

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA  
CURSO ENGENHARIA ELÉTRICA COM HABILITAÇÃO EM ROBÓTICA E  
AUTOMAÇÃO INDUSTRIAL**

**Bruna Mendes Barbosa**

**CONTROLE COMPARTILHADO PARA CADEIRAS DE RODAS ROBÓTICAS  
UTILIZANDO LÓGICA *FUZZY***

**Juiz de Fora  
2018**

**Bruna Mendes Barbosa**

**CONTROLE COMPARTILHADO PARA CADEIRAS DE RODAS ROBÓTICAS  
UTILIZANDO LÓGICA *FUZZY***

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

Orientador: Prof. Leonardo Rocha Olivi

**Juiz de Fora  
2018**

**Imprimir na parte inferior, no verso da folha de rosto a ficha disponível em:**  
**<http://www.ufjf.br/biblioteca/servicos/usando-a-ficha-catalografica/>**

**A FICHA CATALOGRÁFICA SERÁ INSERIDA NA VERSÃO FINAL**

**Bruna Mendes Barbosa**

**CONTROLE COMPARTILHADO PARA CADEIRAS DE RODAS ROBÓTICAS  
UTILIZANDO LÓGICA *FUZZY***

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

**BANCA EXAMINADORA**

---

Prof. Dr. Leonardo Rocha Olivi  
Universidade Federal de Juiz de Fora

---

Prof. Dr. Exuperry Barros Costa  
Universidade Federal de Juiz de Fora

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Ana Sophia Cavalcanti Alves Vilas Boas  
Universidade Federal de Juiz de Fora

---

Prof. Dr. Luís Henrique Lopes Lima  
Universidade Federal de Juiz de Fora

Dedico este trabalho aos meus pais Rosa e Josicler, e também ao meu querido orientador do curso Leonardo.



## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por ter me proporcionado saúde e força para seguir com os meus sonhos e enfrentar as diversas nuances e dificuldades ao longo de minha trajetória.

Agradeço aos meus pais, Rosa e Josicler, por terem acreditado em mim e por todo amor e dedicação que sempre tiveram ao longo de toda a minha vida. Nunca conseguirei retribuir todo o apoio incondicional que tiveram por mim e a confiança de que eu conseguiria realizar esse sonho que não foi só meu, mas compartilhado pela minha família.

Aos meus irmãos, Daniel e João Lucas, que estiveram presentes em todos os momentos e me apoiaram diante dos momentos difíceis. Não existem palavras para descrever o conforto e amor que sinto.

Ao meu orientador Leonardo Olivi por toda a paciência e zelo ao longo da minha trajetória acadêmica. Principalmente por ter tornado a robótica algo ainda mais fascinante. Foi sem dúvidas uma fonte de inspiração e incentivo para mim.

Aos meus amigos e professores, em especial Ana Carolina e Fabiane, que tiveram um papel importante em minha vida acadêmica e pessoal. O apoio vindo de amigos é algo de extrema valia.

“A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo.”

Albert Einstein



## RESUMO

Com o advento da tecnologia, atualmente temos na área da medicina diversas técnicas que podem tornar possível a interação entre equipamentos e pessoas com capacidades motoras restritas. Existem exemplos como a eletroencefalografia e a eletromiografia que foram tecnologias de Interface Humano-Máquina (*Human-Machine Interface*, ou HMI) capazes de fazer a leitura de informações dadas por pessoas que perderam sua mobilidade por alguma razão. Um exemplo de aplicação dessas tecnologias é o uso de robôs móveis, que são capazes de se locomover fazendo a leitura dos comandos de um usuário com capacidades restritas através da HMI. No caso em que é utilizado o robô móvel, torna-se necessário introduzir o controle compartilhado (*shared control*) que será o responsável por supervisionar a navegação tornando-a segura para o usuário. As aplicações são geralmente voltadas para a comutação entre usuário e sistema, de forma que em situações específicas cada qual tenha sua parcela de atuação. Nesta tese é proposto um controle compartilhado utilizando lógica *fuzzy* que torna a navegação pelos mais diversos ambientes, uma navegação segura e livre de riscos de colisão.

Palavras-chave: 1. Shared Control. 2. Lógica Fuzzy. 3. Robótica assistiva.

## **ABSTRACT**

With the advent of technology, we currently have in the medical field several techniques that can make possible the interaction between equipment and people with restricted motor skills. There are examples such as electroencephalography and electromyography which were Human-Machine Interface (HMI) technologies capable of reading information given by people who lost their mobility for some reason. An example of the application of these technologies is the use of mobile robots, which are able to get around by reading the commands of a user with restricted capabilities through the HMI. In the case where the mobile robot is used, it is necessary to introduce the shared control that will be responsible for supervising the navigation making it safe for the user. The applications are generally geared towards switching between the user and the system, so that in specific situations each one has its share of performance. In this thesis is proposed a shared control using fuzzy logic that makes the navigation through the most diverse environments, a safe navigation and free of collision risks.

Keywords: 1. Shared Control. 2. Fuzzy Logic. 3. Assistive Robotics.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura do sistema robótico de eletroencefalografia .....	16
Figura 2 - Função de pertinência clássica .....	17
Figura 3 - Função de pertinência para avaliação de estatura .....	21
Figura 4 - Função de pertinência triangular .....	22
Figura 5 - Função de pertinência de A .....	22
Figura 6 - Função de pertinência de B .....	24
Figura 7 - Intersecção dos conjuntos A e B .....	24
Figura 8 - União dos conjuntos A e B .....	25
Figura 9 - Negação de A .....	25
Figura 10 - Exemplo de aplicação do método de Mamdani .....	26
Figura 11 - Ilustração das entradas do controlador <i>fuzzy</i> .....	29
Figura 12 - Esquema representativo das entradas e saídas do controlador <i>fuzzy</i> .....	30
Figura 13 - Esquema representativo <i>no MATLAB</i> .....	31
Figura 14 - Entrada “rho” do controlador <i>fuzzy</i> .....	31
Figura 15 - Regiões definidas para “rho” no controlador <i>fuzzy</i> .....	32
Figura 16 - Exemplo de obstáculo entre as regiões “Médio” e “Longe” .....	33
Figura 17 - Regiões definidas para “alpha” no controlador <i>fuzzy</i> .....	34
Figura 18 - Exemplo de obstáculo na região a direita do robô .....	35
Figura 19 - Perfil de velocidades de saída do controlador <i>fuzzy</i> .....	35
Figura 20 - Exemplos ilustrativos das condições para “Muito Perto” .....	36
Figura 21 - Exemplos ilustrativos das condições para “Perto” .....	37
Figura 22 - Exemplos ilustrativos das condições para “Médio” .....	37
Figura 23 - Exemplos ilustrativos das condições para “Longe” .....	39
Figura 24 - Exemplos para “Perto” de rho com todas as opções disponíveis em alpha ...	40
Figura 25 - Exemplos para “Perto” de rho com todas as opções disponíveis em alpha ...	42
Figura 26 - Exemplos para “Perto” de rho com todas as opções disponíveis em alpha ...	44
Figura 27 - Percurso 01 – Navegação sem obstáculos .....	48
Figura 28 - Velocidades linear e angular para o percurso 01 .....	49
Figura 29 - Saídas do controlador <i>fuzzy</i> para o percurso 01 .....	50
Figura 30 - Percurso 02 – Navegação com uma parede a direita .....	51
Figura 31 - Velocidades linear e angular para o percurso 02 .....	51
Figura 32 - Saídas do controlador <i>fuzzy</i> para o percurso 02 .....	52

Figura 33 - Percurso 03 – Navegação com uma parede a esquerda .....	52
Figura 34 - Velocidades linear e angular para o percurso 03 .....	52
Figura 35 - Saídas do controlador fuzzy para o percurso 03 .....	53
Figura 36 - Percurso 04 – Navegação por um corredor .....	53
Figura 37 - Velocidades linear e angular para o percurso 04 .....	54
Figura 38 - Saídas do controlador fuzzy para o percurso 04 .....	55
Figura 39 - Percurso 05 – Navegação por uma porta .....	55
Figura 40 - Velocidades linear e angular para o percurso 05 .....	56
Figura 41 - Saídas do controlador fuzzy para o percurso 05 .....	56
Figura 42 - Percurso 06 – Navegação para virar em direção a parede .....	57
Figura 43 - Velocidades linear e angular para o percurso 06 .....	57
Figura 44 - Saídas do controlador fuzzy para o percurso 06 .....	58
Figura 45 - Percurso 07 – Navegação para virar em direção contrária a parede .....	58
Figura 46 - Velocidades linear e angular para o percurso 07 .....	59
Figura 47 - Saídas do controlador fuzzy para o percurso 07 .....	59
Figura 48 - Percurso 08 – Navegação em direção a parede .....	60
Figura 49 - Velocidades linear e angular para o percurso 08 .....	60
Figura 50 - Saídas do controlador fuzzy para o percurso 08 .....	61
Figura 51 - Percurso 09 – Navegação próximo a parede .....	61
Figura 52 - Velocidades linear e angular para o percurso 09 .....	62
Figura 53 - Saídas do controlador fuzzy para o percurso 09 .....	62
Figura 54 - Percurso 10 – Navegação ambiente com obstáculos .....	63
Figura 55 - Velocidades linear e angular para o percurso 10 .....	64
Figura 56 - Saídas do controlador fuzzy para o percurso 10 .....	64

## SUMÁRIO

1	<b>INTRODUÇÃO</b> .....	<b>14</b>
2	<b>FUNDAMENTAÇÃO TÉCNICA</b> .....	<b>16</b>
2.1	COMUNICAÇÃO HUMANO-MÁQUINA .....	16
2.2	CONTROLE COMPARTILHADO .....	17
2.3	LÓGICA <i>FUZZY</i> .....	20
2.3.1	<b>Os conjuntos <i>fuzzy</i></b> .....	<b>20</b>
2.3.2	<b>Definições e operações</b> .....	<b>23</b>
2.3.3	<b>Base de Regras e Método de Inferência</b> .....	<b>27</b>
3	<b>DESENVOLVIMENTO TÉCNICO</b> .....	<b>30</b>
3.1	ENTRADAS E SAÍDAS DO SISTEMA FUZZY .....	30
3.2	CONTROLADOR FUZZY: VARIÁVEIS DE ENTRADAS E SAÍDAS .....	32
3.2.1	<b>Entrada rho</b> .....	<b>32</b>
3.2.2	<b>Entrada alpha</b> .....	<b>34</b>
3.2.3	<b>Velocidades de saída</b> .....	<b>36</b>
3.3	BASE DE REGRAS .....	38
4	<b>RESULTADOS</b> .....	<b>48</b>
4.1	SITUAÇÕES ISOLADAS .....	48
4.2	SITUAÇÕES DESAFIADORAS .....	53
4.3	SITUAÇÕES DE RISCO .....	57
4.4	ILUSTRAÇÃO DE SITUAÇÃO REAL .....	63
5	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> .....	<b>65</b>
	<b>REFERÊNCIAS</b> .....	<b>66</b>

## 1 INTRODUÇÃO

Com o passar das descobertas e pesquisas, a robótica torna-se cada vez mais presente em nossa realidade. Quase 254 mil robôs foram comprados pela indústria de todo o mundo apenas em 2015, segundo a Federação Internacional de Robótica (IFR, *International Federation of Robotics*, em inglês). De quatro a seis cirurgias robotizadas são feitas por semana em um hospital do Rio de Janeiro, que atende pelo Sistema Único de Saúde (SUS). As aplicações desse tipo de tecnologia só crescem e criam novas possibilidades em áreas tradicionais, como a educação [1].

Ainda conforme [1], temos atualmente cirurgias feitas roboticamente no Instituto Nacional do Câncer (INCa), o primeiro hospital do Brasil a oferecer esse tipo de tratamento para pacientes do SUS. A principal vantagem da cirurgia robótica é sua característica menos invasiva e mais rápida, que aproveita cavidades do próprio paciente evitando cortes. Além disso, tem-se o tempo reduzido de operação quando comparado a uma cirurgia tradicional. Enquanto o tempo de operação anteriormente era estimado em cerca de 6 horas, com o uso dos robôs, é possível cair para até 30 minutos.

Já na indústria, os robôs têm sido utilizados há mais tempo e a tendência é que sua utilização cresça ainda mais nos próximos anos. Isto porque o setor deve adquirir 400 mil novas máquinas, conforme estimado pela Federação Internacional de Robótica. Atualmente, a indústria automotiva é a maior consumidora de robôs do mundo todo, segundo [1].

Dentre os temas estudados na Robótica, temos a robótica assistiva, que consiste no estudo e aplicação de conceitos que irão auxiliar o usuário na execução de atividades que lhe foram restritas por alguma razão, seja ela por traumas ou outros incidentes. Dentre as diversas tecnologias desenvolvidas na área médica e utilizadas na robótica assistiva, temos a eletromiografia e a eletroencefalografia, que são tecnologias desenvolvidas em interfaces com usuários que em geral tem suas habilidades motoras restritas e necessitam de auxílio para executar determinadas atividades [2],[3].

O cérebro é um sistema de alta complexidade, tendo por exemplo o córtex frontal como a região onde a maioria dos pensamentos e decisões conscientes é feita. Porém, esta região conduz muito menos do que um décimo da atividade total do cérebro, segundo [3].

A eletroencefalografia é uma tecnologia útil para extrair sinais biológicos de pacientes cujas restrições de movimentos são muito severas, como os portadores de tetraplegia. Uma característica desta tecnologia é a possibilidade de seu uso de maneira não-invasiva. Entretanto, em razão do uso não-invasivo perde-se precisão na leitura das informações do usuário. Dessa

maneira, é importante que se atue em medidas mitigadoras quanto aos erros associados às leituras dos sinais biológicos dos usuários [3].

De acordo com [2] e [3], para mapear a verdadeira atividade do cérebro, é muito importante medir os sinais de diversas estruturas corticais diferentes, localizadas ao redor da superfície do cérebro, não sendo possível mapear esses sinais puramente das regiões frontal e temporal. Sendo assim, um dos grandes desafios nestes casos é o tratamento que será dado aos erros de leitura no equipamento, uma vez que essas interfaces apresentam erros na identificação da ação. O principal objetivo é minimizar os equívocos que podem ocorrer devido a um erro na leitura dos sinais biológicos, visto que o objetivo é proporcionar conforto e segurança ao usuário.

Neste trabalho será proposto um controle compartilhado utilizando a lógica *fuzzy* [4] para minimizar as possíveis situações de risco e tornar a mobilidade segura e confiável. No contexto da robótica assistiva, o usuário irá utilizar a cadeira de rodas robótica com o objetivo de devolver sua mobilidade para que haja uma redução na necessidade de assistência humana na vida do mesmo.



## 2 FUNDAMENTAÇÃO TÉCNICA

Este capítulo se destina a descrever as principais técnicas utilizadas neste trabalho, como a comunicação entre os usuários e a máquina, o controle compartilhado e a lógica *fuzzy*, principais temas explorados neste trabalho.

### 2.1 COMUNICAÇÃO HUMANO-MÁQUINA

A principal comunicação de um indivíduo com deficiência que impede a manipulação de interfaces de interação convencionais é por meio de seus sinais biológicos. Vários tipos de células possuem a capacidade de gerar uma descarga elétrica, fenômeno denominado por potencial de ação. Em especial, o sistema nervoso produz sinais elétricos, por meio de complexas atividades eletroquímicas celulares que percorrem todo o corpo transportando informação. A função desses sinais é, de forma abrangente, controlar as funções do corpo humano, tais como as movimentações voluntárias e involuntárias de órgãos e conjuntos musculares [3].

Para a comunicação dos usuários portadores de imparidades físicas com as máquinas, é possível utilizar as diversas maneiras de aquisição dos sinais biológicos, tais como a eletromiografia (EMG) e a eletroencefalografia (EEG), o rastreamento dos olhos (do inglês, *eye tracking*), dentre outros. Estes podem ser utilizados para a extração das intenções dos usuários de modo a construir uma via de comunicação com a máquina. Neste trabalho, utiliza-se a extração de sinais tipo eletromiografia por meio do EMOTIV [2], conforme a Figura 1.

Figura 1. Equipamento EMOTIV EPOC+.



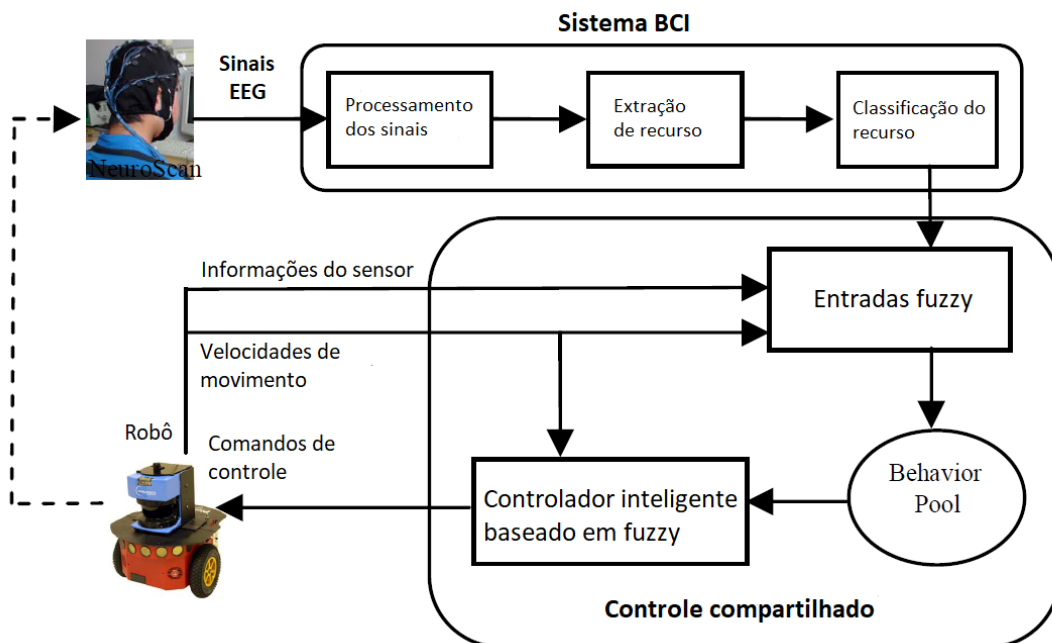
Fonte: Emotiv [2].

## 2.2 CONTROLE COMPARTILHADO

O conceito de controle compartilhado está diretamente relacionado a divisão da responsabilidade entre o usuário e uma lei de controle na atuação e direcionamento de um sistema robótico [5]. Uma lei de controle assistiva é desenvolvida para ajudar o usuário a completar uma tarefa de controle de um robô de modo a maximizar a segurança do usuário enquanto a tarefa é realizada [6].

Conforme mostrado na Figura 2, o usuário dispõe de uma interface humano-máquina assistiva capaz de extrair os comandos adquiridos diretamente do usuário, e os envia ao robô móvel. Portanto, além das leituras do robô e das informações do ambiente, ainda é possível se basear na percepção e direcionamento do usuário. Dessa forma, o principal objetivo do controle compartilhado seria propiciar proteção ao usuário durante a navegação, além de executar correções e minimizar erros da HMI assistiva [7].

Figura 2. Arquitetura do sistema robótico de eletroencefalografia.



Fonte: Liu et. al. [7].

Como forma de ilustrar a utilização do controle compartilhado, tem-se o artigo de Olivi et al. [6], que desenvolve um controle compartilhado baseado em campos vetoriais aplicado para cadeiras de rodas robóticas. No trabalho, os autores descrevem as diversas metodologias de controle compartilhado, dentre as quais a maioria utiliza situações padronizadas para a

solução de situações. No entanto, essas situações padronizadas não atendem bem a diversas situações, e os autores apresentam, então, uma metodologia capaz de se adaptar ao ambiente em que se encontra o usuário e o robô.

Em [6], os autores apresentam diversas situações problemáticas em que o robô é capaz de se adaptar à disposição de obstáculos do ambiente e calcular as melhores velocidades para o robô assistivo, de maneira a auxiliar o usuário a concluir sua navegação pelos diversos cômodos maximizando a segurança do robô, evitando colisões e vencendo os obstáculos mais comuns presentes na rotina de um ambiente.

Outro artigo onde pode-se observar o uso de controle compartilhado, é o artigo citado em [7]. Neste artigo o autor demonstrou o uso de um controle compartilhado entre a coleta de sinais através do EEG e as informações lidas pelos sensores do robô utilizado no experimento. De acordo com [7], os principais problemas da interface entre o computador e o cérebro estão associados ao baixo número de canais e ao alto erro na leitura dos sinais.

Conforme descrito em [7]\*, o autor utiliza um método de controle de movimento assistivo para que os impulsos que comandam o robô sejam confiáveis e em tempo real. Portanto, para que o robô responda com maior eficácia ao ambiente externo, um controle compartilhado baseado em lógica *fuzzy* foi utilizado.

O módulo de controle compartilhado utilizado em [7] tem as entradas *fuzzy*, a chamada pool de comportamento e um controlador inteligente baseado em *fuzzy* como diretrizes para a tomada de decisão. O controle compartilhado traduz então as entradas para gerar comandos para os motores do robô. O repositório de comportamentos terá um deles ativo, uma vez que o controlador compartilhado escolhe o comportamento adequado por meio da combinação dos resultados de reconhecimento e comandos do operador. O experimento mostrou resultados positivos quando comparado ao controle direto (apenas o humano no comando). Foi possível observar resultados relevantes, como o decréscimo do número de colisões e principalmente no tempo de operação do percurso.

O trabalho apresentado por Nishimura et al. [8] descreve um sistema de controle compartilhado utilizado para dirigir um veículo automotivo. O principal objetivo do artigo foi encontrar uma interação suave entre o objetivo do motorista e o objetivo do sistema automatizado do veículo.

Em [8], foi estudada a possibilidade de apoiar o condutor para que acompanhasse a faixa ao mesmo tempo que possibilitasse que o mesmo mudasse de faixa caso desejado. A proposta foi voltada para integrar essas condições contraditórias de forma que o controle compartilhado estivesse trabalhando e ao mesmo tempo considerando a intenção do condutor.

O método proposto em [8] julgou o status cooperativo utilizando o torque de dois agentes e ainda a velocidade lateral do veículo como contribuição de cada agente. O método permitiu aos condutores mudar de faixa sem problemas, porém, no caso de uma inconsistência intencional, foram propostas duas estratégias. No primeiro caso, quando o motorista mantém a intenção de mudar de pista por um tempo especificado, o sistema altera a faixa de destino e conecta suavemente as duas pistas de destino sem diminuir o ganho do sistema de controle. Já no segundo caso, o ganho é ajustado para que o sistema de assistência seja diminuído, ou seja, a vontade do condutor tem maior relevância.

Na avaliação do método de [8], utilizou-se um simulador para que os estudos fossem colocados em prática e os resultados demonstraram a eficácia do método proposto, facilitando as mudanças de faixa, suavizando o movimento e melhorando significativamente o desempenho dentro do trecho. Portanto, nesse artigo foi possível ilustrar as vantagens do uso de controle compartilhado como uma forma de trazer conforto e segurança ao usuário.

Um outro artigo que ilustra o uso do controle compartilhado é o artigo publicado por Soh e Demiris [9]. O trabalho apresenta uma estrutura probabilística modelo e um algoritmo capaz de aprender observando o assistente. A metodologia aplicada, assim como em [8], se refere a Assistência de Aprendizagem por Demonstração (LAD), ou seja, existe uma interação com um operador para que ocorra o treinamento do sistema. A aplicação é voltada para usuários de cadeira de rodas e é demonstrado que o controle compartilhado melhorou o desempenho de condução com uma aceleração de 191% a mais do que na ausência do método.

O método de [9] foi implementado em experimentos práticos, e o cenário considerado foi voltado para o usuário que não possuía habilidades para virar à direita, portanto, sendo necessária a assistência. Afim de incentivar o aprendizado do programa, nenhuma assistência foi dada durante curvas suaves para a direita (casos em que o ângulo de virada era menor do que  $60^\circ$ ), movimentos para a esquerda, para frente ou para trás.

O objetivo de [9] foi permitir uma análise da precisão e facilidade da assistência oferecida, considerando também se foi feita na hora correta. Esta simples representação implementada foi suficiente para o experimento, sendo o modelo capaz de comportar decisões para estados complexos, como foi citado no artigo. Além disso, foram mostrados resultados satisfatórios por parte dos participantes do experimento, mostrando, portanto, que o controle compartilhado de fato alcança o seu objetivo de simplificação de determinadas atividades.

## 2.3 LÓGICA FUZZY

Esta seção aborda a lógica *fuzzy*, desde a teoria até os aspectos práticos aplicados neste trabalho. A lógica *fuzzy*, traduzida como lógica nebulosa, é um tipo de lógica voltada para o suporte aos modos de raciocínio que são aproximados e não exatos e, ainda são técnicas utilizadas no tratamento de informações qualitativas de forma rigorosa. A seguir apresenta-se os principais conceitos ligados à lógica *fuzzy* [4],[10].

### 2.3.1 Os conjuntos *fuzzy*

A noção de um conjunto *fuzzy* foi introduzida inicialmente por Lotfi Zadeh em 1965, que mais tarde desenvolveu muitos dos métodos da lógica *fuzzy* baseados nessa simples noção. Passaram-se algumas décadas para a lógica *fuzzy* ser aplicada em sistemas reais por outros cientistas [11].

Os conjuntos *fuzzy* contrastam com os conjuntos tradicionais da lógica Aristotélica (lógica tradicional) pois permitem graus de pertinência diferentes de 1 ou 0, 100% ou 0%, ou seja, pertence ou não pertence. Os conjuntos da lógica tradicional são sistemas lógicos binários, ou seja, só será possível assumir dois valores distintos, são eles:

$$\begin{cases} \mu_A(u) = 1, \text{ se } u \in A \\ \mu_A(u) = 0, \text{ se } u \notin A \end{cases} \quad (1)$$

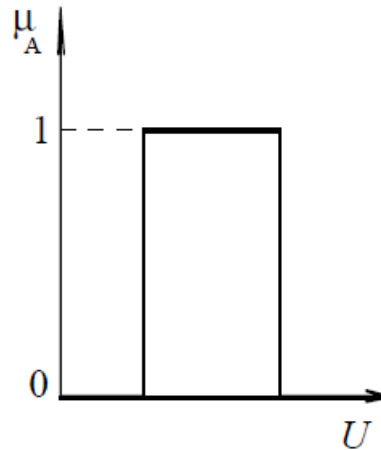
Em que  $U$  é o universo de discurso dos elementos  $u$  permitidos para o conjunto  $A$ , e  $\mu$  é o grau de pertinência do elemento  $u$  ao conjunto  $A$ . Pode-se perceber, conforme a Equação 1 e a Figura 3, que em se tratando da pertinência de elementos  $u$  a um conjunto lógico tradicional, estes elementos possuirão apenas o grau de pertinência  $\mu$  igual 100%, ou seja,  $\mu_A(u) = 1$ . É vedado outro grau de pertinência para elementos que pertencem ao conjunto. Em complementação, todos os elementos  $u$  que não pertencem ao conjunto  $A$  possuem o grau de pertinência de 0%, ou seja,  $\mu_A(u) = 0$ .

Os conjuntos *fuzzy* possibilitam graus de pertinência diferentes de 1 ou 0, ou seja, existe a possibilidade de um elemento pertencer parcialmente a um conjunto, sendo assim, torna-se necessário definir uma função de pertinência, dada por meio da Equação 2:

$$\mu_A(u): U \rightarrow [0,1] \quad (2)$$

Tem-se, então, o universo de discurso  $U$  no qual  $A$  é um subconjunto pertencente a  $U$  e  $\mu$  é uma função que varia entre  $[0,1]$  que define o grau de pertinência dos elementos  $u \in A$ .

Figura 3. Função de pertinência clássica.



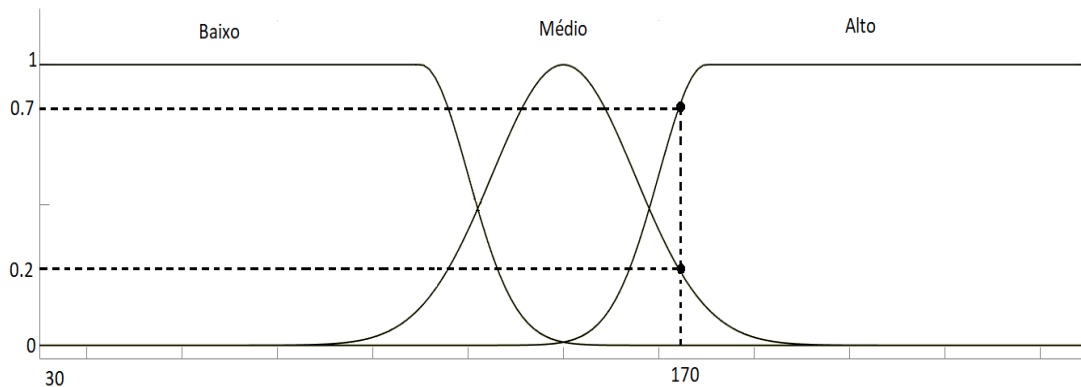
Esta função de pertinência  $\mu$  definirá dentro do intervalo de variação  $[0,1]$  o nível de pertinência dos elementos  $u$  para o universo  $U$ . A função de pertinência  $\mu$  definirá, então, um conjunto *fuzzy*, que permite a elementos  $u \in U$  um grau de pertinência que varia entre 0 e 1. O formato que o conjunto *fuzzy* assumirá depende de como  $\mu$  é definida, em que as formas mais usuais são funções Gaussianas, trapezoidais e triangulares.

Isto significa que esta lógica *fuzzy* expressa, por meio do formato da função de pertinência  $\mu$ , diferentes maneiras de pertencimento de elementos a um conjunto. Como o formato de algumas funções de pertinência se enquadram na maneira linguística dos humanos expressarem quantidades por meio de determinadas palavras que carregam sentido quantitativo, como “muito”, “pouco”, “mais ou menos”, “extremamente”, dentre outras, estas funções foram denominadas por variáveis linguísticas.

Para exemplificar, é possível utilizar o exemplo de variáveis linguísticas “baixa”, “média” e “alta” no contexto aplicado a estatura de seres humanos citado em [11]. Conforme mostrado na Figura 3, tem-se que a altura de 1,7 0m tem um grau de pertinência de 0,2 para média estatura (conjunto *fuzzy* que designa a média estatura) e um grau de pertinência de 0,7 para alta estatura (conjunto *fuzzy* que designa a alta estatura).

No contexto apresentado, estatura de seres humanos, os conjuntos designados para as estaturas variam seus graus de pertinência entre os valores  $[0,1]$ , mostrando que valores intermediários de pertencimento são válidos, denotando a noção humana de pouco ou muito pertencimento a um determinado conjunto. Em outras palavras, é facultado a um elemento o pertencimento a um grupo com um grau de pertinência acima de 0% e abaixo de 100%, se encaixando na maneira humana de expressar quantidades por meio da linguagem.

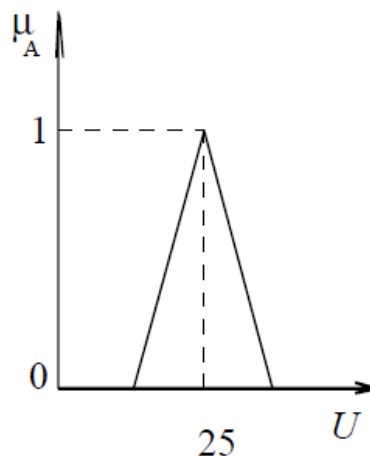
Figura 4. Função de pertinência para avaliação de estatura.



Fonte: autor.

Outro exemplo, segundo [12], seria a ideia de uma temperatura que tem seu valor por volta de  $25^\circ$ , neste caso podemos utilizar uma função de pertinência triangular conforme Figura 5. Esta, por sua vez, representará que, quanto mais próximo do valor de  $25^\circ$ , mais ela se identifica com o conceito representado.

Figura 5. Função de pertinência triangular.



Fonte: [12]

Considerando o universo discreto, pode-se representar a função como sendo:

$$A = \mu(u_1)/u_1 + \mu(u_2)/u_2 + \dots + \mu(u_n)/u_n = \sum \mu_i/u_i \quad (3)$$

Neste caso estamos separando o grau de pertinência de cada membro anterior a barra enquanto que após a barra trata-se do elemento em si. Podemos perceber conforme [12] que na lógica *fuzzy* o raciocínio exato corresponde apenas a um caso limite do raciocínio aproximado, sendo um processo de composição de relações nebulosas.

### 2.3.2 Definições e Operações

Conforme citado na seção anterior, temos definido a função de pertinência, funções estas que podem assumir diferentes formatos que serão definidos de acordo com a aplicação da mesma. Abaixo estão apresentados alguns dos exemplos mais comumente utilizados. São eles:

- **Função Triangular:** a função de pertinência triangular pode modelar matematicamente o conceito de “em torno de”. É definida da seguinte forma:

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \geq a \\ \frac{x-a}{u-a}, & \text{se } a < x \leq u \\ \frac{x-b}{u-b}, & \text{se } u \leq x < b \\ 0, & \text{se } x \geq b \end{cases} \quad (4)$$

- **Função Trapezoidal:** a função de pertinência trapezoidal é definida da seguinte forma:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & \text{se } a \leq x < b \\ 1, & \text{se } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{se } c < x \leq d \\ 0, & \text{caso contrário} \end{cases} \quad (5)$$

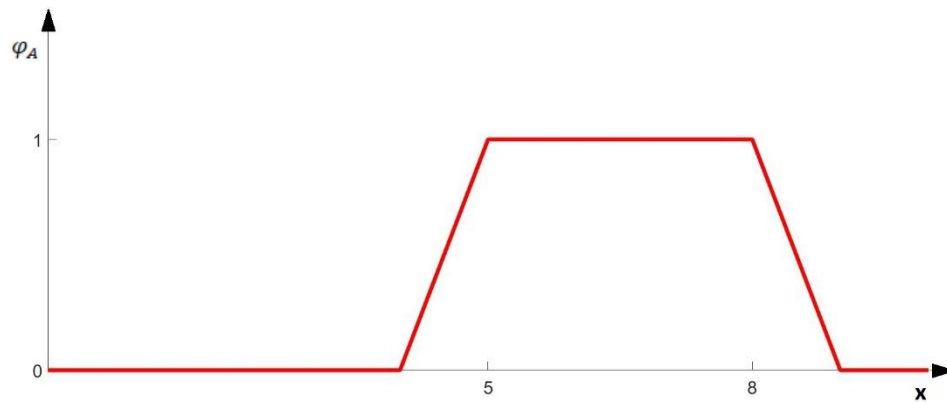
- **Função Sino:** a função de pertinência sino é utilizada quando sua aplicação for suave em relação a um número real. Esta função é definida como:



$$\mu_A(x) = \begin{cases} \exp\left(\frac{-(x-u)^2}{a}\right), & \text{se } u - \delta \leq x \leq u + \delta, \\ 0, & \text{caso contrário} \end{cases} \quad (6)$$

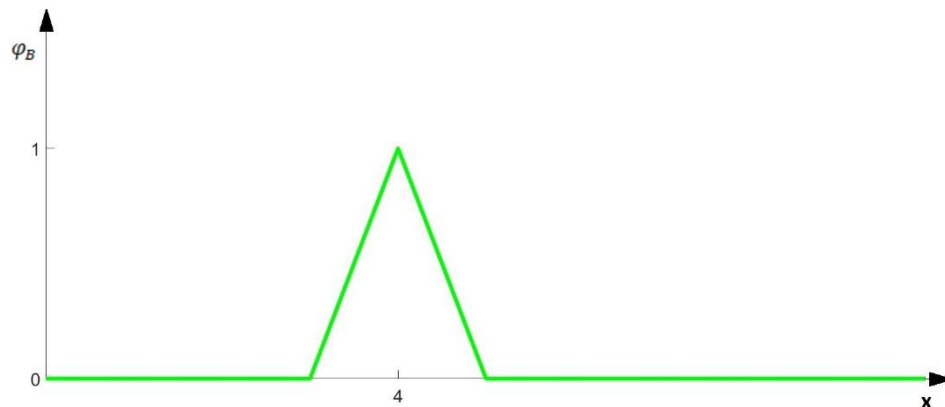
Conforme citado em [13], as operações sobre os conjuntos *fuzzy* são as mesmas utilizadas na teoria dos conjuntos clássicos. Considerando dois conjuntos *fuzzy* A e B, representados por funções de pertinência conforme as Figuras 6 e 7, podem-se definir operações de união, intersecção, negação, etc.

Figura 6. Função de pertinência de A.



Fonte: Autor

Figura 7. Função de pertinência de B.

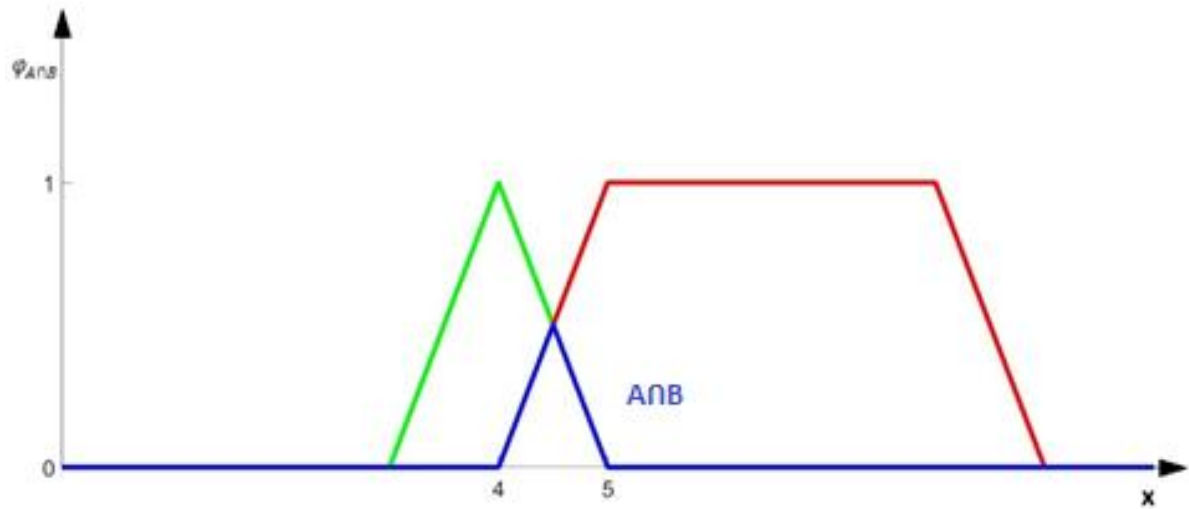


Fonte: Autor

A operação de intersecção de dois conjuntos *fuzzy* A e B é representada por  $A \cap B$  e representa o conceito do conector lógico “AND” ou “E”. Conforme mostrado na Figura 8, a intersecção é dada por:

$$\mu_{A \cap B}(x) = \min(A, B) = A \cap B \quad (7)$$

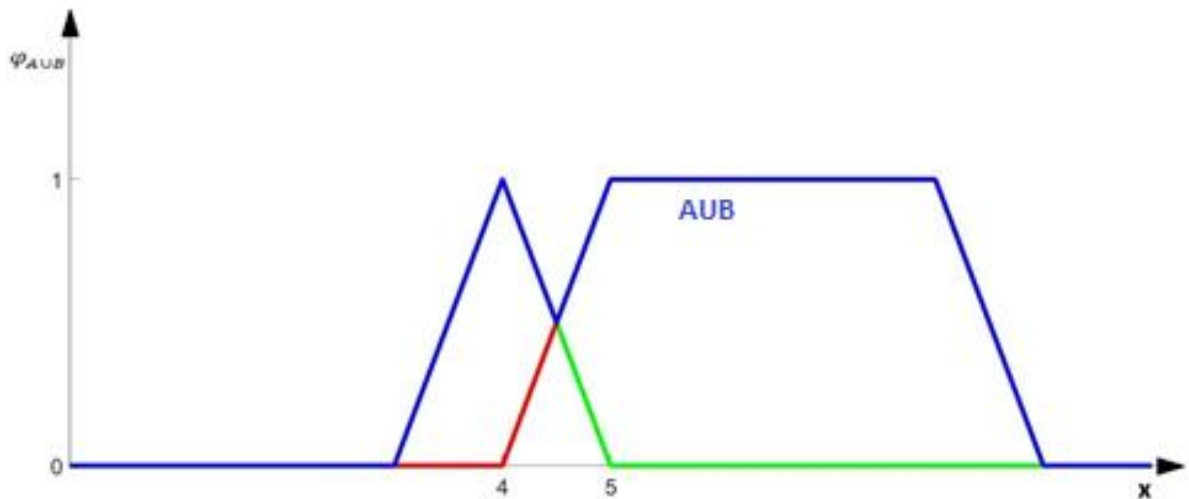
Figura 8. Intersecção dos conjuntos A e B.



A operação de união de dois conjuntos *fuzzy* A e B é representado por  $A \cup B$  e representa o conceito do conector lógico “OR” ou “OU”. Conforme mostrado na Figura 9, a união é:

$$\mu_{A \cup B}(x) = \max(\mu_A, \mu_B) = A \cup B \quad (8)$$

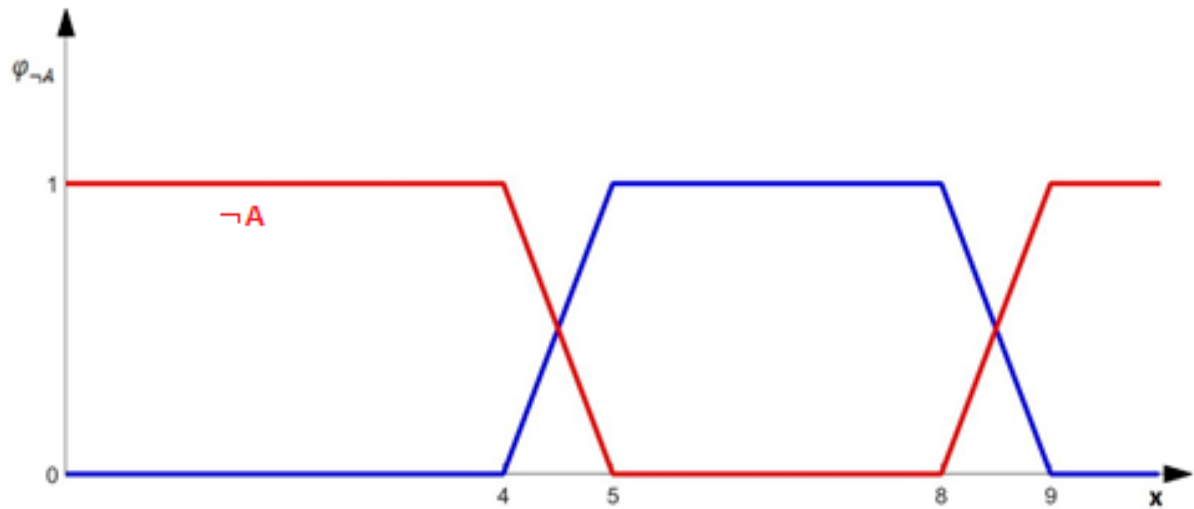
Figura 9. União dos conjuntos A e B.



Por último, nesta descrição, tem-se a operação de negação de um conjunto *fuzzy* também chamado de complemento ou simplesmente o operador lógico “NOT” ou “NÃO”. Conforme a Figura 10, tem-se que a negação é dada por:

$$\bar{\mu}_A(x) = \neg \mu_A(x) = 1 - \mu_A(x) \quad (9)$$

Figura 10. Negação de A.



Existem ainda, segundo [11], diversas outras operações, como:

- Concentração  $CON(A)$  é uma operação que representa o conceito linguístico de “muito” e sua função está mostrada a seguir.

$$\mu_{CON(A)}(x) = (\mu_A(x))^2 \quad (10)$$

- Subconjunto  $A \subseteq B$ :

$$\mu_A(x) \leq \mu_B(x) \quad (11)$$

- Dilatação  $DIL(A)$  é uma operação que representa o conceito linguístico da expressão “mais ou menos” e sua função está mostrada a seguir.

$$\mu_{DIL(A)}(x) = (\mu_A(x))^{0.5} \quad (12)$$

- Produto algébrico  $A \cdot B$ :

$$\mu_{AB}(x) = \mu_A(x) \cdot \mu_B(x) \quad (13)$$

- Soma limitada:

$$\max\{1, \mu_A + \mu_B\} \quad (14)$$

- Diferença limitada A /-/ B:

$$\mu_{A/-/B} = (x) = \min\{0, \mu_A - \mu_B\} \quad (15)$$

- Produto limitado:

$$\max\{0, \mu_A + \mu_B - 1\} \quad (16)$$

- Normalização NORM(A):

$$\mu_{NORM(A)}(x) = \mu_A(x) / MAX\{\mu_A(x)\} \quad (17)$$

- Soma algébrica:

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) \quad (18)$$

As operações de associação, comutação e distribuição são válidas também para a lógica *fuzzy*:

- Associação:  $(a * b) * c = a * (b * c)$
- Comutação:  $a * b = b * a$
- Distribuição:  $a * (b \circ c) = (a \circ b) * (a \circ c)$

Os operadores \* e  $\circ$  podem representar qualquer operação descrita acima.

### 2.3.3 Base de Regras e Métodos de Inferência

Uma vez que se tem definidas as variáveis de entrada e de saída de um determinado sistema que se deseja modelar ou controlar, é necessário estabelecer uma base de regras que irão executar a tarefa de modelagem ou controle designadas para o sistema. Essa base de regras utilizará as diversas operações descritas na seção anterior para relacionar os diferentes grupos

*fuzzy* que compõem as variáveis de entrada e saída do sistema [4], [10], [11]; [12]. O principal objetivo da base de regras é organizar o conhecimento contido dentro da descrição dos conjuntos definidos pelas funções de pertinência, conforme mostrado em [13].

Um sistema *fuzzy* consiste em três partes: (1) valores das variáveis de entrada e saída *fuzzy*; (2) regras *fuzzy*; (3) métodos de inferência *fuzzy*, que podem incluir fuzzificação e desfuzzificação [11].

Em [15], é feita uma classificação dos sistemas de inferência *fuzzy*:

- Raciocínio *fuzzy* tipo 1 (Mamdani)
- Raciocínio *fuzzy* tipo 2 (Larsen)
- Raciocínio *fuzzy* tipo 3 (Tsukamoto)
- Raciocínio *fuzzy* tipo 4 (Takagi e Sugeno).

Dentre os 4 tipos citados, temos, então, o Mamdani, que define sua base de regras como sendo do seguinte tipo:

$$SE\ x\ \acute{e}\ A,\ ENT\tilde{A}O\ y\ \acute{e}\ B$$

Estas são as chamadas proposições *fuzzy* onde  $x$  e  $y$  são variáveis definidas nos universos de discurso  $U$  e  $V$ , respectivamente de entrada e saída. Já  $A$  e  $B$  são definidos como conjuntos *fuzzy* com função de pertinência definida como  $\mu_A: U \rightarrow [0,1]$  e  $\mu_B: V \rightarrow [0,1]$ . O sistema criado por Mamdani [14] é do tipo gráfico e pode ser exemplificado por meio da Figura 11.

Para uma melhor compreensão do exemplo dado na Figura 11, considera-se então que:

$$Regra\ 1: Se\ x = A_1\ e\ y = B_1\ \acute{e}nt\tilde{a}o\ z = C_1$$

$$Regra\ 2: Se\ x = A_2\ e\ y = B_2\ \acute{e}nt\tilde{a}o\ z = C_2$$

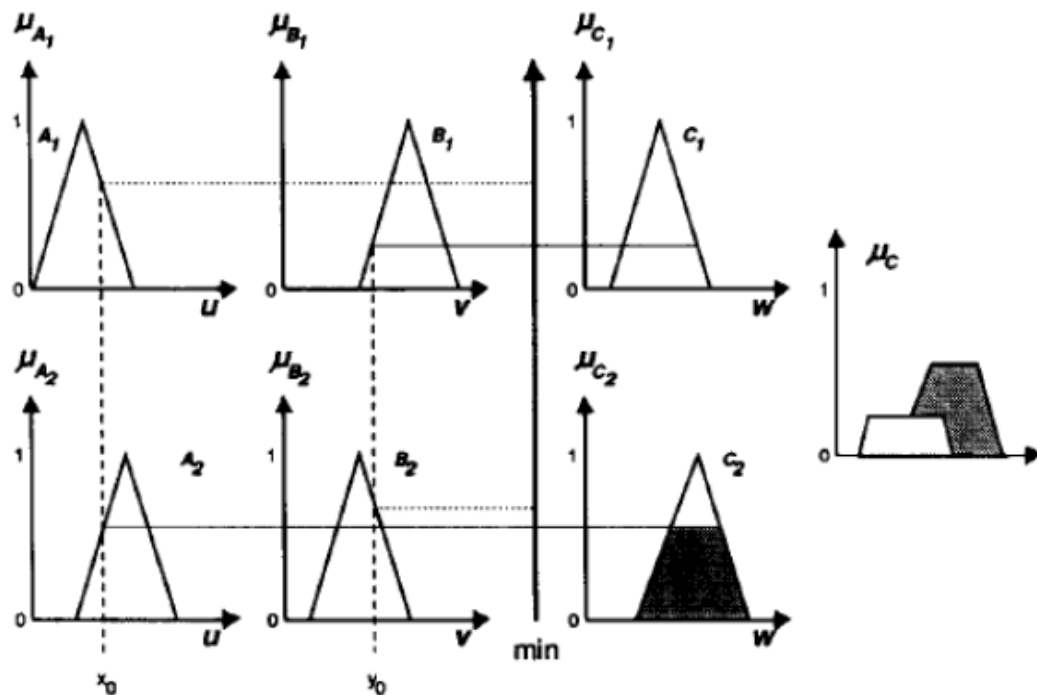
Sendo que  $x$  e  $y$  são variáveis de entrada e  $z$  é a variável de saída,  $A_i$  e  $B_j$  são grupos *fuzzy* que vigoram nas variáveis de entrada  $x$  e  $y$ , respectivamente, e  $C_k$  são os grupos *fuzzy* da variável de saída  $z$ .

Tem-se  $x_0$  e  $y_0$  como os valores de entrada que foram fuzzyficados e com base em seu domínio é possível encontrar sua imagem, no caso  $\mu_{A_1}$  e  $\mu_{B_1}$  e  $\mu_{A_2}$  e  $\mu_{B_2}$ . Uma vez que o conector lógico usado nas duas regras é o “E” ou “AND”, aplica-se o mínimo entre as duas regras. Portanto, o gráfico no termo  $C_1$  será o menor valor dentre as pertinências  $\mu_{A_1}$  e  $\mu_{B_1}$ , conforme é descrito em [13]. Uma descrição matemática do processo descrito é mostrada na sequência:

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

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

Figura 11. Exemplo de aplicação do método de Mamdani.



Fonte: [11]

O último passo deste método é a união dos termos iguais, ou seja, a atribuição será para o maior valor entre  $\mu_{C_1}$  e  $\mu_{C_2}$ .

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

Depois de efetuada a inferência no sistema, o mesmo gera um número *fuzzy*. É necessário converter esse número *fuzzy* em um número real, esse processo é chamado de defuzzyficação [15]. O processo de defuzzyficação é a transformação de um número *fuzzy* em um número real. Existem diversos métodos, tais como: centro de massa, também conhecido como centroide, o centro da soma, média dos máximos, entre outros [15].

## 2.4 CONTRIBUIÇÕES

Este trabalho destina-se a utilizar a lógica *fuzzy* com o modelo de inferência tipo Mamdani para elaborar um sistema de controle compartilhado com o usuário de uma cadeira de rodas robótica cujos sinais biológicos são utilizados para a extração das tomadas de decisão. O sistema compartilhado *fuzzy* auxiliará o usuário a navegar o robô, maximizando sua segurança e permitindo que todas as áreas sejam acessíveis ao robô.

### 3 DESENVOLVIMENTO TÉCNICO

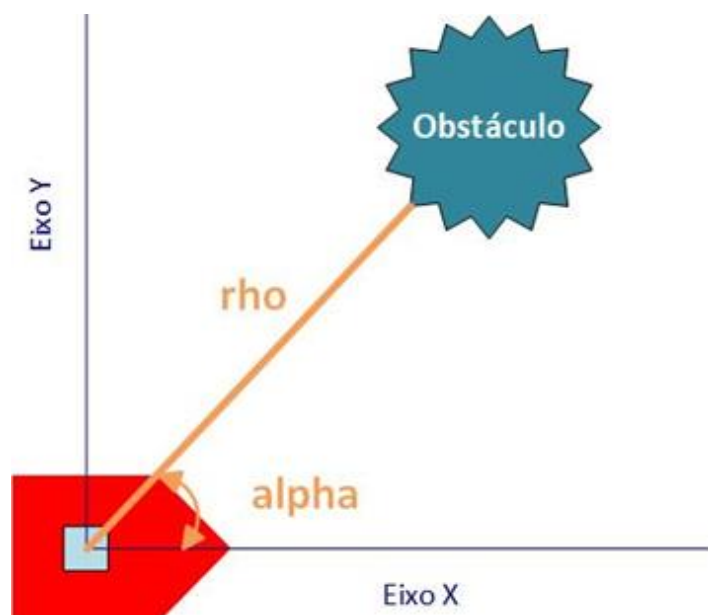
Este capítulo irá descrever o desenvolvimento prático do trabalho, como o sistema *fuzzy* para o controle compartilhado, as variáveis de entrada e saída e ainda a base de regras utilizada nas simulações.

#### 3.1 ENTRADAS E SAÍDAS DO SISTEMA FUZZY

Durante os experimentos, são disponibilizados ao usuário quatro comandos do robô, sendo eles a possibilidade de navegar para frente, para trás (ré), de virar à esquerda e à direita. Esses comandos são escolhidos pelo próprio usuário de acordo com suas necessidades e escolhas, e são extraídos por meio de sinais biológicos do tipo EMG, por meio do EMOTIV. No entanto, esses comandos podem ser extraídos de maneira incorreta, uma vez que diversos ruídos permeiam a captura e processamento dos sinais biológicos, e seus efeitos devem ser mitigados pelo sistema de controle compartilhado.

Neste trabalho, as leituras do ambiente se dão por meio de um sensor LIDAR tipo *laser rangefinder* do fabricante SICK® modelo LMS200 [16], e o robô utilizado para simular a cadeira de rodas é o Pioneer P3-DX do fabricante MobileRobots®. A partir das leituras do sensor serão extraídas as entradas do sistema fuzzy, as quais serão duas, a distância ( $\rho$ ) e o ângulo ( $\alpha$ ) dos obstáculos em relação ao robô, conforme mostrado na Figura 12.

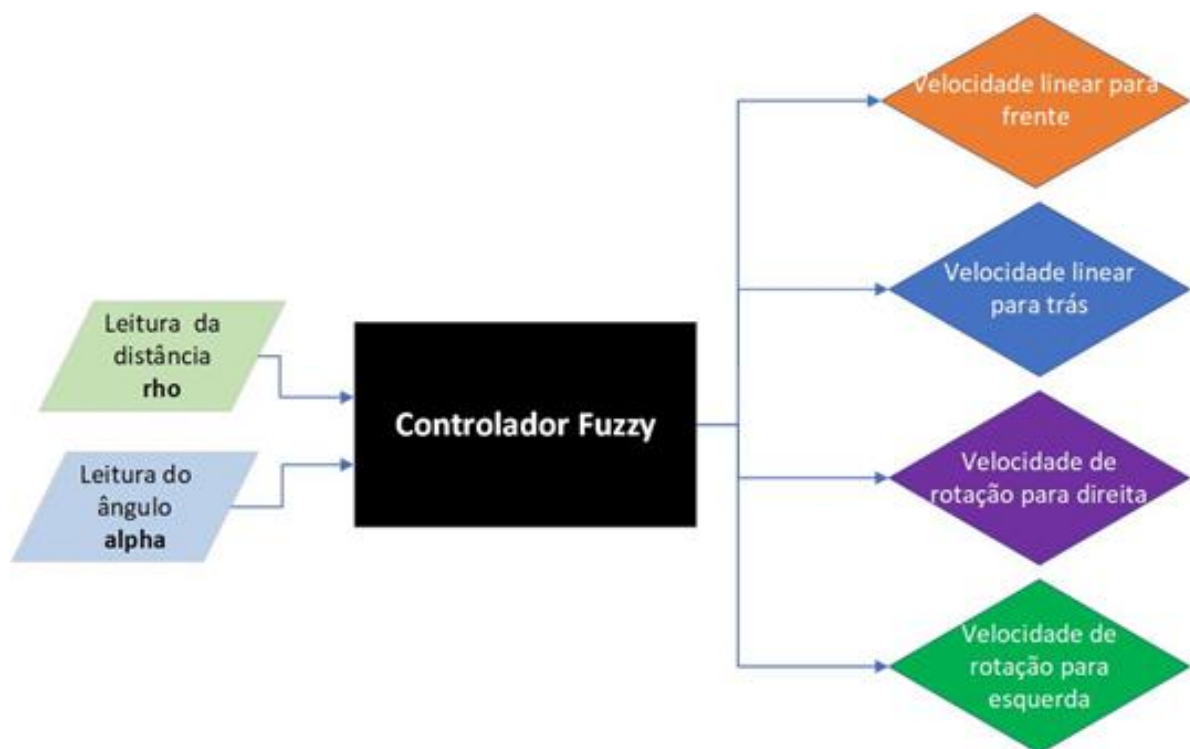
Figura 12. Ilustração das entradas do controlador *fuzzy*.



Fonte: Autor.

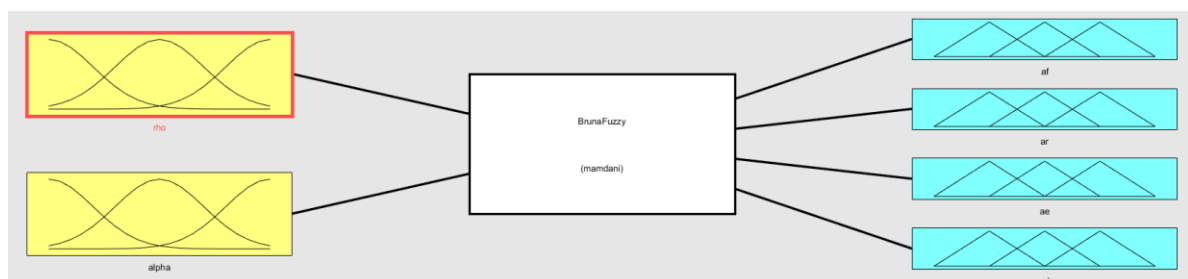
A partir dessas duas variáveis de entrada, o controlador *fuzzy* definirá quais devem ser as variáveis de saída, sendo justamente a regulação das quatro opções disponibilizadas ao usuário, as velocidades lineares para frente e para trás e as velocidades angulares (de rotação) para direita e para a esquerda. Essas serão as velocidades que o robô deverá assumir para fazer uma navegação segura no ambiente, considerando os obstáculos presentes no trajeto. Na Figura 13 pode-se observar o esquema representativo dos conceitos referentes às entradas e saídas do controlador *fuzzy* e na Figura 14 está mostrado como o *fuzzy* é representado no MATLAB.

Figura 13. Esquema representativo das entradas e saídas do controlador *fuzzy*.



Fonte: Autor.

Figura 13 – Esquema representativo no MATLAB.



Fonte: Autor.



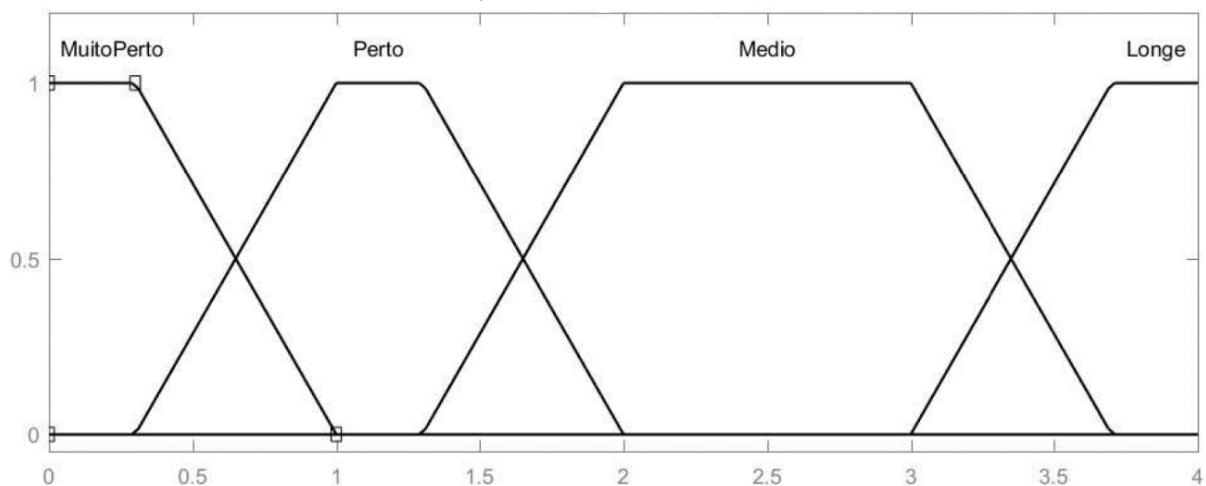
### 3.2 CONTROLADOR *FUZZY*: VARIÁVEIS DE ENTRADAS E SAÍDAS

Esta seção descreve as variáveis utilizadas bem como seus grupos *fuzzy* modelados.

#### 3.2.1 Entrada *rho*

A variável de entrada denominada por *rho* é a distância Euclidiana entre o estado atual do robô e os obstáculos presentes no ambiente. Para esta entrada arquitetou-se, por conveniência, quatro conjuntos fuzzy para designar de maneira linguística a proximidade do robô aos obstáculos, a saber, os conjuntos “Muito Perto”, “Perto”, “Médio” e “Longe”. Todos os grupos foram projetados utilizando a função de pertinência trapezoidal, e são mostrados na Figura 15.

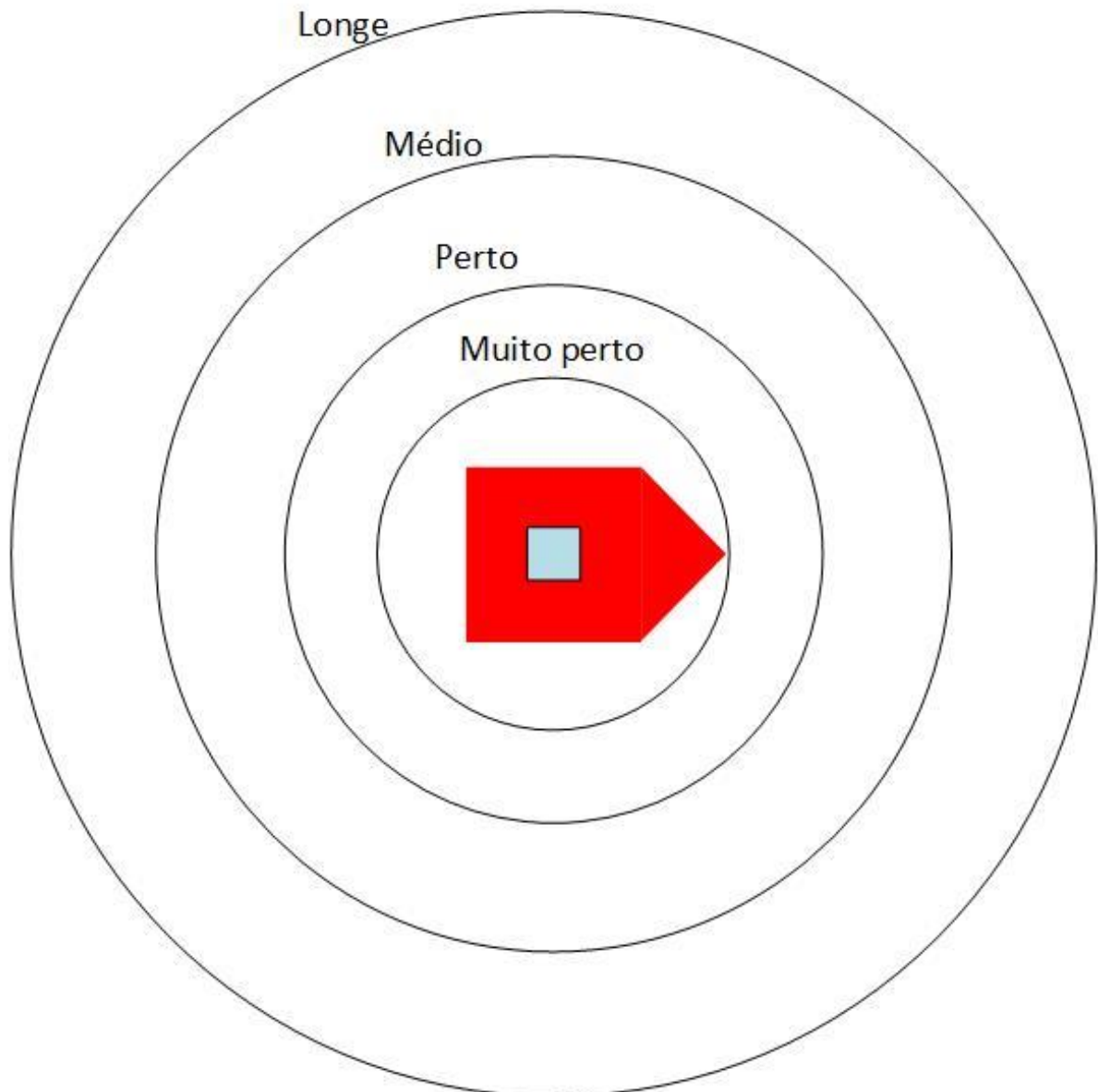
Figura 15. Entrada *rho* e seus grupos do controlador *fuzzy*.



Fonte: Autor.

Para a variável *rho* decidiu-se, de maneira empírica, que a distância máxima de influência de um obstáculo sobre o robô será de 4 metros. Primeiramente, tem-se o caso do grupo “Muito Perto”, com grau de pertinência de 100% entre os valores de 0 a 0,25, decrescendo linearmente entre os valores 0,25 e 1,00, criando o número *fuzzy* trapezoidal [0 0 0,25 1]. Para o caso “Perto”, tem-se o número *fuzzy* trapezoidal [0,25 1 1,25 2] são os principais pontos do trapézio deste grupo *fuzzy*. O mesmo raciocínio é válido para os demais casos, em que o grupo “Médio” é dado por [1,25 2 3 3,75] e “Longe” por [3 3,75 4 4]. Na Figura 16 tem-se, de forma ilustrativa, como estão dispostas as regiões consideradas.

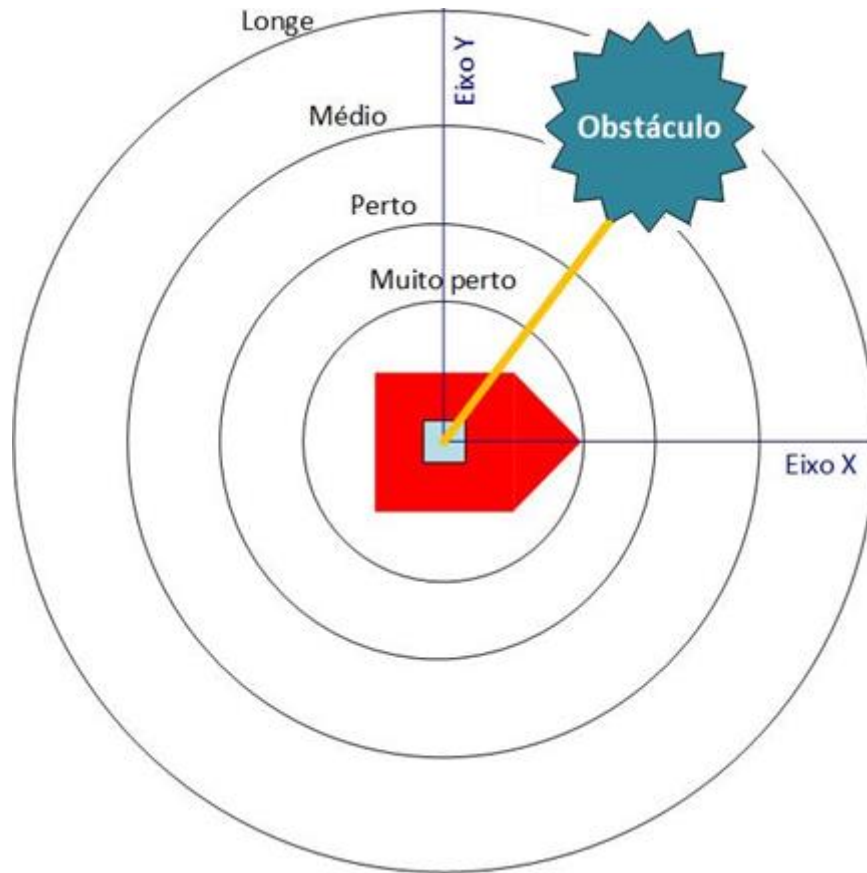
Figura 16. Regiões definidas para  $\rho$  no controlador *fuzzy*.



Fonte: Autor.

Para exemplificar, supõe-se que um obstáculo se encontra a uma distância de 3,5 metros, de acordo com o controlador *fuzzy*, tem-se uma pertinência de 0,2857 para a classificação que considera o obstáculo a uma média distância e uma pertinência de 0,7143 para classificar o obstáculo como longe. Portanto, tem-se um exemplo do conceito de “mais pra longe”. A Figura 17 ilustra a posição do obstáculo citado como exemplo.

Figura 17. Exemplo de obstáculo entre as regiões “Médio” e “Longe”.



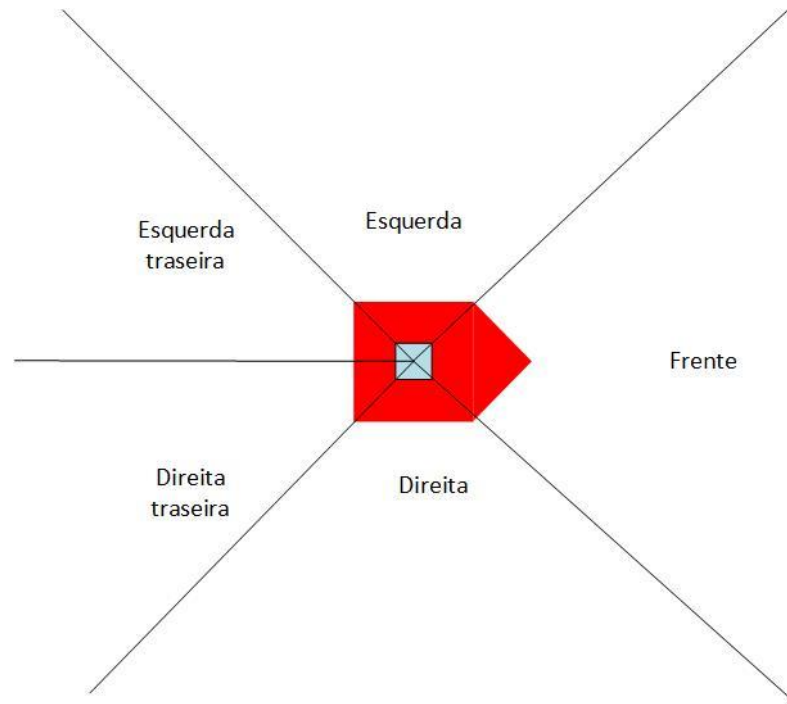
Fonte: Autor.

### 3.2.2 Entrada *alpha*

A variável de entrada denominada por *alpha* é o ângulo que os obstáculos presentes no ambiente se encontram com relação ao sistema de coordenadas do estado atual do robô. Para esta variável são considerados cinco grupos *fuzzy* referentes às regiões do robô, a saber, “Direita Traseira”, “Direita”, “Frente”, “Esquerda” e “Esquerda Traseira”. Tais regiões estão mostradas na Figura 18, e os respectivos grupos *fuzzy* estão mostrados na Figura 19.

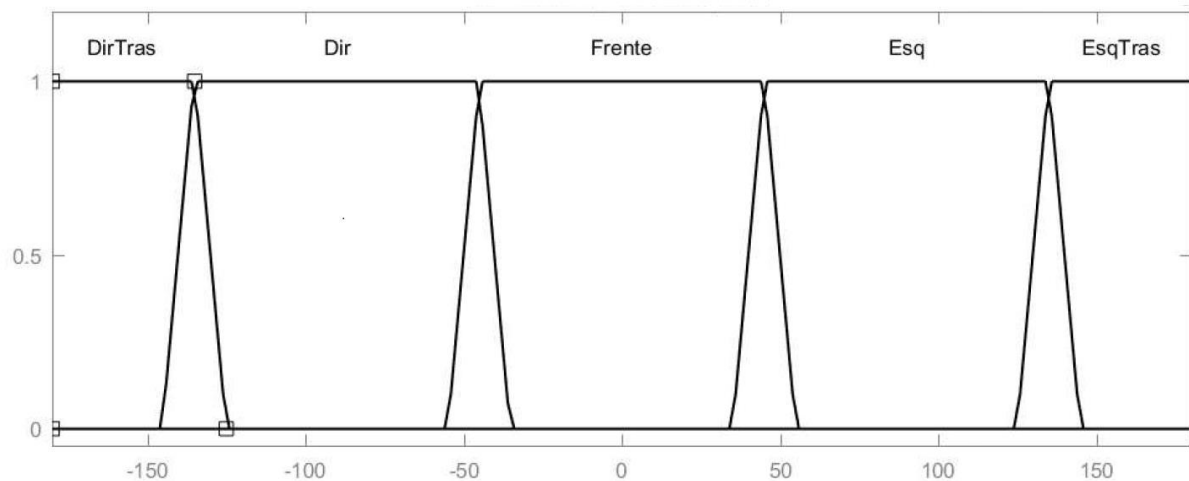
Desta maneira, sempre que um obstáculo se apresentar no entorno do robô, ele poderá ser classificado em uma dessas cinco regiões. Assim, é possível para o sistema decidir quais são as regiões ameaçadas pela proximidade de obstáculos e, assim, calcular a reação apropriada para a dada situação.

Figura 18. Regiões definidas para  $\alpha$  com base no sistema de coordenadas do robô.



Fonte: Autor

Figura 19. Entrada  $\alpha$  e seus grupos, definidos em  $[-180^\circ, 180^\circ]$ , do controlador *fuzzy*.

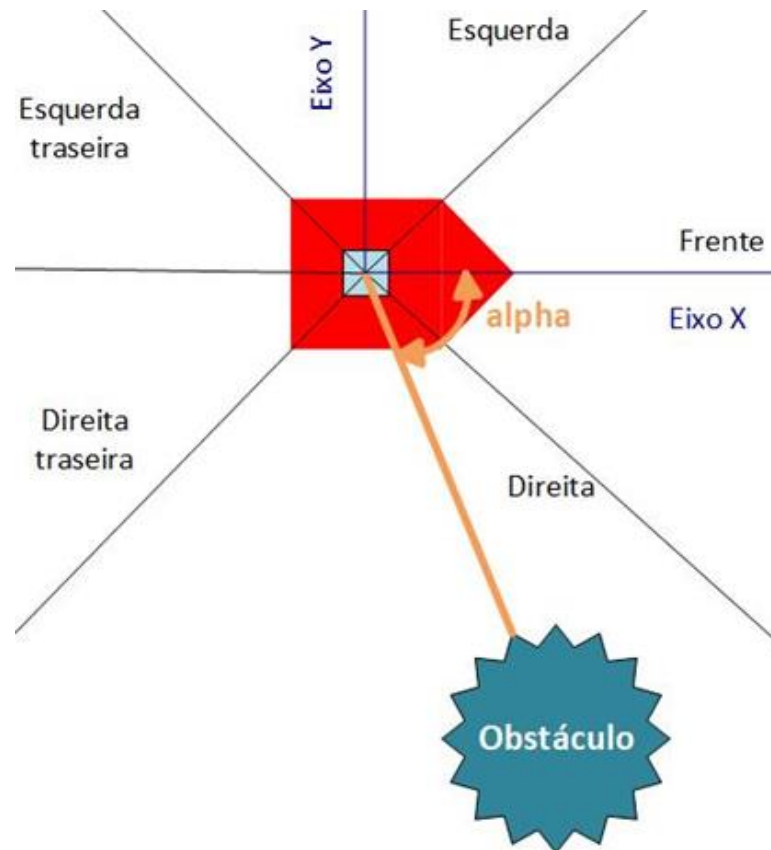


Fonte: Autor.

Se o ângulo for por exemplo de  $55^\circ$  então de acordo com o controlador utilizado, tem-se que o obstáculo se encontra com um grau de pertinência 1 a direita do robô e com grau de pertinência 0 em relação a região frontal do robô. Portanto, neste caso tem-se que o obstáculo está à direita do robô. Existem as regiões de limiar que são as regiões onde deixa de ser considerado “Direita” por exemplo e passa a ser considerado “Frente”. Nestes casos tem-se

valores de pertinência que representarão quanto está pertencendo a uma região e quanto está pertencendo a outra. A Figura 20 apresenta um exemplo do ângulo calculado.

Figura 20. Exemplo de obstáculo na região a direita do robô.



Fonte: Autor.

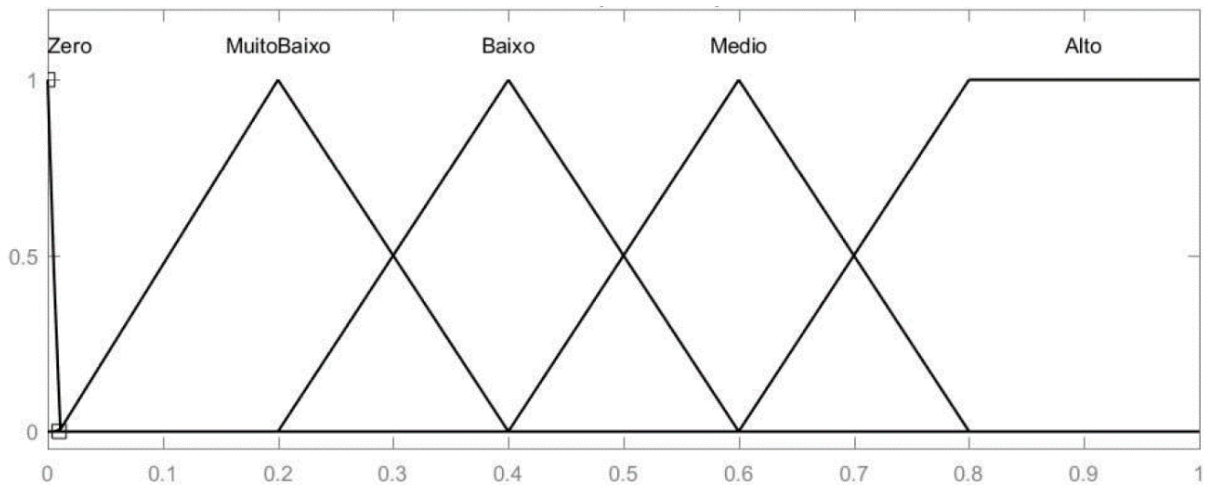
### 3.2.3 Velocidades de saída

Conforme mencionado, as saídas do controlador *fuzzy* serão: velocidade linear para frente e para trás, velocidades de rotação para direita e para esquerda. Essas velocidades serão obtidas por meio de quatro fatores de escada contidos no intervalo  $[0,1]$ , denominados por  $af$  para velocidade para frente,  $ar$  para a velocidade para trás (ré),  $ae$  para rotação à esquerda e  $ad$  para rotação à direita, que serão calculados pelo sistema *fuzzy*. Esses fatores serão, então, multiplicados pelas velocidades linear e angular máximas permitidas ao robô. Dessa maneira, o sistema *fuzzy* produz quatro velocidades, duas lineares (frente e trás) e duas angulares (esquerda e direita), independentes e descorrelacionadas entre si.

O fato de se ter quatro velocidades diferentes pode produzir, por exemplo, velocidades angulares diferentes entre a esquerda e à direita. Elas estarão ligadas ao contexto dos obstáculos

que se apresentam no ambiente. Para tanto, projetou-se cinco grupos a saber, “Zero”, “Muito Baixo”, “Baixo”, “Médio” e “Alto”, e todos os quatro fatores de escala possuem os mesmos grupos dispostos da mesma maneira. A Figura 21 mostra a função de pertinência utilizada para os quatro fatores que produzem as velocidades do robô.

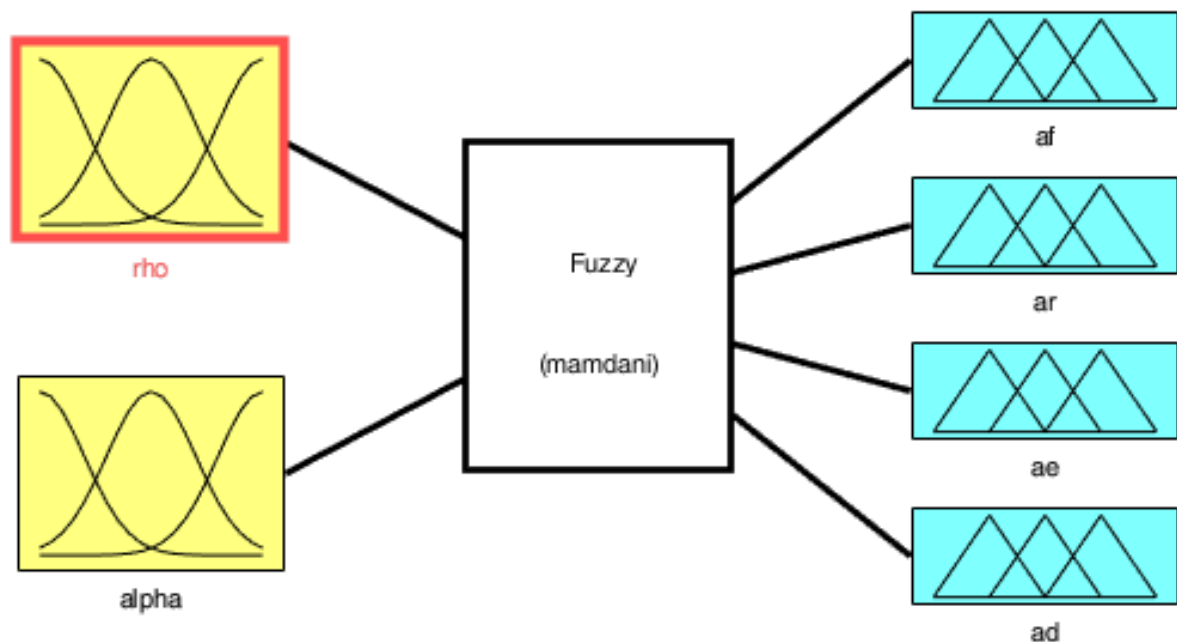
Figura 21. Perfil dos fatores de escala que produzem as velocidades de saída do robô.



Fonte: Autor.

Portanto, de posse das variáveis de entrada e saída, tem-se então o sistema fuzzy modelado para o problema de controle compartilhado, descrito na Figura 22.

Figura 22. Perfil dos fatores de escala que produzem as velocidades de saída do robô.



Fonte: Autor.

### 3.3 BASE DE REGRAS

Quando o controlador *fuzzy* recebe as duas entradas *rho* e *alpha*, o sistema *fuzzy* deverá determinar, com base no contexto dado pelas variáveis de entrada, quais velocidades são as mais adequadas para que o robô possa navegar com a maximização da segurança do usuário. Em outras palavras, as variáveis de entrada deverão estar relacionadas com as variáveis de saída por meio de uma base de regras que, em última instância, será utilizada para determinar o cálculo das velocidades do robô. Sendo assim, se algum obstáculo for detectado, é necessário determinar, por meio da base de regras, quais serão as velocidades do robô naquela situação.

A base de regras é onde organiza-se os conhecimentos já adquiridos do ambiente e é feita a tomada de decisão diante das diversas situações possíveis. O conjunto de regras do sistema *fuzzy* descrito deverá então possuir um conjunto de 20 regras que se devem a combinação das cinco possíveis entradas de *rho* com as quatro possíveis entradas de *alpha*. Tem-se então que para as primeiras 5 regras relacionam *rho* como “Muito perto” conforme mostrado na Figura 23.

Conforme mostrado na Figura 23 (a), o obstáculo se encontra na direita traseira e muito perto do robô, então a velocidade de rotação para a direita deve ser muito baixa pois ela leva a frente do robô ao alinhamento com o obstáculo, a velocidade para trás deve ser nula para evitar a colisão, a velocidade rotacional à esquerda pode ser maior que a direita pois ela leva a uma área livre, e a velocidade para frente pode ser não nula pois ela também leva a uma região livre de obstáculos, ou seja, maximiza a segurança do usuário e do robô.

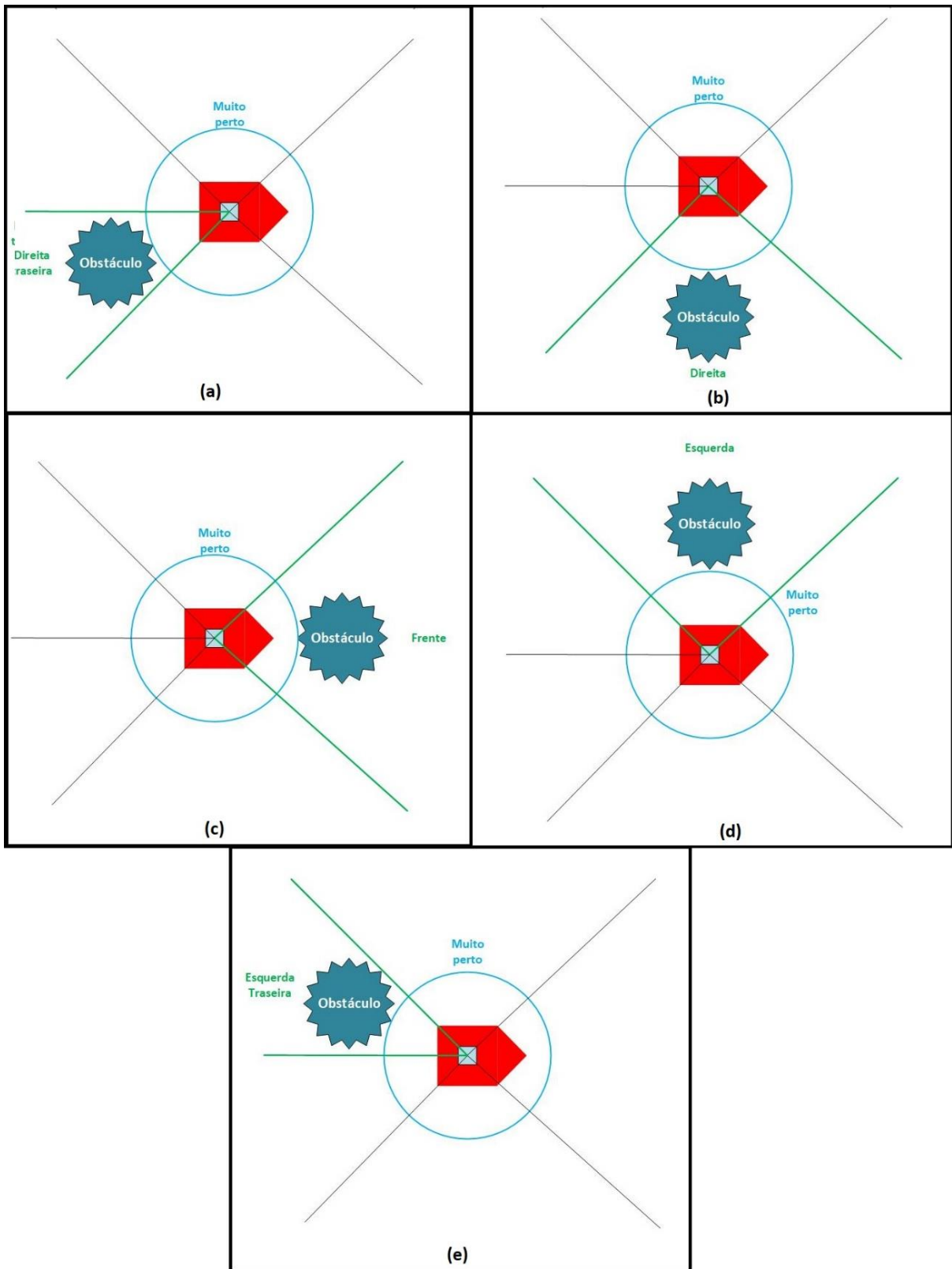
Na Figura 23 (b), o obstáculo se encontra a direita do robô, então não só a velocidade de rotação para a direita deve ser baixa, mas também as velocidades para frente e para trás. Esta é a situação do caso em que o condutor deseja navegar próximo a uma parede ou obstáculo.

Se o obstáculo se encontra a frente e muito perto do robô, como mostrado na Figura 23 (c), então a velocidade associada ao movimento de ir para frente deve ser zero, para que não haja colisão do robô com o obstáculo. Neste caso, cria-se um aviso de que há um risco de colisão iminente, para que o usuário seja notificado a respeito do risco da manobra desejada.

Assim como o caso em que o obstáculo se encontra a direita do robô, nesta situação mostrada na Figura 23 (d), tem-se as velocidades para frente e para trás e também a de rotação para a esquerda definidas como muito baixas.

Finalmente, no caso em que o obstáculo está na direita traseira do robô, nesta situação deve-se tornar muito baixa apenas a velocidade de rotação para a esquerda e zerar a velocidade para trás, a Figura 22 (e) exemplifica este caso.

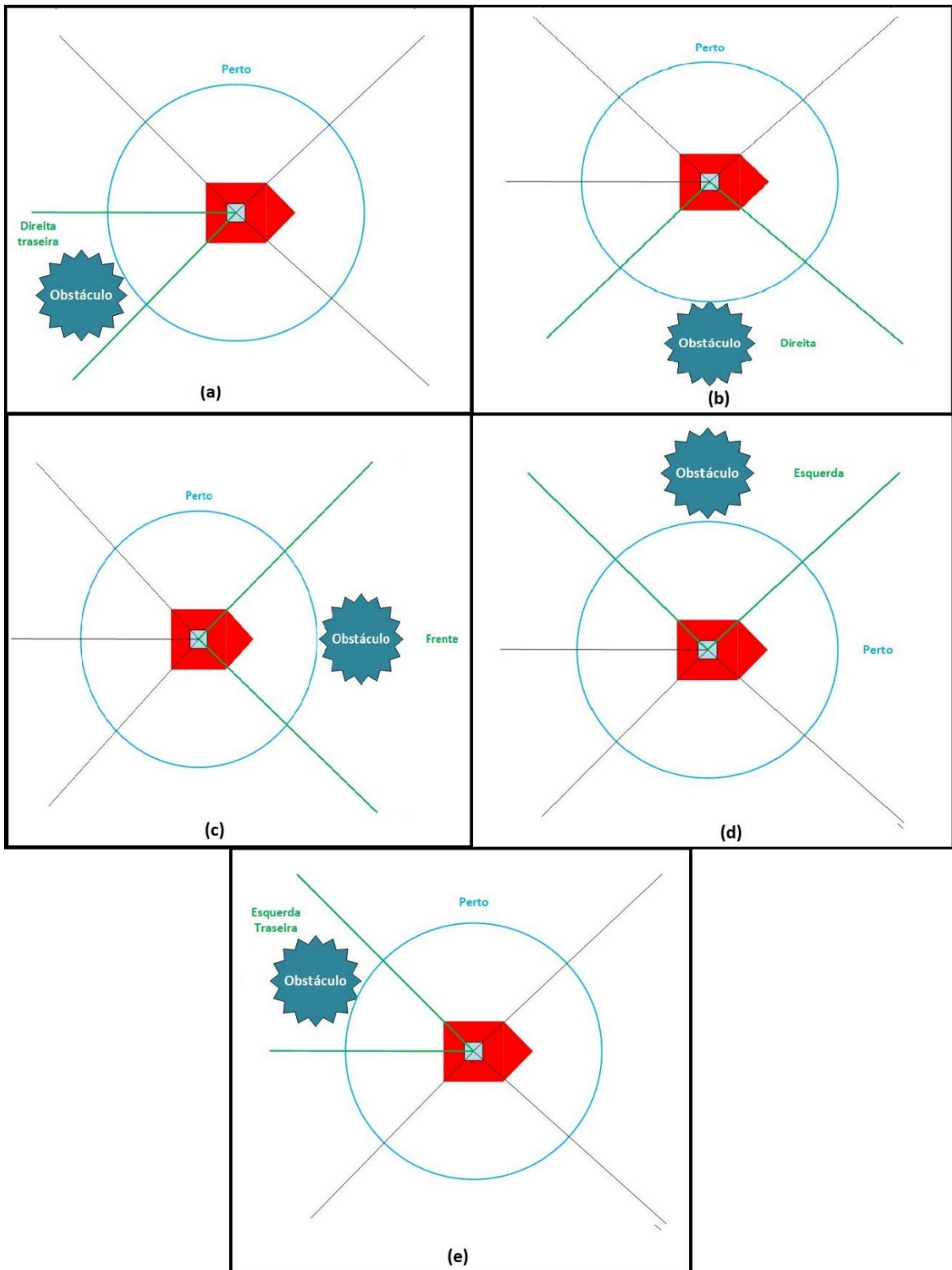
Figura 23. Exemplos para “Muito Perto” de  $\rho$  com todas as opções disponíveis em  $\alpha$ .



Fonte: Autor.



Figura 24. Exemplos para “Perto” de  $\rho$  com todas as opções disponíveis em  $\alpha$ .



Fonte: Autor

Para a Figura 24, tem-se os seguintes casos. Para o caso em que o obstáculo está sendo considerado perto e na direita traseira, associa-se uma velocidade baixa de rotação para a direita e uma velocidade muito baixa de ré, afim de evitar colisões. Pode-se observar que é permitido ao usuário da cadeira de rodas se locomover próximo a obstáculos, porém, com a velocidade consideravelmente reduzido afim de tornar a navegação segura.

No caso em que o obstáculo está perto e a direita do robô, então, para que haja uma navegação segura a velocidade de rotação para a direita será baixa, mas as demais serão média, isto porque seria o caso em que o robô não está necessariamente indo em direção ao obstáculo, mas apenas passando próximo a ele, como por exemplo a navegação próxima a uma parede.

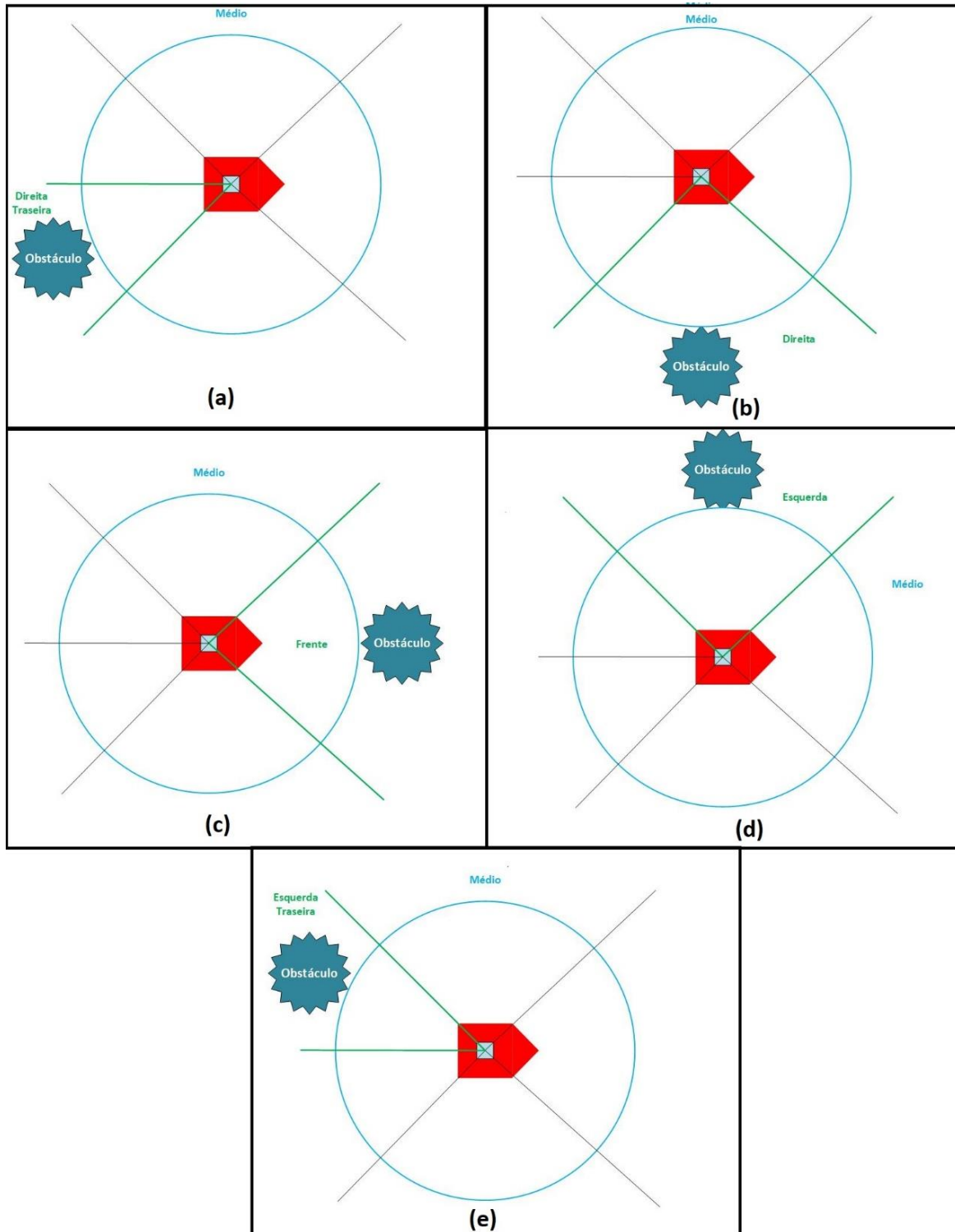
Se o obstáculo está sendo considerado perto e a frente do robô, então é necessário que a velocidade para frente seja muito baixa e as demais podendo ser média, uma vez que são velocidades associadas ao desvio do obstáculo. Neste caso, utiliza-se como exemplo o caso em que o cadeirante deseja ver uma paisagem pela janela, ou mesmo um porta retrato em cima de um móvel e por esta razão deseja ir em direção ao suposto obstáculo. Conseqüentemente, é necessário permitir que o usuário faça a ação, mas minimizando os riscos.

Assim como no caso em que o robô está perto e a direita do obstáculo, neste caso a rotação será baixa na direção que se encontra interrompida e as demais velocidades podendo assumir valores médios já que não representa risco de colidir se o usuário utilizar as demais direções.

Para obstáculos classificados como perto e a esquerda traseira do robô, assim como no caso da direita traseira, deve-se diminuir significativamente a velocidade para trás evitando assim uma colisão, e ainda tornar as velocidades para esquerda e para a direita como baixas.

Na figura 25, tem-se então os cinco casos descritos para o valor de  $\rho$  considerado como uma média distância do obstáculo detectado.

Figura 25. Exemplos para “Médio” de  $\rho$  com todas as opções disponíveis em  $\alpha$ .



Fonte: Autor.

Para a Figura 25 tem-se os seguintes casos. Para os obstáculos que estejam a uma média distância do robô e na parte direita traseira, a tratativa pode ser de forma que apenas a velocidade de ré seja baixa, enquanto para direita e esquerda sejam médias, e para frente seja alta, já que nesta direção o robô está se afastando do obstáculo.

Se o robô se encontra a direita um obstáculo a uma média distância, pode-se considerar que apenas a velocidade para a direita deve ser baixa.

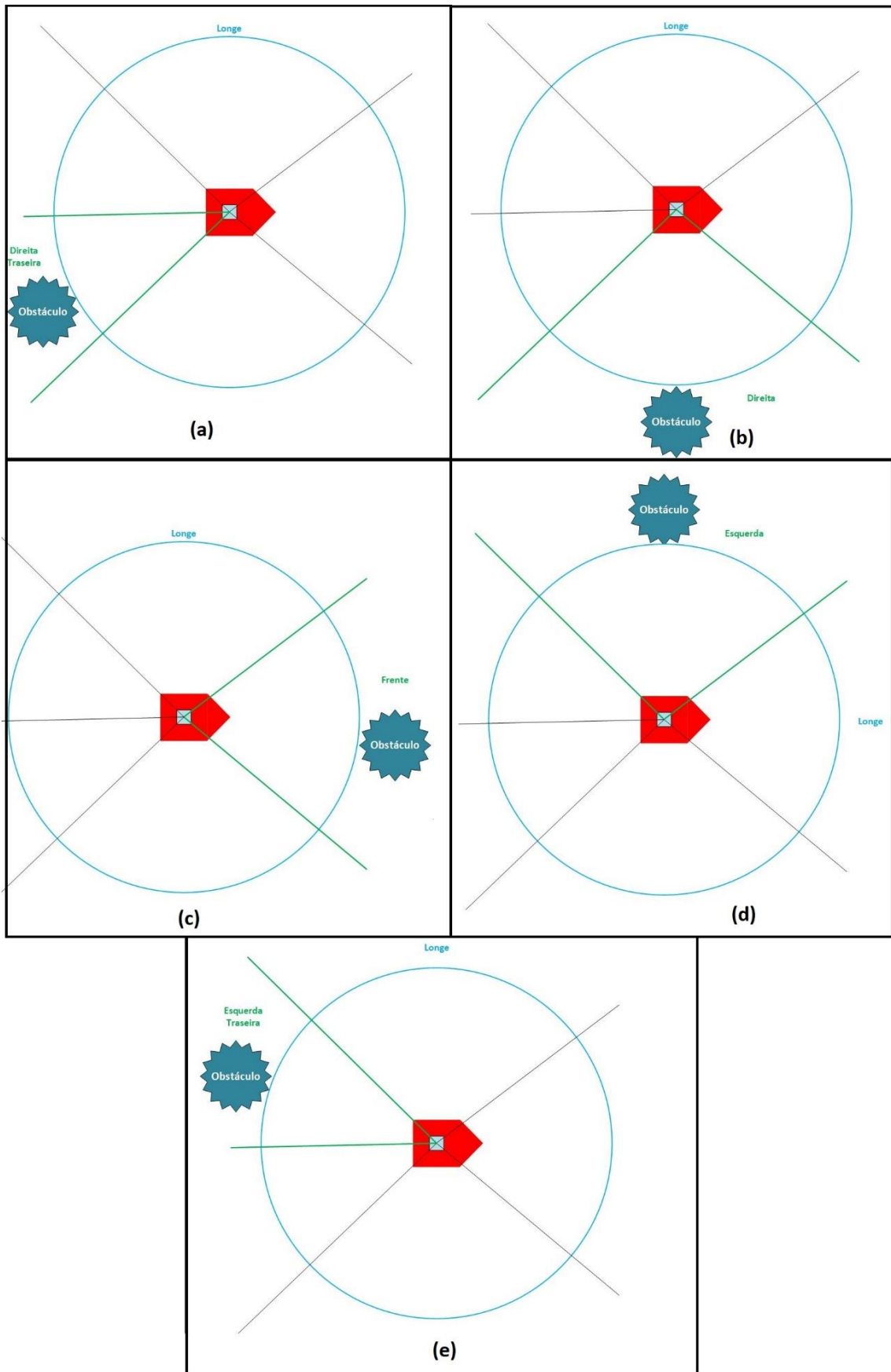
Se o obstáculo se encontrar a uma média distância a frente do robô, então a velocidade para frente deverá ser baixa, e a velocidade na direção contrária ao obstáculo deverá ser a mais alta.

Assim como o caso em que o obstáculo se encontra a esquerda e a distância é considerada média, então a velocidade para a esquerda deverá ser baixa.

Se o obstáculo se encontrar a uma distância média e na esquerda traseira do robô, então a velocidade de ré deverá ser baixa para evitar a colisão e a velocidade para frente deverá ser alta, já que seria a velocidade que faria o robô se movimentar para longe do obstáculo.

Na Figura 26, tem-se então os cinco casos descritos para o valor de  $\rho$  considerado como longe do robô.

Figura 26. Exemplos para “Longe” de  $\rho$  com todas as opções disponíveis em  $\alpha$ .



Fonte: Autor.

Para a Figura 26 tem-se os seguintes casos. Para o caso em que o obstáculo se encontra longe do robô, neste caso, na região da direita traseira do robô, então a única velocidade que deverá ser média, seria a velocidade de ré, já que é nessa direção que se encontra o empecilho.

Para o caso em que o obstáculo se encontra a direita do robô, porém longe dele, então a tratativa seria que o robô virasse para a direita com média velocidade enquanto as demais podem ser velocidades altas.

Para o caso em que o obstáculo se encontra a frente do robô, porém longe, então a velocidade para frente será média e as velocidades para as demais direções podem ser altas.

Assim como no caso em que o obstáculo está longe e a direita do robô, neste caso só precisa haver uma preocupação quanto a velocidade de rotação para a esquerda, que seria na direção do obstáculo. Nas demais direções pode-se permitir a velocidade alta, já que não existem empecilhos sendo detectados.

Por último tem-se o caso em que o obstáculo que se encontra a esquerda traseira do robô está longe do robô. Neste caso, deve-se preocupar apenas com a velocidade de ré que seria em direção ao obstáculo.

Colocando os raciocínios descritos anteriormente no formato de regras tipo “Se/Então” conforme Mamdani, tem-se, finalmente, o conjunto de regras utilizadas para todas as situações possíveis:

1. Se (***rho*** é “Muito Perto”) e (***alpha*** é “Direita Traseira”), então:  
(***af*** é “Baixo”), (***ar*** é “Zero”), (***ae*** é “Médio”) e (***ad*** é “Muito Baixo”).
2. Se (***rho*** é “Muito Perto”) e (***alpha*** é “Direita”), então:  
(***af*** é “Muito Baixo”), (***ar*** é “Muito Baixo”), (***ae*** é “Médio”) e (***ad*** é “Muito Baixo”).
3. Se (***rho*** é “Muito Perto”) e (***alpha*** é “Frente”), então:  
(***af*** é “Zero”), (***ar*** é “Baixo”), (***ae*** é “Médio”) e (***ad*** é “Médio”).
4. Se (***rho*** é “Muito Perto”) e (***alpha*** é “Esquerda”), então:  
(***af*** é “Muito Baixo”), (***ar*** é “Muito Baixo”), (***ae*** é “Muito Baixo”) e (***ad*** é “Médio”).
5. Se (***rho*** é “Muito Perto”) e (***alpha*** é “Esquerda Traseira”), então:  
(***af*** é “Baixo”), (***ar*** é “Zero”), (***ae*** é “Muito Baixo”) e (***ad*** é “Médio”).
6. Se (***rho*** é “Perto”) e (***alpha*** é “Direita Traseira”), então:

- (**af** é “Médio”), (**ar** é “Muito Baixo”), (**ae** é “Médio”) e (**ad** é “Baixo”).
7. Se (**rho** é “Perto”) e (**alpha** é “Direita”), então:  
(**af** é “Médio”), (**ar** é “Médio”), (**ae** é “Médio”) e (**ad** é “Baixo”).
8. Se (**rho** é “Perto”) e (**alpha** é “Frente”), então:  
(**af** é “Muito Baixo”), (**ar** é “Médio”), (**ae** é “Médio”) e (**ad** é “Médio”).
9. Se (**rho** é “Perto”) e (**alpha** é “Esquerda”), então:  
(**af** é “Médio”), (**ar** é “Médio”), (**ae** é “Baixo”) e (**ad** é “Médio”).
10. Se (**rho** é “Perto”) e (**alpha** é “Esquerda Traseira”), então:  
(**af** é “Médio”), (**ar** é “Muito Baixo”), (**ae** é “Baixo”) e (**ad** é “Baixo”).
11. Se (**rho** é “Médio”) e (**alpha** é “Direita Traseira”), então:  
(**af** é “Alto”), (**ar** é “Baixo”), (**ae** é “Médio”) e (**ad** é “Médio”).
12. Se (**rho** é “Médio”) e (**alpha** é “Direita”), então:  
(**af** é “Alto”), (**ar** é “Alto”), (**ae** é “Médio”) e (**ad** é “Baixo”).
13. Se (**rho** é “Médio”) e (**alpha** é “Frente”), então:  
(**af** é “Baixo”), (**ar** é “Alto”), (**ae** é “Médio”) e (**ad** é “Médio”).
14. Se (**rho** é “Médio”) e (**alpha** é “Esquerda”), então:  
(**af** é “Alto”), (**ar** é “Alto”), (**ae** é “Baixo”) e (**ad** é “Médio”).
15. Se (**rho** é “Médio”) e (**alpha** é “Esquerda Traseira”), então:  
(**af** é “Alto”), (**ar** é “Baixo”), (**ae** é “Baixo”) e (**ad** é “Baixo”).
16. Se (**rho** é “Longe”) e (**alpha** é “Direita Traseira”), então:  
(**af** é “Alto”), (**ar** é “Médio”), (**ae** é “Alto”) e (**ad** é “Alto”).
17. Se (**rho** é “Longe”) e (**alpha** é “Direita”), então:  
(**af** é “Alto”), (**ar** é “Alto”), (**ae** é “Alto”) e (**ad** é “Médio”).
18. Se (**rho** é “Longe”) e (**alpha** é “Frente”), então:  
(**af** é “Médio”), (**ar** é “Alto”), (**ae** é “Alto”) e (**ad** é “Alto”).

19. Se ( $\rho$  é “Longe”) e ( $\alpha$  é “Esquerda”), então:

( $af$  é “Alto”), ( $ar$  é “Alto”), ( $ae$  é “Médio”) e ( $ad$  é “Alto”).

20. Se ( $\rho$  é “Longe”) e ( $\alpha$  é “Esquerda Traseira”), então:

( $af$  é “Alto”), ( $ar$  é “Médio”), ( $ae$  é “Alto”) e ( $ad$  é “Alto”).

Pode-se observar nas regras que para casos onde o obstáculo se encontra muito perto, a velocidade associada aquela direção deve ser nula para evitar a colisão. Nos casos onde o obstáculo está perto ainda é permitida uma velocidade muito baixa na direção obstruída enquanto nas demais direções associadas a velocidade é baixa exceto no sentido oposto ao do obstáculo, tudo isto para tornar a navegação segura.



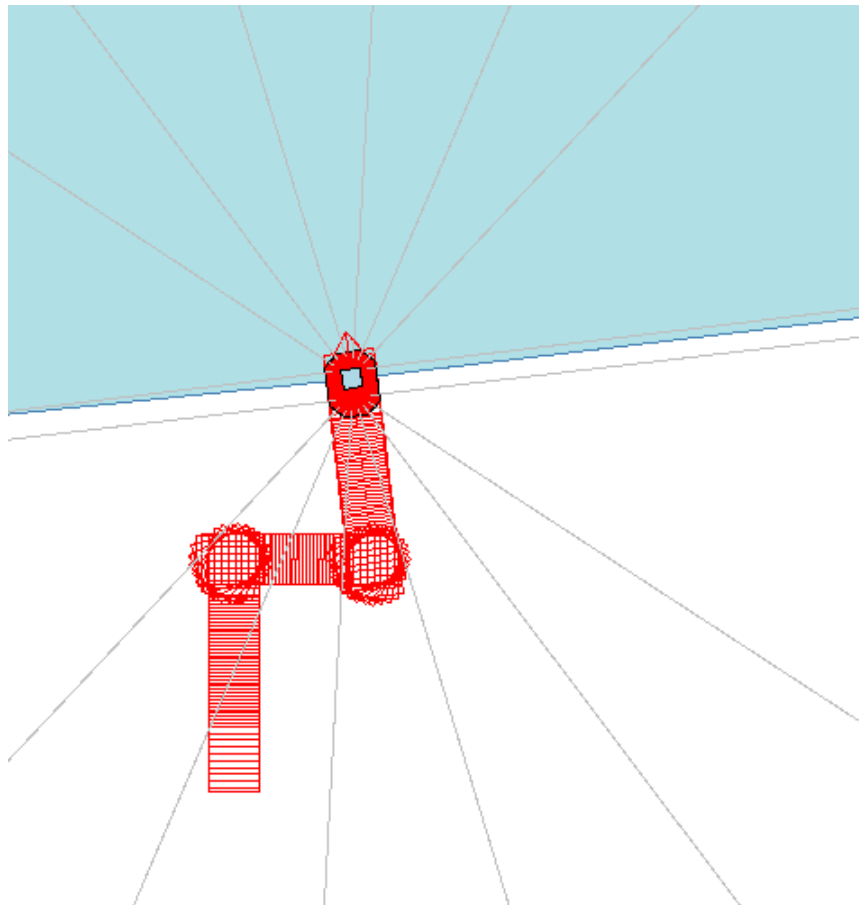
## 4 RESULTADOS

### 4.1 SITUAÇÕES ISOLADAS

Primeiramente, deve-se avaliar os resultados comportamentais do robô de forma isolada, para que seja possível verificar o efeito de cada aspecto tratado pelo controlador *fuzzy*. Desta forma, quando os resultados se tornarem combinados, será possível diferenciar quais foram as combinações que geraram aquela decisão.

O primeiro caso é a navegação livre de obstáculos, tratada na Figura 27. Neste primeiro percurso, isento de qualquer obstáculo, é possível observar a atuação das velocidades linear e angular do robô, que não são influenciadas pelos obstáculos, e, portanto, permanecem constantes durante a navegação, fato que pode ser observado no rastro deixado pelo robô, em que os retângulos estão todos igualmente espaçados. O percurso foi colocado de forma que se possa observar exatamente os momentos em que foram comandos para frente e os momentos em que os comandos foram virar para direita e virar para a esquerda, respectivamente.

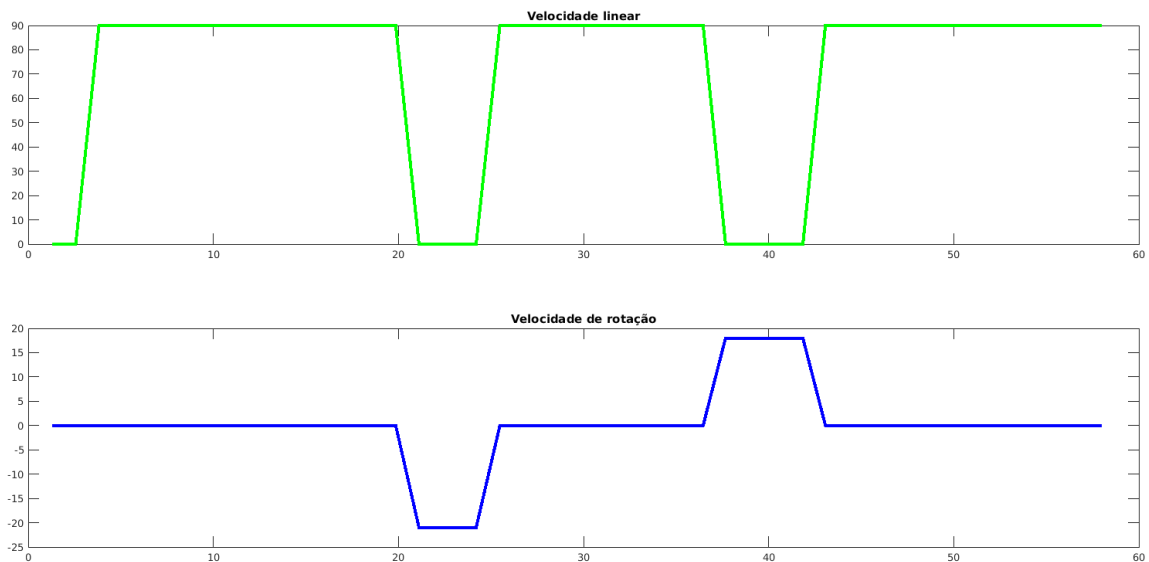
Figura 27. Percurso 01 – Navegação sem obstáculos.



Fonte: Autor.

No entanto, a fim de observar de maneira qualitativa as velocidades linear e angular (de rotação) do percurso exibido na Figura 27, tem-se as velocidades mencionadas identificadas na Figura 28. Observa-se que os momentos em que a velocidade linear é nula, são exatamente os momentos em que a velocidade angular atua, mostrando então os trechos em que o usuário manda diferentes comandos. Além disso, é notável que quando a velocidade para a direita está atuando seu valor no gráfico é representado por valores negativos enquanto nos casos em que a velocidade está para rotacionar a esquerda, o valor desta velocidade é positivo, considerando, portanto, a regra da mão direita.

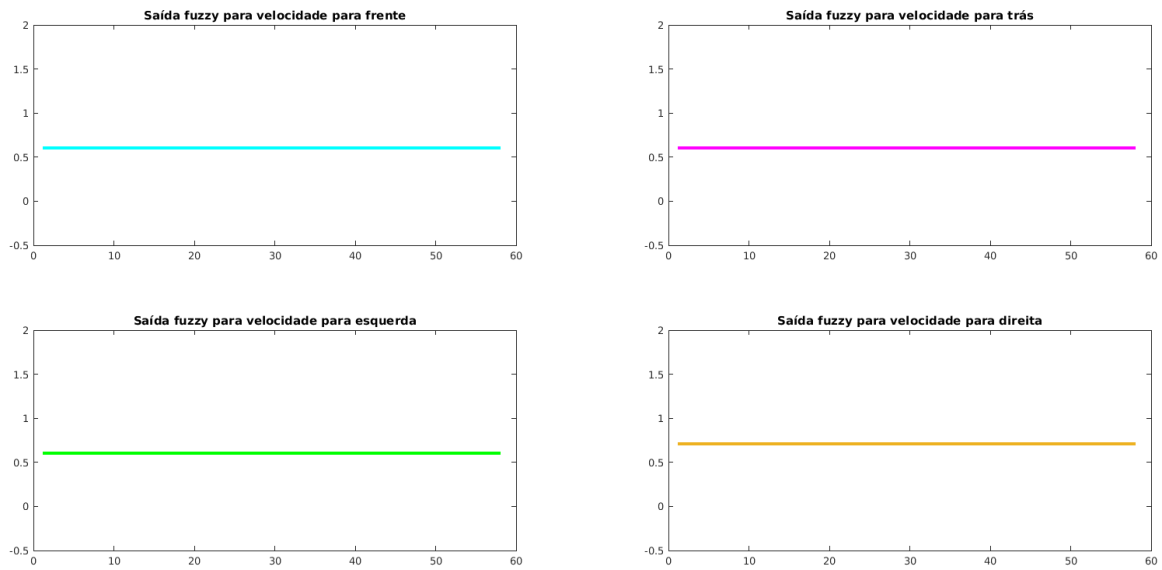
Figura 28. Velocidades linear e angular para o percurso 01.



Fonte: Autor.

Para identificar os parâmetros que deram origem a estas velocidades, os parâmetros  $af$ ,  $ar$ ,  $ae$  e  $ad$ , que são as saídas do controlador *fuzzy*, seus respectivos valores estão mostrados na Figura 29. Como não existem obstáculos na cena, seus valores de saída são os máximos permitidos pelo controlador *fuzzy* e são valores constantes, mostrando então que não foram detectados obstáculos e portanto o robô pode navegar com sua velocidade máxima sem nenhum risco de colisão.

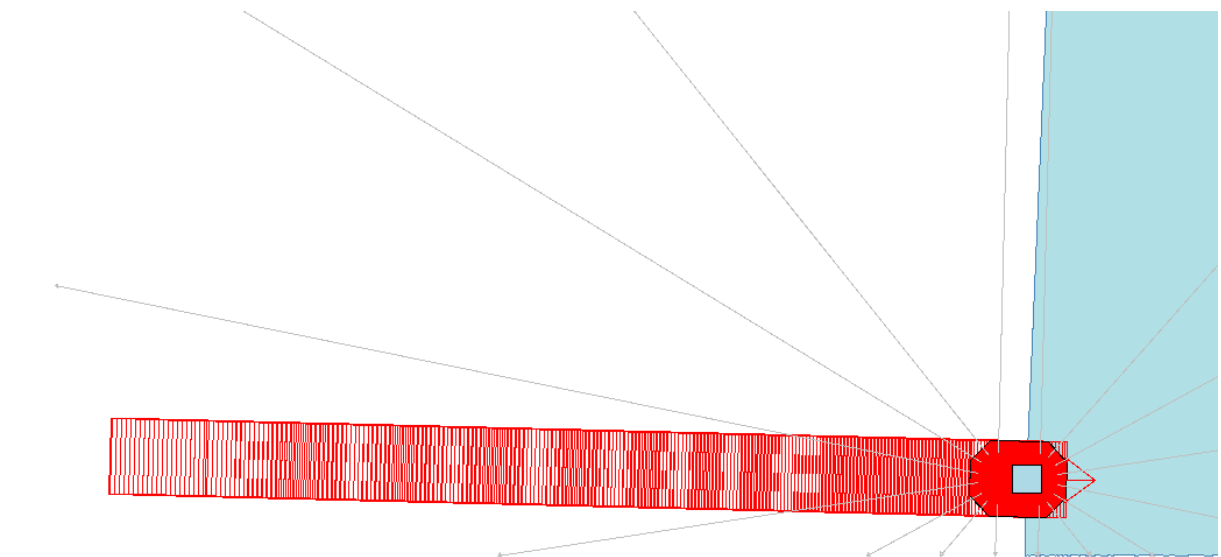
Figura 29. Saídas do controlador *fuzzy* para o percurso 01.



Fonte: Autor.

O segundo percurso tratado é a navegação próxima de uma parede à direita, mostrada na Figura 30. No segundo percurso, é observado o comportamento do robô diante de uma navegação próxima de uma parede a direita. Por se tratar de um comportamento apenas para frente, a velocidade de rotação será nula durante todo o percurso, ou seja, não foram feitos movimentos para direita ou esquerda.

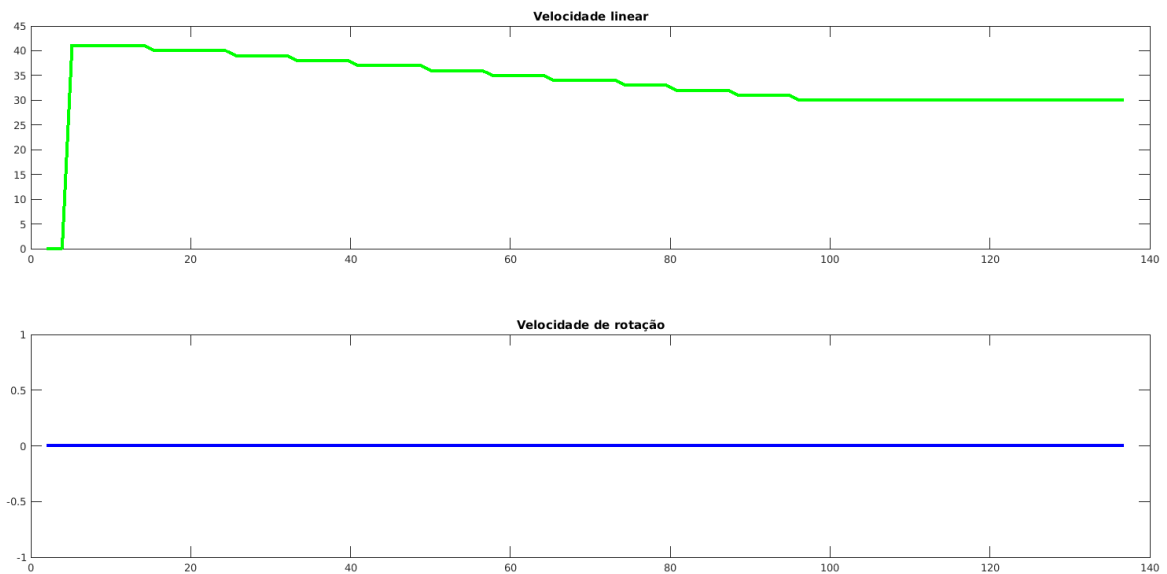
Figura 30. Percurso 02 – Navegação com uma parede a direita.



Fonte: Autor.

Na Figura 31 estão evidenciadas as velocidades linear e angular do respectivo percurso. Além disso, a velocidade linear possui um decaimento leve conforme o robô se aproxima da parede, e se torna constante uma vez que o robô se encontra ao lado da parede.

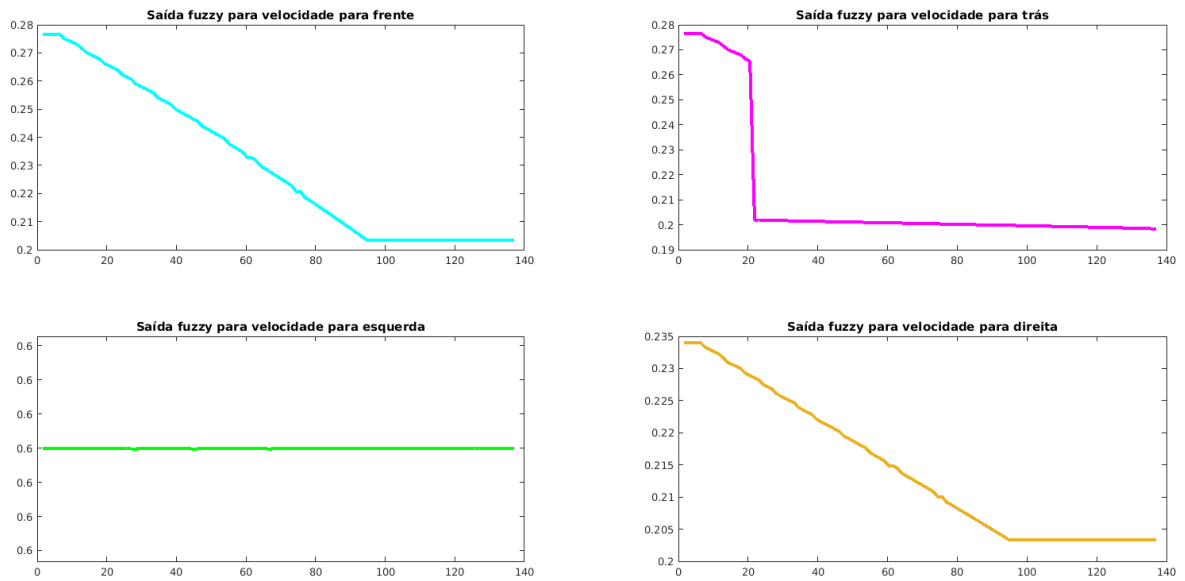
Figura 31. Velocidades linear e angular para o percurso 02.



Fonte: Autor.

As saídas *fuzzy* que produzem as velocidades linear e angular estão evidenciadas na Figura 32. Já nas saídas *fuzzy* é observado que as variáveis proporcionais as velocidades para frente e para direita foram decaindo a medida que o robô se aproximava da parede, enquanto a variável proporcional a velocidade de rotação para a esquerda permaneceu constante e com um valor mais elevado, isto por se tratar de uma situação em que o robô estaria evitando o obstáculo. A variável proporcional a velocidade de ré do robô decaiu no primeiro trecho pois no ponto de partida o robô não tinha nenhuma leitura de obstáculo na parte de trás, porém, a medida que foi avançando a parede foi detectada.

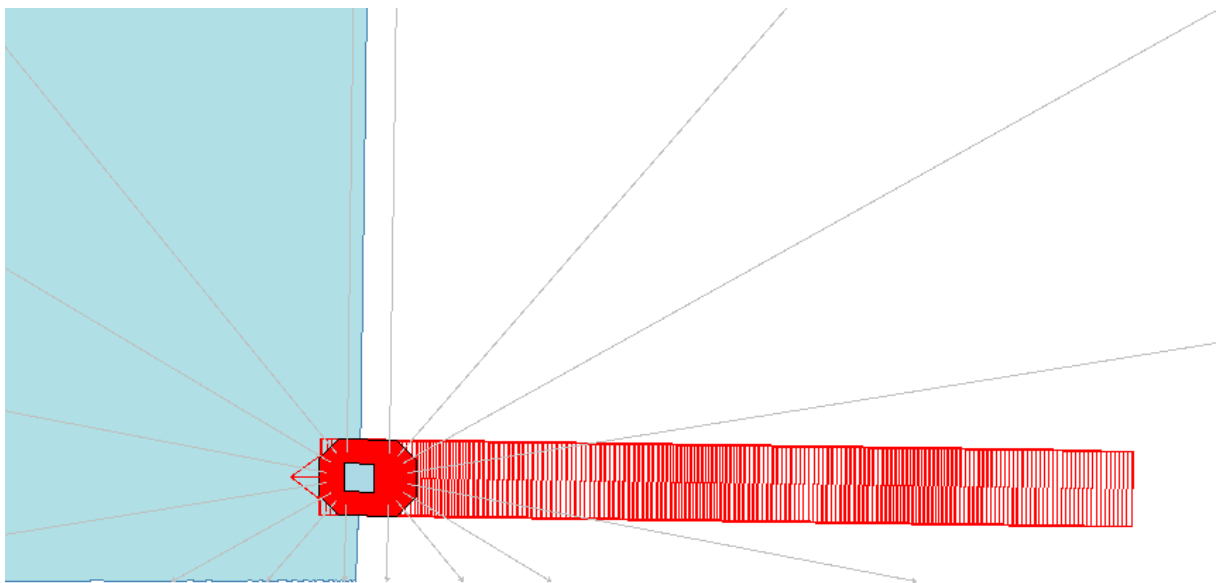
Figura 32. Saídas do controlador *fuzzy* para o percurso 02.



Fonte: Autor.

Analogamente ao caso do percurso 02, tem-se o percurso com a parede à direita do robô, evidenciado na Figura 33. Este percurso foi planejado para mostrar que o controlador trata de maneira equivalente ambos os casos.

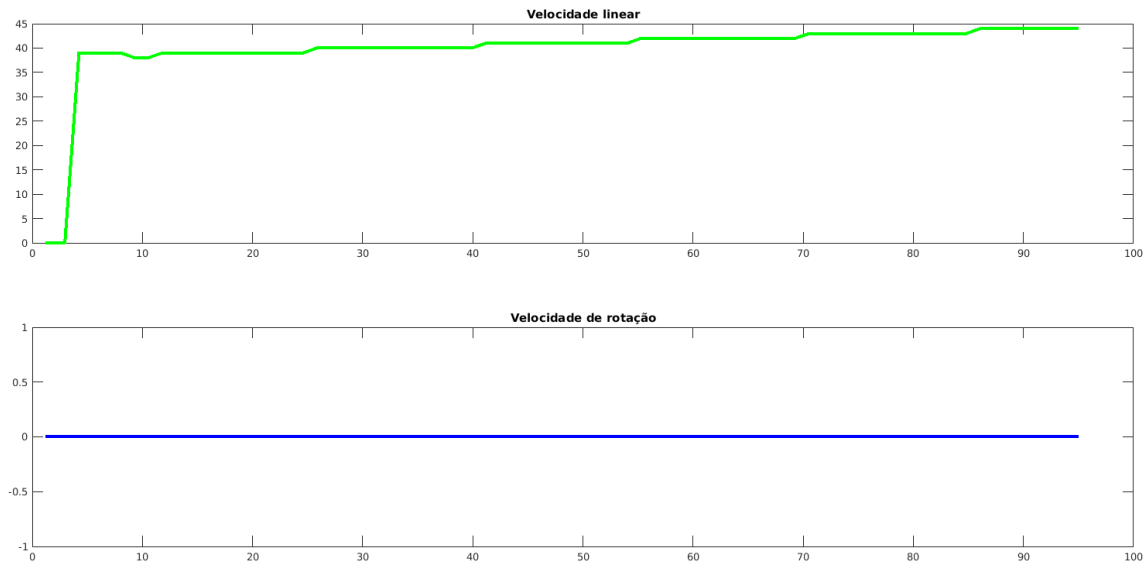
Figura 33. Percurso 03 – Navegação com uma parede a esquerda.



Fonte: Autor.

As velocidades linear e angular estão mostradas na Figura 34. Neste caso a velocidade para frente foi aumentando porque o robô estava vagarosamente se afastando da parede.

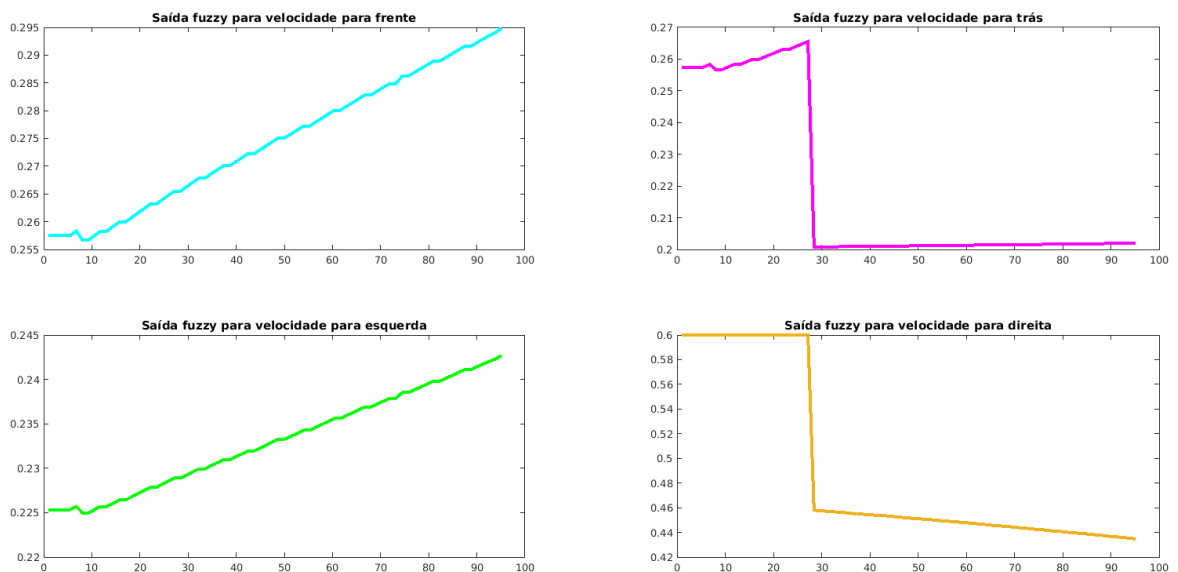
Figura 34. Velocidades linear e angular para o percurso 03.



Fonte: Autor.

As saídas *fuzzy* estão exibidas na Figura 35. É notável que a saída *fuzzy* para o percurso 02 é espelhada do percurso 03, apenas se diferenciando nas variáveis proporcionais para direita e esquerda.

Figura 35. Saídas do controlador *fuzzy* para o percurso 03.

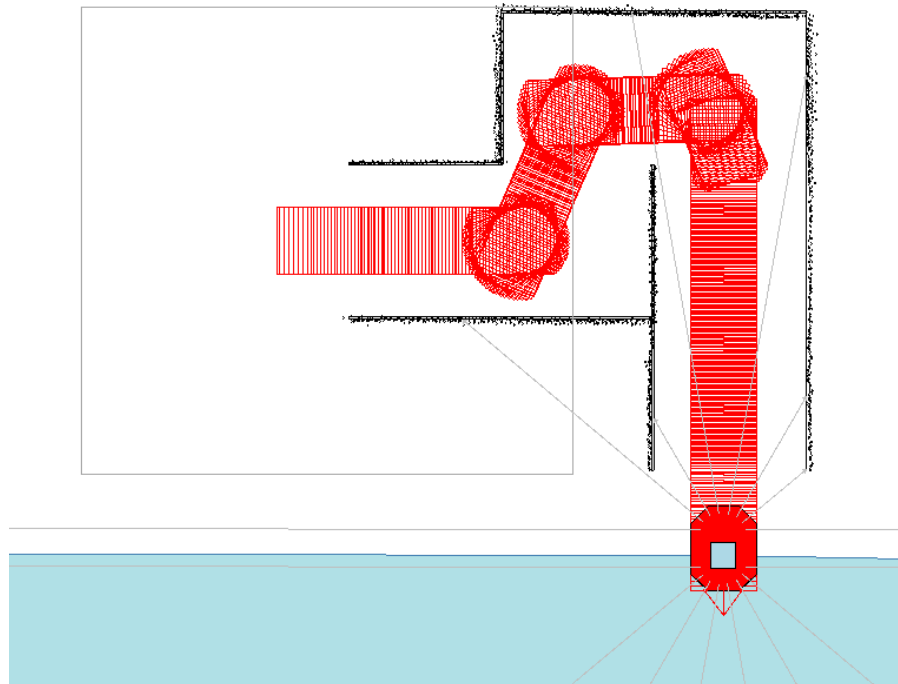


Fonte: Autor.

## 4.2 SITUAÇÕES DESAFIADORAS

No percurso 04, tem-se a navegação por um corredor estreito, conforme mostrado na Figura 36. Este exemplo trata-se de um desafio para o usuário, uma vez que o trecho demonstra diversas possibilidades de colisão, e, ainda assim, é um obstáculo comum do dia a dia de um usuário em uma cadeira de rodas. Pela figura observa-se que, apesar do desafio, o controlador compartilhado *fuzzy* auxilia o usuário a percorrer o trecho sem quaisquer colisões, controlando as velocidades e maximizando a segurança.

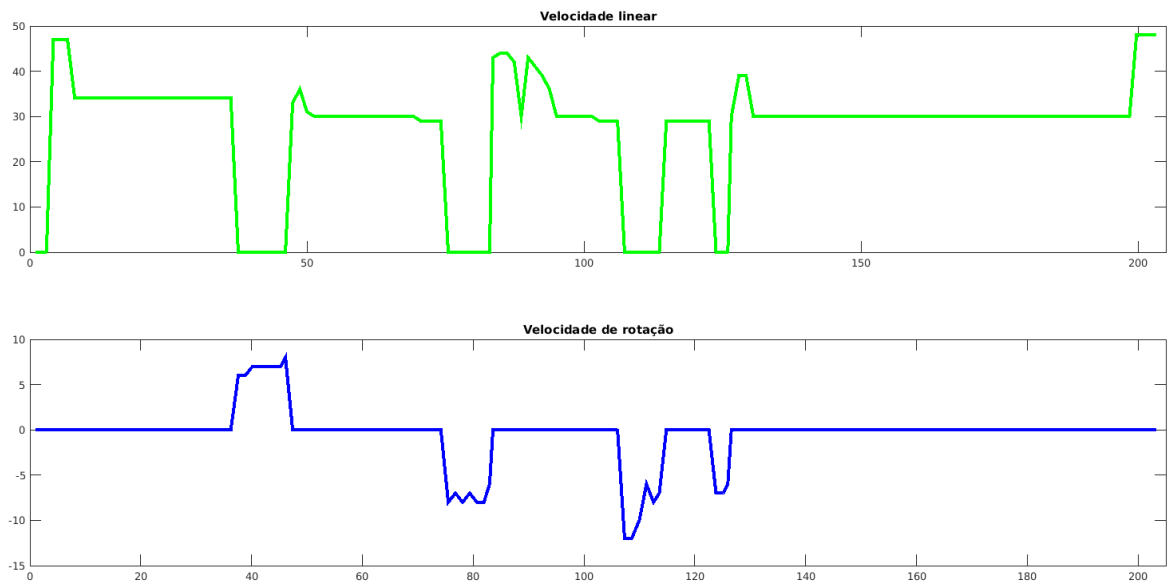
Figura 36. Percurso 04 – Navegação por um corredor.



Fonte: Autor.

As velocidades linear e angular estão evidenciadas na Figura 37. Observa-se através do gráfico de velocidades os exatos momentos em que o robô entra no corredor, onde a velocidade cai consideravelmente logo no início do gráfico e ainda os trechos em que vira primeiramente a esquerda e depois três vezes a direita. Ao final do gráfico tem-se o momento em que o robô deixa o corredor e entra em um espaço aberto. Neste momento a velocidade volta ao mesmo patamar de velocidade para quando o robô ainda não havia entrado no corredor, ou seja, a velocidade para um ambiente sem obstáculos a frente.

Figura 37. Velocidades linear e angular para o percurso 04.



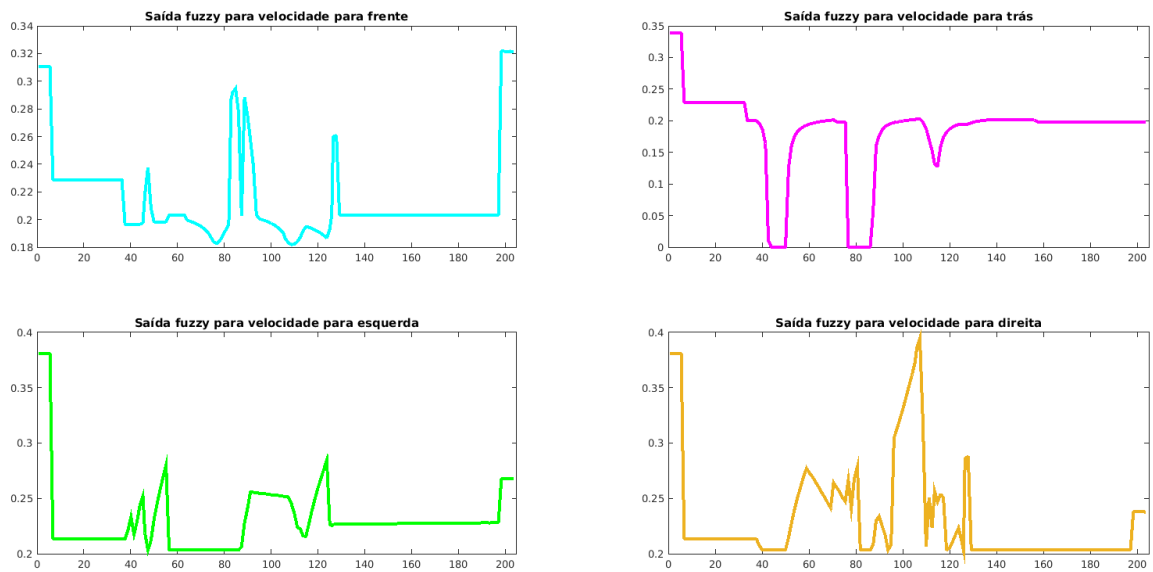
Fonte: Autor.

Além disso, tem-se as variáveis de saída do controlador *fuzzy*, e conforme mostrado na Figura 38, para a variável proporcional a velocidade para frente, é mostrado alguns picos que representam os momentos em que o robô percorreu trechos com a parede a frente mas a uma certa distância. Para a variável associada a velocidade de ré, existem três momentos em que a velocidade deveria decrescer, estes momentos são correspondentes aos três momentos em que o robô executa uma manobra para virar.

Ainda na Figura 38, as variáveis associadas a velocidade rotacional para a esquerda e para a direita apresentam algumas oscilações durante o percurso, isto se deve ao fato de tratar-se de um percurso próximo a parede de ambos os lados, ou seja, nos momentos em que o robô poderia virar para uma direção segura aconteciam os picos.



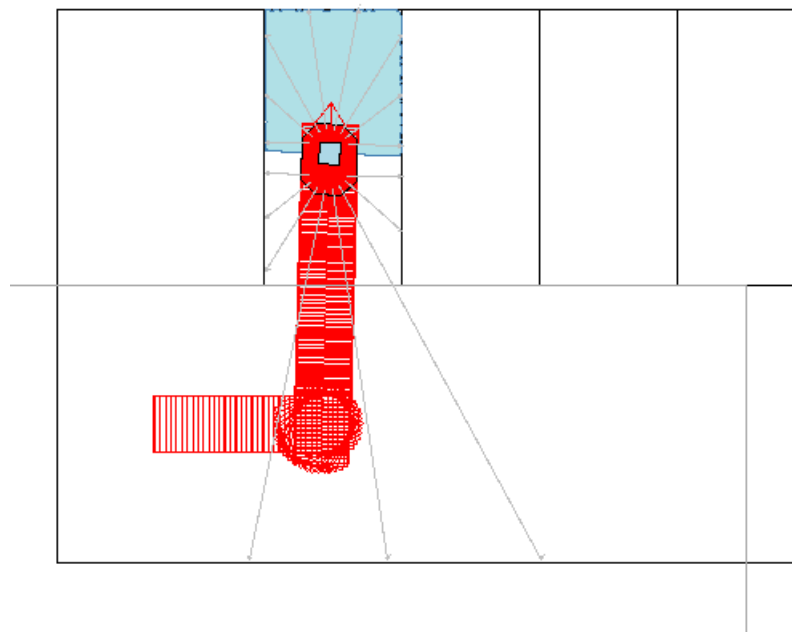
Figura 38. Saídas do controlador *fuzzy* para o percurso 04.



Fonte: Autor.

O percurso 05, mostrado na Figura 39, assim como o percurso 04, representa um desafio para o usuário da cadeira de rodas. Conforme mostrado na Figura 39, o usuário tem como objetivo de passar por uma porta, sendo essa uma situação desafiadora uma vez que a aproximação e travessia de passagens estreitas, com larguras próximas da largura do robô, necessitam de diversos ajustes de controle por parte do usuário.

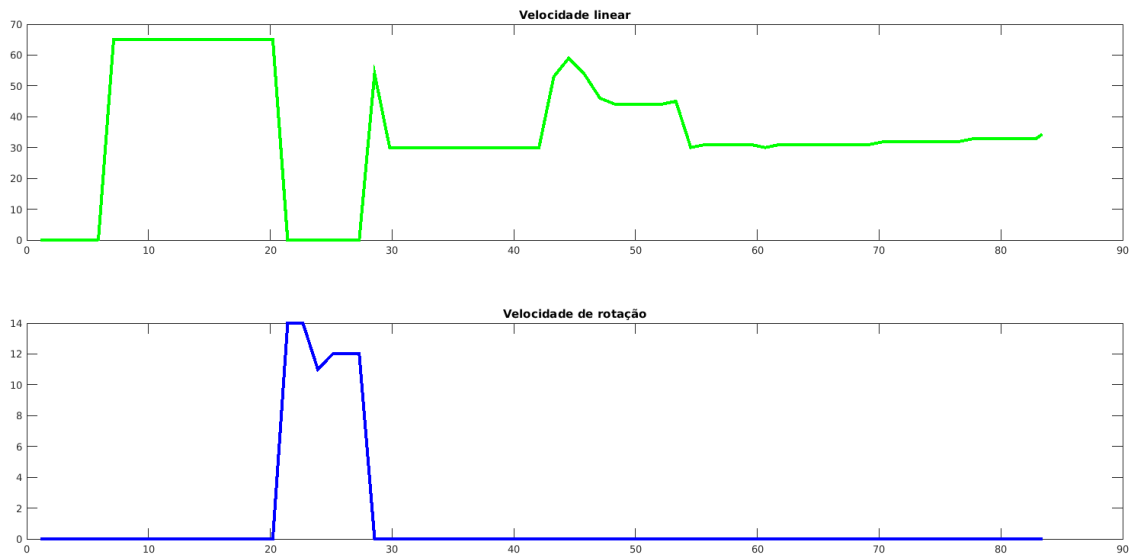
Figura 39. Percurso 05 – Navegação por uma porta.



Fonte: Autor.

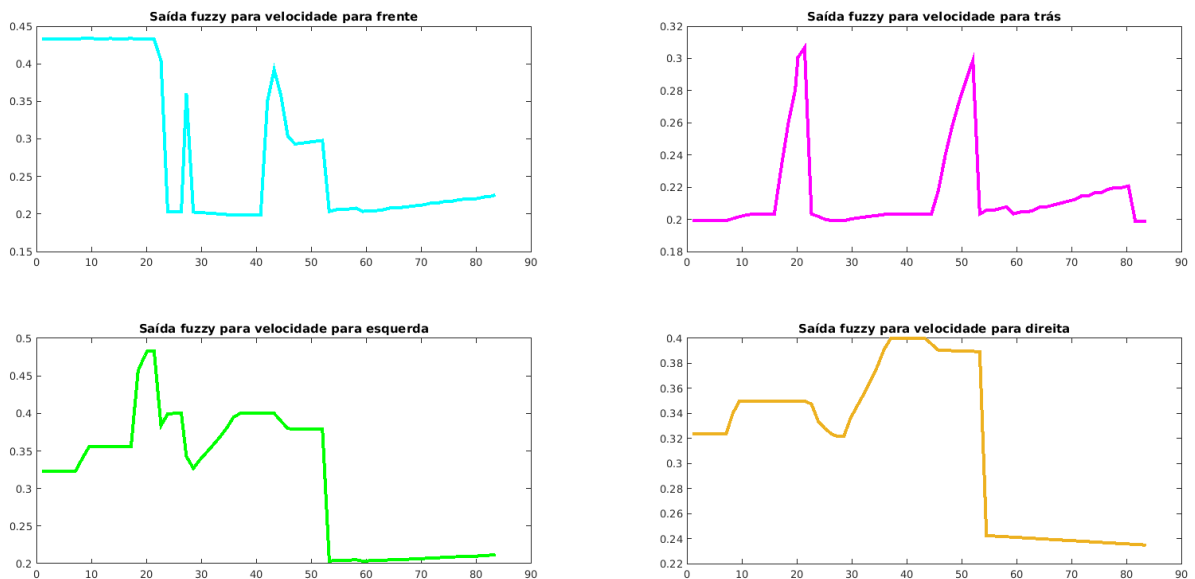
Através do perfil de velocidades, mostrado na Figura 40, é possível observar que enquanto o robô está navegando no espaço aberto, sua velocidade se torna mais alta. No momento em que a velocidade rotacional é acionada, percebe-se que o robô se virou em direção a porta e a partir deste momento, a velocidade linear decresce significativamente e não volta a aumentar devido ao fato de ter um corredor logo após a porta. A Figura 41 mostra as saídas do controlador *fuzzy* para a dada situação, que compuseram as velocidades da Figura 40.

Figura 40. Velocidades linear e angular para o percurso 05.



Fonte: Autor.

Figura 41. Saídas do controlador *fuzzy* para o percurso 05.

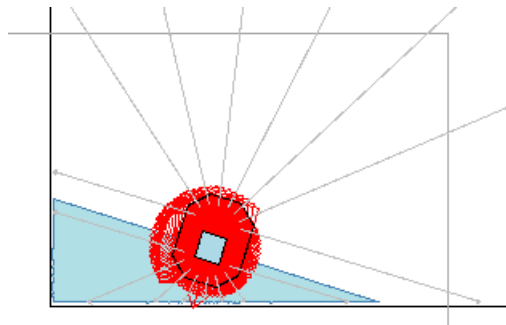


Fonte: Autor.

### 4.3 SITUAÇÕES DE RISCO

O percurso 06, mostrado na Figura 42, é uma manobra em que é desejado virar em direção a parede. Diversas situações de risco como esta, em que o usuário deseja deliberadamente se aproximar de obstáculos, como janelas e mesas, ou seja, locais cujo acesso envolvem a possibilidade de colisão, também devem ser permitidas ao usuário em consideração à sua liberdade de navegação, porém, monitoradas e protegidas pelo controlador compartilhado.

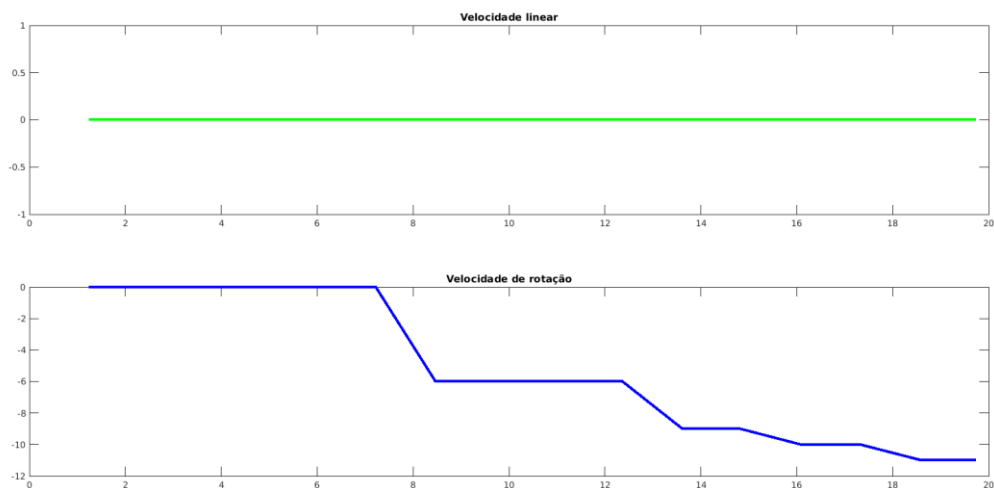
Figura 42. Percurso 06 – Navegação para virar em direção a parede.



Fonte: Autor.

Conforme se mostra na Figura 43 para as velocidades, observa-se uma velocidade linear nula, conforme o esperado, e uma velocidade de rotação que decresce a medida que o robô vai se virando em direção a parede. Neste caso, pode-se ilustrar, portanto, que mesmo com a proximidade do robô da parede, ainda sim é permitido o movimento, mesmo sendo considerado um movimento arriscado. Isto torna possível que o controle compartilhado seja eficaz, uma vez que realiza as vontades do usuário de forma segura.

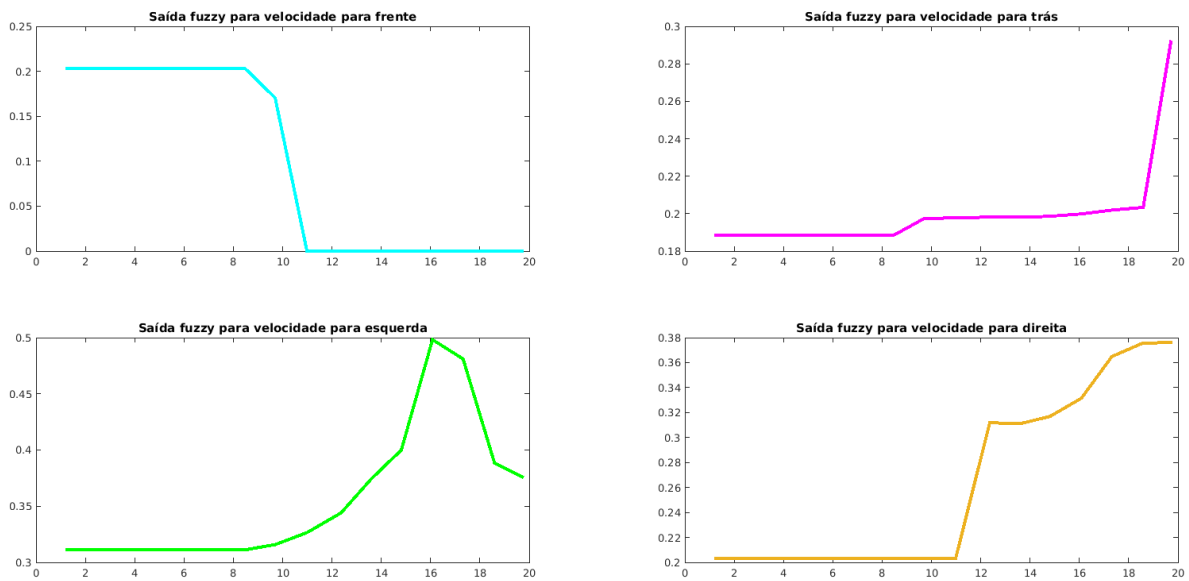
Figura 43. Velocidades linear e angular para o percurso 06.



Fonte: Autor.

Tem-se na Figura 44 o perfil de saídas do controlador *fuzzy*, na qual se observa que a velocidade para frente é zero a partir de um determinado momento, ou seja, o controlador está evitando a colisão já que o robô se encontra muito próximo a parede. As variáveis de giro para direita e para esquerda são baixas até o momento em que o robô se encontra de frente para a parede, neste caso a velocidade para a direita ou para a esquerda poderia aumentar para que fosse possível se afastar do obstáculo.

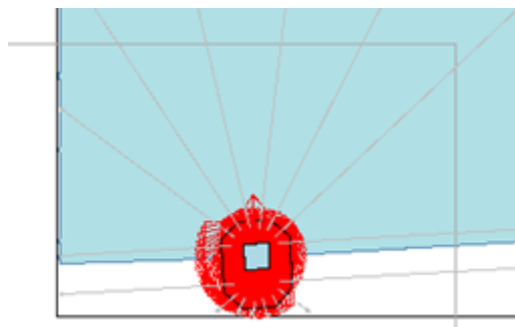
Figura 44. Saídas do controlador *fuzzy* para o percurso 06.



Fonte: Autor.

Já no percurso 07, mostrado na Figura 45, tem-se a situação complementar ao percurso 06, já que neste caso o robô está virando na direção contrária a parede, saindo da situação de perigo para se voltar a um local seguro, ou seja, livre de obstáculos.

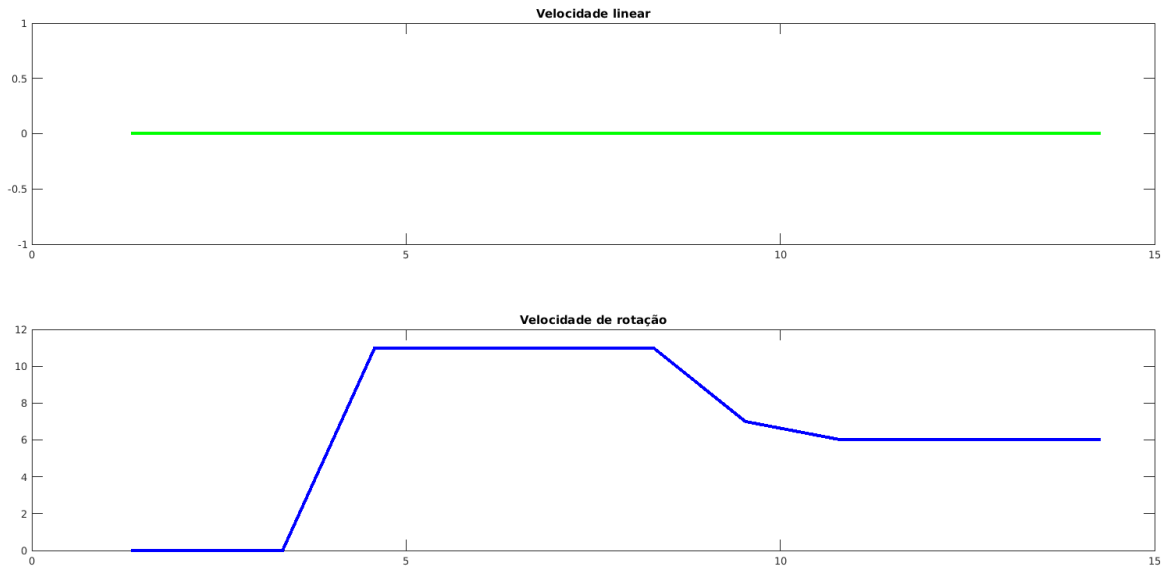
Figura 45. Percurso 07 – Navegação para virar em direção contrária a parede.



Fonte: Autor.

Pode-se observar, na Figura 46, que a velocidade linear é zero, por se tratar de um movimento apenas de rotação e ainda é possível notar que a velocidade rotacional aumenta a medida que o robô tenta se afastar da parede.

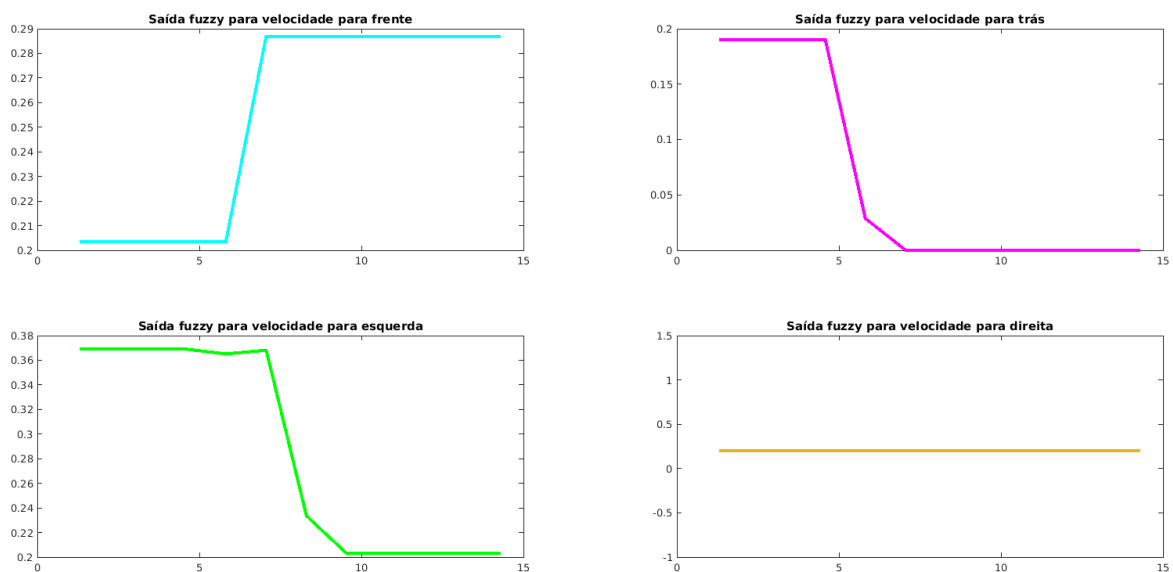
Figura 46. Velocidades linear e angular para o percurso 07.



Fonte: Autor.

As variáveis de saída do controlador *fuzzy*, mostradas na Figura 47, mostram que a medida que o robô se vira para o espaço aberto, a velocidade para frente pode aumentar enquanto a velocidade de ré deve ser zero para evitar a colisão com a parede.

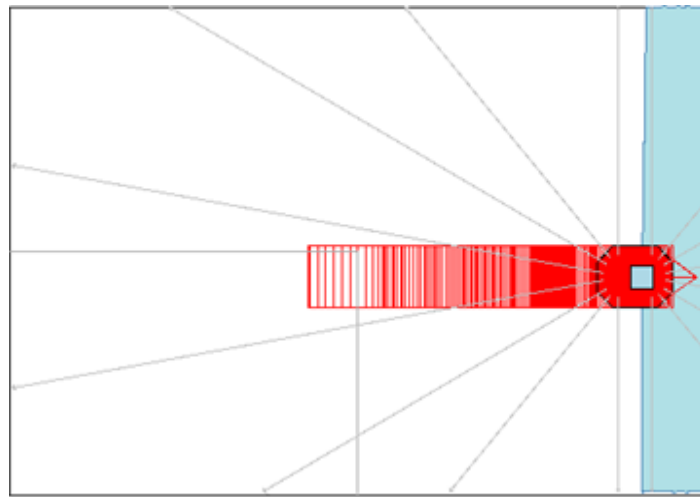
Figura 47. Saídas do controlador *fuzzy* para o percurso 07.



Fonte: Autor.

O percurso 08, mostrado na Figura 48, exemplifica o caso em que o robô se move em direção a parede, ou seja, em direção a um obstáculo podendo colidir caso não exista sem a presença de um controlador compartilhado. Este é o caso, por exemplo, em que o usuário deseja observar através de uma janela colocada na parede que ele se aproxima.

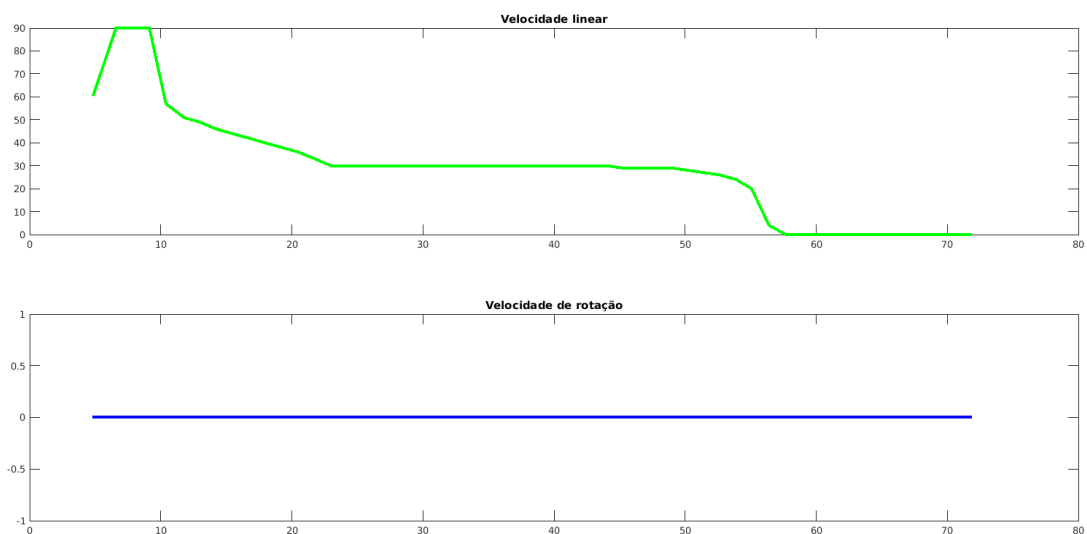
Figura 48. Percurso 08 – Navegação em direção a parede.



Fonte: Autor.

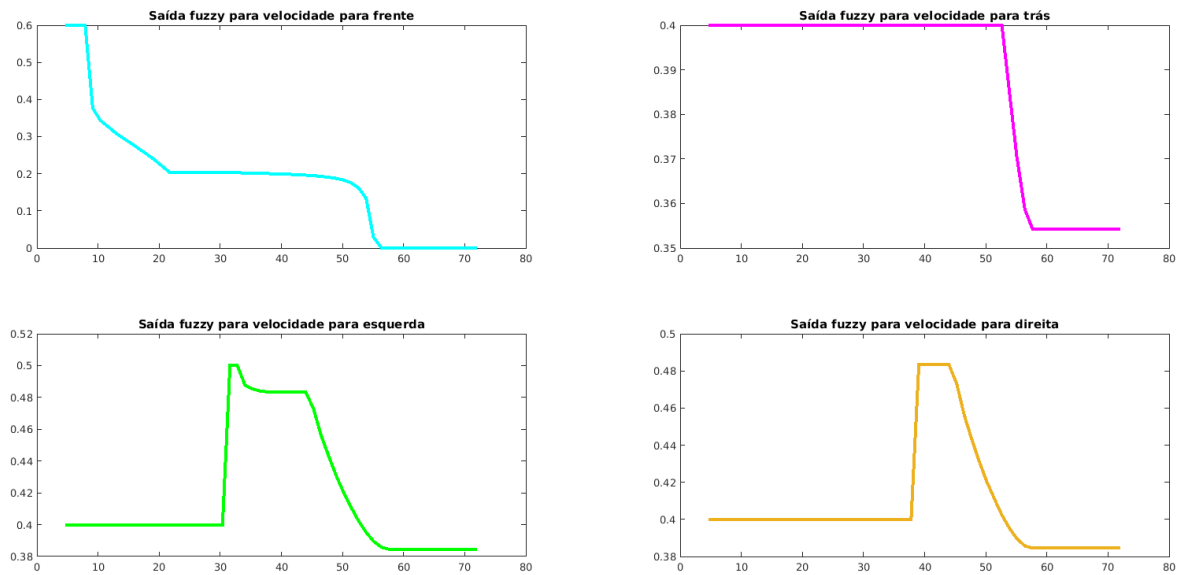
Conforme mostrado na Figura 49, a velocidade para frente é imediatamente zerada quando atinge a distância mínima permitida pelo controlador. Além disso é mostrado no prompt de comando do MATLAB uma mensagem “Risco de colisão iminente”. É possível observar nos gráficos de saída do controlador *fuzzy*, na Figura 50, que as demais variáveis assumem valores elevados afim de tornar possível o rápido afastamento do robô da situação de risco.

Figura 49. Velocidades linear e angular para o percurso 08.



Fonte: Autor.

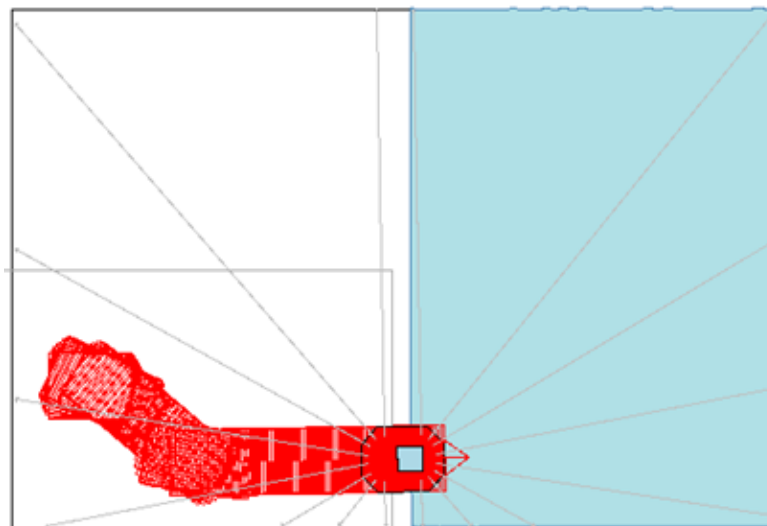
Figura 50. Saídas do controlador *fuzzy* para o percurso 08.



Fonte: Autor.

O percurso 09, na Figura 51, é uma forma de mostrar que mesmo que o robô esteja indo na direção de uma parede, é possível navegar próximo a ela a uma distância segura. Esta é mais uma situação corriqueira no dia a dia de uma pessoa usuária de cadeiras de rodas. O percurso foi feito de forma que inicialmente o robô vai em direção a parede quando então o controlador para o robô por ter atingido a distância limite permitida. É quando então é dado o comando para virar para a esquerda, no sentido de se desviar da parede, mas podendo navegar próximo a ela se esta for a vontade do usuário.

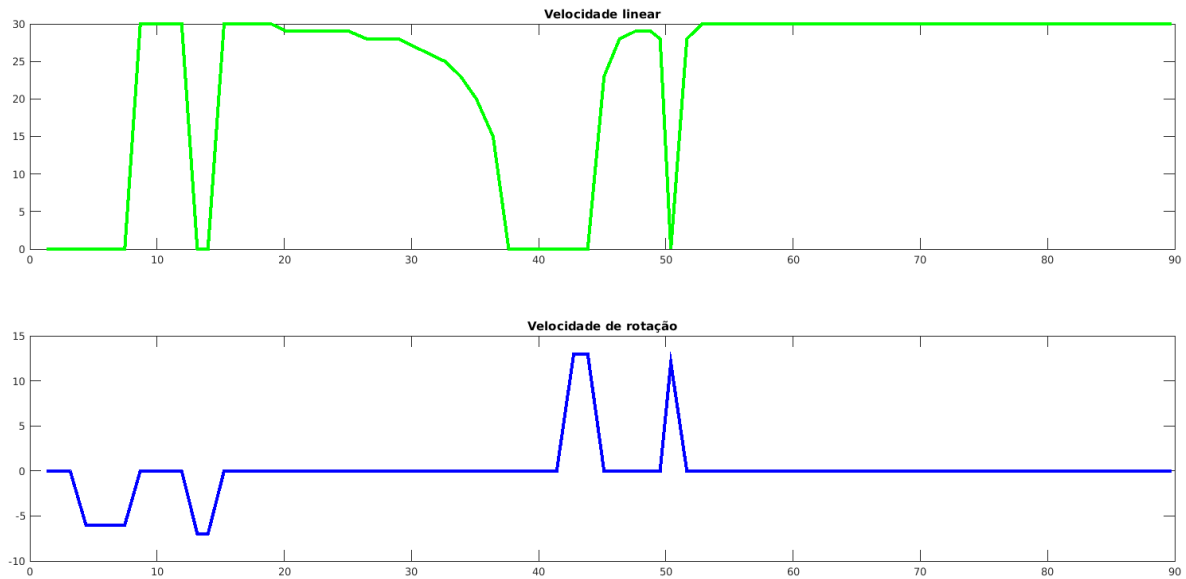
Figura 51. Percurso 09 – Navegação próximo a parede.



Fonte: Autor.

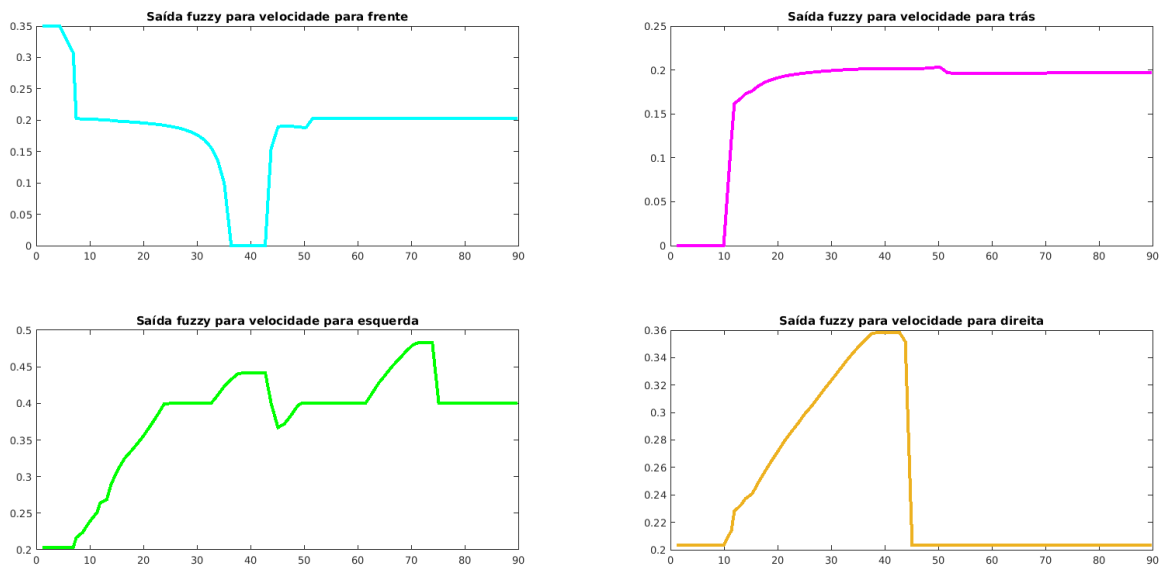
Pode-se observar na Figura 53 que a variável proporcional a velocidade de rotação para a direita é um valor muito baixo, isto porque trata-se da direção em que está a parede. A Figura 54 exibe todas as saídas do controlador *fuzzy* que são utilizadas para compor as velocidades linear e angular, conforme o direcionamento escolhido pelo usuário na navegação.

Figura 52. Velocidades linear e angular para o percurso 09.



Fonte: Autor.

Figura 53. Saídas do controlador *fuzzy* para o percurso 09.



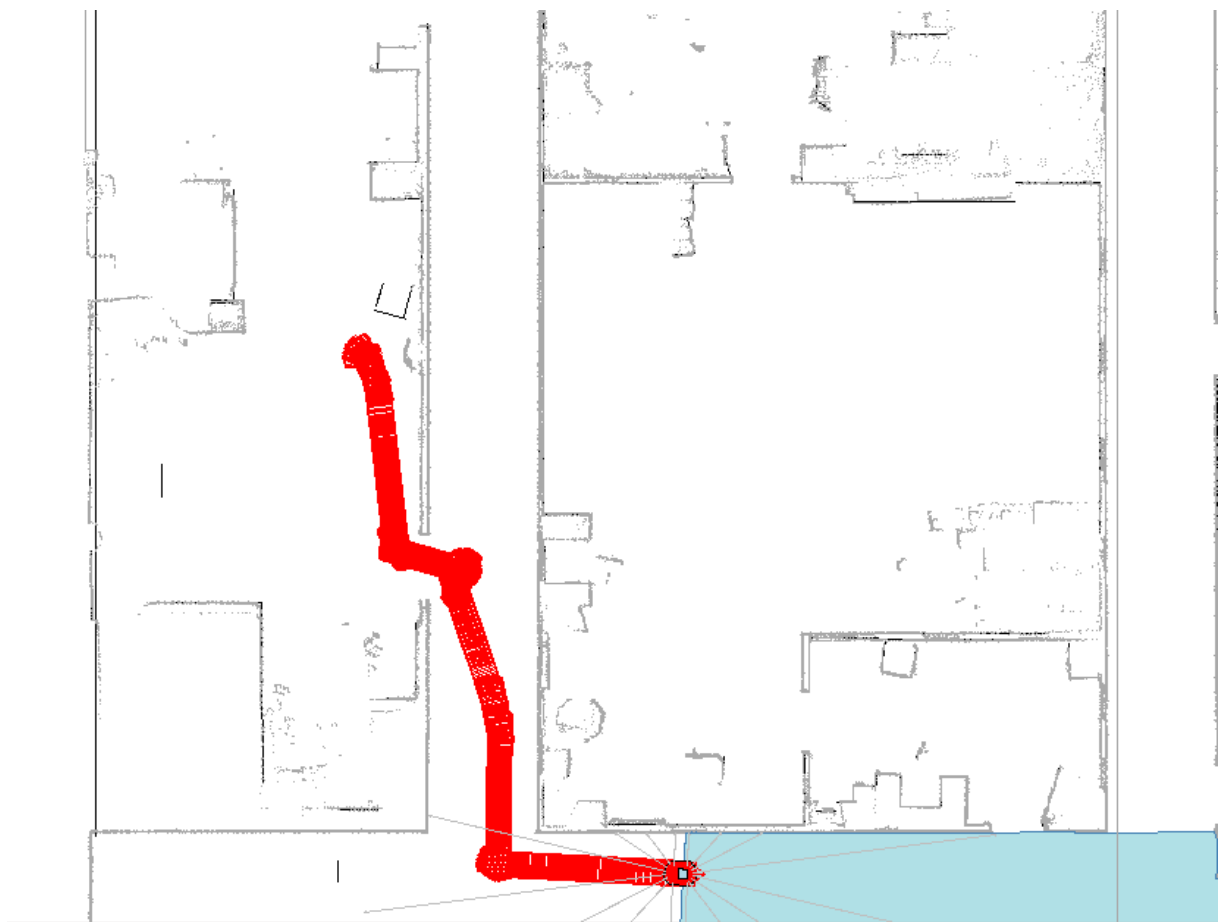
Fonte: Autor.



#### 4.4 ILUSTRAÇÃO DE SITUAÇÃO REAL

Por último temos o percurso 10, exibido na Figura 54, que mostra uma navegação completa em um ambiente grande e dispendo os desafios tratados anteriormente todos em um único local. Tem-se, primeiramente, um trecho em que o robô navega próximo a parede, quando então passa pelo desafio de passar pela porta e então entrar num corredor e virar em outro corredor.

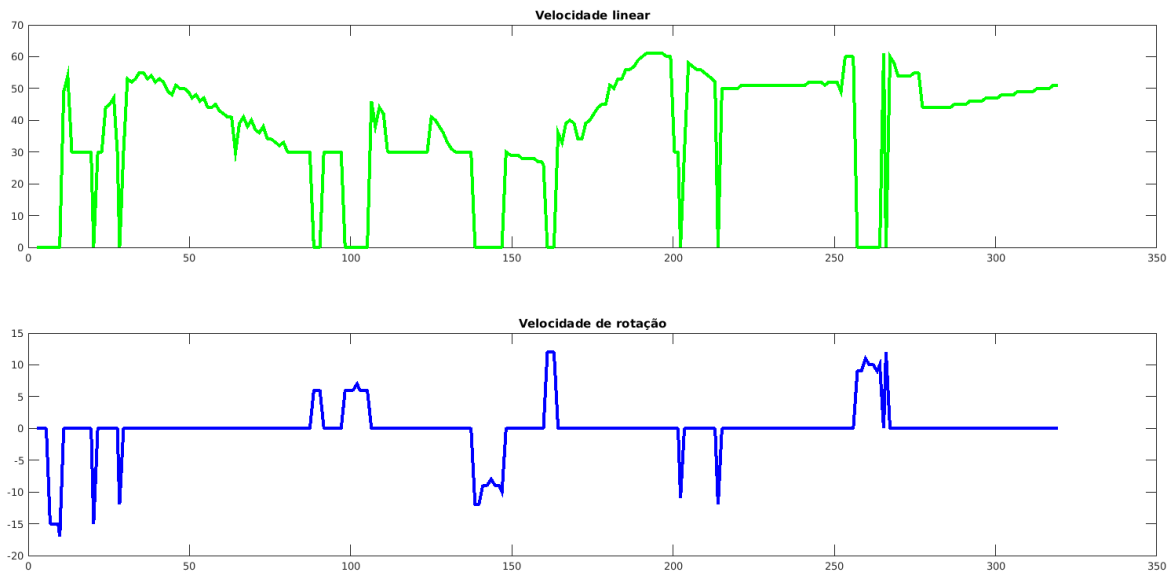
Figura 54. Percurso 10 – Navegação ambiente com obstáculos.



Fonte: Autor.

Na análise do perfil de velocidades do robô neste percurso, mostrada na Figura 55, é possível observar que primeiramente a velocidade linear vai decaindo, já que o robô está cada vez mais próximo da parede. Quando então o robô vira para passar pela porta com velocidade de rotação baixa, para garantir segurança ao passar pelo desafio. Posteriormente a passagem pela porta, a velocidade linear se mantém constante, ainda com momentos em que aparecem picos que representam comandos do usuário para melhor direcionar o robô no corredor.

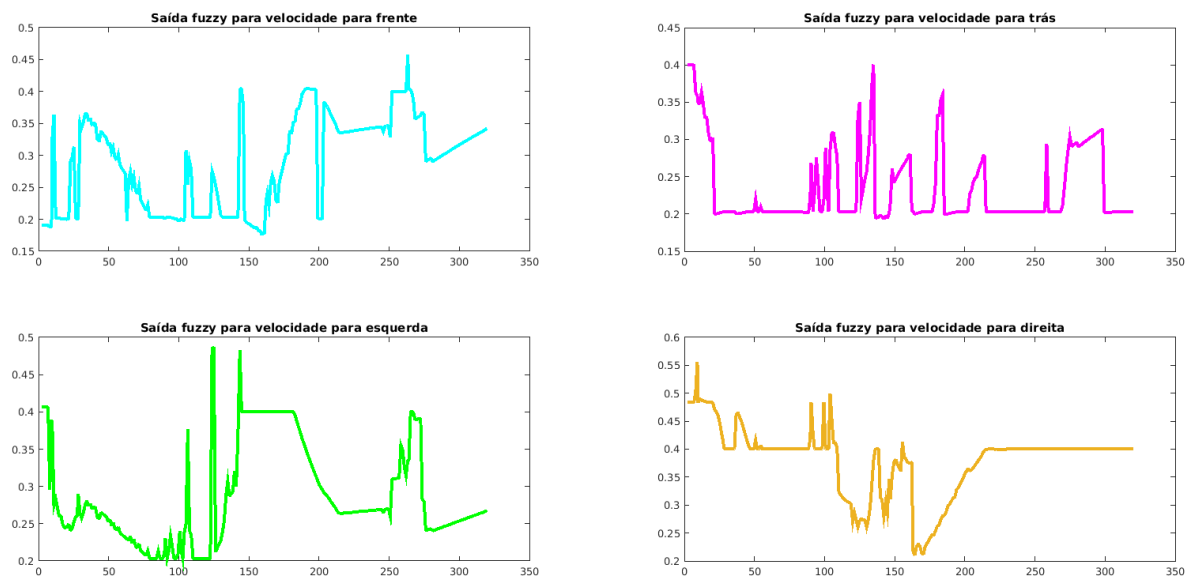
Figura 55. Velocidades linear e angular para o percurso 10.



Fonte: Autor.

Na Figura 56 são exibidas todas as saídas *fuzzy* que compõem as velocidades mostradas no percurso 10. As saídas *fuzzy* são calculadas com base no contexto dos obstáculos do ambiente, com relação a cada um dos estados do robô.

Figura 56. Saídas do controlador *fuzzy* para o percurso 10.



Fonte: Autor.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um controlador *fuzzy* para o uso no contexto de controle compartilhado para cadeiras de roda robóticas, tornando a navegação do usuário segura e livre de riscos de colisão. O objetivo foi ilustrar o comportamento do controlador diante dos diversos obstáculos e dificuldades encontradas ao longo dos percursos.

Em um primeiro momento foram mostrados os diversos conceitos referentes ao controle compartilhado e ao uso da robótica assistiva atualmente. Foram ilustrados exemplos acerca da aplicabilidade da robótica no auxílio das atividades nos mais diversos setores.

Posteriormente foram apresentados os conceitos referentes a lógica *fuzzy*, através de exemplos e também de um desenvolvimento teórico a respeito da formação da base de regras e inferências.

Por fim, foi apresentado o desenvolvimento técnico utilizado, demonstrando com imagens e equações todos os conceitos utilizados na montagem da base de regras bem como a escolha das variáveis de entrada e saída do controlador.

Por último foram mostrados os resultados práticos através do simulador MobileSim, onde foi possível utilizando um programa no MATLAB obter os gráficos representativos das entradas e saídas do controlador e ainda as velocidades reais do robô na navegação pelos ambientes escolhidos.

Conforme mostrado, os resultados foram satisfatórios em todos os testes executados e apresentaram o comportamento esperado pelo controlador. A base de regras foi montada considerando um amplo leque de possibilidades, cobrindo as possíveis situações de risco para o usuário.

Como sugestões de trabalhos futuros, é possível que a base de regras seja ampliada, se tornando ainda mais robusta quanto ao controle compartilhado. Portanto, aumentar o número de regras de forma a contemplar ainda mais possibilidades e tornar o controle mais adaptado aos ambientes nos quais o robô está inserido.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MAIA, M.; “A era dos robôs: tecnologia amplia produtividade, transforma educação e salva vidas”. Agência de notícias, 2017. Disponível em: <http://www.portaldaindustria.com.br/agenciacni/noticias/2017/02/a-era-dos-robos-tecnologia-amplia-productividade-transforma-educacao-e-salva-vidas/>. Acessado em 05/07/2018.
- [2] EMOTIV; “The Science behind our technology”. Disponível em: <https://www.emotiv.com/the-science/>. Acessado em 05/07/2018.
- [3] SANEI, S.; CHAMBERS, J.A.; “*EEG Signal Processing*”. Editora Wiley-Interscience, 2013.
- [4] PEDRYCZ, W.; GOMIDE, F.; “*Fuzzy Systems Engineering: Toward Human-Centric Computing*”. Editora Wiley-Blackwell, 2007.
- [5] STOELLEN, M.F.; TEJADA, V.F.; HUETE, A.J.; BONSIGNORIO, F.; BALAGUER, C.; “*Benchmarking Shared Control for Assistive Manipulators - From Controllability to the Speed-Accuracy Trade-Off*”. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.
- [6] Olivi, L.; Souza, R.; Rohmer, E.; Cardozo, E.; “*Shared Control For Assistive Mobile Robots Based On Vector Fields*”. 10th International Conference on Ubiquitous Robots and Ambient Intelligence, 2013.
- [7] LIU, R.; XUE, K.-Z.; WANG, Y.-X.; YANG, L.; “A Fuzzy-Based Shared Controller For Brain-Actuated Simulated Robotic System”. IEEE Engineering in Medicine and Biology Society, 2011.
- [8] NISHIMURA, R.; WADA, T.; SUGIYAMA, S.; “*Haptic Shared Control In Steering Operation Based On Cooperative Status Between A Driver And A Driver Assistance System*”. Journal of Human-Robot Interaction, v. 4, n. 3, p. 19-37, 2015.
- [9] SOH, H.; DEMIRIS, Y.; “Learning Assistance By Demonstration: Smart Mobility With Shared Control And Paired Haptic Controllers”. Journal of Human-Robot Interaction, v. 4, n. 3, p. 76-100, 2015.
- [10] GOMIDE, F.; GUDWIN, R.; TANSCHHEIT, R.; “*Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações*”. International Fuzzy Systems Association World Congress, 1995.
- [11] KASABOV, N.K.; “*Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*”. Editora The MIT Press, 1996.
- [12] GOMIDE, F.; GUDWIN, R.; “*Modelagem, Controle, Sistemas E Lógica Fuzzy*”. SBA Controle & Automação, v. 4, n. 3, p. 97-115, 1994.
- [13] LEGASPE, E.P.; “*Controlador fuzzy de código aberto para uso em controladores programáveis*”. Dissertação de Mestrado, Universidade de São Paulo, 2012.

[14] MAMDANI, E.H.; ASSILIAN, S.; “An Experiment In Linguistic Synthesis With A Fuzzy Logic Controller”. International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

[15] 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.

[16] SICK, “LMS200”, Disponível em: <https://www.sick.com/br/pt/solucoes-de-medicao-e-deteccao/sensores-2d-lidar/lms2xx/lms200-30106/p/p109843>. Acessado em 05/07/2018.

[17] MOBILEROBOTS, “Pioneer3DX”. Disponível em: <http://www.mobilerobots.com>. Acessado em 05/07/2018.