

UNIVERSIDADE DE CAXIAS DO SUL – UCS
CAMPUS UNIVERSITÁRIO DA REGIÃO DOS VINHEDOS – CARVI
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIA
CURSO DE ENGENHARIA ELÉTRICA

LUCAS VINICIUS COELHO VIEIRA

NAVEGAÇÃO AUTÔNOMA DE CADEIRA DE RODAS MOTORIZADA EM
AMBIENTE PREVIAMENTE CONHECIDO

BENTO GONÇALVES

2019

LUCAS VINICIUS COELHO VIEIRA

**NAVEGAÇÃO AUTÔNOMA DE CADEIRA DE RODAS MOTORIZADA EM
AMBIENTE PREVIAMENTE CONHECIDO**

Trabalho de Conclusão de Curso II
apresentado no Campus Universitário da
Região dos Vinhedos, da Universidade de
Caxias do Sul, como requisito parcial para
obtenção do título de Engenheiro
Eletricista.

Orientador:

Prof. Me. Patric Janner Marques

BENTO GONÇALVES

2019

LUCAS VINICIUS COELHO VIEIRA

**NAVEGAÇÃO AUTÔNOMA DE CADEIRA DE RODAS MOTORIZADA EM
AMBIENTE PREVIAMENTE CONHECIDO**

Trabalho de Conclusão de Curso II
apresentado no Campus Universitário da
Região dos Vinhedos, da Universidade de
Caxias do Sul, como requisito parcial para
obtenção do título de Engenheiro
Eletricista.

Orientador:

Prof. Me. Patric Janner Marques

Aprovado em __/__/__

Banca Examinadora:

Prof. Me. Patric Janner Marques
Universidade de Caxias do Sul – UCS

Prof. Dra. Marilda Machado Spindola
Universidade de Caxias do Sul – UCS

Prof. Me. Felipe Augusto Tondo
Universidade de Caxias do Sul – UCS

AGRADECIMENTOS

Agradeço aos meus pais, Fabiana e Eduardo, que não mediram esforços para me ajudar nessa etapa tão importante da minha vida.

Ao meu irmão Lúcio, pelo todo o companheirismo e momentos de descontração durante esta jornada.

A minha Vó Terezinha, pelos sábios conselhos em momentos importantes.

A minha namorada Laura pelo apoio, compreensão e carinho dando suporte nos momentos mais difíceis.

Aos meus amigos e colegas, em especial ao Vinicius e ao Jacson que me incentivaram todos os dias e ofereceram apoio nos momentos críticos.

Ao Prof. Patric Janner Marques pelas orientações, pelo conhecimento compartilhado e atenção prestada às minhas dúvidas.

Aos funcionários do laboratório Andrei e Francisco, sempre atentos e dispostos a me auxiliar.

*“O que precisamos é de mais pessoas
especializadas no impossível”*

Theodore Roethke

RESUMO

Sistemas autônomos consistem em uma sequência de regras que tem por objetivo a tomada de decisões e a realização de tarefas sem a necessidade de supervisão humana em tempo integral. Cadeira de rodas autônoma é uma tecnologia assistiva que facilita a mobilidade de pessoas com deficiências motoras de locomoção. Com intuito de auxiliar a mobilidade dessas pessoas, foi desenvolvido neste trabalho um sistema embarcado inteligente para a navegação autônoma de uma cadeira de rodas motorizada em um ambiente conhecido. A eletrônica embarcada é constituída de dois microcontroladores, um Raspberry, responsável pelo controle do sistema e o monitoramento dos sensores, e de um ESP32 que controla o acionamento dos motores para a correta locomoção, por meio da lógica de controle fuzzy. Além destes equipamentos o sistema também é constituído por sensores do tipo *encoder* e um sensor de cores, aplicados na geração de dados para a técnica de navegação denominada odometria e reconhecimento de marcos artificiais através de pisos táteis, auxiliando na determinação do posicionamento absoluto da cadeira de rodas. O mapa virtual do local é gerado por intermédio da técnica de mapa discreto e carregado no banco de dados do sistema embarcado, para então um algoritmo de geração de trajetórias ser implementado na elaboração de um caminho possível entre um ponto inicial e um destino escolhido.

Palavras-chave: Cadeira de rodas autônoma, tecnologia assistiva, eletrônica embarcada, odometria, marcos artificiais.

ABSTRACT

Autonomous systems consist in a sequence of rules designed to make decisions and perform tasks without the need for full-time human supervision. Autonomous wheelchair is an assistive technology that facilitates the mobility of people with mobility impairments. In order to assist the mobility of these people, an intelligent embedded system was developed for the autonomous navigation of a motorized wheelchair in a known environment. The onboard electronics will consist of two microcontrollers, a Raspberry that will be responsible for all system control and monitoring of the sensors, and an ESP32 that will drive the motors for correct locomotion through fuzzy control logic. In addition to these devices, the system also consists of encoder sensors and a color sensor applied in the generation of data for the navigation technique called odometry and recognition of artificial landmarks through tactile floors, helping to determine the absolute positioning of the wheelchair. The virtual site map is generated using the discrete map technique and loaded into the embedded system database, so that a trajectory algorithm is implemented to devise a possible path between a starting point and a chosen destination.

Keywords: Autonomous wheelchair, assistive technology, embedded electronics, odometry, artificial landmarks.

LISTA DE FIGURAS

Figura 1 – Modelo cinemático do tipo <i>Hilare-type</i>	21
Figura 2 – Blocos funcionais para a navegação.....	22
Figura 3 – Geometria de uma curva à esquerda por um robô de duas rodas	24
Figura 4 – Mapas discretos e contínuos.....	25
Figura 5 – Mapa discreto, com ocupação de elemento.....	26
Figura 6 – Mapa de Grade e possível trajeto entre os pontos S e G.....	27
Figura 7 – Mapa após aplicado o algoritmo do custo de movimento.....	27
Figura 8 – Giroscópio óptico.....	30
Figura 9 – Sensor do tipo <i>encoder</i> óptico.....	31
Figura 10 – Determinação do grau de certeza.	33
Figura 11 – Determinação da velocidade de resposta de um sistema Fuzzy.	34
Figura 12 – Áreas trapezoidais determinadas em uma configuração Fuzzy.	34
Figura 13 – Sensores aplicados no andador.	37
Figura 14 – Celular adaptado na roda da bicicleta.	37
Figura 15 – Técnica aplicada para redução na quantidade de dados.	39
Figura 16 – Software desenvolvido.	41
Figura 17 – Percurso proposto.	42
Figura 18 – Funcionamento do sistema.	44
Figura 19 – Esquemático dos componentes da cadeira.....	45
Figura 20 – Detalhes construtivos do protótipo desenvolvido.	46
Figura 21 – Defasagem entre canais A e B.....	47
Figura 22 – Condicionamento dos <i>encoders</i>	48
Figura 23 – Marco artificial implementado.....	49
Figura 24 – Supervisório do sistema.	49
Figura 25 – Chave seletora para troca de modo de deslocamento.	50
Figura 26 – Comandos utilizados para o controle fuzzy.	51
Figura 27 – Trajeto gerado até sala 306B.	68

LISTA DE TABELAS

Tabela 1 – Categorização de sensores	29
Tabela 2 – Trabalhos práticos abordados e técnicas utilizadas	36
Tabela 3 – Performance para seleção de trajetória.....	40
Tabela 4 – Performance para parada do sistema	40
Tabela 5 – Níveis de tensão fornecidos pelo joystick.	51
Tabela 6 – Regras fuzzy para movimento retilíneo.	52
Tabela 7 – Regras fuzzy para movimento retilíneo.	52
Tabela 8 – Regras <i>fuzzy</i> para movimento retilíneo com compensação de velocidade angular.	53
Tabela 9 – Regras <i>fuzzy</i> para movimento angular.	54
Tabela 10 – Regras <i>fuzzy</i> para ajuste do movimento angular.....	54
Tabela 11 – Regras <i>fuzzy</i> para movimento em curva.	55
Tabela 12 – Resultado dos testes de leitura digital.	58
Tabela 13 – Resultado dos testes algoritmo Dijkstra	58
Tabela 14 – Resultado dos testes feitos no ESP32.	59
Tabela 15 – Verificação de erro do sistema odométrico.	62
Tabela 16 – Reconhecimento de marcos artificiais.	67

LISTA DE GRÁFICOS

Gráfico 1 – Atuador controlador <i>fuzzy</i> para o comando frente.	60
Gráfico 2 – Variação de deslocamento entre as rodas.....	60
Gráfico 3 – Variação de deslocamento com compensações utilizadas.	61
Gráfico 4 – Deslocamento real para cada caso.....	62
Gráfico 5 – Comando fuzzy necessários para o movimento em 90 graus sob o eixo.	63
Gráfico 6 – Rotação sob o eixo para a esquerda.	64
Gráfico 7 – Rotação sob o eixo para a direita.	64
Gráfico 8 – Rotação sobre o eixo para a direita.	65
Gráfico 9 – Curva 90 graus para esquerda.	66
Gráfico 10 – Curva 90 graus para direita.	66

LISTA DE ABREVIATURAS

GPS	<i>Global Position System</i>
RF	<i>Radio Frequency</i>
CCD	<i>Charged-Couple Device</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
LED	<i>Light Emitting Diode</i>
SNI	<i>Sistema de Navegação Inercial</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
RFID	<i>Radio Frequency Identification</i>
BCI	<i>Brain-Computer Interface</i>
SSVEP	<i>Steady State Visually Evoked Potential</i>
RGB	<i>Red Green Blue</i>
DAC	<i>Digital-to-Analog converter</i>
IHM	<i>Interface human-machine</i>
SO	<i>Sistema Operacional</i>
PCB	<i>Placa de Circuito Impresso</i>
RAM	<i>Random Access Memory</i>
HDMI	<i>Interface Multimídia de Alta Resolução</i>

LISTA DE SÍMBOLOS

δ_m	<i>Grau de manobrabilidade (-)</i>
δ_c	<i>Grau de controlabilidade (-)</i>
δ_s	<i>Estabilidade dinâmica (-)</i>
x_1	<i>Eixo horizontal (m)</i>
x_2	<i>Eixo vertical (m)</i>
x_{r1}	<i>Eixo inercial</i>
θ	<i>Posição angular (°)</i>
q	<i>Orientação</i>
v	<i>Velocidade linear (m/s)</i>
ω	<i>Velocidade angular da roda(rad/s)</i>
u	<i>Vetor de posição (-)</i>
F_{ap}	<i>Força externa (N)</i>
F_i	<i>Força inercial (N)</i>
m	<i>Massa (Kg)</i>
F_{am}	<i>Força de amortecimento (N)</i>
c	<i>Coeficiente de amortecimento (-)</i>
F_m	<i>Força da mola (N)</i>
k	<i>Constante da mola (N/m)</i>
x	<i>Posição de equilíbrio (m)</i>
a	<i>Aceleração (m/s²)</i>
s	<i>Deslocamento linear da roda (m)</i>
n_T	<i>Número de pulsos por revolução (-)</i>
n	<i>Número total de pulsos (-)</i>
C	<i>Comprimento de circunferência da roda (m)</i>
r	<i>Raio da roda (m)</i>
t	<i>Tempo (s)</i>
rd	<i>Roda direita</i>
re	<i>Roda esquerda</i>
b	<i>Distância entre as rodas (m)</i>
R	<i>Raio do deslocamento angular (m)</i>
ω_i	<i>Velocidade angular de deslocamento (rad/s)</i>

d_i	<i>Deslocamento da roda (m)</i>
d_c	<i>Deslocamento do centro da cadeira (m)</i>
S	<i>Ponto de partida</i>
G	<i>Ponto de chegada</i>
C_M	<i>Custo de movimento (-)</i>
C_A	<i>Custo anterior (-)</i>
K	<i>Custo de novo movimento (-)</i>
J	<i>Leitura do sensor</i>
G_c	<i>Grau de certeza (-)</i>
G_{ca}	<i>Grau de certeza a</i>
G_{cb}	<i>Grau de certeza b</i>
w	<i>Base do triângulo de atuação (-)</i>
h	<i>Altura do triângulo de atuação (-)</i>
A_T	<i>Área trapezoidal (-)</i>
S_c	<i>Valor central de atuação (-)</i>
S_a	<i>Valor de atuação no motor (-)</i>

SUMÁRIO

1 INTRODUÇÃO	17
1.1 OBJETIVO GERAL	17
1.2 OBJETIVOS ESPECÍFICOS.....	18
1.3 ESCOPO E RESTRIÇÕES	18
1.4 APRESENTAÇÃO DO TRABALHO.....	19
 2 REVISÃO BIBLIOGRÁFICA	20
2.1 MODELO DE CINEMÁTICA DE CADEIRA DE RODAS	20
2.1.1 Definição de parâmetros.....	20
2.1.2 Modelagem cinemática	21
2.2 NAVEGAÇÃO AUTÔNOMA.....	22
2.2.1 Odometria	23
2.2.2 Marcos artificiais para a navegação	25
2.2.3 Mapa virtual do local.....	25
2.2.4 Geração da trajetória.....	26
2.3 SENSORES	28
2.3.1 Categorias de sensores	28
2.3.2 Sensores utilizados em sistemas de navegação autônoma.....	29
2.3.2.1 Giroscópio óptico.....	30
2.3.2.2 Acelerômetro	30
2.3.2.3 Encoder óptico	31
2.3.2.4 Sensor de espectro	32
2.4 SISTEMA DE CONTROLE BASEADO EM LÓGICA FUZZY.....	32
 3 ABORDAGENS DA LITERATURA	36
 4 DESENVOLVIMENTO DO TRABALHO	43
4.1 MONTAGEM DO HARDWARE.....	44
4.2 INTERLIGAÇÃO DOS EQUIPAMENTOS.....	46
4.2.1 Comunicação Raspberry com ESP32.....	46
4.2.2 Comunicação Raspberry com encoder	47
4.2.3 Comunicação Raspberry com sensor de espectro luminoso	48

4.2.4 Comunicação Raspberry com sistema supervisório	49
4.2.5 Comunicação ESP32 com o driver da cadeira.....	50
4.3 CONTROLADOR FUZZY.....	51
4.3.1.1 Movimento Linear	52
4.3.1.2 Movimento angular sob o eixo.....	53
4.3.1.3 Movimento angular em curva	54
4.4 METODOLOGIA EXPERIMENTAL.....	55
4.4.1 Teste de microcontroladores	55
4.4.2 Testes controlador fuzzy	56
4.4.2.1 Aquisição de dados para análise de deslocamento	56
4.4.2.2 Teste de deslocamento linear	56
4.4.2.3 Teste de deslocamento angular	57
4.4.2.4 Teste de navegação autônoma.....	57
4.4.3 Teste de reconhecimento de marcos artificiais.....	57
 5 RESULTADOS E DISCUSSÕES	 58
5.1 RESPOSTA DOS MICROCONTROLADORES	58
5.2 DESLOCAMENTO LINEAR	59
5.3 DESLOCAMENTO ANGULAR.....	63
5.3.1 Deslocamento angular 90 graus sob o eixo.....	63
5.3.2 Deslocamento angular 90 graus em curva.....	65
5.4 RECONHECIMENTO DE MARCOS ARTIFICIAIS	66
5.5 NAVEGAÇÃO AUTÔNOMA ATRAVÉS DA ODOMETRIA	67
 6 CONCLUSÕES	 69
 REFERÊNCIAS BIBLIOGRÁFICAS	 70
 ANEXOS	 72
ANEXO A – CARACTERÍSTICAS DA TRENA A LASER GLM 40 BOSCH	72
 APÊNDICES	 73
APÊNDICE A – PCB DESENVOLVIDA PARA INTERLIGAÇÃO DOS EQUIPAMENTOS	73
APÊNDICE B – TESTES REALIZADOS NA RASPBERRY PI 3 B+	74

APÊNDICE C – TESTES REALIZADOS NO ESP32	75
---	----

1 INTRODUÇÃO

O suporte à deficientes físicos é um tema que tem crescido no âmbito das pesquisas em tecnologias assistivas nos últimos anos. Um dos fatores que impulsionam esse crescimento é a grande diversidade de funções que essas tecnologias podem desempenhar. Uma das linhas de pesquisa em destaque visa o desenvolvimento de sistemas que tem como propósito dar apoio, ou mesmo eliminar os efeitos da limitação física do paciente (ENCARNAÇÃO; COOK, 2017, p. 4-5).

Em conjunto com tecnologias assistivas, sistemas autônomos são cada vez mais demandados no âmbito de desenvolvimento, pois este tipo de sistema, pode controlar um processo constantemente, sem que seu desempenho seja afetado por uma falta de supervisão humana em tempo integral (SIEGWART, 2011, p. 1-3). Do mesmo modo, devido a precisão que um sistema de controle consegue obter sobre um equipamento, pode ser utilizado para o desenvolvimento de um dispositivo que supra alguma deficiência específica de um indivíduo, fazendo com que esta pessoa tenha sua qualidade de vida melhorada (ENCARNAÇÃO; COOK, 2017, p. 56-58).

Diante do contexto apresentado, este trabalho propõe o desenvolvimento de um sistema inteligente para a navegação autônoma de uma cadeira de rodas motorizada em um ambiente conhecido, através de uma eletrônica embarcada. Um sistema de geração de trajetória automática deverá ser implementado para a elaboração de uma rota possível entre o ponto inicial e o destino escolhido pelo usuário. O mapa do local deverá ser previamente carregado no banco de dados do sistema. Além disso, a cadeira possuirá um sistema de navegação denominado odometria, com a função de realizar o controle da posição em relação a trajetória a ser seguida, tendo em vista o cumprimento da rota proposta.

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo o desenvolvimento de um sistema para a navegação autônoma de uma cadeira de rodas motorizada em ambiente conhecido, fazendo uso das técnicas de odometria, navegação inercial e atualização de posição através de marcos artificiais de referência.

1.2 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral, destacam-se os seguintes objetivos específicos:

- a) definir o dispositivo eletrônico adequado às necessidades de aquisição de informações do ambiente e processamento de dados para a navegação da cadeira;
- b) modelar a dinâmica da cadeira de rodas, bem como o comportamento característico dos motores nos diversos acionamentos possíveis;
- c) desenvolver um algoritmo capaz de gerar trajetórias automaticamente em um mapa previamente conhecido, assim um sistema de controle para adotar o caminho determinado por meio da odometria e marcos artificiais;
- d) realizar ensaios práticos a fim de avaliar o comportamento do sistema em condições reais de aplicação.

1.3 ESCOPO E RESTRIÇÕES

As seguintes restrições são aplicadas ao presente trabalho com a finalidade de restringir-se aos objetivos já discutidos:

- a) o sistema eletrônico desenvolvido se restringirá a sua aplicação no sistema mecânico escolhido;
- b) o mapa local previamente escolhido, está inserido na base de dados do sistema embarcado utilizado;
- c) o sistema a ser desenvolvido se limita a gerar os comandos de navegação de uma cadeira de rodas motorizada por um ambiente conhecido sem considerar possíveis obstáculos encontrados pela trajetória que não são identificáveis no mapa inserido na base de dados do sistema.

1.4 APRESENTAÇÃO DO TRABALHO

Este trabalho se divide em 4 capítulos, sendo que no primeiro capítulo é apresentada uma breve introdução. O segundo capítulo consiste na revisão bibliográfica necessária para a compreensão da proposta deste trabalho, onde são apresentadas as características construtivas, modelos equivalentes, parâmetros essenciais e técnicas de navegação, além dos métodos utilizados para processar as grandezas físicas provenientes dos sensores. No terceiro capítulo é apresentada a análise do estado da arte, evidenciando as principais técnicas para a navegação autônoma. Por fim, o capítulo 4 apresenta uma abordagem quanto a metodologia adotada para a realização da proposta de desenvolvimento deste trabalho, detalhando sobre a escolha dos componentes e técnicas utilizadas para implementação do algoritmo responsável pelo controle e navegação da cadeira de rodas motorizada.

2 REVISÃO BIBLIOGRÁFICA

Com a finalidade de fundamentar a metodologia a ser utilizada no decorrer do desenvolvimento deste estudo, a seguir serão apresentados assuntos considerados necessários para a sua compreensão e elaboração.

2.1 MODELO DE CINEMÁTICA DE CADEIRA DE RODAS

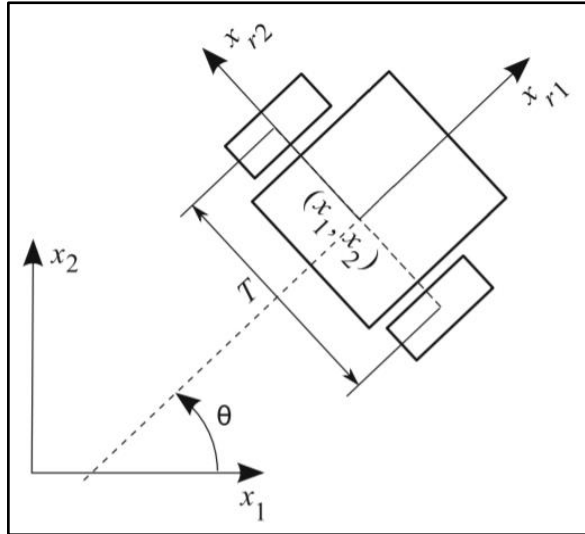
Para a escolha do sistema de rodas mais adequado para um robô móvel, Sicilliano (2008, p. 395-398) desenvolveu uma classificação a respeito de sua mobilidade. Para isto, define duas variáveis, o grau de manobrabilidade (δ_m – *maneuverability*), e o grau de controlabilidade (δ_c – *controllability*). Além destes parâmetros, Siegwart (2011, p. 33) também faz a análise da estabilidade (δ_s – *stability*) da dinâmica do sistema.

A construção do robô determinará o grau destas variáveis, que leva em conta o posicionamento, a quantidade e o tipo de rodas aplicadas. Para uma topologia do tipo de cadeira de rodas, Sicilliano (2008, p. 800) classifica a respeito da manobrabilidade como grau 2, por possuir uma ou várias rodas fixas em eixos independentes. Já referente a controlabilidade classifica como grau 0, pois não possui rodas de direção acopladas.

2.1.1 Definição de parâmetros

Quanto a realização da modelagem matemática da dinâmica do sistema eletrônico, a cadeira de rodas é considerada como um veículo do tipo Hilaré-Type, definida por Fahimi (2009, p. 194) como robô característico por duas rodas fixas com o controle de tração individual conforme apresentado na Figura 1.

Figura 1 – Modelo cinemático do tipo *Hilare-type*.



Fonte: Adaptado de Fahimi (2009, p. 164).

A posição e orientação da cadeira de rodas pode ser definida a partir de três variáveis, sendo elas: as duas componentes lineares de posição x_1 e x_2 em relação à origem definida no mapa virtual e a posição angular θ entre o eixo longitudinal x_1 do mapa e o eixo inercial x_{r1} posicionado horizontalmente no centro da cadeira de rodas, conforme apresentado na Figura 1, (FAHIMI, 2009 p. 166).

2.1.2 Modelagem cinemática

Conforme a Figura 1, a orientação do robô em qualquer dado tempo pode ser agrupada e denotada de acordo com a Equação 1 (FAHIMI, 2009, p. 165):

$$q = \begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix} \quad (1)$$

Considerando que os pontos x_1 e x_2 do robô se movam com uma velocidade linear v , isso permitirá que o veículo possua uma velocidade angular ω . É possível estimar os componentes da velocidade com que o robô se desloca na trajetória conforme as Equações 2 e 3.

$$\dot{x}_1 = v * \cos \theta \quad (2)$$

$$\dot{x}_2 = v * \sin \theta \quad (3)$$

$$\dot{\theta} = \omega \quad (4)$$

O movimento cinemático do robô pode ser descrito de forma matricial de ordem 3x2 por meio da combinação das Equações 2, 3 e 4, conforme descrito na Equação 5.

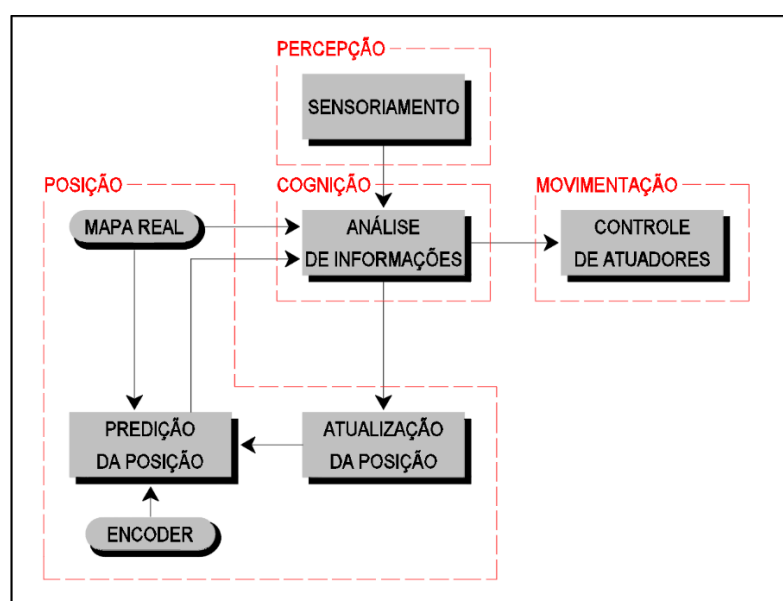
$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} u \quad (5)$$

O vetor u é comumente tratado em função do tempo, como $u = [v \ \omega]^T$, e a integração numérica de u pode descrever o movimento do robô. O movimento pode ser descrito utilizando outras variáveis alternativamente, como v e ω , desde que seja mantida a relação em função do tempo.

2.2 NAVEGAÇÃO AUTÔNOMA

A navegação é uma das competências exigidas em um sistema autônomo. O processo de navegação pode ser dividido em 4 níveis: percepção, posição, cognição e movimentação. A Figura 2 contém uma representação da navegação por meio de blocos funcionais (SIEGWART; 2011, p. 181-183).

Figura 2 – Blocos funcionais para a navegação.



Fonte: Adaptado de Siegwart (2011, p. 182).

Em cada um desses níveis, as seguintes tarefas são realizadas.

- a) percepção: neste nível, o sistema utiliza sensores para extrair dados significativos e criar um modelo contendo as principais características para a correta navegação;
- b) posição: sua principal funcionalidade é determinar a localização do objeto em relação ao destino final e em que posição está inserido dentro do mapa;
- c) cognição: com os dados fornecidos do ambiente pelo nível percepção e da localização, é verificado se a cadeira de rodas já chegou ao destino, e caso necessário são realizados ajustes na trajetória;
- d) movimentação: neste nível os atuadores são controlados de forma que o equipamento respeite a trajetória calculada no nível anterior.

2.2.1 Odometria

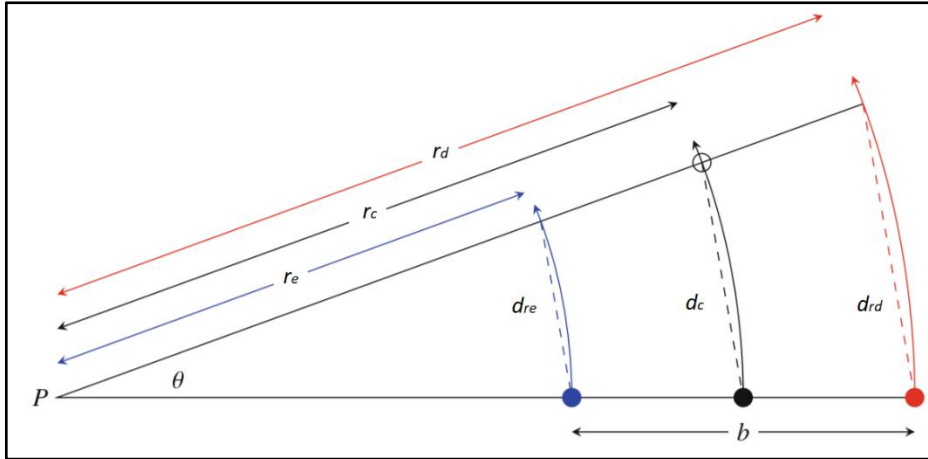
Odometria é um método de localização relativa que pode ser utilizada por sistemas autônomos, tendo como função a determinação da posição atualizada do objeto no meio por medidas de deslocamento em relação a uma base fixa (HOLLAND, 2004, p. 118).

Para Ben-Ari e Mondada (2018, p. 69), o deslocamento linear s pode ser determinado a partir de medidas de tempo t e velocidade v , conforme a Equação 6:

$$s = vt \tag{6}$$

No deslocamento de objetos construídos com duas ou mais rodas de tração independentes, a medida da distância percorrida pode ser distorcida em trajetórias não retilíneas devido a diferença de deslocamento linear em diferentes posições radiais a partir do centro da curva, como apresentado na Figura 3.

Figura 3 – Geometria de uma curva à esquerda por um robô de duas rodas



Fonte: Ben-Ari; Mondada (2018 p. 71).

De acordo com Ben-Ari e Mondada (2018, p. 71-72), em uma curva as rodas direita rd e esquerda re estão separadas por uma distância b , com raio R e velocidade angular ω_i e descrevem trajetórias com comprimentos diferentes. Para ajustar esta diferença, primeiramente a Equação 7 é utilizada para determinar o deslocamento d_i de cada roda em um determinado tempo t .

$$d_i = 2\pi R \omega_i t, \quad i = rd, re \quad (7)$$

A partir dos dados de deslocamento de cada roda é possível definir o deslocamento do centro d_c da cadeira pela Equação 8 e o ângulo θ pela Equação 9.

$$d_c = \frac{d_{rd} + d_{re}}{2} \quad (8)$$

$$\theta = \left| \frac{d_{re} - d_{rd}}{b} \right| \quad (9)$$

As medidas de tempo t podem ser adquiridas a partir do relógio interno do controlador, já para os dados de velocidade v pode ser utilizado um sensor *encoder*. Uma desvantagem desse processo é o fato de que medições ocorrem de forma indireta e estão relacionadas ao movimento das rodas, ou seja, estão propensas a erros de medições como deslizamentos. (BEN-ARI; MONDADA, 2018, p. 69-70).

2.2.2 Marcos artificiais para a navegação

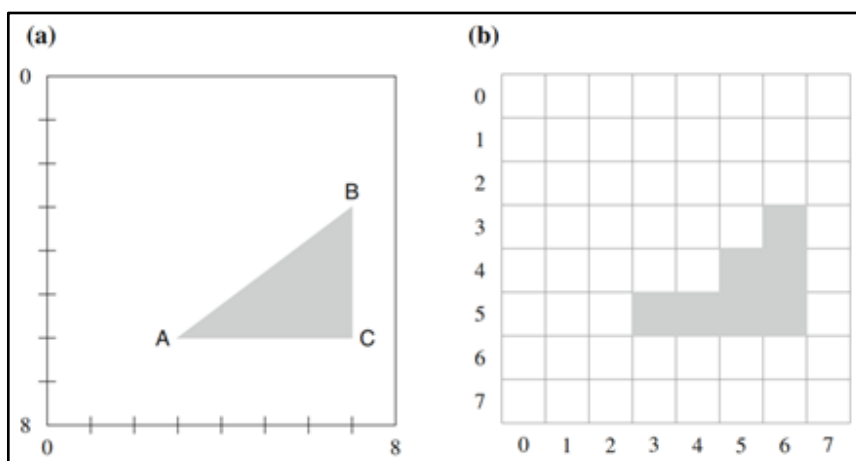
De acordo com Siegwart (2011, p. 245-246) marcos artificiais para a navegação podem ser definidos como padrões passivos e observáveis no ambiente e são utilizados para facilitar a localização de um robô imerso em um sistema de navegação e são tratados em formato de linhas ou placas.

Uma sugestão de montagem desse sistema para Borenstein (2004, p. 10) é descrita como a instalação dos marcos de forma espaçada e com sua localização precisamente conhecida. Também devem ser instalados em locais de fácil reconhecimento pelo sensor como no piso. Quando o padrão for identificado fará com que ocorra a atualização da localização dentro do mapa virtual da edificação. Este processo tem como finalidade a correção de erros propagados advindos de sistemas de navegação baseados em localização relativa.

2.2.3 Mapa virtual do local

Os mapas navegáveis por sistemas autônomos são representações virtuais armazenadas na memória do sistema. Duas técnicas são comumente utilizadas para essa virtualização de mapas: mapa discreto e mapa contínuo (BEN-ARI; MONDADA; 2018, p. 141-146). A Figura 4 mostra uma forma triangular sendo representada de forma discreta e contínua.

Figura 4 – Mapas discretos e contínuos.



Nota: o desenho (a) mapa discreto de um triângulo (b) mapa contínuo de um triângulo
 Fonte: Ben-Ari; Mondada (2018, p. 143).

Em sistemas que não apresentam geometria complexas para análise, o método do mapa contínuo, apresentado na Figura 4a, obtém melhor resultado ao definir o espaço ao ser mapeado, porém em situações com complexa geometria, este mesmo método demanda grandes quantidades de memória e processamento inviabilizando a sua utilização.

Alternativamente, o método discreto, apresentado na Figura 4b, se destaca por permitir a definição da resolução aplicada, ou seja, a quantidade de dados gerados e a necessidade de processamento podem ser controladas.

Para a denotação de ocupação espacial dentro de um mapa de grade, como pode ser visto na Figura 5, pode-se definir 1 como valor lógico denotado como grade ocupada por um objeto, ou 0 para um espaço livre para circulação.

Figura 5 – Mapa discreto, com ocupação de elemento

0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0
4	0	0	0	0	0	1	1	0
5	0	0	0	1	1	1	1	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7

Fonte: Ben-Ari; Mondada (2018, p. 144).

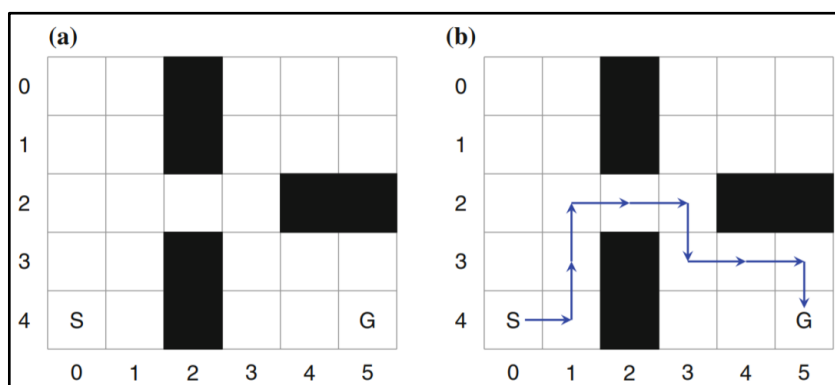
2.2.4 Geração da trajetória

A trajetória é definida como o deslocamento entre dois pontos em que regras de tempo e velocidade devem ser respeitadas enquanto cumpre o objetivo especificado (SICILLIANO; 2009, p. 162).

Para Ben-Ari e Mondada (2018, p.165-169), a geração de trajetória em mapas discretos é definida pelo algoritmo de Dijkstra. A partir de um mapa de grade, como apresentado na Figura 6a, os obstáculos são definidos e representados pelos blocos pretos e os espaços de livre circulação pelos blocos brancos. O sistema de criação de trajetória deverá reconhecer quais dos blocos estão disponíveis para deslocamento

entre dois pontos S e G. A Figura 6b apresenta um dos caminhos possíveis considerando que não sejam realizados movimentos diagonais.

Figura 6 – Mapa de Grade e possível trajeto entre os pontos S e G.



Nota: o desenho (a) ao mapa de base, e o (b) o caminho mais curto encontrado.

Fonte: Ben-Ari; Mondada (2018 p. 166).

Para definir a trajetória o algoritmo utiliza uma lógica baseada em custo de movimento, calculando quantos movimentos serão necessários para se deslocar de um ponto a outro no mapa. A aplicação deste algoritmo se inicia pelo ponto de partida S no qual custo de movimento C_M é igual a 0. A cada novo deslocamento o custo de movimento é determinado pela soma do custo anterior C_A e o custo K de cada novo movimento, conforme a Equação 10.

$$C_M = C_A + K \quad (10)$$

A Figura 7 apresenta o resultado gráfico de um mapa virtual determinado a partir do algoritmo de custo de movimento.

Figura 7 – Mapa após aplicado o algoritmo do custo de movimento.



Fonte: Ben-Ari; Mondada (2018 p. 168).

Sendo K uma constante de custo incremental, neste caso definida igual a 1, podemos determinar os custos de movimentos dos blocos vizinhos a S como 1. Assim, os vizinhos dos blocos com custo 1 deverão receber seu custo igual a 2, e dessa forma sucessivamente até que a quantidade de movimentos necessários para alcançar o destino seja encontrada. Na Figura 7 é apresentado este mapa após feitas 9 interações, que foi a quantidade necessária para que um caminho entre S e G seja traçado.

2.3 SENSORES

De acordo com Webster (1999), a função de um sensor é realizar a conversão de uma grandeza física em padrões de sinais elétricos. Essas grandezas podem ser provenientes de várias origens, como por exemplo mecânica ou eletromagnética, então o sinal deverá ser transmitido a um sistema de processamento para que a extração da informação desejada ocorra.

2.3.1 Categorias de sensores

Para Ben-Ari e Mondada (2018, p. 22) os sensores possuem duas classificações básicas: proprioceptivos ou exteroceptivos.

Os proprioceptivos são sensores utilizados na análise de parâmetros internos do equipamento, por exemplo o velocímetro de um carro medindo a velocidade pela estimativa do tempo em função da revolução de sua roda.

Já os exteroceptivos medem eventos externos ao dispositivo, por exemplo a verificação da distância relativa entre dois objetos com o uso de um equipamento ultrassônico, que relaciona o tempo necessário de reflexão da onda sonora com a velocidade de deslocamento do som em determinado ambiente. Segundo Siegwart (2011, p.104) os sensores exteroceptivos são subdivididos em:

- a) sensores ativos, afetam o ambiente emitindo energia, por exemplo um sonar que utiliza a reflexão das ondas sonoras para determinar a distância relativa entre dois objetos;

- b) sensores passivos, realizam as medições sem que o meio seja afetado, por exemplo uma câmera que registra a luz refletida de um objeto observado para determinar sua geometria.

A Tabela 1 apresenta uma análise qualitativa de sensores conforme sua aplicação.

Tabela 1 – Categorização de sensores

CLASSIFICAÇÃO	APLICAÇÃO	SENSOR	CLASSIFICAÇÃO	
Sensores Táteis	Detecção de contato físico ou proximidade; interruptores de segurança	Interruptores de contato	EC	P
		Barreiras ópticas	EC	A
		Sensores de proximidade sem contato	EC	A
Sensores de roda ou motor	Mede a velocidade e posicionamento da roda ou do motor	Encoders de escova	PC	P
		Potenciômetros	PC	P
		Sincronizadores, resolvers	PC	A
		Encoders ópticos	PC	A
		Encoders magnéticos	PC	A
		Encoders indutivos	PC	A
		Encoders capacitivos	PC	A
Sensores de direção	Mensura a orientação do robô em relação a um quadro de referência fixo	Bússola	EC	P
		Giroscópios	PC	P
		Inclinômetros	EC	A/P
Sensores de aceleração	Mede a aceleração que o corpo sofre	Acelerômetros	PC	P
Marcadores Fixos	Localização com base em uma referência fixa	GPS	EC	A
		Marcadores ativos ópticos ou RF	EC	A
		Marcadores ativos ultrassônicos	EC	A
		Marcadores refletivos	EC	A
Ativos variantes	Usado para refletividade, tempo de voo e triangulação geométrica	Sensores de refletividade	EC	A
		Sensores ultrassônicos	EC	A
		Telêmetro à laser	EC	A
		Triangulação óptica	EC	A
		Luz estruturada	EC	A
Sensores de deslocamento ou de velocidade	Mede a velocidade em relação absoluta ou relativa do corpo	Radar Doppler	EC	A
		Radar Doppler sonoro	EC	A
Sensores de visão	Análise visual da imagem, segmentação e reconhecimento de objetos	Câmeras CCD/CMOS	EC	P
		Pacotes de variação visual	EC	P
		Pacotes de rastreamentos de objeto	EC	P

Nota: Os sensores, conforme sua funcionalidade, são divididos em: proprioceptivos (PC) ou exteroceptivos (EC), em passivos (P) ou ativos (A).

Fonte: Adaptado de Siegwart (2011, p.104) e Webster (1999).

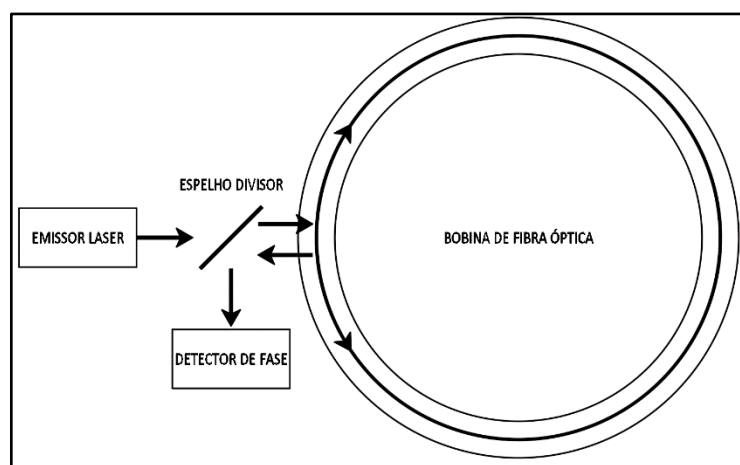
2.3.2 Sensores utilizados em sistemas de navegação autônoma

Segundo Ben-Ari e Mondada (2018, p. 63-79), a navegação autônoma pode ser aplicada com o auxílio de sensores, tendo como os principais modelos utilizados: giroscópio óptico, acelerômetro, *encoder* óptico e o sensor de espectro.

2.3.2.1 Giroscópio óptico

De acordo com Siegwart (2011, p. 118-119), os giroscópios são sensores que medem a variação da orientação de um determinado corpo em relação a um referencial fixo inicial, fornecendo uma medida absoluta de posicionamento de um sistema móvel. Giroscópios ópticos são constituídos de cabos de fibra óptica envoltos a um cilindro, um emissor laser, um espelho divisor e um sensor óptico como está apresentado na Figura 8.

Figura 8 – Giroscópio óptico.



Fonte: Adaptado de Siegwart (2011, p. 119).

O laser é disparado e dividido em dois feixes pelo espelho divisor, seguindo simultaneamente em direções opostas até colidirem no sensor fixado independentemente ao meio em que o laser está sendo propagado, desta forma quando o sensor sofre rotação um dos caminhos se torna relativamente menor ao outro, o que gera uma diferença de fase entre os lasers e essa diferença fornece o dado necessário para o cálculo da velocidade angular do cilindro.

2.3.2.2 Acelerômetro

Acelerômetros são dispositivos capazes de mensurar todas as forças externas que atuam sobre ele, incluindo a gravidade. Seu funcionamento ocorre através de um sistema de massa-mola, no qual a posição tridimensional da massa em relação ao invólucro pode ser medida. Como exemplo, tem-se a aplicação de uma força externa

F_{ap} , como a da gravidade, no qual uma mola ideal sofre uma força proporcional ao seu deslocamento (SIEGWART, 2011, p. 119-120). O sistema pode ser descrito pela Equação 11.

$$F_{ap} = F_i + F_{am} + F_m = m\ddot{x} + c\dot{x} + kx \quad (11)$$

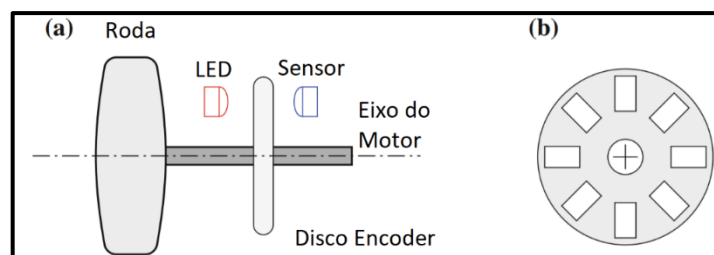
Sendo F_i a força inercial do corpo, m a massa do corpo, F_{am} a força de amortecimento da mola, c o coeficiente de amortecimento, F_m a força da mola, k a constante da mola e x a posição relativa de equilíbrio. Quando a condição $\ddot{x} = 0$ for atendida, podemos obter a aceleração a partir da Equação 12.

$$a = \frac{kx}{m} \quad (12)$$

2.3.2.3 Encoder óptico

Encoders ópticos funcionam pelo princípio de condução da luz. Utiliza-se para isso um disco com furações regulares, onde em um lado deste disco ficará posicionado um LED e no outro lado o sensor óptico conforme apresentado na Figura 9.

Figura 9 – Sensor do tipo *encoder* óptico.



Nota: o desenho (a) se refere ao sensor, e o (b) o disco regularmente vazado.
Fonte: Ben-Ari; Mondada (2018, p. 76).

Durante a rotação do disco a parte vazada é identificada e uma variável de controle é incrementada, então baseado na relação entre o número de pulsos contados e o tempo entre eles é possível determinar o deslocamento e a velocidade angulares do disco. Para determinar o deslocamento e a velocidade do corpo é necessário estabelecer a relação entre o deslocamento angular do disco e o deslocamento linear da roda (BEN-ARI; MONDADA, 2018, p. 76).

Para calcular o deslocamento linear s de uma roda, é necessário definir os parâmetros de número n_T de pulsos por revolução, o número de pulsos lidos n e o comprimento da circunferência C de uma roda que pode ser determinado pela Equação 13, onde r é o raio da roda. A partir da roda apresentado na Figura 9 contendo 8 pulsos por revolução, o deslocamento pode ser calculado pela Equação 14.

$$C = 2\pi r \quad (13)$$

$$s = C \frac{n}{n_T} \quad (14)$$

2.3.2.4 Sensor de espectro

Para Siegwart (2011 p. 143) um sensor também utilizado em sistemas de navegação autônoma são os sensores de espectro que têm seu funcionamento baseado na técnica do limiar constante para identificação de cores regidas pelo sistema de cores RGB, onde o usuário define uma cor ou faixa de cores que devem ser detectadas e o sensor informa quando esta cor for observada.

2.4 SISTEMA DE CONTROLE BASEADO EM LÓGICA FUZZY

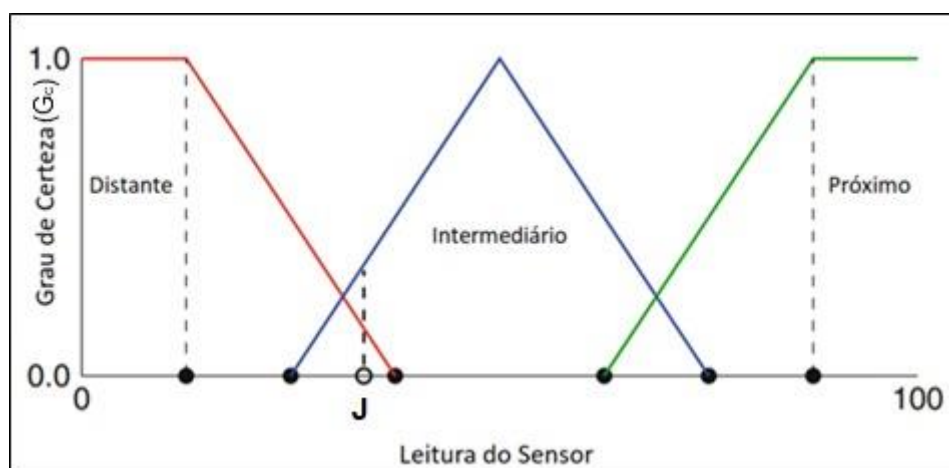
Para Holland (2004, p. 45-46), a lógica *Fuzzy* consiste na análise de um conjunto de valores difusos obtidos de uma combinação de diferentes entradas para obter um resultado que não pode ser medido diretamente, mas que pode ser obtido de modo inferencial a partir da combinação dessas entradas.

De acordo com Ben-Ari e Mondada (2018, p. 179-182) algoritmos baseados na lógica *Fuzzy* possuem três etapas essenciais para a correta operação: *Fuzzyfy*, aplicação de regras e *Defuzzify*.

Na etapa *Fuzzyfy* são realizadas leituras do sensor com objetivo de transformar a informação recebida em um dado numérico denominado grau de certeza, a fim de determinar a situação que a leitura se encontra dentro dos valores esperados pelo sistema. Considerando o sinal recebido em três níveis qualitativos: distante, intermediário e próximo, com possível variação de 0 a 100 e um grau de

certeza de 0 a 1. O grau de certeza pode ser definido através da observação da Figura 10.

Figura 10 – Determinação do grau de certeza.



Fonte: Ben-Ari; Mondada (2018 p. 180).

Os pontos demarcados na Figura 10 representam pontos críticos de leitura de um sensor genérico, e os valores para estes são dependentes de cada sensor a ser utilizado. Para a leitura J com o grau de certeza G_c misto entre dois níveis qualitativos G_{ca} e G_{cb} , o grau de certeza dessa medida pode ser definido pela Equação 15 .

$$G_c = G_{ca} G_{cb} \quad (15)$$

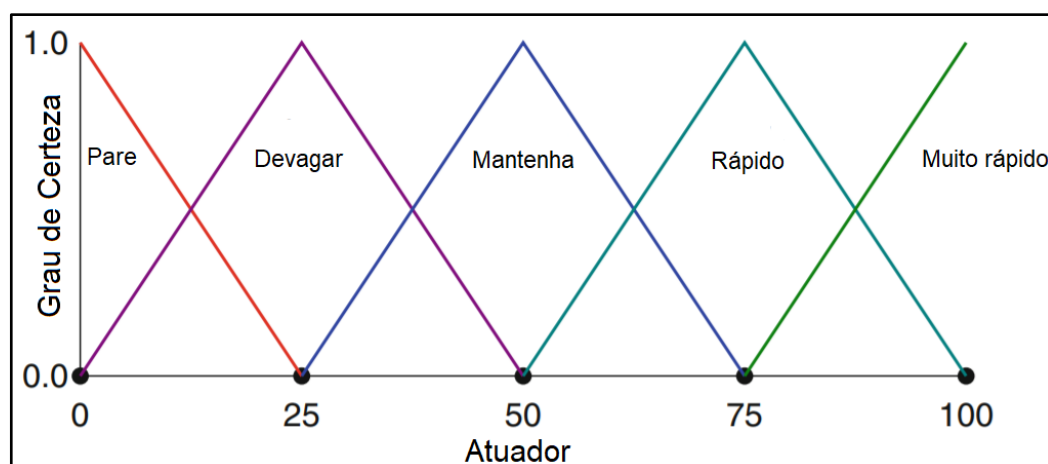
Após ser determinado corretamente o grau de certeza associado para cada medida, cinco regras são definidas para que ações de controle sejam tomadas:

- Se estiver distante, então incremente muito rápido;
- Se estiver entre distante e intermediário, então incremente rápido;
- Se estiver no ponto intermediário, então mantenha a taxa de incremental;
- Se estiver entre o ponto intermediário e próximo, então incremente devagar;
- Se estiver próximo, então pare.

A próxima etapa a ser realizada é a de *Defuzzify* que consiste na combinação das ações de controle que deverão agir sob o sistema. A partir das considerações dos graus de certeza encontrados na etapa *Fuzzify* juntamente com a regra de ação

definida na etapa anterior, o atuador poderá ser acionado corretamente para que as medidas esperadas sejam alcançadas. As ações que o atuador pode receber da lógica de controle estão escaladas conforme apresentado na Figura 11, sendo que a saída desta etapa será um percentual determinando a velocidade de resposta do sistema.

Figura 11 – Determinação da velocidade de resposta de um sistema Fuzzy.

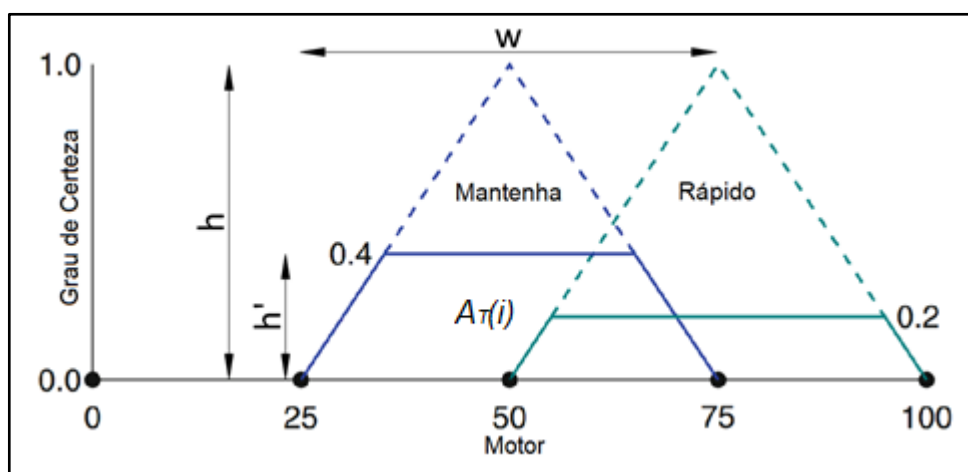


Fonte: Ben-Ari; Mondada (2018 p. 181).

Considerando a base w e a altura h do triângulo de atuação, pode-se definir a altura h' de uma forma trapezoidal a partir da delimitação de um determinado grau de confiabilidade a ser seguido, conforme apresentado na Figura 12. Assim uma determinada área trapezoidal formada $A_T(i)$ pode ser descrita pela Equação 16.

$$A_T(i) = wh' \left(1 - \frac{h'}{2h} \right) \quad (16)$$

Figura 12 – Áreas trapezoidais determinadas em uma configuração Fuzzy.



Fonte: Ben-Ari; Mondada (2018 p. 181).

Com os valores das áreas trapezoidais calculadas $A_T(i)$ e os valores centrais de atuação $S_c(i)$ determinados pelo gráfico possuindo n áreas trapezoidais, cada área deverá possuir seu devido peso perante a saída S_a . É necessário, portanto, uma ponderação conforme apresentada na Equação 17.

$$S_a = \frac{A_T(i)S_c(i) + A_T(i+1)S_c(i+1)}{A_T(i) + A_T(i+1)} \dots \Bigg|_{i=1}^n \quad (17)$$

3 ABORDAGENS DA LITERATURA

Tendo em vista o avanço das tecnologias assistivas em equipamentos modernos, é natural observar diferentes técnicas de navegação autônoma serem aplicadas a cadeiras de rodas motorizadas. Neste capítulo serão apresentados diferentes trabalhos relacionados à navegação autônoma de cadeiras de rodas motorizadas e similares sendo apresentados de forma resumida pela Tabela 2.

Tabela 2 – Trabalhos práticos abordados e técnicas utilizadas

Autor	Trabalho	Técnicas de navegação utilizadas
GERSDORF; FRESE (2013)	Análise da confiabilidade de sistemas de navegação implementados utilizando diferentes técnicas	Odometria
		Navegação inercial
		Filtro de Kalman
YOKOZUKA et al. (2012)	Implementação de técnicas de navegação autônoma visando a quantidade de dados necessários em um sistema SLAM aplicado em uma cadeira de rodas	SLAM
		Amostragem do mapa
		Criação de trajetória
ZHANG et al. (2016)	Análise da performance do sistema autônomo de navegação desenvolvido em uma cadeira de rodas que utiliza interface cérebro-computador	Odometria
		Cérebro-Computador
		Criação de trajetória
WEE-KIAT et al. (2014)	Sistema BCI em uma cadeira de rodas baseado no potencial evocado estacionário para minimização de decisões humanas em um sistema de navegação autônomo	BIC
		Evocado Estacionário
		Desvio de obstáculos
MATSUMOTO et al. (2006)	Sistema de navegação autônomo em percurso pré-definido utilizando tecnologia GPS e marcos de referência	GPS
		Marcos de referência RFID
		Filtro de Kalman
		Desvio de obstáculos
		Criação de trajetória

Fonte: Autor (2019).

Gersdorf e Frese (2013), em seu trabalho sobre navegação autônoma abordam dois sistemas de odometria fazendo uso do filtro de Kalman. Na primeira abordagem, os autores relatam a utilização de um componente microeletrônico, que

possui dois acelerômetros em conjunto com um giroscópio, aplicados em um andador, como pode ser visto na Figura 13.

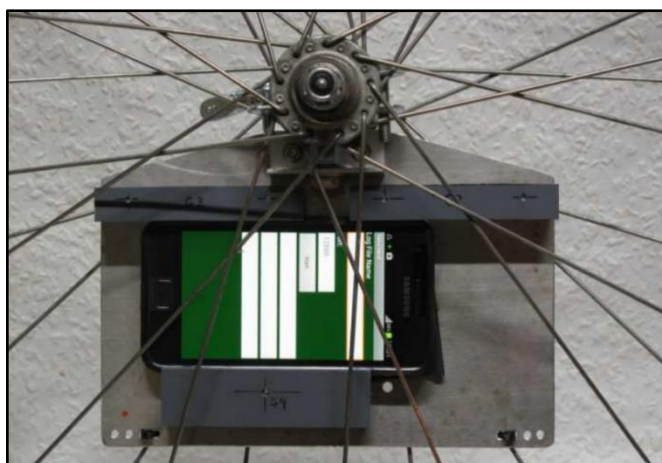
Figura 13 – Sensores aplicados no andador.



Fonte: Adaptado de Gersdorf e Frese (2013).

Já na segunda abordagem, os autores sugerem a adaptação de um celular Samsung Galaxy S2 na roda de uma bicicleta, conforme visto na Figura 14. Este celular possui um sensor tipo acelerômetro de 3 eixos e um giroscópio de 3 eixos nativos, o que oportuniza a medição das variáveis de aceleração e inclinação da bicicleta.

Figura 14 – Celular adaptado na roda da bicicleta.



Fonte: Gersdorf e Frese (2013).

Na fase experimental do trabalho, Gersdorf e Frese (2013) coletaram os dados destes sensores e, posteriormente, aplicaram um filtro de Kalman para

tratamento dos dados. Na sequência, foram analisados os parâmetros de aceleração, inclinação e deslocamento para verificar a confiabilidade da informação adquirida.

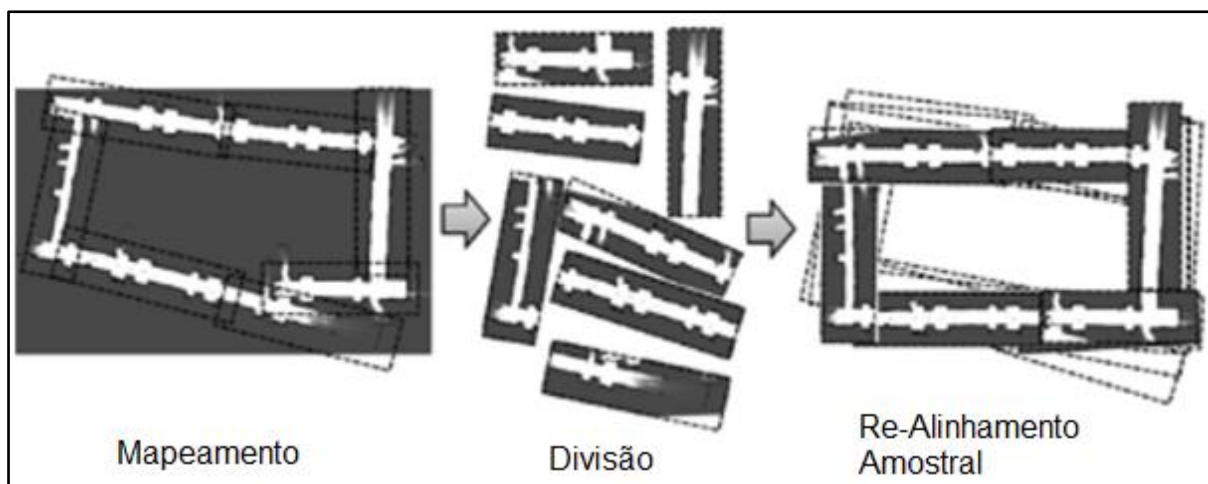
No primeiro teste do andador, foi realizada uma simulação de um idoso sob um terreno irregular de paralelepípedos, chegando a velocidade de 1,5 m/s, que é definida como uma velocidade de locomoção desconfortável para este caso, ainda assim os sensores mantiveram as medições de forma contínua. No segundo teste o andador foi acelerado por 2 segundos até alcançar a velocidade de 4 m/s, fazendo com que o giroscópio e um acelerômetro saturassem, porém devido ao filtro de Kalman as medidas de deslocamento se mantiveram conexas.

Para a bicicleta o objetivo dos testes foi verificar a confiabilidade da medição do deslocamento para longas distâncias, realizado em um ambiente urbano por um percurso de 4 quilômetros com uma velocidade média 6 m/s. Para isso um contador de voltas magnético, não especificado pelo autor, foi utilizado a fim de validar os dados coletados pelos sensores do celular. Com base no teste, a diferença obtida entre o sensor externo e o celular foi menor que um metro em todo percurso, o que representa menos de 0,1%.

Já Yokozuka et al. (2012), em seu sistema de navegação autônoma, optou pelo uso de um sistema baseado no método SLAM. Este método consiste na composição de mapas e navegação simultânea. Porém, a abordagem clássica do SLAM o torna incapaz de gerar um mapa conexo, já que para que o mesmo seja constituído, um número muito grande de amostras se tornariam necessárias, exigindo uma quantidade massiva de dados.

Como proposta de solução ao problema de excesso de dados, os autores buscaram reduzir a quantidade de memória requerida em cada processo através de um método amostral da coleta de dados e pela divisão de um único mapa em mapas menores e posteriormente realinhados de forma amostral, conforme apresentado na Figura 15.

Figura 15 – Técnica aplicada para redução na quantidade de dados.



Fonte: Adaptado de Yokozuka et al. (2012).

Para avaliar o funcionamento da técnica descrita, foi realizado um teste, tomando como base mapas de tamanho de 80x80x12 m e com uma resolução de 20 cm, considerando os elementos adquiridos com valores inteiros de 16 bits, a memória utilizada para um sub mapa é 32 MB. O número total de sub mapas são 34 e a memória utilizada para esta representação gráfica ficou em 1 GB.

Foi feita uma comparação com um método SLAM convencional sob as mesmas condições, porém o mapa todo analisado em um processo. O mapa possui 320x320x12 m e a resolução é de 20 cm, a memória utilizada para a representação desse mapa é 270MB, supondo que 30 aquisições foram feitas, teremos 8,1 GB. O resultado deste teste demonstra que a técnica usada por amostragem do autor necessita oito vezes menos memória que um sistema SLAM convencional.

A abordagem apresentada por Zhang et al. (2016) propõe-se ao controle de uma cadeira de rodas automatizada, utilizando um *notebook* para a escolha do destino, onde o sistema de navegação planeja um caminho e o navega em direção ao destino, e com o auxílio de uma interface cérebro-computador o usuário pode emitir comandos de seleção de trajetória e parada emergencial.

O autor levanta problemas de trabalhos a cerca deste assunto, onde questiona a elevada carga mental necessária para que o usuário desde desempenhe funções de controle para uma cadeira totalmente controlada por uma interface cérebro-computador por longos períodos de tempo. Assim, técnicas foram aplicadas para a redução desta, entre elas estão a implementação de um sistema

semiautônomo, onde o usuário apenas viria a intervir em casos de emergência e o carregamento de mapas prévio.

Para avaliar o desempenho do aparato, foram conduzidos alguns testes com nove voluntários, e posteriormente analisados os seguintes parâmetros:

- a) tempo de concentração para escolha do destino (CT);
- b) seleção de destino incorreto (FS);
- c) tempo de resposta de parada (RT);
- d) taxa de falsa ativação (FA).

O autor compara os parâmetros seleção de trajetória e parada do sistema encontrados em seu sistema(1), em relação a uma cadeira de rodas controlada em tempo integral(2) por uma interface cérebro-computador, e os resultados médios obtidos são apresentados nas Tabela 3 e Tabela 4 .

Tabela 3 – Performance para seleção de trajetória

SISTEMA	CT(s)	FS
1	$12,8 \pm 2,0$	0
2	$24,3 \pm 1,6$	0

Fonte: Adaptado de Zhang et al. (2016).

Tabela 4 – Performance para parada do sistema

SISTEMA	RT(s)	FA
1	$6,2 \pm 1,4$	0
2	$4,5 \pm 0,5$	0

Fonte: Adaptado de Zhang et al. (2016).

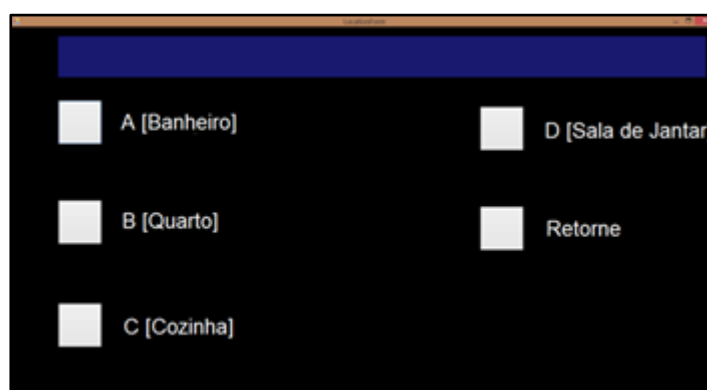
Os resultados obtidos demonstram que no caso de seleção de uma trajetória, a carga mental exercida é aliviada a ponto do usuário conseguir traçar a mesma trajetória pela metade do tempo. Porém ao analisar um comando de parada para a cadeira onde o usuário já estava operando os comandos pela interface cérebro-computador, demonstrou um resultado mais rápido, isso ocorre devido a comandos simples não exercerem uma carga mental significativa, assim quem já estava operando no modo manual, teve uma resposta mais rápida para um sinal de parada da cadeira.

Wee-Kiat et al. (2014) apresentam em seu trabalho a utilização de um sistema de BCI baseado no potencial evocado estacionário (SSEVP) para reduzir o número de decisões que precisam ser concluídas pelo usuário ao se deslocar de um ponto ao

outro com uma cadeira de rodas autônoma, a cadeira também possuirá um sistema de desvio de obstáculos.

O sistema SSEVP se baseia na distinção da frequência oscilatória de um sinal eletromagnético visível. Na Figura 16 está sendo apresentada a interface desenvolvida, onde consiste em 5 quadrados luminosos, possuindo sua frequência oscilatória de 11 a 15 Hz.

Figura 16 – Software desenvolvido.



Fonte: Adaptado de Wee-Kiat (2014).

Sessenta e dois indivíduos saudáveis participaram de um teste baseado no sistema BCI em questão, com o objetivo de uma avaliação se estariam aptos a navegar na cadeira em questão e destes apenas trinta e sete alcançaram um aproveitamento melhor que 50% sendo calculado pela Equação 18.

$$Aproveitamento = \frac{VN + VN}{VP + VN + FP + FN} \quad (18)$$

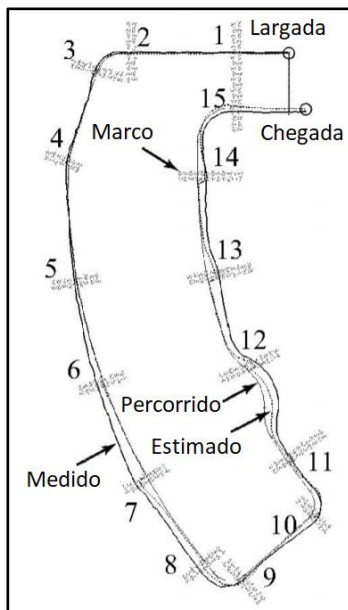
Onde, VP corresponde a verdadeiro positivo, VN a verdadeiro negativo, FP a falso positivo e FN a falso negativo. Apenas trinta e sete pessoas obtiveram o aproveitamento necessário para a realização do segundo teste. No teste com a cadeira efetivamente, foi solicitado que cada pessoa escolhesse um dos destinos apresentados pela Figura 16, olhando para o quadrado correspondente e em 100% dos casos o sistema seguiu até o destino especificado.

Matsumoto et al. (2006) abordam o uso da tecnologia GPS e marcos artificiais baseados em tecnologia RFID, juntamente com um sensor a laser a fim da cadeira detectar a aproximação de possíveis obstáculos. Os drivers da roda são controlados

por uma CPU embarcada que realiza a navegação autônoma utilizando um filtro de Kalman para a fusão das informações do GPS e do RFID.

Foi proposto a execução de um percurso fechado com quinze marcos artificiais distribuídos conforme observado na Figura 17, a fim de analisar o funcionamento do filtro de Kalman.

Figura 17 – Percurso proposto.



Fonte: Adaptado de Matsumoto et al. (2006).

Após percorrer um terço do curso total, o desvio da cadeira de rodas do percurso foi de no máximo 0,4 metros. Analisando a Figura 17 a cadeira demonstra que detectou os marcos artificiais RFID e a cada vez que se deslocou sobre algum, realizou o ajuste de direcionamento para a correção da posição.

4 DESENVOLVIMENTO DO TRABALHO

Este trabalho visa o desenvolvimento de um sistema de navegação autônomo que facilite o deslocamento de pessoas com deficiências motoras de locomoção por meio de uma cadeira de rodas motorizada.

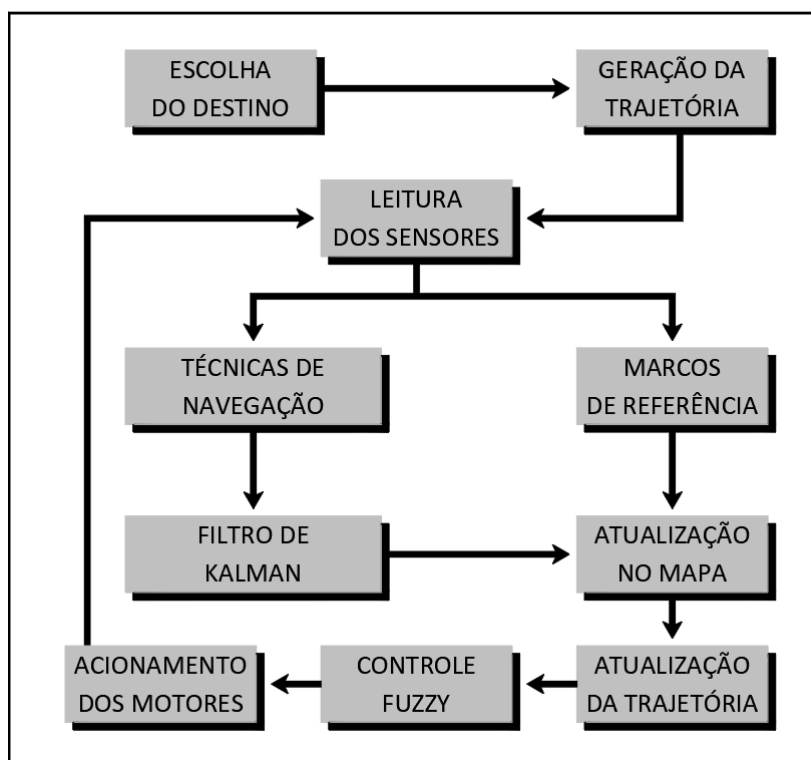
A escolha do destino e a visualização da localização atualizada é feita através da interface homem máquina (IHM) desenvolvida utilizando um *display touch screen*¹ fixado na cadeira. Com o mapa previamente virtualizado sendo descrito pela técnica do mapa discreto, a rota foi traçada de acordo com o destino especificado, e sua execução pode ser acompanhada pelo usuário.

Com o objetivo de realizar o controle do dispositivo, foi implementada uma plataforma embarcada, que tem como atribuição desde a criação da trajetória, a coleta e o processamento de dados para a navegação através da técnica denominada odometria. Estes dados obtidos são linearizados pelo filtro de Kalman, afim de garantir que os valores não lineares sejam devidamente considerados. Em paralelo ao processo de navegação, foi implementado um sistema para correção de possíveis erros odométricos utilizando a técnica de reconhecimento de marcos artificiais com o uso do sensor de espectro RGB e placas de piso tátil com cores específicas distribuídas na edificação.

Após os dados coletados, linearizados e marcos artificiais identificados, é feita a correção da posição e da trajetória com a atuação do controle *fuzzy* e acionamento dos motores para um correto deslocamento da cadeira no percurso, desta forma permitindo que a cadeira execute a trajetória determinada sem a supervisão humana. O diagrama de blocos apresentado na Figura 18 aborda o funcionamento do sistema proposto.

¹ Dispositivo para apresentação de informações de forma visual e sensível ao toque.

Figura 18 – Funcionamento do sistema.



Fonte: Autor (2019).

A metodologia de implementação está dividida em 5 etapas, sendo elas: montagem do *hardware*, interligação dos equipamentos, controle dos motores por lógica *fuzzy* para a navegação autônoma, implementação do filtro de Kalman e detecção de marcos artificiais. Nas seções seguintes, são apresentadas as metodologias para cada uma das etapas de implementação.

4.1 MONTAGEM DO HARDWARE

O processamento dos dados e a geração da trajetória são executados por meio da plataforma embarcada Raspberry Pi 3 B+, a qual conta com um processador *Quad-Core* com 1.4 GHz de *clock*² e memória RAM de 1 *gigabyte*. Alguns conectores de periféricos como o HDMI são nativos de fábrica, que facilita a comunicação com outros equipamentos. O sistema operacional (SO) instalado foi o Raspbian que é um sistema baseado no SO Linux Debian. Este sistema é o recomendado pelo fabricante para que seja otimizada a performance deste componente (RASPBerry, 2019).

² É o número de ciclos de máquina que o processador consegue executar por segundo.

Também foi utilizado um microcontrolador ESP32 Devkit V1, que possui um processador *Dual-Core* com 240 MHz de *clock*. O mesmo foi escolhido por possuir dois conversores digital-analógico (DAC) que possibilitam o envio dos comandos de acionamento ao driver dos motores.

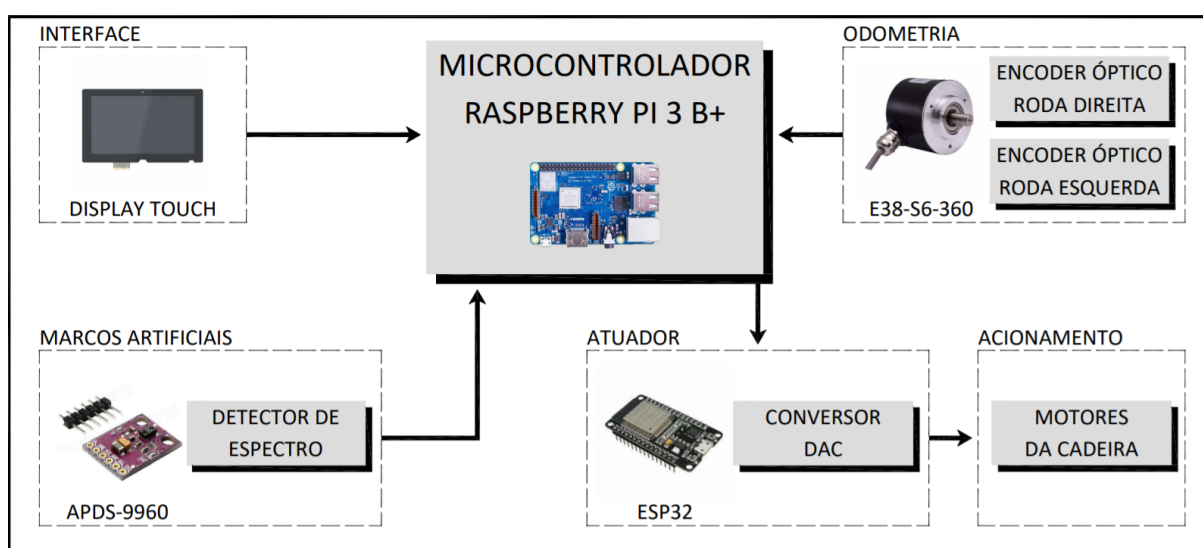
Para o monitoramento do deslocamento do sistema foram utilizados dois *encoders* ópticos acoplados ao eixo do motor de cada roda. Além disso, também foi instalado o sensor de espectro luminoso RGB para o reconhecimento de marcos artificiais, possibilitando a correção de erros advindos do sistema odométrico. Os marcos artificiais estarão dispostos no piso pela edificação em formato de piso tátil colorido.

Foi utilizada uma interface IHM disposta no *display touch screen* de 11 polegadas para a interação com o usuário, onde estão disponíveis as opções de escolha de destino. Nesta interface também poderá ser visualizada a posição atualizada da cadeira dentro do ambiente.

A cadeira de rodas motorizada utilizada é da fabricante Freedom, modelo Compact 13 que possui um driver nativo utilizado para o controle dos motores, com a possibilidade de escolha entre 5 velocidades e com o acréscimo do modo turbo, aplicado em situações que demandam maior força de tração. (FREEDOM, 2019).

O *hardware* descrito está sendo representado esquematicamente pela Figura 19.

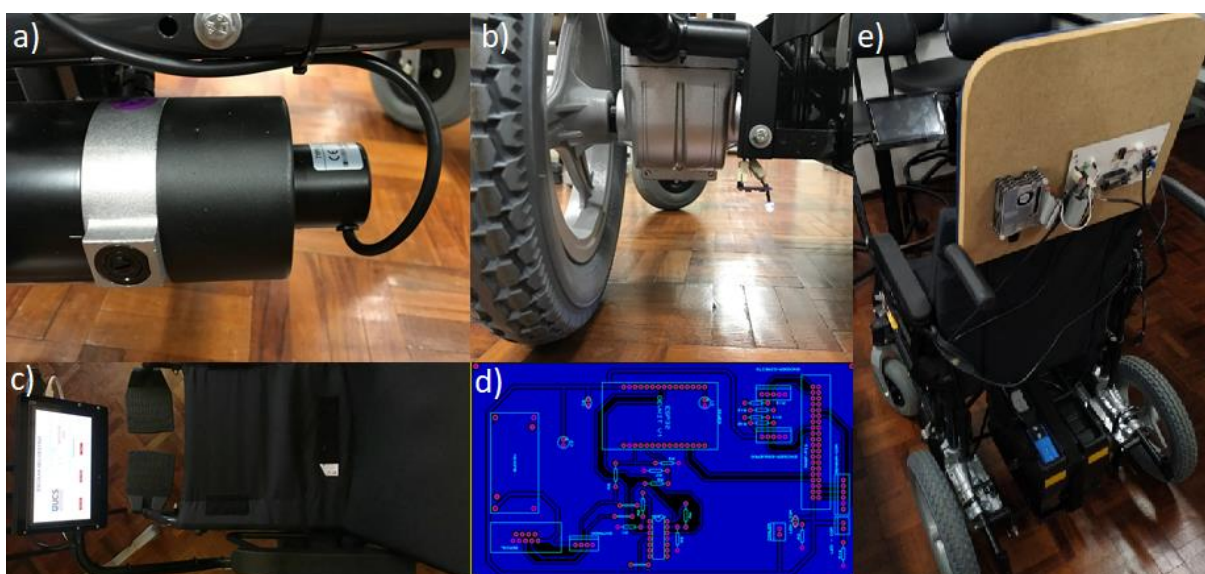
Figura 19 – Esquemático dos componentes da cadeira.



Fonte: Autor (2019).

Com o objetivo de manter o sistema totalmente embarcado foi desenvolvida uma placa de circuito impresso (PCB) conforme o APÊNDICE A, provendo a interligação adequada para os microcontroladores com os demais periféricos do sistema. Para a alimentação elétrica foram utilizadas as baterias de 24 V juntamente com um circuito rebaixador de tensão de 5 V, ambos já instalados na cadeira de rodas. Detalhes construtivos do protótipo desenvolvido estão sendo apresentado na Figura 20, com o intuito de esclarecer pontos relativos a montagem do hardware.

Figura 20 – Detalhes construtivos do protótipo desenvolvido.



Nota: a figura (a) encoder acoplado no eixo do motor, (b) fixação do sensor de espectro luminoso RGB, (c) IHM implementada no display, (d) placa de controle do sistema e (e) protótipo desenvolvido
Fonte: Autor (2019).

4.2 INTERLIGAÇÃO DOS EQUIPAMENTOS

Para alcançar a totalidade de funcionamento da proposta deste trabalho, dispositivos de comunicação foram utilizados de forma paralela atuando com diferentes atribuições, ainda assim, controlados simultaneamente pela Raspberry.

4.2.1 Comunicação Raspberry com ESP32

Para a conexão entre os microcontroladores do sistema de forma exitosa, foi escolhida a comunicação serial, de modo que Raspberry executa o envio dos dados ao ESP32 e este último realiza a atuação nos motores.

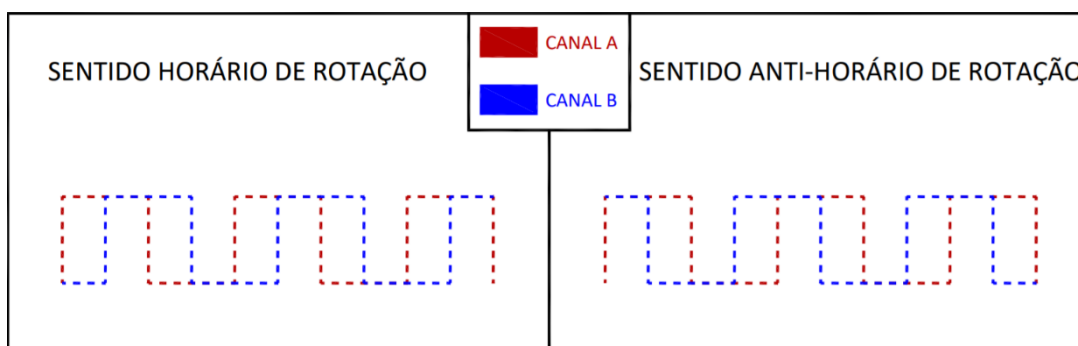
A mensagem enviada da Raspberry ao Esp32 possui o tamanho de 136 bits e o envio dos dados foi realizado a uma taxa de *baud-rate*³ igual a 115200 bits por segundo, possibilitando o envio dos dados 847 vezes em um segundo. Para que a leitura não fosse comprometida, foi definido uma interrupção do tipo *timer*⁴ na Raspberry para envio destes dados atualizados a cada 10 milissegundos.

4.2.2 Comunicação Raspberry com encoder

O *encoder* utilizado possui dois canais digitais, denominados A e B, e para contabilizar uma volta completa são necessários 360 pulsos por canal, porém, no sistema em questão o encoder está fixo na ponta do eixo do motor, e para a interligação do motor com a roda está sendo utilizada uma redução de 1:32, assim, ao final de uma volta completa da roda da cadeira terão sido contabilizados 11520 pulsos por canal no *encoder*.

Tendo em vista a criticidade da leitura do número de voltas executadas por cada roda, e o quanto influencia para a navegação autônoma, foi atribuído uma interrupção por borda de subida na leitura do Canal A. Para a determinação da direção do movimento da roda foi utilizada a defasagem entre os canais, considerando que quando a borda de subida do Canal A é detectada, é feita também uma leitura digital no canal B e caso a leitura digital indique nível baixo, a roda está girando para a direita, caso a leitura esteja em nível alto, a roda está girando para a esquerda.

Figura 21 – Defasagem entre canais A e B.



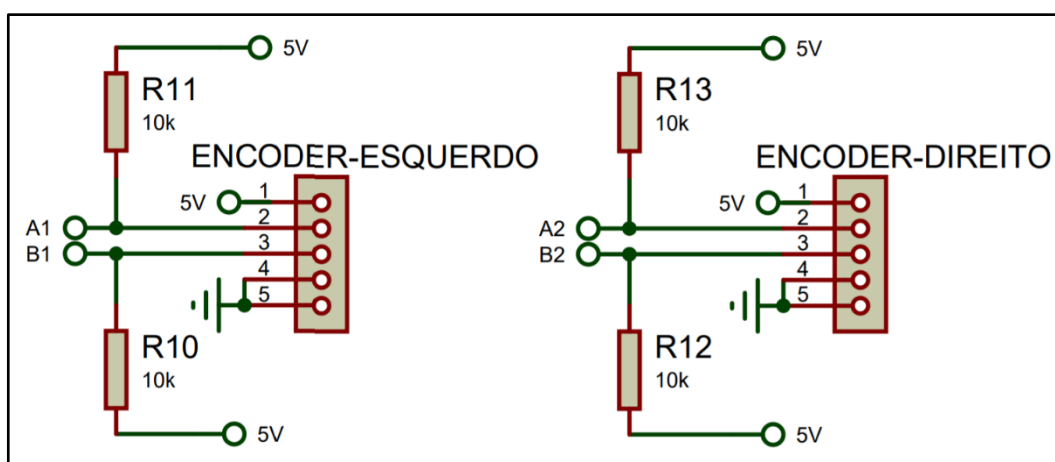
Fonte: Autor (2019).

³ É o número de bits possíveis de envio em um canal de comunicação por segundo.

⁴ Permitem processos temporais, gerar sinais e contar os eventos.

Para o condicionamento do sinal foram definidos dois canais da Raspberry como entradas digitais e posteriormente foi instalado um resistor *pull-up*⁵ de 10 kΩ por canal conforme a Figura 22, com a funcionalidade de garantir a leitura digital coesa, sem que sejam lidos valores flutuantes errôneos durante a operação do equipamento.

Figura 22 – Condicionamento dos *encoders*.



Fonte: Autor (2019).

4.2.3 Comunicação Raspberry com sensor de espectro luminoso

O seu funcionamento é baseado na detecção dos componentes RGB na luz, desta forma ao estar à frente de um plano, determina o percentual de cada um destes componentes neste plano observado. Este tipo de sensor utiliza como base o protocolo de comunicação I2C no endereçamento 0x39 e para leitura dos dados foi utilizado o canal de comunicação I2C da Raspberry.

Os marcos artificiais são compostos de pisos táteis coloridos fixados no piso, onde na edificação estão espalhadas em pontos críticos como: na entrada da edificação e no início de cada corredor longo conforme apresentado na Figura 23.

⁵ Resistor que tem como função enviar nível lógico alto, normalmente utilizado para entradas digitais.

Figura 23 – Marco artificial implementado.



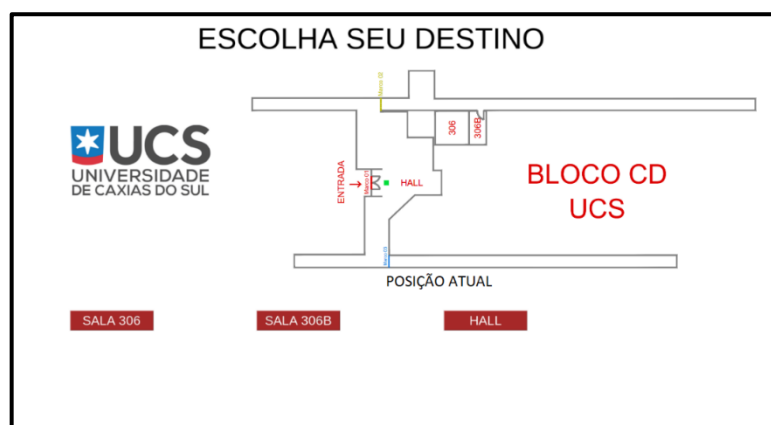
Fonte: Autor (2019).

A implementação do sistema, no momento em que a cadeira passa por cima de um dos marcos, o sensor realiza o reconhecimento do local pela coloração do piso, então uma interrupção é acionada, estabelecendo a atualização da posição da cadeira e a correção na trajetória, se necessário.

4.2.4 Comunicação Raspberry com sistema supervisorio

O supervisorio do sistema tem como objetivo a escolha do destino para onde a cadeira deverá se deslocar, também de monitorar a atual posição da cadeira, informação que será exibida conforme apresentado pela Figura 24 no *display touch screen*.

Figura 24 – Supervisorio do sistema.



Fonte: Autor (2019).

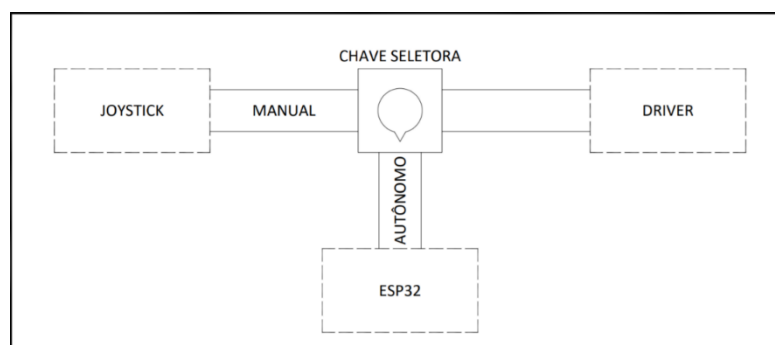
O sistema supervisorio foi desenvolvido em linguagem de programação *python*⁶, assim para ocorrer a interligação com o algoritmo de navegação o mesmo grava um arquivo com as informações referentes à sala escolhida quando algum botão é pressionado, então, a cada 1 segundo o algoritmo de navegação abre este arquivo e analisa se algum destino foi estipulado, caso seja detectado um destino, é calculada a rota, e o deslocamento ocorre.

Em contrapartida, o algoritmo de navegação, grava em outro arquivo as informações da posição atualizada e a cada 1 segundo o supervisorio abre este arquivo e realiza a atualização destes dados. O posicionamento da cadeira é representado por meio de um quadrado na Figura 24.

4.2.5 Comunicação ESP32 com o driver da cadeira

Para o acionamento dos motores, foi utilizado o driver nativo da cadeira de rodas da Freedom, que utiliza dois canais de controle para a realização dos movimentos, porém, para que isso ocorresse foi necessário a instalação de uma chave seletora que possibilitou a seleção entre modo autônomo e modo manual conforme esquemático apresentado pela Figura 25.

Figura 25 – Chave seletora para troca de modo de deslocamento.



Fonte: Autor (2019).

Um dos canais de controle serve apenas para velocidade angular, e o outro para linear, assim, após a adaptação ter sido realizada foi necessário simular os níveis de tensão que *joystick* emitia para o driver utilizando os DAC da ESP32 conforme a Tabela 5.

⁶ Linguagem de programação de alto nível utilizada para a criação de algoritmos.

Tabela 5 – Níveis de tensão fornecidos pelo joystick.

	Canal de Controle Linear		Canal de Controle Angular	
	Frente	Ré	Direita	Esquerda
Tensão Máxima (V)	3,65	1,05	1,08	3,65
Tensão Mínima (V)	2,65	2,13	2,05	2,55
Tensão Parado (V)	2,39		2,30	

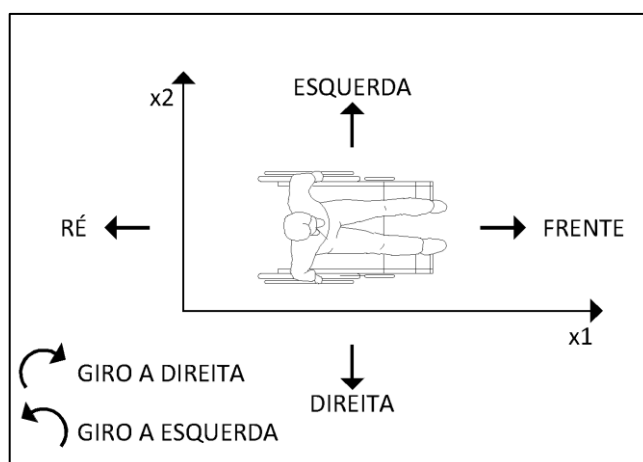
Fonte: Autor (2019).

Para a inicialização do sistema, os níveis necessários de tensão sobre os canais de controle foram definidos iguais aos de tensão parado informados na Tabela 5. Para realizar os demais deslocamentos foram utilizadas as faixas de tensões máximas para velocidades máximas e mínimas para velocidades mínimas.

4.3 CONTROLADOR FUZZY

Para facilitar o entendimento das variáveis utilizadas nas regras do controlador *fuzzy*, foram definidos quatro tipos de comandos: frente, ré, esquerda e direita. A Figura 26 apresenta a correlação de cada comando com o sentido do deslocamento sob os eixos definidos, x_1 e x_2 .

Figura 26 – Comandos utilizados para o controle fuzzy.



Fonte: Autor (2019).

A partir dos comandos definidos, o controle *fuzzy* aplica regras diferenciadas para cada movimento de acordo com a necessidade, sendo as tipologias de

movimento: movimento linear, movimento angular sob o eixo e movimento angular em curva.

Para a realimentação do controle *fuzzy*, cada topologia se baseia em parâmetros como longe, intermediário, perto e objetivo alcançado. Estes parâmetros possuem suas faixas de atuação conforme Tabela 6.

Tabela 6 – Faixa de atuação para cada parâmetro.

DISTÂNCIA / DESTINO	FAIXA DE ATUAÇÃO
LONGE	Enquanto faltar mais que 3 m para o objetivo especificado
INTERMEDIÁRIO	Enquanto faltar menos que 3 m e mais que 1 metro para o objetivo especificado
PERTO	Enquanto faltar menos que 1 m para o objetivo especificado
OBJETIVO ALCANÇADO	Quando a distância especificada for alcançada

Fonte: Autor (2019).

4.3.1.1 Movimento Linear

Neste movimento, foram definidas 4 regras, para o monitoramento da distância percorrida e o restante a percorrer até o destino especificado. Desta forma o sistema realiza ações como incrementar o atuador, manter, decrementar ou parar ao chegar no ponto especificado como demonstra a Tabela 7.

Tabela 7 – Regras fuzzy para movimento retilíneo.

DISTÂNCIA / DESTINO	DESLOCAMENTO LINEAR
LONGE	Incremente nas duas rodas
INTERMEDIÁRIO	Mantenha nas duas rodas
PERTO	Decrementa rapidamente nas duas rodas
OBJETIVO ALCANÇADO	Pare as duas rodas

Fonte: Autor (2019).

Devido a dinâmica de movimento da cadeira, foram percebidas diferenças entre as velocidades lineares das rodas e consequente variação da velocidade angular do sistema apesar de o comando determinado ser de deslocamento retilíneo. Para corrigir esta divergência, foram implementadas 12 novas regras para a execução da compensação deste movimento. Ao verificar que uma das rodas está com a velocidade angular maior que a outra para o movimento retilíneo, há um incremento no atuador da outra roda até que a orientação da cadeira fique alinhada conforme apresentado pela Tabela 8.

Tabela 8 – Regras *fuzzy* para movimento retilíneo com compensação de velocidade angular.

DISTÂNCIA / DESTINO	DESLOCAMENTO LINEAR	RODA ESQUERDA COM VELOCIDADE LINEAR MAIOR	RODA DIREITA COM VELOCIDADE LINEAR MAIOR
LONGE	Incremente nas duas rodas	Incremente velocidade linear na roda direita	Incremente velocidade linear na roda esquerda
INTERMEDIÁRIO	Mantenha nas duas rodas	Mantenha velocidade linear na roda direita	Mantenha velocidade linear na roda esquerda
PERTO	Decrementa nas duas rodas	Decrementa velocidade linear na roda direita	Decrementa velocidade linear na roda esquerda
OBJETIVO ALCANÇADO	Pare as duas rodas	Mantenha as duas rodas	Mantenha as duas rodas

Fonte: Autor (2019).

4.3.1.2 Movimento angular sob o eixo

Controlando apenas o canal de movimento angular, ou seja os comandos esquerda e direita, foram utilizadas 8 regras para a realização de movimentos angulares sob o eixo conforme Tabela 9.

Tabela 9 – Regras *fuzzy* para movimento angular.

ÂNGULO / ÂNGULO DESTINO	DESLOCAMENTO ANGULAR DIREITA	DESLOCAMENTO ANGULAR ESQUERDA
LONGE	Incremente velocidade angular para direita	Incremente velocidade angular para direita
INTERMEDIÁRIO	Mantenha velocidade angular para direita	Mantenha velocidade angular para esquerda
PERTO	Decrementa velocidade angular para direita	Decrementa velocidade angular para esquerda
OBJETIVO ALCANÇADO	Pare as duas rodas	Pare as duas rodas

Fonte: Autor (2019).

Ao retirar a cadeira do repouso, foi percebida uma resistência física na roda do eixo da rotação, submetendo-a à uma rotação menor do que o esperado. Devido a isso, quando o movimento é finalizado, foi necessário a verificação se a rotação esperada nas rodas foi alcançada pela análise do número de pulsos dados nos *encoders*. Com base nesta análise são aplicadas 8 regras adicionais para que ocorra a redundância da conclusão do movimento angular com a angulação devida sendo apresentadas pela Tabela 10.

Tabela 10 – Regras *fuzzy* para ajuste do movimento angular.

ÂNGULO / ÂNGULO DESTINO	AJUSTE ANGULAR
LONGE	Incremente devagar velocidade angular
INTERMEDIÁRIO	Mantenha velocidade angular
PERTO	Decrementa rápido velocidade angular
OBJETIVO ALCANÇADO	Pare

Fonte: Autor (2019).

4.3.1.3 Movimento angular em curva

Nesta topologia de movimento, foi necessário exercer controle simultaneamente na velocidade linear e angular, a fim de executar uma curva de 90 graus. A velocidade angular poderá ser executada no comando esquerda ou direita,

determinando a direção da curva. Desta forma foi necessária a implementação de 12 regras no controlador *fuzzy*, de acordo com a Tabela 11.

Tabela 11 – Regras *fuzzy* para movimento em curva.

DESLOCAMENTO / DESTINO	VELOCIDADE LINEAR	VELOCIDADE ANGULAR
LONGE	Incremente rapidamente velocidade linear	Incremente devagar velocidade angular
INTERMEDIÁRIO	Mantenha velocidade linear	Mantenha velocidade angular
PERTO	Decrementa rápido velocidade linear	Decrementa rápido velocidade angular
OBJETIVO ALCANÇADO	Pare	Pare

Fonte: Autor (2019).

4.4 METODOLOGIA EXPERIMENTAL

A fim de consolidar o funcionamento de cada parte do sistema de forma isolada e conjunta foi elaborada uma rotina de testes que têm como função a averiguação da competência das respostas do sistema perante as ações solicitadas.

4.4.1 Teste de microcontroladores

Com base no *datasheet* fornecido pelo fabricante, foram realizados testes funcionais relacionados a velocidade e precisão de transmissão e leitura de dados presentes nos terminais dos microcontroladores. Desta forma foi possível averiguar o comportamento de cada componente que compõe o sistema.

O microcontrolador Raspberry Pi 3 B+ tem como função realizar o gerenciamento da dinâmica da cadeira de rodas, sendo responsável por adquirir os dados dos sensores e coordenar o atuador para que a cadeira siga a rota desejada, desta forma, deve agir com confiabilidade durante a realização das operações. Para tal, foi realizado uma série de testes a fim de verificar a sua velocidade máxima de leitura. Os testes realizados podem ser observados no APÊNDICE B.

A fim de verificar o funcionamento deste equipamento, foram emitidos sinais percentuais de 0 a 100 para cada tipo de movimento e em seguida foi verificado se as tensões de saída obtidas nos DAC estão de acordo com os valores especificados de acordo com o APÊNDICE C.

4.4.2 Testes controlador fuzzy

4.4.2.1 Aquisição de dados para análise de deslocamento

Para as análises de deslocamento e percurso foi adicionado um sistema de *datalogger*⁷ em que foram registradas as informações referentes aos acionamentos dos motores e deslocamento dos *encoders*. Estas informações foram gravadas em um arquivo com uma velocidade de gravação de 100 Hz.

Este mecanismo auxiliou na análise dos ganhos exercidos pelo controlador *fuzzy*, e a verificação da adequação destes conforme as regras previamente implementadas. Também pode colaborar na reconstrução da trajetória exercida pelas rodas para cada teste com base nas informações registradas pelos *encoders* ao longo do tempo.

4.4.2.2 Teste de deslocamento linear

Considerando a topologia do local definido para realização dos testes estima-se a distância de 5 m de distância suficiente para validação experimental, visto que no trajeto estipulado é a distância que mais se repete.

Para manter a igualdade das condições iniciais do movimento a cadeira foi posicionada em um ponto fixo, assim a partir deste foi requisitado ao sistema a execução do deslocamento linear.

A fim de comparar a validade do algoritmo de controle angular, foram realizados testes de deslocamento linear em duas situações: a primeira com o algoritmo sem o controle angular implementado e o segundo com controle angular implementado.

⁷ Mecanismo auxiliar que coleta e armazena dados de outros dispositivos.

Para a verificação do erro que o sistema está inerente, é comparada a distância percorrida informada pelos *encoders* com a medida realizada por meio de uma trena digital modelo GLM 40, fabricada pela empresa BOSCH, suas características técnicas estão disponíveis no ANEXO A.

4.4.2.3 Teste de deslocamento angular

A cadeira foi posicionada em um marco de ângulo inicial alinhado junto a uma fita posta no piso, posteriormente foi solicitado ao sistema que realizasse os movimentos angulares, juntamente com os arquivos gerados, foram traçadas as tipologias de movimentos realizados.

4.4.2.4 Teste de navegação autônoma

Neste teste, por meio do sistema supervisor, foi solicitado que o equipamento se locomovesse do Hall até a sala 306B, assim, sendo analisada a trajetória. Também foi feita a verificação se os marcos naturais presentes no caminho foram reconhecidos e a devida compensação realizada.

4.4.3 Teste de reconhecimento de marcos artificiais

A fim de verificar a acurácia deste sistema, foi realizada uma rotina de teste, simulando a passagem pelos marcos artificiais especificados e verificado se o sensor foi capaz de identificar corretamente as colorações: amarela, vermelha ou azul. O teste foi realizado dez vezes em cada ponto e apurado em quantos casos o marco foi reconhecido como o ponto da edificação ao qual está relacionado.

Para um melhor reconhecimento do espectro luminoso, foi instalado um LED de alto brilho juntamente com o sensor apontado para o piso, a fim de auxiliar na detecção e reconhecimento das cores compostas na superfície visualizada, e consequentemente o reconhecimento do marco artificial com melhor precisão. Após a instalação desta fonte luminosa os testes foram refeitos e posteriormente comparados com o resultado anteriores.

5 RESULTADOS E DISCUSSÕES

5.1 RESPOSTA DOS MICROCONTROLADORES

Para a leitura de sinais digitais, constatou-se que até 20 kHz a Raspberry tem capacidade de ler bordas de subida que ocorrem no encoder óptico conforme apresentado na Tabela 12.

Tabela 12 – Resultado dos testes de leitura digital.

Teste	Variável de teste	Esperado	Resultado
Leitura de sinais digitais	10 kHz	Detecção de 100000 bordas de subida em um período de 10s	Média de 100000 bordas de subida em 10s
	20 KHz	Detecção de 200000 bordas de subida em um período de 10s	Média de 200000 bordas de subida em 10s
	50 KHz	Detecção de 500000 bordas de subida em um período de 10s	Média de 500000 bordas de subida em 10s
	100 KHz	Detecção de 1000000 bordas de subida em um período de 10s	Média de 950000 bordas de subida em 10s

Fonte: Autor (2019).

No teste do algoritmo de Dijkstra os resultados obtidos se afastaram do valor esperado conforme pode ser verificado na Tabela 13, porém, como esse é calculado antes do deslocamento da cadeira o atraso na resposta não ocasiona atrasos para a navegação da cadeira.

Tabela 13 – Resultado dos testes algoritmo Dijkstra

Teste	Variável de teste	Esperado	Resultado
Geração da trajetória pelo algoritmo de Dijkstra	10 Quadrantes	0,00001 segundo para traçar uma trajetória	0,0001 segundo para traçar uma trajetória
	100 Quadrantes	0,0001 segundo para traçar uma trajetória	0,02 segundo para traçar uma trajetória
	1000 Quadrantes	0,01 segundo para traçar uma trajetória	1 segundo para traçar uma trajetória

Fonte: Autor (2019).

Na verificação realizada nos níveis de tensão gerados pelo DAC do ESP32, pode-se observar os resultados obtidos na Tabela 14.

Tabela 14 – Resultado dos testes feitos no ESP32.

%	DAC Velocidade Linear Esperada		DAC Velocidade Linear Encontrada		DAC Velocidade Linear Esperada		DAC Velocidade Linear Encontrada	
	Frente	Ré	Frente	Ré	Direita	Esquerda	Direita	Esquerda
10	2,75	1,95	2,76	1,94	2,75	1,95	2,75	1,94
20	2,80	1,90	2,80	1,90	2,80	1,90	2,80	1,91
30	2,85	1,85	2,89	1,86	2,85	1,85	2,86	1,88
40	2,90	1,80	2,95	1,81	2,90	1,80	2,93	1,80
50	2,95	1,75	2,97	1,75	2,95	1,75	2,95	1,76
60	3,00	1,70	3,03	1,71	3,00	1,70	3,02	1,70
70	3,05	1,65	3,04	1,65	3,05	1,65	3,06	1,66
80	3,10	1,60	3,12	1,60	3,10	1,60	3,12	1,60
90	3,15	1,55	3,15	1,56	3,15	1,55	3,15	1,55
100	3,20	1,50	3,19	1,49	3,20	1,50	3,18	1,48

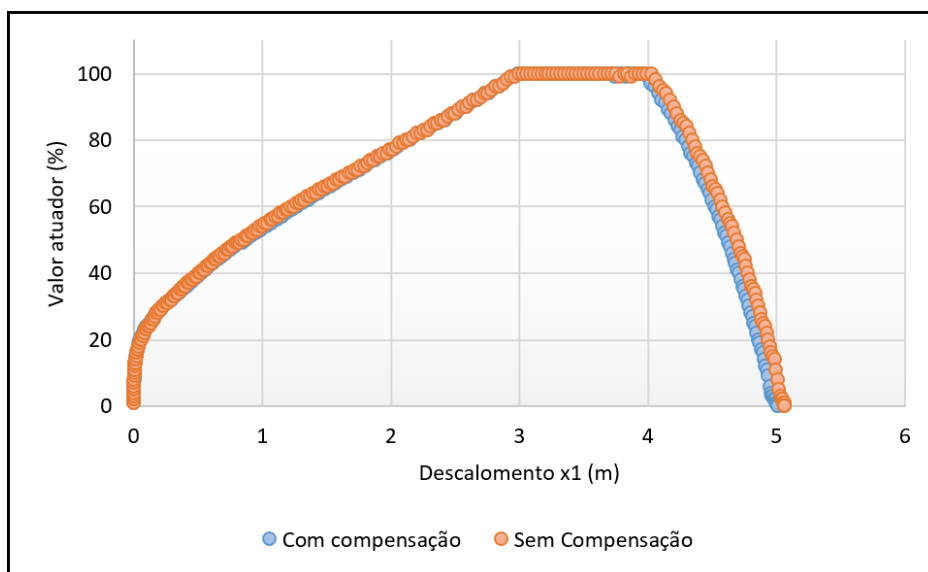
Fonte: Autor (2019) e ESPRESSIF (2019).

Os níveis de tensão gerados mantiveram-se lineares entre 1,5 e 2,8 V, porém ao ultrapassar o valor de 2,8 V o comportamento assume características não lineares nestes canais. Apesar deste comportamento ser atípico conforme o fabricante, não é crítico para o controle *fuzzy* ser aplicado, já que as variáveis de controle da malha são deslocamento linear e angular, e não velocidade, assim sendo uma divergência não crítica para a continuidade do projeto.

5.2 DESLOCAMENTO LINEAR

Em um momento inicial do desenvolvimento desta topologia de deslocamento, foi determinado que não seriam necessárias regras adicionais para o controle do deslocamento, porém, devido as características de cinemática da cadeira de rodas, a mesma não realizou movimentos retilíneos sendo necessária a implementação de novas regras no controlador *fuzzy* para que a trajetória fosse compensada.

Na topologia determinada, a cadeira foi posta no ponto inicial e um comando para se locomover por 5 metros foi emitido. Com base neste movimento, pode-se observar o ganho exercido no comando frente por meio do controlador *fuzzy* para os deslocamentos com e sem compensação sendo apresentado no Gráfico 1.

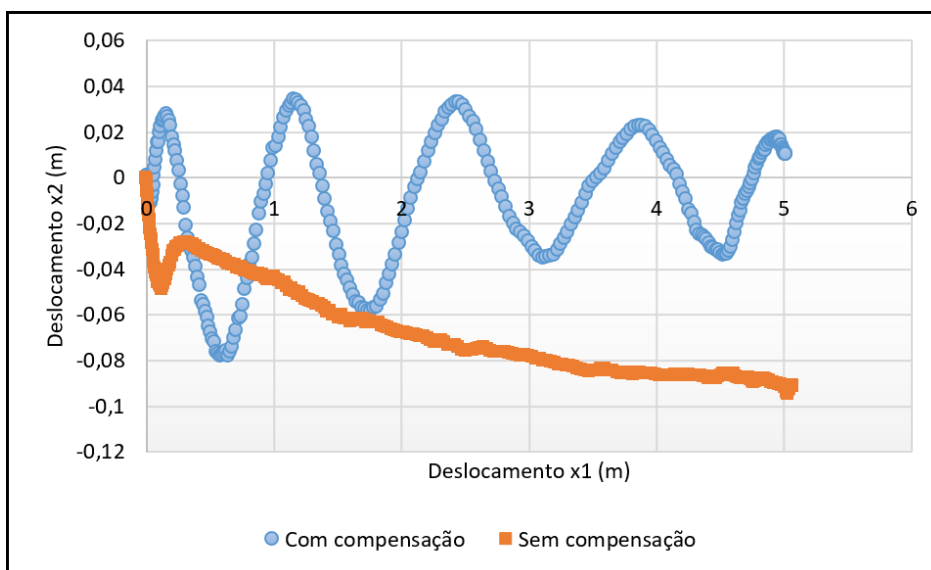
Gráfico 1 – Atuador controlador *fuzzy* para o comando frente.

Fonte: Autor (2019).

Considerando o deslocamento em x_1 , no Gráfico 1, o controlador apresentou resultados satisfatórios para os dois sistemas apresentados de acordo com o esperado perante as regras definidas.

Para a análise de orientação da cadeira durante trajetória, foi verificada a variação do deslocamento que a cadeira sofre e a performance do algoritmo de correção de orientação apresentado pelo Gráfico 2.

Gráfico 2 – Variação de deslocamento entre as rodas.

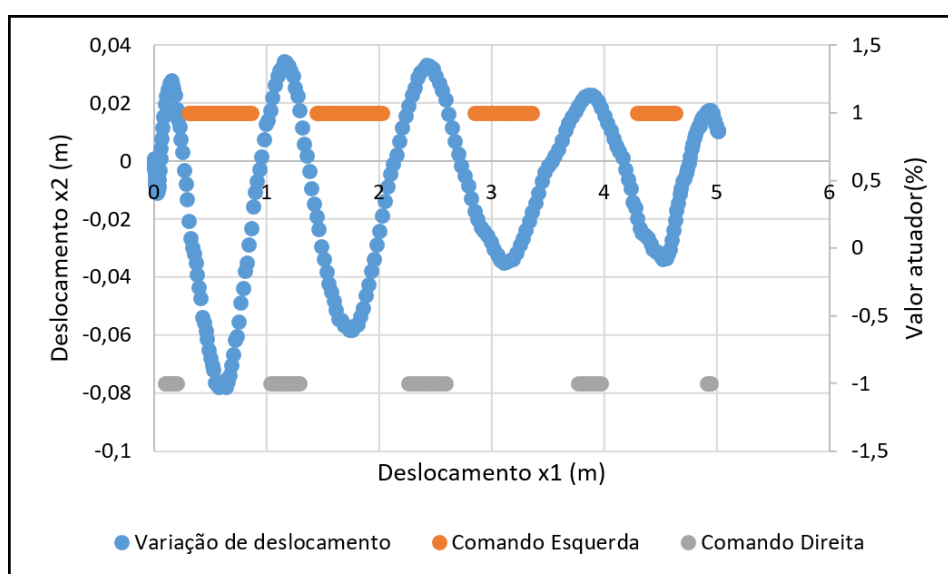


Fonte: Autor (2019).

O valor da compensação no eixo x_2 , como apresentado no Gráfico 2, demonstrou uma tendência de redução de sua variação quando comparado ao sistema sem compensação, visto que realiza compensações sequenciais nos comandos esquerda e direita.

Para uma melhor visualização do método compensatório proposto, será realizada a análise dos comandos esquerda e direita em relação a compensação que está sendo apresentada pelo Gráfico 3.

Gráfico 3 – Variação de deslocamento com compensações utilizadas.

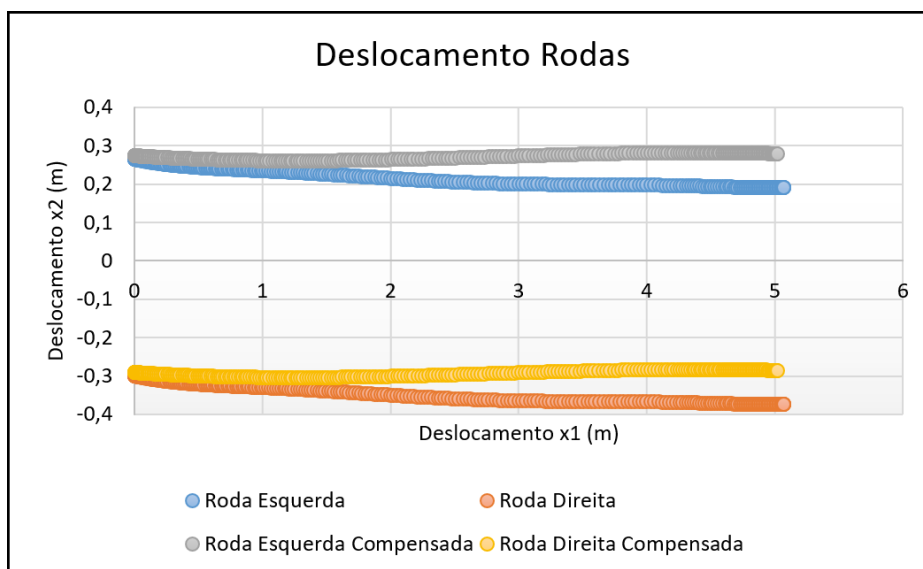


Fonte: Autor (2019).

No Gráfico 3 os comandos direita e esquerda são acionados em momentos que o controlador *fuzzy* detecta a necessidade de realizar a compensação, com a finalidade de reduzir esta variação x_2 permitindo a geração de um movimento retilíneo.

Por fim, no Gráfico 4 foi feita a análise das trajetórias percorridas com e sem a compensação, como a comparação dos resultados para esta mesma situação.

Gráfico 4 – Deslocamento real para cada caso.



Fonte: Autor (2019).

Conforme verificado no Gráfico 4, a implementação da compensação forneceu um melhor resultado no que diz respeito ao deslocamento linear no eixo x1, reduzindo a ocorrência de deslocamentos angulares no eixo x2.

A Tabela 15 apresenta o erro de medição do sistema odométrico, que foi obtido comparando os dados informados pelo sistema e as medições observadas com o auxílio da trena laser.

Tabela 15 – Verificação de erro do sistema odométrico.

DISTANCIA INFORMADA PELO SISTEMA (m)	DISTANCIA TRENA LASER (m)	VARIAÇÃO DE MEDIDA (m)
5,020	4,994	0,02607
5,035	5,052	0,01688
5,034	5,024	0,01022
5,019	4,990	0,02857
5,021	5,010	0,01132
5,022	4,991	0,03092
5,036	5,045	0,894
5,013	4,990	2,272
5,030	5,042	1,157
5,026	5,023	0,348
5,028	5,034	0,606

Fonte: Autor (2019).

O erro relativo entre distância informada pelo sistema e distância medida apresentou resultados satisfatórios, tendo em vista que o maior erro foi de 3,092 cm

para um valor de deslocamento de 5 m, não justificando a necessidade de ajustes no sistema odométrico implementado.

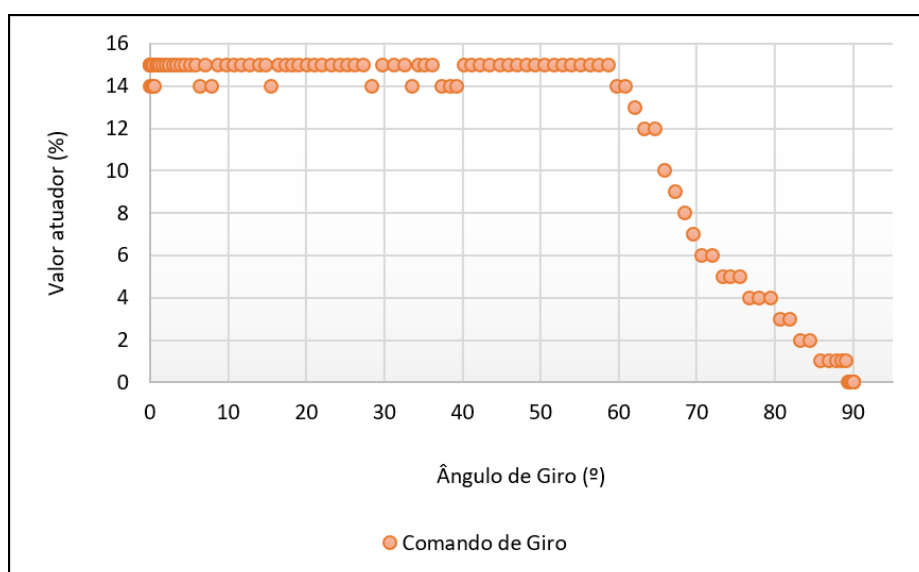
5.3 DESLOCAMENTO ANGULAR

Objetivando a análise da topologia de deslocamento angular proposto, foram realizados testes com deslocamentos com o giro sob o eixo e em curvas, no qual foi realizado um estudo referente a necessidade de implementação de técnicas de correções para possíveis erros existentes neste tipo de movimento. Nesta topologia, para todos os testes, a cadeira foi posta no ponto inicial e determinado um comando para se locomover 90 graus.

5.3.1 Deslocamento angular 90 graus sob o eixo

Para uma movimentação deste tipo é necessária que as rodas da cadeira girem de formas distintas, porém o *fuzzy* para esta topologia de movimento envolve apenas um comando de atuação sendo controlado pelo ângulo de giro conforme apresentado no Gráfico 5.

Gráfico 5 – Comando fuzzy necessários para o movimento em 90 graus sob o eixo.

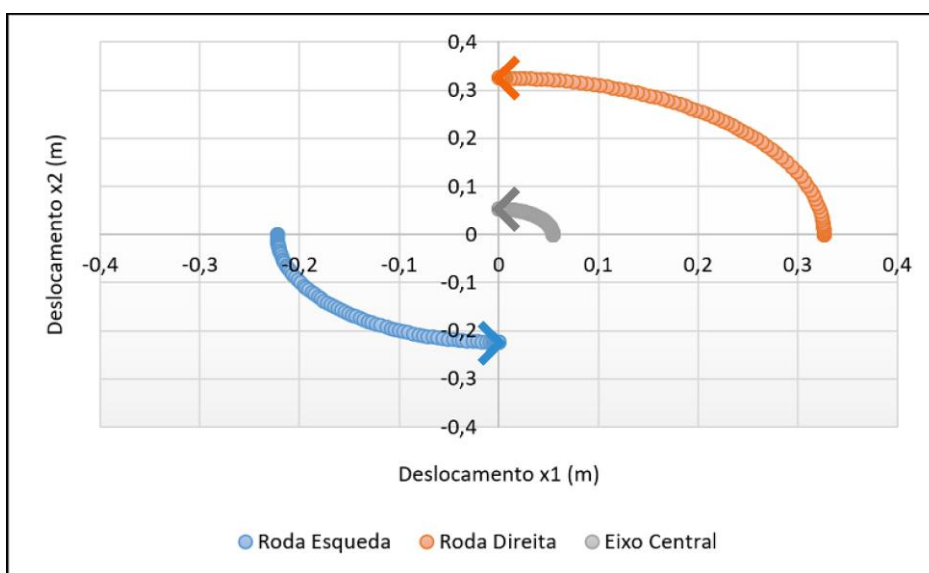


Fonte: Autor (2019).

O comando de giro pode ser definido como comando esquerdo ou direito, onde o controlador fuzzy atua de forma igualitária para os dois casos.

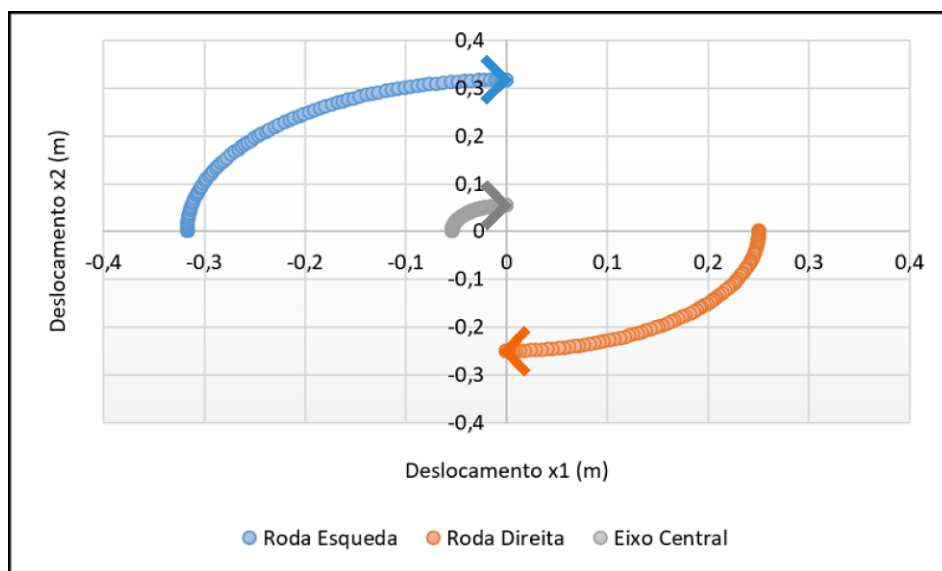
Com base neste comportamento, o deslocamento de rotação para cada lado pode ser traçado e apresentados nos Gráficos 6 e 7.

Gráfico 6 – Rotação sob o eixo para a esquerda.



Fonte: Autor (2019).

Gráfico 7 – Rotação sob o eixo para a direita.



Fonte: Autor (2019).

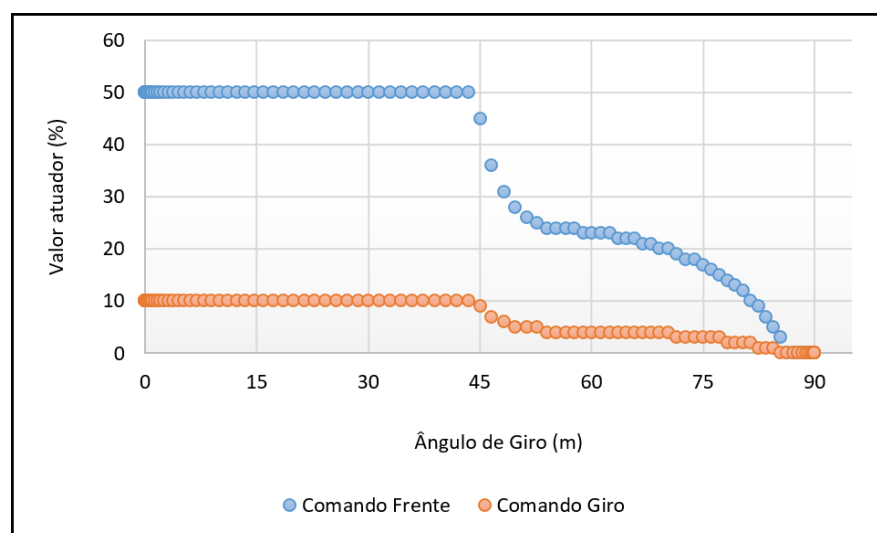
Observa-se que nesses dois deslocamentos apresentados pelos Gráficos 6 e 7, cada roda possui direção rotacional independente conforme indicado pelas setas. Desta forma o movimento que deveria ser puramente angular advindos dos comandos

de giro a esquerda ou giro a direita, não é puramente angular, pois o eixo central acaba sofrendo um leve deslocamento. Essa movimentação não foi considerada como problemática pois o ângulo de rotação definida foi atendido.

5.3.2 Deslocamento angular 90 graus em curva

Nesta topologia de movimento foi necessário realizar o controle sobre o comando frente e o comando de giro simultaneamente conforme apresentado no Gráfico 8 para as regras do controle fuzzy definidas.

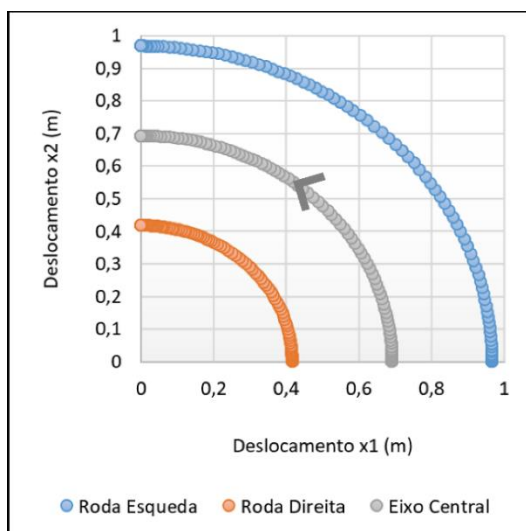
Gráfico 8 – Rotação sobre o eixo para a direita.



Fonte: Autor (2019).

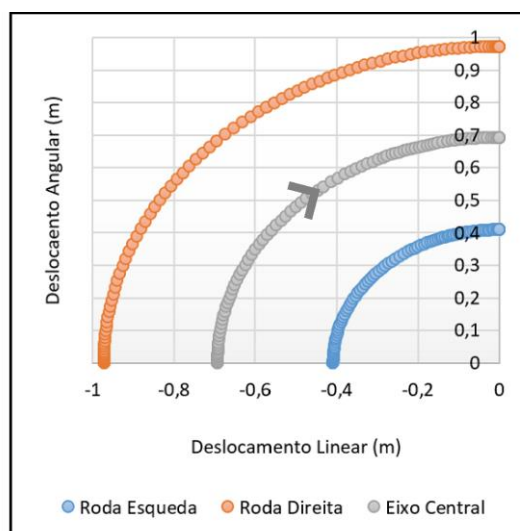
Considerando o ângulo de giro alcançado, o controlador fuzzy praticou a regras pré-definidas, realizando a movimentação à esquerda e à direita de forma satisfatória, conforme indicado nos Gráficos 9 e 10, respectivamente.

Gráfico 9 – Curva 90 graus para esquerda.



Fonte: Autor (2019).

Gráfico 10 – Curva 90 graus para direita.



Fonte: Autor (2019).

Os resultados obtidos nos deslocamentos Gráficos 9 e 10 foram eficazes ao realizarem o deslocamento angular em curva de 90 graus sem erros críticos que impossibilitasse a navegação autônoma.

5.4 RECONHECIMENTO DE MARCOS ARTIFICIAIS

Para o reconhecimento dos marcos artificiais, com o sensor já implementado na cadeira de rodas, foram passadas 10 vezes em cima de cada marco artificial, com e sem iluminação forçada, os resultados podem ser visualizados Tabela 16.

Tabela 16 – Reconhecimento de marcos artificiais.

		Marco Artificial		
		Amarela	Vermelha	Azul
Iluminação natural	Detecções realizadas	8	9	9
	Falhas de detecção	2	1	1
	Falsa detecção	2	0	0
Iluminação forçada	Detecções realizadas	10	10	10
	Falhas de detecção	0	0	0
	Falsa detecção	0	0	0

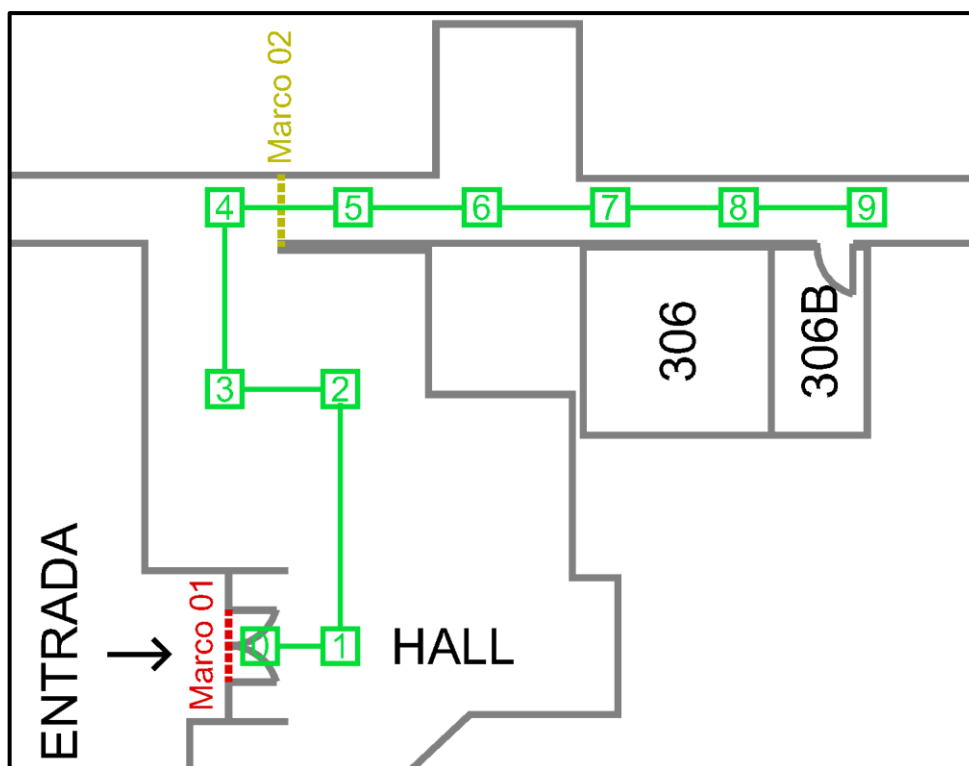
Fonte: Autor (2019).

Conforme observado na Tabela 16, para uma amostragem de 10 verificações, ao utilizar a iluminação natural foram contabilizados um total de 6 erros de detecção, em contrapartida, ao utilizar iluminação forçada não foram contabilizados erros, portanto, para o melhor funcionamento do sistema de marcos artificiais a iluminação forçada se mostrou mais confiável.

5.5 NAVEGAÇÃO AUTÔNOMA ATRAVÉS DA ODOMETRIA

No teste da navegação autônoma, foi definido como destino a sala 306B, desta forma, a partir do ponto de início 0, a trajetória foi definida e assim sendo gerado a rota até o ponto 9, o trajeto pode ser visualizado na Figura 27.

Figura 27 – Trajeto gerado até sala 306B.



Fonte: Autor (2019).

Durante a trajetória, ao identificar o marco artificial 02 a trajetória foi corrigida em um metro evidenciando a correta aplicação deste método de correção. No restante do trajeto a cadeira de rodas autônoma realizou o deslocamento de forma correta identificando o ponto de parada em frente à sala 306B, a qual havia sido definida como destino.

6 CONCLUSÕES

Através dos resultados obtidos foi possível validar o uso de técnicas de navegação, especificamente a odometria com o uso de sensores do tipo encoder. Em auxílio a esta técnica foi utilizado o método de correção da posição baseada em identificação de marcos artificiais identificados a partir de um sensor RGB e pisos táteis coloridos.

A partir dos testes de deslocamento retilíneo com o uso da odometria, pode-se perceber a necessidade da correção de erros da trajetória provenientes das diferenças dinâmicas dos motores da cadeira de rodas. Estes erros foram corrigidos inserindo novas regras no controlador *fuzzy* a fim de igualar o deslocamento realizado pelas duas rodas com o uso das informações lidas pelos *encoders*, e o erro máximo encontrado para um deslocamento de 5 metros foi de 3,092 centímetros.

No deslocamento angular foi verificado dois comportamentos distintos para os dois tipos de deslocamento angular. No deslocamento angular sob o eixo foi constatado uma divergência do eixo de rotação da cadeira fora do esperado, porém esta divergência não apresentou impedimentos no cumprimento do ângulo determinado de movimentação. Já no deslocamento em curva o sistema se comportou conforme o esperado não necessitando ajustes.

O algoritmo de geração de trajetória foi aplicado de acordo com o método de Dijkstra, requerendo um grande número de interações para que a trajetória seja calculada, porém, devido a trajetória ser calculada no início do deslocamento não interferiu no processamento dos outros sistemas implementados.

Para a navegação autônoma realizada até a sala 306B a cadeira chegou até o destino com êxito devido a aplicação da técnica odométrica juntamente com a técnica de controle *fuzzy* aplicadas nos motores da cadeira, para a correção da posição foi utilizada a técnica de de marcos artificiais. Assim sendo constatado que as premissas iniciais para aquisição de dados, métodos de controle e de navegação autônoma tiveram um comportamento conforme o esperado.

Uma possível melhoria a ser implementada no sistema é a aplicação de um método de desvio de obstáculos estáticos e dinâmicos como a implementação de sensores de ultrassom por exemplo, possibilitando o recálculo da rota especificada caso necessário.

REFERÊNCIAS BIBLIOGRÁFICAS

BEN-ARI, Mordechai; MONDADA, Francesco. **Elements of Robotics**. Switzerland: Springer, 2018 324 p. ISBN 978-3-319-62532-4.

BORENSTEIN, Johann, [ed]; EVERRET, H. R.; FENG, Liqiang. **Where am I? Sensors and Methods for Mobile Robot Positioning**. [s.l.]: The University of Michigan, 1996 282 p.

ENCARNAÇÃO, Pedro, [ed.]; COOK, Albert M., [ed.]. **Robotic Assistive Technologies: Principles and Practice**. Boca Raton, EUA: CRC-Press, 2017 381 p. ISBN 978-1-4987-4572-7.

ESPRESSIF, Wi-Fi and Bluetooth chipsets and solutions, 2019. Disponível em: <<https://www.espressif.com/en/products/hardware/esp32/overview>>. Acesso em 18/09/2019.

ESPRESSIF, Wi-Fi and Bluetooth chipsets and solutions, 2019. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em 15/09/2019.

FAHIMI, Farbod. **Autonomous Robots: Modeling, Path Planning, and Control**. New York: Springer, 2009. ISBN 978-0-387-09537-0.

FREEDOM, Cadeiras de rodas e veículos elétricos, 2019. Disponível em: <<http://www.freedom.ind.br/produto/saude/cadeiras-de-rodas-motorizadas/freedom-compact-13/>>. Acesso em 10/08/2019.

GERSDORF, Bernd; FRESE, Udo. **A Kalman Filter for Odometry using a Wheel Mounted Inertial Sensor**. Bremen: German Research Center for Artificial Intelligence.

HOLLAND, John M. **Designing Autonomous Mobile Robots**. s.l.: Elsevier, 2004 335 p. ISBN 0-7506-7683-3.

KORDIĆ, Vedran. **Kalman Filter**. Vukovar, Croatia: Intech, 2010 390 p. ISBN 978-953-307-094-0.

LABBE, Roger R.. **Kalman and Bayesian Filters In Python**. [s.l.]: GitHub, 2014 262 p.

MATSUMOTO, Osamu et al.. **Autonomous Traveling Control of the "TAO Aidce" Intelligent Wheelchair**. Beijing: International Conference on Intelligent Robots and Systems, 2006.

RASPBERRYPI, Raspberry Pi Foundation, 2019. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>>. Acesso em 01/09/2019.

RASPBIAN, free operating system based on Debian optimized for the Raspberry Pi hardware, 2019. Disponível em: <<https://www.raspbian.org/>>. Acesso em 02/09/2019.

SICILIANO, Bruno, [ed]. **Springer Handbook of Robotics**. [s.l.] : Springer, 2008. ISBN 978-3-540-23957-4.

SICILIANO, Bruno et al. **Robotics: Modelling, Planning and control**. London: Springer-Verlag London, 2009 632 p. ISBN 978-1-84628-641-4

SIEGWART, Roland; Nourbakhsh, Illah R.; Scaramuzza, Davide. **Introduction to Autonomous Mobile Robots**. 2. ed. atual Massachusetts: MIT Press, 2011 453 p. ISBN 978-0262015356.

WEBSTER, John G., [ed.]. **The Measurement, Instrumentation, and Sensors Handbook**. [s.l]: CRC-Press, 1999. ISBN 0-8493-8347-1.

WEE-KIAT, Danny et al.. **Development of an Autonomous BCI Wheelchair**. [s.l]: Transactions on Neural Systems and Rehabilitation Engineering, 2014.

WELCH, Greg; BISHOP, Gary. **An introduction to the Kalman filter**. Chapel Hill, NC, USA: University of North Carolina, 2006.

YOKOZUKA, Masashi et al.. **Robotic Wheelchair with Autonomous Traveling Capability for Transportation Assistance in an Urban Environment**. Vilamoura: International Conference on, 2012.

ZHANG, Rui et al.. **Control of a Wheelchair in an Indoor Environment Based on a Brain-Computer Interface and Automated Navigation**. [s.l]: Transactions on Neural Systems and Rehabilitation Engineering, 2016.

ANEXOS

ANEXO A – CARACTERÍSTICAS DA TRENA A LASER GLM 40 BOSCH

Dados técnicos

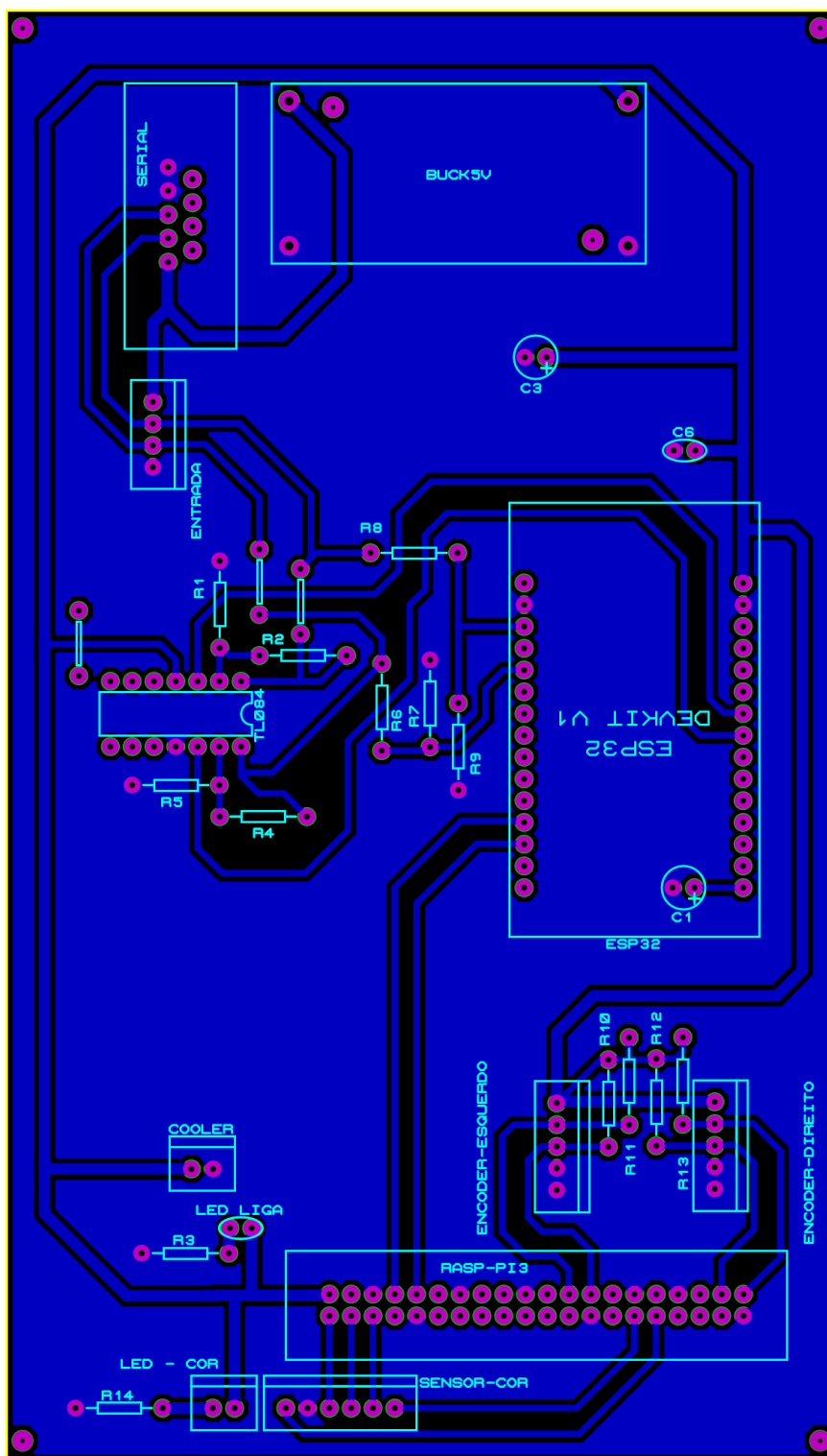
Dados adicionais

Díodo laser*	635 nm, < 1 mW
Faixa de medição*	0,15 – 40,00m
Peso aprox.*	0,09kg
Tempo de medição, típico	< 0,5 s
Faixa de medição, de	0,15m
Faixa de medição, até	40m
Classe de laser	2
Precisão de medição, típica	± 1,5 mm
Tempo de medição máx.	4s
Alimentação elétrica	2 x 1,5 V LR03 (AAA)
Dispositivo de desconexão automática	5min
Unidades de medição	m/cm, pés/polegada
Capacidade da memória (valores)	10
Proteção contra pó e projeções de água	IP 54



APÊNDICES

APÊNDICE A – PCB DESENVOLVIDA PARA INTERLIGAÇÃO DOS EQUIPAMENTOS



APÊNDICE B – TESTES REALIZADOS NA RASPBERRY PI 3 B+

Testes realizados

Teste	Variável de teste	Descrição do teste
Leitura de sinais digitais	10 kHz	Detecção de 100000 bordas de subida
	20 KHz	Detecção de 200000 bordas de subida
	50 KHz	Detecção de 500000 bordas de subida
	100 KHz	Detecção de 1000000 bordas de subida
Geração da trajetória pelo algoritmo de Dijkstra	10 Quadrantes	Análise do tempo necessário para executar cada operação
	100 Quadrantes	
	1000 Quadrantes	

APÊNDICE C – TESTES REALIZADOS NO ESP32

Nível percentual do sinal	DAC Velocidade Linear Esperada (V)		DAC Velocidade Linear Esperada (V)	
	Frente	Ré	Direita	Esquerda
10	2,75	1,95	2,75	1,95
20	2,80	1,90	2,80	1,90
30	2,85	1,85	2,85	1,85
40	2,90	1,80	2,90	1,80
50	2,95	1,75	2,95	1,75
60	3,00	1,70	3,00	1,70
70	3,05	1,65	3,05	1,65
80	3,10	1,60	3,10	1,60
90	3,15	1,55	3,15	1,55
100	3,20	1,50	3,20	1,50