

Universidade Federal de Juiz de Fora  
Engenharia  
Programa de Pós-Graduação em Engenharia Elétrica

**Diego Albuquerque Carvalho**

**Desenvolvimento de um Sistema de Controle para Cadeiras de Rodas  
Automatizada Baseado em Movimentos da Cabeça**

Juiz de Fora

2018

Diego Albuquerque Carvalho

Desenvolvimento de um Sistema de Controle para Cadeiras de Rodas  
Automatizada Baseado em Movimentos da Cabeça

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Eletrônicos, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. André Luís Marques Marcato

Juiz de Fora

2018

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Carvalho, Diego.

Desenvolvimento de um Sistema de Controle para Cadeiras de Rodas  
Automatizada Baseado em Movimentos da Cabeça / Diego Albuquerque  
Carvalho. – 2018.

75 f. : il.

Orientador: Prof. Dr. André Luís Marques Marcato

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2018.

1. Acessibilidade. 2. Tetraplegia. 3. Educação em Engenharia. I. Marcato, André, orient. II. Título.

**Diego Albuquerque Carvalho**

**Desenvolvimento de um Sistema de Controle para Cadeiras de Rodas  
Automatizada Baseado em Movimentos da Cabeça**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Eletrônicos, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em 20 de fevereiro de 2018

**BANCA EXAMINADORA**

---

Prof. Dr. André Luís Marques Marcato - Orientador  
Universidade Federal de Juiz de Fora

---

Professor Dr. Tales Pulinho Ramos  
Instituto Federal de Educação, Ciência e Tecnologia do  
Sudeste de Minas Gerais - Campus Juiz de Fora

---

Professor Dr. Ivo Chaves da Silva Junior  
Universidade Federal de Juiz de Fora

## AGRADECIMENTOS

Agradeço primeiramente a Deus por guiar meu caminho e permitir que buscasse sempre melhorar profissionalmente e principalmente ser uma pessoa melhor. Agradeço aos meus pais por todo o apoio, por sempre estarem comigo em cada novo desafio. Aos meus familiares e à minha noiva também deixo meu Muito Obrigado, pois com certeza, sem vocês talvez não fosse possível.

Ao meu orientador André Marcato, agradeço pela ajuda no desenvolvimento desta dissertação, mostrando os passos a serem seguidos e receptividade com meus questionamentos e desenvolvimentos. Agradeço também aos meus amigos do Grupo de Robótica Inteligente (GRIn - UFJF) por todo o apoio neste ciclo e, em especial, ao Guilherme Marins pelo companheirismo e todos os trabalhos que pudemos desenvolver em conjunto.

Agradeço também às agências de fomento que tornaram este trabalho possível, sendo elas, o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), a Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG) e a Universidade Federal de Juiz de Fora (UFJF).

"'Deficiente' é aquele que não consegue modificar sua vida, aceitando as imposições de outras pessoas ou da sociedade em que vive, sem ter consciência de que é dono do seu destino."

(MÁRIO QUINTANILHA)

## RESUMO

No cenário mundial atual há meio bilhão de pessoas com algum tipo de deficiência. Com relação ao Brasil, cerca de 45 milhões de pessoas possuem deficiência e, dentre elas, 65,9% recebem até um salário mínimo, indicando que esta população é composta majoritariamente por indivíduos pobres e, no entanto, as cadeiras de rodas motorizadas e outros equipamentos de acessibilidade possuem preços, em geral, inacessíveis para as pessoas que necessitam de tais recursos. Nesta perspectiva, este trabalho desenvolve um capacete de baixo custo que percebe os movimentos da cabeça do usuário e aciona uma cadeira de rodas adaptada para indivíduos com tetraplegia. Com isso, utilizou-se componentes de baixo custo para montagem do projeto, tais como Arduino<sup>®</sup>, IMU, entre outros. É necessário realizar calibragem e filtragem dos dados obtidos pelo IMU e implementar simulações para avaliar se a proposta responde corretamente aos estímulos apresentados. Ademais, realizou-se uma simulação em ambiente virtual e também com robô móvel para avaliar desafios com relação a ambientes diversificados e envio de informações através do capacete e alcançar um protótipo funcional e econômico para os indivíduos em questão. Além disso, com uma abordagem voltada para educação em engenharia, desenvolveu-se um tutorial para desenvolvimento deste protótipo por alunos de engenharia, cursos técnicos ou engajados neste ramo de pesquisa auxiliando tanto a formação dos alunos quanto possibilitando o constante desenvolvimento desta estrutura a fim de atender melhor os deficientes que necessitam desta tecnologia.

Palavras-chave: Acessibilidade, Tetraplegia, Educação em Engenharia.

## ABSTRACT

In the world scenario there is half billion of people with some kind of disability. In Brasil, there are around 45 millions of people with disabilities and, among them, 65,9% earn monthly up to the minimum wage, wich indicates that this population is mainly composed by poor individuals and, however, motorized wheelchairs and other acessibility equipments have a market price that is not affordable for this part of population. In this perspective, this work develops a low cost helmet that reads the user head movements and triggers the adapted wheelchair for individuals with tetraplegia. Then, was used low cost components in the project as Arduino<sup>®</sup>, IMU, among others. Simulations were made to evaluate the system in virtual system and a mobile robot that intends determine environment limitations, filtering and achieve an economical and funcional prototype. Furthermore, concerned with engineering education, was developed a tutorial for this prototype development by engineering students or engaged people in this area, working in benefit of education and improving the project constantly.

Key-words: Acessibility, Tetraplegia, Engineering Education



## LISTA DE ILUSTRAÇÕES

Figura 1 – <i>Roll, Pitch e Yaw</i> . . . . .	12
Figura 2 – Diagrama de Movimentos . . . . .	13
Figura 3 – Fluxograma para reconhecimento dos padrões . . . . .	22
Figura 4 – Modelo RNA . . . . .	26
Figura 5 – Estrutura Neurônio Artificial . . . . .	26
Figura 6 – Distribuição Gaussiana . . . . .	29
Figura 7 – Medidas Pioneer 3-DX . . . . .	35
Figura 8 – Classes de Atuação . . . . .	39
Figura 9 – IMU . . . . .	40
Figura 10 – Rotações nos eixos . . . . .	44
Figura 11 – Posição final após as rotações . . . . .	44
Figura 12 – Distribuição de Dados . . . . .	48
Figura 13 – Treinamento de uma RNA . . . . .	49
Figura 14 – Configuração de rede A . . . . .	49
Figura 15 – Configuração de rede B . . . . .	49
Figura 16 – Configuração de rede C . . . . .	50
Figura 17 – Configuração de rede D . . . . .	50
Figura 18 – Configuração de rede E . . . . .	50
Figura 19 – Configuração de rede F . . . . .	50
Figura 20 – Arquitetura de Redes Neurais Artificiais . . . . .	51
Figura 21 – Função <i>Purelin</i> . . . . .	53
Figura 22 – Função <i>Sigmoide</i> . . . . .	54
Figura 23 – Função <i>Radbas</i> . . . . .	55
Figura 24 – Divisão de Classes . . . . .	56
Figura 25 – Mapa utilizado em MatLab . . . . .	57
Figura 26 – Primeiro mapa utilizado em Gazebo <sup>®</sup> . . . . .	58
Figura 27 – Segundo mapa utilizado em Gazebo <sup>®</sup> . . . . .	58
Figura 28 – Teste com robô móvel . . . . .	59
Figura 29 – Conexões Arduino com IMU . . . . .	62
Figura 30 – Fluxograma Arduino . . . . .	62
Figura 31 – Fluxograma MATLAB - Recepção de Dados Serial . . . . .	63
Figura 32 – Fluxograma ROS . . . . .	64
Figura 33 – Resposta do Sistema de Exemplificação . . . . .	67

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
UFJF	Universidade Federal de Juiz de Fora
IBGE	Instituto Brasileiro de Geografia e Estatística
IMU	<i>Inertial Measurement Unit</i>
EMG	Eletromiograma
EEG	Eletroencefalografia
ROS	<i>Robotic Operating System</i>
VoIP	Voz sobre protocolo de internet
BCI	<i>Brain Computer Interface</i>
LMS	<i>Laser Measurement Sensor</i>
RNA	Redes Neurais Artificiais
MMQ	Método dos Mínimos Quadrados
AVA	Ambiente Virtual de Aprendizagem

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>11</b>
1.1	OBJETIVOS GERAIS . . . . .	12
1.2	OBJETIVOS ESPECÍFICOS . . . . .	12
1.3	ESTRUTURA DO TRABALHO . . . . .	13
1.4	PUBLICAÇÕES . . . . .	14
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA . . . . .</b>	<b>16</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>22</b>
3.1	RECONHECIMENTO DE PADRÕES . . . . .	22
3.1.1	<b>Distância Euclidiana . . . . .</b>	24
3.1.2	<b>Distância de Mahalanobis . . . . .</b>	24
3.1.3	<b>Redes Neurais Artificiais . . . . .</b>	25
3.2	COMPONENTES DE HARDWARE . . . . .	27
3.2.1	<b>Orientação . . . . .</b>	27
3.2.2	<b>Arduino® . . . . .</b>	28
3.3	FILTRAGEM DE ERROS . . . . .	28
3.3.1	<b>Filtro de Kalman . . . . .</b>	29
3.4	ROBÓTICA MÓVEL . . . . .	33
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>37</b>
4.1	MATERIAIS . . . . .	37
4.2	PLATAFORMA ROS . . . . .	38
4.3	AQUISIÇÃO E TRATAMENTO DE DADOS . . . . .	38
4.4	ROTINA DE CALIBRAÇÃO . . . . .	39
4.4.1	<b>Calibração do Acelerômetro . . . . .</b>	39
4.4.2	<b>Implementação do Método dos Mínimos Quadrados . . . . .</b>	40
4.4.3	<b>Calibração Individual . . . . .</b>	45
<b>5</b>	<b>IMPLEMENTAÇÃO E RESULTADOS . . . . .</b>	<b>47</b>
5.1	IMPLEMENTAÇÃO DE RNA . . . . .	48
5.1.1	<b>Quantidade de Camadas Ocultas . . . . .</b>	49
5.1.2	<b>Número de Neurônios na Camada Oculta . . . . .</b>	51
5.1.3	<b>Quantidade de Épocas . . . . .</b>	52
5.1.4	<b>Função de Ativação . . . . .</b>	52
5.1.4.1	<i>Função Linear . . . . .</i>	52
5.1.4.2	<i>Função Sigmoide . . . . .</i>	52

5.1.4.3	<i>Radbas</i> . . . . .	53
5.1.5	<b>Comparação entre usuários diferentes</b> . . . . .	54
5.2	SIMULAÇÃO EM MATLAB® . . . . .	55
5.3	SIMULAÇÃO NO GAZEBO® . . . . .	57
5.4	TESTE NO PIONEER 3DX . . . . .	58
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	<b>60</b>
	<b>APÊNDICE A – Tutorial do Módulo Assistivo</b> . . . . .	<b>61</b>
A.1	PLATAFORMA ROS . . . . .	61
A.2	MONTAGEM . . . . .	61
A.3	PROGRAMAÇÃO EM ARDUINO . . . . .	61
A.4	RECEPÇÃO DE DADOS NO MATLAB® . . . . .	63
A.5	PROGRAMAS PARA A PLATAFORMA ROS . . . . .	64
A.6	OBTENÇÃO DA MATRIZ DE CORREÇÃO . . . . .	65
	<b>ANEXO A – Tabelas de Shapiro-Wilk</b> . . . . .	<b>68</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>72</b>

## 1 INTRODUÇÃO

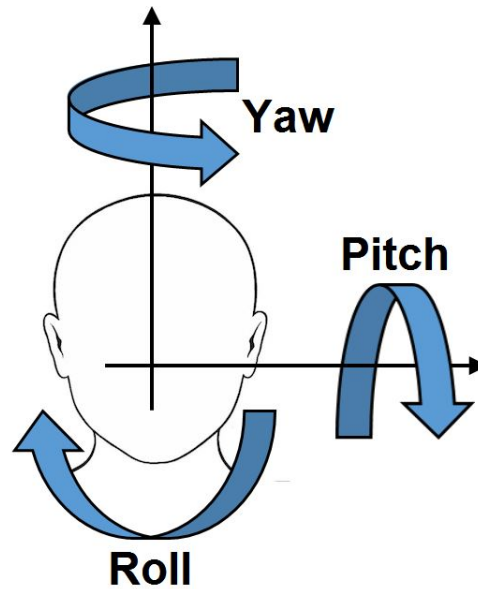
Atualmente, cerca de 500 milhões de pessoas no mundo possuem algum tipo de deficiência [Ceres et al., 2005]. Com isso, a preocupação com as dificuldades diárias que pessoas com deficiência enfrentam aumentaram. Devido a isso, diversos trabalhos estão sendo desenvolvidos na perspectiva de facilitar seu cotidiano [Jayawardena et al., 2013, Duan et al., 2014, Mitchell et al., 2014, Salhi et al., 2016, Nguyen et al., 2016]. Percebeu-se durante a última década um grande esforço na promoção de tecnologias no ramo de acessibilidade. Esse desenvolvimento ocorreu devido ao rápido crescimento das tecnologias, como por exemplo, no ramo de biomateriais, informática, microeletrônica e sensoriamento [Ceres et al., 2005].

De acordo com o censo de 2010 do Instituto Brasileiro de Geografia e Estatística (IBGE) mais de 45 milhões de brasileiros possuem algum tipo de deficiência, o que representa 23,9% da população brasileira. Além disso, 65,9% das pessoas com deficiência recebem até um salário mínimo, ou seja, 30 milhões de brasileiros possuem deficiência e estão em situação de vulnerabilidade econômica [Censo, 2010]. Baseado nisso, este trabalho propõe uma alternativa mais econômica do que as opções praticadas no mercado atualmente para indivíduos com tetraplegia. De acordo com a Organização Mundial da Saúde (OMS), entre 250 e 500 mil pessoas sofrem anualmente lesões na medula espinal [Organization and Society, 2013]. Sabendo que aqueles que sofrem lesões abaixo da quarta vértebra da coluna cervical são, normalmente, capazes de realizar movimentos de pescoço [de Lima, , Young, ]. Desta forma, este trabalho propõe uma alternativa para indivíduos com tetraplegia utilizando movimentos do pescoço.

A fim de modelar os movimentos de uma aeronave utiliza-se as variáveis *Roll*, *Pitch* e *Yaw* que definem movimentos de rotação em torno dos eixos cartesianos. Aplicando ao caso que será descrito neste trabalho, considere um eixo cartesiano na cabeça do usuário e as definições são:

- *Roll*: Movimento da cabeça para a direita e esquerda;
- *Pitch*: Movimento da cabeça para frente e atrás;
- *Yaw*: Movimento circular da cabeça.

Vale destacar que o movimento de *Yaw* não é utilizado nesta aplicação. Os movimentos de *Roll* e *Pitch* são suficientes para a movimentação desejada neste trabalho, ou seja, para o acionamento da cadeira de rodas. Na Figura 1 é apresentado visualmente a explicação das variáveis *Roll*, *Pitch* e *Yaw*.

Figura 1 – *Roll, Pitch e Yaw*

Fonte: [Marins et al., 2017]

## 1.1 OBJETIVOS GERAIS

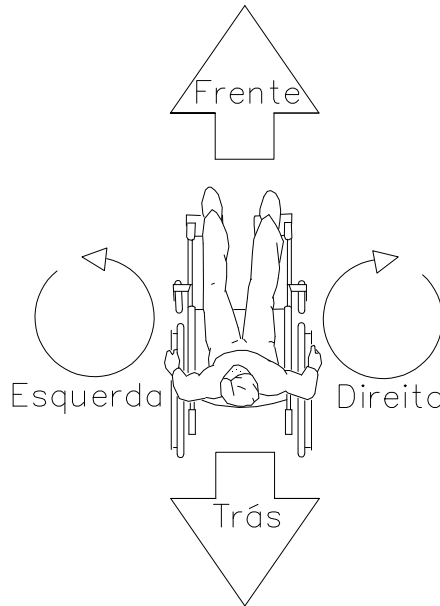
O objetivo do módulo desenvolvido neste trabalho é de atender indivíduos com tetraplegia ou limitações correlatas. A cadeira de rodas é movimentada por movimentos da cabeça do indivíduo, desta forma, torna-se importante que o usuário consiga realizar os movimentos de pescoço satisfatoriamente. A fim de atingir a maior parte do público alvo é necessário desenvolver um módulo econômico com a utilização de componentes de baixo custo. Ademais, realizou-se simulações para avaliar o funcionamento e implementar melhorias na plataforma. A comparação entre o ambiente simulado e o prático permite analisar e implementar melhorias no projeto. Vale salientar que, neste projeto, não foi possível desenvolver um experimento prático em uma cadeira de rodas, no entanto, foi realizada uma análise com um robô móvel diferencial *Pioneer 3-DX* para avaliar a resposta do algoritmo implementado [MobileRobots, ].

## 1.2 OBJETIVOS ESPECÍFICOS

Em linhas gerais, esta dissertação propõe o desenvolvimento de um capacete que seja capaz de compreender os movimentos de cabeça de um usuário com deficiência. Atua-se em uma cadeira de rodas a partir dos dados recebidos do módulo acoplado ao capacete, permitindo que o usuário acione a cadeira com movimentos provenientes da cabeça. Além disso, o desenvolvimento neste trabalho utilizou uma simulação que buscou se aproximar de uma cadeira de rodas e estimar o comportamento do sistema para um caso real. Os componentes utilizados para o desenvolvimento e simulação do modelo serão especificados nas seções 3.2 e 3.4.

Para o reconhecimento dos padrões de movimento foram analisados os métodos da distância Euclidiana, de Mahalanobis e Redes Neurais Artificiais (RNA) conforme detalhados na seção 3.1, atribuindo corretamente cada movimento à sua posição e permitindo ao usuário movimentar a cadeira [Marins et al., 2017]. Com isso, o trabalho desenvolve ambientes de simulação e teste com um robô móvel diferencial para análise da resposta do sistema. Os padrões de movimentos podem ser observados na Figura 2.

Figura 2 – Diagrama de Movimentos



Fonte: [Marins et al., 2017]

Propõe-se neste trabalho, além do desenvolvimento do módulo assistivo, a utilização do desenvolvimento teórico utilizado para ensino em engenharia. Desenvolveu-se um "Tutorial" de forma simplificada para que os alunos do curso de graduação possam primeiramente aprender os conceitos técnicos abordados, implementar modificações, aprimorar o projeto e, principalmente, estimular o trabalho em equipe e o pensamento crítico para que possam desenvolver outras tecnologias para o mercado atual e se tornarem profissionais com base técnica e competências transversais. Todos os códigos e esta dissertação estão disponíveis no [github](https://github.com/diegocarvalho12/wheelchairhm)<sup>®</sup> no link <https://github.com/diegocarvalho12/wheelchairhm>.

### 1.3 ESTRUTURA DO TRABALHO

No fluxo deste trabalho, no capítulo 2 há uma revisão bibliográfica sobre a área de estudo apresentada nesta dissertação, destacando alguns trabalhos relevantes para a proposta apresentada. No capítulo 3 realiza-se uma fundamentação teórica deste trabalho abordando conceitos importantes para esta dissertação a fim de facilitar o entendimento dos leitores da abordagem realizada com lógica *fuzzy*, técnicas de reconhecimento de padrões, filtro de Kalman, robótica móvel e alguns elementos de *hardware* fundamentais

para o direcionamento e entendimento da proposta apresentada. No capítulo 4 apresenta-se os materiais utilizados, a forma de aquisição de dados, a calibração do acelerômetro e o tratamento dos dados. Em seguida, no capítulo 5 apresenta-se a simulação da cadeira de rodas utilizando o *software* MatLab<sup>®</sup>, apresentando os resultados simulados obtidos e no capítulo 5 encontra-se o experimento realizado no *Pioneer 3-DX*.

#### 1.4 PUBLICAÇÕES

Ao decorrer deste trabalho foi produzido um artigo que aborda o desenvolvimento do projeto e apresentado no *Intelligent Systems Conference* (IntelliSys) em 2017 [Marins et al., 2017].

Ademais, de forma decorrente do estudo outros dois artigos com base educacional, buscando alternativas para melhorar a educação em Engenharia, foram publicados no Congresso Brasileiro de Educação em Engenharia em 2016 (COBENGE) [Carvalho et al., 2016] e no Brazilian Power Electronics Conference (COBEP) em 2017 [Lobato et al., 2017].

- O artigo [Marins et al., 2017] apresenta o desenvolvimento da cadeira de rodas com análise dos métodos de Mahalanobis, distância Euclidiana e Redes Neurais Artificiais, apresentando as respostas obtidas nos resultados de simulação e comparações entre os métodos de reconhecimento de padrões a fim de estimar os parâmetros adequados para esta aplicação, pois deve ter alta precisão com o objetivo de aumentar o conforto e a segurança do usuário além de processamento rápido para aplicações embarcadas.
- O artigo [Carvalho et al., 2016] utiliza uma estrutura de dois tanques de água em que realiza transposição e aquecimento de água para o controle de volume e temperatura. A interface desenvolvida para o módulo permite que o estudante interaja com a estrutura, aprendendo conceitos de filtragem de sinais, resposta em degrau, otimização e outros conceitos inerentes às técnicas de controle.
- O artigo [Lobato et al., 2017] implementa uma placa de eletrônica de potência para o aprendizado de alunos do curso de graduação em engenharia elétrica possibilitando a implementação de um conversor *buck*, análise de resposta em frequência e outros conteúdos inerentes às disciplinas de Controle e Eletrônica de Potência auxiliando no ensino de tais disciplinas com uma estrutura que permite, além de desenvolver o conhecimento técnico, possibilite o desenvolvimento do pensamento crítico, análise, desenvolvimento e aprimoramento das tecnologias, dentre outras competências fundamentais necessárias para o engenheiro atualmente.

Neste capítulo foi descrito os objetivos gerais e específicos do trabalho que se baseiam no público alvo que são indivíduos com tetraplegia e de baixa renda, com o



desenvolvimento de uma cadeira de rodas que permita sua locomoção com movimentos do pescoço. Ademais, descreve-se o artigo decorrente deste trabalho e dois artigos correlatos. Com isso, tem-se uma exposição geral do cenário desta dissertação.

## 2 REVISÃO BIBLIOGRÁFICA

Diversas pesquisas foram realizadas a fim de atender a indivíduos que utilizam cadeiras de rodas. Com isso, para abordar as variações de deficiências físicas, grande parte das pesquisas buscam alternativas que auxiliem em deficiências específicas. Neste capítulo serão descritos diversas produções correlatas com a finalidade de subsidiar o trabalho desenvolvido.

O artigo [Jayawardena et al., 2013] apresenta uma cadeira robótica assistiva que permite o controle por voz ou através de uma tela interativa. Além disso, o equipamento realiza aferição de pressão e controle de batimentos cardíacos do usuário. Para o sensoria-mento do ambiente utiliza-se um sensor *kinect*. Vale salientar que esta proposta atende indivíduos tetraplégicos e permite controlar sua saúde de modo facilitado através dos equipamentos conectados de avaliação de saúde. Além disso, permite que usuários com pouca mobilidade no pescoço utilizem o equipamento de forma fácil.

Alternativas para sistemas de controle compartilhado são apresentadas nos artigos [Mitchell et al., 2014, Duan et al., 2014, Rohmer et al., 2015a, Olivi et al., 2013, Li et al., 2016, Tonin et al., 2010, Zhang et al., 2014, Ceres et al., 2005]. Em [Mitchell et al., 2014] tem-se uma alternativa de cadeira de rodas destinada a idosos ou pessoas com mobilidade prejudicada o qual evita colisões, utilizando um sistema de controle compartilhado que identifica obstáculos e ajusta a trajetória do usuário a fim de proporcionar maior facilidade e segurança para o indivíduo. Em [Duan et al., 2014] tem-se um controle compartilhado utilizando uma interface não-invasiva do computador com a mente e utiliza um filtro de Kalman estendido para aquisição dos dados do módulo. O artigo [Rohmer et al., 2015a] utiliza um sensor *kinect* para controle compartilhado e análise do ambiente que avalia a possibilidade de percorrer um trajeto determinado e recebe uma confirmação por movimentos faciais através de um eletromiograma (EMG) colocado sobre a cabeça do usuário. Em [Olivi et al., 2013], o controle compartilhado é assegurado com o uso da teoria de campos potenciais que é capaz de avaliar os obstáculos e desviar com intensidade proporcional à distância ao obstáculo.

O trabalho [Li et al., 2016] utiliza a eletroencefalografia (EEG), que através de eletrodos não-invasivos registra a atividade elétrica do cérebro, permitindo que indivíduos com nenhuma movimentação de membros e do pescoço, realize a condução da cadeira de rodas. Além disso, utiliza um sensor de laser com boa acurácia de distância do tipo SICK Laser Measurement Sensor (LMS) para realizar leituras do ambiente e ajustar a trajetória do usuário e evitar colisões.

Os artigos [Nguyen et al., 2016, Tonin et al., 2010] apresentam controle telepresencial de uma cadeira de rodas ou um módulo de movimentação, mostrando a possibilidade e eficiência de sistemas utilizando tecnologia *VoIP* (*Voice over Internet Protocol*) para enviar

informações para uma cadeira automatizada. O artigo [Nguyen et al., 2016] mostrou-se eficiente em condições de ambientes externos e uma contribuição relevante para envio e tratamento de dados neste tipo de operação, principalmente pelos fatores não controláveis no meio externo como linearidade do solo e exposição de *hardware* a ações do tempo e ainda envio de dados que podem sofrer com maiores interferências no meio externo. Já o trabalho [Tonin et al., 2010] apresenta resultados satisfatórios com relação ao controle do módulo e ainda utilizam nove sensores infravermelhos que evitam colisão, apresentando a possibilidade de controle compartilhado no módulo a fim de melhorar a experiência do usuário final.

O artigo [Zhang et al., 2014] utiliza um conjunto de treze sensores ultrassônicos com o conceito de campo de força virtual em busca de uma alternativa econômica. As alternativas de controle compartilhado facilitam a atividade do usuário, pois, evitam colisões de forma automática. O artigo [Koren and Borenstein, 1991] utiliza campo de força virtual para movimentação de uma robô móvel, reforçando uma técnica alternativa para evitar colisões e aprimorar a movimentação do robô.

Ademais, a utilização de sensores Inerciais como o *Inercial Measurement Unit* (IMU) está presente em diversos trabalhos relacionados à acessibilidade [Sato et al., 2014, Rohmer et al., 2015a]. O IMU é um sensor que permite reconhecer variações de posições, principalmente pela precisão adequada e custo baixo. Utilizar o acionamento com movimentos faciais atende deficientes que conseguem apenas movimento da face, ampliando a aplicabilidade do projeto. Ademais, o *kinect* reconhece o ambiente e, através de suas informações, realiza um controle compartilhado para ajustar a rota aos objetivos desejados pelo usuário.

O artigo [Röfer and Lankenau, 2000] apresenta uma plataforma de reabilitação que utiliza vinte e sete sensores do tipo sonar, buscando uma alternativa econômica para a proposta. Em [Salhi et al., 2016] apresenta-se uma abordagem de controle compartilhado entre o usuário e um cuidador que é capaz de ajustar o percurso remotamente através de um navegador ou *smartphone* com as imagens fornecidas por uma câmera *kinect* acoplada na cadeira, garantindo maior segurança ao indivíduo. Esta plataforma facilita o cuidado de indivíduos com deficiência que necessitam de acompanhamento constante. Vale salientar que para este caso, foi implementado o sistema utilizando a plataforma ROS (*Robotic Operating System*), que é largamente utilizada para aplicações em robótica.

Reconhecimento de padrões de estímulos faciais utilizando EEG também fazem parte das pesquisas recentes. Como exemplo pode-se citar o artigo [Soriano et al., 2013] que realiza um estudo com indivíduos sem deficiência e isquêmicos e compara suas respostas com estímulos musicais. Com uma taxa de sucesso de reconhecimento dos estímulos de 60%, é possível mostrar um caminho de desenvolvimento para interfaces entre o cérebro humano e os computadores. Desta forma, é possível notar o crescimento do uso do EEG

para tais aplicações e melhorias significativas nas taxas de reconhecimento que é uma questão fundamental para implementação de cadeiras de rodas baseadas em EEG ou ainda outras projetos com tal direcionamento.

Outros autores apresentam cadeiras automatizadas com opções de uso, dentre elas o controle manual, compartilhado e autônomo [Rohmer et al., 2015b, Sheikh and Rotake, 2015, Kuno et al., 1999]. O artigo [Rohmer et al., 2015b] apresenta primeiramente o controle completamente manual, o controle compartilhado no qual recebe a informação desejada pelo usuário e ajusta o destino objetivando o desvio de obstáculos e melhorar sua experiência e, por fim, tem-se uma alternativa completamente autônoma que apresenta resultados melhores que os métodos anteriores. No artigo [Sheikh and Rotake, 2015] foi utilizado também o comando de voz para controle compartilhado, como opção para deficientes com lesões mais severas. Adicionalmente, o artigo [Sheikh and Rotake, 2015] faz uso de dois sensores ultrassônicos para localização e movimentação completamente autônoma, no entanto, deve-se salientar que este modelo pode ser desconfortável para o usuário. Desta forma, o controle compartilhado é mostrado como uma alternativa satisfatória e que, além de permitir que o usuário conduza a cadeira, evita colisões da mesma.

O artigo [Kuno et al., 1999] apresenta três tipos de sistemas desenvolvidos. O primeiro tipo é o automático, onde tem-se dezesseis sensores ultrassônicos que evitarão qualquer obstáculo de forma autônoma e para guiar na direção correta o sistema usa reconhecimento facial, pois de acordo com o movimento do rosto, a cadeira irá se locomover na direção desejada. O segundo tipo é o sistema de controle manual com intenções mínimas, onde o usuário é responsável por cada movimento da cadeira, no entanto, para a movimentação é usado o reconhecimento do rosto através de uma câmera. Por fim, o terceiro tipo é de controle da cadeira de forma remota, pois o usuário pode ter dificuldade de se locomover até a cadeira de rodas quando não estiver nela, com isso, o sistema de reconhecimento pode ser usado pela pessoa externamente à cadeira a fim de guiá-la até o usuário. Com isso, este trabalho permite que o deficiente escolha a melhor forma de utilização da cadeira de rodas a fim de tornar a experiência mais agradável e acessível.

De forma semelhante a este trabalho, o artigo [Hu et al., 2010] utiliza uma câmera e o algoritmo Adaboost [Freund and Schapire, 1997] para reconhecimento dos lábios do usuário. Assim como alguns dos artigos anteriormente citados, este artigo utiliza um algoritmo para reconhecimento do padrão de movimentação dos lábios e usa essa informação para movimentação de uma cadeira de rodas, apresentando, portanto, um novo modelo similar ao objeto desta dissertação, porém, aplicada aos lábios do usuário.

Analogamente, o artigo [Lund et al., 2010] descreve um sistema de acionamento da cadeira automatizada utilizando movimentos da língua. Trata-se de um dispositivo intraoral capaz de perceber os movimentos da língua do indivíduo, enviar os dados para o

sistema e, através de inferência *fuzzy*, é capaz de determinar os movimentos e acionar a cadeira corretamente. Um modelo semelhante aos apresentados anteriormente, contudo utiliza a lógica *fuzzy* para reconhecimento do padrão de movimentação da língua do usuário.

O artigo [Kuo et al., 2006] propõe uma aplicação do uso de cadeira de rodas automatizada para ambientes internos. Primeiramente usa rodas omnidirecionais que movimentam a cadeira em qualquer direção. Ademais, utiliza dois modos de operação, sendo o primeiro completamente manual, em que sensores notificarão sobre qualquer obstáculo ao usuário que toma a decisão e o segundo propõe um controle compartilhado em que o *software* não permite colisões, ou seja, depende da ação humana para direcionar e ajusta automaticamente a trajetória para evitar colisões e aprimorar o caminho desejado.

O artigo [Raizer et al., 2013] apresenta uma interface cérebro computador (BCI) capaz de atender diversas necessidades de pessoas com deficiência através de estímulos cerebrais com relação as opções apresentadas em uma tela. Nesse trabalho, em uma cadeira de rodas automatizada foi acoplado um sistema de interface entre o usuário e um computador e possui diversas opções disponíveis, dentre elas pode-se citar a "Chamada por ajuda" que permite ao usuário pedir ajuda em caso de emergências; a "Comunicação" que permite enviar emails e SMS; a opção "Fazer" que execute alguma operação predefinida pelo sistema como "Assistir TV", "Tomar café" e "Almoçar"; a opção de "Ir" para uma determinada posição também está disponível, a qual permite que usuário escolha um destino em sua residência e a cadeira irá se locomover até ele, além de outras funções disponíveis na interface. O sistema desenvolvido mostra que é possível oferecer assistência à pessoas deficientes de forma a atender grande parte de suas necessidades, aumentando sua independência, ao menos, para tarefas cotidianas. Neste trabalho, mantém-se uma plataforma menos completa devido a intenção de reduzir o custo do projeto e tornar viável para os deficientes de baixa renda, fazendo com que o sistema atenda a um menor número de ações.

O artigo [Olivi et al., 2013] apresenta uma abordagem de controle compartilhado para cadeiras automatizadas, o qual permite que o usuário acione a cadeira utilizando o dispositivo *Emotiv EPOC's Expressiv* [Emotiv, ] que envia os sinais do usuário para o sistema através de uma interface. O objetivo do controle compartilhado desenvolvido no trabalho descrito é permitir a autonomia do usuário, utilizando uma inteligência de ajuste de rota baseada em campos vetoriais que adeque o caminho desejado pelo usuário a fim de que ele percorra o caminho de forma segura atendendo o objetivo esperado. Como destacado em outros trabalhos, o controle compartilhado é largamente utilizado neste tipo de aplicação e possui resultados expressivos em diversos trabalhos, permeando o campo de acessibilidade com alternativas eficazes.

O artigo [Ceres et al., 2005] aborda uma plataforma robótica para crianças com

deficiências severas. O módulo é capaz de se locomover de forma autônoma ou como controle compartilhado, obedecendo cinco níveis de autonomia escolhidos pelo usuário que permitem desde autonomia completa até o modelo em que a plataforma é guiada completamente pelo usuário com nenhuma ativação de sensores. Este desenvolvimento busca atender principalmente crianças com paralisia infantil formando uma plataforma de auxílio para sua reabilitação. Além disso, utiliza uma técnica online de detecção e desvio de obstáculos com sensores ultrassônicos escolhidos pela facilidade em termos de configuração e capacidade de processamento. Devido sua característica adaptável, o projeto apresentou um resultado expressivo com a melhoria da interação das crianças com o meio e em suas respostas cognitivas.

O artigo [SANDI et al., 1998] apresenta uma proposta de navegação e guiagem para robôs móveis utilizando um filtro de Kalman e lógica *fuzzy* para navegação. Foi possível utilizar a técnica e ainda proporcionar uma integração de baixo custo utilizando odômetros com *encoders* e bússola digital. O Filtro de Kalman é utilizado também neste trabalho e explicado na seção 3.3.1. A lógica *fuzzy* também é utilizada na dissertação [Nogueira, 2013] com o objetivo de se locomover desviando de obstáculos utilizando três sensores de distância localizados na direita, esquerda e frente do robô e dois motores de passo.

Com base nos trabalhos apresentados neste capítulo, traça-se a linha de trabalho desta dissertação que realiza a aquisição dos dados com um IMU [Sato et al., 2014, Rohmer et al., 2015a], o tratamento com um microcontrolador Arduino<sup>®</sup> [Arduino, ], utilizando uma filtragem inicial pelo Método dos Mínimos Quadrados (MMQ) [Microelectronics, ] e em seguida um Filtro de Kalman [Grewal, 2011] a fim de melhorar a resposta do sistema. Em seguida, utiliza-se o *software* MatLab<sup>®</sup> para simulação da resposta do sistema e também um robô móvel diferencial *Pioneer 3-DX* [MobileRobots, ].

Sabendo que a maioria dos indivíduos com deficiência recebem até um salário mínimo, torna-se fundamental reduzir custos no experimento, com isso, a utilização de um IMU deve-se a sua resposta satisfatória, com mais de 90% de acerto nos testes realizados e principalmente ao baixo custo que está em torno de R\$ 30,00. Outras opções para o sensoriamento facial, labial ou de movimentos da cabeça possuem, em geral, custo mais elevado, como por exemplo o Emotiv [Emotiv, ] utilizado no artigo [Olivi et al., 2013] que custa \$799,00 atualmente e da mesma forma, o EEG utilizado nos artigos [Röfer and Lankenau, 2000, Soriano et al., 2013, Li et al., 2016] possui alto custo comparado ao IMU. Com relação ao controle compartilhado apresentado nos artigos [Mitchell et al., 2014, Duan et al., 2014, Rohmer et al., 2015a, Olivi et al., 2013, Li et al., 2016, Tonin et al., 2010, Zhang et al., 2014, Ceres et al., 2005] optou-se por não utilizar as técnicas para controle compartilhado devido ao preço dos equipamentos de sensoriamento e também ao tempo disponível para implementação. Alternativas mais econômicas como sensores ultrassônicos foram levantadas, porém não foi possível medir sua eficácia no sistema. Com

relação a análise de dados foram testadas três metodologias de reconhecimento de padrões que foram redes neurais artificiais, distância euclidiana e distância de mahalanobis, em que este se mostrou mais eficiente para o caso, conforme apresentado no capítulo 4. Finalmente com relação a atuação do módulo dividiu-se em cinco regiões que foram utilizadas para separar os movimentos da cabeça do indivíduo e acionar a cadeira de rodas adequadamente. A atuação ocorreu em simulação no MatLab<sup>®</sup> e Gazebo<sup>®</sup> a fim de avaliar a atuação do módulo.

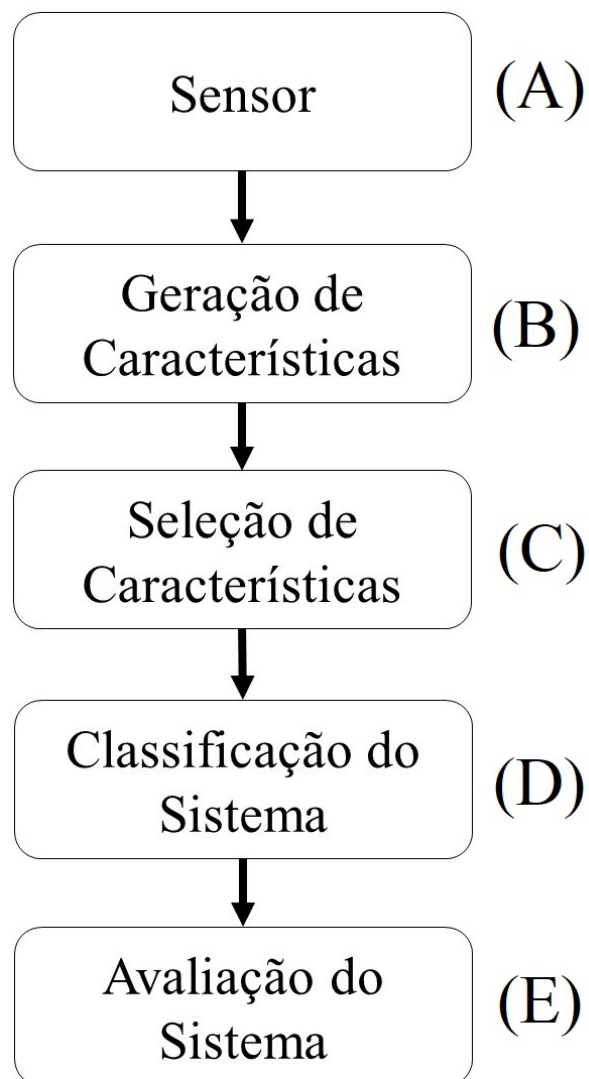
Em resumo, o capítulo apresentou diversos artigos relacionados à proposta desta dissertação, a fim de subsidiar o estudo realizado, apresentando alternativas assistivas para os públicos escolhidos, considerando precificação, tipo de deficiência e tecnologias utilizadas para cada abordagem.

### 3 FUNDAMENTAÇÃO TEÓRICA

#### 3.1 RECONHECIMENTO DE PADRÕES

Os dados recebidos pelo usuário devem ser classificados para determinar corretamente o movimento desejado. O processo de classificação envolve algumas etapas, exemplificadas pelo fluxograma da Figura 3.

Figura 3 – Fluxograma para reconhecimento dos padrões



Fonte: Autoria própria

Para a realização do processo de classificação de dados é necessário percorrer cada uma das etapas apontadas no fluxograma representado na Figura 3 a fim de obter uma avaliação completa do sistema. A seguir, apresenta-se a descrição dos passos para ilustrar



o processo de classificação.

- **Etapa A - Sensor:** Aquisição de dados. Realizada pelo IMU que utiliza acelerômetro, giroscópio e magnetômetro para correta localização da cabeça do usuário. Os dados são enviados para o arduino que realiza a conversão dos dados para *roll* e *pitch*;
- **Etapa B - Geração de Características:** Definição das características a serem trabalhadas para o modelo. No caso em estudo, as características geradas são os valores do IMU que são recebidos no Arduino<sup>®</sup> que são os valores do acelerômetro e giroscópio nos três eixos que serão melhor abordados na Seção 3.2;
- **Etapa C - Seleção de Características:** Refina as características que são relevantes para o sistema. Nos passos seguintes apresenta-se os passos para seleção das características do sistema desenvolvido para esta dissertação:
  1. Inicialmente realiza-se uma calibração do sistema utilizando o método dos Mínimos Quadrados conforme descrito no Apêndice 6 para a calibração inicial do sistema e adequação dos dados conforme recomendado no *datasheet* do IMU;
  2. Em seguida realiza-se a conversão dos dados de Acelerômetro e Giroscópio para *roll* e *pitch* utilizando as Equações 3.1

$$Roll = \frac{A_y}{A_z} \quad (3.1a)$$

$$Pitch = \frac{-A_x}{A_y \cdot \sin(roll) + A_z \cdot \cos(roll)} \quad (3.1b)$$

$A_x$ ,  $A_y$  e  $A_z$  são os valores medidos pelo acelerômetro corrigidos pela matriz de correção. Com os valores corrigidos é calculado *roll* e *pitch*.

3. Realiza-se filtragem utilizando o filtro de Kalman dos dados de *roll* e *pitch* a fim de ajustar os valores e melhorar a resposta do sistema. O filtro de Kalman será descrito na Seção 3.3.1.
- **Etapa D - Classificação do Sistema:** Classifica os dados de acordo com as regras atribuídas e o posicionamento do usuário. A divisão dos dados é feita através de três métodos que são a Distância Euclidiana, a Distância de Mahalanobis e Redes Neurais Artificiais. Os métodos realizam a divisão dos dados em cinco regiões de interesse de acordo com os valores de *roll* e *pitch* de cada posição, sendo que as classes definem as posições de movimento da cadeira de rodas (frente, trás, direita, esquerda e parado);

- **Etapa E - Avaliação do Sistema:** Avalia a resposta do sistema, verificando se é adequada e aplicando no sistema final. Os processos de aquisição, geração e classificação do sistema apresentaram resultados dentro do esperado e no caso da classificação, o método de Mahalanobis mostrou-se mais adequado para o caso apresentado de acordo com as informações apresentadas na Seção ??.

A seguir são apresentadas as metodologias de classificação de dados para esclarecer seu uso no presente trabalho. Os classificadores de distância mínima utilizados e comparados no desenvolvimento deste trabalho são a Distância Euclidiana e a Distância de Mahalanobis que serão descritas nos tópicos 3.1.1 e 3.1.2.

### 3.1.1 Distância Euclidiana

Trata-se de uma determinação de distância entre classes, em que os dados possuem distribuição gaussiana de erro, seguindo as seguintes premissas [Theodoridis et al., 2010]:

- As classes são equiprováveis;
- Os dados de todas as classes seguem distribuição gaussiana;
- A matriz de covariância é a mesma para todas as classe;
- A matriz de covariância é diagonal e possui todos os elementos da diagonal iguais.

Desta forma, a distância Euclidiana utiliza a Equação 3.2 para separar as classes em relação as suas médias aritméticas, dado um vetor no espaço  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  calcula-se a distância euclidiana com relação ao vetor da média  $\boldsymbol{\mu}$  da classe [of Standards and Technology, ].

$$D_E = \sum_{i=1}^n (x_i - \mu_i)^2 \quad (3.2)$$

### 3.1.2 Distância de Mahalanobis

É um classificador que a matriz de covariância não necessita ser diagonal e possuir todos os elementos da diagonal iguais, porém atende os demais requisitos do modelo de Distância Euclidiana. É uma técnica largamente utilizada para classificação de grupos e considera as correlações probabilísticas entre os conjuntos de dados, o que equacionalmente é determinada pela matriz de covariância  $\mathbf{S}$ . A Equação 3.3 descreve a distância de Mahalanobis.

$$D_M(x) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \quad (3.3)$$

Onde  $x$  é o ponto em análise,  $\mu$  é a média da classe e  $S$  a matriz de covariância.

A covariância denota a dispersão entre variáveis bidimensionais ( $X, Y$ ) em relação ao valor esperado ( $E(X), E(Y)$ ). Utilizada quando duas variáveis aleatórias não são independentes. O cálculo de covariância é feito como mostrado na Equação 3.4 [Liberal, ].

$$Cov(X, Y) = E[(X - E(X))(Y - E(Y))] \quad (3.4a)$$

$$Cov(X, Y) = \sum_x \sum_y (x - \mu_x)(y - \mu_y)p(x, y) \quad (3.4b)$$

Reescrevendo

$$Cov(X, Y) = E(XY) - E(X)E(Y) \quad (3.4c)$$

### 3.1.3 Redes Neurais Artificiais

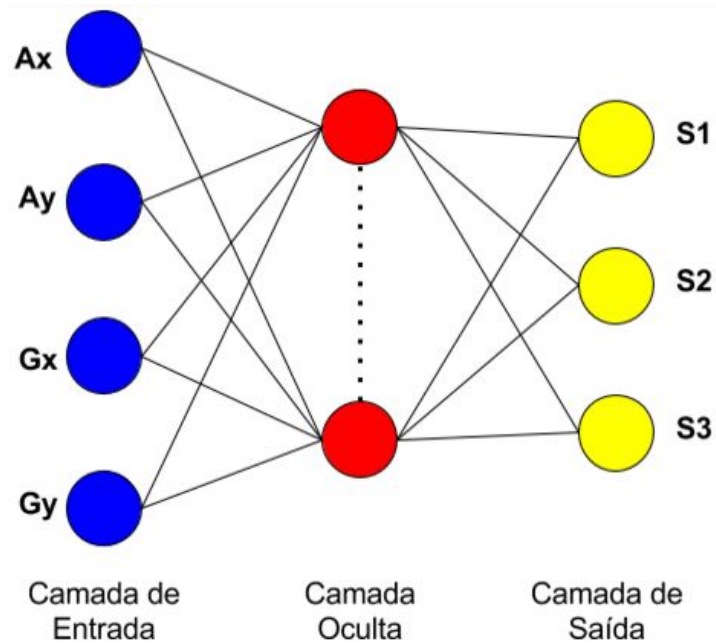
As Redes Neurais Artificiais (RNA) usam como base para funcionamento a estrutura neural de organismos inteligentes e adquirem conhecimento através da experiência. Baseiam-se nas redes neurais humanas que são compostas por neurônios que se comunicam através de sinapses, que é uma região onde dois neurônios entram em contato e os impulsos nervosos são transmitidos entre eles [CARVALHO, 2009].

De acordo com [Rauber, 2005] as redes neurais artificiais buscam quatro características especialmente atrativas:

- **Robustez e tolerância a falhas:** a funcionalidade geral não é afetada pela retirada de alguns neurônios;
- **Capacidade de aprendizagem:** assemelha-se a ideia do cérebro ao ser capaz de aprender novas tarefas;
- **Processamento de informação incerta:** é possível um resultado correto a partir de uma informações incertas;
- **Paralelismo:** vários neurônios ativos ao mesmo tempo.

Com isso, uma RNA busca compor uma configuração com neurônios divididos em três tipos de camadas: na camada de entrada os neurônios recebem as informações iniciais, na camada de saída o sistema envia a resposta final do problema proposto e nas camadas intermediárias (ocultas) tratam os dados realizando o *aprendizado* da rede. O conhecimento da rede está nas sinapses realizadas entre os neurônios, fazendo com que a RNA *aprenda* em cada nova iteração. Nesta dissertação foi utilizada a *toolbox* de RNA para simulações. A Figura 4 exemplifica uma rede neural artificial [Kovács, 2002].

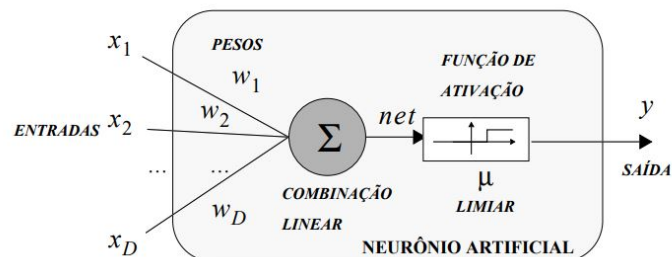
Figura 4 – Modelo RNA



Fonte: Autoria própria

A arquitetura e o algoritmo de aprendizagem são as questões mais importantes na estrutura de uma RNA. O treinamento ocorre por grupos de treino e o algoritmo de aprendizagem generaliza as informações e memoriza o conhecimento dentro dos parâmetros adaptáveis da rede, os pesos. A composição da rede é feita pelos neurônios. Em geral, o processamento de cada neurônio é a combinação linear das entradas com os pesos seguida pela passagem da combinação linear por uma função de ativação [Rauber, 2005]. A Figura 5 apresenta a composição da rede com os pesos e a função de ativação.

Figura 5 – Estrutura Neurônio Artificial



Fonte: [Rauber, 2005]

## 3.2 COMPONENTES DE HARDWARE

Nesta seção são descritos os componentes de hardware utilizados para a construção do módulo.

### 3.2.1 Orientação

Para adquirir a orientação da cabeça do usuário é utilizado um *Inertial Measurement Unit* (IMU) que é um dispositivo eletrônico que mede movimentos lineares e angulares. O equipamento é capaz de informar as especificações de *roll*, *pitch* e *yaw* que serão detalhadas na seção 3.4. IMU tem aplicabilidade em diversas áreas como, por exemplo, sensores vestíveis para esportes ou saúde, navegação de pedestres, robótica autônoma e realidade aumentada [xsens, , fairchildsemi, ].

O IMU possui uma unidade de medição inercial que proporciona detecção de aceleração de três eixos e taxa de rotação angular. A invenção utiliza vários laços de servo-controle convencionais para processar saídas a partir dos receptores e manter a massa do sensor na orientação adequada [Morrison, 1987].

Um das maiores desvantagens ao se usar IMUs para navegação é que realiza acumulação de erro porque o sistema continuamente adiciona mudanças de detecção para as posições previamente calculadas, ou seja, qualquer erro, mesmo pequeno, ao acumular pode provocar uma diferença de medição além da margem esperada. Além disso, devido ao dispositivo coletar dados em um espaço finito de tempo é possível que haja uma grande variação em um tempo específico e o dispositivo irá considerar apenas a média de todo o intervalo [Siciliano and Khatib, 2016].

Ademais, é necessário apresentar os eixos de análise para os movimentos de cabeça. Os eixos de rotação usualmente utilizados para referências de aeronaves e para aplicações próximas ao estudo apresentado são os eixos *Roll*, *Pitch* e *Yaw*. A Figura 1 apresenta esquematicamente os eixos de rotação.

O IMU está em um grupo de sensores chamado Sensores Inerciais que utilizam acelerômetros, magnetômetros e giroscópios para determinar o posicionamento de um corpo. Desta forma, é importante descrever o funcionamento destes equipamentos.

- **Acelerômetro:** Trata-se de transdutores que produzem um sinal proporcional a vibração ou aceleração do movimento na estrutura em que foi submetido. Possui uma grande variedade de aplicações, sendo mais comuns os piezoelétricos [Wilson, 2004]. Um material piezoelétrico produz um sinal de tensão proporcional à força mecânica aplicada sobre o mesmo. O princípio de funcionamento de um acelerômetro eletrônico é por meio de uma massa sobre um cristal piezoelétrico que exercerá uma

força sobre o cristal conforme a aceleração que o dispositivo sofre, seja por mudança inercial ou pela aceleração da gravidade [Maciel, 2016];

- Magnetômetro: Os magnetômetros medem a intensidade, direção e sentido de um campo magnético estático [Wilson, 2004]. Suas aplicações envolvem construção de bússolas digitais, por meio da captação do campo magnético terrestre, e medidores de intensidade de campo magnéticos externos.
- Giroscópio: Sensor que mede a velocidade angular. O princípio de funcionamento são duas massas pareadas oscilando em sentidos opostos com mesma velocidade. A partir da velocidade angular ocorre a movimentação das massas e consequente variação de capacitância interna. Então, a saída do sensor é proporcional à diferença entre as duas variações de capacitância.

### 3.2.2 Arduino®

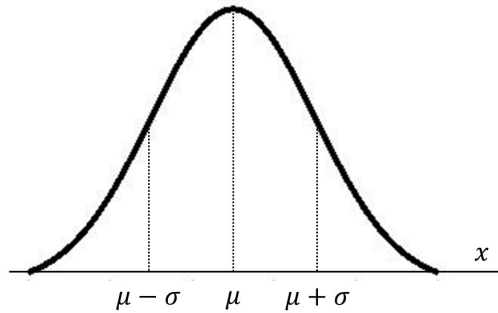
Arduino® é a plataforma de código aberto baseado em um *hardware* e um *software* de fácil uso. É capaz de ler entradas com sensores, entender dados de botões ou mensagens e acionar motores, LEDs e publicar informações na porta serial ou online. Logo, é possível enviar instruções para um microcontrolador com o objetivo de executar tarefas desejadas. Arduino® originou-se no Instituto *Ivrea Interaction Design* como uma forma fácil de prototipar e programar, com o intuito de atingir estudantes com pouco conhecimento de eletrônica e programação. A placa é completamente aberta, assim como o software [Arduino, ].

## 3.3 FILTRAGEM DE ERROS

Qualquer sistema real possui imperfeições e por consequência respostas com erros em suas medições. Contudo, há dois tipos de erros que são os erros determinísticos e não-determinísticos. Os erros determinísticos são os erros que possuem uma origem que pode ser determinada, com isso, é possível tratar e reduzir este tipo de erro, uma vez que são erros que podem ser calibrados, seja pela inclusão de um *offset*, compensação construtiva ou outra forma possível de ser realizada. No entanto, erros não-determinísticos são aqueles em que não é possível determinar sua origem facilmente ou, mesmo que determinados, não é possível tratar de uma forma direta como no primeiro. Então, são erros de origem aleatória e que uma forma de amenizá-los é por meio de filtragens usando processos estocásticos.

A curva Gaussiana é expressa pela função de densidade de probabilidade descrita na Equação 3.5 e sua distribuição é representada na Figura 6. Importante ressaltar que entre  $(\mu - \sigma)$  e  $(\mu + \sigma)$  encontra-se 68% das amostras. Desta forma, o Filtro de Kalman é

Figura 6 – Distribuição Gaussiana



Fonte: Autoria própria

uma opção para tratamento dos erros aleatórios notados neste trabalho, baseado numa curva de distribuição gaussiana que é capaz de modelar os dados obtidos pela IMU.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (3.5)$$

Onde  $\sigma$  é o desvio padrão da função de densidade de probabilidade e  $\mu$  é a esperança do modelo.

Para verificar se um conjunto de dados comporta-se como uma população normalmente distribuída tem-se diversos testes, dentre eles o Teste de Shapiro-Wilk [Shapiro and Wilk, 1965] que é um método estatístico obtido dividindo a raiz de uma combinação linear de amostras estatísticas ordenadas pela estimativa usual de variância. O teste é descrito na Equação 3.6.

$$W = \frac{(\sum_{i=1}^n a_i x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.6)$$

Onde  $x_i$  é a  $i$ -ésima amostra e  $\bar{x}$  é a média simples entre as amostras. Os valores de  $a_i$  pode ser calculado, porém comumente utiliza-se os valores tabelados do teste apresentado no Anexo A. A partir da tabela também é possível obter o valor de  $p$  (Tabela 17 - Anexo A) que define a proximidade dos dados com uma distribuição Normal. Quando tem-se  $p < 0,05$  considera-se que o conjunto não representa uma distribuição Normal.

### 3.3.1 Filtro de Kalman

Proposto em 1959 [Kalman et al., 1960], o filtro de Kalman apresenta uma nova modelagem para o processo de filtragem. Trata-se de um estimador linear capaz de tratar qualquer função quadrática de estimação de erro.

Para controlar um sistema dinâmico, é necessário que primeiramente conhecer o problema a fim de modelar para o correto desenvolvimento. O filtro de Kalman também é usado para prever a dinâmica futura do sistema [Grewal, 2011].

Utiliza-se uma computação recursiva com a finalidade de reduzir a variância do erro. A saída do filtro de Kalman depende do estado corrente e as entradas atuais para realizar a filtragem baseado em equações de sistemas lineares conforme mostrado na Equação 3.7 [Brookner, 1998].

$$x_k = f(x_{k-1}, u_k) + w_k \quad (3.7a)$$

$$z_k = Hx_k + v_k \quad (3.7b)$$

Onde  $x_k$ ,  $x_{k-1}$  representam, respectivamente, o estado atual e o anterior do sistema,  $z_k$  é a observação atual do sistema,  $v_k$  é a medida de ruído,  $u_k$  é a entrada de controle e  $H$  é a matriz de observação que relaciona as variáveis do filtro com as medições reais. No caso em estudo  $H = [1 \ 0]$ .

Objetiva-se obter  $x_k$  com base em  $z_k$ , já que não é possível medir  $x_k$  diretamente, pois os valores observados incluem distorções que são consideradas em  $z_k$ . Neste ponto, a implementação de Kalman pode estimar o estado atual com base na estimativa do anterior. Desta forma, a Equação 3.8 apresenta a estimativa do estado atual utilizando a metodologia do filtro de Kalman [Washington and University, ] iniciando o processo de predição.

$$\hat{x}_k = A_k x_{k-1} + B_k u_k \quad (3.8)$$

Onde  $u_k$  representa a entrada de controle e  $A$  e  $B$  são matrizes específicas da modelagem utilizado no sistema.

Ademais, é necessário formular a predição da covariância. Na Equação 3.9a estima-se a confiabilidade do sistema de predição através da matriz de covariância do erro a priori  $\hat{P}_k$ . Utiliza-se a matriz de covariância do ruído do processo  $Q$ , composta dos desvios padrão escolhidos para as variáveis do vetor  $x$ . Em 3.9b apresenta-se a definição da matriz  $Q$  para o caso de variáveis independentes. Com as Equações 3.9 define-se a etapa de predição do estado.

$$\hat{P}_k = A_k P_{k-1} A_k^T + Q \quad (3.9a)$$



$$Q = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n^2 \end{bmatrix} \quad (3.9b)$$

Para o problema desenvolvido nesta dissertação utiliza-se o filtro de Kalman para as obtenção de variáveis *roll* e *pitch* mais confiáveis para o sistema. O filtro é aplicado separadamente para as duas variáveis em que o *roll* utiliza a variação do giroscópio no eixo x e o *pitch* utiliza a variação do giroscópio no eixo y. O desvio padrão utilizado para *roll* e *pitch* foi 0,003 ( $Q_{angle} = 0,003$ ) e para o valor de giroscópio nos eixos x e y foi de 0,03 ( $Q_{gyro} = 0,03$ ).  $P_k$  foi denominado de  $XP$  no programa em Arduino<sup>®</sup>. A seguir tem-se a manipulação das equações (Equação 3.10) e respectivo programa em Arduino<sup>®</sup>. Ademais, a variação de tempo utilizada foi de 0,05 ( $DT = 0,05$ )

$$\begin{bmatrix} XP_{00} & XP_{01} \\ XP_{10} & XP_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0,05 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} XP_{00} & XP_{01} \\ XP_{10} & XP_{11} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -0,05 & 1 \end{bmatrix} + \begin{bmatrix} 0,003^2 & 0 \\ 0 & 0,03^2 \end{bmatrix} \quad (3.10)$$

---

**Algoritmo 1:** Ler número e imprimir se é par ou não.

---

**Entrada:** número, (*numero*).

**Saída:** Se o número é par ou não

```

1 início
2   ler numero;
3   se numero%2 = 0 então
4     | imprimir numero, "par";
5   senão
6     | imprimir numero, "impar";
7   fim se
8 fim
```

---



---

**Algoritmo 2:** Processo de Predição

---

```

1  $XP_{00+} = -DT * (XP_{10} + XP_{01}) + XP_{11} * DT * DT + Q_{angle} * Q_{angle};$ 
2  $XP_{01+} = -DT * XP_{11};$ 
3  $XP_{10+} = -DT * XP_{11};$ 
4  $XP_{11+} = +Q_{gyro} * Q_{gyro} * DT;$ 
```

---

Em seguida, deve-se prosseguir para o processo de atualização, onde realiza-se a compensação da medida real e calcula-se inicialmente o erro residual  $y$  (Equação 3.11), conhecido como valor de inovação.

$$y_k = z_k - H\hat{x}_k \quad (3.11)$$

Em Arduino<sup>®</sup> o código fica:

---

**Algoritmo 3:** Processo de Atualização

---

1  $y = accAngleX - KFangleX;$

---

A fim de calcular o ganho ótimo do filtro, utiliza-se a técnica dos mínimos quadrados para o erro entre o valor estimado e o real para obter o ganho de Kalman. As equações que definem esse processo são apresentados em 3.12.

$$S_k = H_k \hat{P}_k H_k^T + R_k \quad (3.12a)$$

$$K_k = \hat{P}_k H_k^T S_k^{-1} \quad (3.12b)$$

Onde  $S$  é a matriz do sistema, que calcula a confiabilidade da medição real  $z_k$  em relação ao modelo,  $R$  é a matriz de covariância do ruído dos sensores reais (equivalente da matriz  $Q$ ) e  $K$  é o ganho de Kalman [Kim, 2011].

Na aplicação em estudo:

$$S = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} XP_{00} & XP_{01} \\ XP_{10} & XP_{11} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R_{angle} \quad (3.13)$$

Implementação no Arduino<sup>®</sup>:

---

**Algoritmo 4:** Ganho de Kalman

---

1  $S = XP_{00} + R_{angle};$

2  $K_0 = \frac{XP_{00}}{S};$

3  $K_1 = \frac{XP_{10}}{S};$

---

Por fim, funde-se os dois modelos, o matemático da predição e as medidas reais. Através das Equações 3.14 tem-se o cálculo do valor final filtrado.

$$x_k = \hat{x}_k + K_k y \quad (3.14a)$$

$$P_k = (I - K_k H) \hat{P}_k \quad (3.14b)$$

Onde  $I$  é a matriz identidade de mesma dimensão que  $P$ .

$$x_k = \hat{x}_k + K_k y \quad (3.15a)$$

$$\begin{bmatrix} XP_{00} & XP_{01} \\ XP_{10} & XP_{11} \end{bmatrix} = \begin{bmatrix} 1 - K_0 & 0 \\ -K_1 & 1 \end{bmatrix} \begin{bmatrix} XP_{00} & XP_{01} \\ XP_{10} & XP_{11} \end{bmatrix} \quad (3.15b)$$

Implementando em Arduino<sup>®</sup>:

---

**Algoritmo 5:** Fusão de modelos matemático e real

---

- 1  $KFangleX+ = K_0 * y;$
  - 2  $x_{bias+} = K_1 * y;$
  - 3  $XP_{00-} = K_0 * XP_{00};$
  - 4  $XP_{01+} = K_0 * XP_{01};$
  - 5  $XP_{10+} = K_1 * XP_{00};$
  - 6  $XP_{11+} = K_1 * XP_{01};$
- 

Para o projeto proposto, tem-se a aplicação de um filtro de Kalman em uma IMU, que é eficiente em uma unidade inercial, sendo capaz de produzir dados confiáveis para o sistema *Roll*, *Pitch* e *Yaw* da convenção ZYX. É possível obter os dados dentro deste sistema com a fusão das medidas do acelerômetro, giroscópio e magnetômetro.

A Equação 3.16 modela um sistema rotativo com ângulo  $\theta$  e velocidade angular  $\dot{\theta}$ .

$$\theta = \theta_{k-1} + \dot{\theta}\Delta t \quad (3.16)$$

Devido à rotação da Terra o giroscópio real apresenta um desvio de seu *offset*, dificultando no cálculo da velocidade angular *bias*, uma vez que varia com o tempo [Tajmar et al., 2009, Lauszus, 2012].

A implementação no Arduino<sup>®</sup> foi realizada da seguinte forma:

---

**Algoritmo 6:** Desvio de *Offset*

---

- 1  $KFangleX+ = DT * (gyroRate - x_{bias});$
- 

Vale salientar que todo o processo de filtragem através do filtro de Kalman é repetida para *pitch* com a variável do giroscópio no eixo y.

### 3.4 ROBÓTICA MÓVEL

Robôs móveis têm sido largamente utilizados na indústria, em aplicações residenciais, hospitais a fim de facilitar atividades. Tais robôs são compostos basicamente de um controlador, um *software* para controle, sensores e atuadores a fim de tornar as aplicações possíveis. Para o trabalho apresentado utiliza-se o robô móvel *Pioneer 3-DX* [MobileRobots, ].

O *Pioneer 3-DX* é um robô diferencial ideal para laboratórios internos ou uso em sala de aula. É equipado com um sonar, uma bateria, rodas com *encoders*, um microcontrolador e um software de desenvolvimento. Largamente utilizado e abrange diversas aplicações [MobileRobots, ] .

Para o projeto desenvolvimento o *Pioneer 3-DX* é utilizado como forma de simulação de uma cadeira de rodas para fins de pesquisa e análise. Em um segundo passo, pretende-se implementar para uma cadeira de rodas automatizada e adaptada para a proposta apresentada que busca alternativas econômicas e de fácil embarcação para implementação real.

No trabalho apresentado utiliza-se movimentação apenas no plano, desta forma, não há necessidade de avaliação de um modelo tridimensional. Inicialmente é relevante apresentar o equacionamento do movimento de translação de um robô móvel (Equação 3.17).

$$\begin{bmatrix} x_{(1)} \\ y_{(1)} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{(0)} \\ y_{(0)} \\ 1 \end{bmatrix} \quad (3.17)$$

Onde  $\Delta x$  e  $\Delta y$  são as variações em  $x$  e  $y$  respectivamente do movimento de translação,  $x_{(1)}$  e  $y_{(1)}$  são as coordenadas após o movimento e  $x_{(0)}$  e  $y_{(0)}$  são as coordenadas do sistema original.

Ademais tem-se o equacionamento para o movimento de rotação em um sistema bidimensional é expresso na Equação 3.18.

$$\begin{bmatrix} x_{(1)} \\ y_{(1)} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_{(0)} \\ y_{(0)} \end{bmatrix} \quad (3.18)$$

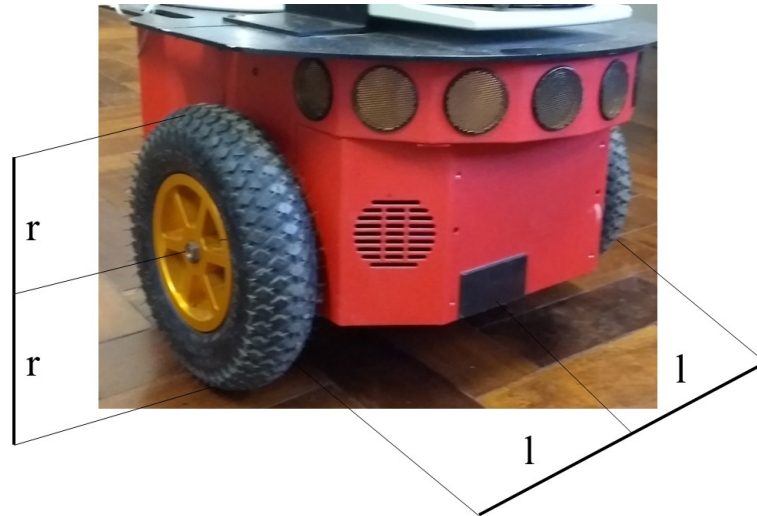
A fim de tratar a dinâmica do robô consideraremos  $l$  a distância entre suas rodas e  $r$  o raio de sua roda, assim como mostrado na Figura 7.

Considerando  $\theta$  a rotação do robô,  $\dot{\varphi}_e$  a rotação angular da roda esquerda e  $\dot{\varphi}_d$  a rotação angular da roda direita pode-se calcular a velocidade linear  $v$  e a velocidade angular  $\omega$ , assim como apresentado nas Equações 3.19a e 3.19b.

$$v_e = r\dot{\varphi}_e \quad (3.19a)$$

$$v_d = r\dot{\varphi}_d \quad (3.19b)$$

Figura 7 – Medidas Pioneer 3-DX



Fonte: Autoria própria

Como  $w = \frac{v}{l}$ , com  $l$  sendo a metade da largura do robô, conforme definido anteriormente. Com isso, a velocidade angular pode ser definida como apresentado na Equação 3.20.

$$w = \frac{r\dot{\varphi}_r - r\dot{\varphi}_l}{2} \rightarrow w = \frac{r}{2l}(\dot{\varphi}_r - \dot{\varphi}_l) \quad (3.20)$$

A partir disso é possível obter o modelo cinemático para o robô diferencial que está representado na Equação 3.21.

$$\begin{bmatrix} \dot{X}_1 \\ \dot{Y}_1 \\ \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (3.21)$$

Ademais é possível inferir o modelo incremental em que a nova pose do robô é computada a cada variação de tempo em função das velocidades das rodas (3.22).

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \begin{bmatrix} \Delta s \cdot \cos(\theta_k + \frac{\Delta\theta}{2}) \\ \Delta s \cdot \sin(\theta_k + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix} \quad (3.22)$$

Com esta fundamentação teórica cria-se embasamento suficiente para o prosseguimento do trabalho. Conforme explicitado, a distância de Mahalanobis será utilizada na implementação do método, o que será exposto no capítulo 4 com detalhamento e comparações, principalmente com relação ao método de Redes Neurais Artificiais a fim de comprovar sua correta aplicação para o método. Além disso, foi apresentado neste

capítulo os componentes do IMU a fim de orientar o estudo com relação ao equipamento, a orientação no domínio *Roll*, *Pitch* e *Yaw* e também o processo de filtragem buscando melhorar a resposta do sistema. Por fim, a modelagem do robô móvel busca orientar para a implementação do modelo no robô móvel *Pioneer 3DX*.

## 4 METODOLOGIA

O presente capítulo apresenta a metodologia proposta para o projeto. São descritos os materiais utilizados, a plataforma e o processo de desenvolvimento.

A cadeira de rodas descrita propõe uma alternativa para deficientes com restrição severa de mobilidade que não são capazes de movimentar os braços, no entanto, possuem mobilidade do pescoço. A cadeira de rodas recebe os comandos de movimentação da cabeça do indivíduo através de uma IMU em um capacete que traduz os movimentos do usuário descrito na seção 4.3 e envia como comando para módulo ou simulador através de um Arduino<sup>®</sup>. O Arduino<sup>®</sup> realiza o tratamento e a filtragem dos dados do IMU e envia para o simulador. Em seguida, ocorre o acionamento da cadeira de rodas de acordo com a movimentação do usuário. No decorrer deste trabalho apresenta-se a plataforma simulada em MatLab<sup>®</sup> e em um robô móvel *Pioneer 3DX* descrito na seção 3.4.

Tem-se razoável precisão nos dados do sensor IMU, no entanto, é necessário realizar ainda alguns ajustes nos dados recebidos para adequá-los ao sistema proposto. Desta forma, utiliza-se o Método dos Mínimos Quadrados para realizar a calibragem inicial dos dados do IMU e convertê-los para o eixo de trabalho, que neste caso, trata-se de *roll*, *pitch* e *yaw* [Microelectronics, ]. A próxima etapa é o tratamento dos dados pelo Arduino<sup>®</sup>, que realiza a implementação do Filtro de Kalman [Kalman et al., 1960] e envio dos dados para o software MatLab<sup>®</sup> que realiza a simulação da cadeira de rodas ou diretamente para o sistema ROS (*Robotic Operating System*) que realizará o tratamento e acionamento do robô diferencial *Pioneer 3-DX* [MobileRobots, ].

### 4.1 MATERIAIS

Esta seção tem por objetivo especificar os materiais utilizados e suas respectivas funções. A maioria já foram descritos no Capítulo 3, desta forma, apresenta-se uma descrição e especificação a fim de ilustrar de forma mais clara o uso de cada componente e sua finalidade para o desenvolvimento proposto.

Primeiramente analisa-se os equipamentos e softwares utilizados para a simulação do projeto.

- **IMU GY-87:** responsável pela aquisição de dados gerado pela movimentação da cabeça do usuário. Foi fixado em um fone de ouvido modelo *headset* e permite o entendimento dos dados enviados pela movimentação do usuário e envio para o Arduino<sup>®</sup>;
- **Arduino<sup>®</sup>:** realiza a recepção de dados provenientes do IMU, a filtragem utilizando Filtro de Kalman e posterior envio para o software MatLab<sup>®</sup>;

- **Software MatLab<sup>®</sup>**: realiza a criação do ambiente simulado e retorno visual para o usuário dos movimentos realizados pela cabeça.

No entanto, para a realização de um teste prático utiliza-se um robô móvel:

- **P3DX**: robô terrestre apresentado detalhamento na seção 3.4;

## 4.2 PLATAFORMA ROS

ROS (*Robot Operating System*) é um *framework* para utilização em sistemas robóticos e integra algoritmos, *hardwares* e robôs. Tem-se uma coleção de ferramentas e bibliotecas que visam simplificar e simular plataformas robóticas. Desta forma, o ROS foi criado para incentivar o desenvolvimento colaborativo da robótica, permitindo que pesquisadores em torno do mundo possam contribuir para o desenvolvimento da robótica de forma direta e efetiva [ROS, ]. Vale salientar que, neste projeto, utiliza-se o ROS com a intenção de interpretar os movimentos e acionar o robô móvel que será especificado na seção 3.4.

Neste trabalho a plataforma ROS é utilizada para envio dos dados pelo Arduino<sup>®</sup> e posteriormente tratamento no robô diferencial *Pioneer 3-DX*.

## 4.3 AQUISIÇÃO E TRATAMENTO DE DADOS

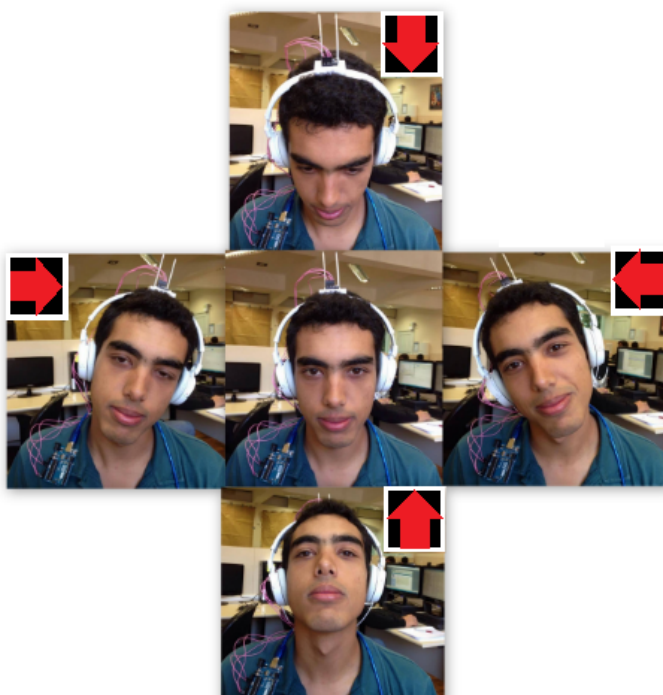
Esta seção tem-se por objetivo descrever o processo de aquisição e tratamento dos dados. Desta forma, utiliza-se a plataforma ROS inicialmente em comunicação com o Arduino<sup>®</sup> recebendo os dados provenientes desta plataforma.

Deseja-se distribuir e classificar os dados dentro de cinco classes conforme descrito na Figura 8. Além disso, define-se as posições com as seguintes intenções do usuário:

- **Neutro**: a posição neutra significa que o usuário deseja que não haja movimento na cadeira de rodas;
- **Atrás**: deseja-se que a cadeira movimente-se lentamente para trás;
- **Frente**: deseja-se que a cadeira movimente-se para frente;
- **Direita**: deseja-se que a cadeira movimente apenas a roda esquerda, fazendo uma curva para a direita;
- **Esquerda**: deseja-se que a cadeira movimente apenas a roda direita, fazendo uma curva para a esquerda.



Figura 8 – Classes de Atuação



Fonte: Autoria própria

Para o tratamento dos dados recebidos utiliza-se um filtro de Kalman incluído no programa do Arduino<sup>®</sup>.

O desenvolvimento do programa em Arduino<sup>®</sup> é fundamental para recepção e tratamento dos dados do IMU que realiza o reconhecimento do movimento e envia para o Arduino<sup>®</sup>. Os dados recebidos passam por um filtro de Kalman a fim de retirar parte dos erros aleatórios; em seguida são separados e enviados para o MatLab<sup>®</sup> seguindo um padrão formulado.

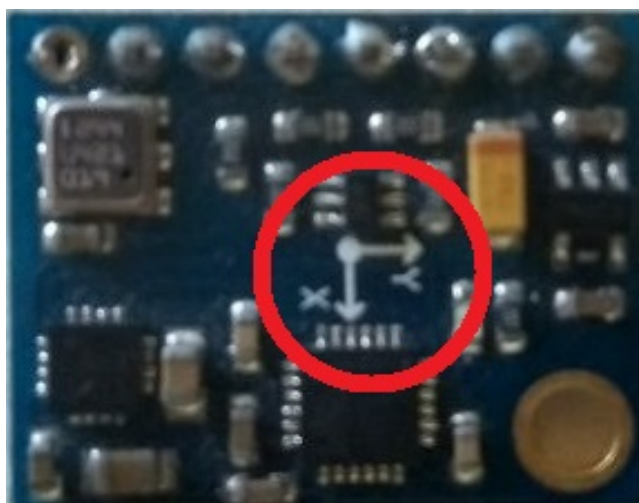
#### 4.4 ROTINA DE CALIBRAÇÃO

Primeiramente, para obter uma resposta mais adequada ao sistema é necessário calibrar as medições realizadas pelo acelerômetro [Microelectronics, ]. Além disso, um segundo ponto a ser destacado é que cada indivíduo realizará a movimentação da cabeça de uma forma diferente. Assim, torna-se necessário uma calibração que será aplicada para cada usuário a fim de definir uma melhor margem de atuação dos equipamentos e melhorar a experiência do usuário.

##### 4.4.1 Calibração do Acelerômetro

Utiliza-se o software MatLab<sup>®</sup> para a calibração inicial. A calibragem é realizada para os parâmetros *roll* e *pitch*. Para algumas aplicações, como por exemplo, rotação

Figura 9 – IMU



Fonte: Autoria própria

da tela de *smartphones* e abertura e fechamento de *notebooks*, porém, nesta aplicação é necessário ajustar os dados a fim de desenvolver um produto final menos susceptível a erros de medição e classificação.

A calibração é realizada com os posicionamentos base do dispositivo, que são indicados como na Figura 9. A calibração recomendada é feita pelo método dos mínimos quadrados [Microelectronics, ].

Para ajustar os valores faz-se um banco de dados seguindo os parâmetros estabelecidos como apresentados na Tabela 1. Na Tabela tem-se os valores sem tratamento do acelerômetro e relacionados à gravidade da Terra ( $g$ ). Utilizando as posições especificadas pode-se montar um banco de dados com as posições do IMU relativo ao posicionamento apresentado na Tabela 1, logo, coloca-se o IMU nas posições indicadas a fim de montar o banco de dados e calcular a matriz de correção que realizará o ajuste dos valores adquiridos pelo equipamento em sua utilização. O banco de dados adquirido é composto por 6400 medições realizadas em uma mesa a fim de posicionar o equipamento de forma acurada com relação aos eixos.

#### 4.4.2 Implementação do Método dos Mínimos Quadrados

Para os dados obtidos do IMU é necessário a criação de uma matriz de correção que ajusta os dados do IMU fornecendo o valor adequado após a operação com a matriz. Utilizando os dados do acelerômetro obtem-se uma matriz  $4 \times 3$  de acordo com a Equação

Tabela 1 – Parametrização do Acelerômetro

Posição Estacionária	Acelerômetro		
	$A_x$	$A_y$	$A_z$
$Z_b$ baixo	0	0	+1 $g$
$Z_b$ cima	0	0	-1 $g$
$Y_b$ baixo	0	+1 $g$	0
$Y_b$ cima	0	-1 $g$	0
$X_b$ baixo	+1 $g$	0	0
$X_b$ cima	-1 $g$	0	0

Fonte: AN4508 Application note [Microelectronics, ]

4.1 [Microelectronics, ].

$$\begin{bmatrix} A_{x1} & A_{y1} & A_{z1} \end{bmatrix} = \begin{bmatrix} A_x & A_y & A_z & 1 \end{bmatrix} \cdot \begin{bmatrix} ACC_{11} & ACC_{21} & ACC_{31} \\ ACC_{12} & ACC_{22} & ACC_{32} \\ ACC_{13} & ACC_{23} & ACC_{33} \\ ACC_{10} & ACC_{20} & ACC_{30} \end{bmatrix} \quad (4.1)$$

ou

$$Y = w \cdot X \quad (4.2)$$

Comparativamente:

$$Y = \begin{bmatrix} A_{x1} & A_{y1} & A_{z1} \end{bmatrix} \quad (4.3a)$$

$$w = \begin{bmatrix} A_x & A_y & A_z & 1 \end{bmatrix} \quad (4.3b)$$

$$X = \begin{bmatrix} ACC_{11} & ACC_{21} & ACC_{31} \\ ACC_{12} & ACC_{22} & ACC_{32} \\ ACC_{13} & ACC_{23} & ACC_{33} \\ ACC_{10} & ACC_{20} & ACC_{30} \end{bmatrix} \quad (4.3c)$$

Desta forma,  $w$  contém os dados adquiridos pelo IMU em seu modo sem qualquer tratamento nas posições estacionárias especificados na Tabela 1. A matriz  $X$  contém os valores correspondentes apresentados também na Tabela 1. Como exemplo deste tratamento, com o IMU posicionado com  $Z_b$  para baixo, o que corresponde a  $A_{x1} = 0$ ,  $A_{y1} = 0$  e  $A_{z1} = +1$ . Os dados sem tratamento dessa posição em um teste inicial foram  $A_x = 1472$ ,  $A_y = 708$  e  $A_z = 18340$ . O objetivo neste ponto é montar a matriz de correção

do sistema que é obtida relacionando os valores esperados de cada posição e os reais sem tratamento utilizando a Equação 4.4 obtida pelo Método dos Mínimos Quadrados.

$$X = [w^T \cdot w]^{-1} \cdot w^T \cdot Y \quad (4.4)$$

De acordo com [Ventura, 2013] a demonstração da obtenção dessa equação é dada conforme equacionamento em 4.5 a 4.8.

$$\|wX\|^2 = \|w(X - X_*) + wX_*\|^2 \quad (4.5a)$$

Onde  $X_*$  representa a inclusão de uma variável mantendo a igualdade no equacionamento. Representa um valor medido de  $X$  com menor precisão conforme descrito no Método dos Mínimos Quadrados detalhado no Apêndice B.

$$\|wX\|^2 = [w(X - X_*) + (wX_*)]^T [w(X - X_*) + (wX_*)] \quad (4.5b)$$

$$\|wX\|^2 = \|w(X - X_*)\|^2 + \|wX_*\|^2 + 2[w(X - X_*)]^T (wX_*) \quad (4.5c)$$

O valor mínimo da equação ocorre quando  $[w(X - X_*)]^T (wX_*) = 0$

$$[w(X - X_*)]^T (wX_*) = (X - X_*)^T w^T [(w(w^T w)^{-1})w^T - I_n] \quad (4.6a)$$

$$[w(X - X_*)]^T (wX_*) = (X - X_*)^T [(w^T w)(w^T w)^{-1}w^T - w^T] \quad (4.6b)$$

$$[w(X - X_*)]^T (wX_*) = (X - X_*)^T (w^T - w^T) \quad (4.6c)$$

$$[w(X - X_*)]^T (wX_*) = 0. \quad (4.6d)$$

Com isso,

$$\|wX\|^2 = \|w(X - X_*)\|^2 + \|wX_*\|^2 \quad (4.7)$$

Se  $X \neq X_*$ , então  $\|w(X - X_*)\| > 0$ . Assim, para todo  $X \neq X_*$ ,

$$\|wX\|^2 > \|wX_*\|^2 \quad (4.8)$$

Logo,  $X = (w^T w)^{-1} w^T$  é o vetor que minimiza  $\|wX\|^2$

Com isso, a matriz de correção é obtida a partir de  $w$  que são os dados sem tratamento do acelerômetro e  $Y$  que é a posição esperada conforme apresentado na Tabela

1. A partir das duas informações é possível posicionar o IMU nas posições padronizadas e obter a matriz de correção adequada para o equipamento em estudo. A matriz de correção obtida com um conjunto amostral de 6000 eventos é apresentada na Equação 4.9.

$$X = \begin{bmatrix} 0,00006147844 & 0,0000606373 & 0,00006056773 \\ 0,0000004262922 & -0,000000407563 & -0,0000009520133 \\ -0,0000002167082 & 0,0000005529877 & -0,0000007607212 \\ -0,04212447 & -0,002551615 & 0,1074834 \end{bmatrix} \quad (4.9)$$

Com a matriz de correção obtida é possível ajustar os valores obtidos diretamente do IMU a fim de melhorar a resposta do sistema. Desta forma, para o exemplo de dados sem tratamento com  $A_x = 1472$ ,  $A_y = 708$  e  $A_z = 18340$  é possível aplicar a matriz de correção apresentada em 4.9 conforme apresentado em 4.10.

$$Y = \begin{bmatrix} 1472 & 708 & 18340 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0,00006147844 & 0,0000606373 & 0,00006056773 \\ 0,0000004262922 & -0,000000407563 & -0,0000009520133 \\ -0,0000002167082 & 0,0000005529877 & -0,0000007607212 \\ -0,04212447 & -0,002551615 & 0,1074834 \end{bmatrix} \quad (4.10a)$$

$$Y = \begin{bmatrix} 0,0447 & 0,0499 & 1,2165 \end{bmatrix} \quad (4.10b)$$

No entanto, os valores do acelerômetro corrigidos ainda não são suficientes para uma análise eficaz do sistema. Desta forma, deve-se realizar a conversão dos valores do acelerômetro para *roll* e *pitch*. Trabalha-se, neste caso, com as rotações na ordem de eixos  $x$ ,  $y$  e  $z$ , com os ângulos  $\phi$  em *roll*,  $\theta$  em *pitch* e  $\psi$  em *yaw*. As matrizes de rotação em cada eixo são apresentadas na Equação 4.11

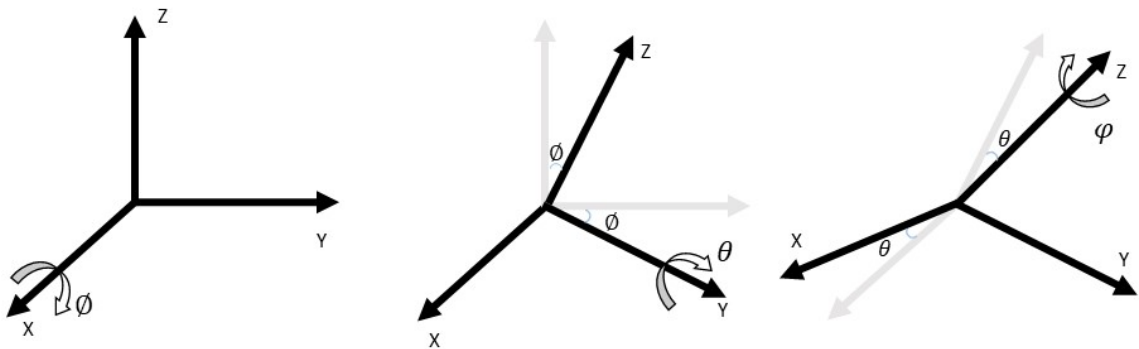
$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (4.11a)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4.11b)$$

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11c)$$

As Figuras 10 e 11 apresentam visualmente o processo de rotação apresentado no equacionamento 4.11.

Figura 10 – Rotações nos eixos

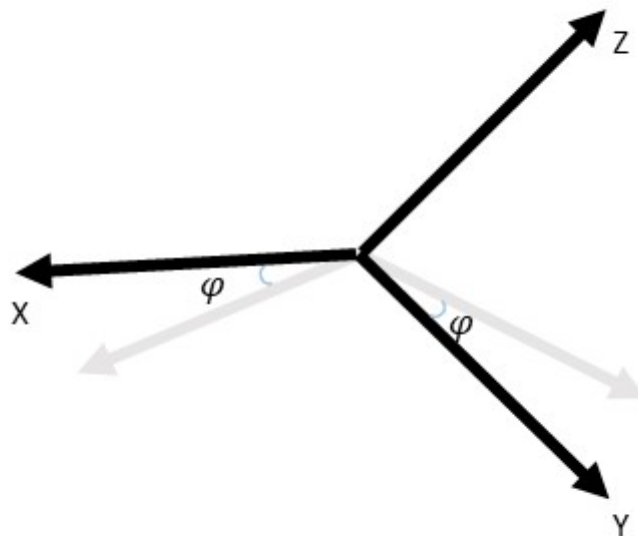


Fonte: Autoria própria

Para realizar a conversão para este domínio utiliza-se a Equação 4.12.

$$Roll = \frac{A_y}{A_z} \quad (4.12a)$$

Figura 11 – Posição final após as rotações



Fonte: Autoria própria

$$Pitch = \frac{-A_x}{A_y \cdot \sin(roll) + A_z \cdot \cos(roll)} \quad (4.12b)$$

Para o caso de exemplificação descrito nesta seção pode-se calcular os valores de *Roll* e *Pitch* com base na Equação 4.12 e atribuindo o valor de *Y* obtido em 4.10 para o caso específico em análise. Os valores calculados são descritos em 4.13.

$$Roll = \frac{0,0499}{1,2165} = 0,0410 \quad (4.13a)$$

$$Pitch = \frac{-0,0447}{0,0499 \cdot \sin(0,0410) + 1,2165 \cdot \cos(0,0410)} = -0,0449 \quad (4.13b)$$

#### 4.4.3 Calibração Individual

Cada indivíduo possui uma referência diferente tratando-se de sua posição confortável da cabeça, desta forma, a posição referencial, ou seja, a posição em que a cadeira de rodas deve permanecer parada precisa ser calibrada individualmente para cada novo indivíduo que utilizar o equipamento. Para atender essa necessidade cria-se no Arduino<sup>®</sup> uma função (apresentada a seguir) que realiza a calibração quando o equipamento é ligado para que todas as vezes em que o capacete seja ligado realiza uma nova calibração do ponto neutro. Destaca-se neste ponto que o usuário deve estar com o equipamento na posição neutra no momento de energização, porém caso não seja possível, após a inicialização pode-se resetar o equipamento e a calibração será realizada novamente.

---

#### Algoritmo 7: Calibração Inicial do Equipamento

---

```

1 void calibrate()
2 float somar = 0;
3 float somap = 0;
4 for(int k = 0; k < 200; k++)
5 request();
6 getangles();
7 somar += roll;
8 somap += pitch;
9 inertialroll = somar/200;
10 inertialpitch = somap/200;

```

---

A função desenvolvida para calibração individual é simples e, em linhas gerais, realiza a requisição e obtenção dos dados do acelerômetro, calibração geral como mostrado na seção 4.4.2 e, em seguida, realiza a média dos valores obtidos e cria o padrão de referência para o restante do programa.

Nesse capítulo foi apresentada inicialmente a metodologia utilizada na dissertação e os materiais utilizados para o teste prático. Além disso, descreve-se a plataforma ROS e

sua importância no trabalho, o formato de aquisição e tratamento de dados com a devida descrição das classes utilizadas no trabalho. Em seguida, foi apresentada a rotina de calibração e a forma de obtenção de valores de *roll* e *pitch* a partir dos dados enviados pelo IMU. Por fim, foi apresentada a calibração individual necessária para cada indivíduo.



## 5 IMPLEMENTAÇÃO E RESULTADOS

As informações do Arduino podem ser recebidos pelo MatLab<sup>®</sup> via serial ou ROS, desta forma, tem-se dois programas em Arduino<sup>®</sup> e MatLab<sup>®</sup> para cada uma das formas de envio dos dados de posicionamento do IMU. Os programas serão detalhados na seção A.4 (Apêndice A). As duas formas de envio de dados são válidas, no entanto, se adequam melhor em casos distintos, por exemplo, uma simulação completa em MatLab<sup>®</sup> indica-se o uso de envio através de serial, pois não envolve um novo sistema operacional e proporciona uma recepção mais rápida no MatLab<sup>®</sup>, no entanto, para simulações em Gazebo<sup>®</sup> o envio de dados através de ROS é mais adequado, pois facilita o uso de ambientes de simulação que utilizam a plataforma ROS. Com isso, o programa de simulação no MatLab<sup>®</sup> utilizou-se o envio via serial e para a simulação em Gazebo<sup>®</sup> utilizou-se o envio de dados via ROS.

Os dados foram desenvolvidos com um conjunto de cerca 75.000 eventos que foram adquiridos com diferentes usuários, atingindo centenas de posicionamentos pré-determinados. A Figura 12 apresenta a projeção de 1000 eventos selecionados randomicamente no plano do acelerômetro e giroscópio, mostrando claramente as regiões sobrepostas de cada classe.

A fim de classificar os dados e identificar o desejo do usuário é necessário escolher métodos de classificação a serem utilizados na plataforma. Os métodos de classificação utilizados neste projeto foram apresentados na seção 3.1 que são a distância euclidiana, distância de mahalanobis e utilizando redes neurais artificiais. Os erros comparando o método Euclidiano e Mahalanobis são apresentados na Tabela 2, mostrando que o método de Mahalanobis é superior com relação a acurácia do método, no entanto, o método Euclidiano mostra-se superior com relação ao tempo, pois utiliza um tempo aproximadamente 40% menor que o tempo utilizado para classificação pelo método de Mahalanobis.

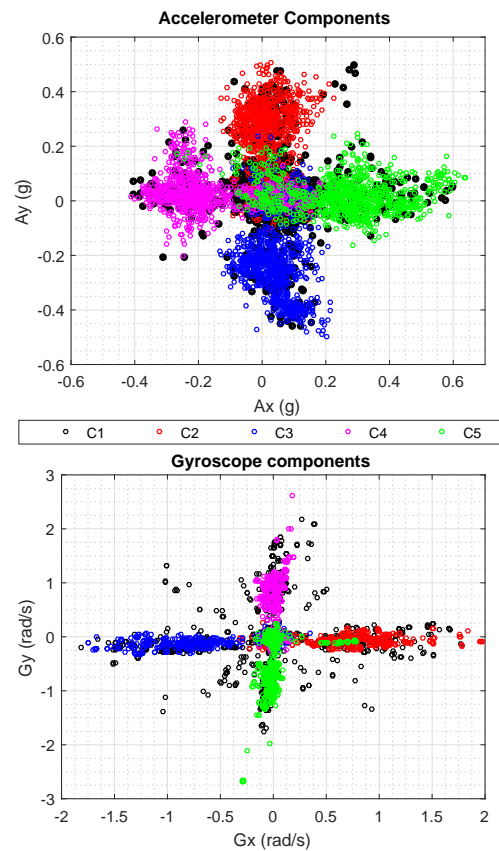
Tabela 2 – Análise dos Métodos de Distância

Método	Taxa de Erro Média	Tempo Médio
Euclidiano	11.53%	3.9 $\mu$ s
Mahalanobis	8.87%	6.4 $\mu$ s

Fonte: [Marins et al., 2017]

Na rede artificial implementada são utilizados quinze neurônios na camada oculta. A comparação entre o método Euclidiano, Mahalanobis e RNA que estão apresentados neste trabalho no Apêndice A resultam na margem de erro apresentada na Tabela 3.

Figura 12 – Distribuição de Dados



Fonte: [Marins et al., 2017]

Tabela 3 – Comparação entre os métodos

Método	Taxa de Erro Média
Euclidiano	11,40%
Mahalanobis	10,23%
RNA	12,05%

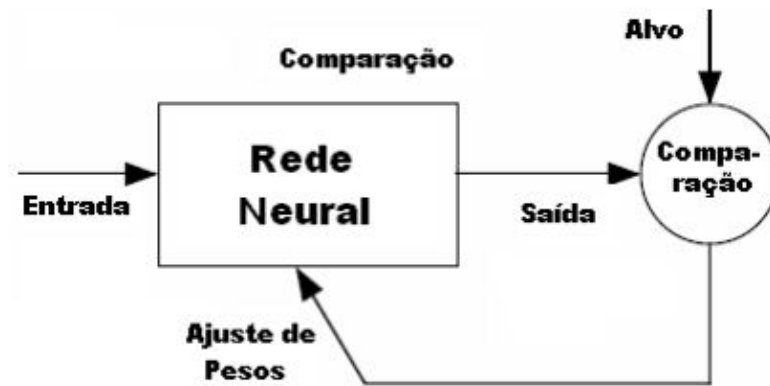
Fonte: Autoria própria

## 5.1 IMPLEMENTAÇÃO DE RNA

A construção de uma rede neural artificial passa pela análise de diversos fatores determinantes. Em um mesmo grupo de dados foi analisado diversos fatores a fim de analisar o melhor desempenho de uma RNA aplicada ao caso proposto. Inicialmente preocupou-se com o número de camadas ocultas, sendo o primeiro teste realizado com quatro camadas ocultas comparativamente com uma. É possível notar na Tabela 5 que a redução de erro para um maior número de camadas ocultas é pequena e, por isso, convencionou-se a utilização de apenas uma camada oculta para as análises seguintes.

Todos os neurônios organizados representam uma estrutura RNA. A parte de treinamento define um conjunto de regras para a solução do aprendizado da rede neural e melhora no desempenho. Além disso, realiza-se o treinamento e a validação. No período de treinamento, o código realiza a alteração do peso sináptico do sistema, realizando o ajuste da rede até a finalização do treinamento. A Figura 13 mostra a comparação realizada no algoritmo de treinamento [PUC-Rio, 2014].

Figura 13 – Treinamento de uma RNA



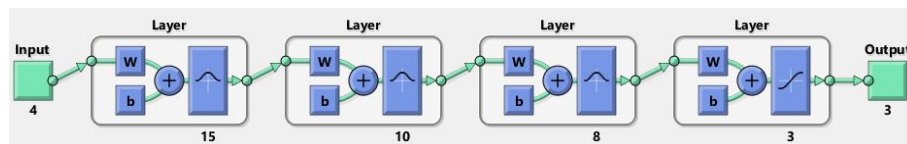
Fonte: [PUC-Rio, 2014]

### 5.1.1 Quantidade de Camadas Ocultas

Na primeira análise procura-se avaliar a diferença na qualidade da resposta com relação à mudança do número de camadas ocultas da rede neural artificial.

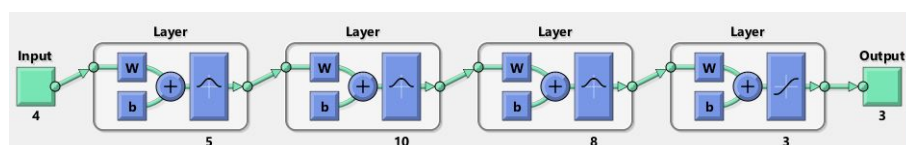
Com uma configuração de três camadas ocultas realiza-se empiricamente a alteração da quantidade de neurônios para analisar a melhor resposta para a rede.

Figura 14 – Configuração de rede A



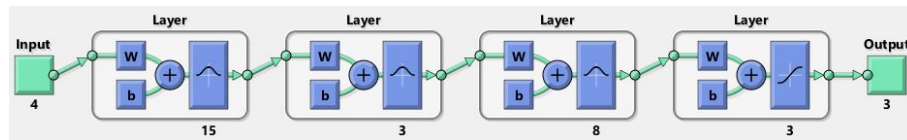
Fonte: Autoria própria

Figura 15 – Configuração de rede B



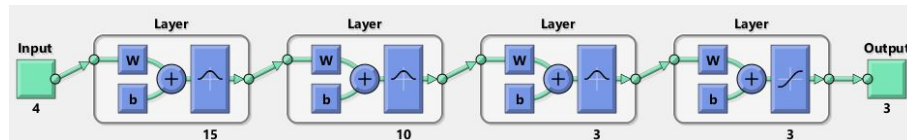
Fonte: Autoria própria

Figura 16 – Configuração de rede C



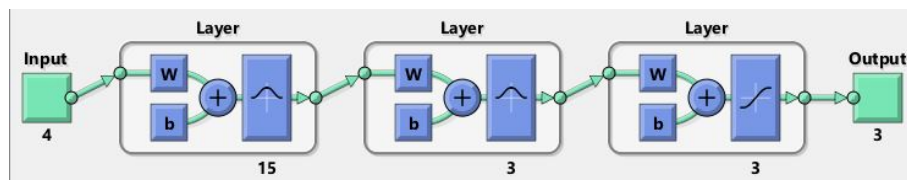
Fonte: Autoria própria

Figura 17 – Configuração de rede D



Fonte: Autoria própria

Figura 18 – Configuração de rede E

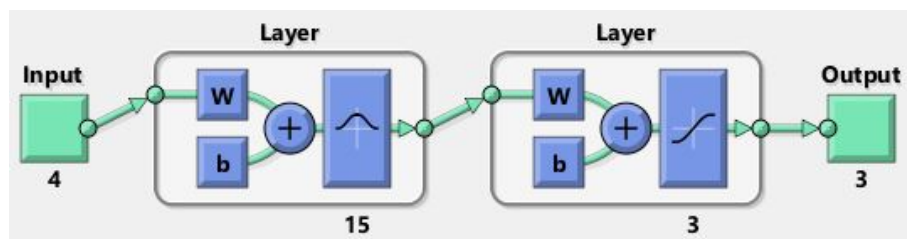


Fonte: Autoria própria

Conforme analisado a quantidade de neurônios para três camadas ocultas, analisa-se na Tabela 5 diferentes configurações de quantidade de camadas ocultas a fim de procurar uma condição mais adequada para o estudo.

Com base nas informações apresentadas, percebe-se que um número maior de camadas ocultas reduz o erro da rede, no entanto, aumenta o tempo de processamento e exige maior poder computacional devido o aumento do número de camadas ocultas.

Figura 19 – Configuração de rede F



Fonte: Autoria própria

Tabela 4 – Variação de número de neurônios para 3 camadas ocultas

Configuração	Taxa de Erro	Tempo (s)
A	9,63%	0,0331
B	14,05%	0,0260
C	11,28%	0,0257
D	9,06%	0,0324

Fonte: Autoria própria

Tabela 5 – Quantidade de Camadas Ocultas

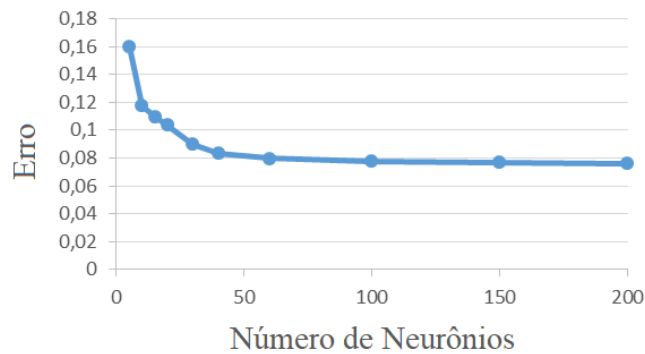
Configuração	Quantidade de Camadas Ocultas	Taxa de Erro	Tempo (s)
D	3	9,06%	0,0324
E	2	10,56%	0,0228
F	1	12,68%	0,0209

Fonte: Autoria própria

### 5.1.2 Número de Neurônios na Camada Oculta

De modo empírico, utilizando o método de Redes Neurais Artificiais, realiza-se a avaliação da classificação do sistema para diferentes valores de neurônios na camada oculta. Desta forma, obtêm-se o gráfico apresentado na Figura 20 que representa o erro na classificação do sistema para quantidades diferentes de neurônios.

Figura 20 – Arquitetura de Redes Neurais Artificiais



Fonte: Adaptado de [Marins et al., 2017]

Na Tabela 6 tem-se três exemplos de estruturas com uma camada oculta e variando a quantidade de neurônios nessa camada. Como é possível observar, a redução do número de neurônios aumenta o erro da rede neural artificial, conforme apresentado na Figura 20.

Na Tabela 6 é apresentado uma análise empírica de três exemplos de redes neurais artificiais a fim de exemplificar a curva apresentada na Figura 20.

Tabela 6 – Número de Neurônios na Camada Oculta

Número de Neurônios	Taxa de Erro	Tempo (s)
20	10,09%	0,0234
15	12,68%	0,0209
10	14,64%	0,0224
5	14,89%	0,0219

Fonte: Autoria própria

### 5.1.3 Quantidade de Épocas

Em seguida, analisa-se a quantidade de épocas a fim de adequar o número de épocas ao sistema.

Tabela 7 – Quantidade de Épocas

Quantidade de Épocas	Taxa de Erro	Tempo (s)
10	30,46%	0,0265
100	17,56%	0,0250
500	14,30%	0,0211
1000	12,80%	0,0283

Fonte: Autoria própria

Com base nas informações apresentadas na Tabela 7 e nas simulações realizadas descreve-se que, na média das simulações a convergência ocorre entre 700 e 900 épocas. Esta análise auxilia no ajuste da rede, pois permite que a quantidade de épocas esteja adequada ao caso em análise.

### 5.1.4 Função de Ativação

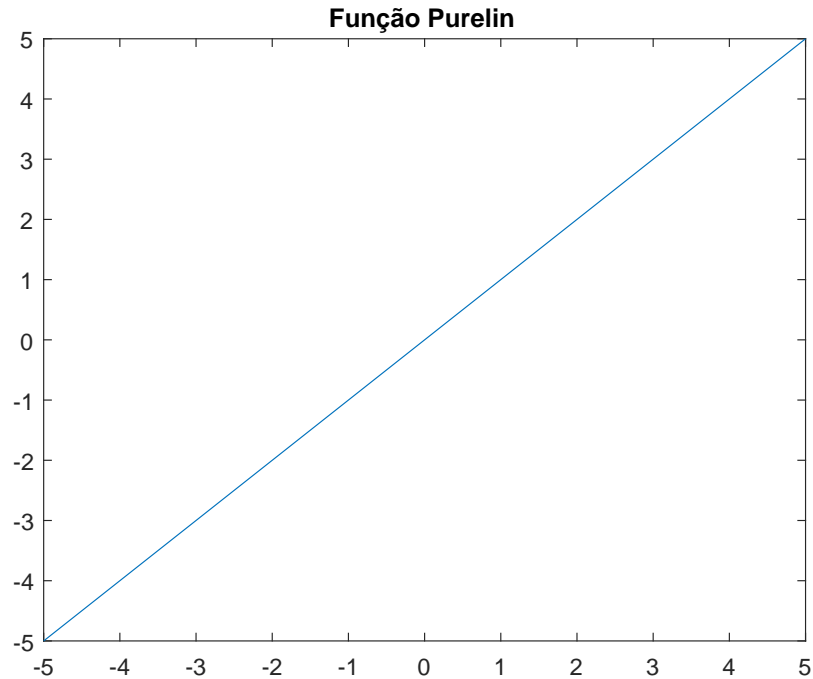
A função de ativação processa o sinal gerado pela combinação linear dos pesos sinápticos. A análise das funções de ativação foi realizada alterando a função utilizada na camada oculta. Para todas as análises foi utilizado um mesmo grupo de dados do **Usuário 1**. Nessa dissertação apresenta-se quatro funções de ativação e seu respectivo comparativo nos resultados.

#### 5.1.4.1 Função Linear

É uma função puramente linear denotada por *purelin* em MatLab<sup>®</sup> e representada pelo gráfico apresentado na Figura 21.

#### 5.1.4.2 Função Sigmoidal

É uma função crescente que apresenta um balanço entre o comportamento linear e não-linear. Sua resposta gráfica tem o formato de **S**. A Equação da função sigmoide é

Figura 21 – Função *Purelin*

Fonte: Autoria própria

apresentada em 5.1 [PUC-Rio, 2014].

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (5.1)$$

Ademais, tem-se as funções Tangente Sigmoid e Elliot Sigmoid que são representadas pelas Equações 5.2a e 5.2b.

$$f(x) = \frac{e^{\lambda x} - e^{-\lambda x}}{e^{\lambda x} + e^{-\lambda x}} \quad (5.2a)$$

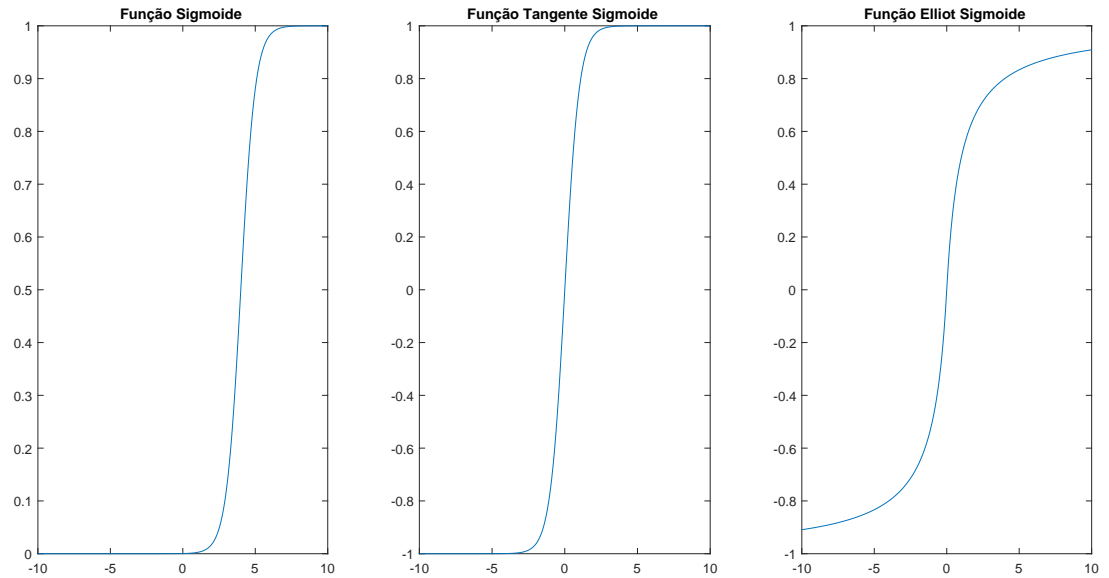
$$f(x) = \frac{\frac{ax}{2}}{1 + ax} + 0,5 \quad (5.2b)$$

A Figura 22 apresenta o comparativo das funções sigmoide.

#### 5.1.4.3 *Radbas*

O *Radbas* é uma função de transferência de base radial, exemplificada pelo gráfico apresentado na Figura 23.

Compara-se as funções de ativação descritas a fim de determinar a função mais adequada para a rede em análise 8.

Figura 22 – Função *Sigmoide*

Fonte: Autoria própria

Tabela 8 – Funções de ativação

Função de ativação	Taxa de Erro	Tempo (s)
Linear	68,28%	0,0238
<i>Radbas</i>	10,60%	0,0193
Tangente Sigmoide	10,97%	0,0283
<i>Elliot Sigmoide</i>	10,75%	0,0202

Fonte: Autoria própria

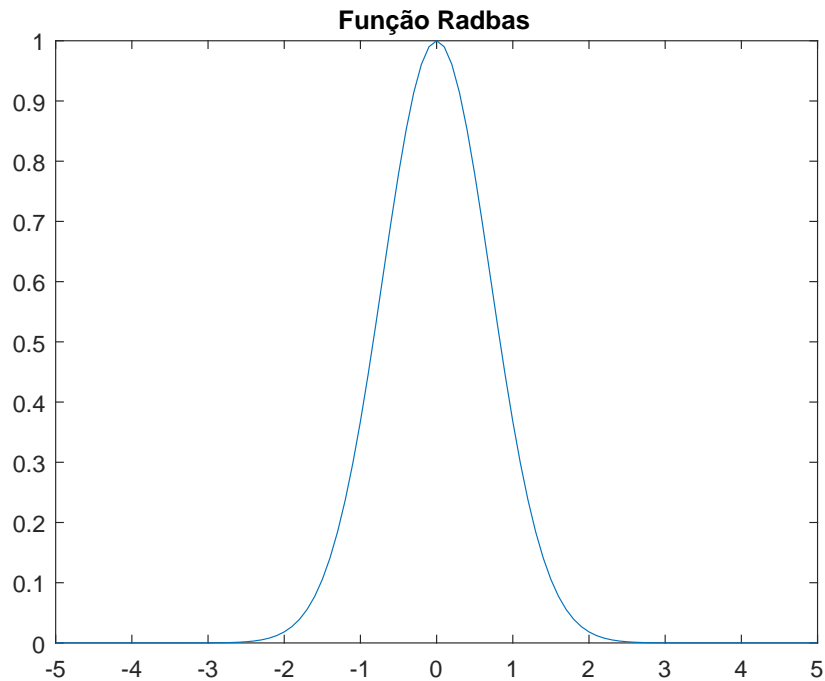
A Tabela 8 mostra um comparativo entre funções de ativação, mostrando que a função *radbas* adequa-se de forma mais efetiva ao caso. A função linear não é capaz de ajustar a rede de forma eficiente e as duas funções sigmoideais possuem respostas próximas ao modelo *radbas*, porém com erros maiores.

Com os ajustes da rede neural artificial é possível obter uma resposta melhor que a obtida no modelo de Mahalanobis. No entanto, a estrutura de Mahalanobis é mais simples e utiliza menor processamento que a estrutura de RNA, o que é vantajoso para estruturas embarcadas.

### 5.1.5 Comparação entre usuários diferentes

Por fim, realiza-se a comparação entre duas redes com dados coletados por dois usuários diferentes com cem mil amostras de dados distribuídas nas cinco classes. Utilizando as condições de rede descritas na Tabela 9 para ambas, obtem-se os erros e tempos de



Figura 23 – Função *Radbas*

Fonte: Autoria própria

conversão apresentados na Tabela 10.

Tabela 9 – Condições da Rede Neural Artificial

Função de ativação <i>radbas</i>	Nº máximo de épocas 1000	Neurônios na camada oculta 15
-------------------------------------	-----------------------------	----------------------------------

Fonte: Autoria própria

Tabela 10 – Comparação entre redes

	Erro	Tempo (s)
<b>Usuário 1</b>	11,50%	0,0202
<b>Usuário 2</b>	15,47%	0,0204

Fonte: Autoria própria

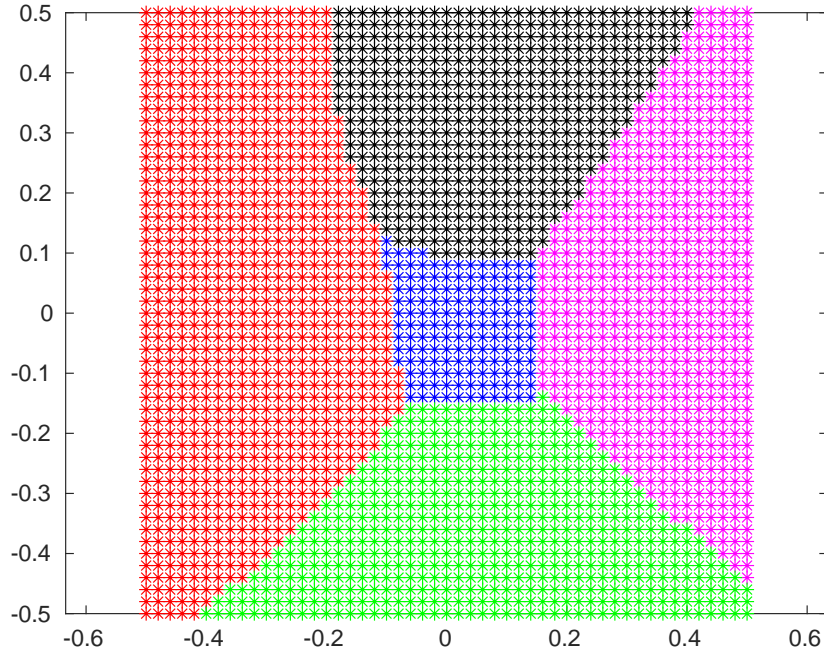
Pela Tabela 10 é possível perceber que a alteração de usuário influencia no resultado final da Rede. De qualquer forma, para aumentar a qualidade do resultado é possível gerar novas redes para o usuário em estudo.

## 5.2 SIMULAÇÃO EM MATLAB®

A partir dos valores obtidos em *Roll* e *Pitch* com base na Equação 4.12 é possível testar todos os pontos da margem escolhida e obter a divisão prevista de classes utilizando

a técnica de Mahalanobis como mostrado na Figura 24.

Figura 24 – Divisão de Classes



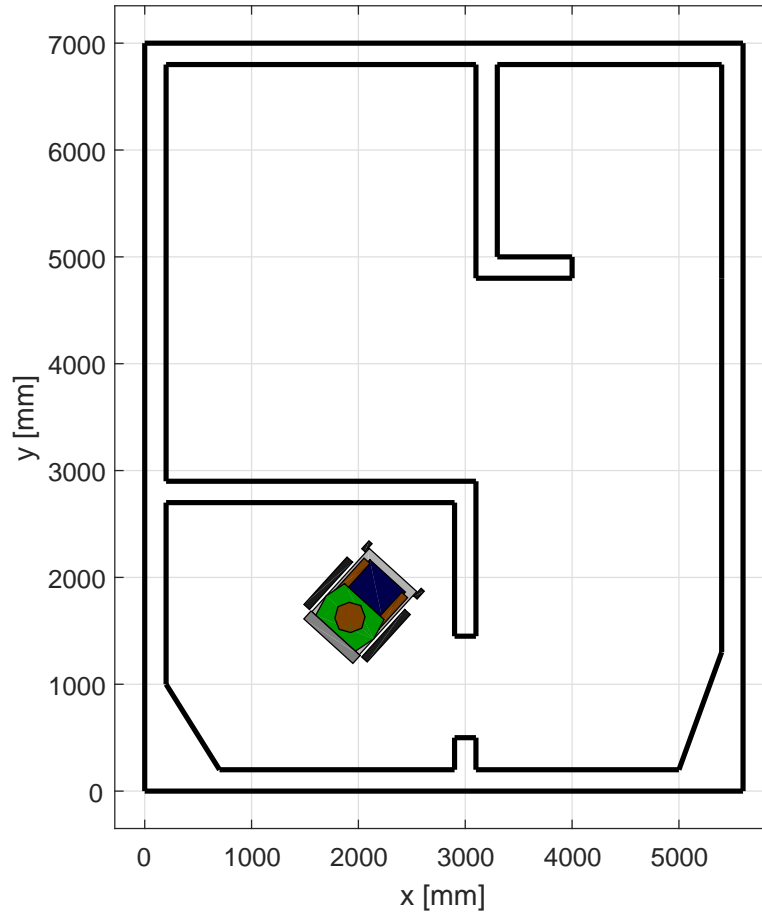
Fonte: Autoria própria

No teste no *Pioneer 3DX* foi possível confirmar a confiabilidade do sistema verificando se a classificação realmente se adequa à resposta esperada do sistema. No teste com o robô móvel (Seção 5.4) foi obtido 86,86% de acertos no sistema, no entanto, vale ressaltar que em todos os casos do teste a classe foi incorretamente classificada como "Classe 1", ou seja, a classe em que a cadeira fica parada. Desta forma, o módulo é capaz de facilitar a locomoção de um usuário de cadeira de rodas com o dispositivo desenvolvido e nos casos de erro de classificação a cadeira deve ficar imóvel.

A estrutura do código de simulação em MatLab<sup>®</sup> é apresentado no Apêndice A, no entanto, é fundamental descrever alguns pontos do desenvolvimento para facilitar o entendimento do programa. Inicialmente, caso o ROS ainda não tenha sido inicializado é necessário inicializar. O programa recebe os dados do giroscópio e também *roll* e *pitch* já convertidos no programa de arduino a partir dos dados de aceleração do IMU, calibração geral e individual apresentadas no Capítulo 4. Após a recepção dos dados do IMU é realizado a avaliação dos dados e atribuição para uma das cinco classes divididas inicialmente através do método de Mahalanobis. Por fim, o simulador em MatLab<sup>®</sup> é inicializado e apresenta visualmente a movimentação do modelo implementado, conforme mostrado na Figura 25.

Figura 25 – Mapa utilizado em MatLab

**Robo<sub>x</sub> = 2055.02 mm, Robo<sub>y</sub> = 1773.60 mm, Robo<sub>th</sub> = 47.75 deg**  
**Mapa = 7**



Fonte: Autoria própria

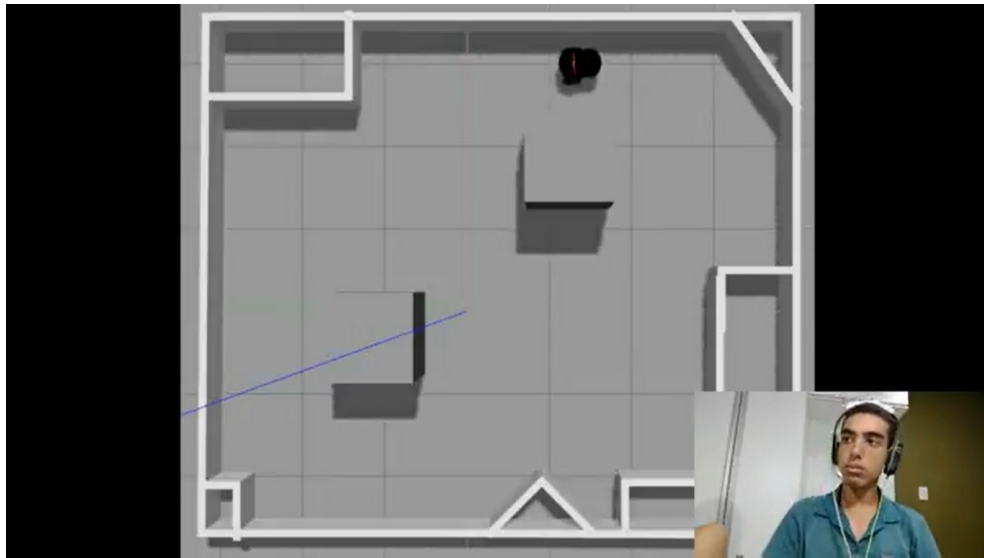
### 5.3 SIMULAÇÃO NO GAZEBO®

O software Gazebo® permite simular ambientes e diversos tipos de robôs a fim de facilitar a implementação real. Permite simulação em ambientes internos e externos.

Desta forma, foi realizado testes em dois ambientes e estão disponíveis em <https://www.youtube.com/watch?v=LtZB4EoizqU> e <https://www.youtube.com/watch?v=BoU48fSTKGA>. As Figuras 26 e 27 mostram os ambientes simulados.

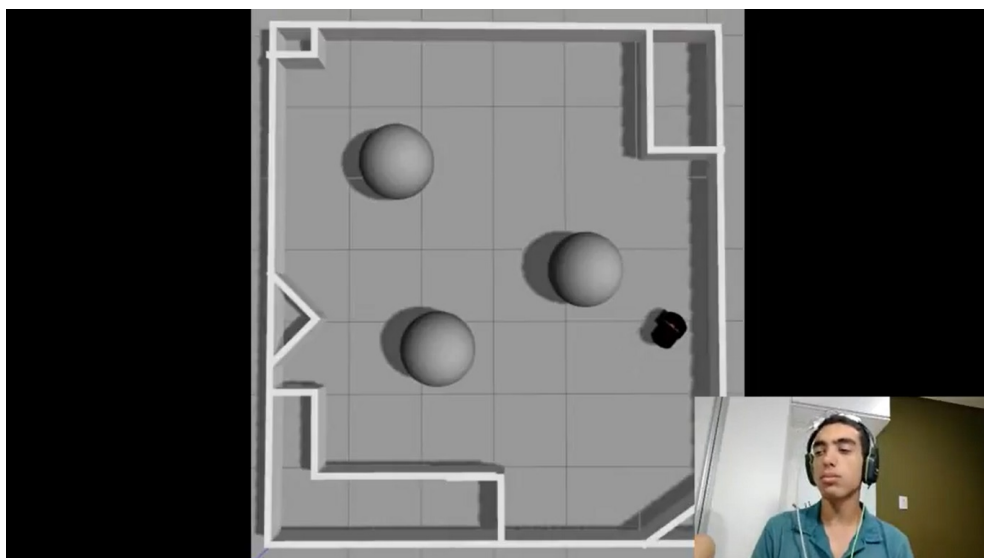
O teste realizado no ambiente virtual Gazebo® mostrou os resultados apresentados na Tabela 11 que comparam os resultados obtidos em cada um dos cenários. Percebe-se que no cenário 2 o erro é maior, o que pode ser atribuído às dificuldades do cenário e erros estocásticos. É importante dizer também que os erros encontrados são, em 99,3% dos casos, colocados na região neutra, ou seja, a maioria dos erros são encontrados na região que não realiza movimento na cadeira de rodas.

Figura 26 – Primeiro mapa utilizado em Gazebo®



Fonte: Autoria própria

Figura 27 – Segundo mapa utilizado em Gazebo®



Fonte: Autoria própria

#### 5.4 TESTE NO PIONEER 3DX

Simulação é fundamental no campo da robótica para que seja possível uma análise acurada do elementos trabalhados e ajustar os códigos e implementações para a implementação em robôs físicos. O *Pioneer 3DX* que é objeto de estudo neste trabalho [Koenig and Howard, 2013] foi utilizado para realizar uma simulação do módulo desenvolvido.

O experimento conta com a recepção de dados do movimento da cabeça do usuário através de uma IMU que envia dados de acelerômetro e giroscópio para o Arduino. Os

Tabela 11 – Comparativo entre Ambientes

Erro - Ambiente 1	Erro - Ambiente 2
10,58%	11,37%

Fonte: Autoria própria

dados são tratados através de filtro de Kalman e convertidos para *roll* e *pitch*, classificados por Mahalanobis em MatLab e atuado utilizando a plataforma ROS no *Pioneer 3DX* atuando o robô móvel. Na Figura 28 tem uma imagem do robô atuado por movimentos da cabeça do ambiente de teste. O vídeo disponível no link <https://youtu.be/FMwy883uywQ> mostra um teste realizado no robô móvel utilizado.

Figura 28 – Teste com robô móvel



Fonte: Autoria própria

No teste com o robô móvel foi constatado um erro de 11,3% comparando os comandos enviados com a resposta ao movimento. É importante destacar que, em todos os casos desse teste, quando o robô erra no movimento, a classificação ocorre na posição neutra, ou seja, em casos de classificação errônea o robô não realizará movimento, característica fundamental no caso em estudo para que o usuário de cadeira de rodas não realize um movimento em uma direção não desejada.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Com base no trabalho apresentado é possível concluir que foi desenvolvido uma plataforma assistiva econômica para atender pessoas com tetraplegia. Deve-se destacar a preocupação uma plataforma com preço menor que o praticado comercialmente a fim de que a população com menor renda possa comprar o tipo de tecnologia desenvolvido. O desenvolvimento ocorreu apenas em ambientes de simulação, desta forma, não é possível inferir um preço final para o produto, no entanto, os componentes utilizados para controle foram escolhidos com base na eficiência para a aplicação e economia.

Com relação a proposta educacional foi desenvolvido um tutorial com o desenvolvimento completo, incluído os códigos, a fim de que outros alunos de engenharia possam continuar o projeto e implementar melhorias a fim de atingir um produto comercial. Apesar dessa preocupação, o principal objetivo com o desenvolvimento do tutorial é estimular os alunos a ter novos modelos de implementação no curso de engenharia, com aplicações práticas em que são desafiados a entender e implementar novos projetos e, no decorrer do desenvolvimento, aprimorar competências transversais como trabalho em equipe, liderança, exposição de ideias, dentre outras que são fundamentais para os profissionais no mercado de trabalho atual.

No desenvolvimento do projeto da cadeira de rodas sugere-se como trabalho futuro a implementação de um novo IMU para que tenha um IMU como referência e outro para a movimentação a fim de que a resposta fique mais adequada para a implementação. Recomenda-se também a inclusão de sonares na estrutura da cadeira de rodas, já que são dispositivos de baixo custo comparado a sensores como laser e *kinect* a fim de implementar um controle compartilhado, evitar colisões e ajustar rotas na condução da cadeira de rodas. Ademais, uma alternativa para reduzir o custo final da cadeira de rodas é implementar toda a automatização dela a partir de uma cadeira de modelo básico.

## APÊNDICE A – Tutorial do Módulo Assistivo

### A.1 PLATAFORMA ROS

ROS (*Robot Operating System*) é um *framework* para utilização em sistemas robóticos e integra algoritmos, *hardwares* e robôs. Tem-se uma coleção de ferramentas e bibliotecas que visam simplificar e simular plataformas robóticas. Desta forma, o ROS foi criado para incentivar o desenvolvimento colaborativo da robótica, permitindo que pesquisadores em torno do mundo possam contribuir para o desenvolvimento da robótica de forma direta e efetiva [ROS, ].

No desenvolvimento deste tutorial foi utilizada a Plataforma ROS Indigo, no entanto, o estudante poderá utilizar a versão mais recente da plataforma sendo apenas necessário atenção nas configurações necessárias para a utilização da plataforma. No site [ros.org](http://ros.org) há uma descrição detalhada do processo de instalação do ROS. A plataforma é destinada ao sistema operacional Linux, desta forma, caso não seja usuário de Linux é necessário realizar a instalação deste sistema operacional ou instalar um ambiente que possa executar Linux dentro de outro sistema operacional.

Com a plataforma ROS instalada, o próximo passo é realizar os tutoriais também disponíveis no site [ros.org](http://ros.org) a fim de entender o funcionamento da plataforma e ser capaz de desenvolver aplicações utilizando-a.

### A.2 MONTAGEM

Para montagem do circuito são necessários os seguintes equipamentos:

- Um Arduino com cabo USB (no exemplo utiliza-se o Arduino UNO);
- Um IMU;
- Uma protoboard;
- Um suporte para colocar sobre a cabeça;
- Um robô diferencial *Pioneer 3DX* (uso apenas no teste real).

Realize as conexões conforme esboço descrito na figura 29.

### A.3 PROGRAMAÇÃO EM ARDUINO

A programação é feita em duas vertentes. A primeira trata do envio de dados no sistema operacional Windows e a segunda o envio de informações para o ROS. O primeiro está descrito como `enviodadosimu.ino` e o segundo `ProgramaROS.ino`.

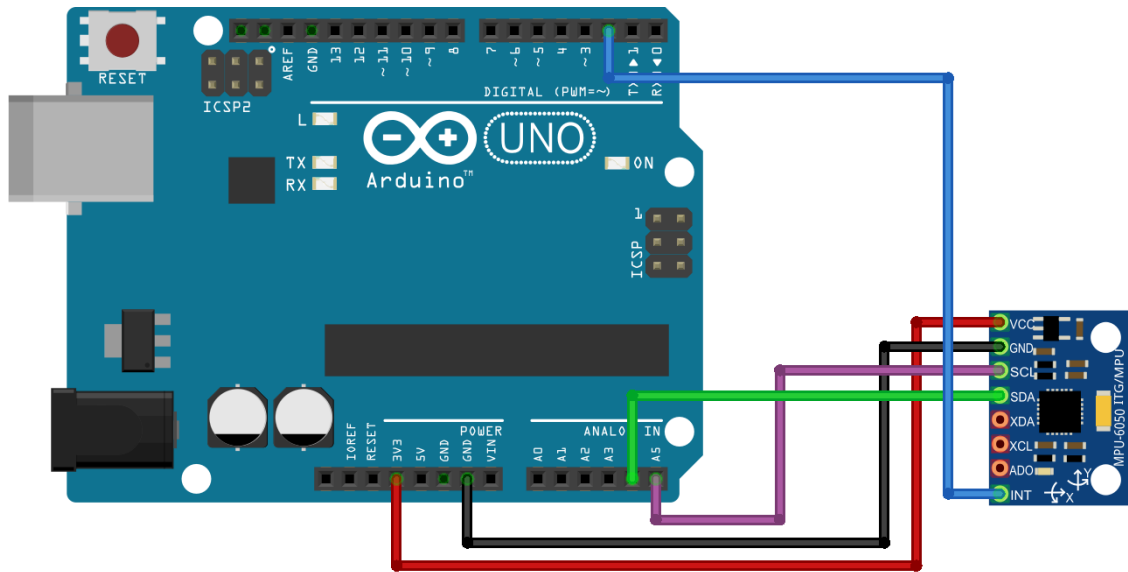
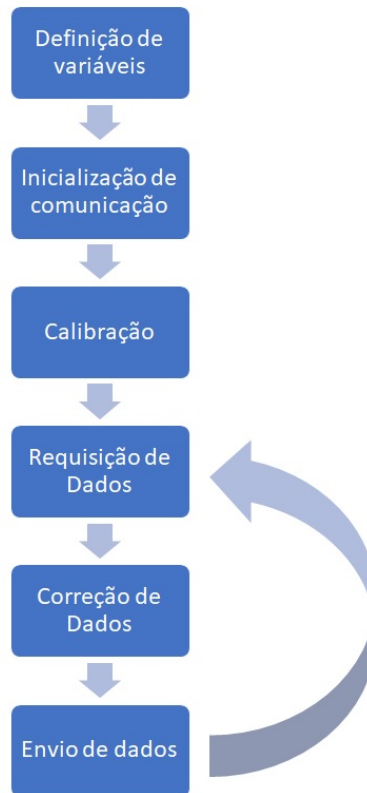


Figura 29 – Conexões Arduino com IMU

Ambos os códigos encontram-se no github e o fluxograma seguinte representa o fluxo lógico a fim de ilustrar os passos necessários nessa parte do desenvolvimento.

Figura 30 – Fluxograma Arduino



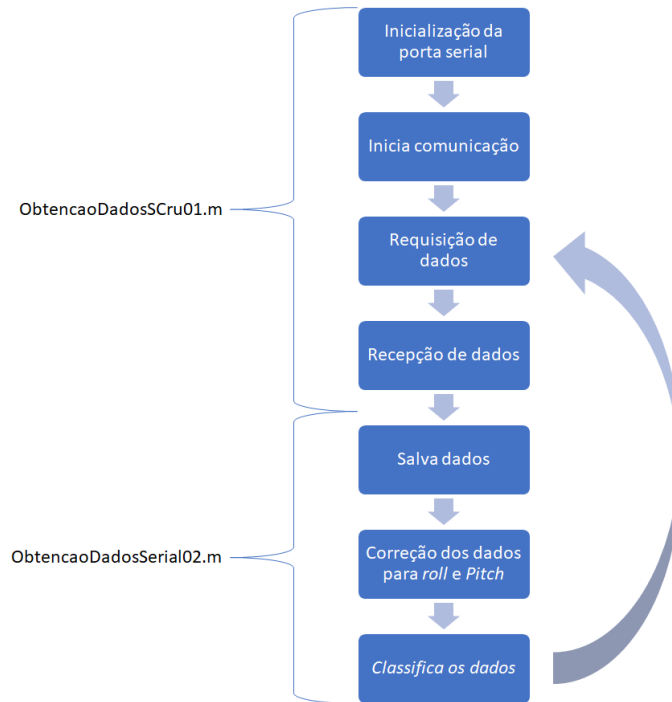
Fonte: Autoria própria



#### A.4 RECEPÇÃO DE DADOS NO MATLAB®

O programa do Arduino® realiza as leituras dos dados e envia para o Software MATLAB®. Com isso, o código de MATLAB® realiza a obtenção dos dados da porta serial (ObtencaoDadosCru01.m) e correção dos valores para *roll* e *pitch* (ObtencaoDadosSerial02.m). O fluxo apresentado na Figura 31 mostra o funcionamento do código descrito.

Figura 31 – Fluxograma MATLAB - Recepção de Dados Serial

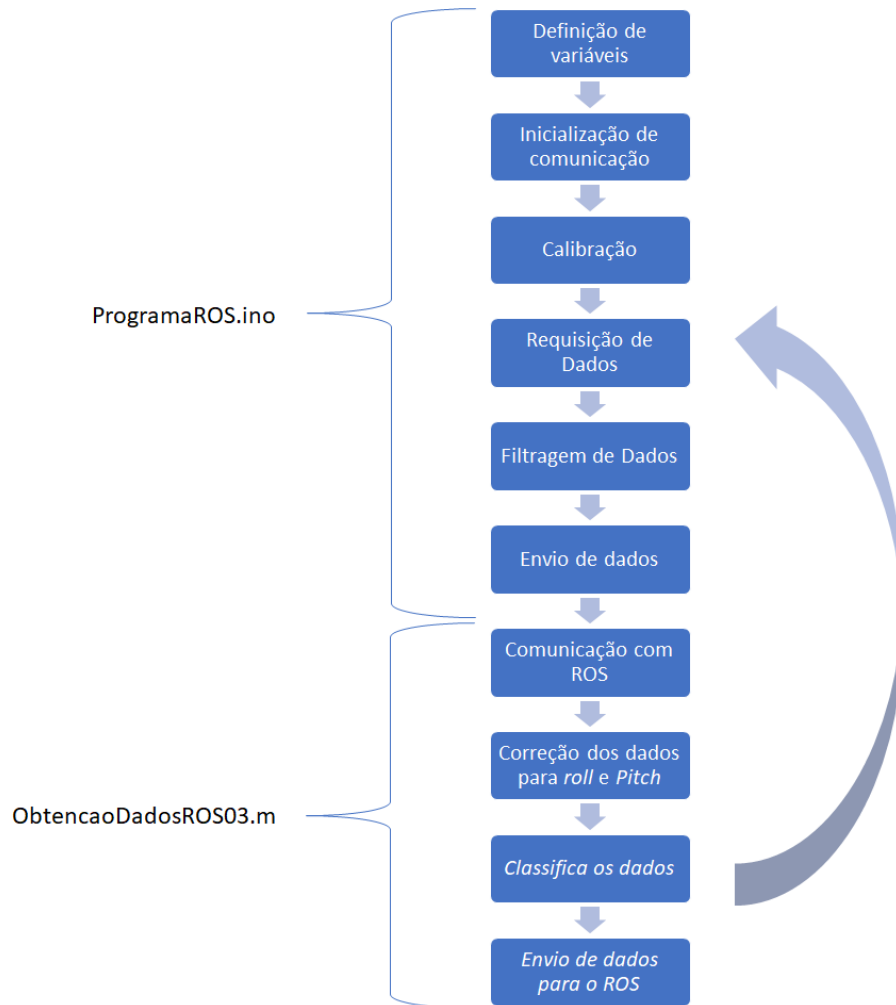


Fonte: Autoria própria

## A.5 PROGRAMAS PARA A PLATAFORMA ROS

A seguir tem-se o Programa Arduino<sup>®</sup> com comunicação para o ROS denominado ProgramaROS.ino. O programa em Arduino<sup>®</sup> realiza a leitura, filtragem e envio de dados para o MATLAB<sup>®</sup> (ObtencaoDadosROS03.m) que realiza a recepção, classificação e envio de dados para a plataforma ROS.

Figura 32 – Fluxograma ROS



Fonte: Autoria própria

## A.6 OBTENÇÃO DA MATRIZ DE CORREÇÃO

Conforme descrito neste trabalho, os dados obtidos através do IMU necessitam ser tratados a fim de serem utilizados para o sistema de coordenadas *Roll*, *Pitch* e *Yaw*. O código que realiza a obtenção da matriz de correção é `ObtencaoMatrizCorrecao04.m`.

### **APÊNDICE A - Tutorial do Módulo Assistivo**

## APÊNDICE B - Método dos Mínimos Quadrados

O Método dos Mínimos Quadrados (MMQ) é um método estatístico utilizado para obter estimativas de dados experimentais sujeitos a erros aleatórios. Usada também para estimar funções, é uma técnica largamente utilizada e eficiente para análises estimativas de dados experimentais. A fim de ilustrar melhor a aplicação do MMQ apresenta-se um exemplo a seguir [Helene, 2006].

Um estudante realizou medidas de um quadrado com uma régua e encontrou os seguintes valores:

- Lado = 5 cm;
- Área = 30 cm<sup>2</sup>;
- Perímetro = 15 cm.

Deseja-se obter o melhor valor para o lado do quadrado apresentado, por isso, utiliza-se o método dos mínimos quadrados. É necessário manipular as equações para aplicação do método.

$$\text{Lado} = l, \text{ como Lado} = 5\text{cm}, \text{ tem-se } l - 5 = 0$$

$$\text{Área} = l^2, \text{ como Área} = 30\text{cm}^2, \text{ tem-se } l^2 - 30 = 0$$

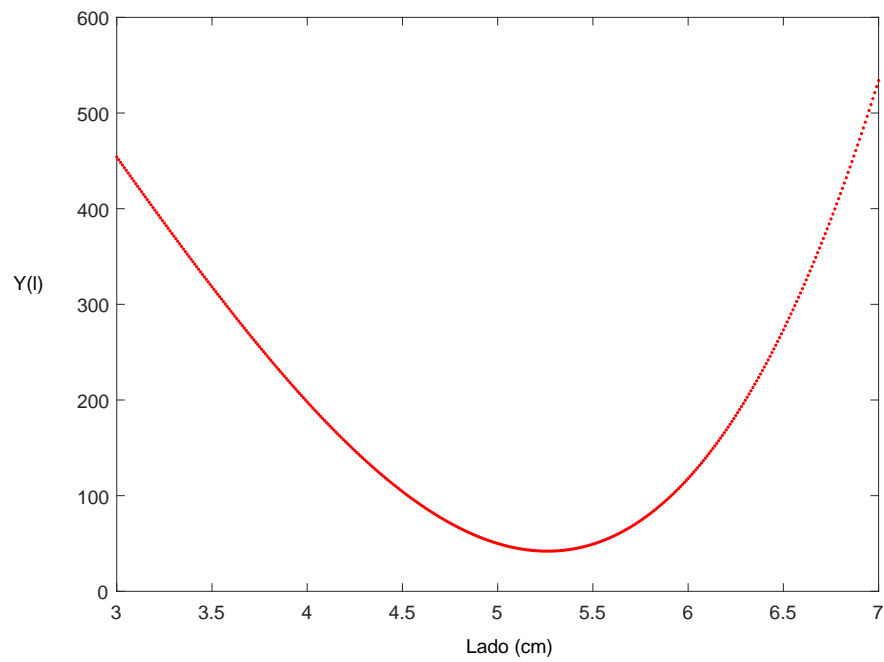
$$\text{Perímetro} = 4l, \text{ como Perímetro} = 15\text{cm}, \text{ tem-se } 4l - 15 = 0$$

O valor adequado para o lado do quadrado ( $l$ ) é encontrado com base no valor mínimo da Equação 1 obtida elevando cada equação obtida ao quadrado e igualando a zero.

$$Y(l) = (l - 5)^2 + (l^2 - 30)^2 + (4l - 15)^2 \quad (1)$$

O gráfico apresentado na Figura 33 representa a análise do sistema para valores de  $l$  na faixa de 3 a 7 com precisão de duas casas decimais. Com base no gráfico obtem-se que o menor valor de  $Y(l)$  é 41,9893 e o correspondente valor do lado pelo MMQ é 5,26.

Figura 33 – Resposta do Sistema de Exemplificação



Fonte: Autoria própria

## APÊNDICE B - Método dos Mínimos Quadrados

## ANEXO A – Tabelas de Shapiro-Wilk

Neste anexo são apresentados os coeficientes  $a_i$  e os valores críticos  $p$  para o teste Shapiro-Wilk [Verrill and Johnson, 1988].

Tabela 12 – Shapiro-Wilk: Coeficientes  $a_i$ : Parte 1

$i \backslash n$	2	3	4	5	6	7	8	9	10
1	0,7071	0,7071	0,6872	0,6646	0,6431	0,6233	0,6062	0,5888	0,5739
2			0,1677	0,2413	0,2806	0,3031	0,3164	0,3244	0,3291
3					0,0875	0,1401	0,1743	0,1976	0,2141
4							0,0561	0,0947	0,1224
5									0,0399

Tabela 13 – Shapiro-Wilk: Coeficientes  $a_i$ : Parte 2

$i \backslash n$	11	12	13	14	15	16	17	18	19	20
1	0,5601	0,5475	0,5359	0,5251	0,5150	0,5056	0,4968	0,4886	0,4808	0,4734
2	0,3315	0,3325	0,3325	0,3318	0,3306	0,3290	0,3273	0,3253	0,3232	0,3211
3	0,2260	0,2347	0,2412	0,2460	0,2495	0,2521	0,2540	0,2553	0,2561	0,2565
4	0,1429	0,1586	0,1707	0,1802	0,1878	0,1939	0,1988	0,2027	0,2059	0,2085
5	0,0695	0,0922	0,1099	0,1240	0,1353	0,1447	0,1524	0,1587	0,1641	0,1686
6		0,0303	0,0539	0,0727	0,0880	0,1005	0,1109	0,1197	0,1271	0,1334
7				0,0240	0,0433	0,0593	0,0725	0,0837	0,0932	0,1013
8						0,0196	0,0359	0,0496	0,0612	0,0711
9								0,0163	0,0303	0,0422
10										0,0140

Tabela 14 – Shapiro-Wilk: Coeficientes  $a_i$ : Parte 3

$i \backslash n$	21	22	23	24	25	26	27	28	29	30
1	0,4643	0,4590	0,4542	0,4493	0,4450	0,4407	0,4366	0,4328	0,4291	0,4254
2	0,3185	0,3156	0,3126	0,3098	0,3069	0,3043	0,3018	0,2992	0,2968	0,2944
3	0,2578	0,2571	0,2563	0,2554	0,2543	0,2533	0,2522	0,2510	0,2499	0,2487
4	0,2119	0,2131	0,2139	0,2145	0,2148	0,2151	0,2152	0,2151	0,2150	0,2148
5	0,1736	0,1764	0,1787	0,1807	0,1822	0,1836	0,1848	0,1857	0,1864	0,1870
6	0,1399	0,1443	0,1480	0,1512	0,1539	0,1563	0,1584	0,1601	0,1616	0,1630
7	0,1092	0,115	0,1201	0,1245	0,1283	0,1316	0,1346	0,1372	0,1395	0,1415
8	0,0804	0,0878	0,0941	0,0997	0,1046	0,1089	0,1128	0,1162	0,1192	0,1219
9	0,0530	0,0618	0,0696	0,0764	0,0823	0,0876	0,0923	0,0965	0,1002	0,1036
10	0,0263	0,0368	0,0459	0,0539	0,061	0,0672	0,0728	0,0778	0,0822	0,0862
11		0,0122	0,0228	0,0321	0,0403	0,0476	0,0540	0,0598	0,065	0,0697
12				0,0107	0,0200	0,0284	0,0358	0,0424	0,0483	0,0537
13					0,0000	0,0094	0,0178	0,0253	0,032	0,0381
14							0,0000	0,0084	0,0159	0,0227
15									0	0,0076

Tabela 15 – Shapiro-Wilk: Coeficientes  $a_i$ : Parte 4

$i \backslash n$	31	32	33	34	35	36	37	38	39	40
1	0,4220	0,4188	0,4156	0,4127	0,4096	0,4068	0,4040	0,4015	0,3989	0,3964
2	0,2921	0,2898	0,2876	0,2854	0,2834	0,2813	0,2794	0,2774	0,2755	0,2737
3	0,2475	0,2463	0,2451	0,2439	0,2427	0,2415	0,2403	0,2391	0,2380	0,2368
4	0,2145	0,2141	0,2137	0,2132	0,1227	0,2121	0,2116	0,2110	0,2104	0,2098
5	0,1874	0,1878	0,1880	0,1882	0,1883	0,1883	0,1883	0,1881	0,1880	0,1878
6	0,1641	0,1651	0,1660	0,1667	0,1673	0,1678	0,1683	0,1686	0,1689	0,1691
7	0,1433	0,1449	0,1463	0,1475	0,1487	0,1496	0,1505	0,1513	0,1520	0,1526
8	0,1243	0,1265	0,1284	0,1301	0,1317	0,1331	0,1344	0,1356	0,1366	0,1376
9	0,1066	0,1093	0,1118	0,1140	0,1160	0,1179	0,1196	0,1211	0,1225	0,1237
10	0,0899	0,0931	0,0961	0,0988	0,1013	0,1036	0,1056	0,1075	0,1092	0,1108
11	0,0739	0,0777	0,0812	0,0844	0,0873	0,0900	0,0924	0,0947	0,0967	0,0986
12	0,0585	0,0629	0,0669	0,0706	0,0739	0,0770	0,0798	0,0824	0,0848	0,0870
13	0,0435	0,0485	0,0530	0,0572	0,0610	0,0645	0,0677	0,0706	0,0733	0,0759
14	0,0289	0,0344	0,0395	0,0441	0,0484	0,0523	0,0559	0,0592	0,0622	0,0651
15	0,0144	0,0206	0,0262	0,0314	0,0361	0,0404	0,0444	0,0481	0,0515	0,0546
16	0,0000	0,0068	0,0131	0,0187	0,0239	0,0287	0,0331	0,0372	0,0409	0,0444
17			0,0000	0,0062	0,0119	0,0172	0,0220	0,0264	0,0305	0,0343
18					0,0000	0,0057	0,0110	0,0158	0,0203	0,0244
19							0,0000	0,0053	0,0101	0,0146
20									0,0000	0,0049

Tabela 16 – Shapiro-Wilk: Coeficientes  $a_i$ : Parte 5

$i \backslash n$	41	42	43	44	45	46	47	48	49	50
1	0,3940	0,3917	0,3894	0,3872	0,3850	0,3830	0,3808	0,3789	0,3770	0,3751
2	0,2719	0,2701	0,2684	0,2667	0,2651	0,2635	0,2620	0,2604	0,2589	0,2574
3	0,2357	0,2345	0,2334	0,2323	0,2313	0,2302	0,2291	0,2281	0,2271	0,2260
4	0,2091	0,2085	0,2078	0,2072	0,2065	0,2058	0,2052	0,2045	0,2038	0,2032
5	0,1876	0,1874	0,1871	0,1868	0,1865	0,1862	0,1859	0,1855	0,1851	0,1847
6	0,1693	0,1694	0,1695	0,1695	0,1695	0,1695	0,1695	0,1693	0,1692	0,1691
7	0,1531	0,1535	0,1539	0,1542	0,1545	0,1548	0,1550	0,1551	0,1553	0,1554
8	0,1384	0,1392	0,1398	0,1405	0,1410	0,1415	0,1420	0,1423	0,1427	0,1430
9	0,1249	0,1259	0,1269	0,1278	0,1286	0,1293	0,1300	0,1306	0,1312	0,1317
10	0,1123	0,1136	0,1149	0,1160	0,1170	0,1180	0,1189	0,1197	0,1205	0,1212
11	0,1004	0,1020	0,1035	0,1049	0,1062	0,1073	0,1085	0,1095	0,1105	0,1113
12	0,0891	0,0909	0,0927	0,0943	0,0959	0,0972	0,0986	0,0998	0,1010	0,1020
13	0,0782	0,0804	0,0824	0,0842	0,0860	0,0876	0,0892	0,0906	0,0919	0,0932
14	0,0677	0,0701	0,0724	0,0745	0,0765	0,0783	0,0801	0,0817	0,0832	0,0846
15	0,0575	0,0602	0,0628	0,0651	0,0673	0,0694	0,0713	0,0731	0,0748	0,0764
16	0,0476	0,0506	0,0534	0,0560	0,0584	0,0607	0,0628	0,0648	0,0667	0,0685
17	0,0379	0,0411	0,0442	0,0471	0,0497	0,0522	0,0546	0,0568	0,0588	0,0608
18	0,0283	0,0318	0,0352	0,0383	0,0412	0,0439	0,0465	0,0489	0,0511	0,0532
19	0,0188	0,0227	0,0263	0,0296	0,0328	0,0357	0,0385	0,0411	0,0436	0,0459
20	0,0094	0,0136	0,0175	0,0211	0,0245	0,0277	0,0307	0,0335	0,0361	0,0386
21	0,0000	0,0045	0,0087	0,0126	0,0163	0,0197	0,0229	0,0259	0,0288	0,0314
22			0,0000	0,0042	0,0081	0,0118	0,0153	0,0185	0,0215	0,0244
23					0,0000	0,0039	0,0076	0,0111	0,0143	0,0174
24							0,0000	0,0037	0,0071	0,0104
25									0,0000	0,0350



Tabela 17 – Valores Críticos  $p$ 

N	Nível de significância ( $p$ )								
	0,01	0,02	0,05	0,1	0,5	0,9	0,95	0,98	0,99
3	0,753	0,756	0,767	0,789	0,959	0,998	0,999	1,000	1,000
4	0,687	0,707	0,748	0,792	0,935	0,987	0,992	0,996	0,997
5	0,686	0,715	0,762	0,806	0,927	0,979	0,986	0,991	0,993
6	0,713	0,743	0,788	0,826	0,927	0,974	0,981	0,986	0,989
7	0,730	0,760	0,803	0,838	0,928	0,972	0,979	0,985	0,988
8	0,749	0,778	0,818	0,851	0,932	0,972	0,978	0,984	0,987
9	0,764	0,791	0,829	0,859	0,935	0,972	0,978	0,984	0,986
10	0,781	0,806	0,842	0,869	0,938	0,972	0,978	0,983	0,986
11	0,792	0,817	0,850	0,876	0,940	0,973	0,979	0,984	0,986
12	0,805	0,828	0,859	0,883	0,943	0,973	0,979	0,984	0,986
13	0,814	0,837	0,866	0,889	0,945	0,974	0,979	0,984	0,986
14	0,825	0,846	0,874	0,895	0,947	0,975	0,980	0,984	0,986
15	0,835	0,855	0,881	0,901	0,950	0,975	0,980	0,984	0,987
16	0,844	0,863	0,887	0,906	0,952	0,976	0,981	0,985	0,987
17	0,851	0,869	0,892	0,910	0,954	0,977	0,981	0,985	0,987
18	0,858	0,874	0,897	0,914	0,956	0,978	0,982	0,986	0,988
19	0,863	0,879	0,901	0,917	0,957	0,978	0,982	0,986	0,988
20	0,868	0,884	0,905	0,920	0,959	0,979	0,983	0,986	0,988
21	0,873	0,888	0,908	0,923	0,960	0,980	0,983	0,987	0,989
22	0,878	0,892	0,911	0,926	0,961	0,980	0,984	0,987	0,989
23	0,881	0,895	0,914	0,928	0,962	0,981	0,984	0,987	0,989
24	0,884	0,898	0,916	0,930	0,963	0,981	0,984	0,987	0,989
25	0,888	0,901	0,918	0,931	0,964	0,981	0,985	0,988	0,989
26	0,891	0,904	0,920	0,933	0,965	0,982	0,985	0,988	0,989
27	0,894	0,906	0,923	0,935	0,965	0,982	0,985	0,988	0,990
28	0,896	0,908	0,924	0,936	0,966	0,982	0,985	0,988	0,990
29	0,898	0,910	0,926	0,937	0,966	0,982	0,985	0,988	0,990
30	0,900	0,912	0,927	0,939	0,967	0,983	0,985	0,988	0,990
31	0,902	0,914	0,929	0,940	0,967	0,983	0,986	0,988	0,990
32	0,904	0,915	0,930	0,941	0,968	0,983	0,986	0,988	0,990
33	0,906	0,917	0,931	0,942	0,968	0,983	0,986	0,989	0,990
34	0,908	0,919	0,933	0,943	0,969	0,983	0,986	0,989	0,990
35	0,910	0,920	0,934	0,944	0,969	0,984	0,986	0,989	0,990
36	0,912	0,922	0,935	0,945	0,970	0,984	0,986	0,989	0,990
37	0,914	0,924	0,936	0,946	0,970	0,984	0,987	0,989	0,990
38	0,916	0,925	0,938	0,947	0,971	0,984	0,987	0,989	0,990
39	0,917	0,927	0,939	0,948	0,971	0,984	0,987	0,989	0,991
40	0,919	0,928	0,940	0,949	0,972	0,985	0,987	0,989	0,991
41	0,920	0,929	0,941	0,950	0,972	0,985	0,987	0,989	0,991
42	0,922	0,930	0,942	0,951	0,972	0,985	0,987	0,989	0,991
43	0,923	0,932	0,943	0,951	0,973	0,985	0,987	0,990	0,991
44	0,924	0,933	0,944	0,952	0,973	0,985	0,987	0,990	0,991
45	0,926	0,934	0,945	0,953	0,973	0,985	0,988	0,990	0,991
46	0,927	0,935	0,945	0,953	0,974	0,985	0,988	0,990	0,991
47	0,928	0,936	0,946	0,954	0,974	0,985	0,988	0,990	0,991
48	0,929	0,937	0,947	0,954	0,974	0,985	0,988	0,990	0,991
49	0,929	0,938	0,947	0,955	0,974	0,985	0,988	0,990	0,991
50	0,930	0,939	0,947	0,955	0,974	0,985	0,988	0,990	0,991

## REFERÊNCIAS

- [Arduino, ] Arduino. Introduction.
- [Brookner, 1998] Brookner, E. (1998). *Frontmatter and Index*. Wiley Online Library.
- [CARVALHO, 2009] CARVALHO, A. (2009). S1, sd. *Redes Neurais Artificiais*. Disponível em <http://www.icmc.usp.br/~andre/research/neural>. Acesso em, 29.
- [Carvalho et al., 2016] Carvalho, D., Ferreira, D., and Gomes, F. (2016). Desenvolvimento de uma plataforma educacional baseada em pjl para ensino de controle. In *Congresso Brasileiro de Educação em Engenharia*.
- [Censo, 2010] Censo, I. (2010). Instituto brasileiro de geografia e estatística. *Fornecido em meio eletrônico:[www.ibge.gov.br/home/estatistica/populacao/censo2010/] Acessado em 20-02-2017*, 20(03):2012.
- [Ceres et al., 2005] Ceres, R., Pons, J. L., Calderon, L., Jimenez, A. R., and Azevedo, L. (2005). A robotic vehicle for disabled children. *IEEE Engineering in Medicine and Biology magazine*, 24(6):55–63.
- [de Lima, ] de Lima, A. K. P. Fisioterapia na lesão medular.
- [Duan et al., 2014] Duan, J., Li, Z., Yang, C., and Xu, P. (2014). Shared control of a brain-actuated intelligent wheelchair. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pages 341–346. IEEE.
- [Emotiv, ] Emotiv. The emotiv epoc interface.
- [fairchildsemi, ] fairchildsemi. Mems inertial measurement unit (imu).
- [Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.
- [Grewal, 2011] Grewal, M. S. (2011). *Kalman filtering*. Springer.
- [Helene, 2006] Helene, O. (2006). *Método dos Mínimos Quadrados*. Editora Livraria da Física.
- [Hu et al., 2010] Hu, Z.-f., Li, L., Luo, Y., Zhang, Y., and Wei, X. (2010). A novel intelligent wheelchair control approach based on head gesture recognition. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 6, pages V6–159. IEEE.
- [Jayawardena et al., 2013] Jayawardena, C., Baghaei, N., Ganeshan, K., and Sarrafzadeh, A. (2013). Designing a socially assistive companion robotic wheel chair: Robochair. In *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 231–236. IEEE.
- [Kalman et al., 1960] Kalman, R. E. et al. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

- [Kim, 2011] Kim, P. (2011). *Kalman filter for beginners: with MATLAB examples*. CreateSpace.
- [Koenig and Howard, 2013] Koenig, N. and Howard, A. (2013). Gazebo-3d multiple robot simulator with dynamics (2003). *URL: <http://gazebo.org>*, 3.
- [Koren and Borenstein, 1991] Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398–1404. IEEE.
- [Kovács, 2002] Kovács, Z. L. (2002). *Redes neurais artificiais*. Editora Livraria da Física.
- [Kuno et al., 1999] Kuno, Y., Nakanishi, S., Murashima, T., Shimada, N., and Shirai, Y. (1999). Robotic wheelchair with three control modes. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 2590–2595. IEEE.
- [Kuo et al., 2006] Kuo, C.-H. et al. (2006). Human-oriented design of autonomous navigation assisted robotic wheelchair for indoor environments. In *2006 IEEE International Conference on Mechatronics*, pages 230–235. IEEE.
- [Lauszus, 2012] Lauszus, K. (2012). A practical approach to kalman filter and how to implement it. *Guides, TKJ Electronics, September*.
- [Li et al., 2016] Li, Z., Zhao, S., Su, C.-Y., Yang, C., Duan, J., and Zhao, X. (2016). Human cooperative wheelchair with brain machine interaction based on shared control strategy. *IEEE/ASME Transactions on Mechatronics*.
- [Liberal, ] Liberal, T. Covariância e coeficiente de correlação - probabilidade ii.
- [Lobato et al., 2017] Lobato, S., Carvalho, D., Ferreira, A., and Montagner, V. (2017). Didactic prototype to model and to design linear control applied to a rlc plant. In *Brazilian Power Electronics Conference*. IEEE.
- [Lund et al., 2010] Lund, M. E., Christensen, H. V., Caltenco, H. A., Lontis, E. R., Bentsen, B., and Struijk, L. N. A. (2010). Inductive tongue control of powered wheelchairs. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 3361–3364. IEEE.
- [Maciel, 2016] Maciel, G. M. (2016). Compensação de posicionamento angular de um manipulador robótico sobre uma base com três graus de liberdade.
- [Marins et al., 2017] Marins, G., Carvalho, D., Marcato, A., and Júnior, I. (2017). Development of a control system for electric wheelchairs based on head movements. In *Intelligent Systems Conference*. IEEE.
- [Microelectronics, ] Microelectronics, S. An4508 application note, parameters and calibration of a low-g 3-axis accelerometer.
- [Mitchell et al., 2014] Mitchell, I. M., Viswanathan, P., Adhikari, B., Rothfels, E., and Mackworth, A. K. (2014). Shared control policies for safe wheelchair navigation of elderly adults with cognitive and mobility impairments: Designing a wizard of oz study. In *2014 American Control Conference*, pages 4087–4094. IEEE.

- [MobileRobots, ] MobileRobots, A. Pioneer 3-dx.
- [Morrison, 1987] Morrison, M. (1987). Inertial measurement unit. US Patent 4,711,125.
- [Nguyen et al., 2016] Nguyen, T. N., Nguyen, H. T., et al. (2016). A telepresence wheelchair using cellular network infrastructure in outdoor environments. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 5352–5355. IEEE.
- [Nogueira, 2013] Nogueira, M. M. (2013). Aplicando lógica fuzzy no controle de robôs móveis usando dispositivos lógicos programáveis e a linguagem vhdl.
- [of Standards and Technology, ] of Standards, N. I. and Technology. Euclidian distance.
- [Olivi et al., 2013] Olivi, L., Souza, R. S., Rohmer, E., and Cardozo, E. (2013). Shared control for assistive mobile robots based on vector fields. In *URAI*, pages 96–101.
- [Organization and Society, 2013] Organization, W. H. and Society, I. S. C. (2013). *International perspectives on spinal cord injury*. World Health Organization.
- [PUC-Rio, 2014] PUC-Rio (2014). Redes neurais artificiais. *Pontifícia Universidade Católica do Rio de Janeiro*.
- [Raizer et al., 2013] Raizer, K., Rohmer, E., Paraense, A. L., and Gudwin, R. R. (2013). Effects of behavior network as a suggestion system to assist bci users. In *Computational intelligence in rehabilitation and assistive technologies (cirat), 2013 ieee symposium on*, pages 40–47. IEEE.
- [Rauber, 2005] Rauber, T. W. (2005). Redes neurais artificiais. *Universidade Federal do Espírito Santo*.
- [Röfer and Lankenau, 2000] Röfer, T. and Lankenau, A. (2000). Architecture and applications of the bremen autonomous wheelchair. *Information Sciences*, 126(1):1–20.
- [Rohmer et al., 2015a] Rohmer, E., Pinheiro, P., Cardozo, E., Bellone, M., and Reina, G. (2015a). Laser based driving assistance for smart robotic wheelchairs. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE.
- [Rohmer et al., 2015b] Rohmer, E., Pinheiro, P., Raizer, K., Olivi, L., and Cardozo, E. (2015b). A novel platform supporting multiple control strategies for assistive robots. In *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*, pages 763–769. IEEE.
- [ROS, ] ROS. About ros.
- [Salhi et al., 2016] Salhi, K., Alimi, A. M., Gorce, P., and Khelifa, M. M. B. (2016). Navigation assistance to disabled persons with powered wheelchairs using tracking system and cloud computing technologies. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pages 1–6. IEEE.
- [SANDI et al., 1998] SANDI, F. A., Hemerly, E. M., and Lages, W. F. (1998). Sistema para navegação e guiagem de robôs móveis autônomos. *SBA Controle e Automação*.

- [Sato et al., 2014] Sato, M., Kaneko, J., and Kojima, K. (2014). Development of pavement surface inspection system for wheel chair comfortability. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pages 219–220. IEEE.
- [Shapiro and Wilk, 1965] Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.
- [Sheikh and Rotake, 2015] Sheikh, S. A. M. S. and Rotake, D. R. (2015). An evolutionary approach for smart wheelchair system. In *Communications and Signal Processing (ICCSP), 2015 International Conference on*, pages 1811–1815. IEEE.
- [Siciliano and Khatib, 2016] Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer.
- [Soriano et al., 2013] Soriano, D., Silva, E. L., Slenes, G. F., Lima, F. O., Uribe, L. F., Coelho, G. P., Rohmer, E., Venâncio, T. D., Beltramini, G. C., Campos, B. M., et al. (2013). Music versus motor imagery for bci systems a study using fmri and eeg: Preliminary results. In *Biosignals and Birobotics Conference (BRC), 2013 ISSNIP*, pages 1–6. IEEE.
- [Tajmar et al., 2009] Tajmar, M., Plesescu, F., and Seifert, B. (2009). Anomalous fiber optic gyroscope signals observed above spinning rings at low temperature. In *Journal of Physics: Conference Series*, volume 150, page 032101. IOP Publishing.
- [Theodoridis et al., 2010] Theodoridis, S., Pikrakis, A., Koutroumbas, K., and Cavouras, D. (2010). *Introduction to pattern recognition: a matlab approach*. Academic Press.
- [Tonin et al., 2010] Tonin, L., Leeb, R., Tavella, M., Perdakis, S., and Millán, J. d. R. (2010). The role of shared-control in bci-based telepresence. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 1462–1466. IEEE.
- [Ventura, 2013] Ventura, A. R. C. (2013). Método dos mínimos quadrados lineares. pages 4–5.
- [Verrill and Johnson, 1988] Verrill, S. and Johnson, R. A. (1988). Tables and large-sample distribution theory for censored-data correlation statistics for testing normality. *Journal of the American Statistical Association*, 83(404):1192–1197.
- [Washington and University, ] Washington and University, L. The extended kalman filter: An interactive tutorial for non-experts.
- [Wilson, 2004] Wilson, J. S. (2004). *Sensor technology handbook*. Elsevier.
- [xsens, ] xsens. Timu inertial measurement unit.
- [Young, ] Young, W. Spinal cord injury levels and classification.
- [Zhang et al., 2014] Zhang, J., Wang, J., and Chen, W. (2014). A control system of driver assistance and human following for smart wheelchair. In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pages 1927–1932. IEEE.