

Relatório de Atividades

Equipe UFPE

DETEB 2018

Professores:

Alana Elza Fontes Da Gama ^{1,3}

Guilherme Nunes Melo ^{2,4}

João Marcelo Xavier Natário Teixeira ^{1,2,4}

Marilú Gomes Netto Monte da Silva ³

Estudantes:

José Gomes da Silva Neto ¹

Marina Cardoso de Oliveira ³

Mirella Santos Pessoa de Melo ¹

Thiago Buarque de Gusmão Lafayette ²

Willams de Lima Costa ¹

¹ Voxar Labs, Centro de Informática, Universidade Federal de Pernambuco

² Maracatronics, Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco

³ Grupo de Pesquisa em Engenharia de Reabilitação, Departamento de Engenharia Biomédica, Universidade Federal de Pernambuco

⁴ Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco

Sumário

1. Introdução	3
2. Metodologia	6
3. Cronograma	7
4. Implementação	8
4.1 Controle por meio de software de simulação	8
4.1.1 Captura de sinais mioelétricos através do Myo Armband	8
4.1.2 V-REP	9
4.1.3 Integração V-REP com Myo Armband	10
4.2 Braço Robótico 6-DOF	11
4.2.1 Montagem	11
4.2.2 Esquema elétrico geral	11
4.2.3 Testes de Funcionamento do Braço Robótico	12
4.2.4 Confeção do circuito impresso	16
4.3 Captação de sinais mioelétricos	17
4.3.1 Circuito para Captação dos Sinais Mioelétricos	17
4.3.2 Teste de Funcionamento do Circuito	19
4.3.3 Protocolo de Captura de Sinal de Eletromiografia	19

4.3.4 Resultados do Sistema de Aquisição do Sinal Eletromiográfico	21
4.4 Cinemática de Robôs	22
4.4.1. Cinemática Direta	23
4.4.2.1. Equações de cinemática direta	25
4.4.3 Cinemática inversa	25
4.4.3.1. Método Geométrico para cinemática inversa	25
4.5 Aprendizado de Máquina	27
4.5.1 Abordagem por aprendizado raso	28
4.5.1.1 Extração de Características	29
4.5.1.2 <i>Multilayer Perceptron</i>	29
Fonte: Autores (2018).	30
4.5.1.3 <i>Random Forests</i>	30
4.5.1.4 <i>Gradient Boosting</i>	31
4.5.2 Abordagem por aprendizado profundo	31
4.5.2.1 <i>DenseNet</i>	32
4.5.2.2 <i>Inception-V3</i>	32
4.5.2.3 Discussão	33
4.6 Custos	34
4.6.1 Custos do sistema mioelétrico elaborado	34
4.4.2 Custos do controle utilizando Myo	35
5. Conclusão	36
6. Referências	37
7. Apêndice I – Código OpenCV	40
8. Apêndice II – Código do Arduino 1	43
8. Apêndice III – Código do Arduino 2	45

1. Introdução

Este relatório visa descrever as atividades exercidas pela equipe da UFPE até o presente momento para o Desafio Temático em Engenharia Biomédica (DETEB), incluindo passos executados, implementações, testes, resultados e limitações encontradas durante o processo. Este documento traz também o planejamento até o dia do desafio, previsto para 24 de Outubro de 2018 na cidade de Búzios, no Rio de Janeiro.

O DETEB terá sua primeira edição e ocorrerá no XXVI Congresso Brasileiro de Engenharia Biomédica (CBEB) com objetivo de estimular o desenvolvimento de soluções na área de braços robóticos controlados por sinais de eletromiografia (EMG) ou mecanomiografia (MMG) para problemas na área de Engenharia Biomédica. Dessa forma, busca incentivar envolvimento de alunos na área em questão, além de estimular o desenvolvimento técnico na área nos centros de Engenharia Biomédica do país, bem como aumentar a visibilidade do campo de Engenharia Biomédica para a Sociedade e potenciais empresários.

Com avanços tecnológicos, o uso de braços robóticos em saúde tem crescido significativamente a cada dia abrangendo soluções em diferentes aplicações nessa área, variando desde robôs para cirurgias, que requerem posicionamento minucioso de ferramentas (**Annamdas e Soh, 2018**), a braços robóticos para substituição de membros perdidos (próteses de membros superiores), que requerem formas alternativas de controle que sejam inerentes ao usuário (**May e Lockard, 2011; Carey, Stevens e Highsmith, 2017**). Dentre as aplicações tem-se também braços robóticos para auxílio à manipulação em pessoas com mobilidade reduzida, por exemplo, acopláveis a cadeiras de rodas para pegar objetos (**Cooper, et al., 2006**).


Relacionado a braços robóticos em saúde, as pessoas envolvidas neste desafio tem trabalhado com diferentes aplicações em engenharia de reabilitação que serão publicados no XXVI CBEB. Um exemplo é o uso de braços robóticos acoplados a uma plataforma móvel para auxiliar pessoas com dificuldade de locomoção quando não estão em cadeiras de rodas (**Melo et al., 2018**). Outra grande área de braços robóticos em saúde é na terapia robótica, onde eles são utilizados para auxiliar ou resistir a um exercício. Esse tipo de terapia tem tido resultados positivos principalmente na área de reabilitação de pacientes de Acidente Vascular Encefálico (AVE) (**Veerbeek et al., 2017; Maciejasz et al., 2014**). Na área de controle para esse tipo de aplicação estamos trabalhando com o uso de uma miniatura de braço robótico construída usando impressão 3D com seis eixos articulados com potenciômetros, que são responsáveis pela captura do movimento que o usuário realiza no controle (**Neto et al., 2018**).

Na área de braços robóticos em engenharia biomédica o uso de controle por sinal de eletromiografia ou mecanomiografia, caso específico desse desafio, é crescente principalmente na área de próteses mioelétricas. Os movimentos executados pela extremidade terminal das próteses mioelétricas são controlados pela atividade elétrica muscular do usuário de uma parte do corpo íntegra **(Carvalho, 2004; Barros, 2005)**. Esse tipo de controle requer menor gasto energético que as mecânicas e permite explorar maior número de movimentos e motricidade mais fina. Porém, essas próteses costumam ter alto custo e peso elevado para suportar os componentes e circuitos **(Carey, Stevens e Highsmith, 2017; May e Lockard, 2011)**.

A atividade muscular pode ser capturada por eletrodos internos ou de superfície que são posicionados no membro residual ou região mais proximal do membro, detecta a atividade dos músculos e por meio de processamento decide e envia comando para a resposta desejada **(Carvalho, 2004; Barros, 2005; May e Lockard, 2011)**. Com uso de eletrodos internos as próteses oferecem uma maior variedade de movimentos, contudo, o usuário precisa se submeter a uma cirurgia para implantar os eletrodos. Além disso, os eletrodos implantados podem causar dor e desconforto. No caso da eletromiografia de superfície, os eletrodos são fixados à pele, mas, devido à interferência do sinal de um músculo sobre o sinal de outro (*cross-talk*) é mais difícil a diferenciação de qual músculo está sendo ativado e assim tem-se uma menor possibilidade para controle de movimentos **(May e Lockard, 2011; Carvalho, 2004; Duque, de Godoi e Díaz, 2014)**.

Pesquisas vêm sendo feitas tentando aprimorar o uso de eletrodos de superfície para controle dessas próteses, a fim de atingir o número de diferenciação de movimentos e assim ampliar as possibilidades de uso. Para tal, duas estratégias têm sido propostas o uso de um maior número de eletrodos e/ou um reconhecimento dos padrões dos sinais mioelétricos mais eficiente. Embora certos movimentos possam ser reconhecidos usando apenas dois eletrodos, como por exemplo, um eletrodo no grupo extensor e outro no grupo flexor do antebraço, com o uso de mais eletrodos o sistema deve ampliar o número de possibilidades **(Ortolan, 2002)**. Devido a isso nossa equipe dividiu seus esforços em duas estratégias diferentes, uma com o uso do Myo Armband, um bracelete que possui oito eletrodos de eletromiografia e outra com desenvolvimento de circuito de eletromiografia próprio, que fosse simplificado para possível uso em próteses mioelétricas.

Nesse âmbito tomamos como ponto inicial o trabalho desenvolvido pelos colaboradores do grupo desse desafio que será publicado no XXVI CBEB, no qual estão desenvolvendo prótese mioelétrica em 3D **(De Oliveira et al., 2018)**. Nessa pesquisa investigamos a atividade elétrica muscular do membro superior em diferentes movimentos, para assim investigar o padrão de recrutamento dos mesmos e identificar quais músculos seriam os mais adequados para controlar prótese mioelétrica **(De Sousa et al., 2018)**. Com base nesses sinais trabalhamos



no desenvolvimento de processamento de sinais com base em aprendizagem de máquina para reconhecimento do padrão de recrutamento muscular e então gerar maiores possibilidades de controle.

A equipe formada para esse desafio é interdisciplinar com integrantes da Engenharia Biomédica, Engenharia de Automação e Controle, Mecânica, Eletrônica, Computação e Fisioterapia. A interdisciplinaridade é uma característica importante na engenharia biomédica, sendo fundamental no desenvolvimento de tecnologias inovadoras **(Royal Society, 2010)**.

Nas próximas seções serão apresentadas as metodologias, técnicas e processos utilizados e as etapas desenvolvidas e seus custos até o momento pela equipe para o DETEB. O relatório irá descrever ainda o planejamento para as próximas etapas até o dia do desafio.

2. Metodologia

Para iniciar as atividades referentes ao DETEB, primeiramente fez-se um estudo que diz respeito aos sinais mioelétricos, bem como o levantamento de possíveis soluções de forma a entender como funcionam as tecnologias do estado da arte. A partir de um primeiro entendimento, optou-se por dar partida às práticas considerando o material que a equipe já havia posse, no caso, a plataforma Myo Armband.

A etapa com o Myo Armband, descrita na Seção 4.1, envolveu testes referentes à leitura do gesto do usuário de forma a interpretá-lo, por exemplo, quando este abria a mão, fechava ou outros movimentos secundários. Também foi utilizada uma simulação, via software V-REP, de um braço robótico com seis graus de liberdade controlado por meio dos sinais mioelétricos captados e interpretados pelo Myo. Com a finalidade de aumentar a assertividade na detecção do movimento também foram aplicadas técnicas de aprendizagem de máquina que envolveram treinamento em rede neural, descritas na Seção 4.4.

Em paralelo aos estudos com o dispositivo citado acima, foi realizada a montagem do braço robótico e diversos testes usando a plataforma Arduino, de forma a verificar conexões e determinar limitações dos motores (Seção 4.2). Também foram realizados testes funcionais onde foram reproduzidos movimentos funcionais que são comumente realizados para simular atividades da vida diária dos usuários de próteses. Esses testes são descritos na Seção 4.6.

Trabalhou-se também em uma outra possível solução, sendo esta com circuito integrado (CI) INA128P, que com as devidas conexões é capaz de sentir e amplificar a tensão exercida pelo músculo no momento de contração. Essa etapa é descrita na Seção 4.5. Até a data deste relatório os resultados se apresentam promissores, sendo capaz de detectar impulsos a partir dos eletrodos de superfícies aplicados à pele com um ganho de até mil vezes maior. Espera-se avaliar melhor os sinais para classificá-los quanto ao movimento realizado.

Preocupou-se em documentar as etapas citadas anteriormente em formato de vídeo, onde os respectivos *links* podem ser encontrados durante a leitura deste relatório.

3. Cronograma

Para desenvolvimento das atividades necessárias para o desafio o seguinte cronograma, apresentado no Quadro 1, foi e está sendo seguido:

Quadro 1 - Cronograma de atividades.

Atividade	Jun	Jul	Ago	Set	Out
Estudo e levantamento de pesquisas sobre sinais mioelétricos					
Propostas de soluções e determinação do grau de viabilidade das mesmas					
Montagem do braço robótico					
Testes com o braço e levantamento de limitações e dificuldades					
Myo Armband: calibração e testes					
Myo Armband: validação por meio de ambiente virtual					
Myo Armband: validação por meio físico (braço robótico)					
CI: montagem do circuito e testes					
Integração dos sistemas					
Testes finais de controle após integração					

Fonte: Autores (2018)

4. Implementação

Esta seção aborda todas as etapas desenvolvidas referente à implementação do braço robótico até o seu devido funcionamento.

4.1 Controle por meio de software de simulação

Um meio de validar a correta movimentação do braço robótico através de sinais mioelétricos é por meio de software de simulação. Essa alternativa é pouco custosa financeiramente e é capaz de nos dar informação do correto desenvolvimento em software, uma vez que elimina as problemáticas que envolvem o hardware.

4.1.1 Captura de sinais mioelétricos através do Myo Armband

O sinal eletromiográfico é uma representação do potencial elétrico gerado pela despolarização da membrana externa da fibra muscular. Esses sinais são formados através das variações fisiológicas no estado das fibras musculares atuando em conjunto **(Merletti e Farina 2016)**. A eletromiografia é uma técnica experimental voltada para o desenvolvimento e a análise de sinais mioelétricos. Além de proporcionar as condições para estudos fisiológicos, a eletromiografia também é estabelecida como uma ferramenta para avaliação na reabilitação, treinamento de esportes e interações do corpo humano a métodos e condições de trabalho **(Konrad, 2005)**.

Para a prototipação da ferramenta final, utilizamos os sensores de eletromiografia do Myo Armband¹ (Figura 1), um aparelho para o reconhecimento de gestos e poses de mão. O aparelho é apresentado como um bracelete que possui sensores inerciais, permitindo o reconhecimento do movimento, orientação e rotação do antebraço, e oito eletrodos por onde a atividade muscular é detectada. Esses dados são transmitidos através do protocolo de transmissão Bluetooth, juntamente com o sinal de eletromiografia puro, e são usados para controlar computadores, drones e até mesmo próteses mioelétricas **(Abraham, 2015; Thalmic Labs, 2017; Bernardt, 2015)**. Os dados de eletromiografia foram usados para realizar o treinamento de modelos de aprendizado de máquina, como descrito na seção [Aprendizado de Máquina](#).

¹ "Myo Gesture Control Armband | Wearable Technology by Thalmic Labs." <https://www.myo.com/>. Acessado em 14 Set. 2018.

Figura 1 - Myo Armband

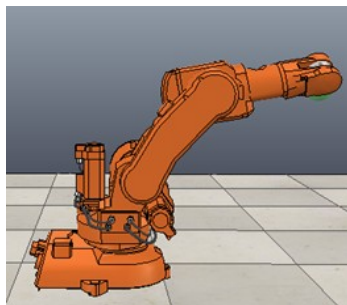


Fonte: Thalmic Labs (2018)

4.1.2 V-REP

Simuladores virtuais representam uma alternativa simples e barata para validar sistemas complexos, plataformas ou protótipos. O V-REP (**Freese, 2018**) foi desenvolvido para resolver este problema. Vários projetos podem ser simulados antes de serem implementados e, portanto, escolheu-se um modelo virtual que representa aproximadamente o braço robótico físico enviado pelo DETEB/CBEB, de seis graus de liberdade, como ilustrado na Figura 2.

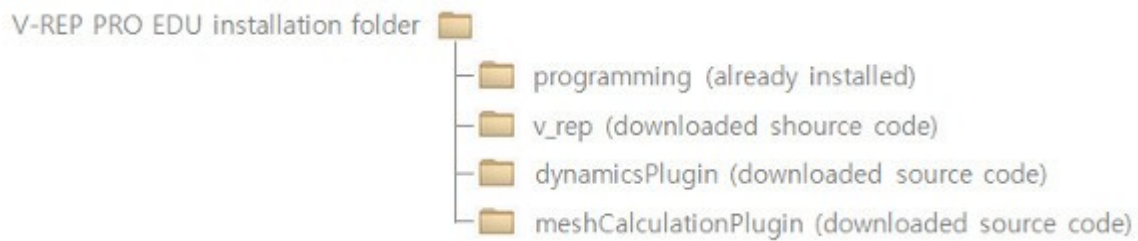
Figura 2 - Braço virtual robótico usado na simulação



Fonte: Autores (2018)

A comunicação do V-REP com uma aplicação externa foi por meio de *API* de comunicação do próprio V-REP. Fez-se uso da biblioteca em C/C++ disponibilizada pelo software, que também oferece funções em diversas outras linguagens, tais como Java, Python e Lua. Depois de criar o arquivo de extensão *.cpp*, modificou-se as propriedades de configuração nos diretórios adicionais de forma a associá-lo com a pasta “programming”, ilustrada na Figura 3, que é gerada juntamente com a instalação do simulador, e está localizada dentro da pasta V-REP. Após isso, o aplicativo externo é capaz de acessar as pastas de cabeçalhos e, conseqüentemente, fazer uso das funções do V-REP.

Figura 3 -Organização das pastas



Fonte: Autores (2018)

4.1.3 Integração V-REP com Myo Armband

Através do kit de desenvolvimento² disponibilizado pela Thalmic Labs, as poses de mão que são detectadas pelo Myo Armband podem ser usadas para desenvolver soluções baseadas em controle remoto. Para o controle através do simulador V-REP, um módulo de integração foi desenvolvido em C++ usando o kit citado anteriormente.

O kit de desenvolvimento apresenta métodos que permitem a captura do sinal de eletromiografia puro e também a captura da pose de mão já processada. Para a integração com o V-REP, apenas as informações de pose de mão foram usadas no controle. De acordo com a pose capturada, uma ação era enviada ao V-REP através da API de controle remoto³. O Quadro 2 descreve quais ações eram tomadas de acordo com cada gesto detectado.

Quadro 2 - Ações implementadas no simulador de acordo com o gesto detectado no Myo Armband.

Gesto	Ação
fist	Não implementado
waveOut	Incremento na rotação da junta
waveIn	Decremento na rotação da junta
fingerSpread	Não implementado
doubleTap	Mudança de junta
rest	Não implementado

Fonte: Autores (2018)

² "Myo SDK 0.9.0: The Myo SDK - Myo Developer - Thalmic Labs."

https://developer.thalmic.com/docs/api_reference/platform/the-sdk.html. Acessado em 17 Set. 2018.

³ "Remote API - Coppelia Robotics."

<http://www.coppeliarobotics.com/helpFiles/en/remoteApiOverview.htm>. Acessado em 17 Set. 2018.

O resultado prático dessa etapa de testes pode ser visualizado pelo link⁴ disponível.

4.2 Braço Robótico 6-DOF

Os subtópicos desta seção 4.2 são referentes ao braço robótico de seis graus de liberdade cedido pela organização do CBEB/DETEB.

4.2.1 Montagem

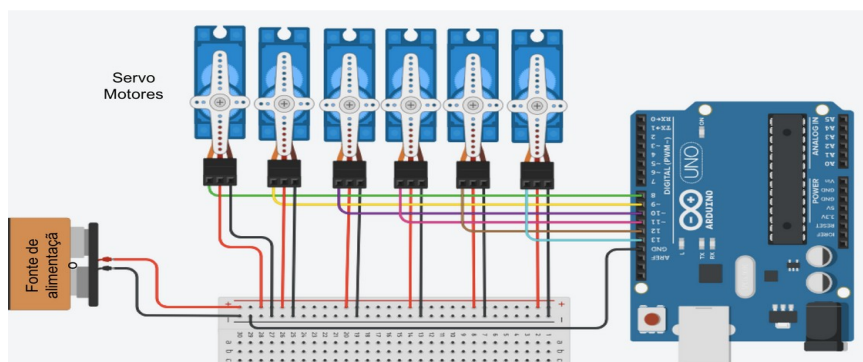
Após a obtenção do braço robótico que foi disponibilizado pela organização do DETEB iniciou-se sua montagem. O kit recebido contém seis motores MGXXX, a estrutura física do braço e os componentes extras necessários para a montagem, como parafusos e porcas. As ferramentas utilizadas para a montagem incluem um alicate de bico e uma chave de fenda.

A execução da montagem foi toda documentada em formato de vídeo e pode ser acessada pelo link⁵ disponível. Um manual de instruções⁶ também foi desenvolvido pela equipe.

4.2.2 Esquema elétrico geral

O esquemático exposto na Figura 4 ilustra como foram realizadas as conexões que envolvem o controle dos motores. É possível perceber pela Figura 5 que o Arduino e os motores são alimentados por uma fonte de alimentação de corrente contínua de 5V, a qual foi disponibilizada pela organização do CBEB/DETEB. As portas digitais do Arduino conectadas como descritas na Figura 4, enviam comandos de incremento ou decremento do ângulo do servo motor.

Figura 4 - Esquemático do circuito de controle dos motores



Fonte: Autores (2018)

4 "Vídeo_1: integração do Myo com o simulador V-REP. - YouTube." https://youtu.be/gTtdZVL_bj0 Acessado em 30 de Set. de 2018.

5 "Vídeo_2: montagem do braço robótico 6GdL. - YouTube." <https://www.youtube.com/watch?v=P6sjPyC-ZVg&t=111s> Acessado em 20 de Set. de 2018.

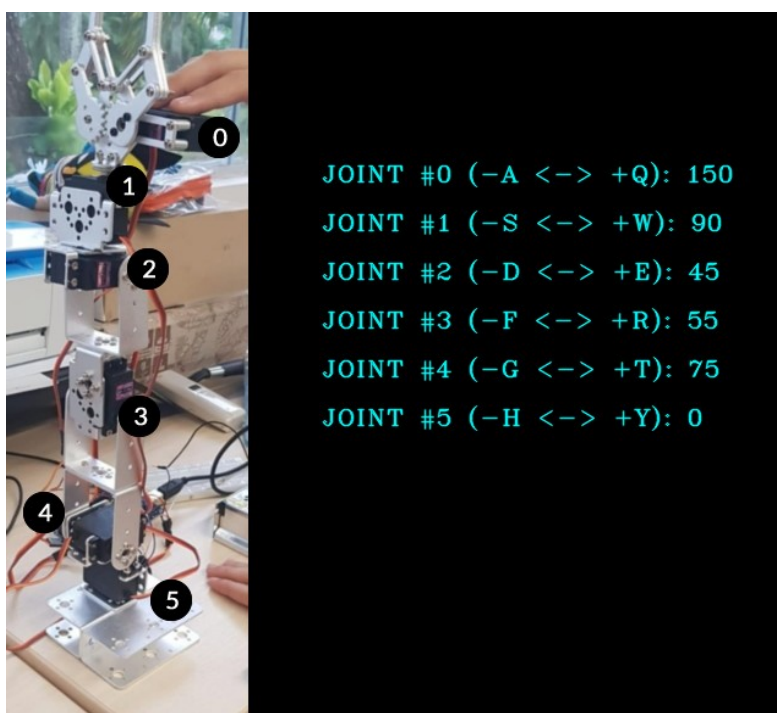
6 "Manual de montagem do braço 6-GdL" http://cin.ufpe.br/~wlc2/deteb/montagem_braco_6gdl.pdf. Acessado em 20 de Set. de 2018.

4.2.3 Testes de Funcionamento do Braço Robótico

Após feita a montagem do braço e realizadas as conexões citadas na subseção 4.2.2, fez-se necessária validação do pleno funcionamento do braço robótico, portanto, desenvolveu-se, em linguagem C++ e com auxílio da biblioteca *OpenCV* (Apêndice I) uma interface para manipulação dos 6 servo-motores. O resultado final desse *layout* pode ser visualizado na Figura 5. É importante citar que essa interface, desenvolvida através do software de desenvolvimento Visual Studio, integra também o código do Arduino (Apêndice II) e, assim, nos permite movimentar cada servo-motor individualmente.

Pela Figura 5, é possível perceber que a junta 0, por exemplo, incrementa o ângulo atual do servo motor quando a tecla "Q" é apertada, e decrementa este ângulo quando a tecla "A" é clicada. Essa lógica segue para as outras juntas, onde os valores da direita representam as limitações angulares do respectivo motor.

Figura 5 - *Layout* da solução que utiliza openCV



Fonte: Autores (2018)

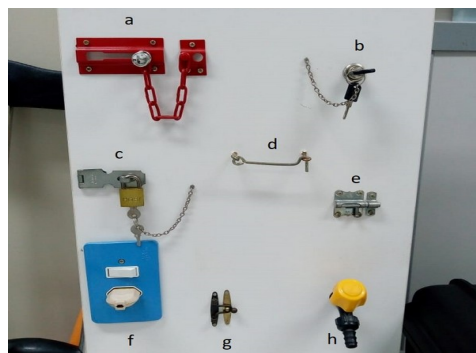
4.2.3.2 Testes funcionais

O braço robótico foi testado em termos de funcionalidade. Para avaliação da funcionalidade foram utilizadas a tábua de Atividades da Vida Diária (AVDs) e testes recomendados pela Academy's Ninth State of the Science Conference para avaliação de resultados de membro superior (Miller e Swanson, 2009), incluindo o teste de Jebsen-Taylor (J-T) (Jebsen e Taylor et al., 1969), o teste de caixa e

blocos (Kontson e Marcus *et al.*, 2017). Esses testes são recomendados como os mais promissores para avaliação funcional de próteses (Miller e Swanson, 2009).

A tábua de AVDs é composta por uma madeira com uma base de apoio onde são afixados objetos de atividade da vida diária como, por exemplo, ferrolhos e interruptores (Figura 6). Essa tábua é comumente utilizada durante a reabilitação como forma de treinar as AVDs. Visto que boa parte das funções realizadas por braços robóticos na área de engenharia biomédica deve estar relacionada a realização de funções como essas, seja em próteses de membros superiores ou em braços robóticos acoplados a cadeiras de rodas, foram realizados testes de como o braço robótico do desafio trabalharia nessas circunstâncias.

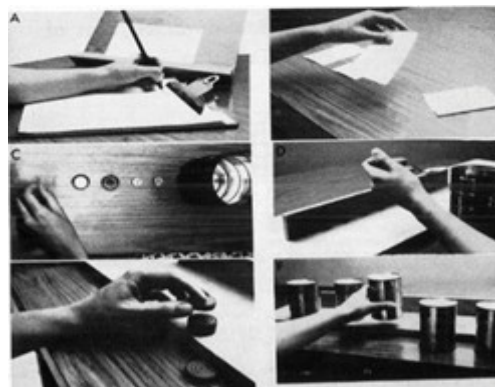
Figura 6 - Tábua de AVDs.



Fonte: Autores (2018).

O teste de J-T (Jebsen-Taylor Hand Function Test®) é padronizado e compõe-se de sete subtestes que avaliam diversas habilidades do paciente: escrever, virar cartas, pegar objetos pequenos, alimentar-se, empilhar blocos, pegar objetos grandes e leves e pegar objetos grandes e pesados (Figura 7). Esses subtestes são utilizados para estimular atividades funcionais, manuais e unilaterais (Jebsen e Taylor *et al.*, 1969).

Figura 7 - Testes funcionais do Jebsen-Taylor Hand Function Test



Fonte: Jebsen *et al.* (1969)

O Teste de Caixa e Blocos (Box and Blocks Test – BBT) (Kontson, Marcus *et al.* 2017) é composto por uma caixa de madeira (53,7 cm de comprimento) com uma divisória mais alta que as suas bordas, separando-a em dois compartimentos de iguais dimensões e 150 cubos de 2,5 cm (7,8) (Figura 8). O sujeito desloca os blocos de um lado a outro da caixa, sendo orientado a fazer inicialmente por 15 segundos para a familiarização com o teste. Na sequência, um minuto é cronometrado e conta-se o número de blocos transportados.

Figura 8 - Testes funcionais do Jebsen-Taylor Hand Function Test



Fonte: Autores (2018).

4.2.3.3 Resultado dos testes funcionais

Dentro do teste funcional com uso da tábua de AVDs o braço robótico fornecido pelo DETEB conseguiu realizar as atividades a seguir, cujo vídeo pode ser visualizado no link⁷.

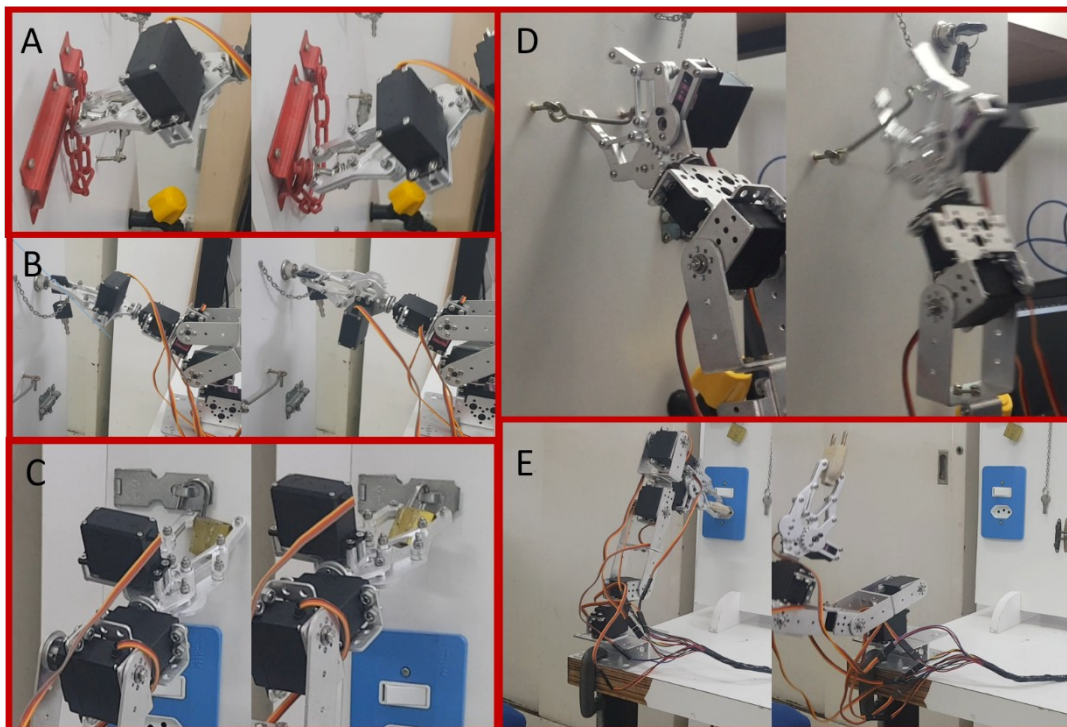
- Figura 6.a. Abrir o ferrolho deslizando e puxando o trinco (Figura 9.A);
- Figura 6.b. Girar a chave, sem remover a mesma, pois devido ao baixo atrito o braço robótico escapava (Figura 9.B);
- Figura 6.c. Remover o cadeado; resultado dependente da posição inicial do mesmo (Figura 9.C);
- Figura 6.d. Abrir o ferrolho levantando o trinco (Figura 9.D);
- Figura 6.f : Puxar tomada (Figura 9.E).

As tarefas da figura 6.e, 6.g, 6.h não conseguiram ser realizadas devido a minuciosidade requerida no controle. Pelo mesmo motivo grande parte dos testes de J-T não puderam ser realizados, como, por exemplo, pegar a moeda.

No teste J-T as atividades com latas, que pareciam ser adequadas para serem realizadas com o braço robótico do desafio não obtiveram sucesso devido a largura das mesmas. A abertura da pinça do braço não atingia distância suficiente para agarrar as latas padronizadas.

⁷ "Vídeo_3: testes funcionais com o braço robótico. - YouTube." <https://youtu.be/HpZy9KaicjQ>
Acessado em 30 de Set. de 2018.

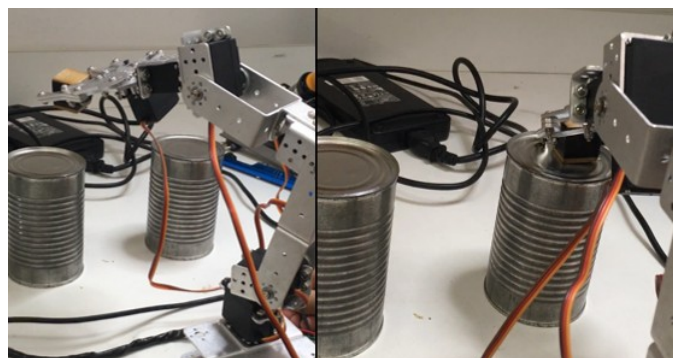
Figura 9 - Execução do braço robótico nos testes funcionais com a tábua de AVDs.



Fonte: Autores (2018).

No teste de Caixa e Blocos as atividades foram difíceis para o braço robótico devido ao baixo atrito e a necessidade de motricidade fina. O teste então foi realizado parcialmente, com fitas aderidas ao cubo para melhorar o atrito e apenas um bloco foi movido de um local para outro (Figura 10). A agilidade com o controle do braço com 6 graus de liberdade é difícil inviabilizando o movimento de vários blocos em um minuto.

Figura 10 - Execução do braço robótico no teste funcional de Caixa e Blocos.



Fonte: Autores (2018).

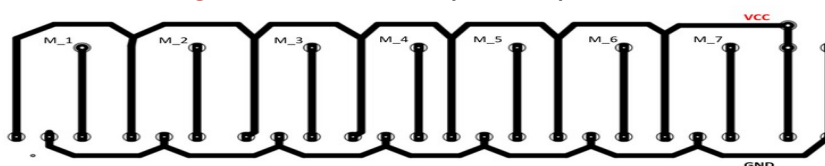
As limitações ocorridas devido ao baixo atrito poderiam ser reduzidas com uso de silicone na extremidade terminal do braço robótico. Porém, seguindo o regulamento do desafio, não realizamos nenhuma alteração no braço.

4.2.4 Confeção do circuito impresso

Com relação à conexão dos motores, optou-se por desenvolver uma placa de circuito impresso a fim de solucionar problemas de mau contato e sintetizar a conectividade dos fios. A confecção da placa seguiu o seguinte passo-a-passo:

1. Criação do *layout* da placa: desenvolvida através do software Proteus e tem seu resultado ilustrado na Figura 11. Seja “n” um número inteiro de 1 à 7, tem-se na Figura 11 que “M_n” indica a conexão referente ao pino de entrada de digital do servo motor;

Figura 11 - Circuito para impressão.



Fonte: Autores (2018).

2. Impressão do desenho em papel fotográfico com tinta a laser;
3. Limpeza da placa de fenolite já cortada em seu devido tamanho com um solvente de material volátil e lã de aço. Em seguida, finalizar a limpeza com algodão e álcool isopropílico;
4. Fixação do papel fotográfico com o lado da impressão contra a placa e realização da transferência por meio de um ferro de passar devidamente aquecido sobre o papel de maneira a realizar movimentos firmes e constantes. Estima-se que 3 minutos sejam suficientes para realizar toda a transferência da tinta do papel para a placa de fenolite;
5. Correção de possíveis trechos transferidos com falhas por meio de caneta de tinta permanente;
6. Furação com broca 0,5mm;
7. Corrosão do cobre por imersão em perclorato de ferro, realizando verificação a cada 5 min;
8. Soldagem dos fios de passe e componentes;
9. Testes de conectividade.

O resultado dessa confecção é ilustrado na Figura 12.

Figura 12 – Placa de circuito impresso.



Fonte: Autores (2018).

4.3 Captação de sinais mioelétricos

4.3.1 Circuito para Captação dos Sinais Mioelétricos

Para a aquisição do sinal mioelétrico foi elaborado um circuito de eletromiografia simples contendo dois canais. Cada canal é composto por um amplificador de instrumentação INA 128, um capacitor eletrolítico de 47 uF, um diodo de retificação e filtros para eliminar eventuais ruídos.

Amplificadores de instrumentação são circuitos altamente utilizados na instrumentação para amplificação de pequenos sinais. Eles são caracterizados por apresentarem entradas diferenciais e uma alta impedância de entrada, por isso eliminam a necessidade do casamento de impedância com a resistência de entrada. As entradas diferenciais são responsáveis por eliminar o sinal de modo comum. Além disso, os amplificadores de instrumentação possuem ganho programado por meio de um único resistor e alta rejeição a ruído. Pode-se observar o esquema do amplificador de instrumentação na Figura 13.

O circuito elaborado neste projeto utiliza uma impedância reativa de capacitores conectada aos pinos de ganho (1 e 8), o qual define um ganho dependente de frequência variável, dessa forma amplifica somente frequências mais altas, resultando em um efeito similar a um filtro passa alta. A impedância do capacitor é calculada conforme a Equação 1.

$$Z = \frac{-j}{2 \times \pi \times f \times 47 \mu F}$$

Equação 1

O cálculo do ganho é realizado conforme descrito pelas Equações 2, 3 e 4.

$$G = 1 + \frac{50 k\Omega}{Z}$$

Equação 2

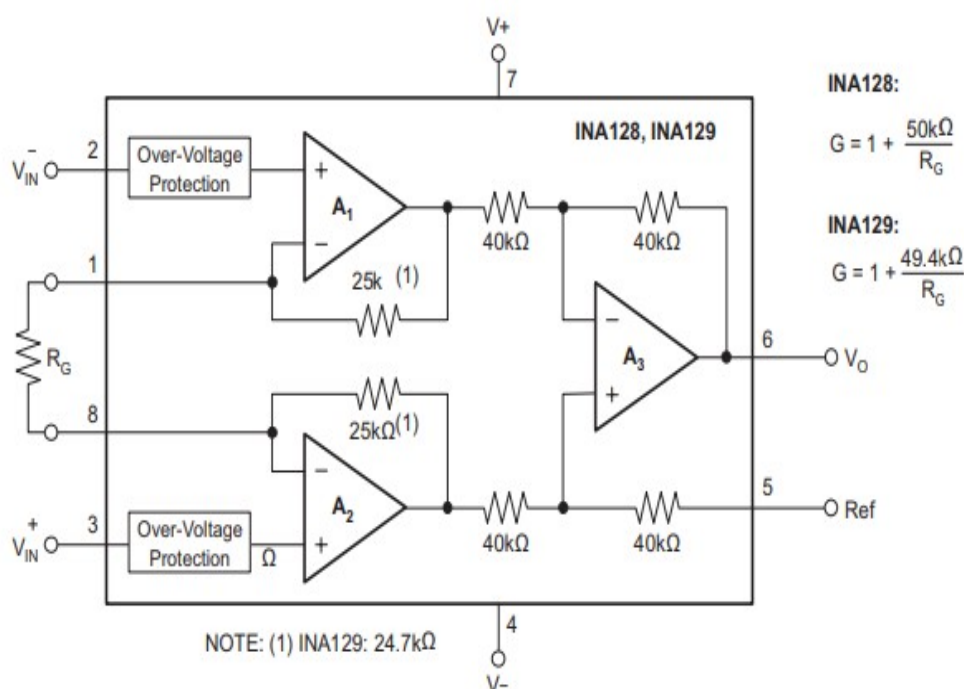
$$G = 1 + j 14,771$$

Equação 3

$$|G| = \sqrt{1 + 14,771^2} \quad \theta = \arctg(14,771)$$

Equação 4

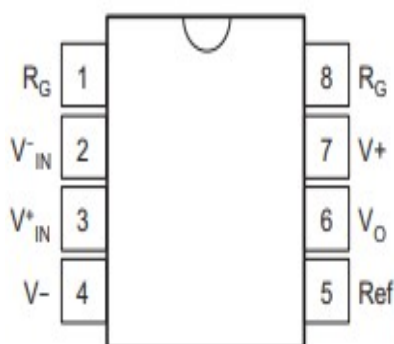
Figura 13 - Esquema do amplificador de instrumentação.



Fonte: INSTRUMENTS, Texas. Texas instruments incorporated ina128 datasheet. 2009.

As portas 2 e 3 correspondem às entradas do sinal mioelétrico, em que são conectados os cabos referentes aos eletrodos. A alimentação utilizada é simétrica e é conectada nas portas 4 e 7, - Vcc e + Vcc respectivamente. A saída do sinal corresponde à porta 6, em que é conectado o diodo de retificação. A referência do C.I. situa-se na porta 7. A correspondência dos pinos está descrita na Figura 14.

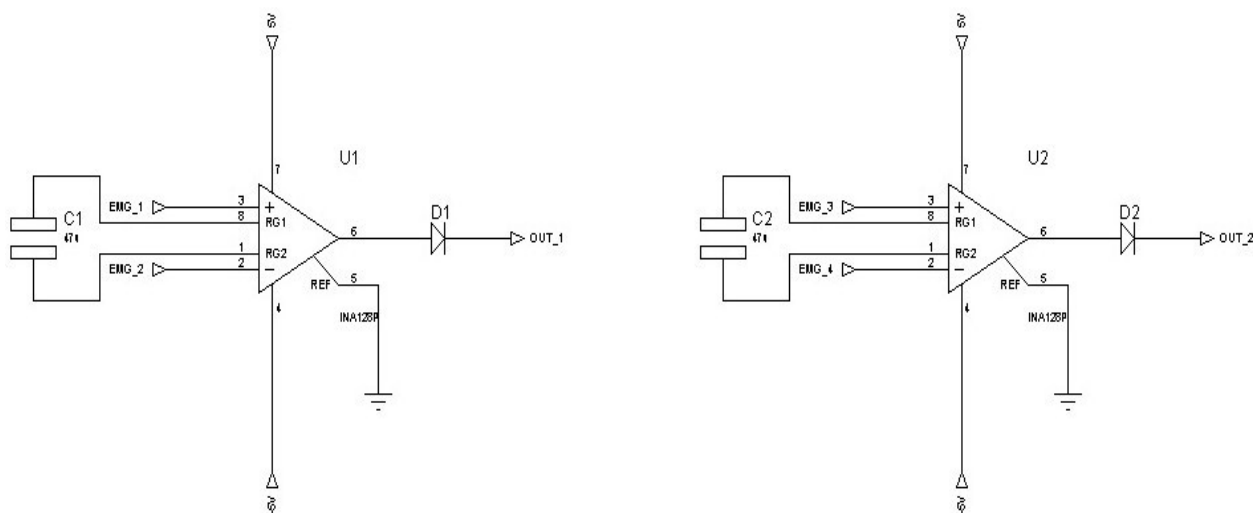
Figura 14 - Pinagem do CI INA 128



Fonte: INSTRUMENTS, Texas. Texas instruments incorporated ina128 datasheet. 2009.

O esquemático do sistema de aquisição de EMG completo pode ser observado na Figura 15.

Figura 15 - Esquemático do circuito de EMG



Fonte: Autores (2018).

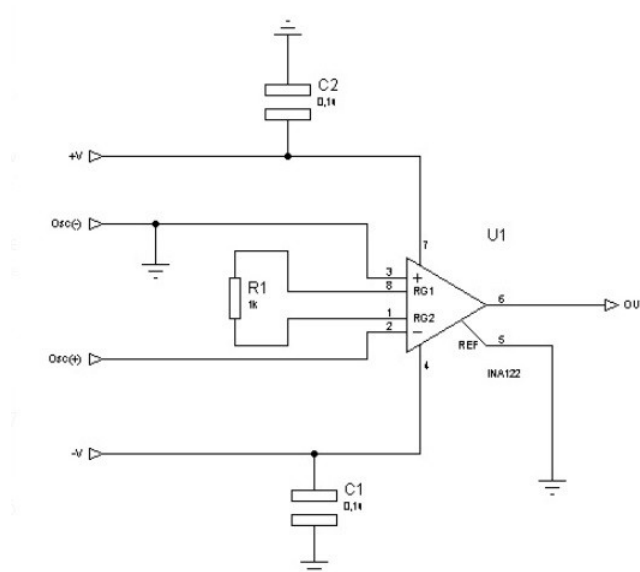
4.3.2 Teste de Funcionamento do Circuito

Os testes de funcionamento descritos a seguir tem como objetivo a validação do pleno funcionamento do circuito de captação dos sinais mioelétricos, isto é, se há amplificação do sinal de entrada da maneira projetada. Dessa forma, com o auxílio de um osciloscópio é possível visualizar a existência da amplificação do sinal. A Figura 16.A ilustra como o circuito foi montado para realização do teste, onde "osc" indica a palavra "osciloscópio". Na Figura 16.B é possível visualizar o circuito completo dos dois canais conectado ao Arduino.

4.3.3 Protocolo de Captura de Sinal de Eletromiografia

Para captura do sinal eletromiográfico os eletrodos foram posicionados seguindo protocolos das Recomendações Europeias para Eletromiografia de Superfície (SENIAM) (**Hermens et al., 2018**). Foram realizados testes preliminares com sinal dos flexores do cotovelo, cujo motor principal é bíceps braquial. Os eletrodos então foram posicionados na linha entre o acrômio, ponto referência do ombro, e a fossa ulnar, ponto referência na região do cotovelo (Figura 17) (**Hermens et al., 2018**).

Figura 16.A - Circuito para teste de funcionamento do circuito de captação dos sinais.



Fonte: Autores (2018)

Figura 16.B – Esquemático completo contemplando dois canais de eletromiografia e conexão com Arduino.

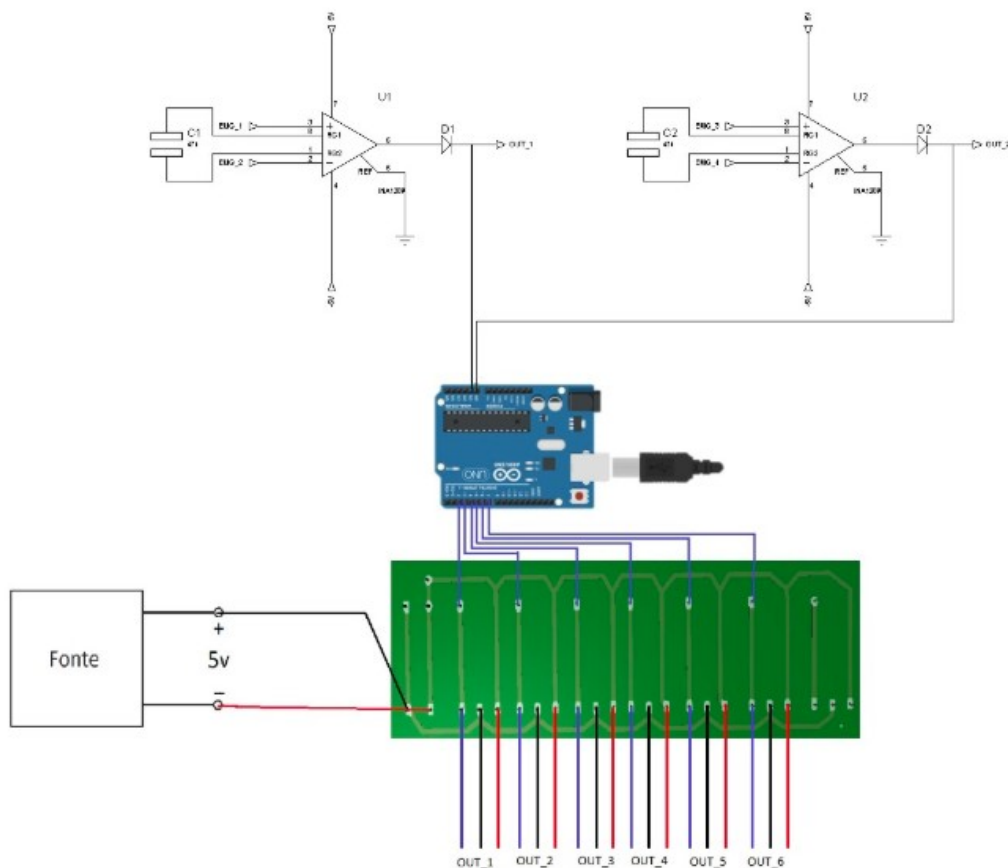
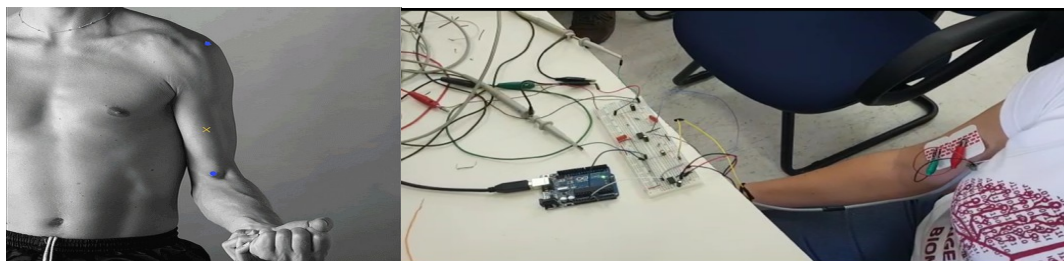


Figura 17 - Posicionamento do eletrodo no bíceps braquial. À esquerda recomendação da SENIAM e à direita posicionamento dos testes.



Fonte: SENIAM (Hermens et al., 2018) / Autores (2018).

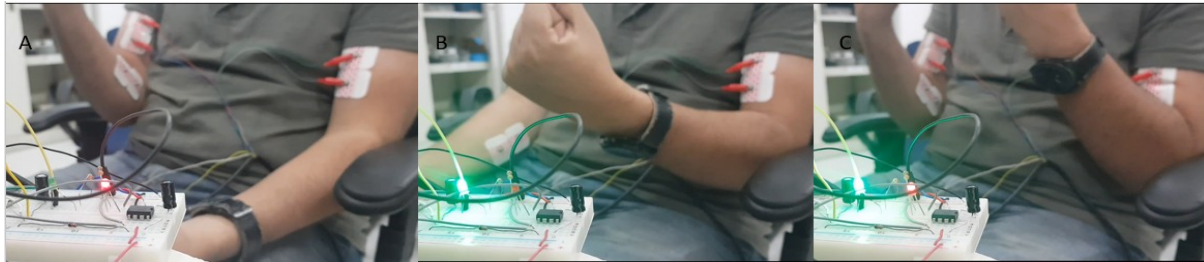
4.3.4 Resultados do Sistema de Aquisição do Sinal Eletromiográfico

De modo a verificar o funcionamento do circuito elaborado para conversão em controles de ativação, foram colocados leds de sinalização, os quais acendiam quando o limiar de contração muscular era atingido.

Nas imagens abaixo (Figura 18) observa-se que o LED vermelho acende quando o canal 1 é ativado a partir da contração muscular do bíceps direito; o LED verde

acende quando o canal 2, biceps esquerdo, é ativado e ambos os LEDs acendem quando a contração é realizada nos dois braços simultaneamente.

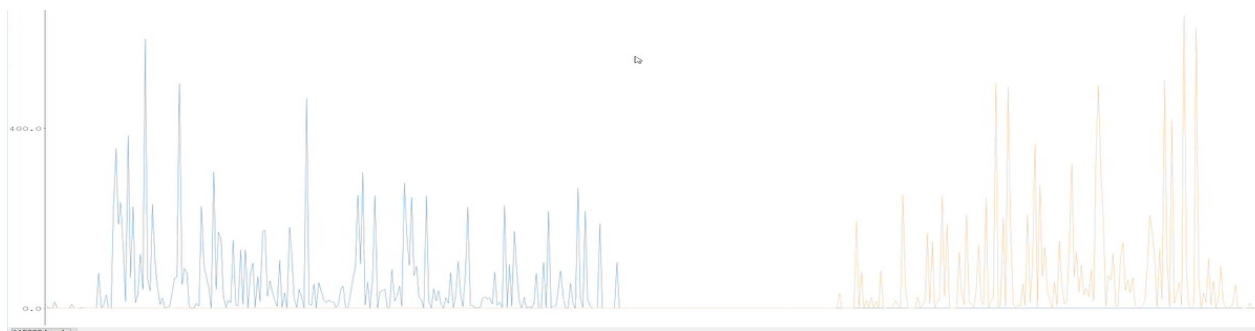
Figura 18 - Sinalização de LEDs ao contrair músculos



Fonte: Autores (2018).

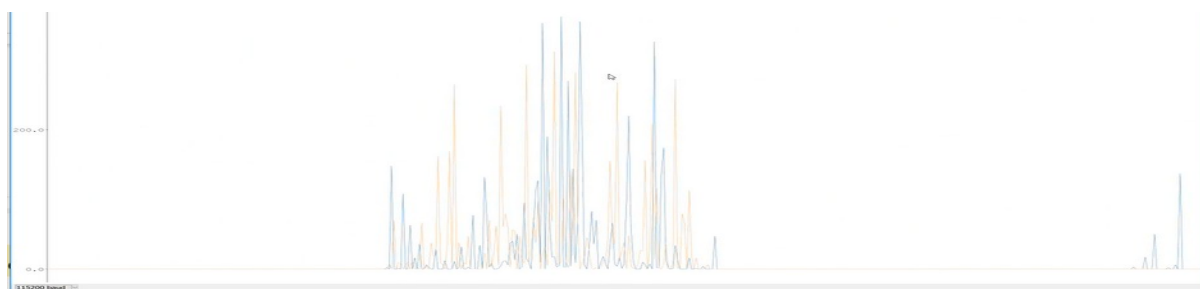
O sinal resultante das contrações de ambos os canais podem ser vistos nas Figuras 19 e 20.

Figura 19 - Sinal da contração muscular para o canal 1 e 2, azul e amarelo respectivamente, durante contração individual.



Fonte: Autores (2018).

Figura 20 - Sinal da contração muscular para os canais 1 e 2, azul e amarelo respectivamente, durante contração simultânea.



Fonte: Autores (2018).

Pelo link⁸ disponível, é possível ver, em formato de vídeo, uma representação quantitativa dos sinais mioelétricos captados do braço do usuário, enquanto que em um outro link⁹ disponível, a aplicação com os LEDs também pode ter seu funcionamento visualizado em vídeo.

4.4 Cinemática de Robôs

O robô (braço robótico) proposto possui 6 graus de liberdade (GDL) sendo dois graus diretamente ligados ao atuador da garra, isto é, consiste em 4-GDL de movimentação e 2-GDL de atuação (**Junior, 2014**). O manipulador em questão pode ser classificado como uma configuração resoluta pela presença de juntas prismáticas que transformam movimentos de rotação em translação.

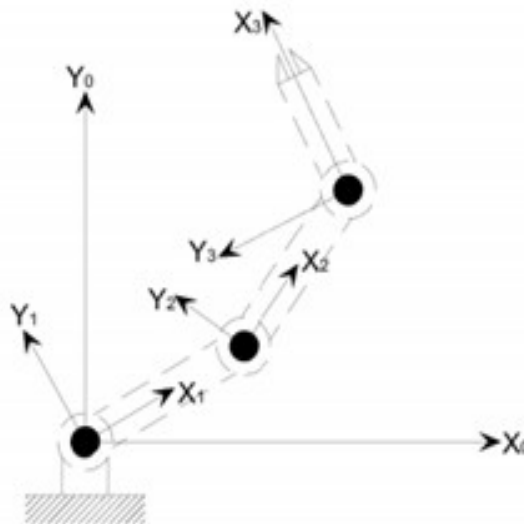
A configuração de construção do manipulador permite a simplificação de um sistema espacial (x, y, z) para um sistema plana com rotação no espaço (Ω, x, y) , com essa simplificação o controle do motor da base da estrutura pode ser feito de

⁸ “Vídeo_4: leitura dos sinais mioelétricos. - YouTube” <https://youtu.be/OdBUCEMv-yQ> Acessado em 30 de Set. de 2018.

⁹ “Vídeo_5: Vídeo_5: ascendendo LEDs com sinais mioelétricos. - YouTube” <https://youtu.be/vdQIsHqRMB4> Acessado em 30 de Set. de 2018.

forma independente reduzindo o manipulador a uma estrutura de 3-GDL tornando os cálculos mais simples para a cinemática direta e inversa (Figura 21) (Junior, 2014; Tartari Filho, 2006; Niku, 2010).

Figura 21 – Representação de um manipulador simplificado para forma planar com 3-GDL



Fonte: Junior (2014).

4.4.1. Cinemática Direta

É a nomenclatura do conjunto de equações capazes de determinar a posição, velocidade e aceleração de determinado ponto do corpo do robô. De forma conveniente é utilizado a ponta do atuador como ponto de referência para o equacionamento (Niku, 2010).

Existem diversas técnicas para o modelamento da cinemática direta, mas para o modelamento do braço robótico do desafio foi usado o Método de Denavit-Hartenberg por ser um método sistemático que analisa a orientação de posição de dois elementos sequenciais baseado em transformações homogêneas.

O método de Denavit-Hartenberg é o produto de 4 transformações, sendo 2 translações e 2 rotações, tendo como parâmetros o ângulo entre os elementos subsequentes, orientação, comprimento e deslocamento linear do mesmo. Sempre apresentação do mesmo padrão (Junior, 2014).

As vantagens do método consistem na simplicidade da análise do manipulador seguindo um passo a passo simples descritos por Junior (2014) em sua dissertação.

Passo 1: Nomear os eixos e as juntas de z_0, \dots, z_{n-1} . O eixo z_i deve ser o eixo de atuação da junta $i+1$. Assim, z_0 está no eixo de atuação da junta 1 e z_1 está

no eixo de atuação da junta 2. Existem dois casos a serem considerados: se a junta $i+1$ é de revolução, z_i é o eixo de revolução da junta $i+1$; se a junta $i+1$ é prismática, z_i é o eixo de translação da junta $i+1$.

Passo 2: Determine o *frame* da base. Coloque a origem em qualquer lugar ao longo do eixo z_0 . Os eixos x_0 e y_0 são escolhidos de acordo com a regra da mão direita.

Passo 3: Localize a origem o_i onde a normal comum de z_i e z_{i-1} intercepta z_i . Se z_i intercepta z_{i-1} , coloque o_i nessa interseção. Se z_i e z_{i-1} são paralelos, coloque o_i em qualquer posição conveniente ao longo de z_i .

Passo 4: Estabeleça x_i ao longo da normal comum entre z_{i-1} e z_i através de o_i ou na direção normal ao plano $z_{i-1}-z_i$ se z_{i-1} e z_i se cruzam.

Passo 5: Estabeleça y_i de acordo com a regra da mão direita.

Passo 6: Estabeleça o *frame* do *end-effector* $o_n x_n y_n z_n$. Assuma que a n -ésima junta é de revolução e estabeleça z_n na direção de Z_{n-1} . Estabeleça a origem o_n convenientemente ao longo de z_n , preferivelmente no centro da garra. Faça y_n na direção do fechamento da garra e x_n como $y_n \times z_n$. Se o *end-effector* não possuir uma garra, estabeleça x_n e y_n de forma conveniente de acordo com a regra da mão direita.

Passo 7: Crie uma tabela com os parâmetros dos elos, θ_i , a_i , d_i e α_i :

- θ_i : ângulo entre x_{i-1} e x_i medido sobre z_{i-1} , θ_i será variável se a junta i for de revolução.
- a_i : distância ao longo do eixo x_i de o_i até a interseção dos eixos x_i e z_{i-1} .
- d_i : distância ao longo de z_{i-1} de o_{i-1} até a interseção dos eixos x_i e z_i , d_i será variável se a junta i for prismática.
- α_i : ângulo entre z_{i-1} e z_i medido sobre x_i .

Passo 8: Forme a matriz de transformação homogênea A_i substituindo os parâmetros dos elos na Equação 5, descrita mais abaixo.

Passo 9: Faça ${}^0T_n = A_1 \dots A_n$. Assim obtêm-se a posição e orientação da ferramenta em relação ao sistema de coordenadas da base do robô.

Considerando o passo-a-passo do método de Devanit-Hartenberg descrito acima, chegou-se a modelagem do braço robótico 6-GDL apresentada no Quadro 3:

Quadro 3 - Modelagem segundo o método de Denavit-Hartenberg.

Elo	θ (rad)	d (mm)	α (°)	L (mm)
0-1	θ	0	0	103
1-2	α	0	0	98
2-3	$0.1258 + \sigma$	0	0	167.82

Fonte: Autores (2018).

4.4.2.1. Equações de cinemática direta

A equação de cinemática direta está representada na Equação 5, onde as 3 primeiras colunas mostram a orientação da garra do braço robótico, e a última coluna fornece as coordenadas espaciais da garra (x,y,z). A última linha é uma constante padrão para deixar a matriz homogênea.

$$T_3 = \begin{bmatrix} \cos(\theta + \alpha + \beta + 0.125) & -\sin(\theta + \alpha + \beta + 0.125) & 0 & 103 * \cos(\theta) + 98 * \cos(\theta + \alpha) + 167.82 * \cos(\theta + \alpha + \beta + 0.125) \\ \sin(\theta + \alpha + \beta + 0.125) & \cos(\theta + \alpha + \beta + 0.125) & 0 & 103 * \sin(\theta) + 98 * \sin(\theta + \alpha) + 167.82 * \sin(\theta + \alpha + \beta + 0.125) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equação 5

4.4.3 Cinemática inversa

A cinemática inversa é o conjunto de equações definidas a fim de determinar os valores das variáveis das juntas do manipulador tendo como entrada a posição espacial e orientação desejada para o atuador final (end-effector). A vantagem da utilização dessas equações é a possibilidade de determinar os valores coerentes para a posição final e controlá-los de forma simultânea e dependente, facilitando o controle e tornando mais fluido o movimento (**Junior, 2014**).

Embora as equações da cinemática inversa tenham sido determinadas, para o braço robótico, as mesmas não foram utilizadas por inviabilidade mecânica da mesma. Isso ocorreu devido à limitação dos motores disponibilizados, os quais não apresentavam torque suficiente para realizar os movimentos em algumas posições o qual o próprio peso do braço robótico é maior que o torque. Outro fator foi o tempo de resposta e o sistema eletrônico do motor ser em malha aberta.

4.4.3.1. Método Geométrico para cinemática inversa

Com base nas expressões de cinemática direta foi realizado o método geométrico para determinar as equações cinemática inversa, para determinar os ângulos α , θ e σ descritos no Quadro 3.

Para as equações de cinemática inversa tem-se as coordenadas (ϕ, x, y) , onde ϕ é a orientação da garra, isto é o ângulo que a garra faz com a horizontal. A primeira equação é definida de forma geométrica, sendo:

$$\phi = 0.125 + \alpha + \sigma - \theta$$

Equação 6

Sendo x_4 e y_4 as coordenadas da ponta da garra e x_3 e y_3 a junta anterior ao seguimento da garra. Dessa forma, com os pontos x_3 e y_4 é possível aplicar leis do cosseno no passo seguinte:

$$\begin{cases} x_3 = x_4 - L_3 \cos(\varphi) \\ y_3 = y_4 - L_3 \sin(\varphi) \end{cases}$$

Equação 7

Usando a lei dos cossenos é possível estabelecer a relação:

$$x_3^2 + y_3^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos(\alpha)$$

Equação 8

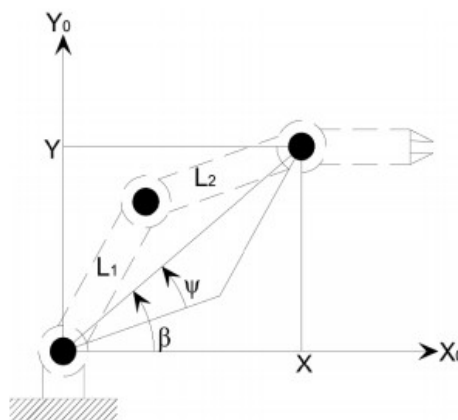
Logo,

$$\alpha = \cos^{-1} \left(\frac{x_3^2 + y_3^2 - (L_1^2 + L_2^2)}{2L_1L_2} \right)$$

Equação 9

O ângulo θ é determinado através dos ângulos auxiliares, como exibidos na Figura 22, onde β é o ângulo entre a tangente entre as coordenadas (x_3, y_3) e ψ agulho do vértice (x_1, y_1) do triângulo formado pelas três primeiras juntas do braço robótico em questão.

Figura 22 - Definição de ângulos auxiliares para a cinemática inversa



Fonte: Junior (2014).

Sendo β :

$$\beta = \tan^{-1} \left(\frac{y_3}{x_3} \right)$$

Equação 10

Usando novamente a lei dos cossenos para determinar ψ :

$$\psi = \cos^{-1} \left(\frac{x_3^2 + y_3^2 - (L_1^2 + L_2^2)}{2L_1\sqrt{x_3^2 + y_3^2}} \right)$$

Equação 11

Desse modo obtém-se:

$$\theta = \beta \pm \psi$$

Equação 12

É importante notar que pode assumir valores positivos e negativos, de tal forma que o sinal será determinado pelo valor do ângulo α , se $\alpha < 0$ $\psi > 0$ e $\alpha > 0$ $\psi < 0$. De modos práticos pelo controlados e variação de parâmetros, α não tente a assumir valores negativos.

Dessa forma tem-se que todas as equações de cinemática inversa para o braço robótico fornecido pelo desafio, onde α , θ , e σ foram definidos pela equação 9, 12 e 6 respectivamente, são:

$$\left\{ \begin{array}{l} x_3 = x_4 - L_3 \cos(\varphi) \\ y_3 = y_4 - L_3 \sin(\varphi) \\ \alpha = \cos^{-1} \left(\frac{x_3^2 + y_3^2 - (L_1^2 + L_2^2)}{2L_1L_2} \right) \\ \theta = \tan^{-1} \left(\frac{y_3}{x_3} \right) \pm \cos^{-1} \left(\frac{x_3^2 + y_3^2 - (L_1^2 + L_2^2)}{2L_1\sqrt{x_3^2 + y_3^2}} \right) \\ \sigma = \varphi + \theta - (0.125 + \alpha) \end{array} \right.$$

Equação 13

4.5 Aprendizado de Máquina

Por ter uma quantidade pré-definida de poses, o Myo Armband limita o usuário a desenvolver uma aplicação dependente dessas poses. Para agilizar o processo do controle do braço robótico, técnicas de aprendizagem de máquina foram aplicadas em um conjunto de dados colhidos pela equipe, com o objetivo de estender a quantidade original de poses.

As poses foram estendidas para 8, onde 6 são para a seleção de juntas no braço robótico e 2 para o controle de incremento e decremento naquela junta. O Quadro 4 informa a quantidade de dados para cada uma das poses descritas.

Quadro 4 - Quantidade de dados para cada pose.

Código	Descrição da pose	Quantidade de dados
EPN	Extensão do punho (neutro)	25.556
EPP	Extensão do punho (pronado)	25.271
FPN	Flexão do punho (neutro)	24.938
FPP	Flexão do punho (pronado)	25.033
MAN	Mão aberta (neutra)	25.433
MAP	Mão aberta (pronada)	25.660
MFN	Mão fechada (neutra)	25.442
MFP	Mão fechada (pronada)	25.538

Fonte: Autores (2018).

4.5.1 Abordagem por aprendizado raso

As técnicas de aprendizado raso abordam o problema sem a necessidade de implementação de classificadores grandes (em tamanho). Porém, essas técnicas trazem a necessidade de extração de características (*feature engineering*), já que a capacidade de generalização destes modelos é limitada pela sua falta de profundidade.

Para problemas comuns, como classificar entradas em um conjunto de dados, as abordagens por aprendizado raso costumam ter bons resultados. Porém, para algo como um sinal de eletromiografia, a variação dos dados de entrada não é tão diferente se analisadas como um todo, fazendo então com que haja a necessidade de extração de características.

Para todas as técnicas de aprendizado raso, algoritmos de busca em malha foram aplicados para fixar os hiperparâmetros dos modelos. Os hiperparâmetros são as características personalizáveis do modelo que permitem uma diferente generalização dos dados, logo, impactando diretamente no resultado final. Para cada técnica aplicada, os hiperparâmetros variados estarão descritos na seção da técnica. Utilizamos a média da validação cruzada com 3 *folds* como métrica.

4.5.1.1 Extração de Características

Ao total, 7 algoritmos foram aplicados para a extração de características. Esses algoritmos são descritos abaixo (Tkach, Huang e Huike, 2010):

- **Waveform length:** $wl = \log i$
- **Absolute mean value:** $\underline{x}_i = \frac{\sqrt{2}}{N} \sum_{k=1}^N i \vee x_k \vee i$
- **Difference absolute mean value:** $\underline{\Delta x}_i = \frac{1}{N-1} \sum_{k=1}^{N-1} i x_{k+1} - x_k \vee i i$
- **Root mean square:** $RMS = \sqrt{\frac{1}{N} \sum_{k=1}^N [x_k]^2}$
- **Average rectified value:** $x_i = \frac{1}{N} \sum_{k=1}^N i x_k \vee i i$
- **V-Order:** $vOrder = \sqrt[v]{E\{\vee x_k i^v\}}$, onde $v=2$
- **Log Detector:** $e^{\frac{1}{N} \sum_{k=1}^N \log i i i}$

4.5.1.2 Multilayer Perceptron¹⁰

As redes neurais artificiais foram aplicadas neste problema com funções de ativação não-lineares, com o objetivo de alcançar a não-linearidade do problema. O algoritmo de busca em malha foi aplicado, e o Quadro 5 indica os hiperparâmetros¹¹ que foram variados e quais obtiveram o melhor resultado.

¹⁰ "1.17. Neural network models (supervised) — scikit-learn 0.19.2"

http://scikit-learn.org/stable/modules/neural_networks_supervised.html. Acessado em 18 Set. 2018

¹¹ "sklearn.neural_network.MLPClassifier — scikit-learn 0.19.2" http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. Acessado em 18 Set. 2018

Quadro 5 - Busca em malha para o *Multilayer Perceptron*

Hiperparâmetro	Descrição	Variação	Melhor
hidden_layer_sizes	O enésimo elemento representa a quantidade de neurônios na enésima camada escondida.	(8), (16), (32), (64, 32), (64, 32, 16), (64, 32, 16, 8), (128, 64, 32, 16, 8)	(64, 32)
activation	Função de ativação para as camadas escondidas.	Sigmóide, ReLU, tanh	tanh
Solver	O otimizador usado para a variação dos pesos da rede neural.	SGD (Robins e Monro, 1985) e Adam (Kingma e Ba, 2014)	SGD
learning_rate	Comportamento da taxa de aprendizado de acordo com as épocas.	1) Constante : usa o valor em <i>learning_rate_init</i> ; 2) Adaptivo : divide o valor de <i>learning_rate_init</i> por 5 sempre que o sistema estiver se aproximando de um sobreajuste.	Adaptivo
learning_rate_init	A taxa inicial que controla o <i>step</i> na atualização dos pesos.	0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95	0.55
momentum	A taxa do Momento de Nesterov	[0., 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95]	0.9
Tempo total de execução	5.2 horas	Melhor % de acerto:	47%

Fonte: Autores (2018).

4.5.1.3 *Random Forests*¹²

Este algoritmo cria uma combinação de árvores de decisão, outra técnica de aprendizagem que originalmente não possui a capacidade de aproximação da

¹² "1.11. Ensemble methods — scikit-learn 0.19.2 documentation."

<http://scikit-learn.org/stable/modules/ensemble.html#forest>. Acessado em 18 Set. 2018

função que define a fronteira dos dados. Ao combinar várias árvores de decisão, uma curva não-linear consegue ser aproximada com mais acurácia.

O Quadro 6 indica quais parâmetros¹³ foram variados na busca em malha, juntamente com os parâmetros que obtiveram o melhor resultado.

4.5.1.4 Gradient Boosting¹⁴

O Gradient Boosting é uma generalização de um método de boosting que é robusto para outliers no espaço de saída. Foi analisado por possuir uma boa resistência ao sobreajuste.

O Quadro 7 indica quais parâmetros¹⁵ foram variados na busca em malha, juntamente com os parâmetros que obtiveram o melhor resultado.

Quadro 6 - Busca em malha para o *Random Forests*

Hiperparâmetro	Descrição	Variação	Melhor
n_estimators	O número de árvores na floresta	[5, 10, 15, 20, 25, 30, 35, 50, 100, 200, 500]	500
max_features	O número de features a ser considerado na hora de realizar a divisão da árvore.	auto, sqrt, log2	sqrt
max_depth	A profundidade máxima de cada árvore.	None, 4, 5, 6, 7, 8	None
Criterion	A função para medir a qualidade de cada divisão.	gini, entropy	'gini'
Tempo total de execução:	0.61 horas	Melhor % de acerto:	32%

Fonte: Autores (2018).

13 "3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn" <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>. Acessado em 18 Set. 2018

14 "1.11. Ensemble methods — scikit-learn 0.19.2 documentation." <http://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting>. Acessado em 19 Set. 2018

15 "3.2.4.3.5. sklearn.ensemble.GradientBoostingClassifier — scikit-learn" <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>. Acessado em 19 Set. 2018

4.5.2 Abordagem por aprendizado profundo

As técnicas de aprendizado profundo tornam a solução de problemas mais simples, pois automatiza completamente o passo mais crucial do fluxo de trabalho da abordagem rasa: a extração de características (Chollet, 2018).

Para a tarefa de detecção de pose de mãos, aplicamos redes neurais convolucionais profundas (DCNNs) no reconhecimento de Wavelets que foram gerados a partir dos dados de eletromiografia usando a biblioteca *PyWavelets*¹⁶. Os Wavelets são gerados ao fazer uma janela de 4 leituras, reduzindo então o tamanho do conjunto de dados para 25% do tamanho anterior. A Figura 23 mostra os wavelets gerados para cada movimento. O atrativo dessa abordagem é a não-necessidade de aplicação de técnicas de extração de características.

Quadro 7 - Busca em malha para o *Gradient Boosting*

Hiperparâmetro	Descrição	Variação	Melhor
learning_rate	Diminui a contribuição de cada árvore para o processo.	[0.01, 0.05, 0.1, 0.2]	0.05
max_depth	A profundidade máxima de cada árvore.	[1,3,5,8]	8
max_features	O número de features a ser considerado na hora de realizar a divisão da árvore.	log2, sqrt	sqrt
Subsample	A fração de amostras usadas para treinar cada classificador individual.	[0.5, 0.618, 0.8, 0.85, 0.9, 0.95, 1.0]	0.9
n_estimators	O número de classificadores na floresta.	[100, 200, 400, 500]	500
Tempo total de execução:	3.165 horas	Melhor % de acerto:	33%

Fonte: Autores (2018).

4.5.2.1 DenseNet

DenseNet é uma arquitetura onde cada camada é diretamente conectada com todas as outras camadas em um sentido direcional, agrupada em blocos. Para

¹⁶ "PyWavelets - Wavelet Transforms in Python" <https://pywavelets.readthedocs.io/>. Acessado em 18 Set. 2018

cada camada, os mapas de características de todas as camadas são tratadas como entradas, onde os seus mapas de características são tratadas como entradas nas próximas camadas. Apesar de sua densidade, essa arquitetura apresenta uma razoável quantidade de parâmetros.

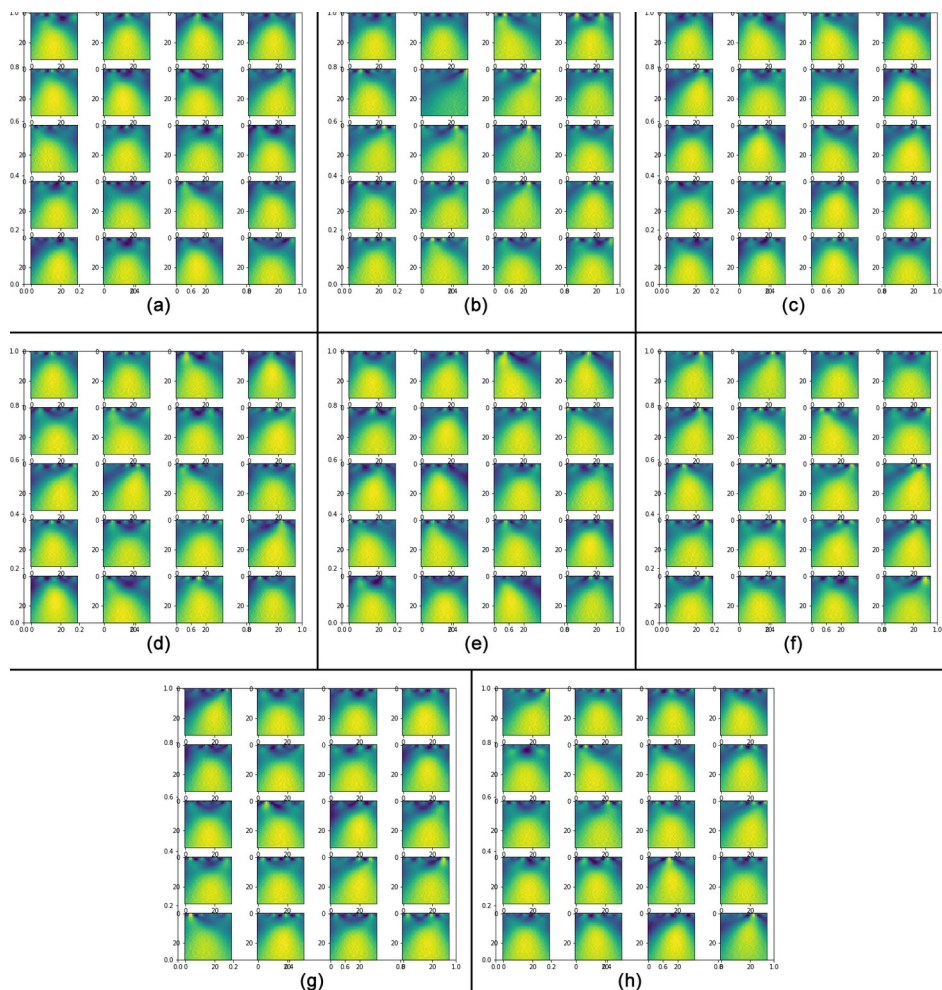
A DenseNet não obteve um bom resultado, pois não conseguiu generalizar corretamente os dados do treinamento. Atingiu apenas uma taxa de 12%, com sobreajuste a partir da 4ª época. Os blocos conectados de forma densa levaram ao sobreajuste em pouco tempo.

4.5.2.2 Inception-V3

As redes Inception fazem uso dos módulos inception. Esses módulos foram inspirados nas decisões que são necessárias em cada camada, como qual o tamanho dos filtros de convolução. A ideia é usar todas as configurações e permitir que um módulo de Inception escolha qual é a melhor para o modelo. Esses módulos permitem então que características locais sejam generalizadas por convoluções menores, e características globais sejam generalizadas por convoluções maiores.

A Inception-V3 obteve um resultado de 13% de acerto no conjunto de testes para as 8 poses, com o sobreajuste a partir da 8ª época. Para uma rede com esse nível de profundidade, mais dados seriam necessários para o treinamento.

Figura 23 - Wavelets gerados para cada movimento. **a)** EPN **b)** MFP **c)** MAN **d)** MFN **e)** MAP **f)** FPP **g)** FPN **h)** EPP



Fonte: Autores (2018).

Dois modelos de redes convolucionais foram aplicados: a DenseNet (Huang *et al.*, 2017) e a Inception-V3 (Szegedy *et al.*, 2015).

4.5.2.3 Discussão

Os dois modelos neurais profundos sofreram de sobreajuste com os dados de treinamento. O sobreajuste acontece quando o modelo não consegue generalizar os dados de teste, ficando muito ajustado aos dados de treinamento. Duas abordagens podem ser analisadas para resolver esses problemas: modelos mais rasos podem ser desenvolvidos para o treinamento, porém a falta de parâmetros livres para serem aprendidos levará o modelo a não conseguir aproximar a função de separação das classes, e mais dados podem ser colhidos para evitar o sobreajuste, que acontece com um conjunto de dados pequeno.

Para testarmos o desempenho com mais dados de treinamento, usamos um conjunto de dados maior, com movimentos diferentes do que os descritos anteriormente. Com esse conjunto de dados, conseguimos uma taxa de acerto de 90.4% usando o modelo Inception-V3, porém essa abordagem se mostrou ineficaz

para o uso *on-the-fly* no cenário de aplicação, pois seria necessária uma placa de vídeo com alta capacidade de memória, aumentando o custo final da solução.

4.6 Custos

De forma a propor uma solução que facilite o acesso à tecnologia a todos, o projeto foi elaborado visando um baixo custo financeiro. Nesta seção estão descritas duas tabelas de custo, sendo a primeira contendo o custo para a implementação do circuito mioelétrico, o qual utiliza dois canais; e a segunda com a solução utilizando o Myo Armband, o qual utiliza oito canais de sinais eletromiográficos.

4.6.1 Custos do sistema mioelétrico elaborado

Quadro 8 - Custos para o sistema mioelétrico

Componente	Quantidade	Valor (por unidade)
INA128p	2	R\$ 40,00
Capacitores Eletrolíticos	4	R\$ 0,15
Capacitores Cerâmicos	2	R\$ 0,25
Resistores	4	R\$ 0,15
Diodos	2	R\$ 1,28
Pacote de eletrodos descartáveis	1	R\$ 25,00
		Total = R\$ 109,26

4.4.2 Custos do controle utilizando Myo

Quadro 9 - Custos utilizando Myo

Componente	Quantidade	Valor (por unidade)
Myo Armband	1	R\$ 800,00
		Total = R\$ 800,00

5. Conclusão

Este relatório mostrou resultados preliminares positivos no controle do braço robótico por meio de eletromiografia, tanto com uso do Myo Armband como no uso do circuito de eletromiografia desenvolvido pela equipe. Para uma avaliação mais completa, ainda se faz necessário treino de coordenação do usuário para controle adequado no braço em diferentes atividades.

Foram realizados diferentes testes com o braço robótico em questão, que mostraram suas capacidades e limitações. Dentre suas principais limitações está a baixa potência dos motores das articulações de base, que impossibilita a execução de algumas posições, e consequentemente também a cinemática inversa. A mobilidade do braço robótico obteve sucesso na execução de algumas atividades funcionais, mas não em todas, principalmente devido à dificuldade em movimentos finos e ao baixo atrito na garra.

O Myo Armband foi utilizado para analisar as possibilidades do uso de múltiplos sinais como entrada para aprendizagem de máquina. Foram obtidos resultados positivos, porém com uso de processamento em computadores de alta performance, inviabilizando seu uso em tempo real de forma móvel, o que seria necessário para controle de próteses por exemplo. Além disso, a baixa taxa de amostragem do sinal obtido com o Myo Armband deixa o sinal eletromiográfico com baixa qualidade, dificultando a aprendizagem por técnicas de aprendizado de máquina otimizadas para tais circunstâncias.

Até o atual momento há duas soluções em vista. Uma delas é fazendo uso da plataforma Myo Armband e outra se dá pelo CI INA128P. Espera-se até o momento da demonstração aprimorar os testes e definir qual a melhor estratégia para a apresentação final do protótipo.

6. Referências

ABRAHAM, Zachary et al. Control of an affordable hand and wrist prosthesis. In: **15th Research Symposium, Rutgers School of Engineering**. 2015. p. 1-13.

ANNAMDAS, Venu Gopal Madhav; SOH, Chee Kiong. Application of Metamaterial Surface Plasmon and Waveguide for Robotic-Arm Based Structural Health Monitoring. **Journal of Nondestructive Evaluation**, v. 37, n. 2, p. 34, 2018.

BARROS, Kety Rosa de et al. Metodologia para classificação de sinais EMG para controle de próteses com baixo esforço computacional. 2005.

BERNHARDT, Paul. #MyoCraft: EMG in the Bluetooth Protocol. 2015. Myo BLOG. Disponível em: <http://developerblog.myo.com/myocraft-emg-in-the-bluetooth-protocol/> Acessado em: 22 de julho de 2017.

BERNHARDT, Paul. #MyoCraft Fly your Parrot AR.Drone 2.0 with AutoFlight. 2015. Myo BLOG. Disponível em: <http://developerblog.myo.com/myocraft-fly-your-parrot-ardrone-2-0-with-autoflight/> Acessado em: 22 de julho de 2017.

Carey, S. L., Stevens, P. M., e Highsmith, M. J. Differences in Myoelectric and Body-Powered Upper-Limb Prostheses: Systematic Literature Review Update 2013–2016. **Journal of Prosthetics and Orthotics**, 29, P17-P20, 2017.

CARVALHO, Gustavo Longhi de. **Proposta de um método de projeto de próteses de membros superiores com a utilização da engenharia e análise do valor**. 2004. Tese de Doutorado. Universidade de São Paulo.

Cooper, R. A., H. Ohnabe and D. A. Hobson, **An Introduction to Rehabilitation Engineering**, CRC Press, 2006.

De Oliveira, M. et al. Modelagem CAE de Próteses Mioelétricas **Proceedings do XXVI Congresso Brasileiro de Engenharia Biomédica**, 2018.

DE SOUSA, G. et al. Upper Limb Muscle Activation: An EMG Analysis Using Myo® Armband. **IFMBE Proceedings do XXVI Congresso Brasileiro de Engenharia Biomédica**, 2018.

DUQUE, Luciano Henrique; DE GODOI, Tomás da Silva Martins; DÍAZ, Francisco Javier De Obaldia. Prótese mioelétrica controlada por redes neurais. **Universitas: Gestão e TI**, v. 3, n. 2, 2014.

Hermens, H. J., B. Freriks, R. Merletti, G. Rau, C. Disselhorst-Klug, D. F. Stegeman and G. M. Hägg. **"SENIAM Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles."** from <http://www.seniam.org/>, 2018.

HUANG, Gao et al. Densely Connected Convolutional Networks. In: **CVPR**. p.3. 2017.

JEBSEN, Robert H. et al. An objective and standardized test of hand function. **Archives of physical medicine and rehabilitation**, v. 50, n. 6, p. 311-319, 1969.

JUNIOR, F. Estudo e implementação de redes neurais e algoritmos genéticos para resolução de cinemática inversa de um manipulador robótico com 5 graus de liberdade. **Universidade Estadual de Campinas, Campinas**, 2014.

KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KONRAD, Peter. The abc of emg. **A practical introduction to kinesiological electromyography**, v. 1, p. 30-35, 2005.

KONTSON, Kimberly et al. Targeted box and blocks test: Normative data and comparison to standard tests. **PloS one**, v. 12, n. 5, p. e0177965, 2017.

MACIEJASZ, P., et al. "A survey on robotic devices for upper limb rehabilitation." **Journal of NeuroEngineering and Rehabilitation** 11(1): 3, 2014.

MAY, Bella J.; LOCKARD, Margery A. **Prosthetics & orthotics in clinical practice: a case study approach**. FA Davis, 2011.

MELO, M. et al. Tecnologia Assistiva de manipulação para pessoas com mobilidade reduzida. **Proceedings do XXVI Congresso Brasileiro de Engenharia Biomédica**, 2018.

MERLETTI, Roberto; FARINA, Dario (Ed.). **Surface Electromyography: Physiology, Engineering, and Applications**. John Wiley & Sons, 2016.

MILLER, Laura A.; SWANSON, Shawn. Summary and recommendations of the academy's state of the science conference on upper limb prosthetic outcome measures. **JPO: Journal of Prosthetics and Orthotics**, v. 21, n. 9, p. P83-P89, 2009.

NETO, J. et al. Controle em miniatura de braço robótico aplicado à Reabilitação Motora. **Proceedings do XXVI Congresso Brasileiro de Engenharia Biomédica**, 2018.

NIKU, S. (2010). **An introduction to robotics analysis, systems, applications**. Hoboken, N.J.: Wiley.

ORTOLAN, R. L. **Estudo e Avaliação de Técnicas de Processamento de Sinal Mioelétrico para Controle de Sistemas de Reabilitação.** 133 p. 2002. Tese de Doutorado. Dissertação (Mestrado)-Universidade de São Paulo (USP), São Carlos-SP.

Royal Society. **The scientific century: securing our future prosperity.** RS Policy document 02/10. London: Royal Society; 2010.

TARTARI FILHO, Sylvio Celso. **Modelagem e otimização de um robô de arquitetura paralela para aplicações industriais.** 2006. Tese de Doutorado. Universidade de São Paulo.

ROBBINS, Herbert; MONRO, Sutton. A stochastic approximation method. In: **Herbert Robbins Selected Papers.** Springer, New York, NY, 1985. p. 102-109.

ROBOTICS, Coppelio. Virtual robot experimentation platform: User manual. **URL:** <http://www.coppeliarobotics.com/helpFiles/index.html>, 2015.

SZEGEDY, Christian et al. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition.** 2015. p. 1-9.

Thalmic Labs. Deliver Presentations with the Myo Armband. Disponível em: <<https://www.myo.com/present>> Acessado em: 22 de julho de 2017.

TKACH, Dennis; HUANG, He; KUIKEN, Todd A. Study of stability of time-domain features for electromyographic pattern recognition. **Journal of neuroengineering and rehabilitation**, v. 7, n. 1, p. 21, 2010.

VEERBEEK, Janne M. et al. Effects of robot-assisted therapy for the upper limb after stroke: a systematic review and meta-analysis. **Neurorehabilitation and neural repair**, v. 31, n. 2, p. 107-121, 2017.

7. Apêndice I – Código OpenCV

```
#include "SerialPort.h"
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>

using namespace cv;

void main() {

    SerialPort port;
    port.connect((char *)"COM2");

    char keys[] = {'Q','A','W','S', 'E','D','R','F','T','G','Y','H'};

    unsigned char angles[6];
    angles[0] = 150;
    angles[1] = 90;
    angles[2] = 45;
    angles[3] = 55;
    angles[4] = 75;
    angles[5] = 0;

    unsigned char answer[1];
```

```

Mat window;

window.create(640, 640, CV_8UC3);

Mat aninha = imread("aninha.jpg");

aninha.copyTo(window(Rect(Point(0, 0), Point(aninha.cols, aninha.rows))));

char jointtext[20];

while(true) {
    port.sendArray(angles, sizeof(char) * 6);

    rectangle(window, Rect(aninha.cols, 0, window.cols - aninha.cols,
aninha.rows), Scalar(0,0,0), -1);

    for (int i = 0; i < 6; i++) {
        sprintf(jointtext, "JOINT #%d (-%c <-> +%c): %d", i, keys[i*2+1],
keys[i*2+0], angles[i]);

        putText(window, jointtext, cvPoint(aninha.cols + 60, 200 + i*40),
                FONT_HERSHEY_COMPLEX_SMALL, 1, cvScalar(255,
255, 0), 1, CV_AA);
    }

    imshow("DETEB :: ARM Serial Control", window);

    int c = waitKey(100);

    int delta = 10;

    switch (c) {
        case 'q': angles[0] = min(angles[0] + delta, 180); break;
        case 'w': angles[1] = min(angles[1] + delta, 180); break;
        case 'e': angles[2] = min(angles[2] + delta, 180); break;
        case 'r': angles[3] = min(angles[3] + delta, 180); break;
        case 't': angles[4] = min(angles[4] + delta, 180); break;
        case 'y': angles[5] = min(angles[5] + delta, 180); break;
        case 'a': angles[0] = max(angles[0] - delta, 0); break;
    }
}

```



```
        case 's': angles[1] = max(angles[1] - delta, 0); break;
        case 'd': angles[2] = max(angles[2] - delta, 0); break;
        case 'f': angles[3] = max(angles[3] - delta, 0); break;
        case 'g': angles[4] = max(angles[4] - delta, 0); break;
        case 'h': angles[5] = max(angles[5] - delta, 0); break;
        case 27: exit(0);
    }
    port.getArray(answer, sizeof(char) * 1);
}
}
```

8. Apêndice II – Código do Arduino 1

```
#include <Servo.h>
```

```
Servo servos[6];
```

```
int pins[] = {7,6,5,4,3,2};
```

```
//unsigned char angles[] = {150,90,45,55,75,0};
```

```
unsigned char angles[] = {0,0,0,0,0,0};
```

```
void setup() {
```


```
    Serial.begin(9600);
```

```
    for(int i = 0; i < 6; i++) {  
        servos[i].attach(pins[i]);  
        servos[i].write(angles[i]);  
    }  
}
```

```
void loop() {
```

```
    unsigned char temp;
```

```
    if(Serial.available()>=6) {  
        for(int i = 0; i < 6; i++) {
```



```
temp = (unsigned char) Serial.read();
if(temp != angles[i]) {
    angles[i] = temp;
    //if(i < 1) {
    servos[i].write(angles[i]);
    //}
}

}

Serial.write(42); // envia resposta apenas para indicar que recebeu os 6 angulos
corretamente
}

delay(15);
}
```


8. Apêndice III - Código do Arduino 2

```
#include <Servo.h>

#define THRESH 10

#define DELAY 30


Servo servos[6];


int pins[] = {8,9,10,11,12,13};
//unsigned char angles[] = {150,90,45,55,75,0};
unsigned char angles[] = {0,0,0,0,0,0};


long currentTime = 0;
int currentJoint = 5;


int counter1 = 0;
int counter2 = 0;


void setup() {

    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);

    Serial.begin(115200);
```

```
for(int i = 0; i < 6; i++) {  
    servos[i].attach(pins[i]);  
    servos[i].write(angles[i]);  
}  
  
    currentTime = millis();  
}  
void loop() {  
  
    int sensorValue = analogRead(A0);  
    int sensorValue1 = analogRead(A1);  
  
    if(sensorValue>THRESH) counter1 = DELAY;  
    if(sensorValue1>THRESH) counter2 = DELAY;  
  
    bool on1 = counter1>0?true:false;  
    bool on2 = counter2>0?true:false;  
  
    if(on1) {  
        digitalWrite(6, HIGH);  
    } else {  
        digitalWrite(6, LOW);  
    }  
  
    if(on2) {  
        digitalWrite(7, HIGH);  
    } else {
```

```
digitalWrite(7, LOW);
}

if(millis()-currentTime>1000) {
  currentTime = millis();
  if(on1 && !on2) {
    angles[currentJoint] = min(180, angles[currentJoint]+10);
    servos[currentJoint].write(angles[currentJoint]);
  } else if(!on1 && on2) {
    angles[currentJoint] = max(0, angles[currentJoint]-10);
    servos[currentJoint].write(angles[currentJoint]);
  } else if(on1 && on2) {
    currentJoint = (currentJoint +1)%6;
  }
  Serial.print(currentJoint);
  Serial.print(" ");
  Serial.println(angles[currentJoint]);
}

counter1 = (counter1 > 0)?counter1 - 1:0;
counter2 = (counter2 > 0)?counter2 - 1:0;

delay(10);
}
```