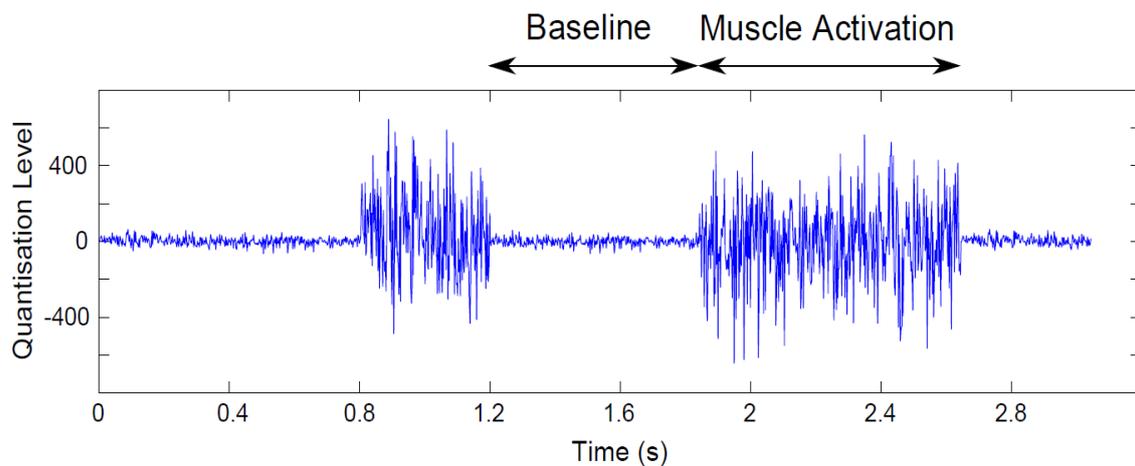


Figura 1: Amostra de sinal eletromiográfico de superfície durante o repouso e a ativação muscular  
Fonte: Ever, 2018

A fim de se obterem estes sinais, se faz necessária a utilização de um componente elétrico associado a um circuito de aquisição. O componente é composto por eletrodos (variando-se de um a n eletrodos) e mais um eletrodo de referência (terra).

## Objetivo

O objetivo deste trabalho é analisar e classificar sinais eletromiográficos através de ferramentas de aprendizado de máquina, permitindo o acionamento de uma mão robótica de acordo com as posturas identificadas.

## Referencial teórico

O sinal mioelétrico é proveniente do potencial de ação que percorre a fibra muscular, levando-a à contração. Este sinal deriva da unidade motora, que é a menor unidade muscular controlável, constituída por um neurônio motor, suas junções neuromusculares e as fibras musculares inervadas por este neurônio (ORTOLAN, 2002).

Dessa forma, o potencial de ação pode ser descrito como variações muito rápidas dos potenciais externo e interno da membrana da célula nervosa (ORTOLAN, 2002), presentes nas fibras musculares. Estes potenciais, então, se deslocam ao longo da fibra nervosa, partindo de um potencial inicial (denominado de potencial de repouso) para um potencial positivo (Figura 2), até que termina com um retorno ao potencial negativo. Ao enviar um potencial de ação, o neurônio motor o faz para todas as fibras musculares presentes na sua unidade motora, que são estimuladas. Sendo assim, a soma de todos os potenciais de ação de um determinado conjunto fibras é denominada de potencial de ação da unidade motora (ORTOLAN, 2002).

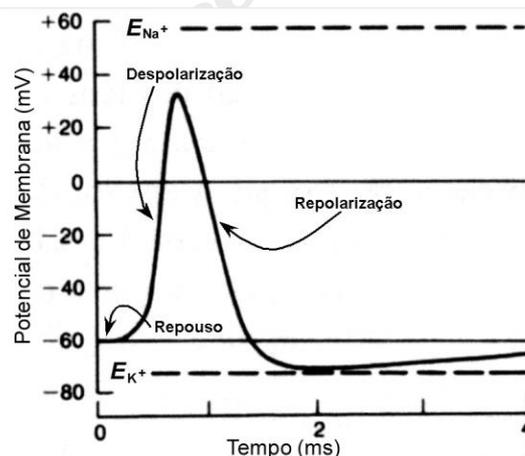


Figura 2: Forma de onda de um potencial de ação  
Fonte: Ortolan, 2002

Estes potenciais podem ser capturados através de uma técnica denominada eletromiografia (FAVIERO, 2009). A implementação dessa técnica consiste em dispor n eletrodos mais um eletrodo de referência (terra) e adquirir um sinal elétrico na superfície. Como os eletrodos são posicionados superficialmente na pele, o sinal obtido

diz respeito a uma "janela" muscular (uma área de tamanho definida que compreende uma quantidade de músculos). Sendo assim, o sinal possui várias componentes, o que faz necessário um tratamento após sua obtenção (Figura 3).

A fim de se utilizarem estes sinais, extraem-se então as denominadas *features* (características). Estas *features* são tratamentos dos dados coletados, visto que estes dados seriam apenas valores de intensidade.

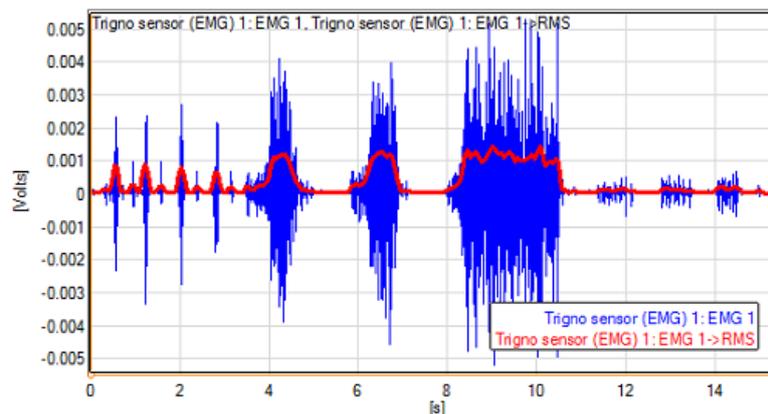


Figura 3: Exemplo de um tratamento de sinal utilizando a técnica RMS  
Fonte: Delsys, 2014

Tendo muitos dados, estas características são utilizadas para realizar uma redução dimensional, eliminando assim parte da redundância que pode ser encontrada nos dados, bem como realizar um ajuste de valores discrepantes. Como exemplo de *features* utilizadas, têm-se: a média, a variância, o valor quadrado médio (RMS), entre outras.

Inspirando-se em neurônios biológicos do sistema nervoso, é possível estabelecer um modelo computacional primitivo (HAYKIN, 1994) (Figura 4). A este tipo de modelo se dá o nome de Perceptron, que foi idealizado por Frank Rosenblatt, em 1958.

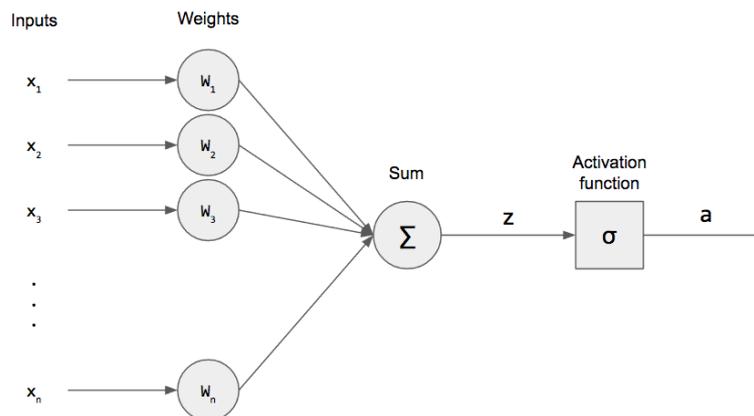


Figura 4: Diagrama de um simples neurônio Perceptron  
Fonte: Python Machine Learning, 2013

Neste modelo, os sinais de entradas são os dados/informações disponíveis, em que cada entrada possui um peso correspondente a fim de ponderá-lo. Em seguida, os dados pós-ponderados são somados, gerando uma saída.

Finalmente, o resultado é aplicado na denominada função de ativação, que é responsável por limitar e/ou demonstrar o comportamento da saída dentro de uma certa faixa e condição, sendo algumas dessas funções: degrau, sigmóide, hiperbólica etc.

No entanto, devido à complexidade dos problemas, um Perceptron simples não se faz suficiente. Para isso, se adicionam mais camadas entre a camada de entrada e a de saída, as denominadas *hidden layers* (camadas escondidas) (Figura 5) e com isso tem-se a MLP (*Multilayer Perceptron*). Cada uma dessas camadas possui, como a de entrada, os neurônios anteriores e seus respectivos pesos para ponderação dos valores. Com estas camadas escondidas, pode-se obter uma maior separação dos dados, pois tem-se não mais uma reta separando estes dados, mas n retas.

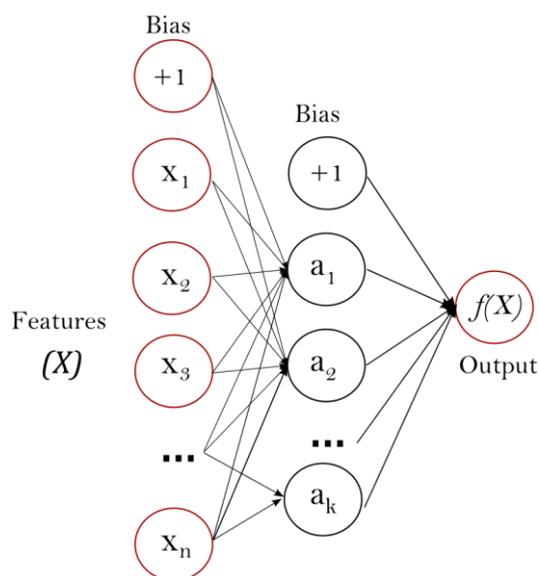


Figura 5: Perceptron Multicamadas formando uma rede  
Fonte: Pedregosa et al., 2011

Para se definirem os pesos de ponderação, se aplica também a técnica de *Backpropagation* (retropropagação) (Figura 6). Dado o valor de saída da rede, este é comparado com o valor esperado, gerando um valor de erro. Este valor de erro então é utilizado, propagando-se de volta na rede para reajuste dos pesos de todos os outros neurônios (HAYKIN, 1994).

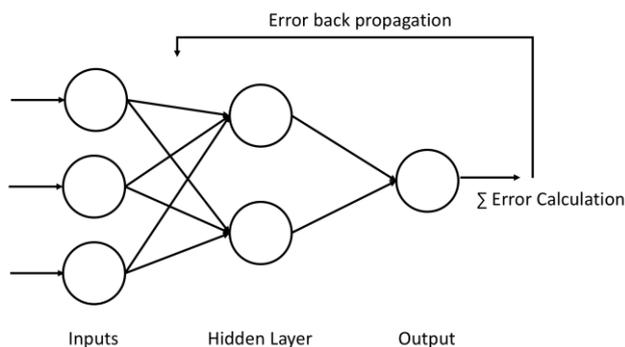


Figura 6: Diagrama de blocos explicando o funcionamento da retropropagação  
Fonte: Renals, 2017

## Metodologia

Para a coleta dos dados, foi escolhida uma ferramenta de mercado chamada Myo Armband. Com ela, além de ser possível transferir os dados via Bluetooth 4.0 (*Low Energy*), certos tipos de ruídos não precisam ser tratados, como o de rede (60Hz) ou o

advindo da movimentação dos cabos (para eletromiógrafos convencionais). Outro ponto a se destacar é que sua utilização não exige uma prévia tricotomia (raspagem dos pelos e aplicação de gel/pasta abrasiva) no local de interesse.

A Myo Armband conta com oito pares de eletrodos por divisão (Figura 7). Ela também conta com uma IMU (*Inertial Measurement Unit*), que possui recursos como giroscópio e acelerômetro. Os dados de SEMGs recebidos da Myo são dados que tem seu valor entre -127 e 127, representando então valores positivos e negativos (adimensionais).

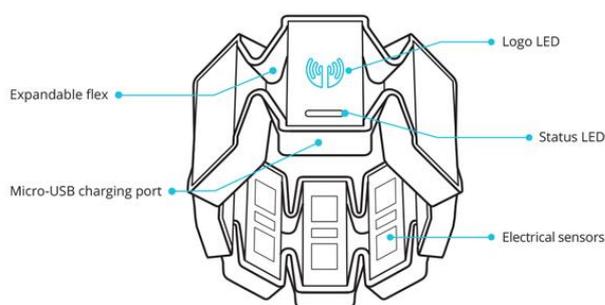


Figura 7: Diagrama de posicionamento dos elementos da MYO  
Fonte: Thalmic Labs, 2013

Para a realização das medições, foi escolhido apenas um voluntário, sendo este um dos autores deste trabalho. Posicionou-se a Myo® sempre com seu LED de *status* perpendicular ao dedo médio. A altura utilizada foi em torno de 5cm abaixo da divisão entre o braço e antebraço (Figura 8).

Os dados foram coletados realizando-se as posições (flexão) em seu percurso máximo, ou seja, para o caso de Mão Fechada, Polegar, Indicador, Médio, Anelar e Mínimo e em repouso para Mão Aberta. Foi utilizada para os movimentos uma força moderada, não causando assim grande fadiga.



Figura 8: Posicionamento da Myo® Armband em relação às distâncias e orientação  
Fonte: Autores

Utiliza-se então, na Raspberry, um programa em Python que realiza captura dos SEMGs. Esse *script* permite salvar os sinais classificados por posição, escolhendo a desejada (nome da posição), por quantas vezes irão se realizar as medições, bem como por quanto tempo cada medição será feita (em segundos). Alguns *scripts* utilizados foram obtidos de um repositório, Open-Myo® (VILLOSLADA, 2017), e adaptados conforme a necessidade. No projeto, o autor realiza as coletas temporizadas, e não por quantidade de coletas, o que foi adotado também neste projeto.

Após todas as coletas, é gerado um arquivo contendo todas as medições para cada posição. Este arquivo então é aberto em outro *script* que será responsável por carregar esses dados, separá-los e extrair oito *features*, como o valor eficaz das medidas.

O algoritmo então separa 20% das amostras para a realização de testes e o restante para o treino da rede neural do tipo MLP, implementada pelo pacote Scikit-Learn (PEDREGOSA, 2011).

Abaixo vemos os fluxogramas de captura dos dados, processamento, treinamento e teste da rede (Figura 9). Em seguida, o de teste em tempo real para acionamento da mão (Figura 10).

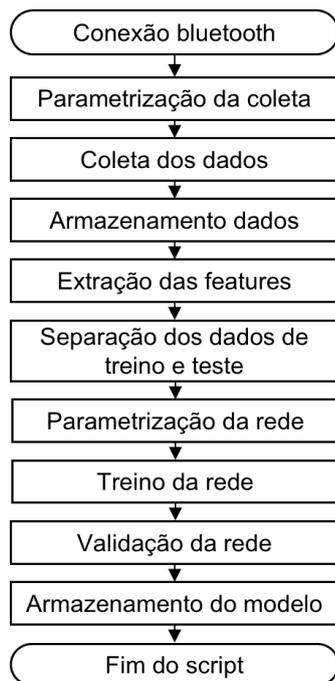


Figura 9: Fluxograma do processo de captura e treino de um modelo  
Fonte: Autores

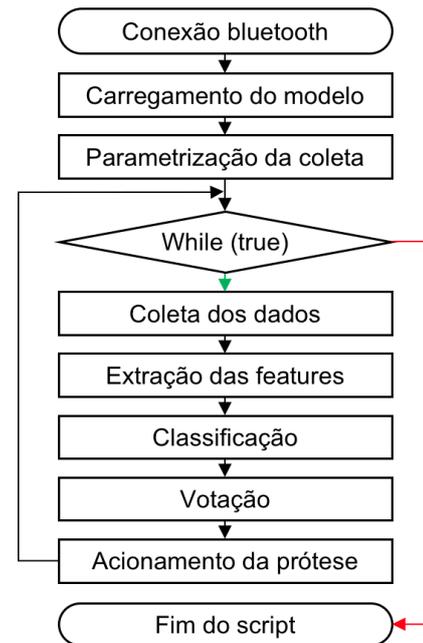


Figura 10: Fluxograma do teste em tempo real de um modelo treinado  
Fonte: Autores

Assim que o treino é finalizado, uma das funções fica responsável por utilizar a parte das amostras de teste sob o resultado da rede neural treinada e retornar a porcentagem de acerto para cada posição, bem como a porcentagem total de acerto, representados em uma matriz de confusão. Ao final, a rede treinada é salva para posterior utilização.

### Arquitetura de *hardware*

Com a necessidade de poder de processamento e de comunicação e portabilidade, foi decidido usar uma Raspberry Pi Zero W Bluetooth 4.0.

Este computador é capaz de executar uma coletânea de ferramentas de *software*, uma vez que possui uma versão embarcada do sistema operacional Linux compatível com a arquitetura ARM. Com a necessidade de expandir as saídas PWM, se fez necessária a expansão para cinco ou mais canais PWM, devido à plataforma robótica utilizar cinco servos-motores que fazem o controle de cada dedo individualmente. Dessa

forma, foi utilizado um *hardware* expensor com comunicação i2c dedicado a saídas PWM. O circuito integrado escolhido, PCA9685, contempla 16 saídas PWM, com resolução de 12bits.

Para o projeto eletrônico, *layout* e modelagem 3D, foi utilizado o *software* KiCad. Nele foram desenvolvidas todas as partes que compõem o HAT (*Hardware Attached on Top*). Os arquivos foram enviados para a produção e, logo em seguida, montados e acoplados à Raspberry (Figura 11) para início dos testes.

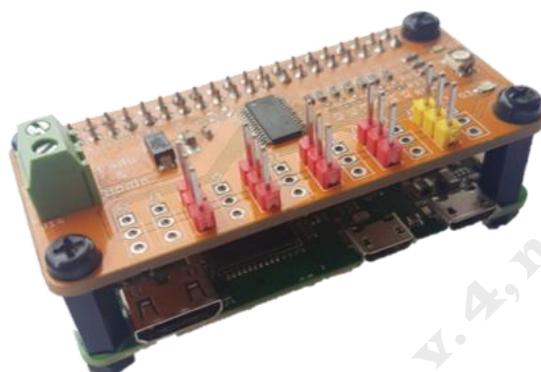


Figura 11: HAT concluída e montada sobre a Raspberry Pi Zero W  
Fonte: Autores

## Resultados

Após o treino dos classificadores, os testes foram realizados com um novo conjunto de dados retornando os apresentados a seguir. Os testes foram realizados com um total de 665 amostras para treino e 665 novas amostras para validação, de todas as posições estudadas (Mão Fechada, Mão Aberta, Flexão do Polegar, Flexão do Indicador, Flexão do Médio, Flexão do Anelar e Flexão do Mínimo), com a utilização de uma força moderada para cada posição, bem como capturada sempre a posição final do movimento.

LR	0.0001	0.001	0.001	0.0001
Neurônios por camada	32,64,128	32,64,128	12,32,64	12,32,64
Tempo de treino	57s	26s	21s	1m3s
Épocas	216	48	57	720
Média	58%	73%	73%	76%

Tabela 1: Comparativos com redes MLP  
Fonte: Autores

Feita a validação, realizada com os testes estáticos, os resultados foram demonstrados através de uma mão eletromecânica, que foi reproduzida e adaptada do projeto *open-source* InMoov (LANGEVIN, 2014). O braço robótico foi construído utilizando peças produzidas em impressão 3D e, após montada a estrutura, recebeu a eletrônica embarcada desenvolvida neste trabalho (Figura 12).



Figura 12: Braço eletromecânico sob domo de acrílico com eletrônica embarcada  
Fonte: Autores

## Discussões

Segundo os testes realizados, apesar de se terem utilizado menos neurônios por camada, pode-se obter um resultado melhor do que aquele em que são utilizados mais neurônios por camada, havendo assim muita separação dos dados, e possivelmente maiores chances de haver resultados com porcentagens menores.

O *learning rate* utilizado, apesar de bastante baixo, foi importante para o resultado final, apesar de que, quanto menor, faz com que a rede treine por mais épocas, tendo o seu resultado final mais demorado, e, dependendo do caso, podendo causar o chamado *overfitting*, que é quando a rede é treinada por grandes números de épocas e faz com que a mesma fique "especialista" em apenas alguns casos (como os apresentados para teste).

## Conclusões

Conclui-se que - partindo-se da dispersão dos dados coletados, e posteriormente verificando-se o resultado da rede classificadora -, apesar de se obter um resultado de 76%, o problema trazido pela proposta deste projeto não é trivial, ou seja, a separação dos dados (classificação) se mostra complicada, em especial para as posições Indicador, Médio e Mínimo.

Diante dos resultados obtidos, foi possível concluir também que, para este devido problema, considerando-se que foram tratados apenas os movimentos finais de cada posição, em que se tem apenas o valor destas em sua excursão final (e não em movimento) e com tendo como base uma quantidade razoável de amostras (665 no total), uma rede classificadora do tipo MLP pode obter resultados com acertos médios a altos (utilizando como parâmetros de 50% a 80%).

## Referências

DESHPANDE, Mohit. **The First Neural Networks** Disponível em <https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks>. Acesso em: 05 nov. 2018.

FAVIERO, Gabriela Winkler. Controle de uma prótese experimental do segmento mão-braço por sinais mioelétricos e redes neurais artificiais. 2009. Disponível em <https://lume.ufrgs.br/handle/10183/18554>.

HAYKIN, Simon. **Neural networks: a comprehensive foundation**. Prentice Hall PTR, 1994.

LANGEVIN, G. InMoov-Open Source 3D printed life-size robot. 2014 Disponível em <http://inmoov.fr>, License: [creativecommons.org/licenses/by-nc/3.0/legalcode](https://creativecommons.org/licenses/by-nc/3.0/legalcode).

ORTOLAN, Rodrigo Lício. Estudo e Avaliação de Técnicas de Processamento do Sinal Mioelétrico para o controle de Sistemas de Reabilitação. 2002. Dissertação (Mestrado em Engenharia Elétrica) - Escola de Engenharia de São Carlos. São Carlos, 2002.

PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. **Journal of machine learning research**, v. 12, n. Out, 2011. p. 2825-2830.

REAZ, Mamun Bin Ibne; HUSSAIN, M. S.; MOHD-YASIN, Faisal. Techniques of EMG signal analysis: detection, processing, classification and applications. **Biological procedures online**, v. 8, n. 1, 2006.

VILLOSLADA, Alvaro. Python module to get data from myo armband using a generic ble interface. 2017. Disponível em <https://github.com/Alvipe/Open-Myo>.

VUJAKLIJA, Ivan; FARINA, Dario; ASZMANN, Oskar C. New developments in prosthetic arm systems. **Orthopedic Research and Reviews**, v. 8, 2016. p. 31-39.

WANG, Jianhua et al. A portable artificial robotic hand controlled by EMG signal using ANN classifier. In: **Information and Automation, 2015 IEEE International Conference on**. IEEE, 2015. p. 2709-2714.

### **APPLICATION OF MACHINE LEARNING TECHNIQUES FOR CONTROL OF A ROBOTIC HAND**

#### **ABSTRACT**

*The purpose of this work is to activate a robotic hand using an electromyography data collection system for analysis and classification through machine learning tools. The capture of the electromyographic surface signals will be performed from Myo® Armband, storing and processing the data in a Raspberry Pi 3B. In this work, an algorithm of the type MLP (Multilayer Perceptron) will be used to classify the sampled signal and with this, to carry out the activation of an electromechanical hand according to the identified gestures.*

**Keywords:** Myo®; Machine Learning; Robotic Hand.

**Envio: dezembro/2018**  
**Aceito para publicação: abril/2019**