

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA – HABILITAÇÃO EM  
ROBÓTICA E AUTOMAÇÃO INDUSTRIAL**

**RAPHAELA MANSUR JOSÉ**

**UTILIZAÇÃO DE LÓGICA *FUZZY* E DIAGRAMA DE VORONOI PARA  
CONTROLAR UM ROBÔ MÓVEL ASSISTIVO**

**JUIZ DE FORA**

**2022**

RAPHAELA MANSUR JOSÉ

**UTILIZAÇÃO DE LÓGICA *FUZZY* E DIAGRAMA DE VORONOI PARA  
CONTROLAR UM ROBÔ MÓVEL ASSISTIVO**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica – Habilitação em Robótica e Automação Industrial da Universidade Federal de Juiz de Fora, como requisito parcial para a obtenção do título de Engenheira Eletricista.

Orientador: Prof. Leonardo Rocha Olivi

**JUIZ DE FORA**

**2022**

***Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)***

Mansur José, Raphaela.

UTILIZAÇÃO DE LÓGICA FUZZY E DIAGRAMA DE VORONOI  
PARA CONTROLAR UM ROBÔ MÓVEL ASSISTIVO / Raphaela  
Mansur José. -- 2022.

72 f.

Orientador: Leonardo Rocha Olivi  
Trabalho de Conclusão de Curso (graduação) - Universidade  
Federal de Juiz de Fora, Faculdade de Engenharia, 2022.

1. Lógica Fuzzy. 2. Diagrama de Voronoi. 3. Controle. I. Rocha  
Olivi, Leonardo, orient. II. Título.



## ATA DE APRESENTAÇÃO DE TRABALHO FINAL DE CURSO

DATA DA DEFESA: **24/02/2022**

CANDIDATA: **RAPHAELA MANSUR JOSÉ** ORIENTADOR: **PROF. LEONARDO ROCHA OLIVI**

TÍTULO DO TRABALHO: **UTILIZAÇÃO DE LÓGICA FUZZY E DIAGRAMA DE VORONOI PARA CONTROLAR UM ROBÔ MÓVEL ASSISTIVO**

BANCA EXAMINADORA/INSTITUIÇÃO:

PRESIDENTE: **PROF. LEONARDO ROCHA OLIVI – UNIVERSIDADE FEDERAL DE JUIZ DE FORA**

AVALIADOR: **PROF. EXUPERRY BARROS COSTA – UNIVERSIDADE FEDERAL DE JUIZ DE FORA**

**1 LOCAL: AMBIENTE VIRTUAL (ERE – ENSINO EMERGENCIAL REMOTO EM FUNÇÃO DA COVID19)**

Nesta data, em sessão pública, após exposição oral de \_\_\_\_\_ minutos, o candidato foi arguido pelos membros da banca. Em decorrência desta arguição, a banca considerou o candidato:

(  ) APROVADO

(  ) REPROVADO

Na forma regulamentar foi lavrada a presente Ata que é abaixo assinada pelos membros da banca na ordem determinada e pelo candidato:

PRESIDENTE: Leonardo Rocha Olivi

AVALIADOR: Exuperry Barros Costa

CANDIDATO: Raphaela Mansur José

## AGRADECIMENTOS

Agradeço, primeiramente, à Deus, pois sem Ele eu nada seria. Em segundo lugar, mas não menos importante, à minha família pelo apoio incondicional. Vocês sorriram e choraram comigo ao longo de toda a graduação, me ajudaram a me levantar quando eu acreditava ser impossível. Meu eterno agradecimento não será suficiente para reconhecer todo o amor e força que minha mãe, Rogéria, meu pai, Fernando, minha irmã, Lais e minha amada vó Elza me deram. E, claro, agradeço à minha tia Vani, que sempre me ajudou em tudo. Saibam que essa vitória é tão minha quanto de vocês e eu amo muito cada um.

Essa jornada de vitórias e derrotas também não poderia ter sido concluída sem a ajuda e atenção do meu querido orientador Leonardo Olivi, do professor Exuperry e do professor David. Foram vocês que me motivaram a seguir caminhando na engenharia e me motivaram a acreditar em mim mesma. Espero sinceramente que outros alunos possam ter professores e mentores tão bons quanto vocês foram pra mim. Acima de tudo, obrigada pela paciência nos momentos de surto.

A todos os meus amigos e familiares, obrigada pela compreensão quando eu precisava me fazer ausente e por me ajudar a me divertir muito quando eu podia fazer parte do momento. Eu sei que vocês torciam por mim e me ajudavam em tudo que podiam, e até no que não podiam. Obrigada por atender telefonemas desesperados na madrugada, mensagens surtadas e até por estudar comigo pras provas, mesmo sendo de áreas completamente diferentes e odiarem matemática. Amo vocês.

Por fim, agradeço a toda equipe e famílias participantes deste projeto tão lindo chamado *Adapt*. Que possamos beneficiar cada vez mais pessoas!

Se hoje cheguei aqui, é porque cada pessoa que fez parte da minha história contribuiu pra isso. Obrigada a todos que escolheram fazer parte dela.

## RESUMO

Este trabalho tem por objetivo modelar um controlador que utilize a lógica *Fuzzy* e o Diagrama de Voronoi para auxiliar uma criança com paralisia cerebral a conduzir um robô diferencial. Foram modeladas duas cenas no *software* CoppeliaSIM contendo obstáculos tanto no plano XY quanto no eixo Z. Os obstáculos foram avaliados em termos de distância relativa ao robô, posicionamento em relação ao mesmo e se podiam ser detectados ou não. Também foram atribuídas três zonas para atuação do controle, sendo elas a zona segura, na qual o usuário gozava de total controle, a zona de atenção e a zona de risco. Nestas últimas atuavam o controlador *fuzzy* e o controle desenvolvido a partir do Diagrama de Voronoi, respectivamente. Os testes foram realizados tanto com o usuário tentando constantemente seguir em frente, quanto com a geração de comandos aleatórios incluindo ré, virar a direita ou a esquerda. Concluiu-se que a aplicação combinada de um controlador *fuzzy* e outro com base no Diagrama de Voronoi ofereceu uma redução de risco de colisão ou queda satisfatória para os ambientes testados, incluindo passagens por portas e corredores. Contudo, diante de cenários mais complexos, este controlador pode não apresentar uma resposta tão interessante, necessitando de refinamento.

**Palavras-chave:** 1. Controle Compartilhado. 2. Lógica *Fuzzy*. 3. Diagrama de Voronoi.

## **ABSTRACT**

This work aims to model a controller that uses Fuzzy logic and the Voronoi Diagram to help a child with cerebral palsy to drive a differential robot. Two scenes were modeled in CoppeliaSIM software containing obstacles both in the XY plane and in the Z axis. The obstacles were evaluated in terms of relative distance to the robot, positioning in relation to it and whether they could be detected or not. Three zones were also assigned to control performance, named the safe zone, in which the user had full control, the attention zone and the risk zone. In each of the latter, the fuzzy controller and the control developed based on the Voronoi Diagram acted, respectively. Tests were performed both with the user constantly trying to move forward, and with the generation of random commands including reverse, turn right or left. It was concluded that the combined application of a fuzzy controller and another based on the Voronoi Diagram offered a satisfactory reduction in the risk of collision or fall for the tested environments, including passages through doors and corridors. However, in the face of more complex scenarios, this controller may not present such an interesting response, requiring refinement.

**Keywords:** 1. Shared Control. 2. Fuzzy Logic. 3. Voronoi Diagram.

## LISTA DE FIGURAS

Figura 1: Controle Compartilhado. Fonte: ROMERO, 2014.....	14
Figura 2: Comparação entre uma função característica da lógica crisp e uma função de pertinência da lógica <i>fuzzy</i> . Fonte: Autora.....	16
Figura 3: Formato de conjunto linear. Fonte: Autora. ....	18
Figura 4: Formato de conjunto triangular. Fonte: Autora. ....	19
Figura 5: Formato de conjunto trapezoidal. Fonte: Autora.....	20
Figura 6: Formato de conjunto Gaussiano. Fonte: Autora. ....	21
Figura 7: Representação gráfica das operações em <i>fuzzy</i> dos conjuntos A e B. (a) conjuntos A e B; (b) união; (c) interseção; (d) complemento. Fonte: SILVA, 2013 adaptado de FERNANDES, 2005. ....	22
Figura 8: Exemplo de aplicação do método Mamdani. Fonte: (Mamdani, 1975). ....	24
Figura 9: Ilustração do Diagrama de Voronoi para 10 pontos randômicos contidos no intervalo de 0 a 1. Fonte: Autora. ....	27
Figura 10: Visão da <i>toolbox</i> ofertada pelo MATLAB para implementação da lógica <i>fuzzy</i> . Fonte: Autora.....	30
Figura 11: A figura (a) ilustra a imagem do robô Pioneer p3dx no menu de seleção de robôs móveis do CoppeliaSIM, enquanto a figura (b) indica as posições de instalação dos sensores horizontais de acordo com o manual do fabricante. Fonte: Autora. ....	31
Figura 12: A Figura (A) mostra o posicionamento dos sonares frontais que realizam a leitura do chão, enquanto a figura (B) ilustra os sonares traseiros. Fonte: Autora....	32
Figura 13: Ilustração da divisão de zonas horizontais do robô. Fonte: Autora.....	33
Figura 14: Zonas de Segurança definidas para a detecção de obstáculos. A zona de segurança corresponde a cor verde, a zona de atenção a cor amarelo escuro e a zona de risco a cor vermelha. Fonte: Autora.....	34
Figura 15: <i>Toolbox</i> modelada para implementação no controlador utilizando 4 variáveis de entrada e 4 variáveis de saída. Fonte: Autora. ....	38
Figura 16: Funções de pertinência da variável de entrada "frente". Fonte: Autora. ...	39
Figura 17: Funções de pertinência da variável de entrada "re". Fonte: Autora. ....	39
Figura 18: Funções de pertinência da variável de entrada "direita". Fonte: Autora...	40
Figura 19: Funções de pertinência da variável de entrada "direita". Fonte: Autora...	40
Figura 20: Funções de pertinência da variável de saída "velocidade_linear_frente". Fonte: Autora. ....	41



Figura 21: Funções de pertinência da variável de saída "velocidade_angular_direita". Fonte: Autora. ....	42
Figura 22: Funções de pertinência da variável de saída "velocidade_linear_re". Fonte: Autora. ....	42
Figura 23: Funções de pertinência da variável de saída "velocidade_angular_esquerda". Fonte: Autora. ....	43
Figura 24: Ilustração da utilização do Diagrama de Voronoi para obtenção do ponto objetivo. Fonte: Autora. ....	46
Figura 25: Cena 1 criada no CoppeliaSIM. Fonte: Autora. ....	47
Figura 26: Cena 2 criada no CoppeliaSIM. Fonte: Autora. ....	48
Figura 27: Ponto inicial da trajetória do robô. Fonte: Autora. ....	49
Figura 28: Ponto final onde a simulação foi pausada. Em vermelho tem-se a trajetória do robô. Fonte: Autora. ....	50
Figura 29: O obstáculo encontra-se dentro da zona de atenção, porém, a própria criança está se retirando do perigo e não há necessidade para atuação do <i>fuzzy</i> . Fonte: Autora. ....	53
Figura 30: Situação 2 - Navegação em um corredor com um desnível profundo ao final. Fonte: Autora. ....	53
Figura 31: Momento em que o robô retorna ao salão após o percurso. Fonte: Autora. .....	54
Figura 32: Plot do mapa local de obstáculos associado aos vértices do Diagrama de Voronoi traçado. O ponto objetivo está identificado de vermelho. Fonte: Autora.....	57
Figura 33: Situação para a qual o protocolo de contingência é acionado. Fonte: Autora. .....	58
Figura 34: Trajeto realizado pelo robô evidenciado o retorno realizado diante do obstáculo no chão. Fonte: Autora.....	58
Figura 35: Plotagem do mapa local evidenciando apenas um obstáculo de solo e a ausência de vértices do Diagrama de Voronoi. Fonte: Autora.....	61
Figura 36: (A) Momento do início da trajetória. (B) Momento em que o programa foi finalizado, evidenciando a trajetória realizada. Fonte: Autora.....	62

## LISTA DE TABELAS

Tabela 1: Exemplos de situações enfrentadas pelo usuário e a resposta dada pelo controlador.....	36
--	----

## LISTA DE GRÁFICOS

Gráfico 1: Distância mínima dos objetos detectados em relação ao robô durante o tempo da simulação. Fonte: Autora.....	51
Gráfico 2: Velocidades lineares aplicadas ao robô ao longo da trajetória comparadas às velocidades dadas pelo usuário e pelo controlador <i>fuzzy</i> . Fonte: Autora.....	51
Gráfico 3: Velocidades angulares aplicadas ao robô ao longo da trajetória comparadas às velocidades dadas pelo usuário e pelo controlador <i>fuzzy</i> . Fonte: Autora.....	52
Gráfico 4: Distância mínima para os objetos detectados. Fonte: Autora .....	55
Gráfico 5: Velocidades lineares obtidas durante o percurso. Fonte: Autora. ....	55
Gráfico 6: Velocidades angulares obtidas no trajeto. Fonte: Autora. ....	56
Gráfico 7: Distâncias mínimas detectadas mostrando que não houve detecção de objetos na horizontal. Fonte: Autora.....	59
Gráfico 8: Velocidades lineares associadas ao trajeto. Fonte: Autora. ....	59
Gráfico 9: Velocidades angulares associadas ao trajeto. Fonte: Autora. ....	60
Gráfico 10: Distâncias mínimas detectadas em relação aos objetos na horizontal ao longo da trajetória. Fonte: Autora. ....	62
Gráfico 11: Velocidades lineares atribuídas ao longo da trajetória. Fonte: Autora....	63
Gráfico 12: Velocidades angulares atribuídas ao longo da trajetória. Fonte: Autora.	63

## LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

<b>PC</b>	<b>Paralísia Cerebral</b>
-----------	---------------------------

## SUMÁRIO

<b><u>1 LOCAL: AMBIENTE VIRTUAL (ERE – ENSINO EMERGENCIAL REMOTO EM FUNÇÃO DA COVID19)</u></b>	<b>4</b>
<b><u>2 INTRODUÇÃO</u></b>	<b>11</b>
<b><u>3 FUNDAMENTAÇÃO TÉCNICA</u></b>	<b>13</b>
3.1 CONTROLE COMPARTILHADO	13
3.2 LÓGICA <i>FUZZY</i>	15
3.2.1 CONJUNTO <i>FUZZY</i>	16
3.2.2 FUNÇÕES DE PERTINÊNCIA	17
3.2.3 PROPRIEDADES	21
3.2.4 OPERAÇÕES	22
3.2.5 REGRAS E MÉTODOS DE INFERÊNCIA	23
3.2.6 DEFUZZIFICAÇÃO	25
3.3 DIAGRAMA DE VORONOI	26
<b><u>4 DESENVOLVIMENTO</u></b>	<b>29</b>
4.1 OBJETIVO	29
4.2 SENSORIAMENTO E DETERMINAÇÃO DAS ZONAS DE SEGURANÇA	30
4.3 CONTROLADOR <i>FUZZY</i>	37
4.4 CONTROLADOR BASEADO NO DIAGRAMA DE VORONOI	45
<b><u>5 RESULTADOS E DISCUSSÕES</u></b>	<b>47</b>
5.1 PRIMEIRA SITUAÇÃO: OBSTÁCULOS E PASSAGEM PELA PORTA	49
5.2 SEGUNDA SITUAÇÃO: CORREDOR E DESNÍVEL	53
5.3 TERCEIRA SITUAÇÃO: DESNÍVEL SEM A PRESENÇA DE OBSTÁCULOS HORIZONTAIS	57
5.4 QUARTA SITUAÇÃO: TODOS OS CONTROLES DE COMANDO HABILITADOS	61
<b><u>6 CONCLUSÕES</u></b>	<b>65</b>
<b><u>REFERÊNCIAS</u></b>	<b>66</b>

## 2 INTRODUÇÃO

A Robótica Assistiva é um campo da robótica que desenvolve tecnologias que permitam ou facilitem as interações humanas com o ambiente e a execução de tarefas. Dentro deste campo, uma das linhas de atuação mais nobres se encontra na melhora da qualidade de vida de pessoas com algum grau de deficiência motora e/ou neurológica (OLIVI *et al.* 2013).

Considerando que a acessibilidade é uma necessidade básica dos indivíduos, é prudente afirmar que uma melhora da qualidade de vida de indivíduos que apresentem limitações fisiológicas ou mentais impacta não somente as suas próprias vidas, mas a de todos em seu convívio, principalmente tratando-se de crianças e idosos.

Uma das patologias neurológicas permanentes mais comuns na infância é a Paralisia Cerebral (PC), causada por uma encefalopatia não progressiva nos períodos pré, peri ou pós-natal. Esta encefalopatia conduz a um quadro clínico de distúrbios de motricidade com alterações de movimento, postura, equilíbrio e a presença de movimentos involuntários. Essas desordens motoras podem ainda ser acompanhadas de distúrbios sensoriais, perceptivos, cognitivos, de comunicação e comportamentais (CANS *et al.*, 2007; O'SHEA, 2008).

Estima-se que a incidência de PC está entre 1,5 a 5,9 por 1000 nascidos vivos nos países desenvolvidos e em países em desenvolvimento este valor é ainda maior, correspondendo a 7 por 1000. Segundo a cartilha do Ministério da Saúde do Brasil, denominada "Diretrizes de Atenção à Pessoa com Paralisia Cerebral" publicada em 2013, ainda há uma carência de estudos no Brasil que investiguem a prevalência e incidência da PC no país (BRASIL, 2013).

Diante do cenário apresentado, o desenvolvimento de um carro diferencial que auxilie na locomoção do indivíduo e ainda ofereça possibilidades fisioterápicas para o mesmo se torna extremamente interessante, pois se torna uma ferramenta de promoção de mobilidade e desenvolvimento de habilidades psicomotoras (EVANS & BAINES, 2017; LIVINGSTONE & FIELD, 2014). Contudo, devido a fragilidade desta criança, a preocupação com a segurança da mesma é alta, exigindo adaptações físicas do carrinho e tornando a aplicação de um sistema de controle compartilhado prioritária.

Por outro lado, se a segurança é prioritária, também o é o desenvolvimento da criança. Essa ponderação levanta a questão de qual controlador utilizar para permitir que a vontade da criança seja cumprida, mas de forma a não colocá-la em risco, principalmente considerando a proposta fisioterápica na qual esse estudo também está inserido.

Inspirado no projeto *Go Baby Go* fundado pelo professor Cole Galloway na Universidade de Delaware (GALLOWAY, 2022), foi criado o projeto de extensão e treinamento profissional chamado *Adapt* na Universidade Federal de Juiz de Fora no ano de 2021. Este projeto visa oferecer mobilidade a crianças com Paralisia Cerebral através do uso compartilhado de carrinhos adaptados. Esses mesmos carrinhos são emprestados para as crianças para que as mesmas possam usá-los no seu dia-a-dia e também como ferramentas de trabalho em sessões de fisioterapia acompanhadas por fisioterapeutas qualificados.

Considerando o escopo do projeto, duas soluções interessantes para a modelagem do controle compartilhado são a utilização da Lógica *Fuzzy*, que busca apresentar uma solução algorítmica que se assemelhe ao pensamento humano, e a aplicação do Diagrama de Voronoi, o qual mostra as áreas seguras de um mapa, uma vez que os obstáculos presentes no mesmo são identificados (MAMDANI & ASSILIAN, 1975; BHATTACHARYA & GAVRILOVA, 2007).

Neste trabalho será proposto um Controle Compartilhado para o robô móvel que carrega uma criança que possui paralisia cerebral. Este controle será baseado na combinação da Lógica *Fuzzy* e do Diagrama de Voronoi, em diferentes níveis de segurança, para minimizar situações de risco a que uma criança com PC possa ser exposta, mas ainda promovendo sua mobilidade de forma segura e apoiando o processo de fisioterapia.

### 3 FUNDAMENTAÇÃO TÉCNICA

Este capítulo se destina a uma breve revisão sobre as técnicas utilizadas neste trabalho sendo elas o Controle Compartilhado e as estratégias de decisão baseadas em Lógica *Fuzzy* e o Diagrama de Voronoi.

#### 3.1 CONTROLE COMPARTILHADO

Uma arquitetura de controle é uma forma de estruturar um sistema de controle, a partir da definição individual do projeto dos conceitos de Percepção, Planejamento e Atuação (ROMERO, 2014, p. 47).

A percepção está relacionada a escolha, aplicação e integração de sensores para extração de informações úteis sobre o meio em que o robô está inserido para que seja feita a modelagem e reconhecimento do mundo real. A partir das informações coletadas pelos sensores, é possível realizar o planejamento, o qual envolve o planejamento de tarefas, a sincronização e o monitoramento da execução de toda a atividade do robô. Atividades estas que serão executadas de forma controlada pelos atuadores na etapa de atuação.

A definição de como serão dadas as interações entre os comandos dados por um usuário através de uma interface e a resposta dos atuadores após a avaliação dos dados enviados pelos sensores determina o grau de automação do robô e o tipo de arquitetura de controle. Estas interações podem ser do tipo deliberativas, reativas ou ambas e determinam se o controle será do tipo reativo, híbrido ou deliberativo (MATARIC, 2014).

É importante ressaltar que tratamos deliberação como a tomada de decisão após análise dos dados modelados. Dessa forma, um controle totalmente reativo seria um controle operado em uma curta escala de tempo reagindo à informação sensorial atual, enquanto um controle deliberativo corresponderia a um conjunto de regras estabelecidas a partir da definição prévia de um mapa, ou seja, considera informações passadas também.

A arquitetura híbrida, a qual é uma combinação das arquiteturas reativa e deliberativa foi desenvolvida para permitir a interação entre o usuário e o robô por ele operado em diferentes níveis de autonomia.



Assim, um robô autônomo realiza a tomada de decisão, de forma reativa ou deliberativa, sem que o usuário tenha qualquer capacidade de interferência na mesma. Por sua vez, o robô teleoperado apenas replica os comandos dados pelo usuário de forma deliberativa, sem interferir na tomada de decisão. O meio termo entre as situações é um robô com controle compartilhado no qual a tomada de decisão é ponderada entre o que foi deliberado pelo usuário e um conjunto pré-programado de restrições e ações que o robô deve seguir de forma reativa.

A Figura 1 ilustra uma arquitetura híbrida com controle compartilhado que permite que um operador humano interaja com um robô em diferentes níveis de controle para executar uma tarefa de navegação.

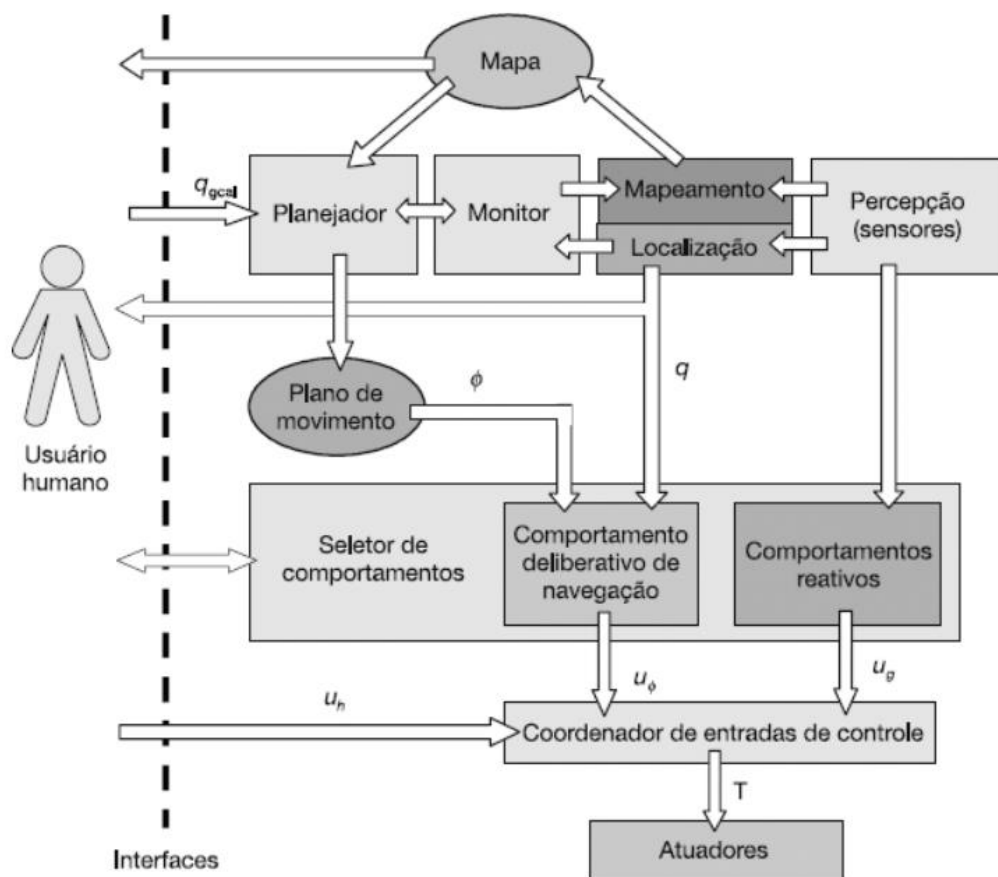


Figura 1: Controle Compartilhado. Fonte: ROMERO, 2014.

Leis de controle assistivas tem por objetivo dividir a responsabilidade da execução da tarefa entre o usuário que envia o comando do que deve ser feito e o robô que analisa se esse comando pode ser executado de acordo com o conjunto de regras pré-estabelecidas por um programador (OLIVI *et al.*, 2013).

Um exemplo de utilização de controle compartilhado para execução de trajetórias é o trabalho realizado por OLIVI *et al.*, 2013, cujo controle é realizado a partir da aplicação de campos vetoriais à uma cadeira de rodas robótica.

O trabalho realizado demonstra que o controle compartilhado nessa situação assiste o usuário na realização de uma trajetória bem-sucedida, evitando obstáculos que possam ser identificados pelos sensores e que poderiam ser atingidos caso o usuário, que possui funções motoras restritas, tentasse ir de encontro a eles. Neste caso, o controle compartilhado oferece a liberdade do deslocamento aliada a maximização da segurança na performance do trajeto.

Com premissas semelhantes, porém utilizando a Lógica *Fuzzy* para criação do controle compartilhado, os autores de BARBOSA *et al.*, 2019, desenvolveram um controlador fuzzy utilizando uma simulação em MATLAB de um ambiente 2-D, para que o robô utilizado na simulação performasse uma trajetória de forma eficiente e evitando obstáculos, reduzindo, assim, os riscos de colisões.

### 3.2 LÓGICA FUZZY

A modelagem de um problema que envolva dados ou soluções com valores intermediários pode ser complexa quando a lógica clássica, também chamada de lógica *crisp* (ZADEH, 1965), é aplicada, uma vez que a mesma atua apenas com os estados verdadeiro ou falso.

Como uma proposta de solução para esse tipo de problema, os conceitos relacionados a lógica *fuzzy* foram primeiramente introduzidos por Lofti Zadeh em 1965 (ZADEH, 1965) como elementos pertencentes a um conjunto *fuzzy* e apresentando um certo grau de pertinência. Assim, obtém-se um método que traduz expressões verbais qualitativas e imprecisas, comumente utilizadas na comunicação humana, em valores numéricos (SIMÕES, 2007).

Essas características permitem a modelagem de problemas complexos, simulando o processo de tomada de decisão de forma mais simples, apoiado no conhecimento de múltiplos especialistas envolvidos no processo modelado. Além disso, pode-se manipular incertezas de forma consistente e matemática (ZADEH, 1965; ZADEH, 1999).

### 3.2.1 Conjunto *Fuzzy*

A lógica clássica utiliza a teoria de conjuntos para determinar se um elemento  $x$  pertence ou não a determinado conjunto  $A$  a partir de uma função de pertinência  $\mu_A(x)$ . Assim, tem-se:

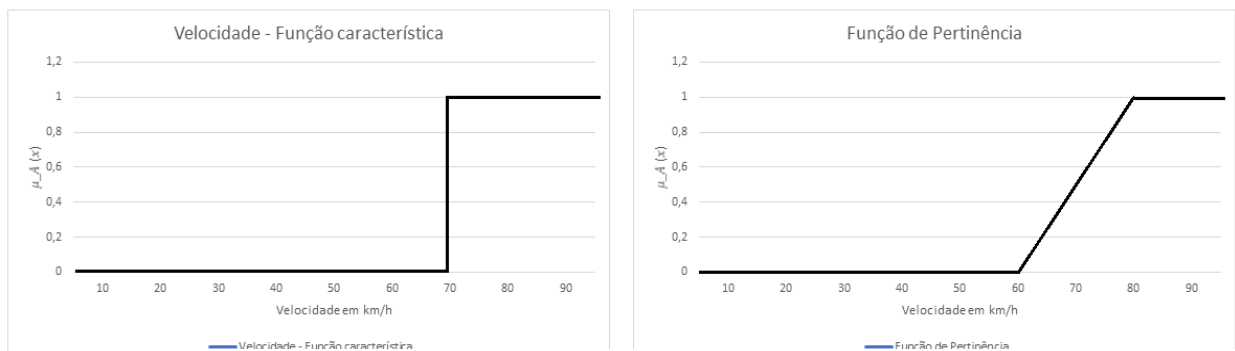
$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (1.1)$$

Quando tentamos aplicar essa ideia ao mundo real, percebemos a limitação dessa lógica. Se perguntarmos a diferentes pessoas se um carro a 70km/h está andando rápido ou devagar, a opinião de cada indivíduo variará. Assim, para uma pessoa essa velocidade é alta e o carro está andando rápido, para outro indivíduo essa velocidade é média, contudo, para um piloto de fórmula 1, essa velocidade seria considerada baixa.

A lógica *fuzzy* busca associar essa visão qualitativa a um conjunto infinito de valores dentro do intervalo  $[0,1]$ . Nesse caso, a função de pertinência de um conjunto  $A$  em um universo  $U$  é dada por:

$$\mu_A(x) = X \rightarrow [0,1] \quad (1.2)$$

A Figura 2 ilustra a função característica da lógica tradicional para a ideia de um carro rápido a 70km/h em comparação com a função de pertinência da lógica em que um carro na mesma velocidade pode ser considerado rápido ou não, dependendo de quem está analisando.



**Figura 2: Comparação entre uma função característica da lógica crisp e uma função de pertinência da lógica *fuzzy*. Fonte: Autora.**

Logo, a teoria *fuzzy* provê um método para manipulação de conjuntos que refletem imprecisões e incertezas a partir de uma medida de possibilidade de pertencimento do elemento ao conjunto (SIMÕES, 2007).

Esses conjuntos carregam propriedades como altura, normalização, domínio, convexidade, conjunto  $\alpha$ -cut e universo de discurso (ZADEH, 1965).

A normalização impõe que a altura máxima da função de pertinência de um conjunto *fuzzy* seja 1, como mostrado na equação abaixo:

$$X \text{ deve ser } normal: \max \mu_X(u) = 1, \quad u \in U \quad (1.3)$$

O domínio do conjunto é o universo total de valores possíveis para os elementos pertencentes a ele. O domínio varia de acordo com o contexto em que a lógica está inserida.

A convexidade, por sua vez, é uma característica importante por carregar a informação sobre a conectividade interior e o formato do número *fuzzy*. É interessante ressaltar que funções não-convexas não possibilitam unicidade na avaliação numérica do valor no eixo horizontal.

$$X \text{ deve ser } convexo: \mu_X[\lambda u_1 + (1 - \lambda)u_2] \geq \min [\mu_X(u_1), \mu_X(u_2)] \quad (1.4)$$

para  $u_1, u_2 \in U, \lambda \in [0,1]$

Caso o conjunto apresente apenas um único ponto em  $u$  em que  $\mu_X(u) = 1$ , o mesmo é chamado de Conjunto *Singleton*. Por sua vez, o conjunto  $\alpha$ -cut representa uma restrição imposta a um domínio para otimizar funções com longos *tails*, ou seja, que possuem valores muito baixos de  $\mu_X(u)$  por um domínio extenso. Nesse caso, esses valores serão eliminados baseados no valor de  $\alpha$ . Por fim, o universo de discurso é o espaço completo de variação de uma variável do modelo (ZADEH, 1965).

Para caracterizar os fenômenos complexos, incertos e qualitativos que envolvem o universo de discurso e os conjuntos *fuzzy* são utilizadas as variáveis linguísticas. Essas variáveis nomeiam os valores atribuídos aos conjuntos e utilizam palavras que reflitam a avaliação do especialista, de modo a facilitam o entendimento do modelo que está sendo criado. Palavras como “alto”, “médio”, “rápido”, “devagar”, “bonito”, “feio” são variáveis linguísticas aplicadas a conjuntos *fuzzy* e relevantes para uma lógica de controle contextualizada.

### 3.2.2 Funções de Pertinência

Existem diversos tipos de funções de pertinência, incluindo aquelas modeladas pelo próprio usuário para uma aplicação específica. Porém, existem as funções mais comumente usadas que definem o formato dos conjuntos *fuzzy*. São elas:

1. **Linear:** é a função de pertinência mais simples possível e, portanto, uma boa escolha para aproximação de conceitos que ainda não são bem compreendidos. A Figura 3 ilustra um conjunto linear.

$$X\mu_A(x) = x, \quad x \in [0,1] \quad (1.5)$$

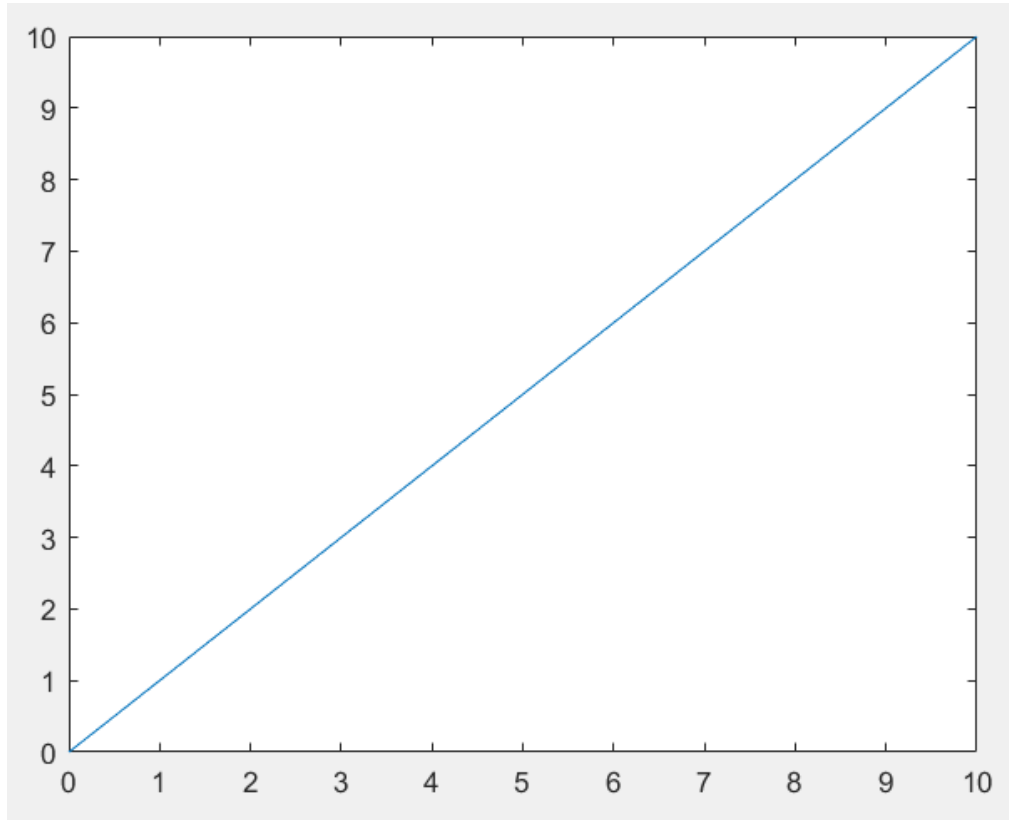
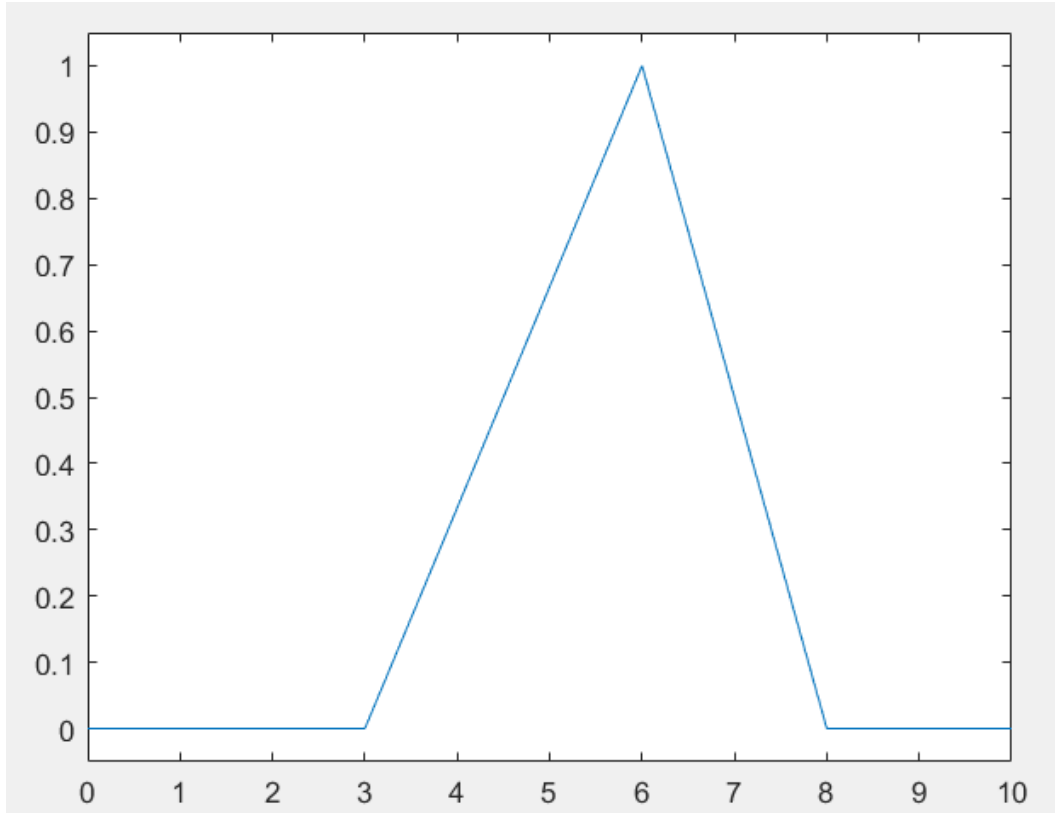


Figura 3: Formato de conjunto linear. Fonte: Autora.

2. **Triangular:** A função de pertinência triangular não apresenta descontinuidade entre as mudanças de crescimento da função. É comumente utilizada para modelar matematicamente o conceito de “em torno de”. A Figura 4 ilustra uma função triangular de um conjunto *fuzzy*.

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \geq a \\ \frac{x-a}{u-a}, & \text{se } a < x \leq u \\ \frac{x-b}{u-b}, & \text{se } u \leq x < b \\ 0, & \text{se } x \geq b \end{cases} \quad (1.6)$$

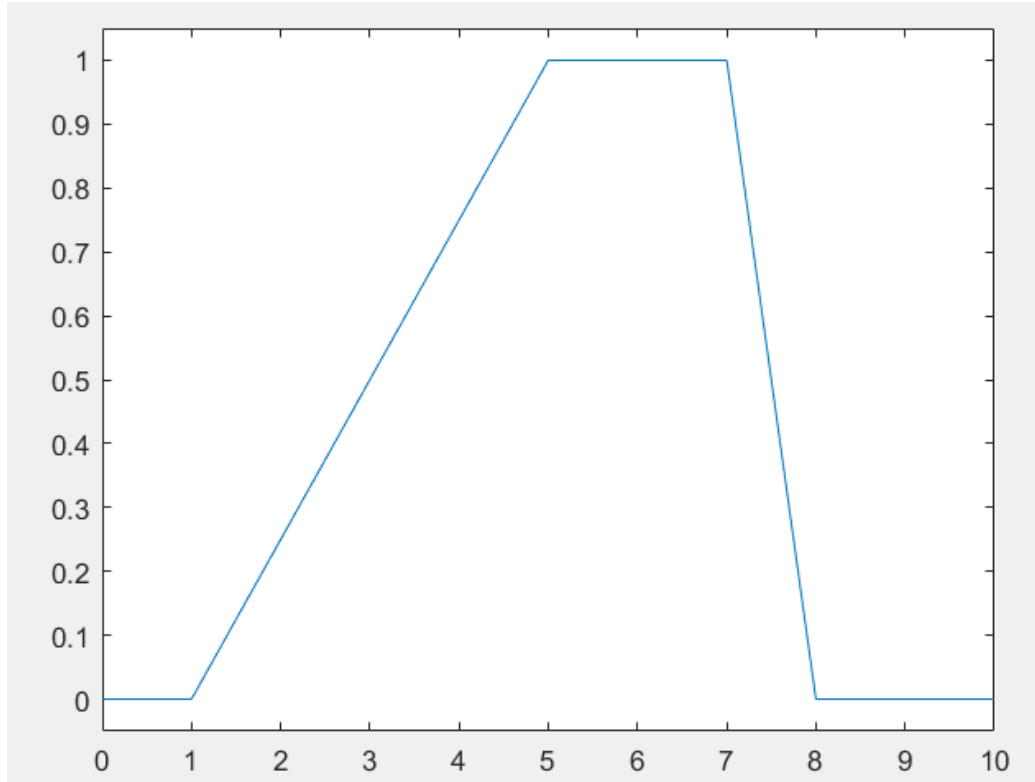


**Figura 4: Formato de conjunto triangular.**  
**Fonte: Autora.**

3. **Trapezoidal:** Com uma fórmula simples, a função trapezoidal é comumente utilizada por facilmente representar conjuntos incertos e demandar pouco esforço computacional. Essa função também pode ser transformada em triângulo, contanto que sua altura máxima seja 1. Sua definição é:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & \text{se } a \leq x < b \\ 1, & \text{se } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{se } c < x \leq d \\ 0, & \text{caso contrário} \end{cases} \quad (1.7)$$

A Figura 5 ilustra uma função trapezoidal.



**Figura 5: Formato de conjunto trapezoidal. Fonte: Autora.**

4. **Gaussiana:** A função gaussiana possui uma distribuição normal tendendo a zero para valores muito maiores ou muito menores do que a média. Assim como as outras funções, sua altura máxima é 1.

A definição desta função é:

$$\mu_A(x) = e^{-\frac{(x-\mu)^2}{\sigma^2}} \quad (1.8)$$

A figura 6 representa uma função de pertinência gaussiana. Como não há vértices, as respostas obtidas por essa função são mais suaves quando comparadas as respostas das funções triangulares e trapezoidais.

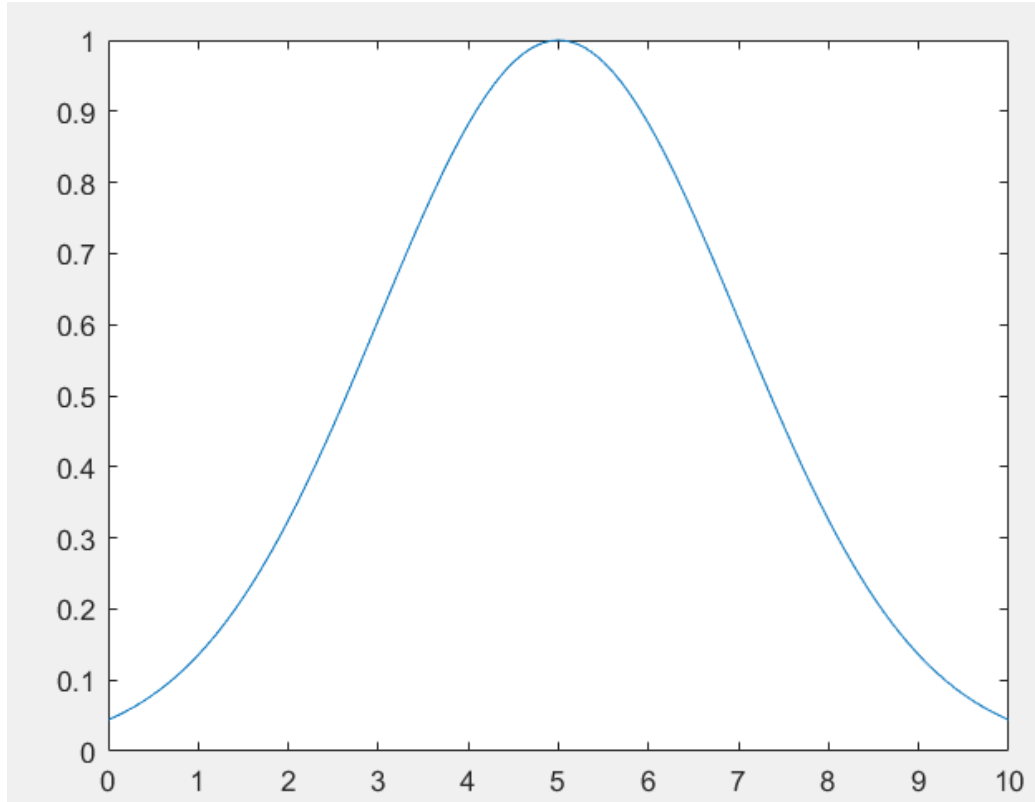


Figura 6: Formato de conjunto Gaussiano. Fonte: Autora.

### 3.2.3 Propriedades

Assim como a lógica clássica apresenta propriedades para conjuntos ordinários o mesmo ocorre para os conjuntos *fuzzy*. Dentre essas, tem-se conforme (Tanscheit ou o livro em inglês) para conjuntos A, B e C:

- Involução:  $(A')' = A$  (1.9)

- Idempotência:  $\begin{cases} A \cap A = A \\ A \cup A = A \end{cases}$  (1.10)

- Comutatividade:  $\begin{cases} A \cap B = B \cap A \\ A \cup B = B \cup A \end{cases}$  (1.11)

- Associatividade:  $\begin{cases} (A \cap B) \cap C = A \cap (B \cap C) \\ (A \cup B) \cup C = A \cup (B \cup C) \end{cases}$  (1.12)

- Distributividade:  $\begin{cases} A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \end{cases}$  (1.13)

- Absorção:  $\begin{cases} A \cap (A \cup B) = A \\ A \cup (A \cap B) = A \end{cases}$  (1.14)

- Lei Transitiva: se  $A \subset B$  e  $B \subset C \Rightarrow A \subset C$  (1.15)

- Lei de De Morgan:  $\begin{cases} (A \cap B)' = A' \cup B' \\ (A \cup B)' = A' \cap B' \end{cases}$  (1.16)



### 3.2.4 Operações

As funções de pertinência são utilizadas para realizar as operações em *fuzzy*. O conjunto de operações básicas foi apresentado por Zadeh em 1965 e refere-se as operações de interseção, união e complemento entre funções com grau de pertinência maior que 0 (zero) e, no caso de  $\alpha$ -cuts, acima do valor de  $\alpha$ . Matematicamente, essas operações são representadas, respectivamente, por:

$$(A \cap B)(x) = \min(A(x), B(x)) \quad (1.17)$$

$$(A \cup B)(x) = \max(A(x), B(x)) \quad (1.18)$$

$$\bar{A} = 1 - A(x) \quad (1.19)$$

A Figura 7 é uma representação gráfica dessas operações sendo realizadas entre um conjunto A com função de pertinência trapezoidal e um conjunto B cuja função de pertinência é triangular. Nesse contexto, entende-se que A e B representados na figura 7a são os *inputs* da operação e os resultados mostrados nas figuras 7b, 7c e 7d são os *outputs* obtidos.

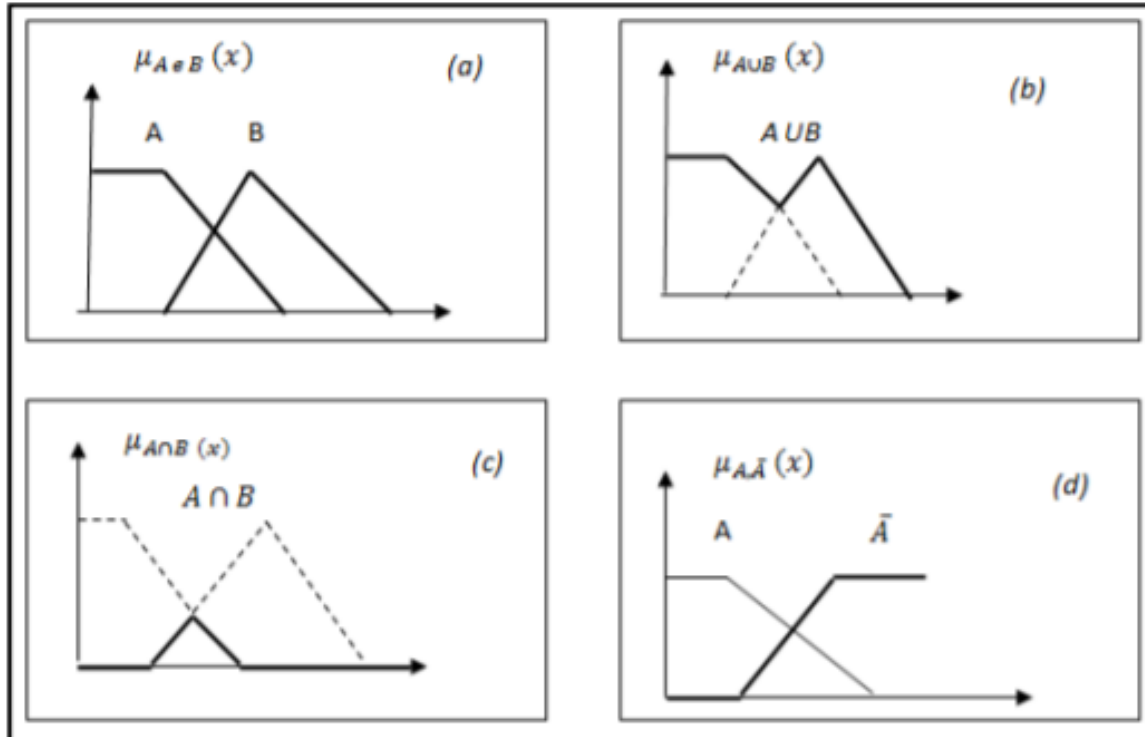


Figura 7: Representação gráfica das operações em *fuzzy* dos conjuntos A e B. (a) conjuntos A e B; (b) união; (c) interseção; (d) complemento. Fonte: SILVA, 2013 adaptado de FERNANDES, 2005.

Comparando as operações entre conjuntos *fuzzy* e conectores lógicos, tem-se que a interseção é equivalente ao operador “AND” ou “E”, que a união equivale ao operador “OR” ou “OU” e, por fim, o complementar assemelha-se ao “NOT” ou “NÃO”.

Essa noção auxilia na escolha da melhor operação de acordo com o objetivo a ser alcançado. Existem diversas outras operações como dilatação, concentração, soma limitada, diferença limitada, entre outras, que não serão referenciadas nesse estudo.

### 3.2.5 Regras e Métodos de Inferência

Após a modelagem do problema e definição das variáveis de entrada e de saída, o passo seguinte é determinar como os *inputs* se relacionam para gerar um *output* determinado. Para isso, cria-se um conjunto de regras baseadas no conhecimento do especialista acerca do tema estudado, as quais as operações disponibilizadas para sistemas *fuzzy* para alcançar o resultado desejado (LEE, 1990).

Uma possível classificação dos sistemas de inferência *fuzzy* é (LEE, 1990):

- Mamdani ou raciocínio *fuzzy* tipo 1; este é um dos sistemas de inferência mais conhecidos e lembrados quando se discute sistemas *fuzzy*;
- Larsen ou raciocínio *fuzzy* tipo 2;
- Takagi e Sugeno ou raciocínio *fuzzy* tipo 3
- Tsukamoto ou raciocínio *fuzzy* tipo 4;

O modelo de Mamdani utiliza regras linguísticas do tipo “SE”/ “ENTÃO”. As entradas são computadas utilizando os valores mínimos, enquanto as saídas são computadas e combinadas utilizando os valores máximo, implicando em uma inferência do tipo max-min. Um exemplo dado em SIMÕES, 2007, que ilustra esse raciocínio é: “SE o trânsito está PESADO na Rua #2 ENTÃO mantenha o semáforo verde MAIS TEMPO ACESSO”.

Neste caso, PESADO e MAIS TEMPO ACESO são conjuntos *fuzzy*, para os quais PESADO é uma função que define o grau de densidade do trânsito e MAIS TEMPO ACESO é uma outra função que define o grau de duração do tempo de operação do semáforo

Se considerarmos  $A_i$  e  $B_i$  como grupos *fuzzy* de entrada e  $C_k$  como o grupo da variável de saída, o processo pode ser explicado matematicamente da seguinte forma:

$$\mu_A: U \rightarrow [0,1] \text{ e } \mu_B: V \rightarrow [0,1] \quad (1.20)$$

$$\mu_{C_1} = \mu_{A_1} \cap \mu_{B_1} = \min(\mu_{A_1}, \mu_{B_1}) \quad (1.21)$$

$$\mu_{C_2} = \mu_{A_2} \cap \mu_{B_2} = \min(\mu_{A_2}, \mu_{B_2}) \quad (1.22)$$

$$\mu_C = \mu_{C_1} \cup \mu_{C_2} = \max(A, B) \quad (1.23)$$

A Figura 8 exemplifica graficamente o modelo matemático acima representado para o caso em que o conector lógico utilizado para aplicar as regras seja do tipo “AND”.

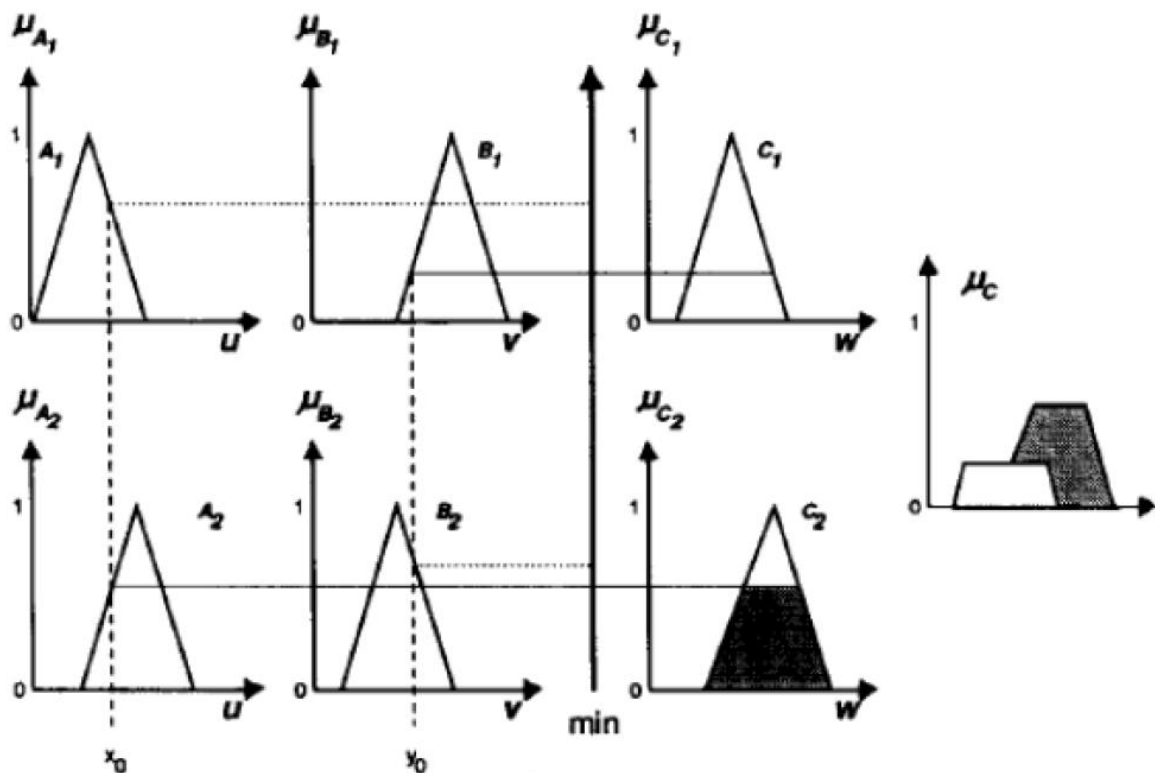


Figura 8: Exemplo de aplicação do método Mamdani. Fonte: (Mamdani, 1975).

No método de inferência de Larsen, as implicações de entrada são calculadas pela norma do produto e as saídas são calculadas pelo máximo do operador de agregação.

O método de Takagi-Segeno é interessante por utilizar funções *crisp* e, por consequência, não necessitar de um modelo de defuzificação para extração do resultado obtido. A operação de agregação consiste na soma das contribuições das funções representativas de cada regra e o resultado direto é a própria saída.

O raciocínio *fuzzy* e, ou método de Tsukamoto, utiliza interpolação como uma simplificação do método de Mamdani, porém a saída é calculada como no método de Takagi-Segeno.

### 3.2.6 Defuzzificação

Após a aplicação dos métodos de inferência, o último passo da aplicação da lógica *fuzzy* é a interpretação da saída obtida. Isso porque, tecnicamente falando, os atuadores operam com saídas precisas, *crisp*, e não atuam diante de uma saída com variável linguística “MAIS RÁPIDO” sem uma atribuição quantitativa.

Para esta etapa, pode-se empregar diferentes métodos. Contudo, os mais comuns são o máximo, o centróide ou centro de gravidade e a média dos máximos.

O primeiro, como o nome sugere, avalia o valor no universo de discurso da variável de saída para o qual o grau de pertinência é máximo.

Para o centróide, a saída precisa é o valor no universo de discurso que corresponde ao centro de gravidade do conjunto *fuzzy* de saída, uma vez que este é o ponto que divide a área de  $\mu_{SAÍDA}$  em duas partes iguais. O valor real da saída ( $u$ ) pode ser calculado como:

$$u^* = \frac{\sum_{j=1}^N u_j \mu_{SAÍDA}(u_j)}{\sum_{j=1}^N \mu_{SAÍDA}(u_j)} \quad (1.24)$$

Este método pode apresentar problemas caso as funções de pertinência não se sobreponham ou caso mais de uma regra apresente a mesma saída *fuzzy* pois, neste caso, a sobreposição de áreas não é devidamente contabilizada.

Por fim, a Média dos Máximos aborda casos em que a saída possui mais de um máximo no universo de discurso da saída. Neste caso, utiliza-se a solução mais plausível calculada pela média entre os dois elementos extremos com os maiores valores da função de pertinência de saída  $\mu_{SAÍDA}$ . Matematicamente, isto pode ser representado por:

$$u^* = \sum_{m=1}^M \frac{u_m}{M} \quad (1.25)$$

Após a defuzzificação, a saída *fuzzy* representa um valor real, o qual pode ser aplicada a um atuador ou a alguma etapa de controle de um controlador.

### 3.3 DIAGRAMA DE VORONOI

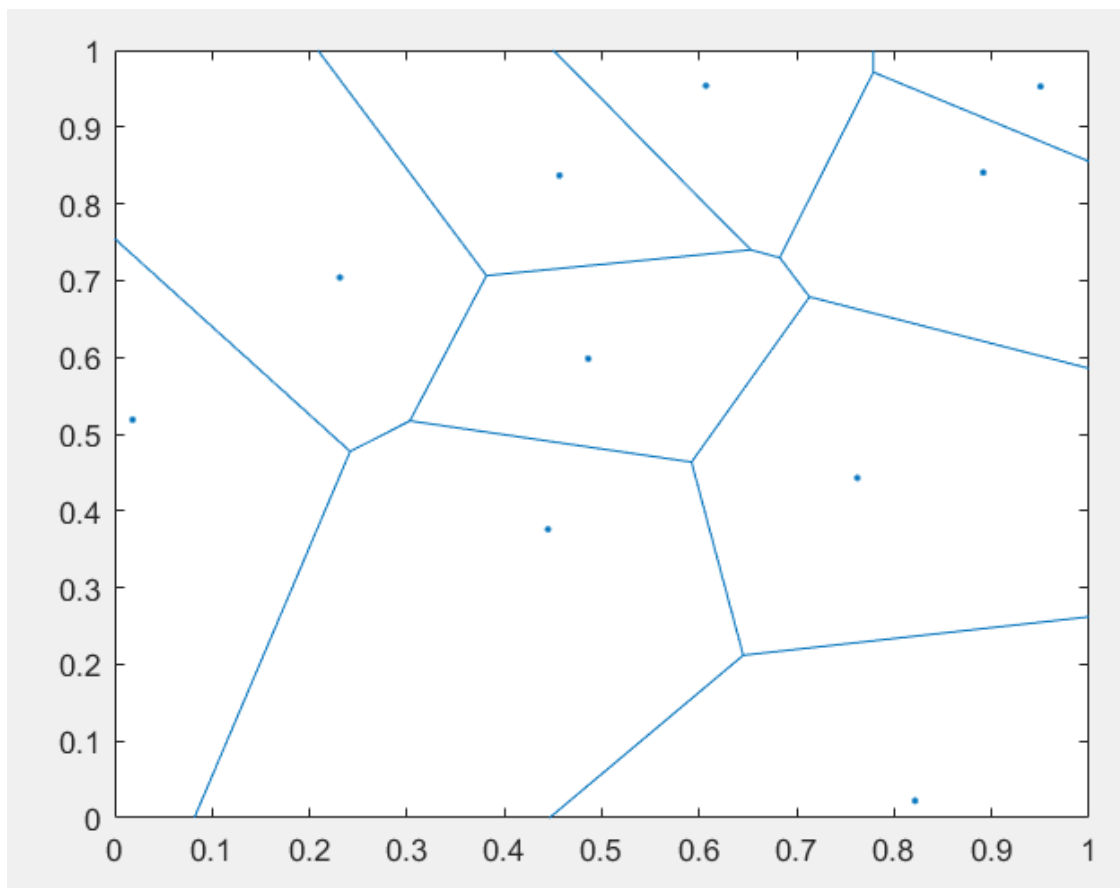
O Diagrama de Voronoi, também conhecido como Polígono de Thiessen ou Tesselação de Delaunay, é empregado nas mais diversas áreas, tais como arqueologia, biologia, cartografia, mineralogia, meteorologia, fisiologia, estatística e ciência da computação (BOOTS *et al.*, 1995).

A ideia base do diagrama pode ser encontrada em trabalhos de René Descartes no ano de 1644 chamados “*Le monde de Mr. Descartes*” e “*Le Traité de La Lumière*”. Entretanto, a criação e estruturação Diagrama de Voronoi, como conhecido atualmente, se deu em 1908 através do trabalho de Georgy Voronoi. O matemático russo utilizou um mosaico para representar a decomposição de um plano conforme a detecção de um vértice próximo ao ponto do plano que está sendo analisado (AURENHAMMER, 1991).

Em 1911, Thiessen empregou o Diagrama de Voronoi em seu trabalho para representar espacialmente a relação entre áreas geográficas e estimar com mais acurácia as médias regionais de precipitação de chuvas. Apesar do pesquisador ter denominado o mapeamento como Polígono de Thiessen, o trabalho contribuiu para a expansão do uso do diagrama para diferentes áreas (RHYNSBURGER, 1973).

O teorema que define formalmente o Diagrama de Voronoi dita que: seja  $P$  um conjunto de  $n$  pontos no plano. Se todos os sítios são colineares,  $Vor(P)$  possui  $n-1$  linhas paralelas e  $n$  células. Em caso contrário,  $Vor(P)$  é convexo e suas arestas são segmentos ou semi-linhas que formam as regiões de Voronoi. O objetivo, então, é determinar para cada ponto  $n$  de  $P$  qual a região  $Vor(P)$  dos pontos do plano estão mais próximos de  $n$  do que qualquer outro ponto de  $P$  (CHEN *et al.*, 2013).

A Figura 9 ilustra um Diagrama de Voronoi, seus vértices e células para um conjunto de 10 pontos gerados randomicamente no *software* MATLAB. De forma simplificada, cada aresta visualizada no modelo corresponde ao conjunto de pontos equidistante de seus dois sítios vizinhos e conecta dois vértices. Os vértices, também chamados de nós, são os pontos equidistantes a três ou mais sítios, também chamados de células (AURENHAMMER, 1991).



**Figura 9:** Ilustração do Diagrama de Voronoi para 10 pontos randômicos contidos no intervalo de 0 a 1. Fonte: Autora.

Em robótica, o diagrama é empregado em algoritmos de planejamento de caminhos como uma forma de esqueletização, ou seja, a redução do espaço livre de um robô a uma representação unidimensional. O auxílio provido pela técnica se dá ao mostrar todos os pontos equidistantes de dois ou mais obstáculos, os quais são fornecidos como *inputs* do grafo. Com isso, tem-se o caminho mais seguro, mais distante dos obstáculos, o qual o robô pode seguir até o destino programado (NORVIG & RUSSEL, 2013).

Os autores de HSU *et al.*, 2012, DONG. *et al.*, 2010 e CHEN *et al.*, 2013 utilizaram o diagrama para planejar os caminhos ótimos para robôs móveis, nos dois primeiros trabalhos, e para uma aeronave no último.

Planejar trajetórias pode exigir um grande esforço computacional e a precisão da análise é necessária para reduzir os riscos advindos da presença de obstáculos desconhecidos e dinâmicos. Os três artigos concluem que a solução otimiza o esforço computacional, oferecendo resultados satisfatórios com riscos reduzidos.

CHEN *et al.*, 2013 e DONG. *et al.*, 2010 combinam o Polígono de Thiessen ao algoritmo de Dijkstra para obtenção do caminho global mais curto, uma vez que o algoritmo analisa a ligação entre todos os vértices para determinar o caminho ótimo.

## 4 DESENVOLVIMENTO

### 4.1 OBJETIVO

O objetivo deste trabalho é contribuir para o desenvolvimento de um controlador para os carrinhos adaptados no projeto *Adapt* utilizando, para isso, o sistema de lógica *Fuzzy* combinado ao uso do Diagrama de Voronoi.

Os carros em questão equivalem a robôs diferenciais controlados por uma criança com Paralisia Cerebral (PC) que possui mobilidade reduzida e pode ser acometida por diferentes níveis de comprometimento físicos e neurológicos. Assim, a implementação do sistema de controle compartilhado deve garantir sua segurança, porém sem privá-la de sua liberdade de escolha no âmbito da locomoção, de forma a contribuir para seu desenvolvimento fisiológico e melhorar suas funções sociais ao promover uma melhor interação dentro do ambiente em que ela está inserida.

HUANG *et al.*, 2018, estudou os impactos fisiológicos e sociais da utilização de carrinhos modificados por crianças com mobilidade reduzida. O grupo amostrado consistia de 29 crianças entre 1 e 3 anos de idade com comprometimentos motores que as impediam de rolar, andar e caminhar, mas que fossem capazes de alcançar um objeto utilizando uma ou as duas mãos e suportassem se manter sentadas e apoiadas por até 30 minutos.

Este grupo foi dividido entre um grupo de controle e um grupo de teste, o qual recebeu o carrinho adaptado e um programa de treinamento social para interação com adultos. Os carrinhos de cada indivíduo foram adaptados para atender as suas necessidades particulares e o controle compartilhado se dava através de um *joystick* sem fio que poderia ser controlado pelo terapeuta da criança, enquanto a mesma tinha acesso ao seu próprio *joystick* no carrinho.

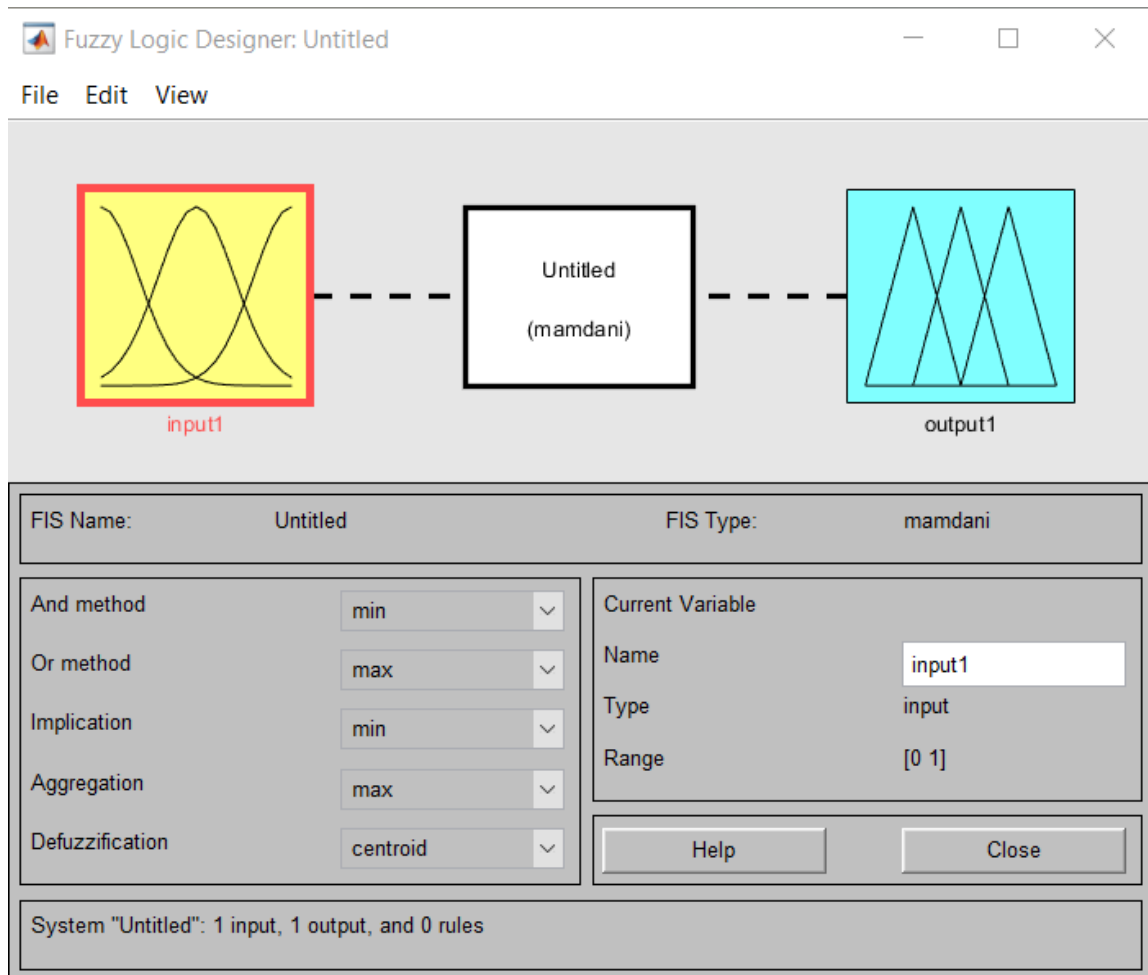
O resultado deste estudo revelou uma melhora significativa nas habilidades motoras das crianças, mas, mais que isso, uma melhora de sua função social pelo uso dos carros adaptados.

Considerando todas as variáveis citadas e, ainda que, durante o período de execução deste trabalho o Brasil enfrentava a pandemia do novo coronavírus (Sars-Cov-1) que impunha restrições de contato social, o controle compartilhado escolhido como objeto de estudo foi desenvolvido com o auxílio de simuladores.



Para tanto, o *software* MATLAB foi escolhido para o desenvolvimento do algoritmo de controle por ofertar a *toolbox* de lógica *fuzzy* denominada *Fuzzy Logic Designer*, encontrada na aba APPS e mostrada na Figura 10, e métodos implementados de utilização do Diagrama de Voronoi chamados *voronoi* e *voronoin*.

Como simulador, o CoppeliaSIM foi eleito por oferecer cenas modeláveis, um modelo do robô *Pioneer p3dx* e uma API de conexão ao MATLAB.



**Figura 10:** Visão da *toolbox* ofertada pelo MATLAB para implementação da lógica *fuzzy*. Fonte: Autora.

#### 4.2 SENSORIAMENTO E DETERMINAÇÃO DAS ZONAS DE SEGURANÇA

O robô escolhido para a simulação foi o *Pioneer 3-DX* modelado e nomeado no CoppeliaSIM como *Pioneer p3dx*, como ilustrado na Figura 11(A). Este robô diferencial possui 16 sensores de medição de distância do tipo sonar distribuídos ao longo da circunferência do robô, como mostrado na Figura 11(B).

A distância de detecção de um objeto pode ser definida manualmente no simulador e, para simular a leitura dos sonares que serão usados no projeto real, estipulou-se que a distância máxima de detecção é de 2m para esses 16 sonares pré-instalados. Este grupo de sonares será denominado, ao longo deste trabalho, de “sonares horizontais”.

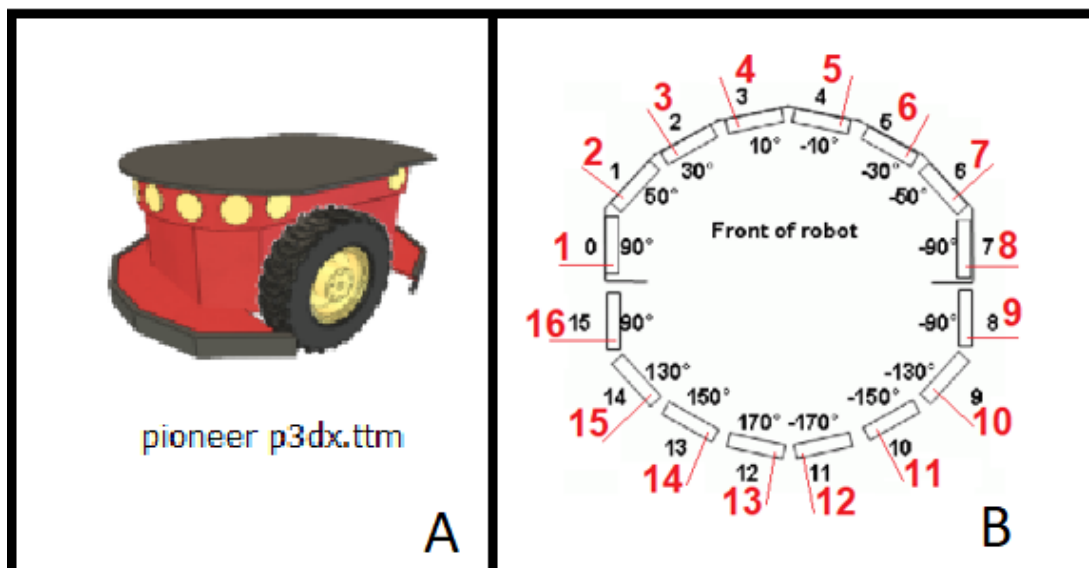
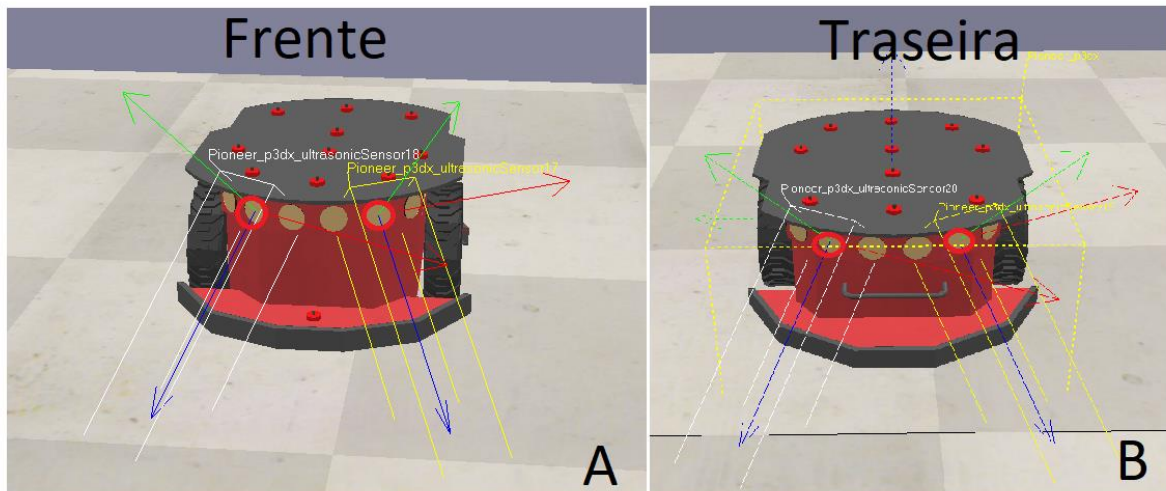


Figura 11: A figura (a) ilustra a imagem do robô Pioneer p3dx no menu de seleção de robôs móveis do CoppeliaSIM, enquanto a figura (b) indica as posições de instalação dos sensores horizontais de acordo com o manual do fabricante. Fonte: Autora.

Além destes, foram instalados mais 4 sonares para detecção de obstáculos no solo, sendo dois na frente – Figura 12 (a) representando os sensores 17 e 18 – e dois na traseira, como mostrados na Figura 12 (b), a qual mostra os sensores 19 e 20. A distância máxima de detecção destes últimos sensores foi definida como 50cm ou 0,5m. Isso porque eles devem ser capazes de detectar buracos, declives, piscinas, degraus ou qualquer outra forma de obstáculo em profundidade do solo. A denominação escolhida para este grupo particular foi “sonares de leitura do chão”.



**Figura 12:** A Figura (A) mostra o posicionamento dos sonares frontais que realizam a leitura do chão, enquanto a figura (B) ilustra os sonares traseiros. Fonte: Autora.

É interessante ressaltar que o simulador faz a leitura individual de cada sensor e indica ao MATLAB se um obstáculo foi detectado ou não, para cada um deles, através da lógica 0 para não-deteccção e 1 para detectado. Logo, para o grupo de sonares horizontais, a preocupação existe para os casos em que o obstáculo é detectado enquanto, para o segundo grupo, só há risco em casos de não-deteccção. Isso porque os sensores de leitura do chão estão constantemente detectando o “obstáculo chão” e, ao retornarem 0, estão indicando que há uma alteração importante em profundidade do solo naquele instante e naquele local.

Para todos os sonares foi mantida a configuração padrão de deteccção mínima de 0,05m após a qual o objeto não é mais detectado. Ademais, os ângulos de deteccção de todos os sensores foram mantidos com a configuração inicial de 45°.

Assim, após essa etapa de configuração, determinou-se as zonas do robô de forma que os obstáculos pudessem ser posicionados em relação ao *Pioneer* após a sua deteccção. Foram escolhidas 4 zonas, mostradas na Figura 13, para agrupar os sensores horizontais e 2 zonas para agrupar os sensores de leitura do chão. São elas:

- Zona da frente: sensores 3, 4, 5 e 6;
- Zona traseira: sensores 10, 11, 12, 13, 14 e 15;
- Zona direita: 6, 7, 8, 9, 10 e 11;
- Zona esquerda: 1, 2, 3, 14, 15 e 16.
- Zona frente chão: sensores 17 e 18;
- Zona traseira chão: sensores 19 e 20.

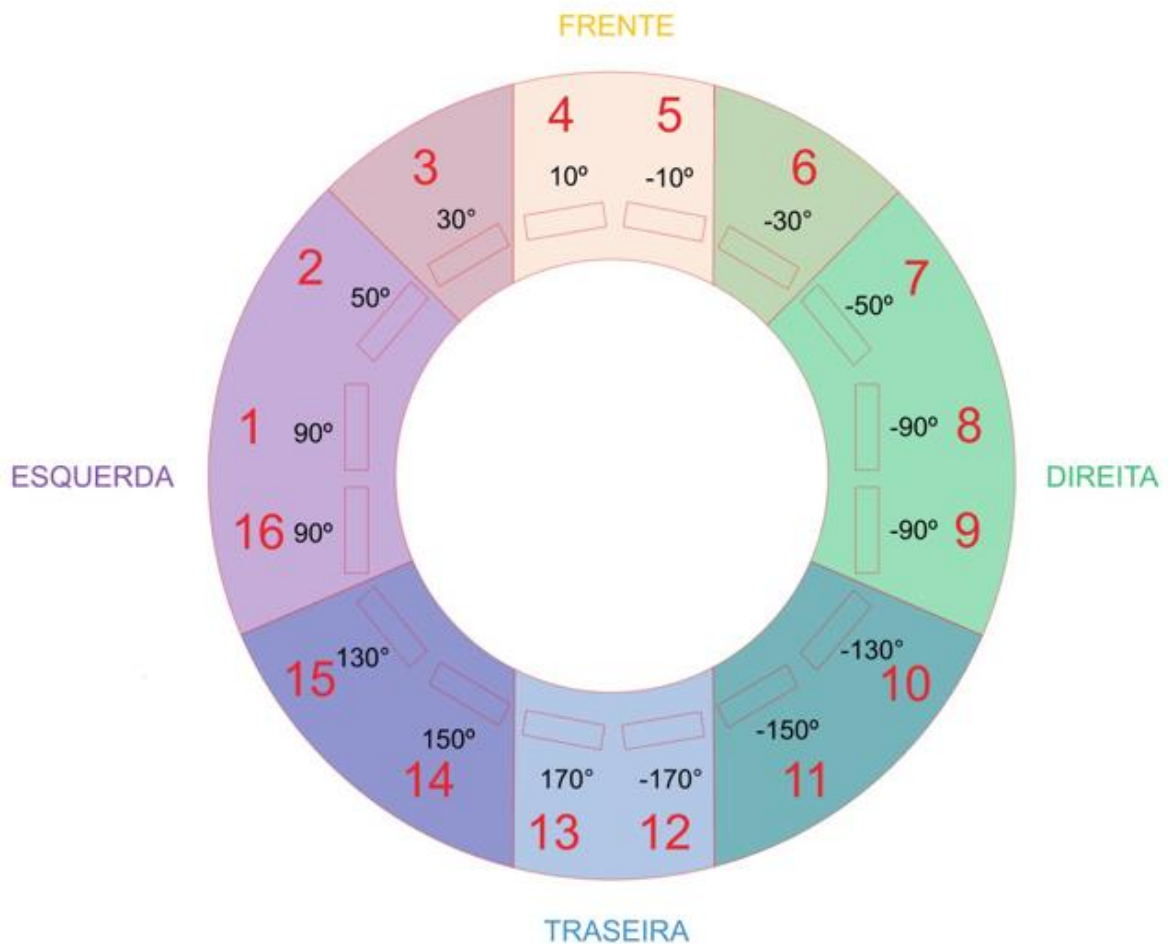


Figura 13: Ilustração da divisão de zonas horizontais do robô. Fonte: Autora

Nota-se que ocorre uma sobreposição proposital entre as zonas da frente e traseira com as zonas direita e esquerda porque, em avaliações qualitativas de posição, atribuímos noções de frente direita ou traseira esquerda para um elemento em relação a outro. Assim, um obstáculo pode pertencer tanto a frente do robô quanto a direita ao mesmo tempo, por exemplo.

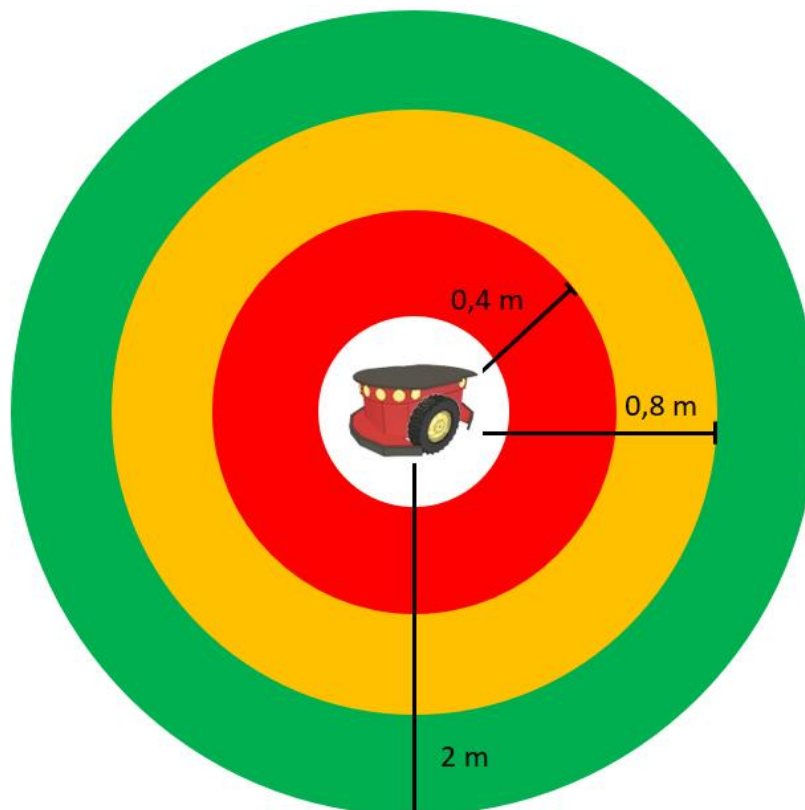
Esta divisão também está alinhada as opções de comando do usuário, as quais consistem em “ir para frente”, “ré”, “virar à direita” ou “virar à esquerda”. A cada um dos comandos são atribuídas as seguintes velocidades:

- Ir para frente: velocidade linear máxima positiva e velocidade angular 0;
- Ré: velocidade linear máxima negativa e velocidade angular 0;
- Virar à direita: velocidade linear zero e velocidade angular máxima negativa;
- Virar à esquerda: velocidade linear zero e velocidade angular máxima positiva.

A atribuição positiva ou negativa à velocidade angular se deu pela aplicação da regra da mão direita, considerando que o robô se movimenta pelo plano XY e o eixo de rotação das rodas é o X.

Além das zonas referentes a posição dos obstáculos em relação ao robô, também foram criadas as “zonas de segurança”. Os limites dessas zonas são um dos critérios que definem o funcionamento do controle compartilhado em termos de a quem o controle é atribuído: se ao usuário, ao controlador Fuzzy ou ao controlador baseado no diagrama de Voronoi.

A Figura 14 exhibe as áreas correspondentes às zonas de segurança para sensores horizontais e seus raios. A zona denominada segura corresponde a área verde, na qual o controle seguido é fornecido pela criança que pilota o carro adaptado. A área amarela-escuro, por sua vez, corresponde à zona de atenção, na qual atua o controlador *fuzzy*. Por fim, existe a área vermelha considerada uma zona de risco, onde quem atua é o controlador baseado no Diagrama de Voronoi.



**Figura 14: Zonas de Segurança definidas para a detecção de obstáculos. A zona de segurança corresponde a cor verde, a zona de atenção a cor amarelo escuro e a zona de risco a cor vermelha.**

**Fonte: Autora.**

Desta forma, se um obstáculo for detectado a uma distância menor ou igual 2m e maior ou igual 0,8m, o mesmo se encontra dentro da zona segura e o comando do usuário será respeitado. Contudo, para distâncias menores que 0,8m e maiores ou iguais a 0,2m o controlador *fuzzy* pode começar a atuar. Caso a aproximação permaneça e o obstáculo se encontre a uma distância menor que 0,2m, ou se o obstáculo detectado se encontrar no solo, o controle será baseado no Diagrama de Voronoi.

O segundo critério para determinação da atuação do controle compartilhado se dá na interseção entre o comando dado pela criança e a posição do obstáculo. Por exemplo, se o comando corresponder a “Ir pra frente” e o obstáculo encontra-se na região “traseira” e na “zona de atenção”, nada será feito. Isso porque esse obstáculo não oferece risco por não pertencer ao caminho que o usuário deseja seguir. Contudo, se o usuário estabelecer um comando de “Virar à direita” que culmine neste obstáculo se posicionando na região “Direita” do robô e dentro da “zona de atenção”, o controlador *fuzzy* será ativado.

Entretanto, se a qualquer momento a criança adentrar a região da zona de risco, o controlador de Voronoi será ativado. Isso porque entende-se que há o risco de colisão iminente devido à proximidade.

É importante lembrar que usuários com paralisia cerebral podem apresentar espasmos musculares que levam a uma rigidez muscular espasmódica. Caso essa rigidez ocorra no braço que atua no *joystick* de controle, a criança não estará enviando um sinal voluntário ao carrinho e poderá se colocar em uma situação de risco. Por esta razão, o controle baseado no Diagrama de Voronoi atua independentemente do que a criança ordenou dentro das situações já mencionadas.

A Tabela 1 ilustra algumas situações passíveis de ocorrerem relacionando o comando dado pelo usuário, as zonas de agrupamento dos sensores, as zonas de segurança e a atuação dos controladores.

**Tabela 1: Exemplos de situações enfrentadas pelo usuário e a resposta dada pelo controlador.**

Comando Criança	Deteção de objeto	Zona de deteção do objeto	Zona de segurança	Controle atuante
Ir pra frente	Não-detectado	-	-	Comando criança: “Ir pra frente”
Ir pra frente	Detectado	Traseira	Zona de Atenção	Comando criança: “Ir pra frente”
Ir pra frente	Detectado	Direita	Zona de Atenção	Comando criança: “Ir pra frente”
Virar à direita	Detectado	Direita	Zona de Atenção	Controlador <i>fuzzy</i>
Virar à direita	Detectado	Esquerda	Zona de Risco	Controlador de Voronoi
Ré	Detectado	Ré	Zona de Atenção	Controlador <i>fuzzy</i>
Ré	Detectado	Ré	Zona de Risco	Controlador de Voronoi
Ré	Detectado	Direita	Zona de Atenção	Comando criança: “Ré”
Virar à esquerda	Não-detectado	-	-	Comando criança: “Virar à esquerda”
Virar à esquerda	Detectado	Esquerda	Zona de Atenção	Controlador <i>fuzzy</i>
Virar à esquerda	Detectado	Zona da frente chão	Zona de Risco	Controlador de Voronoi
Ré	Detectado	Zona da frente chão	Zona de Risco	Controlador de Voronoi

A última situação mostrada na tabela, por exemplo, ocorre devido ao risco de um espasmo muscular da criança. Ainda que ela esteja fugindo do obstáculo do chão, devido à proximidade do obstáculo, menos de 0,2m de distância, se ela tiver um espasmo muscular que enrijeça seu braço e empurre o *joystick* pra frente, ela atingirá o obstáculo. Esse risco não pode ser tolerado e, portanto, o controlador de Voronoi atuará de forma independente.

### 4.3 CONTROLADOR FUZZY

O sistema de controle baseado na lógica *fuzzy* foi criado a partir da *toolbox* disponibilizada pelo MATLAB denominada *Logic Fuzzy Designer*, como mencionado anteriormente.

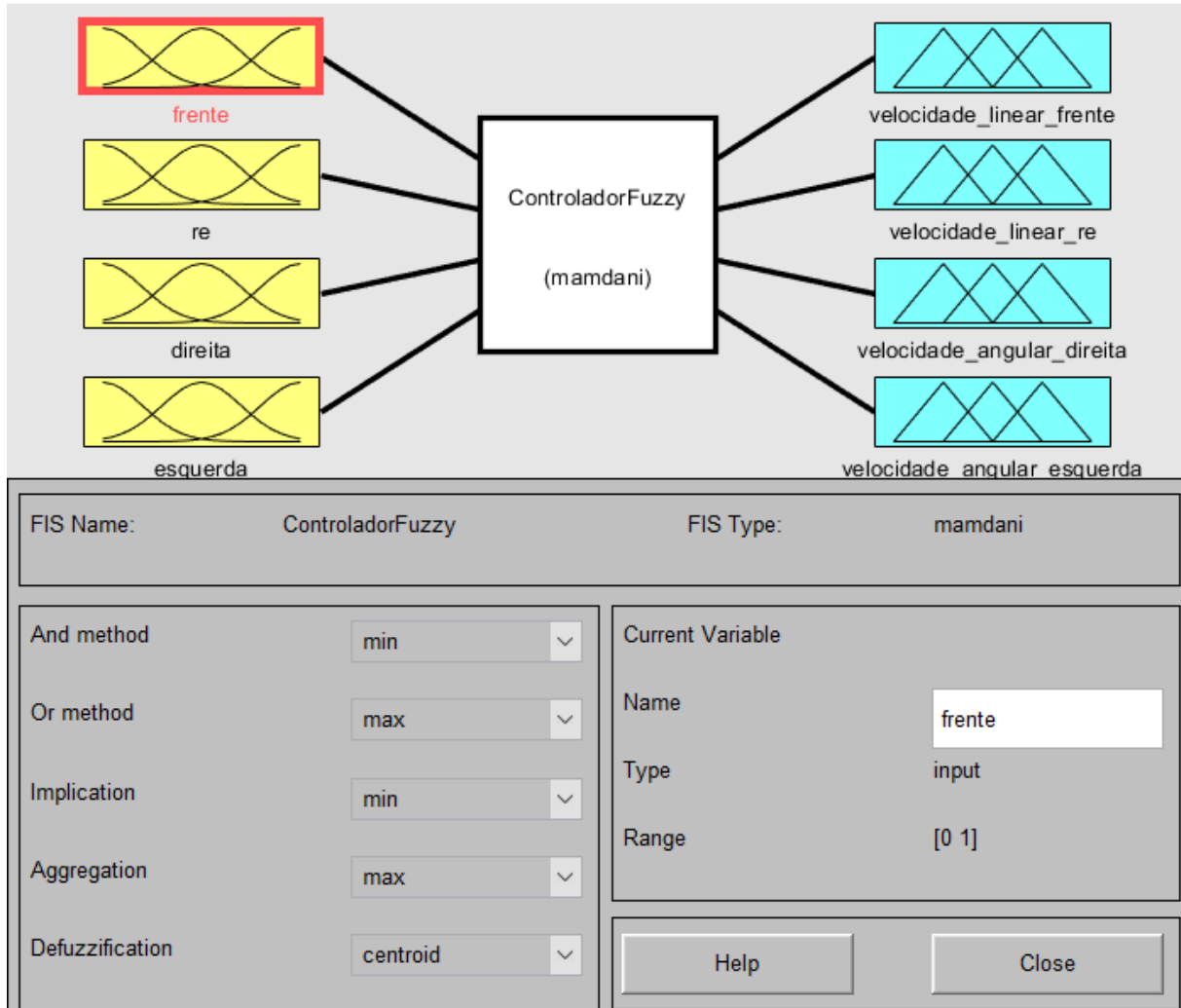
A premissa para determinação das variáveis de entrada e de saída foi a de modular tanto a velocidade linear quanto a velocidade angular a partir da identificação de em qual zona o obstáculo se encontra.

No algoritmo, a condição de avaliação dos parâmetros de entrada para obtenção dos valores de saída é obtida quando um obstáculo é detectado na zona do robô para a qual a criança está ativamente se direcionando.

A detecção de obstáculos, assim como a distância que eles se encontram do robô, são constantemente avaliados e calculados. Logo, se a distância mínima do obstáculo detectado é menor ou igual a 40% da distância máxima de detecção, 0,8m no cenário testado, e maior ou igual a 20% da distância máxima de detecção, equivalente a 0,4m no valor estipulado neste trabalho, como citado anteriormente, o robô entra na “zona de atenção” e o controlador *fuzzy* é ativado.

A Figura 15 mostra o controlador *fuzzy* modelado no formato Mamdani com suas 4 variáveis de entrada, cada uma correspondendo à uma zona de agrupamento de sensores na horizontal, e as 4 variáveis de saída que correspondem à velocidade linear para mover-se para frente (*velocidade\_linear\_frente*), velocidade linear para ré (*velocidade\_linear\_re*), velocidade angular para girar para a direita (*velocidade\_angular\_direita*) e velocidade angular para girar para a esquerda (*velocidade\_angular\_esquerda*). O método de defuzzificação escolhido foi o centróide, pois se mostra o que melhor baliza a concentração de área dos conjuntos *fuzzy* agregados para este trabalho em específico.





**Figura 15: Toolbox modelada para implementação no controlador utilizando 4 variáveis de entrada e 4 variáveis de saída. Fonte: Autora.**

Para cada variável de entrada foram atribuídas duas variáveis linguísticas denominadas “Perto” e “Longe” e, assim como o nome sugere, elas expressam a ideia de o obstáculo estar perto ou longe do robô dentro da zona avaliada. O alcance dos valores, como evidenciado nas Figuras 16, 17, 18 e 19 varia entre 0 e 1. Isso porque foi usada a ideia de porcentagem em que 0 = 0% e 1 = 100%. Dessa forma, o controlador é válido para qualquer sistema, desde que seja determinada uma distância máxima de detecção.

Outro detalhe evidenciado na imagem está relacionado a função de pertinência. Em todos os casos optou-se pela trapezoidal. Logo, todas as variáveis “Perto” possuem o vetor de valores trapezoidais igual a [0 0 0.3 0.6] e as variáveis “Longe” correspondem a [0.3 0.55 1 1].

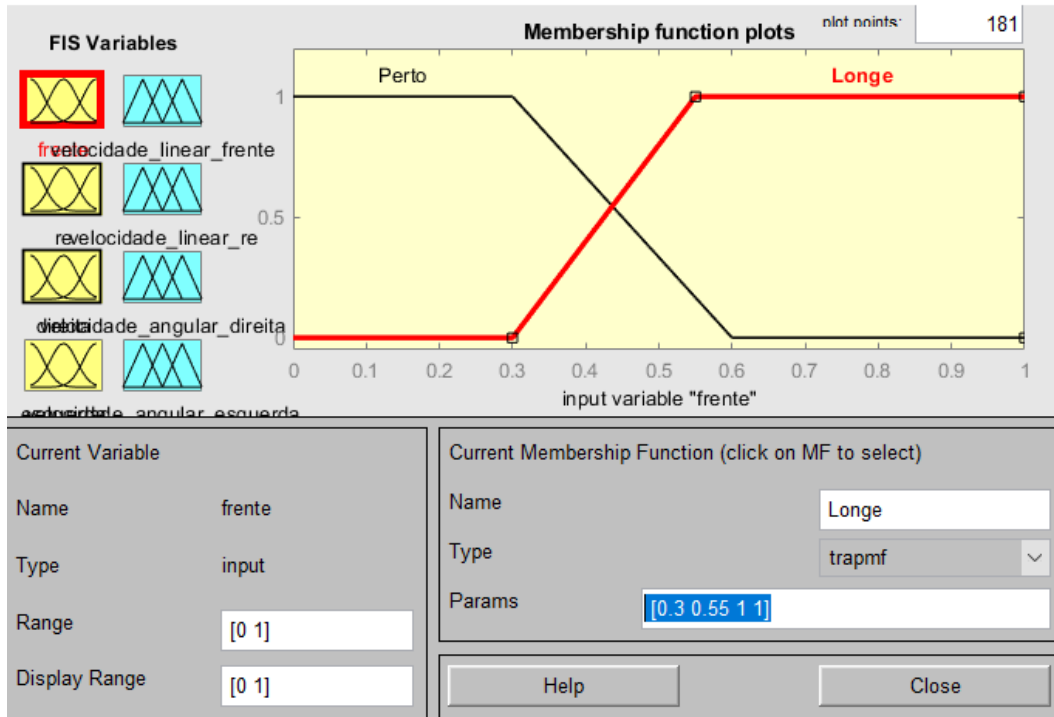


Figura 16: Funções de pertinência da variável de entrada "frente". Fonte: Autora.

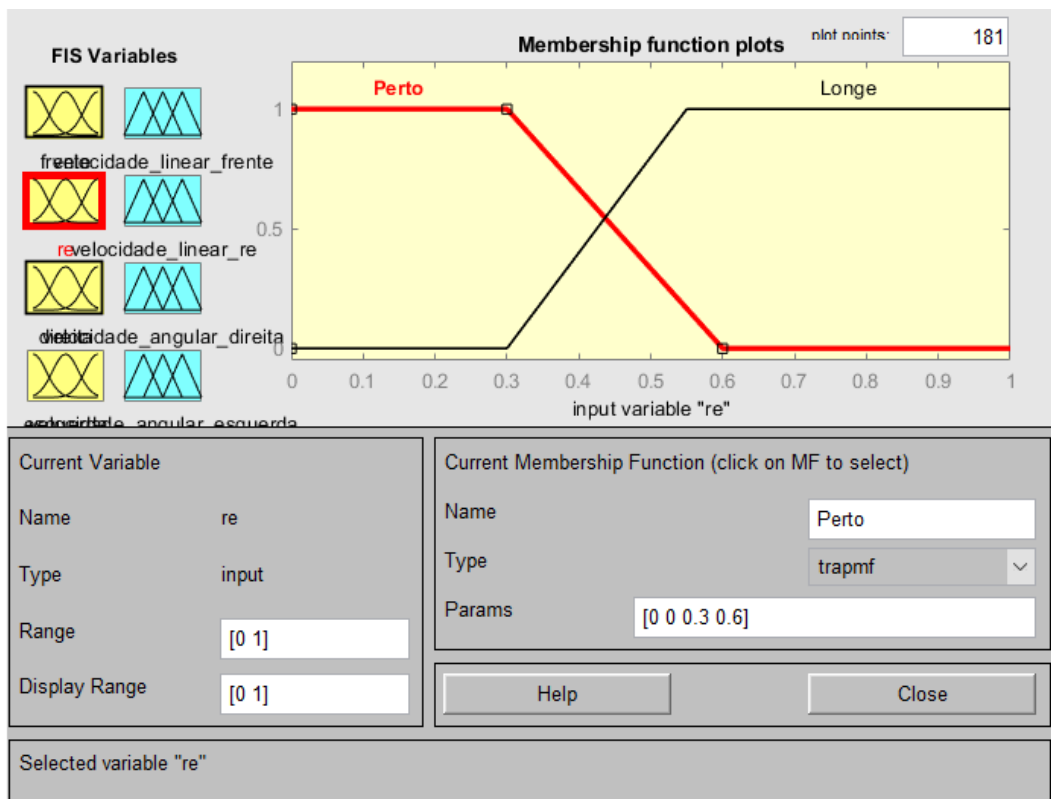


Figura 17: Funções de pertinência da variável de entrada "re". Fonte: Autora.

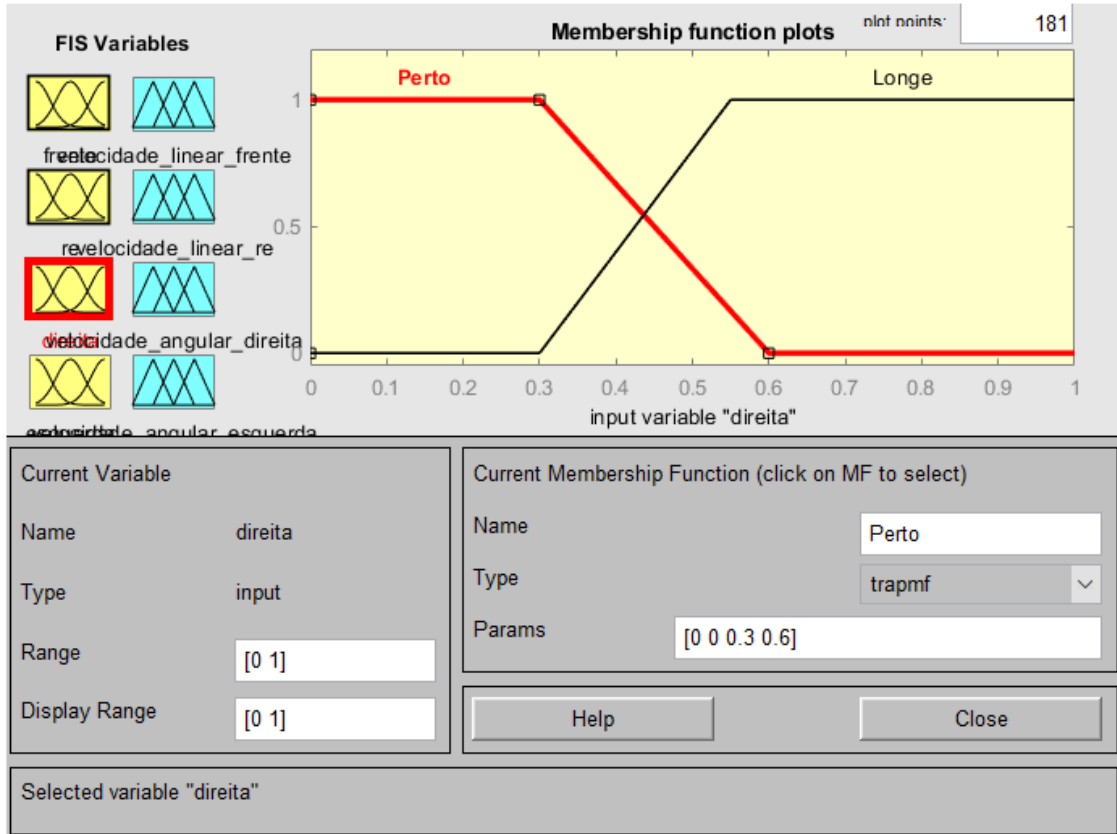


Figura 19: Funções de pertinência da variável de entrada "direita". Fonte: Autora.

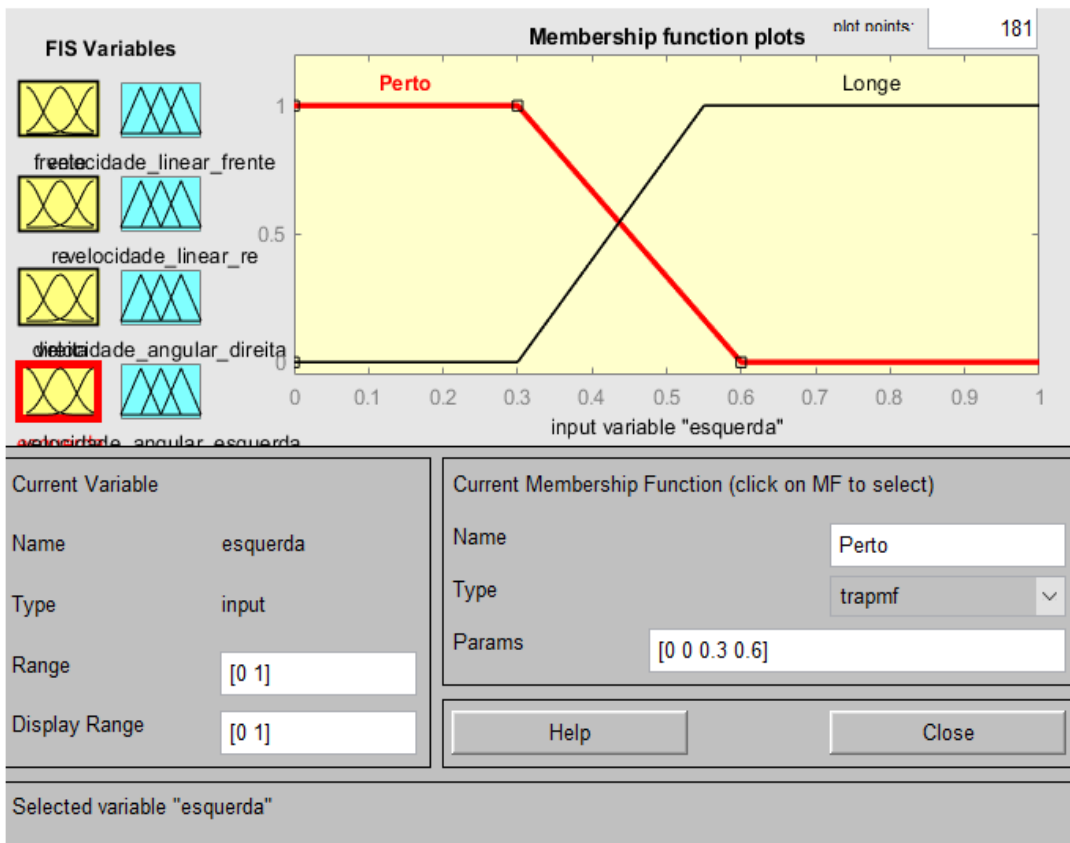


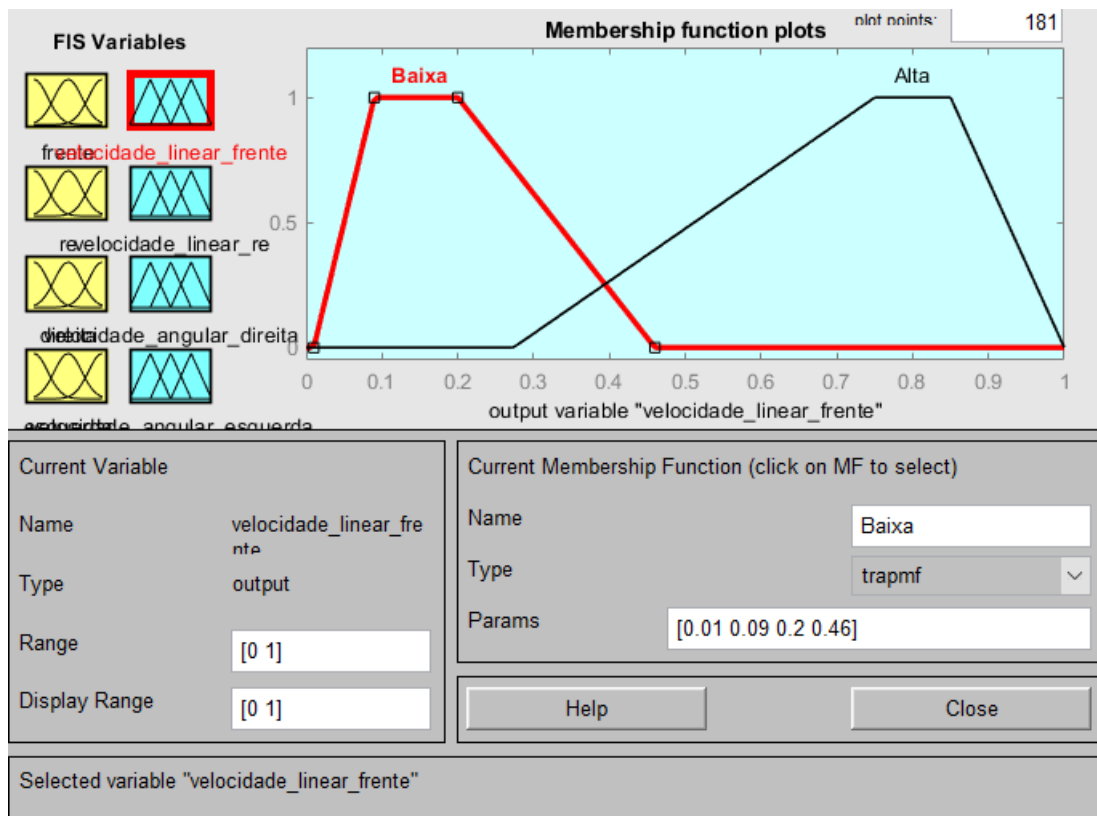
Figura 18: Funções de pertinência da variável de entrada "direita". Fonte: Autora.

A particularização de uma distância “Média” não foi expressada em termos de uma variável linguística, mas sim pela interseção entre “Perto” e “Longe”. Isto foi feito para reduzir a complexidade do conjunto de regras *fuzzy*.

Ademais, todos os valores atribuídos à função trapezoidal foram obtidos de forma empírica.

No caso das variáveis de saída, as variáveis linguísticas atribuídas a elas foram “Baixa” e “Alta”, sendo estas relacionadas a ideia de uma velocidade mais baixa ou mais alta. Assim como no caso das entradas, a ideia “Média” é atribuída como uma interseção entre as funções de pertinência trapezoidais.

Uma diferença evidente entre as funções citadas encontra-se na variável de saída “velocidade\_linear\_frente”. Isto foi feito de forma proposital para privilegiar a mobilidade na região frontal do robô, uma vez que esta zona se encontra no campo direto de visão da criança. Este fato pode ser observado através da comparação entre a Figura 20 e as Figuras 21, 22 e 23. De forma semelhante as variáveis de entrada, os valores das funções de pertinência foram obtidos de forma empírica.



**Figura 20: Funções de pertinência da variável de saída "velocidade\_linear\_frente". Fonte: Autora.**

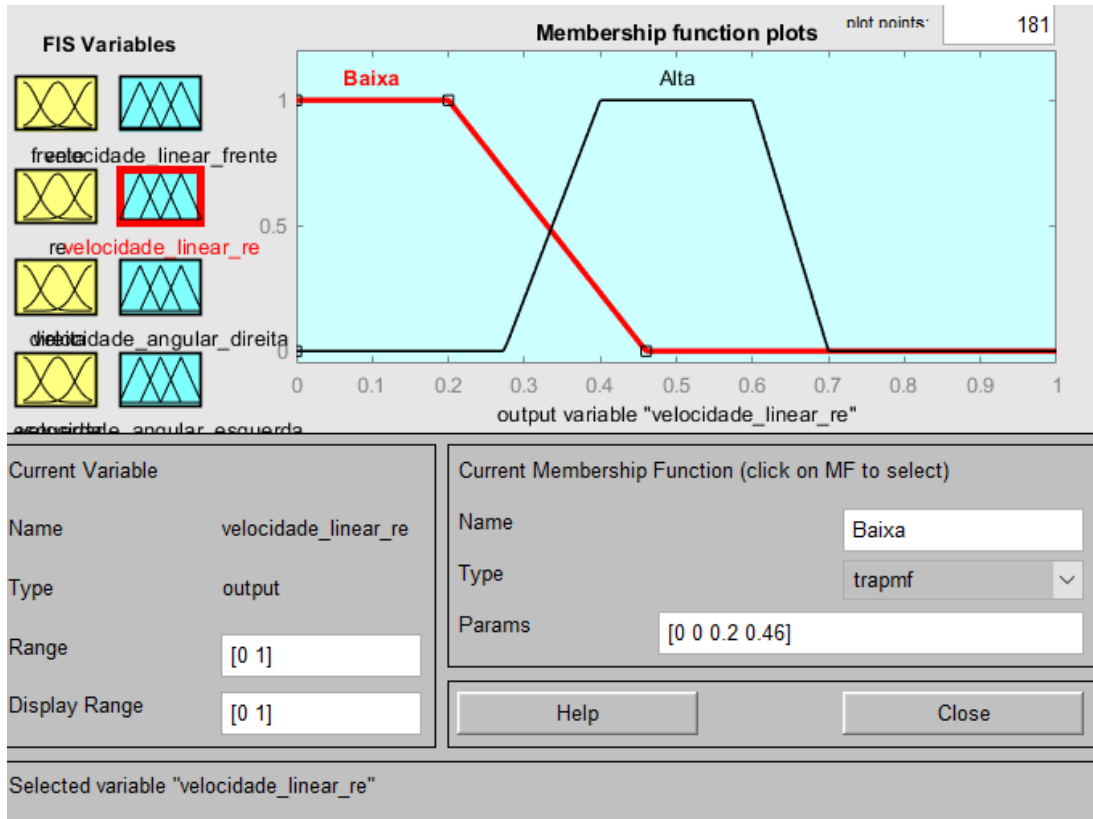


Figura 22: Funções de pertinência da variável de saída "velocidade\_linear\_re". Fonte: Autora.

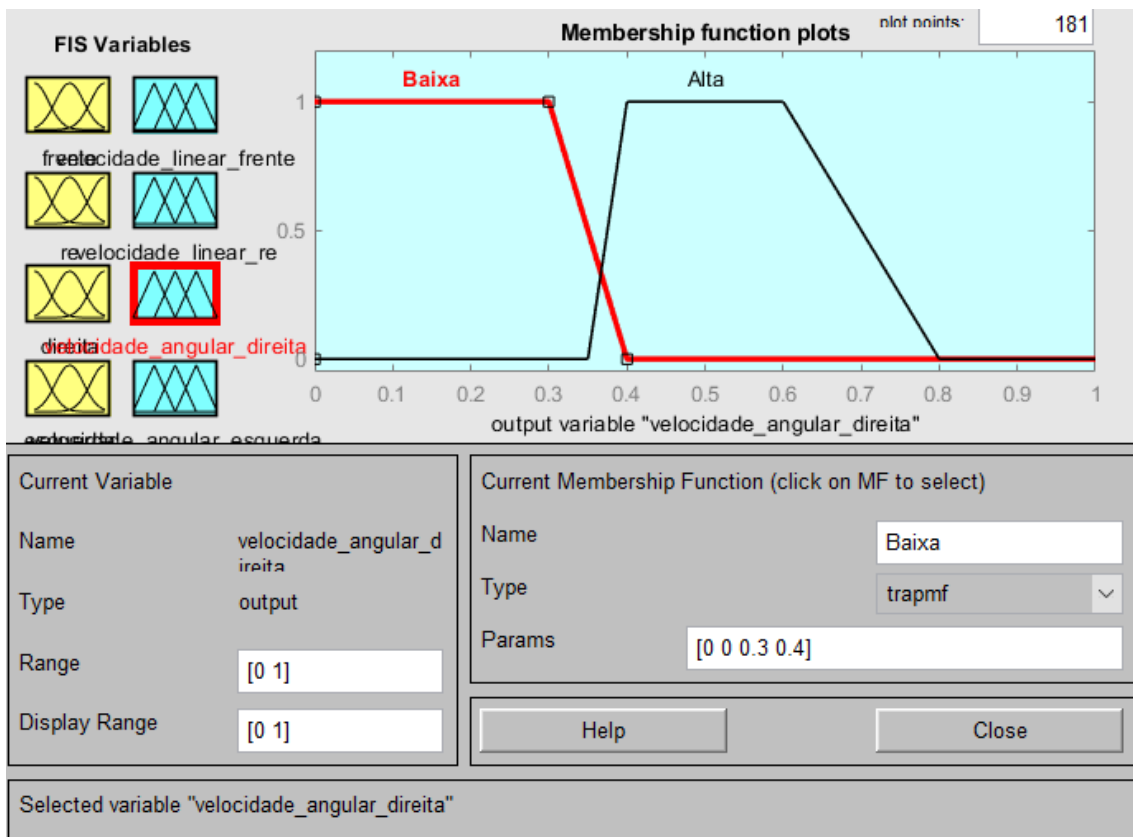


Figura 21: Funções de pertinência da variável de saída "velocidade\_angular\_direita". Fonte: Autora.

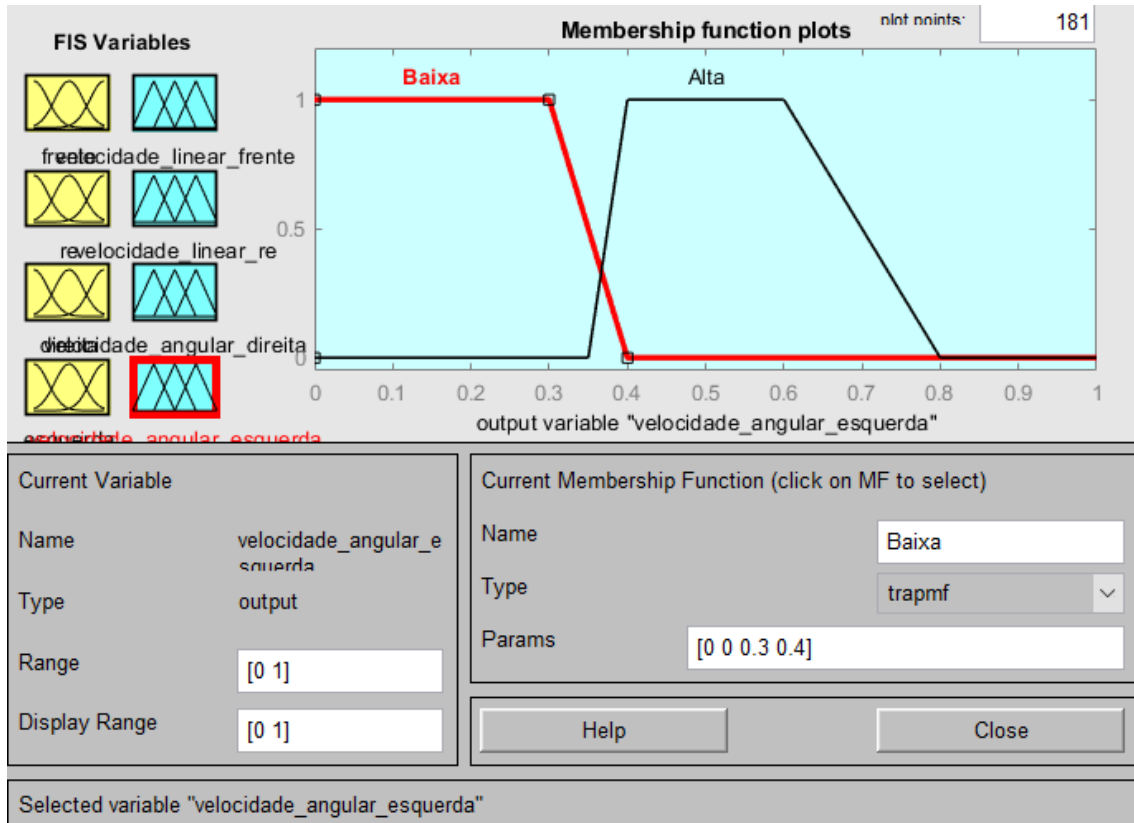


Figura 23: Funções de pertinência da variável de saída "velocidade\_angular\_esquerda". Fonte: Autora.

Após a escolha das variáveis de entrada e saída e a modelagem de suas funções de pertinência, o determinou-se as regras que irão reger as relações entre *inputs* e *outputs*. As mesmas são explicitadas abaixo:

- Se "frente" é "Perto" e "re" é "Perto" e "direita" é "Perto" e "esquerda" é "Perto", então "velocidade\_linear\_frente" é "Baixa" e "velocidade\_linear\_re" é "Baixa" e "velocidade\_angular\_direita" é "Baixa" e "velocidade\_angular\_esquerda" é "Baixa".
- Se "frente" é "Perto" e "re" é "Perto" e "direita" é "Perto" e "esquerda" é "Longe", então "velocidade\_linear\_frente" é "Baixa" e "velocidade\_linear\_re" é "Baixa" e "velocidade\_angular\_direita" é "Baixa" e "velocidade\_angular\_esquerda" é "Alta".
- Se "frente" é "Perto" e "re" é "Perto" e "direita" é "Longe" e "esquerda" é "Perto", então "velocidade\_linear\_frente" é "Baixa" e "velocidade\_linear\_re" é "Baixa" e "velocidade\_angular\_direita" é "Alta" e "velocidade\_angular\_esquerda" é "Baixa".
- Se "frente" é "Perto" e "re" é "Perto" e "direita" é "Longe" e "esquerda" é "Longe", então "velocidade\_angular\_frente" é "Baixa" e



“velocidade\_linear\_re” é “Baixa” e “velocidade\_angular\_direita” é “Alta” e “velocidade\_angular\_esquerda” é “Alta”.

- Se “frente” é “Longe” e “re” é “Longe” e “direita” é “Perto” e “esquerda” é “Perto”, então “velocidade\_linear\_frente” é “Alta” e “velocidade\_linear\_re” é “Alta” e “velocidade\_angular\_direita” é “Baixa” e “velocidade\_angular\_esquerda” é “Baixa”.
- Se “frente” é “Longe” e “re” é “Longe” e “direita” é “Perto” e “esquerda” é “Longe”, então “velocidade\_linear\_frente” é “Alta” e “velocidade\_linear\_re” é “Alta” e “velocidade\_angular\_direita” é “Baixa” e “velocidade\_angular\_esquerda” é “Alta”.
- Se “frente” é “Longe” e “re” é “Longe” e “direita” é “Longe” e “esquerda” é “Perto”, então “velocidade\_linear\_frente” é “Alta” e “velocidade\_linear\_re” é “Alta” e “velocidade\_angular\_direita” é “Alta” e “velocidade\_angular\_esquerda” é “Baixa”.
- Se “frente” é “Longe” e “re” é “Longe” e “direita” é “Longe” e “esquerda” é “Longe”, então “velocidade\_linear\_frente” é “Alta” e “velocidade\_linear\_re” é “Alta” e “velocidade\_angular\_direita” é “Alta” e “velocidade\_angular\_esquerda” é “Alta”.

#### 4.4 CONTROLADOR BASEADO NO DIAGRAMA DE VORONOI

O segundo sistema de controle utiliza as coordenadas cartesianas dos obstáculos detectados, tanto pelos sonares horizontais quanto pelos sonares de leitura do chão para gerar um Diagrama de Voronoi.

É importante observar que esse diagrama, no MATLAB, é traçado utilizando a triangulação de Delaunay, o que implica na existência de no mínimo três obstáculos detectados para que o grafo possa ser gerado. Caso esse mínimo não seja respeitado, o algoritmo apresentará erro e faz-se necessário trabalhar com uma contingência.

Caso exista o mínimo de três pontos ou mais e aplicando apenas as suas coordenadas de X e Y na função *voronoi*, obtém-se os vértices e células. Contudo, apenas as coordenadas dos vértices são aproveitadas neste trabalho.

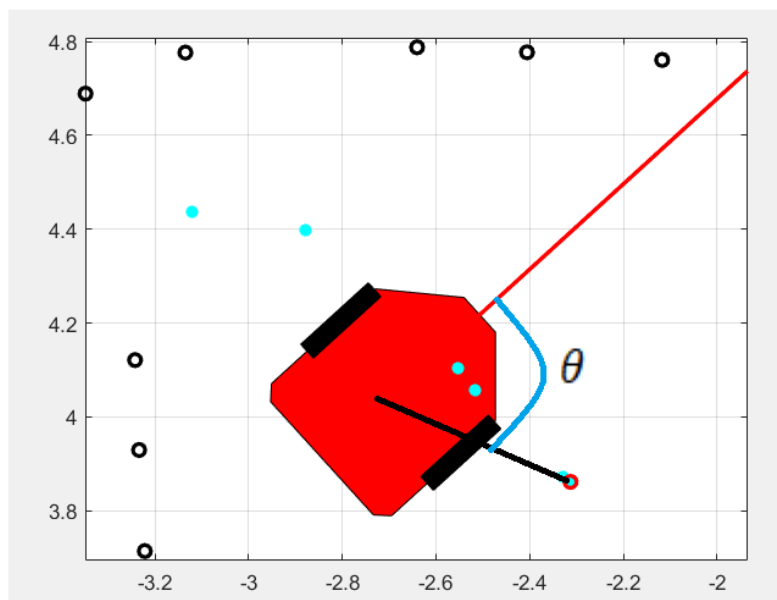
Primeiramente, faz-se necessário filtrar os valores obtidos para remover vértices que estejam no infinito e os *outliers*. Os últimos são retirados através de sucessivas iterações que comparam as coordenadas dos vértices com as



coordenadas mínimas e máximas (em X e Y) dos obstáculos. Assim, é possível gerar um mapa local.

O passo seguinte é identificar qual vértice válido encontra-se mais distante de todos os obstáculos e utilizar suas coordenadas para determinar o ângulo  $\varphi$  entre a posição do robô e a posição do ponto, agora denominado como objetivo. Por último, a velocidade linear é zerada e, de forma iterativa, o robô realiza uma rotação em direção ao ponto objetivo escolhido respeitando, sempre, a velocidade angular máxima permitida.

A Figura 24 ilustra como essa situação é plotada no MATLAB. Os círculos pretos correspondem aos obstáculos detectados e os pontos em ciano representam os vértices do Diagrama de Voronoi determinados e já filtrados. O ponto em ciano circulado em vermelho identifica o vértice selecionado como mais distante de todos os obstáculos e para qual o robô deverá se direcionar. A linha vermelha, por sua vez, determina a frente do robô e para onde ela aponta.



**Figura 24: Ilustração da utilização do Diagrama de Voronoi para obtenção do ponto objetivo. Fonte: Autora.**

O ângulo  $\varphi$  entre o ponto objetivo e o robô é calculado em relação ao centro do mesmo através da seguinte equação:

$$\varphi = \operatorname{tg}^{-1} \left( \frac{y_{\text{objetivo}} - y_{\text{robô}}}{x_{\text{objetivo}} - x_{\text{robô}}} \right) - \theta \quad (3.1)$$

Nesta equação,  $\theta$  é fornecido pelo próprio simulador e corresponde ao ângulo do robô em relação ao *frame* global.

## 5 RESULTADOS E DISCUSSÕES

As simulações foram realizadas em duas cenas criadas no CoppeliaSIM na qual a cena da Figura 25 privilegia a existência de obstáculos, corredores e a passagem por portas como desafio, enquanto a cena da Figura 26 privilegia um espaço com menos obstáculos horizontais e mais obstáculos em profundidade.



Figura 25: Cena 1 criada no CoppeliaSIM. Fonte: Autora.

Baseados no manual do fabricante do robô *Pioneer*, no tempo necessário para realizar as leituras dos sensores e rodar os scripts, tanto dentro do CoppeliaSIM quanto no MATLAB, determinou-se, de forma empírica, que a velocidade linear máxima será de 0,05m/s e a velocidade angular máxima de 10 rad/s.

Essas velocidades são baixas porque: (I) existe uma criança com limitações motoras e neurológicas que não deve, em hipótese nenhuma, ser exposta a riscos, já que não teria um tempo de reação ótimo; (II) essas crianças também possuem dificuldade em sustentar seus troncos e cabeça, o que torna pouco interessante o

emprego de velocidades mais altas; (III) essas velocidades se mostraram seguras para que todo o processamento do sistema de controle ocorresse.



**Figura 26: Cena 2 criada no CoppeliaSIM. Fonte: Autora.**

Após essa definição, os primeiros testes realizados simularam os comandos dados pela criança como sendo sempre “Ir pra frente”. Isto porque fez-se necessário avaliar a resposta dos controladores em um trajeto mais homogêneo.

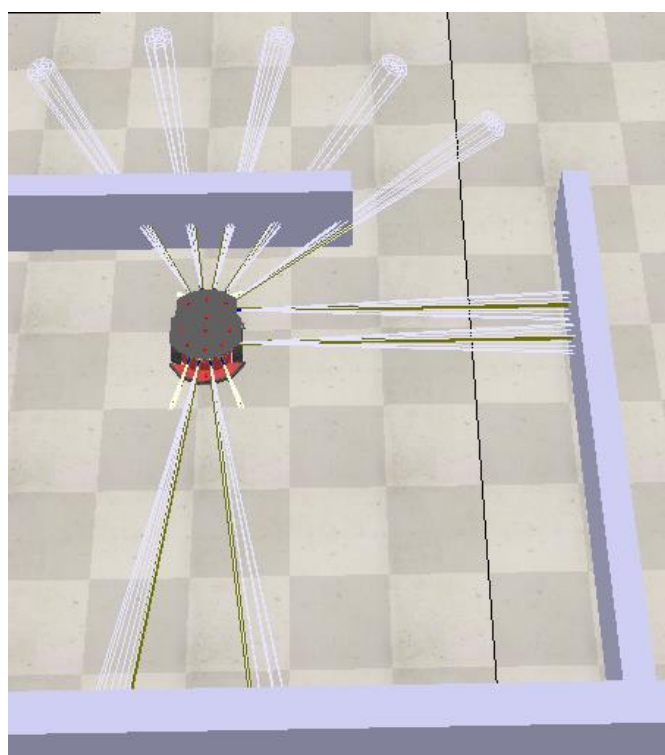
Os testes seguintes se deram pela aplicação dos quatro comandos possíveis, sendo gerados de forma randômica pelo algoritmo, para que pudessem ser analisadas todas as situações que a criança poderia enfrentar. Este tipo de simulação dos controles dados pelo usuário cria situações inusitadas como “Virar à direita” de forma constantes, descrevendo círculos sequencialmente, seguido por um “Virar à esquerda” que ocorre da mesma forma.

Situações como essa não foram excluídas ou tratadas pelos controladores visto que, se a criança não se encontra em risco de queda ou colisão, é de sua vontade permanecer girando e a ordem do usuário é soberana nesse contexto.

## 5.1 PRIMEIRA SITUAÇÃO: OBSTÁCULOS E PASSAGEM PELA PORTA

Na primeira situação demonstrada pela Figura 27, o robô encontra-se dentro de uma sala e cercado por paredes, porém, sem nenhum obstáculo no chão. Os sensores que estão detectando algum tipo de obstáculo podem ser visualizados.

Perceba que os 4 sensores de leitura do chão estarão sempre ativados pois estão sempre detectando o mesmo, uma vez que não há desnível. O robô também foi posicionado de forma que a distância mínima entre o obstáculo mais próximo e o carrinho fosse de 0,5147 m. Este obstáculo, como visualizado na imagem, encontra-se na traseira do carro e, o comando inicial dado pelo usuário, indicará que o deslocamento ocorrerá para a frente.

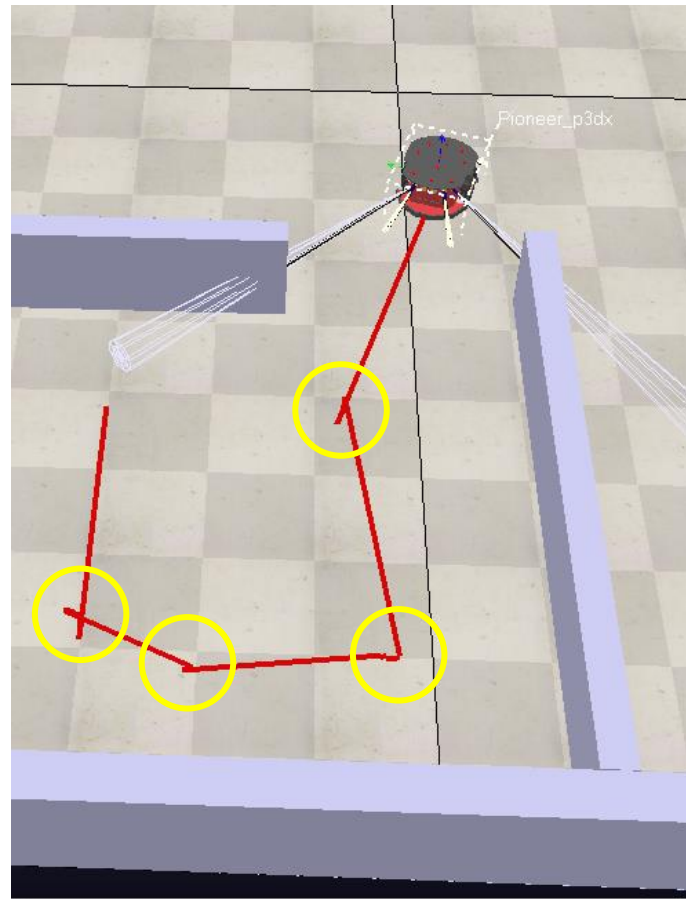


**Figura 27: Ponto inicial da trajetória do robô. Fonte: Autora.**

Uma vez que o robô completou a tarefa inicial de se locomover dentro do cômodo, buscando sempre que possível “Ir pra frente” como o usuário deseja, e passar pela porta existente entre os ambientes, a simulação foi pausada. Este momento é mostrado na Figura 28, na qual também é possível observar o rastro vermelho que evidencia o caminho percorrido pelo robô desde o início.

É muito interessante perceber que o sistema de controle compartilhado foi eficiente em lidar com situações com objetos detectados nas mais diferentes zonas

do robô, além de ser capaz de ajustar a posição para que a criança realizasse a passagem pela porta sem nenhuma colisão.

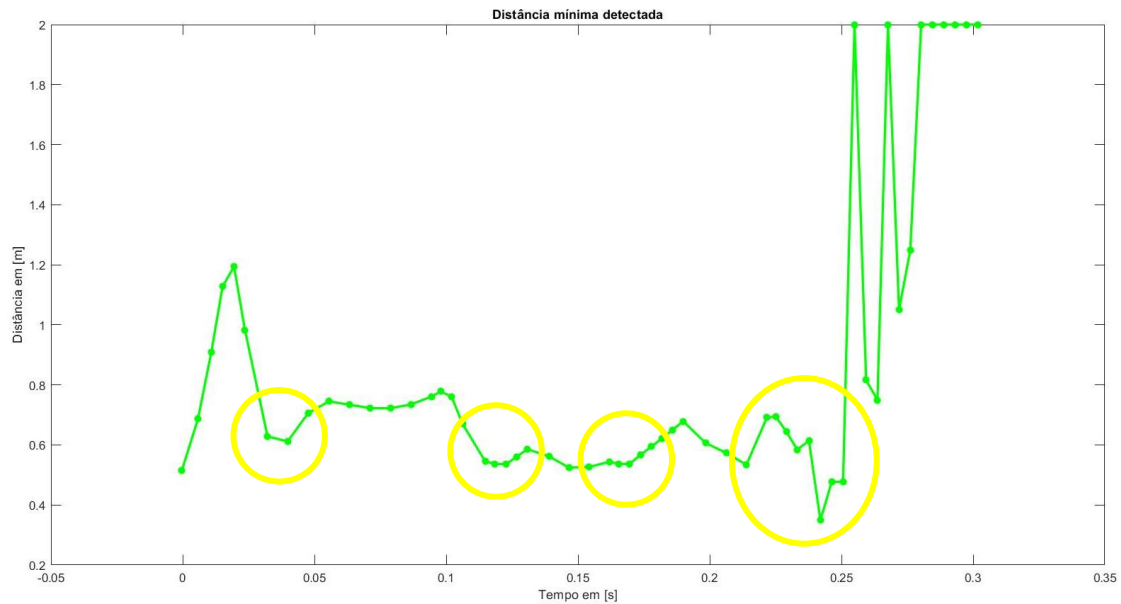


**Figura 28: Ponto final onde a simulação foi pausada. Em vermelho tem-se a trajetória do robô.**  
**Fonte: Autora**

O Gráfico 1 mostra os valores mínimos detectados pelos sonares ultrassônicos horizontais. Percebe-se que, ao final, essas distâncias encontram-se muito próximas ou iguais a 2m. Isso porque, em caso de não-detecção de objetos por um sonar, este assume o valor máximo igual a 2.

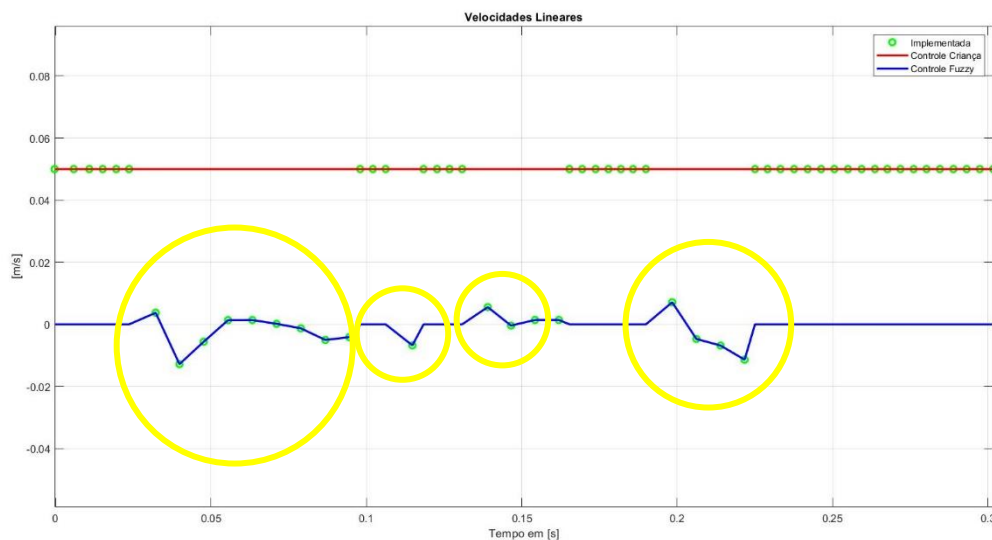
Outro fator digno de nota é a variação de valores que se encontravam entre 2m e aproximadamente 1m durante o tempo de 0,25s a 0,3s. Este foi o momento em que o robô passava pela porta e adentrava um amplo espaço com poucos obstáculos. Assim, com a leitura dos sensores ocorrendo de forma sequencial e, apenas após a execução do algoritmo em MATLAB, essa obtenção de dados foi um pouco prejudicada, uma vez que, como a porta é fina, ela poderia em dado momento, estar no ponto cego dos sonares, como ocorreu.

Como é possível observar, em nenhum momento o robô adentrou a zona de risco (distâncias menores que 0,4m) e, portanto, o controle realizado com base no Diagrama de Voronoi não foi aplicado.



**Gráfico 1: Distância mínima dos objetos detectados em relação ao robô durante o tempo da simulação. Fonte: Autora.**

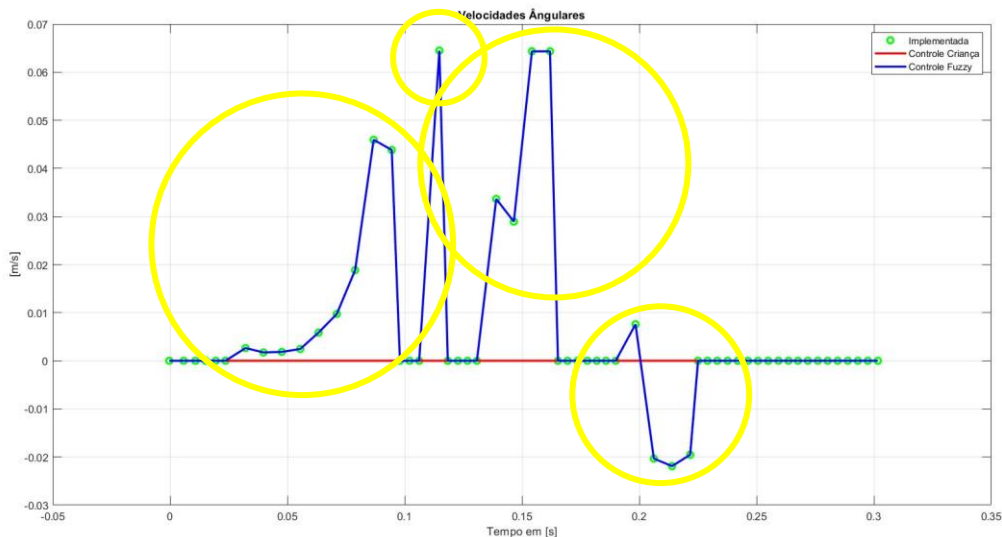
O Gráfico 2 apresenta as velocidades lineares aplicadas ao longo do percurso, assim como a velocidade linear correspondente ao comando dado pela criança que, para “Ir pra frente”, a velocidade máxima linear de 0,05m/s. Quando o controle *fuzzy* é ativado, sua velocidade varia, caso contrário, é informada como 0. Os círculos



**Gráfico 2: Velocidades lineares aplicadas ao robô ao longo da trajetória comparadas às velocidades dadas pelo usuário e pelo controlador *fuzzy*. Fonte: Autora**

verdes mostram qual velocidade de fato foi aplicada no robô, se a do usuário ou a fornecida pelo *fuzzy*.

O Gráfico 3, por sua vez, mostra as velocidades angulares para a mesma situação. Neste caso, ao performar o comando “Ir pra frente”, a velocidade angular empregada pelo usuário é 0. Assim como no gráfico anterior, quando *fuzzy* não estava ativado, o valor registrado é 0. Isso se dá pela disposição dos dados na planilha que gerou o gráfico.



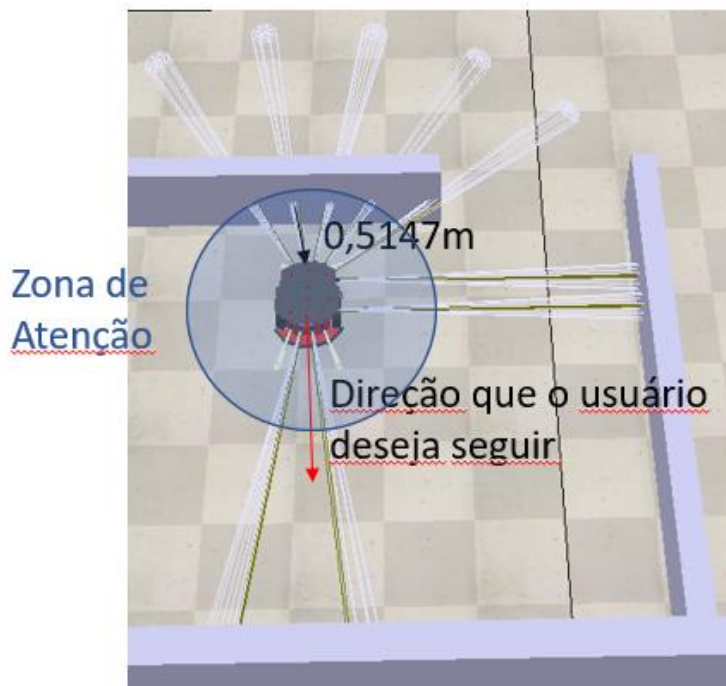
**Gráfico 3: Velocidades angulares aplicadas ao robô ao longo da trajetória comparadas às velocidades dadas pelo usuário e pelo controlador fuzzy. Fonte: Autora**

Fica evidente, então, a forma de atuação do controle compartilhado quando comparamos o Gráfico 1, contendo as distâncias mínimas, o Gráfico 2, contendo as velocidades lineares e Gráfico 3, o qual, por sua vez, evidência as velocidades angulares.

Isso porque, mesmo com distâncias mínimas suficientes para adentrar a zona de atenção, se a zona do robô que contenha o obstáculo não é a mesma que a criança deseja seguir, não há problema. Ela está se tirando, sozinha, da situação de risco e o controlador não deve interferir. A Figura 29 ilustra essa ideia de forma mais clara.

O importante é ressaltar a principal ideia sendo discutida ao longo de todo este trabalho: a criança deve ter direito a expressar suas vontades ao controlar o carrinho, contanto que ela não se coloque em risco. Os controladores foram desenhados para auxiliar a condução do robô, e não realizar o trabalho todo pela criança.

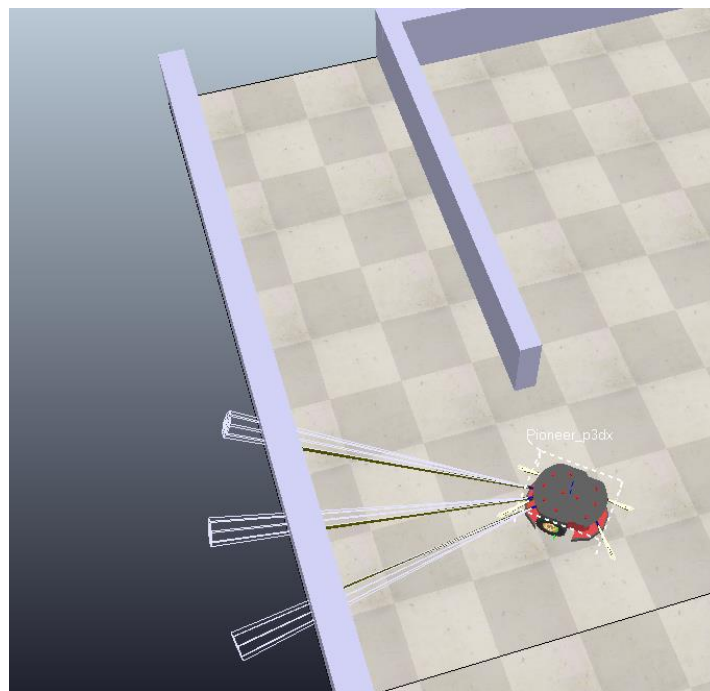




**Figura 29:** O obstáculo encontra-se dentro da zona de atenção, porém, a própria criança está se retirando do perigo e não há necessidade para atuação do *fuzzy*. Fonte: Autora.

## 5.2 SEGUNDA SITUAÇÃO: CORREDOR E DESNÍVEL

A segunda situação avaliada mostra um trajeto contendo um corredor com um desnível ao final, como mostrado na Figura 30.



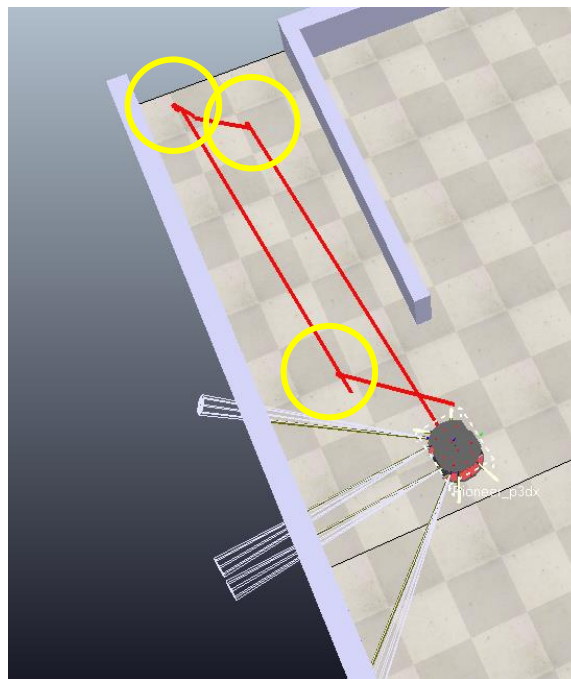
**Figura 30:** Situação 2 - Navegação em um corredor com um desnível profundo ao final. Fonte: Autora.



O ponto final da trajetória, considerado para interromper a simulação, se deu quando o *Pioneer* saiu do corredor em direção ao amplo salão, como mostrado na Figura 31. Em vermelho, podemos observar que a trajetória sofreu ajustes, e o robô não seguiu em uma linha reta centralizada em relação ao corredor, porém, foi capaz de realizar todo o percurso em segurança.

Neste experimento a representação da criança sobre o robô foi conduzida ao longo de todo o corredor com o comando “Ir pra frente” acionado, isto é, a criança escolheu sempre ir para frente independente dos possíveis desafios afrente, o que pode ser uma vontade da criança, ou então, uma situação de espasmos involuntários, que a criança não controla o robô por vontade própria e sim porque seus músculos estão enrijecidos, e dá o comando de seguir em frente.

Ao ser deparada com o desnível, o robô retirou o controle da criança e se reorientou girando de forma a voltar-se novamente para o corredor e percorrê-lo no sentido de retorno ao salão.



**Figura 31: Momento em que o robô retorna ao salão após o percurso. Fonte: Autora.**

O Gráfico 4 mostra os valores avaliados para as distâncias mínimas dos objetos detectados. Ao correlacionarmos estes valores com as velocidades lineares, Gráfico 5, e angulares, Gráfico 6, percebemos os momentos críticos de acionamento de cada um dos controladores, principalmente quando Voronoi se faz necessário. Este é o ponto em que o robô se encontra frente ao desnível.

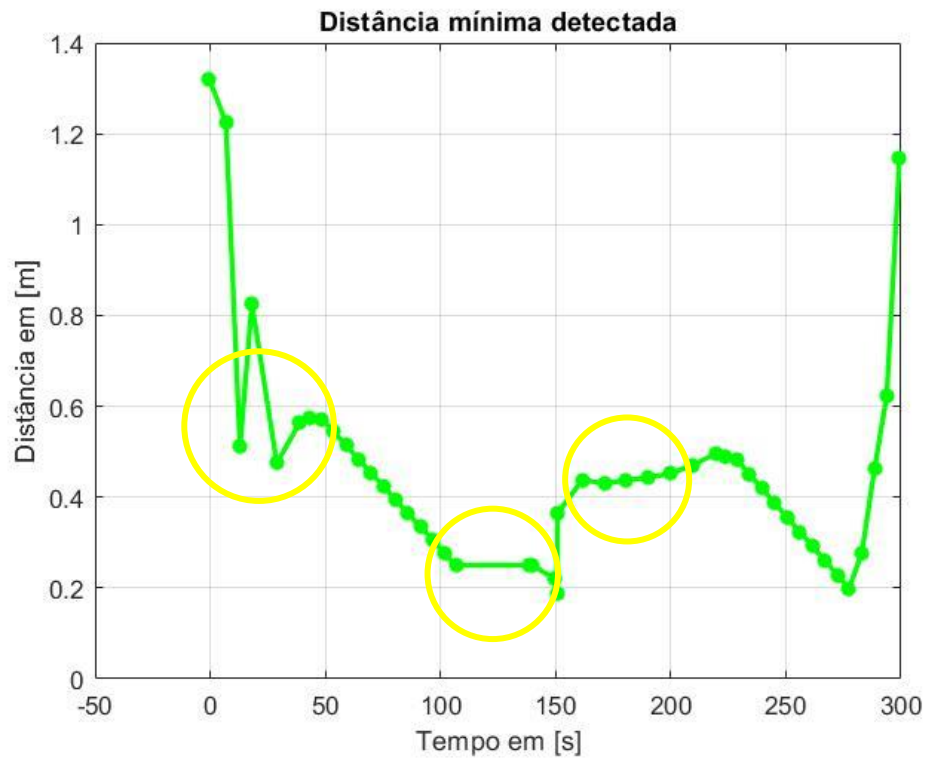


Gráfico 4: Distância mínima para os objetos detectados. Fonte: Autora

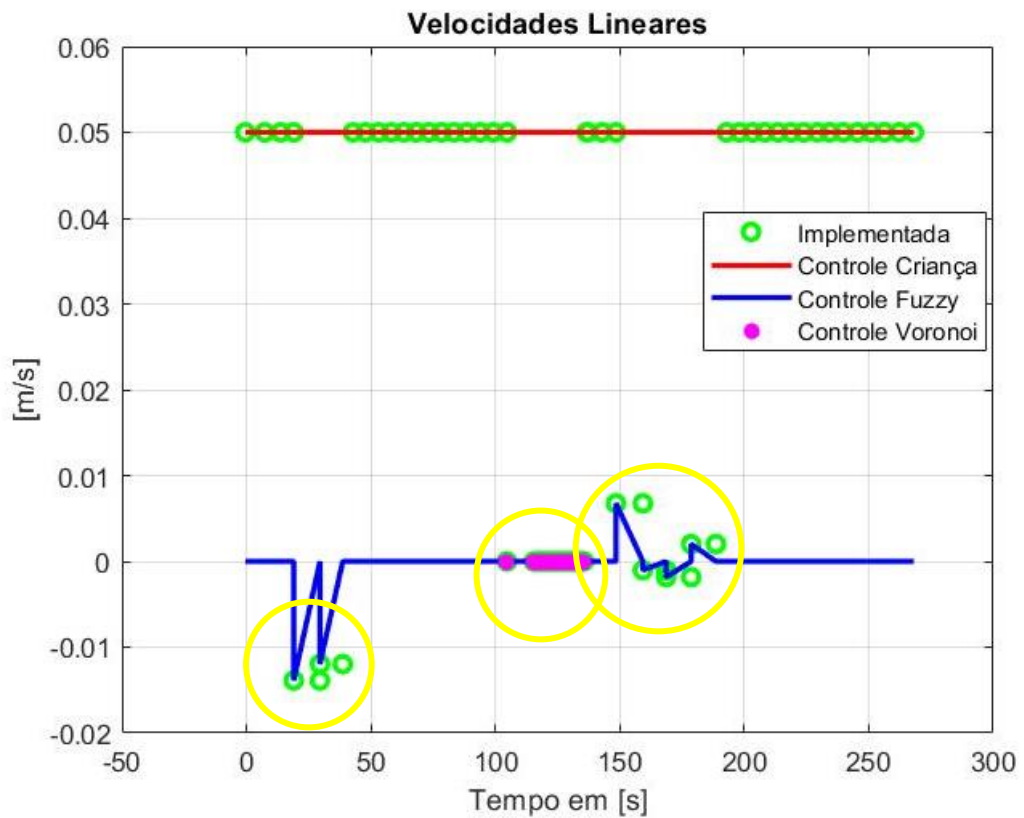
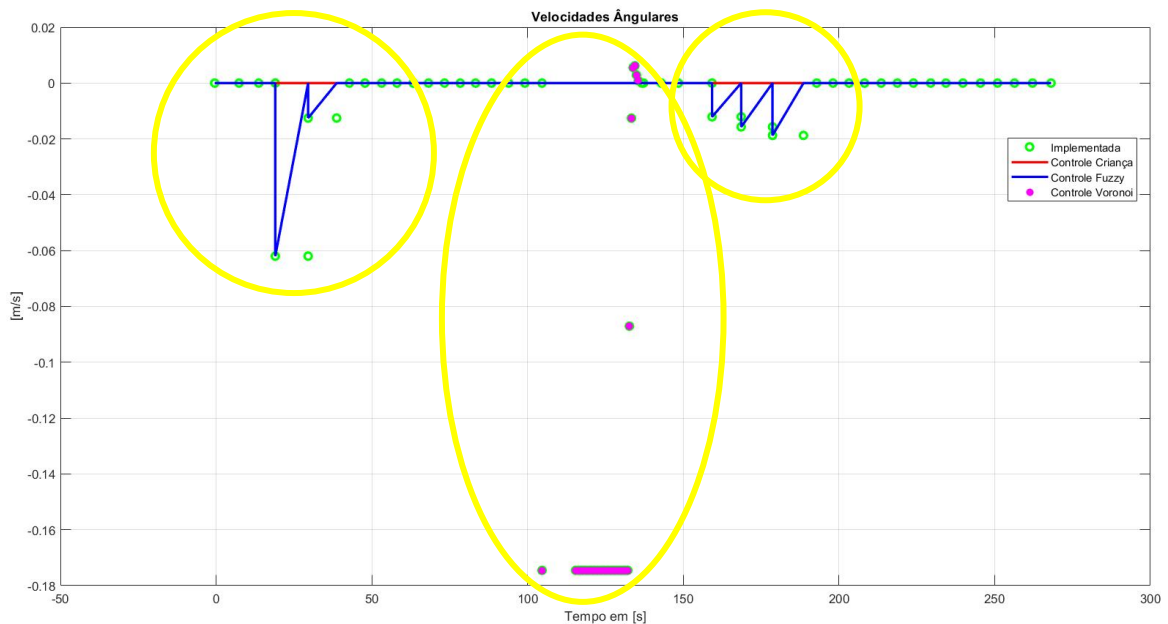


Gráfico 5: Velocidades lineares obtidas durante o percurso. Fonte: Autora.

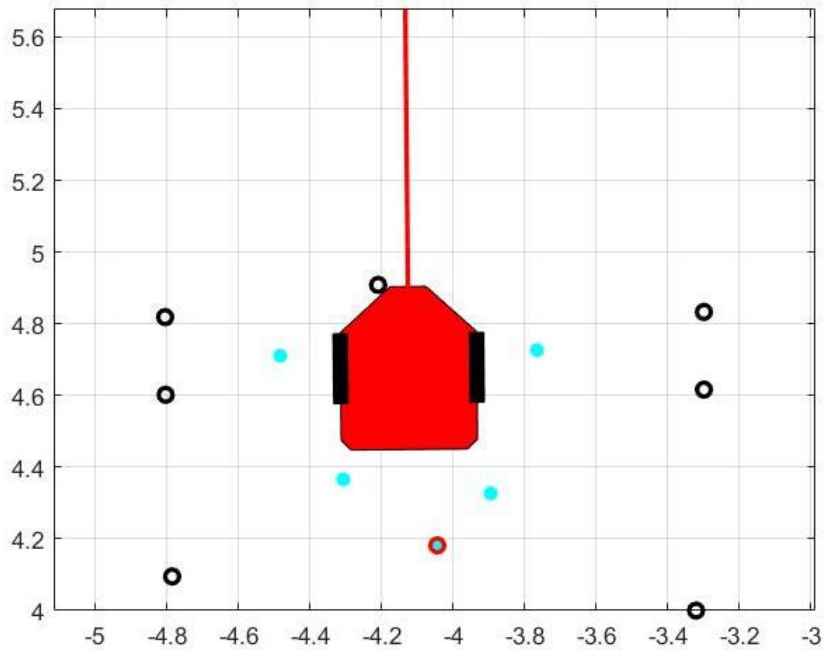
Um fator interessante relacionado ao Gráfico 5 é a presença de *outliers*. Provavelmente, esse fato se deu pela diferença entre o momento em que a saída do controlador *fuzzy* foi obtida e o momento de registro da mesma no *array* de dados. Isto aconteceu apenas em momentos em que o controlador *fuzzy* atuava e exige maior investigação. Contudo, ainda sim, esses *outliers* são consistentes com formato da curva assumido pelo sistema de controle mencionado. De forma muito semelhante se comportam os *outliers* do Gráfico 6.



**Gráfico 6: Velocidades angulares obtidas no trajeto. Fonte: Autora.**

Entretanto, o fator mais importante e relevante, observado tanto no Gráfico 5 quanto no Gráfico 6, é a atuação pontual do controlador baseado em Voronoi. Este freia completamente o robô, reduzindo sua velocidade linear a zero durante o período de atuação, atuando apenas sobre a velocidade angular para efetuar o giro. Percebe-se também que o valor desta velocidade angular é negativo, mostrando que o robô girou para a direita, como evidenciado no rastro da Figura 31.

Por fim, a Figura 32 mostra o *plot* gerado do mapa local advindo do sensoriamento. Os obstáculos são exibidos como círculos negros vazados, inclusive o declive que pode ser visualizado bem próximo a região a frente e a esquerda do robô. Os vértices do Diagrama de Voronoi podem ser vistos em ciano e, o que foi circulado de vermelho, corresponde ao ponto objetivo escolhido para que o robô se redirecione para evitar a queda.



**Figura 32: Plot do mapa local de obstáculos associado aos vértices do Diagrama de Voronoi traçado. O ponto objetivo está identificado de vermelho. Fonte: Autora.**

### 5.3 TERCEIRA SITUAÇÃO: DESNÍVEL SEM A PRESENÇA DE OBSTÁCULOS HORIZONTAIS

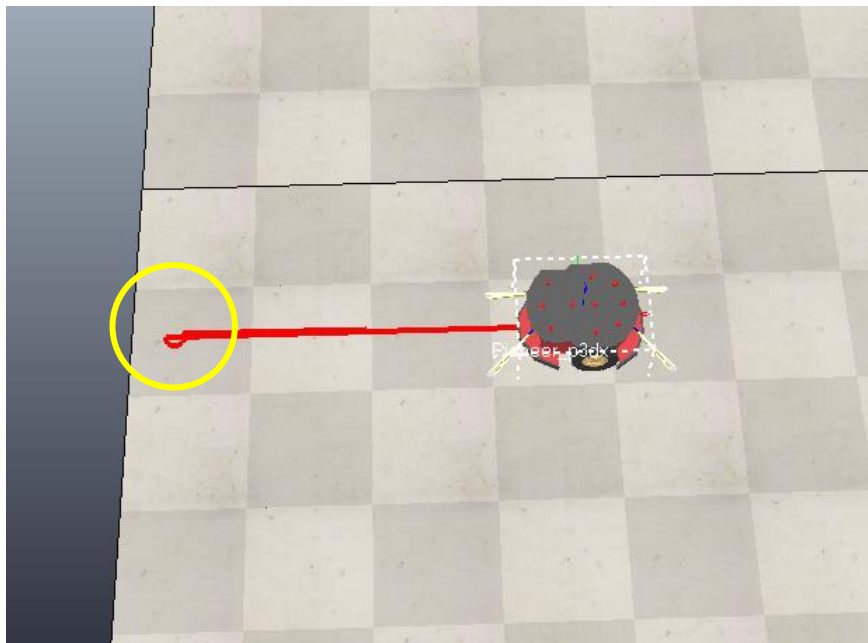
Esta situação foi, sem sombra de dúvidas, a mais desafiadora deste trabalho, uma vez que a ausência de obstáculos horizontais impede que o mínimo de três pontos para a construção do diagrama de Voronoi seja obtido. Diagrama este que é a base para atuação do controle em situações críticas. O uso do controlador *fuzzy* também não era uma opção interessante visto que o foco de atuação dos mesmos relacionava-se à presença de obstáculos detectados pelos sensores horizontais.

Assim sendo, para situações como a da Figura 33, foi estabelecido um protocolo de contingência para o qual, caso fosse detectado um declive, porém, não houvesse outros pontos de apoio para cálculo do diagrama de Voronoi, a velocidade linear do robô seria reduzida a zero e o mesmo efetuaria um giro de  $180^\circ$  para se opor ao obstáculo do chão. Assim, a criança poderia se dirigir a uma área segura, evitando a ocorrência de um acidente.



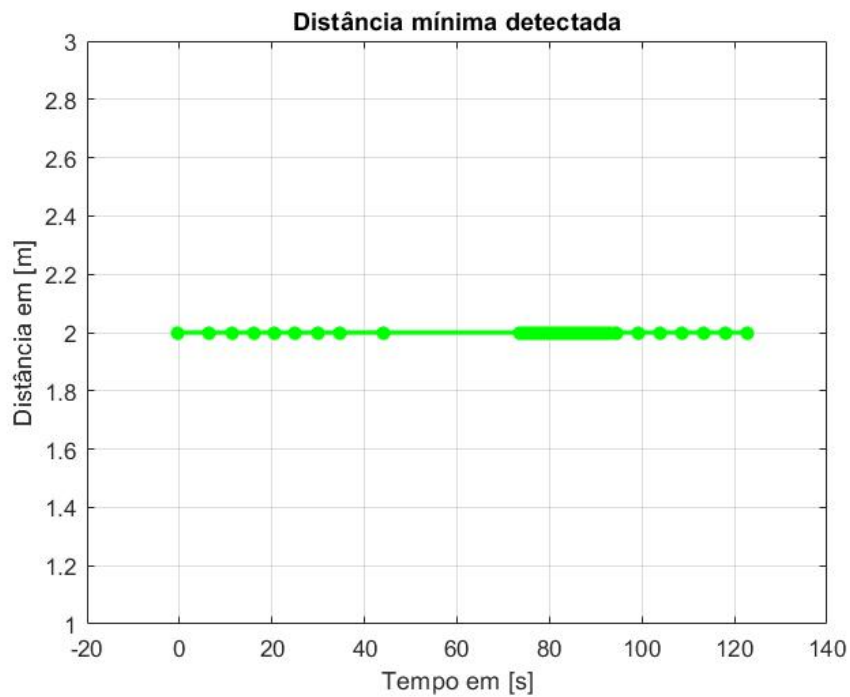
**Figura 33: Situação para a qual o protocolo de contingência é acionado.**  
Fonte: Autora.

A Figura 34 mostra o rastro do trajeto evidenciado que o robô seguiu em frente, em direção ao declive, devido ao comando “Ir pra frente” fornecido pelo usuário, mas, após a identificação do obstáculo no solo, realizou o giro em sentido contrário e retornou pelo mesmo caminho, retomando o comando dado pela criança.

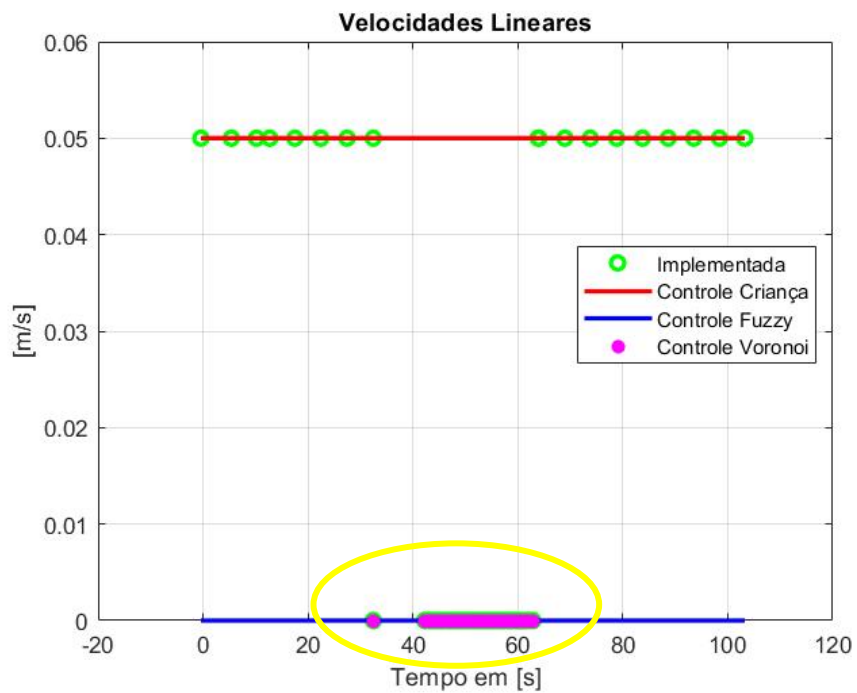


**Figura 34: Trajeto realizado pelo robô evidenciado o retorno realizado diante do obstáculo no chão.** Fonte: Autora.

Os Gráficos 7, 8 e 9 mostram o contraste entre as distâncias mínimas detectadas, as velocidades lineares e angulares aplicadas.



**Gráfico 7: Distâncias mínimas detectadas mostrando que não houve detecção de objetos na horizontal.**  
Fonte: Autora.



**Gráfico 8: Velocidades lineares associadas ao trajeto.** Fonte: Autora.

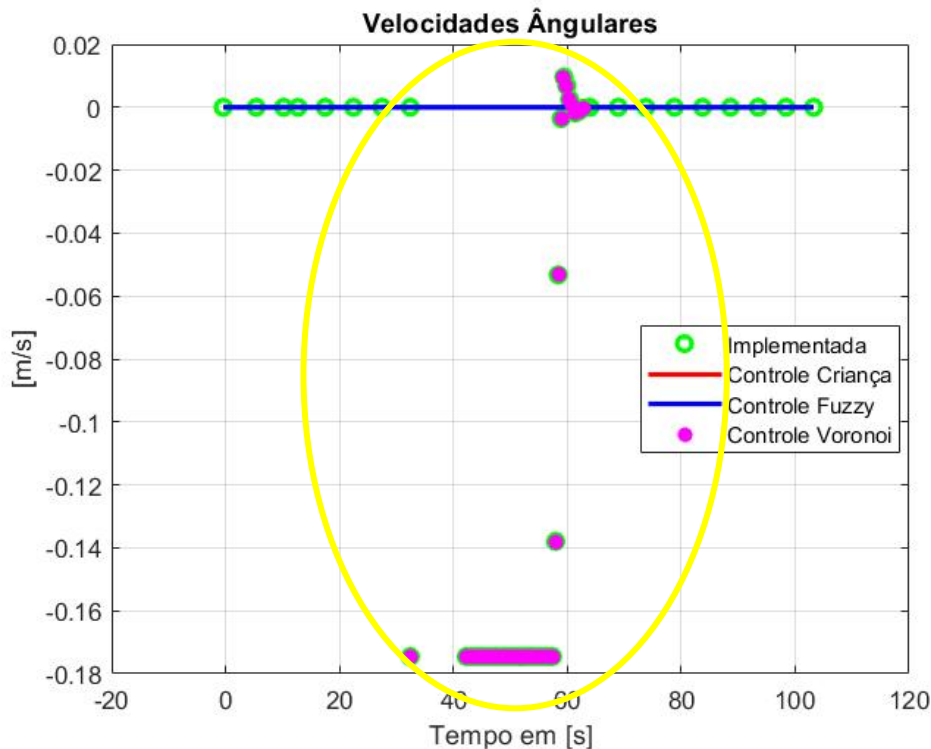


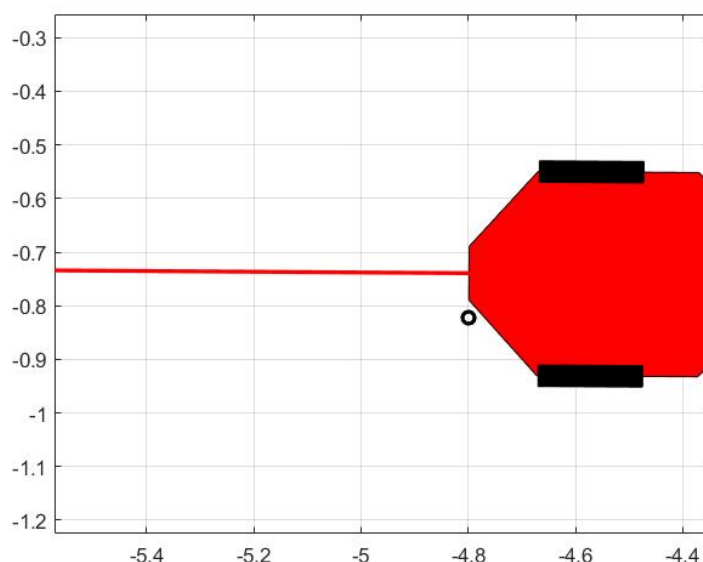
Gráfico 9: Velocidades angulares associadas ao trajeto. Fonte: Autora.

Como não há detecção de nenhum obstáculo pelos sensores na horizontal, as distâncias permanecem sempre iguais a 2m, correspondendo à distância máxima atribuída. É interessante perceber que, por consequência disso, em momento nenhum o controle *fuzzy* atuou. Dessa forma, a contingência atuou criando um ponto objetivo equivalente a um vértice do Diagrama de Voronoi e direcionando o robô para ele.

Um ponto interessante do Gráfico 9 é que entre 30s, aproximadamente, e quase 60s, o robô realiza o giro para a direita direcionando-se ao ponto objetivo. Porém, perto da marca de 60s, ele se desvia da direção e faz uma correção de trajetória virando à esquerda. Após este evento, o controle retorna para a criança, que segue atribuindo a direção “Ir pra frente”.

Também é relevante relatar que ocorre uma sobreposição entre a curva relativa ao controle da criança, uma vez que a velocidade angular associada a seguir em frente é zero, e a curva do *Fuzzy*, que, como o mesmo não atuou, é retratada como zero.

Por fim, a Figura 35 mostra a plotagem do mapa local de sensoriamento, evidenciando a ausência de obstáculos suficientes para que o Diagrama de Voronoi fosse desenhado e, por consequência, a ausência de vértices do mesmo. O obstáculo pode ser visualizado juntamente a frente esquerda do robô.



**Figura 35: Plotagem do mapa local evidenciando apenas um obstáculo de solo e a ausência de vértices do Diagrama de Voronoi. Fonte: Autora.**

Ressalta-se que apenas um obstáculo é evidenciado por ter sido o primeiro ponto em que o sensor deixou de detectar o solo. O segundo sonar frontal passa pelo mesmo processo em sequência, porém, o robô já está atuando na correção da posição e esse dado é considerado, internamente, sem ser plotado.

#### 5.4 QUARTA SITUAÇÃO: TODOS OS CONTROLES DE COMANDO HABILITADOS

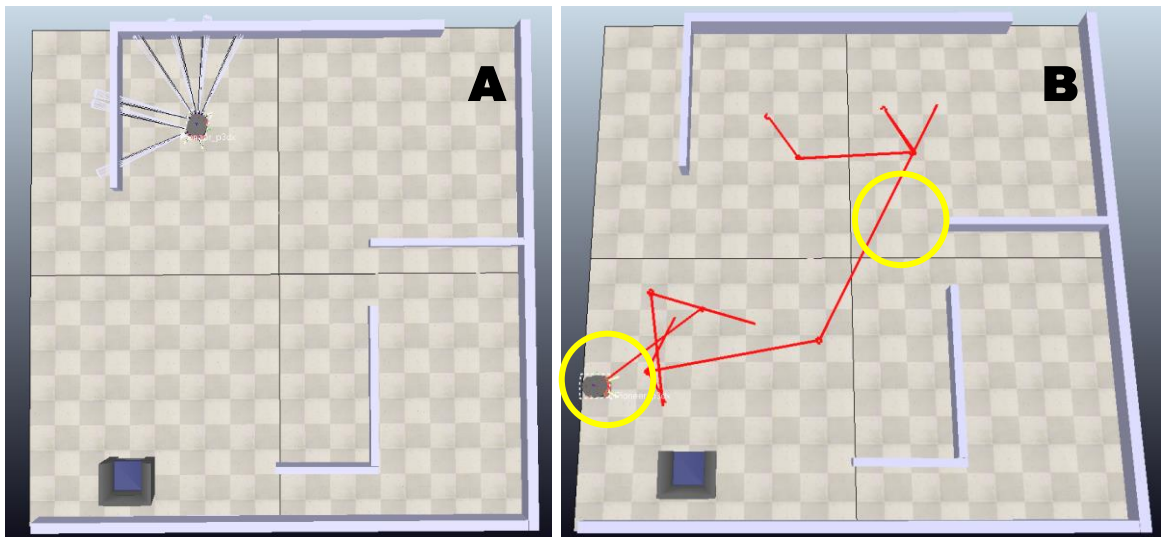
A quarta e última situação mostrada neste trabalho associa todas as possibilidades de controle por parte da criança (“Ir pra frente”, “Ré”, “Virar à direita” e “Virar à esquerda”) a uma circulação livre pela segunda cena que oferece tanto os obstáculos horizontais quanto uma maior quantidade de obstáculos de solo.

A Figura 36 (A) mostra o ponto de partida do robô e a Figura 36 (B) o ponto em que o programa foi finalizado devido ao tempo de execução. Em vermelho, encontra-se a trajetória do robô, a qual é notoriamente randômica.

O que ocorre nessa simulação é uma atribuição aleatória de comandos do usuário para o carrinho em um universo em que toda e qualquer sequência de ordens é possível. Justamente por essa questão aleatória, o teste específico dos sistemas de controle seria complexo, caso esta fosse a única forma de fazê-lo. Isto porque o



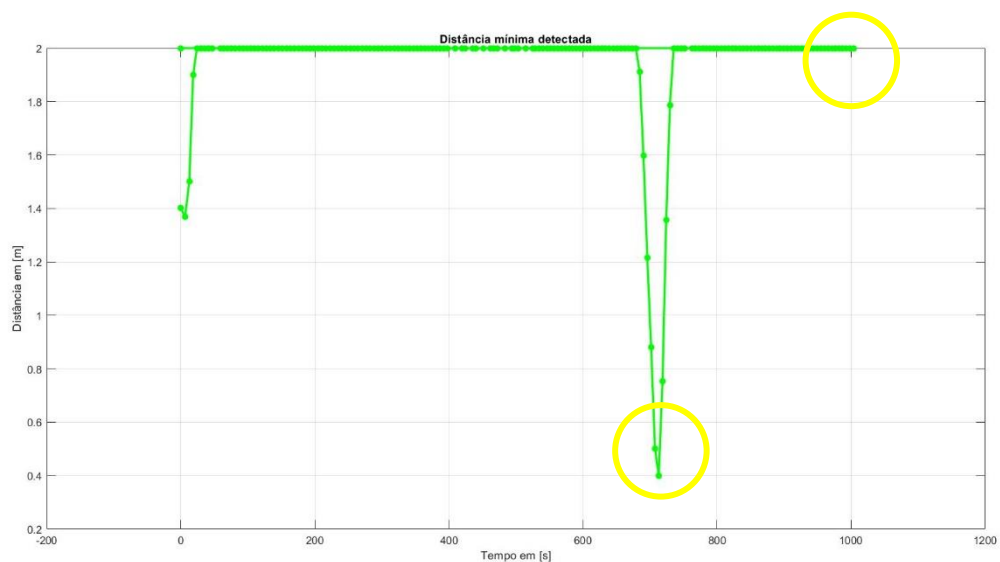
programa pode rodar inteiramente sem que, em nenhum momento, a criança se coloque em uma situação de risco de queda ou colisão.



**Figura 36: (A) Momento do início da trajetória. (B) Momento em que o programa foi finalizado, evidenciando a trajetória realizada. Fonte: Autora.**

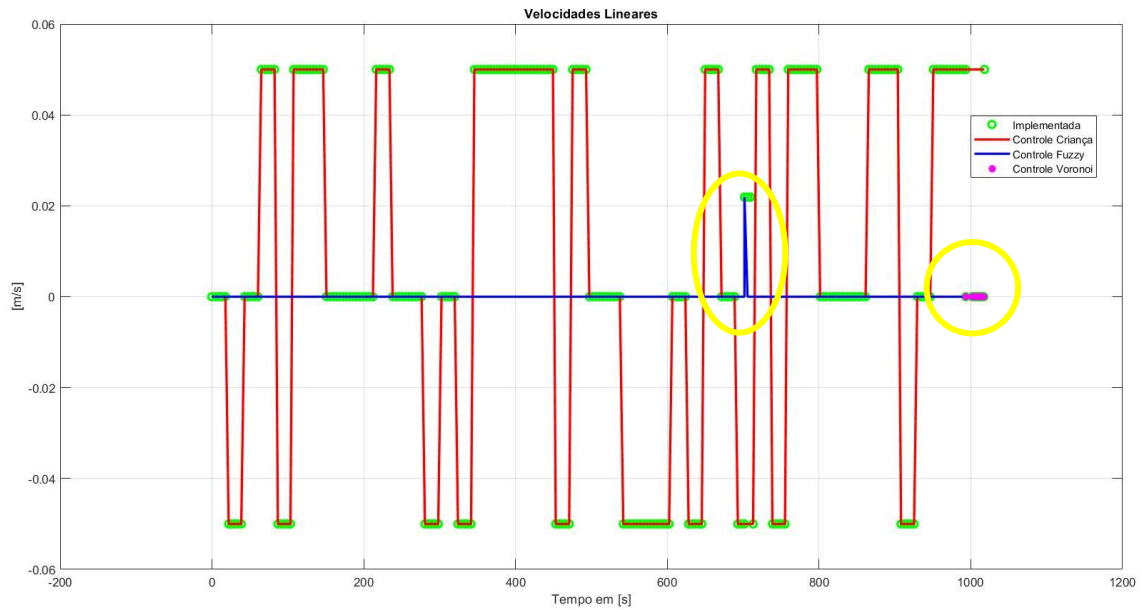
Para esta trajetória, o momento mais crítico ocorreu pouco antes do programa ser finalizado. O robô se dirigia ao obstáculo de solo, porém, a contingência atuou redirecionando-o a um local seguro, então o programa foi finalizado.

O Gráfico 10 mostra as distâncias mínimas detectadas para obstáculos na horizontal. Contudo, raros foram os momentos em que a criança se encontrava diante de ou cercada por obstáculos.

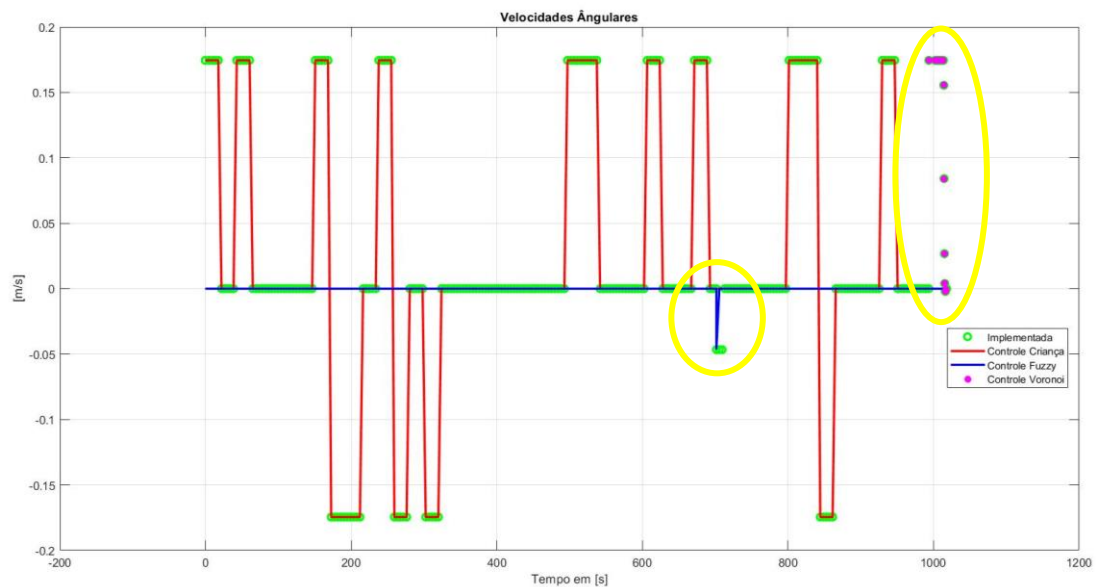


**Gráfico 10: Distâncias mínimas detectadas em relação aos objetos na horizontal ao longo da trajetória. Fonte: Autora.**

Os Gráficos 11 e 12 mostram as velocidades lineares e angulares atribuídas as trajetórias, seja por comando do usuário, seja por atuação do controlador.



**Gráfico 11: Velocidades lineares atribuídas ao longo da trajetória. Fonte: Autora.**



**Gráfico 12: Velocidades angulares atribuídas ao longo da trajetória. Fonte: Autora.**

Percebe-se claramente o momento em que o controlador *fuzzy* atuou, sendo este o momento em que o robô se encontrava diante da cadeira, buscando seguir em frente. Também notamos que, no início do percurso, ainda que o robô se encontrasse

perto de obstáculos, os mesmos não ofereciam risco ao prosseguimento da trajetória estabelecida pelo usuário. Dessa forma, não houve atuação do controle.

Por fim, nota-se a atuação do controle baseado no Diagrama de Voronoi, o qual efetuou um giro a esquerda para que o robô saísse da borda pela qual poderia cair, voltando-se para a sala novamente.

Os desafios promovidos por controles aleatórios poderiam ser testados a exaustão em ambientes de simulação, mas são onerosos em termos de horas dedicadas para que, talvez, nada significativo possa ser extraído. São testes necessários, mas devem ser feitos de forma comedida.

## 6 CONCLUSÕES

Diante de um cenário desafiador, envolvendo uma criança com mobilidade reduzida devido a paralisia cerebral, vivendo em um mundo pouco adaptado e preparado para atender suas necessidades, concluímos que um controle compartilhado que apresente diferentes níveis de segurança e, ainda garanta a liberdade de movimentação do indivíduo, é possível através da combinação da Lógica *Fuzzy* com o Diagrama de Voronoi.

Entretanto, apesar de ter atendido de forma satisfatória a proposta deste trabalho, cenários complexos podem conduzir à falha do sistema de controle e um possível acidente.

Como situações assim não são desejáveis, o próximo passo para a continuidade deste trabalho se dá pela melhoria do sistema de controle existente e uma modelagem mais acurada para circunstâncias de alta complexidade ou para as quais o algoritmo não prevê resposta. Para tanto, é necessário não apenas a melhoria do algoritmo em si, mas a conversa com especialistas na área de fisioterapia e familiares das crianças para entender quais situações complexas poderiam ocorrer.

Em seguida, a continuação natural será a implantação do controlador nos carrinhos desenvolvidos no projeto *Adapt*. A partir disso, será possível refinar os parâmetros do mesmo e verificar se atendem a necessidade da criança em ambientes reais, controlados ou não.

Uma vez que o mundo real apresenta mais desafios do que os estudados nesse trabalho, outras teorias e lógicas de controle poderão ser incorporadas para que a criança possa utilizar o carrinho em seu máximo potencial e se beneficiar da liberdade de locomoção e melhoria do convívio social.

## REFERÊNCIAS

**AURENHAMMER, F.** “*Voronoi diagrams - A survey of a fundamental geometric data structure*”. ACM Computing Surveys, n. 233, v. 3, p. 45-405, 1991.

**BARBOSA, B. M.; OLIVI, L. R.; COSTA, E. B.** “*Shared Control for Robotic Wheelchair Using Fuzzy Logic*”, 14<sup>o</sup> Simpósio Brasileiro de Automação Inteligente (SBAI 2019), 2019.

**BHATTACHARYA, P.; GAVRILOVA, M. L.** “*Voronoi diagram in optimal path planning*”, 4th International Symposium on Voronoi Diagrams in Science and Engineering, pp. 38-47, 2007.

**BOOTS, B. N.; OKABE, A.; KOKICHI, S.** “*Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*”, John Wiley and Sons Ltd., 1995.

**BRASIL.** “*Diretrizes de atenção à pessoa com paralisia cerebral*”, Ministério da Saúde, Secretaria de Atenção à Saúde, Departamento de Ações Programáticas Estratégicas, Brasília, 2013. Disponível em: [http://bvsmms.saude.gov.br/bvs/publicacoes/diretrizes\\_atencao\\_paralisia\\_cerebral.pdf](http://bvsmms.saude.gov.br/bvs/publicacoes/diretrizes_atencao_paralisia_cerebral.pdf), último acesso 19/02/2022.

**CANS, C.; DOLK, H.; PLATT, M. J.; COLVER, A.; PRASAUSKIENE, A.; KRÄGELOH-MANN, I.** “*Recommendations from the SCPE collaborative group for defining and classifying cerebral palsy*”. Developmental Medicine and Child Neurology, v. 49, p. 35-38, 2007.

**CHEN, P.; XIAOQING, L.; JIVANG, D.; LINFEI, Y.** “*Research of path planning method based on the improved Voronoi diagram*”, 25th Chinese Control and Decision Conference (CCDC), pp. 2940-2944, 2013.

**DONG, H.; ZHU, W.; LI, J.; DUAN, S.** “*The Path Planning for Mobile Robot Based on Voronoi Diagram*”, Third International Conference on Intelligent Networks and Intelligent Systems, pp. 446-449, 2010.

**EVANS, N.; BAINES, R.** “*Trends, goals and outcomes for children and families using early powered mobility in a charitable loan scheme*”, Journal of Enabling Technologies, Vol. 11 No. 4, pp. 138-147, 2017. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/JET-08-2017-0032/full/html>, último acesso 19/02/2022.

**FERNANDES, R. T.** “*Supervisão de um Sistema Híbrido eólico/diesel usando Lógica Fuzzy*”. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Mato Grosso do Sul, 2005.

**GALLOWAY, C.** “Go Baby Go”, Cerebral Palsy Foundation, Disponível em: <https://www.yourcpf.org/cpproduct/go-baby-go-the-ultimate-toy-hack/>, último acesso: 19/02/2022;

**HSU, C.; CHEN, Y.; LU, M.; LI, S.** “Optimal path planning incorporating global and local search for mobile robots”, The 1st IEEE Global Conference on Consumer Electronics pp. 668-671, 2012.

**HUANG, H.H.; CHEN, Y.M.; HUANG, H.W.; SHIH, M.K.; HSIEH, Y.H.; CHEN, C.L.** “Modified Ride-On Cars and Young Children with Disabilities: Effects of Combining Mobility and Social Training”. *Frontiers in Pediatrics* pp. 5-299, 2018.

**KENYON, L. K.; HOSTNICK, L.; McELROY, R.; PETERSON, C.; FARRIS, J. P.** “Power Mobility Training Methods for Children: A Systematic Review”, *Pediatric physical therapy: the official publication of the Section on Pediatrics of the American Physical Therapy Association* n. 30, v. 1, p. 2-8., 2018.

**LEE, C.-C.** “Fuzzy logic in control systems: fuzzy logic controller I”. *IEEE Transactions on systems, man, and cybernetics*, v. 20, n. 2, p. 404-418, 1990.

**LEITE, J. M. R. S.; PRADO, G. F.** “Paralisia cerebral Aspectos Fisioterapêuticos e Clínicos”. *Revista Neurociências*, v. 12, n. 1, p. 41–45, 2004. Disponível em: <https://periodicos.unifesp.br/index.php/neurociencias/article/view/8886>, último acesso: 15/02/2022.

**LIVINGSTONE, R.; FIELD, D.** “Systematic review of power mobility outcomes for infants, children and adolescents with mobility limitations”, *Clinical Rehabilitation*, Vol. 28 No. 10, pp. 954-64, 2014.

**LIVINGSTONE, R.; FIELD, D.** “The child and family experience of power mobility: a qualitative synthesis”, *Developmental Medicine and Child Neurology*, Vol. 57 No. 4, pp. 317-27, 2014.

**MAMDANI, E.H.; ASSILIAN, S.** “An Experiment In Linguistic Synthesis With A Fuzzy Logic Controller”. *International Journal of Man-Machine Studies*, v. 7, n. 1, p. 1-13, 1975.

**MANCINI, M. C.; ALVES, A. C. M.; SCHAPER, C.; FIGUEIREDO, E. M.; SAMPAIO, R. F.; COELHO, Z. A. C.; TIRADO, M. G. A.** “Gravidade da paralisia cerebral e desempenho funcional”. *Revista Brasileira de Fisioterapia*, v. 8, n. 3, p. 253-60, 2004.

**MATARIC, M. J.** “Introdução à robótica”. São Paulo: Editora Blucher-unesp, 2014. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521208549/>, último acesso: 16/02/2022.

**NORVIG, P; RUSSEL, S.** “*Artificial Intelligence: A Modern Approach*”, Pearson Education Limited; 3ª edição, 2013.

**O’SHEA, T. M.** “*Diagnosis, treatment, and prevention of cerebral palsy*”, *Clinical Obstetrics and Gynecology*, v. 51, n. 4, p. 816-828, 2008.

**OLIVI, L.; SOUZA, R.; ROHME, E.; CARDOZO, E.** “*Shared Control For Assistive Mobile Robots Based On Vector Fields*”, 10th International Conference on Ubiquitous Robots and Ambient Intelligence, 2013.

**PATO, T. R.; PATO, T. R.; SOUZA, D. R. de; LEITE, H. P.** “*Epidemiologia da paralisia cerebral*”, *Acta Fisiátrica*, v. 9, n. 2, 2002. Disponível em: <https://www.revistas.usp.br/actafisiatrica/article/view/102365>, último acesso: 19/02/2022.

**RHYSBURGER, D.** “*Analytic delineation of Thiessen Polygons*”, *Geographical Analysis*, v. 5, n. 2, p. 133-144, 1973.

**ROMERO, R.** “*Robótica Móvel*”. Rio de Janeiro, 2014.

**SILVA, L. M.** “*Modelagem Fuzzy como subsídios para a espacialização da vulnerabilidade costeira à erosão*”. Dissertação de Mestrado, Programa de PósGraduação em Ciências Geodésicas e Tecnologias da Geoinformação, Universidade Federal de Pernambuco, 2013.

**SILVA, L. H.; ABREU E LIMA, E.** “*Projeto e protótipo de uma cadeira de rodas automatizada de baixo custo utilizando tubos de PVC*”. *Revista CIATEC-UPF*, v. 11, n. 2, p. 57-72, 2019.

**SIMÕES, M. G.** “*Controle e modelagem fuzzy*”, Editora Blucher, 2007.

**YOUSSEF, O. A. S.** “*Applications of fuzzy inference mechanisms to power system relaying*”, *IEEE Power Systems Conference and Exposition*, v.1, p. 560-567, 2004.

**ZADEH, L. A.** “*Fuzzy sets*”. *Information and Control*, v. 8, n. 3, p. 338-353, 1965.

**ZADEH, L. A.** “*Fuzzy sets as a basis for a theory of possibility*”. *Fuzzy Sets and Systems*, v. 100, n. 1, p. 9-34, 1999.



UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA

**Termo de Declaração de Autenticidade de Autoria**

Declaro, sob as penas da lei e para os devidos fins, junto à Universidade Federal de Juiz de Fora, que meu Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia Elétrica - Habilitação em Robótica e Automação Industrial é original, de minha única e exclusiva autoria. E não se trata de cópia integral ou parcial de textos e trabalhos de autoria de outrem, seja em formato de papel, eletrônico, digital, áudio-visual ou qualquer outro meio.

Declaro ainda ter total conhecimento e compreensão do que é considerado plágio, não apenas a cópia integral do trabalho, mas também de parte dele, inclusive de artigos e/ou parágrafos, sem citação do autor ou de sua fonte.

Declaro, por fim, ter total conhecimento e compreensão das punições decorrentes da prática de plágio, através das sanções civis previstas na lei do direito autoral<sup>1</sup> e criminais previstas no Código Penal<sup>2</sup>, além das cominações administrativas e acadêmicas que poderão resultar em reprovação no Trabalho de Conclusão de Curso.

Juiz de Fora, \_\_\_\_ de \_\_\_\_\_ de 20\_\_\_\_.

\_\_\_\_\_  
NOME LEGÍVEL DO ALUNO (A)

\_\_\_\_\_  
Matrícula

\_\_\_\_\_  
ASSINATURA

\_\_\_\_\_  
CPF

<sup>1</sup> LEI N° 9.610, DE 19 DE FEVEREIRO DE 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.

<sup>2</sup> Art. 184. Violar direitos de autor e os que lhe são conexos: Pena - detenção, de 3 (três) meses a 1 (um) ano, ou multa.