

UNIVERSIDADE FEDERAL FLUMINENSE
POLO UNIVERSITÁRIO DE RIO DAS OSTRAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Victor Santos Brito

Desenvolvimento de Sistemas Inteligentes para Robôs Lego®.

Rio das Ostras - RJ

2018

VICTOR SANTOS BRITO

DESENVOLVIMENTO DE SISTEMAS INTELIGENTES PARA ROBÔS LEGO®.

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel. Área de Concentração: Inteligência Artificial e Robótica.

Orientadores: Prof. Dr. PATRICK BARBOSA MORATORI e Prof. Dr. SERGIO CRESPO

Rio das Ostras-RJ

2018

VICTOR SANTOS BRITO

DESENVOLVIMENTO DE SISTEMAS INTELIGENTES PARA ROBÔS LEGO®.

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel. Área de Concentração: Inteligencia Artificial.

Aprovada em JULHO de 2018.

BANCA EXAMINADORA

Prof. Dr. SERGIO CRESPO - Co-orientador
Universidade Federal Fluminense

Prof. Dr. CARLOS BAZILIO MARTINS
Universidade Federal Fluminense

Prof. Dr. DALESSANDRO VIANA
Universidade Federal Fluminense

Rio das Ostras-RJ
2018

À meus pais, minha tia, minha namorada e a todos aqueles que já me deixaram.

“Sorte ou azar, eu não sei. Só sei que foi assim.”

Agradecimentos

Primeiramente, agradeço a minha mãe, Ivanise, meu pai, Nelson e a minha tia, Lourdes, por todo amor incondicional e apoio irrestrito.

À meu tio, Walkírio e minha avó, Issandra, que já me deixaram, mas que foram cruciais para meu crescimento, não somente acadêmico, mas também pessoal.

À minha namorada, Thais, por todo amor, carinho e compreensão. Sem suas insistências em me fazer ir para a frente, este trabalho jamais teria sido concluído.

À meus professores, em especial ao professor Patrick Moratori, que me orientou durante este trabalho e durante minha monitoria, a professora Flavia Bernardini, que me orientou durante minha iniciação científica e ao professor Sergio Crespo, que me orientou na reta final deste trabalho.

À Mulekadinha, amigos que fiz durante a faculdade e que agora me acompanham na vida após ela.

E a todos que, de alguma forma, contribuÃram para o desenvolvimento deste trabalho.

Lista de Figuras

2.1	Robô - Visão geral	6
2.2	Robô - Módulo NXT	6
2.3	Robô - Parte de baixo	7
2.4	Robô - Servo motor	7
2.5	Robô - Sensores	7
2.6	Diagrama de bloco de comunicação entre computador e bloco inteligente	9
3.1	Frenagem - Overview	11
3.2	Frenagem - Conjuntos de entrada	11
3.3	Frenagem - Conjunto de Saída Velocidade	12
4.1	Seguidor de Linha - Circuito	16
4.2	Seguidor de Linha 1 - Overview	16
4.3	Seguidor de Linha 1 - Conjuntos de entrada	17
4.4	Seguidor de Linha 1 - Conjuntos de saída	18
4.5	Seguidor de Linha 2 - Overview	18
4.6	Seguidor de Linha 2 - Conjuntos de entrada	19
4.7	Seguidor de Linha 2 - Conjuntos de saída	20
5.1	Desvio - Overview	23
5.2	Desvio - Conjuntos de Entrada	23
5.3	Desvio - Conjuntos de Saída	24

Lista de Tabelas

2.1	Métodos	5
3.1	Frenagem - Resultados estatísticos	14
4.1	Seguidor de Linha - Dados ANFIS	21
5.1	Desvio - Estatísticas 30 Centímetros	24
5.2	Desvio - Estatísticas 50 Centímetros	24
D.1	Frenagem - 150 Centímetros	43
D.2	Frenagem - 100 Centímetros	45
D.3	Frenagem - 80 Centímetros	46
D.4	Frenagem - 50 Centímetros	47
D.5	Frenagem - 30 Centímetros	48
E.1	Seguidor de Linha - Sistema 1	49
E.2	Seguidor de Linha - Sistema 2	52
E.3	Seguidor de Linha - Sistema ANFIS	55
F.1	Desvio - Distância Inicial 30 centímetros	58
F.2	Desvio - Distância Inicial 50 centímetros	58

Sumário

Agradecimentos	v
Lista de Figuras	vi
Lista de Tabelas	vii
Resumo	x
Abstract	xi
1 Introdução	1
1.1 Motivação	1
1.2 Objetivo	2
1.3 Revisão da Literatura	2
1.4 Organização do Texto	3
2 Tecnologias Utilizadas	4
2.1 Lógica Nebulosa	4
2.2 Lego® Mindstorms NXT	5
2.3 Softwares Utilizados	8
3 Cenário 1- Frenagem	10
3.1 Descrição do Problema	10
3.2 Sistema Desenvolvido	10
3.3 Resultados Obtidos	13
4 Cenário 2 - Seguidor de Linha	15
4.1 Descrição do Problema	15
4.2 Sistema Desenvolvido	16
4.2.1 Sistema 1	16
4.2.2 Sistema 2	18
4.2.3 Sistema ANFIS	20
4.3 Resultados Obtidos	20

5	Cenário 3 - Desvio de Obstáculos	22
5.1	Descrição do Problema	22
5.2	Sistema Desenvolvido	22
5.3	Resultados Obtidos	24
6	Conclusão	26
7	Trabalhos Futuros	28
A	Sistema de Frenagem	29
B	Sistema de Seguidor de linha	31
B.1	Sistema 1	31
B.2	Sistema 2	33
B.3	Sistema AnfisEsq	35
B.4	Sistema AnfisDir	36
C	Sistema de Desvio	39
D	Resultados Sistema de Frenagem	42
E	Resultados Seguidor de Linha	49
F	Resultados Sistemas de Desvio	58

Resumo

Redes neurais e sistemas nebulosos tem sido usados para criação de sistemas inteligentes que melhoram o dia-a-dia da sociedade, identificando e resolvendo falhas de funcionamento nas mais diversas áreas e ferramentas. Como a robótica pode ser descrita como uma área interdisciplinar, que visa a automação de processos repetitivos, a associação com técnicas de inteligência artificial é inevitável.

Embora tenha ênfase maior na educação tecnológica infantil, a linha de robôs Mindstorm da LEGO® possui uma praticidade que faz com que os modelos que a compõem sejam ferramentas versáteis em outras áreas de pesquisa. A interação com a plataforma MATLAB® permite a criação de sistemas inteligentes e testes práticos dos mesmos de maneira rápida e fácil.

Este trabalho tem como objetivo o desenvolvimento de sistemas inteligentes para robôs LEGO® modelo NXT, utilizando a plataforma MATLAB®. Tais sistemas podem servir de base para sistemas implementados em outros mecanismos autômatos com as mais variadas finalidades, como drones e robôs, além de oferecerem uma visão sobre questões de segurança, praticidade e efetividade dos métodos implementados.

Serão aplicados três cenários de teste para verificar a eficácia dos sistemas desenvolvidos. Nestes cenários, o robô deverá desviar de obstáculos, seguir trajetos determinados por uma linha e frear da melhor maneira. Também será avaliado a eficácia de diferentes técnicas de inteligência artificial no desenvolvimento dos sistemas.

Abstract

Neural networks and fuzzy systems have been used in the creation of intelligent systems to improve society's, identifying and solving functioning flaws on a bunch of areas and tools. As robotics may be described as a interdisciplinary area, an association between robotics and artificial inteligency is inevitable.

This work has, as main goal, intention of developing intelligent systems for LEGO® NXT robots, using MATLAB software. Those systems may be used as base to other systems created for the more farreaching mechanism, like drones and robots as well. Will be applied, three different scenarios where the robot will have to follow a line on the ground, bypass an obstacle or just stop as close as possible to it.

Capítulo 1

Introdução

1.1 Motivação

Nas grandes indústrias, robôs já executam tarefas como a soldagem de peças, montagem de componentes, limpeza de partes, etc. No dia-a-dia, é cada vez mais comum ver tarefas sendo automatizadas. Entretanto, tais tarefas, embora importantes, são sequenciais, repetitivas e não apresentam um desafio para a área de inteligência artificial. Na verdade, estes robôs podem ser vistos como um desperdício de potencial, quando se imagina o que seriam capazes de fazer se houvesse um sistema inteligente por trás.

Hoje em dia, robôs móveis, controlados por sistemas inteligentes, já são capazes de realizar tarefas que vão desde o transporte de peças até a exploração de locais perigosos como crateras de vulcões e até mesmo outros planetas[9]). Estes robôs não são um desperdício de potencial, e sim o futuro das indústrias, embora seja verdade que o desafio é muito maior.

Atualmente, os automóveis podem ser vistos como computadores de alta performance, capazes de estacionar sozinhos, antecipar possíveis colisões, manter a velocidade desejada, dentre outras tecnologias menos populares. Como em quase todas as áreas, o próximo passo na evolução é a eliminação do fator humano, o que neste caso seriam os veículos autônomos, capazes de levar e buscar o passageiro no seu destino. [5] cita uma pesquisa do MIT¹ que mostrou que é possível levar cada passageiro ao seu destino, dentro do tempo necessário, com 80% menos carros, apenas dividindo a viagem. A automatização dos veículos contribuiria para fazer do cenário descrito na pesquisa, uma realidade. Além disso, veículos autônomos não se perdem, criam menos engarrafamentos, tem um menor tempo de viagem e são mais seguros, o que contribui para um aumento na qualidade de vida dos cidadãos e redução de custos governamentais com rodovias [5].

Podemos citar os veículos autônomos e os robôs inteligentes como o futuro da robótica e da inteligência artificial, e é este futuro que motivou este trabalho.

¹Massachusetts Institute of Technology

1.2 Objetivo

Este trabalho tem por objetivo a geração de três sistemas inteligentes para robôs Lego NXT. Visando analisar, através de uma abordagem prática, o desenvolvimento destes sistemas, este trabalho procurará fazer um breve estudo sobre a experiência de desenvolvimento.

Os cenários escolhidos são cenários comuns em competições de robótica, como a OBR [6].

Os sistemas nebulosos serão capazes de fazer com que um robô LEGO@NXT execute determinadas tarefas, como seguir uma linha, desviar de obstáculos e frear suavemente. Além disso, este trabalho visa analisar a efetividade e a praticidade das ferramentas no desenvolvimento da ideia.

Cada sistema será testado em um cenário desenvolvido para avaliar a performance do mesmo na tentativa de alcançar seu objetivo. Posteriormente, um outro sistema, com o mesmo objetivo e gerado a partir de um sistema híbrido de redes neurais com lógica nebulosa, fará o mesmo percurso, sob as mesmas condições, de maneira que seja possível comparar os resultados. Este segundo sistema será gerado a partir dos resultados obtidos no primeiro sistema.

O objetivo é comparar os resultados obtidos pelos sistemas desenvolvidos quando comparados com o sistema gerado automaticamente. Desta maneira, é possível analisar quem teve um desempenho melhor. Além disso, este trabalho tem como segundo objetivo, verificar o grau de dificuldade de desenvolvimento, aprimoramento e manutenção dos sistemas gerados utilizando as tecnologias escolhidas, visando entender os principais problemas desta abordagem.

1.3 Revisão da Literatura

A robótica é uma realidade em diversas áreas. Seu impacto é sempre alvo de trabalhos acadêmicos dos mais variados. Hoje em dia, existem estudo do uso da robótica nas mais diversas áreas, como na medicina [18], no desenvolvimento social infantil [21] ou no deslocamento urbano[5] .

A criação de sistemas inteligentes para robôs NXT também é alvo constante de trabalhos acadêmicos. Desde artigos de iniciação científica [17] [3] até projetos de mestrado [7], o tema é explorado de diversas maneiras.

Isso evidencia que a junção da robótica com a inteligência artificial é um caminho natural a ser explorado e também mostra que a utilização de robôs NXT é uma via interessante aos olhos do mundo acadêmico. A questão, então, é o que este trabalho pode agregar de novo.

A diferença entre este trabalho e alguns dos outros publicados é o uso do MATLAB em sua totalidade. Simonsen [17] e Benevides [3] desenvolveram sistemas nebulosos no MATLAB, mas usaram o software LabVIEW para a comunicação entre robô e computador. O LabVIEW permite a programação a partir de ícones, que representam funções, variáveis e etc. e possui interação com o toolbox de lógica nebulosa do MATLAB. Além disso, nenhum trabalho acadêmico lido faz uso de um sistema gerado utilizando redes neurais, como este.

Em alguns casos, o MATLAB não é usado em momento algum, optando por utilizar linguagens de programação mais populares, como Java. Uma abordagem vista algumas vezes foi a utilização da linguagem Java junto do LeJOS, uma máquina virtual Java que pode ser instalada no controlador do

NXT [22]. Em comparação com este trabalho, esta abordagem pode se mostrar desvantajosa, uma vez que os softwares utilizados por este trabalho apresentam uma performance computacional melhor ao fazer o processamento no computador [16]. Vale ressaltar que nestes casos não houve uso de sistemas inteligentes.

Vale ressaltar que lógica nebulosa data da década de 60, com o artigo de Zadeh[23]. Além disso, foi lido um artigo da década de 90 [8] a respeito do uso de sistemas inteligentes em robôs, que já citava artigos da década de 80 também sobre o assunto, o que evidencia à quanto tempo este problema é discutido no meio acadêmico.

1.4 Organização do Texto

Este texto esta organizado da seguinte maneira: o capítulo 2 possui uma explicação para as tecnologias usadas neste trabalho. Nos capítulos 3, 4 e 5 estão explicitados o cenário, sistema desenvolvido e resultados obtidos para cada problema. Em seguida, vem a conclusão deste trabalho no capítulo 6, onde existe uma análise de tudo que foi dito neste trabalho. Após a conclusão, vem os futuros trabalhos, os quais se intenciona realizar, no capítulo 7.

Capítulo 2

Tecnologias Utilizadas

2.1 Lógica Nebulosa

Na opinião de Pawlikowski[15] “[...] lógica nebulosa representa mais fielmente o pensamento humano do que os fundamentos da lógica clássica. A lógica nebulosa permite o usuário descrever o sistema usando palavras naturais, e criar um mecanismo de controle baseado em “regras de expert“ ou, em outras palavras, regras desenvolvidas por experiência“¹. Em outras palavras, a lógica nebulosa é a lógica que trabalha com a incerteza, com o aspecto vago da informação. Nesta lógica, um determinado valor possui um grau de pertinência em um determinado conjunto. Por exemplo, na lógica tradicional, um ser humano com idade entre 13 e 17 anos pode ser considerado um adolescente, mas um ser humano com 12 anos e 11 meses não. Na lógica nebulosa, este ser humano teria um grau de pertinência no conjunto de pessoas "adolescente". Além disso, um ser humano com 13 anos poderia ter um grau de pertinência no conjunto "adolescente" menor que um ser humano com 16 anos, enquanto que na lógica tradicional ambos seriam adolescentes sem distinção.

Isso tudo é permitido pelo uso de conjuntos nebulosos. Zadeh[23] define um conjunto nebuloso como sendo “uma classe de objetos com grau de pertinência a um conjunto“².

Por este motivo, a lógica nebulosa se apresenta como uma ótima solução para problemas em que a lógica tradicional não é satisfatória.

Um sistema do tipo Mamdani é um sistema que possui um controlador *fuzzy* baseado no trabalho de Mamdani[12], em que a ação do controlador é definida pela agregação das regras que se aplicam, dada a entrada. Então, esta aplicação define uma saída, que é um conjunto de saída baseado nos conjuntos de saída definidos pelo usuário. Uma função de defuzzificação é aplicada no conjunto de saída, resultando no valor efetivo de saída[2].

Um sistema do tipo Sugeno é um sistema que possui um controlador *fuzzy* baseado no trabalho de Takagi e Sugeno[19] e pode ser visto como uma simplificação do sistema Mamdani. Neste sistema, as

¹ “[...] *fuzzy* logic more closely represents human thought than the fundamentals of classical control. *fuzzy* logic allows a user to describe a system using natural words, and create a control mechanism based on ?expert rules? or in other words, rules developed through experience.“

² “A *fuzzy* set is a class of objects with a continuum of grades of membership.“

regras resultam em valores numéricos. Numa simplificação, pode-se definir uma regra como sendo: se x é A e y é B , então $z = f(x, y)$ [2]. A saída do sistema é uma média ponderada das respostas obtidas por z .

Todos os sistemas desenvolvidos foram do tipo Mamdani[12], com exceção do sistema gerado pelo ANFIS, que é do tipo Sugeno[19]. O ANFIS é um toolbox para MATLAB que será melhor explicado na seção 2.3.

Tabela 2.1: Métodos

Conjunção	'min'
Disjunção	'max'
Implicação	'min'
Agregação	'max'
Defuzzificação	'centróide'

Como os sistemas desenvolvidos pelo autor foram do tipo Mamdani, as funções de conjunção, disjunção, implicação e agregação escolhidas, assim como a função de defuzzificação, estão explicitadas na tabela 2.1. No caso dos sistemas gerados pelo *ANFIS*, o controlador era um controlador Sugeno. Os métodos escolhidos estão explicitados em suas seções.

2.2 Lego® Mindstorms NXT

Iniciado em 1998, o projeto LEGO Mindstorms é tido como o melhor em vendas da empresa [10]. A linha NXT foi lançada em 2006 e hoje em dia, a versão mais atual é a EV3, lançada em 2013. Sua maior utilização é na educação de jovens, facilitando a entrada deles no mundo da programação. Porém, por se tratar de um robô de fácil uso, também é utilizado em pesquisas acadêmicas, como neste trabalho, em [22] ou nos trabalhos de iniciação científica da PUC-RJ dos alunos Gabriel Simonsen [17] e Leonardo Benevides [3]

Atualmente, o modelo NXT 2.0 não é mais vendido pelo site oficial da LEGO [11]. Entretanto, o kit completo pode ser encontrado na Amazon por R\$1.902,22³[1]. Este kit é a versão completa, com todos os sensores e três servo motores.

Neste trabalho, o robô é composto pelos seguintes componentes: um bloco inteligente, que se comunica com o computador; dois sensores de ultrassom, para medir a distância entre o robô e os obstáculos; um sensor de cor, usado para fazer o robô seguir a linha que determina o trajeto; dois motores, responsáveis pela movimentação do robô.

A figura 2.2 é uma foto do bloco de controle do NXT. É para ele que são enviados os comandos e é ele quem envia para o computador as leituras dos sensores.

A figura 2.3 mostra uma foto da parte de baixo do robô, mostrando a estrutura montada para manter os servos motores fixos no módulo de controle e uma roda auxiliar, centralizada para ajudar no equilíbrio do robô, além da quantidade de pilhas necessárias para fazer o robô funcionar. Em versões

³\$509,98. Dólar a R\$3,73

Figura 2.1: Robô - Visão geral



Figura 2.2: Robô - Módulo NXT



mais atuais do projeto Mindstorm, os robôs vem com uma bateria recarregável no lugar da pilha.

A figura 2.4 é uma foto do servo motor, enquanto a figura 2.5 é uma foto da parte da frente do robô, mostrando a estrutura montada para posicionar os sensores.

Figura 2.3: Robô - Parte de baixo

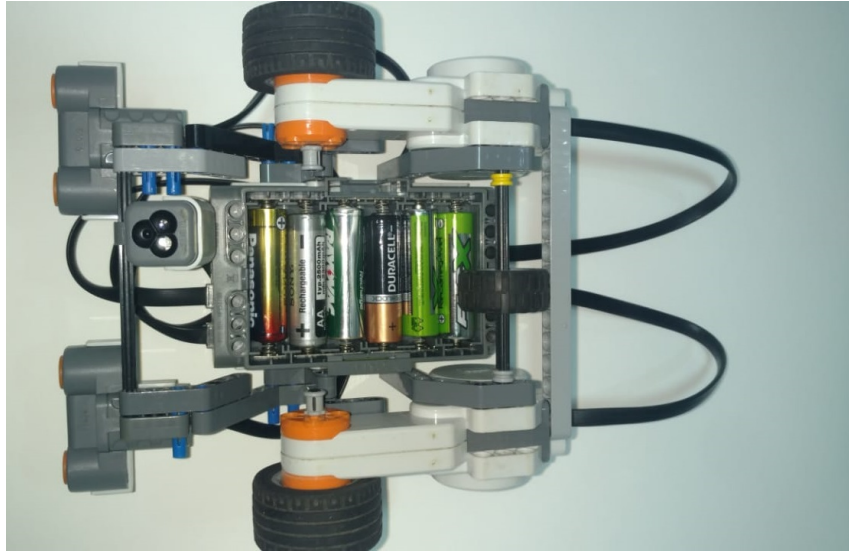


Figura 2.4: Robô - Servo motor

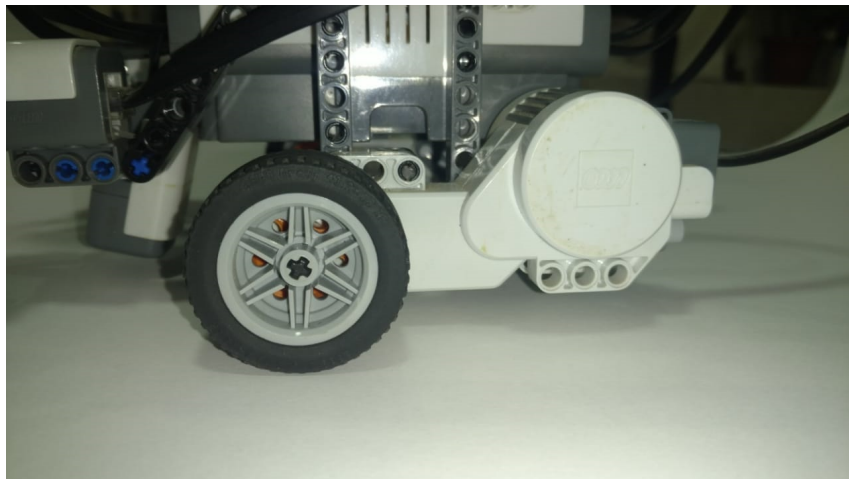
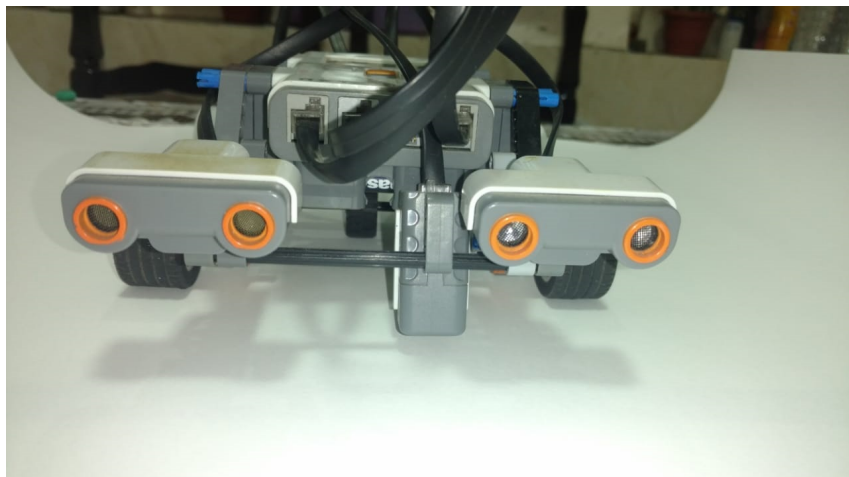


Figura 2.5: Robô - Sensores



2.3 Softwares Utilizados

Para conectar o robô ao computador é preciso que haja uma comunicação via Bluetooth, ou seja, que o computador possua um conector bluetooth, muito comum em notebooks. Para computadores de mesa, provavelmente será necessário um conector bluetooth externo, já que não é comum que venha de fábrica.

Para realizar a conexão via cabo USB, é necessário instalar o software da Lego para o primeiro uso. Após isso, é possível comandar o robô sem o software. Como este jeito é inviável para um robô móvel, uma vez que o tamanho do cabo não é suficiente e nem seria inteligente ficar seguindo o robô com notebook na mão, o método escolhido foi o do bluetooth. Portanto, este trabalho não detalhará o uso do robô com conexão USB.

Para o processamento dos dados e geração dos sistemas nebulosos, foi utilizado o MATLAB. Um software pago, que se propõe a ser uma solução para análise de dados, comunicação wireless, processamento de sinais, deep learning, entre outros assuntos. O site da MathWorks, empresa dona do MATLAB, diz que “O MATLAB combina um ambiente otimizado para análises iterativas e processos de design com uma linguagem de programação que expressa matrizes e arrays matemáticos diretamente”⁴[14]. A licença do MATLAB varia de R\$108,17⁵ para estudantes, até R\$8.765,50⁶[14], na licença padrão.

Para fazer a interação entre MATLAB e robô, foi adicionado ao programa o toolbox RWTH. Diferente de outras técnicas, o toolbox envia apenas comandos para o robô, deixando o processamento ser feito pelo computador e não pelo bloco inteligente [16]. A figura 2.6 explicita o caminho para comunicação entre o computador e o robô. Como fica claro, os comandos enviados para o robô são compilados e enviados ao protocolo de comunicação da LEGO. Neste trabalho, foi utilizada a comunicação via Bluetooth, assim, os comandos são enviados as camadas referentes ao Bluetooth e então enviados ao robô.

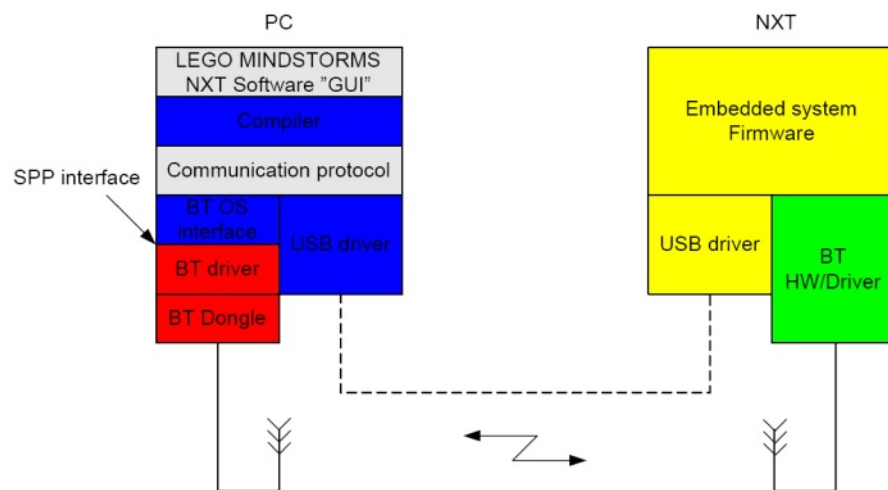
Na figura 2.6, o MATLAB faz o papel de “LEGO MINDSTORMS NXT Software GUI” e a seta com as palavras ”SPP interface“ aponta a interface de Serial Port Profile, que define um “perfil” esperado para o dispositivo que se conectará aquela porta[4]. Este perfil é uma série de protocolos e procedimentos que o dispositivo a se conectar deve ter e ser capaz de executar para que haja comunicação entre eles. Uma vez que a comunicação é estabelecida, ela é mantida até que um dos dispositivos a encerre. Por não ser o alvo deste trabalho, os detalhes mais técnicos a respeito do funcionamento do Bluetooth e o papel da SPP foram deixados de fora. A comunicação possui um alcance de aproximadamente 10 metros e o adaptador Bluetooth do computador é da versão 2.0. O bloco inteligente não se comunica com adaptadores de versões mais velhas que estas [16].

⁴“MATLAB® combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly.

⁵\$29,00. Dólar a R\$3,73

⁶\$2.350,00. Dólar a R\$3,73

Figura 2.6: Diagrama de bloco de comunicação entre computador e bloco inteligente



Capítulo 3

Cenário 1- Frenagem

3.1 Descrição do Problema

Neste cenário, o objetivo do sistema é realizar a frenagem do robô de maneira suave, quando o mesmo se aproxima de um obstáculo. Por questões de modelagem, a definição de quando o robô para de fato é feita de maneira booleana. Em outras palavras, quando o robô chega a uma distância determinada de um obstáculo, ele parará completamente. Esta distância foi estipulada em 10 centímetros pensando na integridade do robô. Como os sensores possuem uma margem de erro de mais ou menos 2 centímetros, essa distância de segurança poderia ser até menor, como por exemplo 5 centímetros. Porém, neste caso, existiria pouca margem de erro para o sistema, colocando em risco a integridade do robô.

Outro exemplo de uso em que este cenário seria útil é para desenvolvimento de frenagens emergenciais. Pode-se pensar na possibilidade do sistema assumir o controle de um carro caso o mesmo esteja acima de uma determinada velocidade e próximo de algum obstáculo. Um sistema como tal, seria útil para evitar acidentes de trânsito onde o motorista dorme no volante ou tenha seu tempo de reação debilitado por uso de substâncias alucinógenas, como álcool ou drogas. Oportunamente, durante a realização deste trabalho, os caminhoneiros fizeram uma greve [20] que trouxe a tona uma discussão sobre as ferrovias no Brasil. Este tema é relevante para este trabalho porque se torna mais um caso de uso para este cenário. Dada as características deste sistema, é possível utilizá-lo como um sistema emergencial para frear um trem, quando identificado um obstáculo nos trilhos.

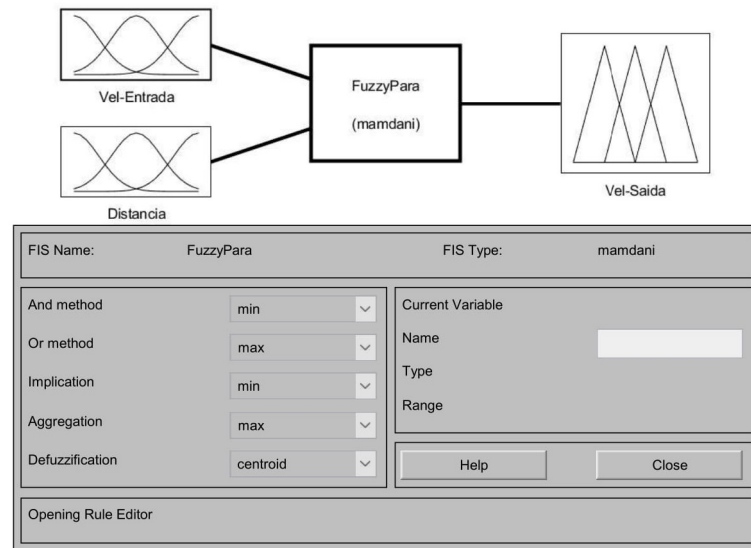
No geral, o papel do sistema é reduzir a velocidade de maneira com que, ao chegar a distância estabelecida, o robô possa parar completamente realizando uma desaceleração menos brusca possível.

3.2 Sistema Desenvolvido

O sistema desenvolvido está disponível, no formato do MATLAB, no apêndice A.

Neste sistema foi necessário fazer uso de uma condição booleana. Dado que o método de defuzzificação utilizado foi o método do centroide, não foi possível fazer o robô estacionar completamente. Utilizando outros tipos de conjunto, como o conjunto trapezoidal ou triangular, o robô não parava com-

Figura 3.1: Frenagem - Overview



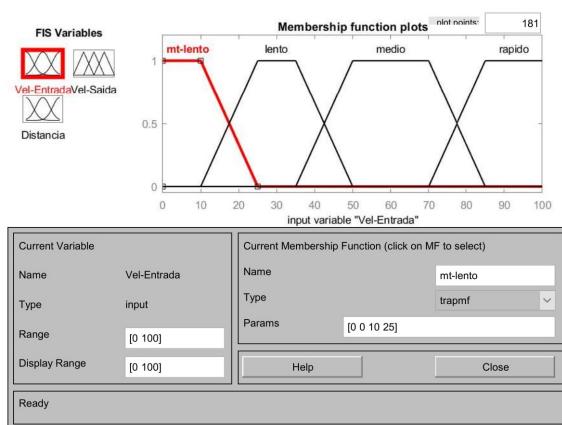
pletamente, uma vez que o centroide do conjunto de saída não se localizava na extremidade do mesmo. Assim, foi colocada uma condição booleana no código que faz com que o robô pare quando a distância é menor que 10 centímetros.

Uma outra solução pensada, para não fazer uso da lógica tradicional, seria utilizar um conjunto do tipo Singleton, que produziria o mesmo resultado. Porém, ao tentar esta abordagem, o robô não respondeu como deveria, então a solução escolhida foi fazer uso da lógica booleana.

Foram utilizados dois conjuntos de entrada: velocidade e distância, conforme as figuras 3.2a e 3.2b. Estes conjuntos representam a velocidade que o robô está e a distância para o obstáculo, respectivamente.

Figura 3.2: Frenagem - Conjuntos de entrada

(a) Frenagem - Conjunto Velocidade



(b) Frenagem - Conjunto Distância



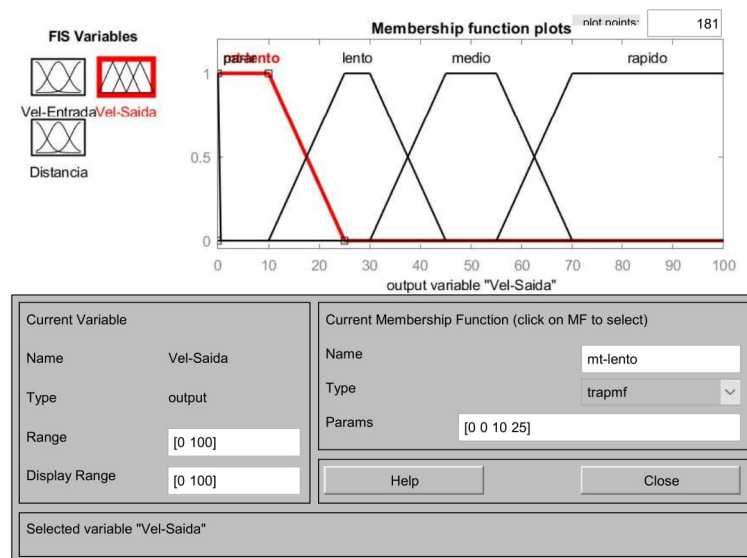
A escolha pelo formato dos conjuntos se deu pelo entendimento da situação. Foi considerado que se o obstáculo está a mais de 100 centímetros, ele está longe do robô. Como os sensores leem valores entre 0 e 255, percebe-se que a maior parte deste intervalo será considerado longe do robô.

Sobre o conjunto de entrada de velocidade, a ideia foi de ter algo mais balanceado, com conjuntos tendo tamanhos parecidos. Depois de alguns ajustes, chegou-se ao resultado visto na figura 3.2a. Um valor terá grau de pertinência máximo no conjunto 'lento' quando seu valor está entre 25 e 35, o que representa um intervalo de 10 unidades, enquanto um valor terá grau de pertinência máximo no conjunto 'médio' quando estiver entre 50 e 70, o que representa um intervalo de 20 unidades, o dobro do que no conjunto 'lento'. Desta maneira, percebe-se que a ideia de ter conjuntos de tamanhos iguais não foi para frente.

Vale ressaltar que o robô possui dois sensores de distância, logo ele realiza duas leituras que podem resultar em valores diferentes. Como o sistema foi pensado para ser usado de maneira emergencial, foi definido que a distância que valeria seria a menor entre as duas. Uma opção seria tirar a média destas leituras e utiliza-la como valor válido.

Foi utilizado um conjunto de saída: velocidade, conforme a figura 3.3. Este conjunto representa a velocidade de saída que o robô teve ter.

Figura 3.3: Frenagem - Conjunto de Saída Velocidade



Num primeiro momento, foi idealizado que o conjunto de saída fosse idêntico ao conjunto de velocidade de entrada. Depois de alguns testes, viu-se que esta não era a melhor opção.

Quando a distância era 'longe', a velocidade do robô era mantida e essa ideia foi a única razão pela qual o conjunto 'rápido' foi mantido na variável de saída. Só existe uma regra em que o resultado é este conjunto, que é quando o robô tem uma velocidade de entrada pertencente ao conjunto 'rápido' e a distância é 'longe'. Porém, como este conjunto começa com valores mais baixos que seu homônimo na variável de entrada, até nestas situações já existe uma diminuição da velocidade do robô.

Para os testes deste sistema, foi utilizado um obstáculo e o código foi testado a partir de várias distâncias do robô até o obstáculo diferente. Foi pensando em utilizar diversas velocidades iniciais para avaliar o comportamento do robô controlado pelo sistema, mas a velocidade sempre era ajustada nas primeiras interações, fazendo o robô se comportar de maneira similar em todos os casos.

Isso levou a conclusão de que a velocidade inicial não era fator no sistema, desde que ela não seja tão pequena. Uma vez que a premissa do sistema é a de frear, não faz sentido que ele aumente a velocidade do robô, salvo em situações que o obstáculo esteja muito próximo. Então, uma velocidade inicial muito baixa dificilmente seria aumentada pelo sistema, fazendo com que ela limite as possibilidades.

Devido a limitações dos sensores, foi necessário fazer uma correção em algumas distâncias lidas. Em determinados momentos, o ultrassom dos sensores identificava outros objetos que não eram o obstáculo que ele deveria frear. Um exemplo disso eram os cabos dos sensores, que, vez ou outra, entravam na frente dos sensores, devido ao movimento do robô.

Essas leituras faziam com que o robô fizesse uma redução maior do que a necessária. Para evitar este problema, as leituras equivocadas foram substituídas por leituras projetadas. Primeiramente, foi preciso identificar quais leituras eram equivocadas. Para isso, foi definido que se houvesse uma diferença entre a leitura anterior e a atual maior que 16, então a leitura atual era equivocada e precisava ser substituída. Em outras palavras, se $(\text{distância_anterior} - \text{distância_atual}) > 16$ então a distância atual estava errada.

Para gerar um valor válido para aquela leitura, foi estipulado que a distância projetada seria equivalente a distância anterior menos 6 unidades. Por exemplo, caso a distância anterior fosse de 55 centímetros e a leitura atual resultasse em 20 centímetros, ela seria corrigida para 49 centímetros.

Tanto o valor 16 quanto valor 6 foram valores concluídos a partir de observação e da média da variação no geral. São valores obtidos de forma empírica. Novamente, uma vez que sistema não acelera o robô, uma leitura errada implicaria em uma resposta muito baixa por parte do sistema, fazendo com que o robô se movesse de maneira lenta sem possibilidades de aumentar a velocidade novamente.

Vale ressaltar que o sistema foi desenvolvido pensando em cenários estáticos, onde o obstáculo estaria fixo desde o início do movimento do robô. Para cenários variáveis, onde um obstáculo pudesse surgir após a movimentação do robô ter sido iniciada, seria necessário alterar as correções de leitura.

Este cenário não é comum em competições de robótica, por ser muito simples de desenvolver utilizando a lógica booleana. Porém, justamente por ser tão simples, é um ótimo cenário inicial para um desenvolvimento mais rebuscado, adicionando a componente da lógica nebulosa.

3.3 Resultados Obtidos

Os resultados obtidos estão disponíveis no apêndice D. Do ponto de vista analítico, os resultados estão apresentados na tabela 3.1. O robô tinha uma velocidade inicial de 50 e foi testado com as distâncias iniciais conforme a primeira coluna da tabela 3.1. Isso permite analisar o comportamento dele de acordo com a distância.

Curiosamente, quando a distância inicial foi igual a 80 centímetros, o robô apresentou uma velocidade média maior, assim como um desvio padrão da velocidade também. Isso não se repetiu na desaceleração média, onde, quanto mais próximo do obstáculo era a distância inicial, maior foi a média da desaceleração.

A distância média e o desvio padrão da distância permite analisar o comportamento do robô

Tabela 3.1: Frenagem - Resultados estatísticos

Dist. Inicial	Média Velocidade	DP Velocidade	Média Desaceleração	Média Dist.	DPDist
150 cm	27.01	7.89	0.73	81.33	55.13
100 cm	26.95	9.56	1.28	66.33	29.16
80 cm	27.37	10.35	1.31	59.67	17.61
50 cm	23.48	8.71	1.85	46.00	6.08
30 cm	20.78	8.38	2.78	35.00	13.07

conforme se aproximava do obstáculo. Caso ele chegue perto com uma velocidade menor, a tendência é que existam mais leituras da distância, com valores menores, puxando estas duas métricas para baixo.

No geral, o sistema teve um desempenho considerado bom, por isso não foi utilizada nenhuma outra tentativa de aprimoração, além da calibragem manual.

Capítulo 4

Cenário 2 - Seguidor de Linha

4.1 Descrição do Problema

Neste cenário, o objetivo do sistema é fazer com que o robô siga um trajeto pré-estabelecido por uma linha no chão. Para isso, foi utilizada uma fita isolante preta sob um papel cartolina. A fita é o trajeto que o robô deve seguir, conforme a figura 4.1

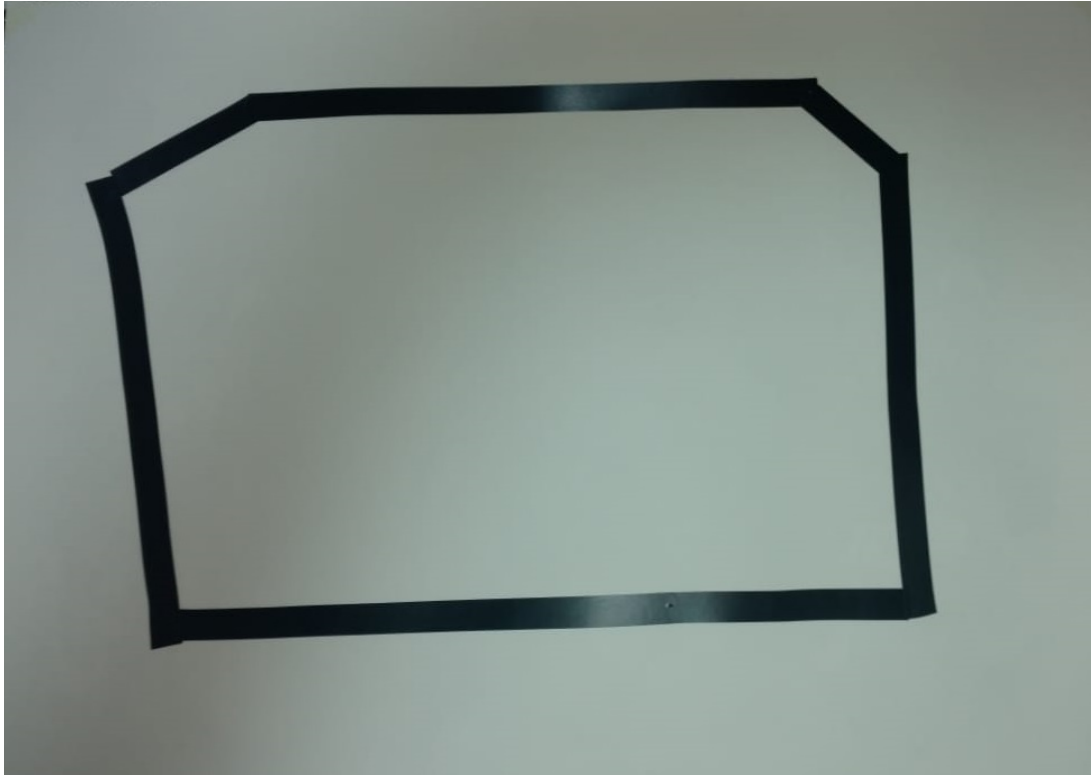
A figura 4.1 é uma foto do circuito ao qual o robô foi submetido. Ele possui quatro curvas para a esquerda em diferentes graus. O robô começa na parte inferior do circuito, da esquerda para a direita. A primeira curva que ele deve realizar é uma curva para a esquerda a 90° . A próxima curva é uma curva mais suave, a 45° graus

Neste sistema, foi definido que a cor preta indica que o robô está muito a esquerda do trajeto e a cor branca indica que o robô está muito a direita. Assim, o robô tende a seguir o trajeto predefinido ao iniciar uma curva para a direita quando lê a cor preta e tende a iniciar uma curva para a esquerda quando lê a cor branca. Em outras palavras, ele vai "zigzagueando" ao andar em linha reta, uma vez que ele não anda em linha reta de fato.

Dada as limitações do sensor de luminosidade, foi necessário ter cuidado com a velocidade do robô. Essencialmente, uma velocidade muito alta faz com que o robô se mova muito rápido, de um lado para o outro, fazendo com que o sensor leia valores muito extremos, que acabam sendo classificados como branco ou preto. Isso faz com que os movimentos do robô, em resposta as leituras, também sejam extremos. Por isso, não foi utilizado toda a potência dos motores.

A montagem do robô foi feita utilizando dois servo-motores e um sensor de luminosidade, como demonstrado na figura 2.5. Cada servo-motor ficou de um lado do robô e foram responsáveis pela movimentação do mesmo. O sensor de luminosidade foi posicionado na parte frontal do robô, posicionado no centro e virado para o chão, como visto na figura 2.3. A proximidade do sensor de luminosidade com o chão é importante para diminuir a influência da luminosidade do ambiente nas leituras.

Figura 4.1: Seguidor de Linha - Circuito



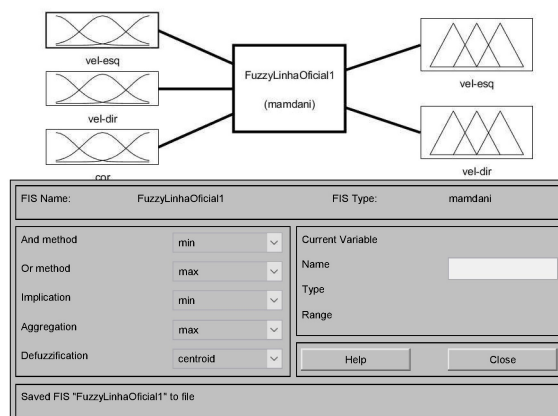
4.2 Sistema Desenvolvido

Uma vez que não foi possível atingir resultados satisfatórios com o robô, foram desenvolvidos três sistemas diferentes. Eles estão disponíveis, no formato do MATLAB, no apêndice B.

4.2.1 Sistema 1

A visão geral do sistema está na figura 4.2. Este sistema teve um bom desempenho em seguir as linhas retas do circuito, mas não conseguiu realizar curvas de maneira satisfatória.

Figura 4.2: Seguidor de Linha 1 - Overview



O sistema possui três variáveis de entrada: 'vel-esq', 'vel-dir' e 'cor', representados nas figuras

4.3a, 4.3b e 4.3c.

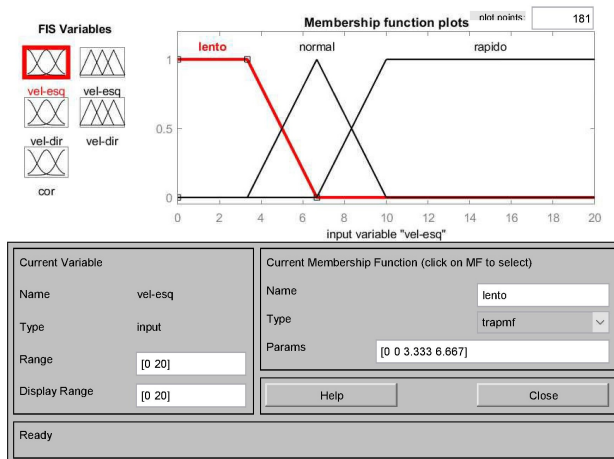
As variáveis 'vel-esq' e 'vel-dir' são compostas por três conjuntos, sendo o 'lento' e o 'rápido' conjuntos trapezoidais e o 'normal' um conjunto triangular. O intervalo de valores para as variáveis é entre 0 e 20. Este intervalo não representa todos os valores aceitáveis pelos motores do NXT. Isso foi uma escolha, devido ao fato de que o robô se comporta muito mal quando tem uma velocidade alta.

O fato de que o conjunto 'normal' tem formato triangular, faz com que o conjunto 'rápido' comece exatamente quando o conjunto 'lento' termine, que é o ponto ápice do conjunto 'normal'.

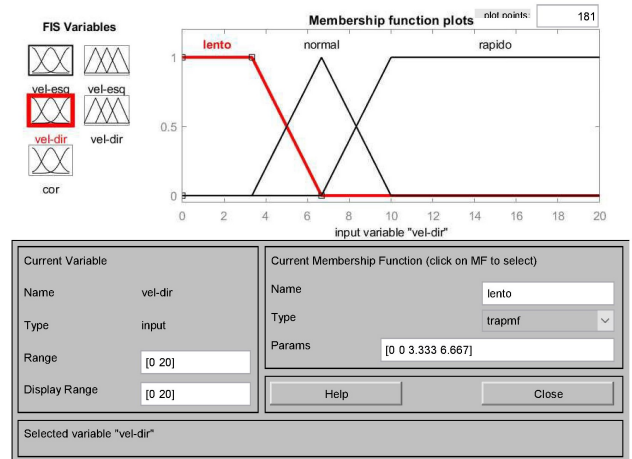
A variável 'cor' é composta por dois conjuntos trapezoidais, que representam a cor lida pelo sensor. O intervalo de valores é entre 0 e 700. Foram escolhidos estes dois conjuntos por questões de desempenho. Inicialmente, foi pensando em três conjuntos, sendo o terceiro equivalente a cor cinza. Esta ideia foi descartada devido ao desempenho do robô. Assim, a cor cinza acaba por ser representada no trecho de indecisão entre ambos os conjuntos, que é o intervalo entre 250 e 500.

Figura 4.3: Seguidor de Linha 1 - Conjuntos de entrada

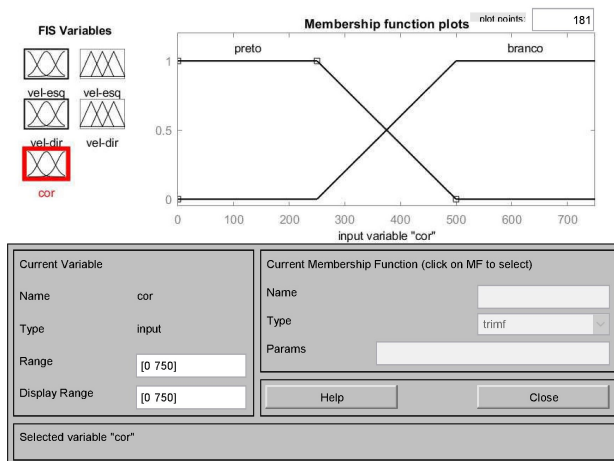
(a) Entrada - Velocidade da esquerda



(b) Entrada - Velocidade da direita



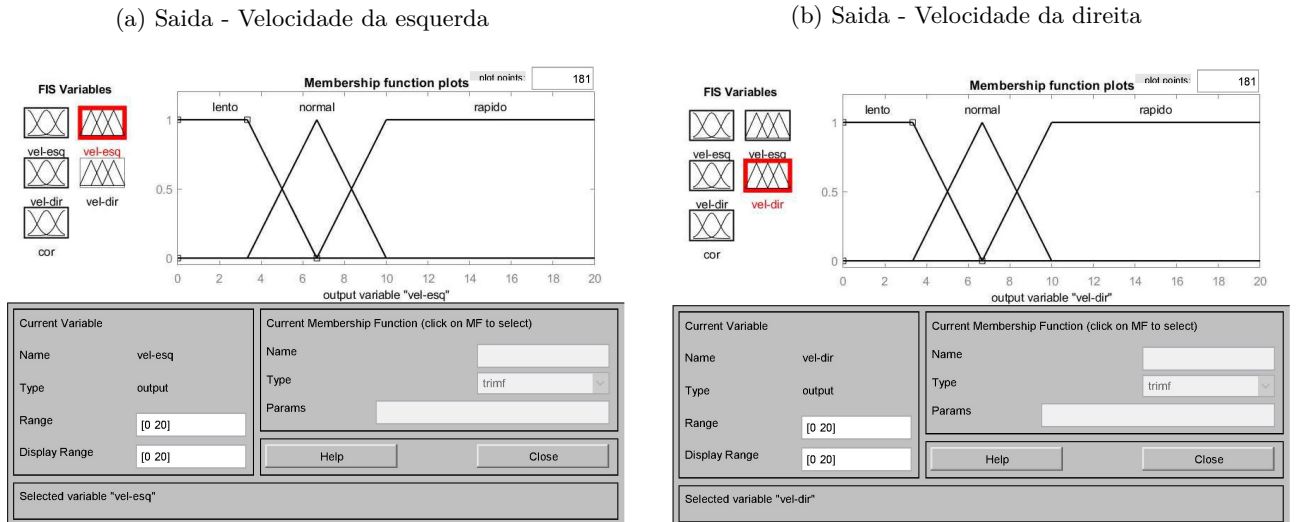
(c) Entrada - Cor



O sistema possui duas variáveis de saída: 'vel-esq' e 'vel-dir', são similares as variáveis de entrada

homônimas, e estão representados nas figuras 4.4a e 4.4b, respectivamente. Seus conjuntos são idênticos aos das variáveis de entrada homônimas.

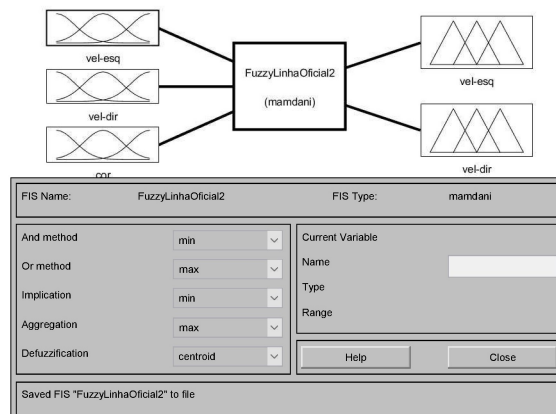
Figura 4.4: Seguidor de Linha 1 - Conjuntos de saída



4.2.2 Sistema 2

A visão geral do sistema está na figura 4.5. Este sistema teve um bom desempenho em realizar as curvas do circuito, embora precisasse de mais espaço do que o desejo para fazê-lo. Já nas retas, ele foi inconsistente, tendo momentos em que quase avançava a linha preta e ia para a parte interior do sistema.

Figura 4.5: Seguidor de Linha 2 - Overview



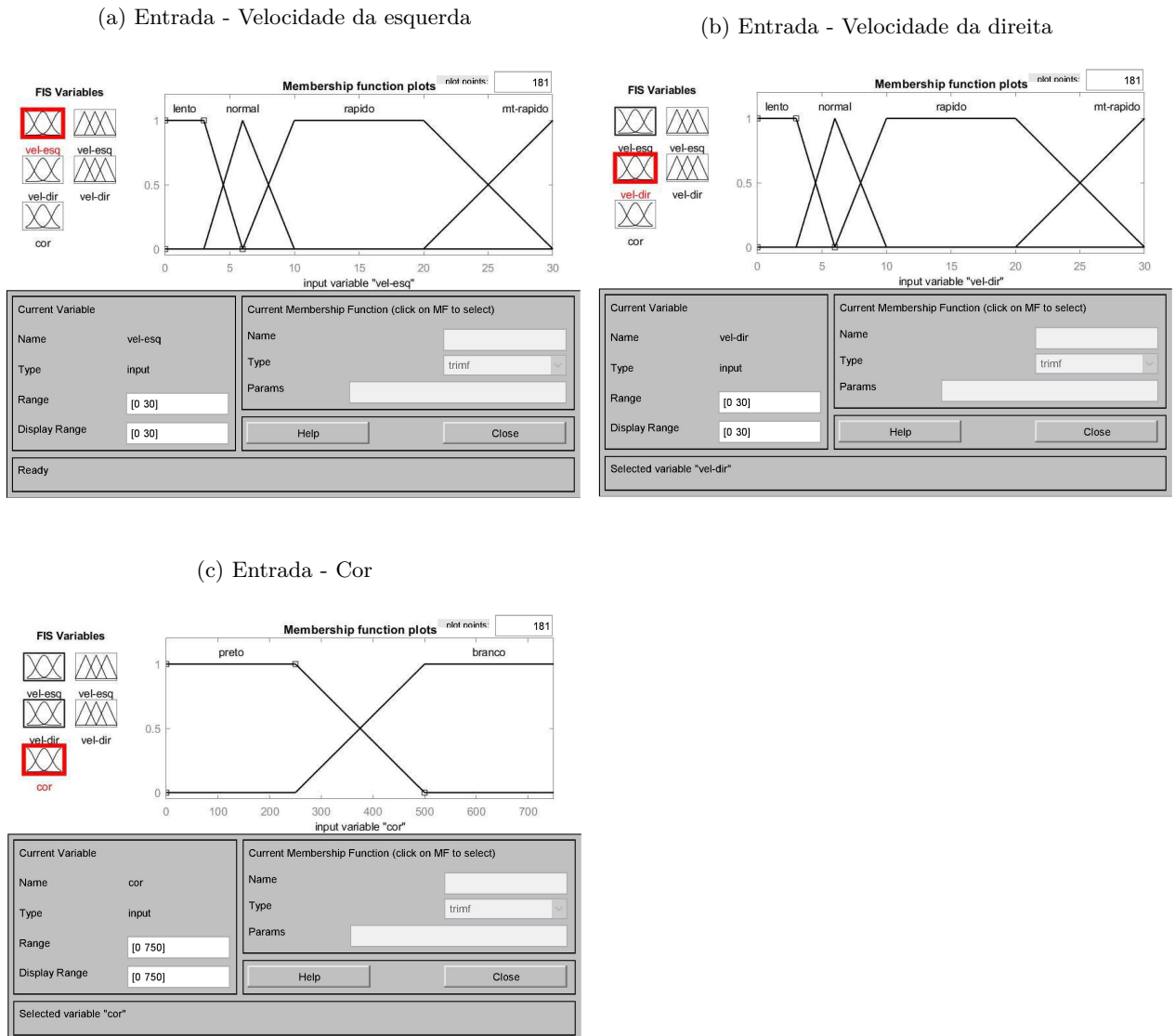
O sistema possui três variáveis de entrada: 'vel-esq', 'vel-dir' e 'cor', representados nas figuras 4.6a, 4.6b e 4.6c.

As variáveis 'vel-esq' e 'vel-dir' são compostas por quatro conjuntos, sendo o 'lento' e o 'rapido' conjuntos trapezoidais e o 'normal' e o 'mt-rapido' conjuntos triangulares. O intervalo de valores para as variáveis é entre 0 e 20. Diferente do sistema 4.2.1, as variáveis de entrada de velocidade possuem um

conjunto que representa velocidades muito rápidas. O papel deste sistema é maior nas variáveis de saída, porém, foi incorporado às variáveis de entrada para manter a igualdade entre as variáveis.

A variável 'cor' é composta por dois conjuntos trapezoidais, que representam a cor lida pelo sensor. O intervalo de valores é entre 0 e 700. Está variável é idêntica à do 4.2.1, com a mesma explicação para a definição dos valores e conjuntos.

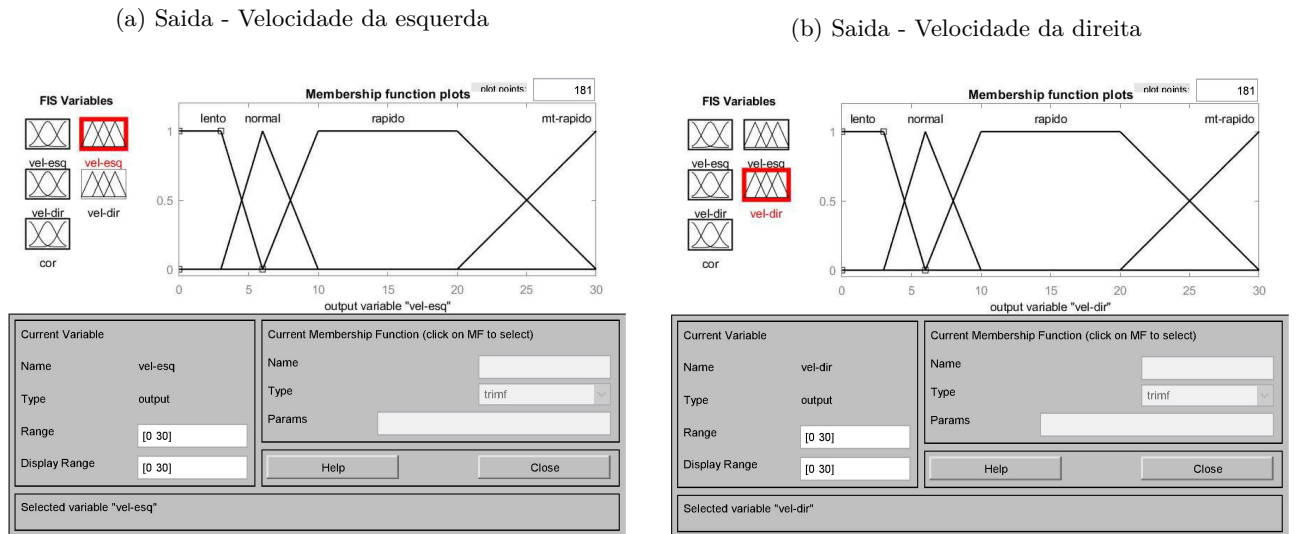
Figura 4.6: Seguidor de Linha 2 - Conjuntos de entrada



O sistema possui duas variáveis de saída: 'vel-esq' e 'vel-dir', são similares as variáveis de entrada homônimas, e estão representados nas figuras 4.7a e 4.7b, respectivamente.

O conjunto 'mt-rapido' tem papel importante aqui. Ele é utilizado quando o robô já está realizando uma curva, ou seja, está com o sensor sobre a fita preta ou sobre a parte branca e possui velocidade maior em um motor. O objetivo é que este conjunto faça o robô realizar a curva mais rápido, necessitando de menos espaço para fazê-lo. Até certo ponto, esta abordagem foi boa. De fato, o robô realiza curvas melhor que com o sistema 4.2.1, mas ainda não as faz de maneira satisfatória.

Figura 4.7: Seguidor de Linha 2 - Conjuntos de saída



4.2.3 Sistema ANFIS

Este sistema foi gerado através da concatenação dos resultados dos sistemas 4.2.1 e 4.2.2. Foi utilizado o comando *anfis*[13], que gera um sistema nebuloso do tipo Sugeno, com uma saída. Para utilizar, é necessário um conjunto de treino de tamanho $N + 1$. no caso deste trabalho, o conjunto de treino foi uma matriz de 3 colunas e 200 linhas, onde a primeira coluna era a velocidade de entrada do motor, a segunda coluna era a cor lida e a terceira era o resultado apresentado pelos outros sistemas.

Como o sistema gerado só possui uma saída, foram utilizados dois sistemas, um para cada motor. Desta maneira, a primeira coluna conjunto de treino do motor da direita era a velocidade de entrada do motor da direita nos outros sistemas. O mesmo foi válido para o sistema da esquerda.

O algoritmo de treino utiliza uma combinação de backpropagation com o método dos mínimos quadrados [13]. Ambos sistemas geraram resultados similares, como esperado. Os números estão expostos na tabela 4.1. Vale explicar que a função de defuzzificação, chamada de $f(x, y)$ na seção 2.1 é a função 'wtaver'.

A única diferença entre os sistemas durante a geração foi o RMSE mínimo, que é um valor atrelado ao treino da rede neural utilizada pelo *anfis* na geração. No sistema da direita, o valor foi de 0.483600, enquanto que no da esquerda, o valor foi de 1.004721.

4.3 Resultados Obtidos

Para avaliar os sistema, inicialmente o robô foi seguindo uma linha reta. Posteriormente, o trajeto possuía uma curva para a direita. Depois, possuía uma curva para a direita e uma para a esquerda. Finalmente, o robô foi submetido a um percurso em formato retangular. Foi testado o robô em ambos os sentidos possíveis do cenário de testes, de maneira que o robô realizava curvas para a direita e curvas para a esquerda também.

O grande problema nestes sistemas vem da dificuldade de fazer o sensor de cor ler valores inter-

Tabela 4.1: Seguidor de Linha - Dados ANFIS

Nº de nós	35
Nº de parâmetros lineares	27
Nº de parâmetros não-lineares	18
Nº total de parâmetros	45
Nº de pares do conjunto de treino	200
Nº de pares do conjunto de checagem	0
Nº de regras nebulosas	9
Tipo	'sugeno'
Conjunção	'prod'
Disjunção	'max'
Defuzzificação	'wtaver'
Implicação	'prod'
Agregação	'max'
Entrada	2 conjuntos
Saida	1 conjunto
Regras	9 regras

mediários entre o preto e o branco. Isso impediu que os sistemas desenvolvidos fossem um pouco mais sofisticados, necessitando manter apenas dois conjuntos na variável de entrada 'cor'. Também por causa disto, o robô oscilava entre seguir uma linha reta de maneira mais suave e precisar fazer curvas mais abertas e conseguir fazer curvas mais fechadas, porém sendo menos estável ao seguir em linha reta. Na parte de calibragem, este foi o grande desafio, achar um termo entre fazer curvas muito abertas e seguir a linha reta de maneira suave.

Durante testes, foi tentado alterar o método de defuzzificação para 'mom', que significa a 'média do máximo', ou seja, a média do maior conjunto de saída, o robô apresentou comportamento em linhas retas que não foi satisfatório, já que o mesmo não andava em linha reta e ficava indo para um lado e para o outro.

A abordagem escolhida para tentar contornar estes problemas, foi tentar usar o *anfis* para gerar um sistema intermediário. Em partes, a ideia funcionou. De fato, o sistema gerado teve um desempenho um pouco melhor ao seguir a linha reta e ao realizar curvas. Porém, eles esteve mais sujeito a falhas, como gerar valores de saída para os motores maiores do que os permitidos, ou fazer curvas fechado demais, invadindo a parte de dentro do circuito.

Uma questão sobre este sistema é que a luz ambiente influencia consideravelmente sobre o desempenho. Dependendo do ambiente, é necessário realizar ajustes na variável de entrada de cor.

Capítulo 5

Cenário 3 - Desvio de Obstáculos

5.1 Descrição do Problema

Neste cenário, o objetivo do sistema é desviar o robô ao encontrar um obstáculo. Vale ressaltar que o robô não segue um trajeto pré-definido, portanto ele não precisa dar uma volta no obstáculo, somente evitar a colisão, mesmo que isto signifique ir para uma direção diferente da original.

A montagem do robô foi feita utilizando dois servo-motores e dois sensores de ultrassom. Cada servo-motor ficou de um lado do robô e foram responsáveis pela movimentação do mesmo. Os sensores de luminosidade foram posicionados na parte frontal do robô, sendo um sensor posicionado mais a esquerda e outro mais a direita do mesmo.

Os sensores de ultrassom emitem ondas que são rebatem nos obstáculos, permitindo calcular a distância aproximada, em centímetros, do robô aos obstáculos.

5.2 Sistema Desenvolvido

O sistema desenvolvido está disponível, no formato do MATLAB, no apêndice C.

Neste sistema foram utilizados duas variáveis de entrada e duas variáveis de saída, conforme a figura 5.1.

Foi necessário escolher um lado para onde o robô deveria virar, afim de evitar a colisão. Foi escolhido que o robô passaria a ter uma velocidade maior no motor da direita, de maneira que ele faria uma curva para a esquerda.

As variáveis de entrada representam a distância entre o robô e um obstáculo, nomeadas 'dist-esq' e 'dist-dir', conforme as figuras 5.2a e 5.2b. Os conjuntos 'perto' e 'medio' possuem formato triangular, enquanto os outros conjunto possuem formato trapezoidal. Além disso, cada variável aceita valores que variam entre 0 e 255.

Para efeito de melhor visualização, as figuras mostram apenas valores entre 0 e 100. Caso fosse mostrar todo o intervalo, o resultado seria de difícil visualização;

Diferente dos sistemas desenvolvidos para os outros cenários, este possui bastante conjuntos.

Figura 5.1: Desvio - Overview

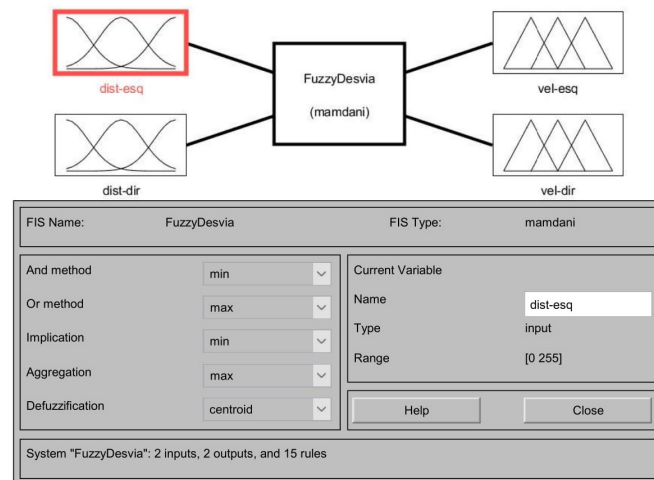
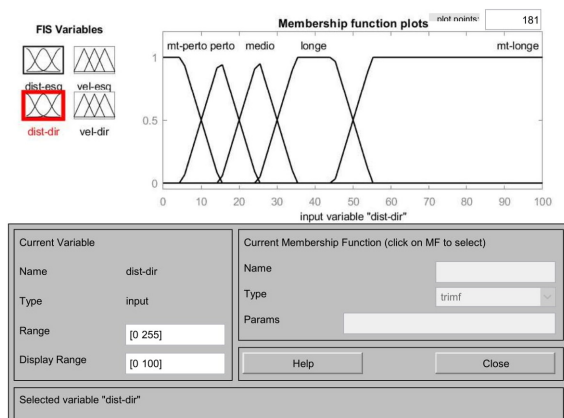
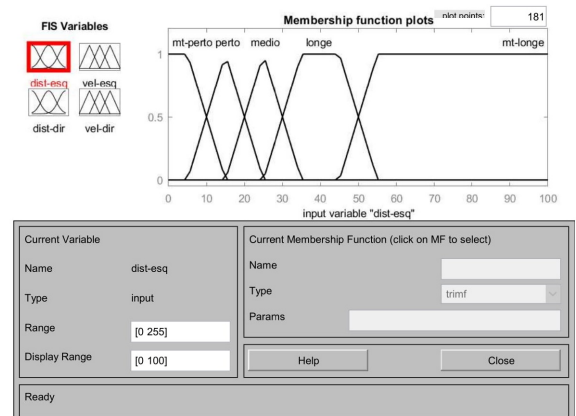


Figura 5.2: Desvio - Conjuntos de Entrada

(a) Distância do sensor direito



(b) Distância do sensor esquerdo



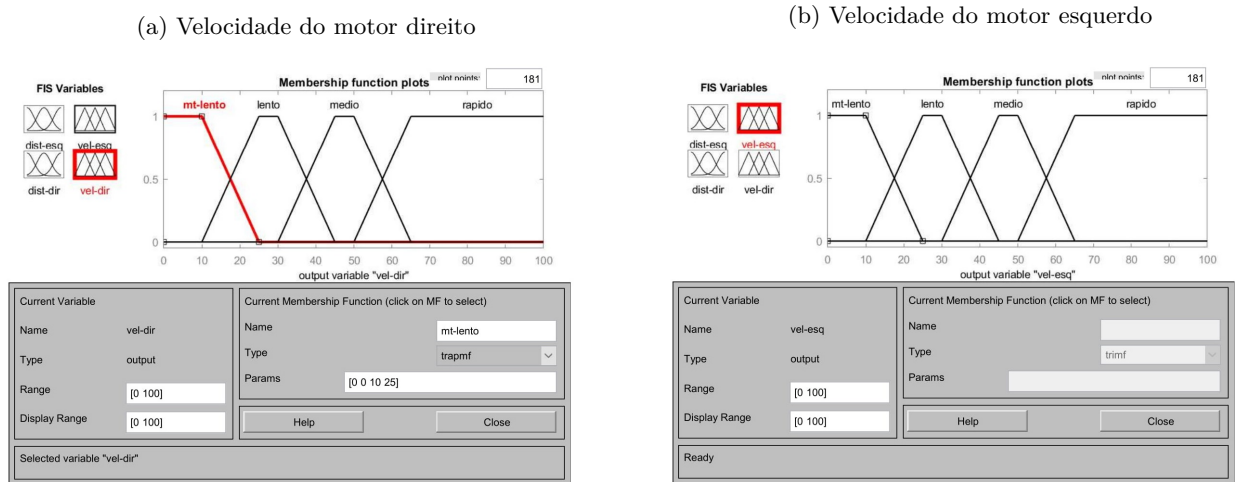
Isso implica em um número maior de regras e, conseqüentemente, uma dificuldade maior na hora de dar manutenção. Este foi um cuidado tido nos outros casos, mas que aqui não teve muito como ser contornado.

A opção por não levar em consideração a velocidade atual do robô se deu pela ideia de que este sistema teria controle total sobre o mesmo, de maneira que seria acabaria sendo pouco relevante para o resultado final. Como já foi dito, este sistema possui muitas regras, caso houvessem mais duas variáveis de entrada, representado as velocidades e similares a dos outros sistemas, o número de regras seria em torno de 200.

Já as variáveis de saída representam a velocidade a ser passada para os motores e estão exibidas nas figuras 5.3a e 5.3b. Cada variável possui quatro conjuntos e são eles: 'mt-lento', 'lento', 'medio' e 'rapido'. Todos os conjuntos são trapezoidais e o intervalo de valores é de 0 a 100.

Existe uma simetria entre os conjuntos 'lento' e 'medio' que se dá pela escolha do autor. Tendo esta simetria, a variável fica mais fácil de ser compreendida. Como isso não afetou o desempenho, foi

Figura 5.3: Desvio - Conjuntos de Saída



mantido. Percebe-se também que o conjunto 'rápido' é o maior. Isso se dá porque o sistema pode precisar acelerar um motor de maneira com que seja feito o desvio, mas é preciso tomar cuidado para não acelerar demais, impossibilitando que o mesmo aconteça.

Como este sistema assume total controle do robô e não apenas em situação emergencial, a velocidade aumentada também quando o obstáculo está longe.

5.3 Resultados Obtidos

Os resultados estão expostos no apêndice F.

Para obter os resultados, o sistema foi testado em duas situações diferentes. Primeiro, ele foi posicionado a uma distância de 30 centímetros do obstáculo. Uma análise estatística está exposta na tabela 5.1.

Tabela 5.1: Desvio - Estatísticas 30 Centímetros

	Dist. Esq	Dist. Dir.	Vel. Entrada Esq.	Vel. Entrada Dir.	Vel. Saida Esq.	Vel Saida Dir.
Média	30,00	30,00	43,50	58,50	44,64	58,55
DP	119,08	115,95	25,29	22,28	22,65	22,92

Uma análise estatística para a situação em que o robô estava a 50 centímetros do obstáculo está exposta na tabela 5.2.

Tabela 5.2: Desvio - Estatísticas 50 Centímetros

	Dist. Esq	Dist. Dir.	Vel. Entrada Esq.	Vel. Entrada Dir.	Vel. Saida Esq.	Vel Saida Dir.
Média	45,00	51,00	50,00	77,00	50,00	77,43
DP	90,06	92,45	21,54	22,16	18,65	22,08

Comparando as duas tabelas, pode-se observar que, quando teve mais espaço, o robô realizou variações menores nas velocidades. Outro fato, que fica mais evidente na tabela 5.2 é que a média da velocidade de saída da esquerda foi menor que a da direita, mostrando que o robô tendia para a esquerda quando queria desviar do obstáculo.

Uma curiosidade é que o desvio padrão das distâncias são muito maiores que as médias das mesmas. Isso se dá porque as distâncias lidas crescem muito quando o robô não está em frente ao obstáculo. Nestas situações, a leitura é o valor total do sensor, 225 centímetros, daí a variação grande mostrada nos desvios padrões.

Capítulo 6

Conclusão

Primeiramente, é preciso explicitar o grande problema encontrado durante todo o trabalho: consumo de energia do robô. O robô faz uso de seis pilhas AA e elas duram em torno de 40 minutos. Isso foi um grande problema, pois era preciso trocar de pilha constantemente durante o desenvolvimento. Além disso, quando as pilhas tinham pouca carga, o robô apresentava resultados estranhos, como uma velocidade que não era condizendo com a passa ou leituras erradas dos sensores.

Ainda na questão do robô, um ponto negativo foi a demora na comunicação entre computador e robô. Isso foi importante para se manter uma distância de segurança maior no sistema de frenagem, apresentado no capítulo 3, onde era arriscado reduzir a distância de segurança justamente por causa da demora na leitura; e na dificuldade no desenvolvimento do sistema de seguidor de linha, apresentado no capítulo 4. Neste último, o robô acabava por andar muito sem ler a cor. Isso motivou a uma redução da velocidade no sistema, que mesmo assim não resolveu o problema.

Por parte da lógica nebulosa, é fato que é uma grande ferramenta que só tem a adicionar no desenvolvimento tecnológico da sociedade. Este trabalho considera que a grande dificuldade encontrada ao utiliza-la é o número de regras. Diversas vezes foi preciso levar isto em consideração durante o desenvolvimento. Seria extremamente difícil criar e manter um sistema com 200 regras, por exemplo.

Por falar em ferramental, conclui-se que a utilização dos MATLAB foi aceitável. O software peca em alguns detalhes, como na geração de gráfico, que foi tarefa árdua neste desenvolvimento. Mas em outros aspectos, é um software poderoso, que consegue processar uma quantidade razoável de informação em um tempo aceitável.

Por parte do toolbox para criação de sistemas nebulosos, a interface gráfica se mostra obsoleta e antiquada, como é possível ver nas figuras 5.1 ou 3.1. Isso é um limitador razoável, na opinião do autor. De fato, a criação e manutenção dos sistemas se mostrou um desafio maior que o esperado, ao ponto de ser necessário utilizar de outros meios para chegar em resultados imaginados. Constantemente, foi mais prático editar o arquivo .fis diretamente, do que utilizar a interface gráfica do toolbox. Por sorte, os arquivos não são codificados e, por isso, é fácil de editar.

Ponto positivo para o MATLAB é a existência de uma ferramenta para criação de sistemas nebulosos, utilizando redes neurais. O *anfis* é, sem dúvida, uma ferramenta poderosa no software. A ideia

de utiliza-lo afim de otimizar os sistemas, se deu pela facilidade que é sua utilização. Com poucas linhas de comando e com extrema rapidez, é possível gerar um sistema para ser testado. A respeito da qualidade dos sistemas gerados, é complicado analisar, uma vez que ele reproduz tendências e, conseqüentemente, erros dos sistemas no qual ele se baseia. Por exemplo, foi definido no seguidor de linha, que ao ler a cor preta, o robô deveria iniciar uma curva para a direita. Isso foi reproduzido no sistema gerado pelo *anfis* sem necessidade de intervenção por parte do autor.

No final, conclui-se que os sistemas desenvolvidos são bons sistemas, que podem servir de base para outros no futuro. Porém, é de se ressaltar que são sistemas básicos, utilizados em um robô limitado para tais fins. Eles precisam de melhorias, principalmente o seguidor de linha, que não obteve sucesso.

Na questão do ferramental, é preciso dizer que o MATLAB é um software poderoso que, no final, acaba cumprindo com as expectativas. É possível utiliza-lo e conseguir bons resultados. Já o robô NXT, é um robô muito limitado para o que foi tentado aqui. Claramente, seu enfoque é no ensino de crianças e nisso ele aparenta ser muito bom. No geral, é um robô interessante, mas limitado.

Capítulo 7

Trabalhos Futuros

No futuro, o autor pretende criar um cenário no qual os três sistemas desenvolvidos sejam utilizados. Este cenário seria de extremo interesse para a validação da ideia dos sistemas. Porém, existe o receio de que o ferramental aqui utilizado não dê conta da tarefa.

Uma vez que a tentativa de melhora do sistema de seguidor de linha utilizando o anfis apresentou resultados interessantes, a ideia é trazer esta abordagem para os outros sistemas. Isso não foi feito durante a realização deste trabalho pois os resultados por parte dos sistemas foi considerado aceitável, diferente do que aconteceu com o seguidor de linha.

Apêndice A

Sistema de Frenagem

```
[System]
Name='FuzzyPara'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=16
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

    [Input1]
Name='Vel-Entrada'
Range=[0 100]
NumMFs=4
MF1='mt-lento':'trapmf',[0 0 10 25]
MF2='lento':'trapmf',[10 25 35 50]
MF3='medio':'trapmf',[35 50 70 85]
MF4='rapido':'trapmf',[70 85 100 100]

    [Input2]
Name='Distancia'
Range=[0 255]
NumMFs=4
MF1='mt-perto':'trapmf',[0 0 10 30]
MF2='perto':'trapmf',[10 30 40 60]
MF3='medio':'trapmf',[40 60 80 100]
```



```
MF4='Longe':'trapmf',[80 100 255 255]
```

```
    [Output1]
```

```
Name='Vel-Saida'
```

```
Range=[0 100]
```

```
NumMFs=5
```

```
MF1='mt-lento':'trapmf',[0 0 10 25]
```

```
MF2='lento':'trapmf',[10 25 30 45]
```

```
MF3='medio':'trapmf',[30 45 55 70]
```

```
MF4='rapido':'trapmf',[55 70 100 100]
```

```
MF5='parar':'trimf',[0 0 0]
```

```
    [Rules]
```

```
1 1, 5 (1) : 1
```

```
1 2, 1 (1) : 1
```

```
1 3, 1 (1) : 1
```

```
1 4, 1 (1) : 1
```

```
2 1, 1 (1) : 1
```

```
2 2, 2 (1) : 1
```

```
2 3, 2 (1) : 1
```

```
2 4, 2 (1) : 1
```

```
3 1, 1 (1) : 1
```

```
3 2, 2 (1) : 1
```

```
3 3, 3 (1) : 1
```

```
3 4, 3 (1) : 1
```

```
4 1, 1 (1) : 1
```

```
4 2, 2 (1) : 1
```

```
4 3, 3 (1) : 1
```

```
4 4, 4 (1) : 1
```

Apêndice B

Sistema de Seguidor de linha

B.1 Sistema 1

[System]

Name='FuzzyLinhaOficial1'

Type='mamdani'

Version=2.0

NumInputs=3

NumOutputs=2

NumRules=18

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'

[Input1]

Name='vel-esq'

Range=[0 20]

NumMFs=3

MF1='lento': 'trapmf', [0 0 3.333 6.667]

MF2='normal': 'trimf', [3.333 6.667 10]

MF3='rapido': 'trapmf', [6.667 10 20 20]

[Input2]

Name='vel-dir'

Range=[0 20]

NumMFs=3

```
MF1='lento':'trapmf',[0 0 3.333 6.667]
MF2='normal':'trimf',[3.333 6.667 10]
MF3='rapido':'trapmf',[6.667 10 20 20]
```

[Input3]

```
Name='cor'
Range=[0 750]
NumMFs=2
MF1='preto':'trapmf',[0 0 250 500]
MF2='branco':'trapmf',[250 500 750 750]
```

[Output1]

```
Name='vel-esq'
Range=[0 20]
NumMFs=3
MF1='lento':'trapmf',[0 0 3.333 6.667]
MF2='normal':'trimf',[3.333 6.667 10]
MF3='rapido':'trapmf',[6.667 10 20 20]
```

[Output2]

```
Name='vel-dir'
Range=[0 20]
NumMFs=3
MF1='lento':'trapmf',[0 0 3.333 6.667]
MF2='normal':'trimf',[3.333 6.667 10]
MF3='rapido':'trapmf',[6.667 10 20 20]
```

[Rules]

```
1 1 1, 2 1 (1) : 1
2 2 1, 3 2 (1) : 1
3 3 1, 3 2 (1) : 1
1 1 2, 1 2 (1) : 1
2 2 2, 2 3 (1) : 1
3 3 2, 2 3 (1) : 1
1 2 1, 2 1 (1) : 1
1 3 1, 2 1 (1) : 1
2 1 1, 3 2 (1) : 1
2 3 1, 3 2 (1) : 1
3 1 1, 3 1 (1) : 1
```

3 2 1, 3 2 (1) : 1
 1 2 2, 2 3 (1) : 1
 1 3 2, 1 3 (1) : 1
 2 1 2, 1 2 (1) : 1
 2 3 2, 2 3 (1) : 1
 3 1 2, 1 2 (1) : 1
 3 2 2, 2 3 (1) : 1

B.2 Sistema 2

[System]

Name='FuzzyLinha3'

Type='mamdani'

Version=2.0

NumInputs=3

NumOutputs=2

NumRules=18

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'

[Input1]

Name='vel-esq'

Range=[0 30]

NumMFs=4

MF1='lento': 'trapmf', [0 0 3 6]

MF2='normal': 'trimf', [3 6 10]

MF3='rapido': 'trapmf', [6 10 20 30]

MF4='mt-rapido': 'trimf', [20 30 30]

[Input2]

Name='vel-dir'

Range=[0 30]

NumMFs=4

MF1='lento': 'trapmf', [0 0 3 6]

MF2='normal': 'trimf', [3 6 10]

MF3='rapido':'trapmf',[6 10 20 30]

MF4='mt-rapido':'trimf',[20 30 30]

[Input3]

Name='cor'

Range=[0 750]

NumMFs=2

MF1='preto':'trapmf',[0 0 250 500]

MF2='branco':'trapmf',[250 500 750 750]

[Output1]

Name='vel-esq'

Range=[0 30]

NumMFs=4

MF1='lento':'trapmf',[0 0 3 6]

MF2='normal':'trimf',[3 6 10]

MF3='rapido':'trapmf',[6 10 20 30]

MF4='mt-rapido':'trimf',[20 30 30]

[Output2]

Name='vel-dir'

Range=[0 30]

NumMFs=4

MF1='lento':'trapmf',[0 0 3 6]

MF2='normal':'trimf',[3 6 10]

MF3='rapido':'trapmf',[6 10 20 30]

MF4='mt-rapido':'trimf',[20 30 30]

[Rules]

1 1 1, 2 1 (1) : 1

2 2 1, 3 2 (1) : 1

3 3 1, 3 2 (1) : 1

1 1 2, 1 2 (1) : 1

2 2 2, 2 3 (1) : 1

3 3 2, 2 3 (1) : 1

1 2 1, 2 1 (1) : 1

1 3 1, 2 1 (1) : 1

2 1 1, 3 2 (1) : 1

2 3 1, 3 2 (1) : 1

3 1 1, 3 1 (1) : 1
 3 2 1, 4 2 (1) : 1
 1 2 2, 2 3 (1) : 1
 1 3 2, 1 3 (1) : 1
 2 1 2, 1 2 (1) : 1
 2 3 2, 2 4 (1) : 1
 3 1 2, 1 2 (1) : 1
 3 2 2, 2 3 (1) : 1

B.3 Sistema AnfisEsq

[System]

Name='FuzzyAnfisEsq'

Type='sugeno'

Version=2.0

NumInputs=2

NumOutputs=1

NumRules=9

AndMethod='prod'

OrMethod='max'

ImpMethod='prod'

AggMethod='max'

DefuzzMethod='wtaver'

[Input1]

Name='input1'

Range=[5 27]

NumMFs=3

MF1='in1mf1':'gbellmf',[5.86839472592555 1.01149857310175 5.08019419081619]

MF2='in1mf2':'gbellmf',[5.31992912107993 0.247449594236159 16.0140093095384]

MF3='in1mf3':'gbellmf',[6.67298696955375 2.93382517753495 25.5581115479491]

[Input2]

Name='input2'

Range=[166 608]

NumMFs=3

MF1='in2mf1':'gbellmf',[110.494129727139 2.02360383269039 165.994972007636]

MF2='in2mf2':'gbellmf',[110.495553488039 1.87786591986441 386.986634624917]

MF3='in2mf3':'gbellmf',[110.510659329395 2.16672612880184 607.987144458123]

[Output1]

Name='output'

Range=[6.335974304 26.76534653]

NumMFs=9

MF1='out1mf1':'linear',[3.67081159488117 -0.034054425059601 84.4538861090872]

MF2='out1mf2':'linear',[-5.52488376675513 -0.0309795627729337 14.1643957471139]

MF3='out1mf3':'linear',[1.58763581143363 0.0365101100787894 -12.4268616395377]

MF4='out1mf4':'linear',[16.9128520492383 0.0411915165988426 -282.502207201615]

MF5='out1mf5':'linear',[-9.09321010350688 0.00423265707223878 178.193129314778]

MF6='out1mf6':'linear',[2.32493419401284 0.00844032011188434 -43.1633866781791]

MF7='out1mf7':'linear',[-0.885159510266965 0.0636272680847264 -32.9690607781752]

MF8='out1mf8':'linear',[1.83089397526544 0.0293099698924053 1.85709577873634]

MF9='out1mf9':'linear',[0.0761163709356837 0.0630509682680602 -45.1958062371614]

[Rules]

1 1, 1 (1) : 1

1 2, 2 (1) : 1

1 3, 3 (1) : 1

2 1, 4 (1) : 1

2 2, 5 (1) : 1

2 3, 6 (1) : 1

3 1, 7 (1) : 1

3 2, 8 (1) : 1

3 3, 9 (1) : 1

B.4 Sistema AnfisDir

[System]

Name='FuzzyAnfisDir'

Type='sugeno'

Version=2.0

NumInputs=2

NumOutputs=1

NumRules=9

AndMethod='prod'

OrMethod='max'

ImpMethod='prod'

AggMethod='max'

DefuzzMethod='wtaver'

[Input1]

Name='input1'

Range=[5 27]

NumMFs=3

MF1='in1mf1':'gbellmf',[6.52746408133069 1.06643839751541 5.71239648778939]

MF2='in1mf2':'gbellmf',[5.70111304631562 0.146075817163115 15.9088742761993]

MF3='in1mf3':'gbellmf',[5.27201671036587 2.12265297376781 27.1760186292525]

[Input2]

Name='input2'

Range=[166 608]

NumMFs=3

MF1='in2mf1':'gbellmf',[110.51135834055 1.97107600742046 166.010965871898]

MF2='in2mf2':'gbellmf',[110.492002534101 2.09967866313238 387.011408752576]

MF3='in2mf3':'gbellmf',[110.466955802267 2.54189605124273 608.022040685593]

[Output1]

Name='output'

Range=[4.54459989895457 26.76534653]

NumMFs=9

MF1='out1mf1':'linear',[0.511577702191357 -0.0259435534809963 27.8898293558551]

MF2='out1mf2':'linear',[-1.54056988651706 0.0295482222907011 8.27704135728336]

MF3='out1mf3':'linear',[0.291381841177083 0.00117886472931029 91.8258572365462]

MF4='out1mf4':'linear',[3.80959841259244 0.0138530382333983 -63.0392838908092]

MF5='out1mf5':'linear',[0.0406835405874752 -0.00223410013612131 24.2207009374295]

MF6='out1mf6':'linear',[13.8004381791916 -0.00445174802612908 -230.86588270442]

MF7='out1mf7':'linear',[6.19697724468257 -0.0355458031411795 -167.712320597289]

MF8='out1mf8':'linear',[3.14613534456316 -0.0124703848853381 -57.2705222578772]

MF9='out1mf9':'linear',[-5.54860248088482 -0.0456793705622534 146.979760662228]

[Rules]

1 1, 1 (1) : 1

1 2, 2 (1) : 1

1 3, 3 (1) : 1

2 1, 4 (1) : 1

2 2, 5 (1) : 1

2 3, 6 (1) : 1

3 1, 7 (1) : 1

3 2, 8 (1) : 1

3 3, 9 (1) : 1

Apêndice C

Sistema de Desvio

[System]

Name='FuzzyDesvia'

Type='mamdani'

Version=2.0

NumInputs=2

NumOutputs=2

NumRules=15

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'

[Input1]

Name='dist-esq'

Range=[0 255]

NumMFs=5

MF1='mt-perto': 'trapmf', [0 0 5 15]

MF2='perto': 'trimf', [5 15 25]

MF3='medio': 'trimf', [15 25 35]

MF4='longe': 'trapmf', [25 35 45 55]

MF5='mt-longe': 'trapmf', [45 55 255 255]

[Input2]

Name='dist-dir'

Range=[0 255]

NumMFs=5

MF1='mt-perto':'trapmf',[0 0 5 15]
 MF2='perto':'trimf',[5 15 25]
 MF3='medio':'trimf',[15 25 35]
 MF4='longe':'trapmf',[25 35 45 55]
 MF5='mt-longe':'trapmf',[45 55 255 255]

[Output1]

Name='vel-esq'
 Range=[0 100]
 NumMFs=4
 MF1='mt-lento':'trapmf',[0 0 10 25]
 MF2='lento':'trapmf',[10 25 30 45]
 MF3='medio':'trapmf',[30 45 50 65]
 MF4='rapido':'trapmf',[50 65 100 100]

[Output2]

Name='vel-dir'
 Range=[0 100]
 NumMFs=4
 MF1='mt-lento':'trapmf',[0 0 10 25]
 MF2='lento':'trapmf',[10 25 30 45]
 MF3='medio':'trapmf',[30 45 50 65]
 MF4='rapido':'trapmf',[50 65 100 100]

[Rules]

1 1, 1 2 (1) : 1
 2 2, 1 2 (1) : 1
 3 3, 2 3 (1) : 1
 4 4, 3 4 (1) : 1
 5 5, 4 4 (1) : 1
 1 2, 2 1 (1) : 1
 1 3, 3 1 (1) : 1
 1 4, 4 1 (1) : 1
 1 5, 4 1 (1) : 1
 2 3, 3 1 (1) : 1
 2 4, 4 2 (1) : 1
 2 5, 4 2 (1) : 1
 3 4, 3 4 (1) : 1
 3 5, 3 4 (1) : 1

4 5, 4 4 (1) : 1

Apêndice D

Resultados Sistema de Frenagem

Tabela D.1: Frenagem - 150 Centímetros

Distância	Vel. atual	Vel. futura
150	49	48
144	48	47
140	47	45
135	45	43
130	43	40
126	40	36
122	36	29
120	29	27
117	27	27
115	27	27
112	27	27
110	27	27
108	27	27
107	27	27
104	27	27
103	27	27
100	27	27
98	27	27
96	27	27
94	27	27
92	27	27
90	27	27
88	27	27
86	27	27
84	27	27
81	27	27
80	27	27
78	27	27
76	27	27
73	27	27
71	27	27
69	27	27
67	27	27
64	27	27
63	27	27
61	27	27

58	27	27
55	27	27
53	27	27
51	27	27
48	27	27
47	27	27
41	27	27
42	27	27
39	27	27
38	27	27
36	27	27
30	27	27
31	27	27
30	27	27
27	27	25
19	25	19
24	19	22
19	22	19
22	19	22
21	22	21
20	21	20
19	20	19
17	19	18
15	18	17
15	17	17
14	17	16
14	16	17
12	17	14
11	14	13
9	13	10
10	10	0

Tabela D.2: Frenagem - 100 Centímetros

Distância	Velocidade atual	Velocidade futura
100	50	49
100	49	48
90	48	45
86	45	43
82	43	40
71	40	36
67	36	29
64	29	27
59	27	27
56	27	27
51	27	27
53	27	27
50	27	27
48	27	27
46	27	27
44	27	27
41	27	27
40	27	27
38	27	27
24	27	23
33	23	25
31	25	27
25	27	24
28	24	26
26	26	24
25	24	24
23	24	22
22	22	22
21	22	21
20	21	20
19	20	19
17	19	18
16	18	18
15	18	17
14	17	16
14	16	17
12	17	14
10	14	10
8	10	0

Tabela D.3: Frenagem - 80 Centímetros

Distância	Velocidade atual	Velocidade futura
80	50	49
81	49	48
77	48	47
73	47	45
69	45	43
63	43	40
60	40	36
58	36	29
56	29	27
53	27	27
51	27	27
49	27	27
47	27	27
44	27	27
43	27	27
40	27	27
38	27	27
33	27	27
34	27	27
31	27	27
30	27	27
27	27	25
26	25	24
24	24	23
23	23	22
21	22	21
20	21	20
20	20	20
20	20	20
19	20	19
17	19	18
15	18	17
14	17	16
14	16	17
13	17	15
12	15	15
10	15	10
10	10	0

Tabela D.4: Frenagem - 50 Centímetros

Distância	Velocidade atual	Velocidade futura
49	50	39
49	39	36
48	36	30
44	30	27
42	27	27
39	27	27
37	27	27
35	27	27
32	27	27
30	27	27
28	27	26
26	26	24
24	24	23
23	23	22
21	22	21
20	21	20
18	20	18
17	18	18
16	18	18
15	18	17
14	17	16
13	16	16
12	16	14
11	14	13
11	13	14
10	14	10
9	10	0

Tabela D.5: Frenagem - 30 Centímetros

Distância	Velocidade atual	Velocidade futura
29	50	26
19	26	19
27	19	22
25	22	24
24	24	23
21	23	21
22	21	22
21	22	21
20	21	20
20	20	20
18	20	18
16	18	18
15	18	17
14	17	16
13	16	16
11	16	12
9	12	9
9	9	0

Apêndice E

Resultados Seguidor de Linha

Tabela E.1: Seguidor de Linha - Sistema 1

Vel. Entrada Esq.	Vel. Entrada Dir	Cor	Vel. Saida Esq.	Vel. Saida Dir.
5.00	5.00	209.00	16.17	5.05
16.00	5.00	207.00	20.04	5.05
20.00	5.00	223.00	20.04	5.05
20.00	5.00	392.00	14.97	14.67
15.00	15.00	595.00	6.34	16.71
6.00	17.00	596.00	6.34	26.77
6.00	27.00	591.00	6.43	25.78
6.00	26.00	264.00	17.18	10.13
17.00	10.00	198.00	16.71	6.34
17.00	6.00	190.00	26.77	6.34
27.00	6.00	211.00	25.78	6.43
26.00	6.00	601.00	6.41	17.43
6.00	17.00	601.00	6.34	26.77
6.00	27.00	608.00	6.43	25.78
6.00	26.00	607.00	6.41	25.99
6.00	26.00	594.00	6.41	25.99
6.00	26.00	602.00	6.41	25.99
6.00	26.00	600.00	6.41	25.99
6.00	26.00	595.00	6.41	25.99
6.00	26.00	598.00	6.41	25.99
6.00	26.00	602.00	6.41	25.99
6.00	26.00	598.00	6.41	25.99
6.00	26.00	603.00	6.41	25.99

6.00	26.00	592.00	6.41	25.99
6.00	26.00	598.00	6.41	25.99
6.00	26.00	597.00	6.41	25.99
6.00	26.00	602.00	6.41	25.99
6.00	26.00	601.00	6.41	25.99
6.00	26.00	600.00	6.41	25.99
6.00	26.00	601.00	6.41	25.99
6.00	26.00	591.00	6.41	25.99
6.00	26.00	594.00	6.41	25.99
6.00	26.00	596.00	6.41	25.99
6.00	26.00	595.00	6.41	25.99
6.00	26.00	603.00	6.41	25.99
6.00	26.00	606.00	6.41	25.99
6.00	26.00	596.00	6.41	25.99
6.00	26.00	600.00	6.41	25.99
6.00	26.00	601.00	6.41	25.99
6.00	26.00	605.00	6.41	25.99
6.00	26.00	565.00	6.41	25.99
6.00	26.00	204.00	17.43	6.41
17.00	6.00	602.00	6.34	16.71
6.00	17.00	598.00	6.34	26.77
6.00	27.00	596.00	6.43	25.78
6.00	26.00	605.00	6.41	25.99
6.00	26.00	605.00	6.41	25.99
6.00	26.00	603.00	6.41	25.99
6.00	26.00	601.00	6.41	25.99
6.00	26.00	601.00	6.41	25.99
6.00	26.00	603.00	6.41	25.99
6.00	26.00	310.00	16.42	15.82
16.00	16.00	179.00	16.71	6.34
17.00	6.00	240.00	26.77	6.34
27.00	6.00	448.00	16.29	16.43
16.00	16.00	424.00	14.50	16.24
14.00	16.00	576.00	6.34	16.71
6.00	17.00	590.00	6.34	26.77
6.00	27.00	593.00	6.43	25.78
6.00	26.00	413.00	15.66	18.53
16.00	19.00	194.00	16.71	6.34
17.00	6.00	227.00	26.77	6.34

27.00	6.00	184.00	25.78	6.43
26.00	6.00	324.00	19.12	15.41
19.00	15.00	216.00	16.71	6.34
17.00	6.00	519.00	6.34	16.71
6.00	17.00	558.00	6.34	26.77
6.00	27.00	518.00	6.43	25.78
6.00	26.00	427.00	15.39	19.18
15.00	19.00	173.00	16.71	6.34
17.00	6.00	192.00	26.77	6.34
27.00	6.00	183.00	25.78	6.43
26.00	6.00	191.00	25.99	6.41
26.00	6.00	532.00	6.41	17.43
6.00	17.00	590.00	6.34	26.77
6.00	27.00	594.00	6.43	25.78
6.00	26.00	591.00	6.41	25.99
6.00	26.00	510.00	6.41	25.99
6.00	26.00	234.00	17.43	6.41
17.00	6.00	187.00	26.77	6.34
27.00	6.00	187.00	25.78	6.43
26.00	6.00	186.00	25.99	6.41
26.00	6.00	410.00	17.58	15.97
18.00	16.00	595.00	6.34	16.71
6.00	17.00	607.00	6.34	26.77
6.00	27.00	608.00	6.43	25.78
6.00	26.00	554.00	6.41	25.99
6.00	26.00	223.00	17.43	6.41
17.00	6.00	204.00	26.77	6.34
27.00	6.00	210.00	25.78	6.43
26.00	6.00	209.00	25.99	6.41
26.00	6.00	212.00	25.99	6.41
26.00	6.00	570.00	6.41	17.43
6.00	17.00	604.00	6.34	26.77
6.00	27.00	604.00	6.43	25.78
6.00	26.00	397.00	15.82	18.01
16.00	18.00	187.00	16.71	6.34
17.00	6.00	187.00	26.77	6.34
27.00	6.00	189.00	25.78	6.43
26.00	6.00	195.00	25.99	6.41

Tabela E.2: Seguidor de Linha - Sistema 2

Vel. Entrada Esq.	Vel. Entrada Dir	Cor	Vel. Saida Esq.	Vel. Saida Dir.
5.00	5.00	170.00	12.12	4.54
12.00	5.00	168.00	13.79	4.54
14.00	5.00	167.00	13.79	4.54
14.00	5.00	166.00	13.79	4.54
14.00	5.00	216.00	13.79	4.54
14.00	5.00	309.00	11.61	8.14
12.00	8.00	460.00	9.73	13.30
10.00	13.00	565.00	6.67	14.18
7.00	14.00	580.00	6.67	14.10
7.00	14.00	588.00	6.67	14.10
7.00	14.00	588.00	6.67	14.10
7.00	14.00	575.00	6.67	14.10
7.00	14.00	586.00	6.67	14.10
7.00	14.00	575.00	6.67	14.10
7.00	14.00	559.00	6.67	14.10
7.00	14.00	495.00	7.15	14.04
7.00	14.00	445.00	10.11	13.39
10.00	13.00	327.00	13.03	10.89
13.00	11.00	215.00	14.18	6.67
14.00	7.00	217.00	14.10	6.67
14.00	7.00	264.00	13.95	7.90
14.00	8.00	438.00	10.61	13.03
11.00	13.00	566.00	6.67	14.18
7.00	14.00	581.00	6.67	14.10
7.00	14.00	554.00	6.67	14.10
7.00	14.00	406.00	11.38	12.74
11.00	13.00	203.00	14.18	6.67
14.00	7.00	194.00	14.10	6.67
14.00	7.00	204.00	14.10	6.67
14.00	7.00	215.00	14.10	6.67
14.00	7.00	317.00	13.20	10.56
13.00	11.00	443.00	10.19	13.36
10.00	13.00	509.00	6.67	14.18
7.00	14.00	517.00	6.67	14.10
7.00	14.00	469.00	8.99	13.75
9.00	14.00	345.00	12.72	11.41
13.00	11.00	216.00	14.18	6.67

14.00	7.00	229.00	14.10	6.67
14.00	7.00	293.00	13.57	9.60
14.00	10.00	430.00	10.66	13.15
11.00	13.00	493.00	7.32	14.08
7.00	14.00	527.00	6.67	14.10
7.00	14.00	467.00	9.10	13.72
9.00	14.00	265.00	13.75	8.06
14.00	8.00	186.00	13.87	6.67
14.00	7.00	188.00	14.10	6.67
14.00	7.00	292.00	13.59	9.55
14.00	10.00	419.00	11.01	12.96
11.00	13.00	462.00	9.36	13.65
9.00	14.00	460.00	9.58	13.44
10.00	13.00	416.00	11.10	12.91
11.00	13.00	362.00	12.39	11.83
12.00	12.00	303.00	13.42	10.03
13.00	10.00	227.00	14.18	6.67
14.00	7.00	325.00	13.07	10.83
13.00	11.00	481.00	8.25	13.92
8.00	14.00	593.00	6.67	13.87
7.00	14.00	591.00	6.67	14.10
7.00	14.00	579.00	6.67	14.10
7.00	14.00	583.00	6.67	14.10
7.00	14.00	589.00	6.67	14.10
7.00	14.00	583.00	6.67	14.10
7.00	14.00	588.00	6.67	14.10
7.00	14.00	586.00	6.67	14.10
7.00	14.00	586.00	6.67	14.10
7.00	14.00	582.00	6.67	14.10
7.00	14.00	589.00	6.67	14.10
7.00	14.00	600.00	6.67	14.10
7.00	14.00	587.00	6.67	14.10
7.00	14.00	603.00	6.67	14.10
7.00	14.00	507.00	6.67	14.10
7.00	14.00	337.00	12.86	11.19
13.00	11.00	468.00	9.04	13.73
9.00	14.00	305.00	13.27	10.20
13.00	10.00	182.00	14.18	6.67

14.00	7.00	247.00	14.10	6.67
14.00	7.00	335.00	12.90	11.13
13.00	11.00	425.00	10.83	13.07
11.00	13.00	551.00	6.67	14.18
7.00	14.00	574.00	6.67	14.10
7.00	14.00	577.00	6.67	14.10
7.00	14.00	553.00	6.67	14.10
7.00	14.00	505.00	6.67	14.10
7.00	14.00	410.00	11.27	12.81
11.00	13.00	321.00	13.13	10.70
13.00	11.00	235.00	14.18	6.67
14.00	7.00	223.00	14.10	6.67
14.00	7.00	272.00	13.86	8.45
14.00	8.00	411.00	11.32	12.73
11.00	13.00	478.00	8.45	13.87
8.00	14.00	468.00	9.32	13.40
9.00	13.00	390.00	11.79	12.43
12.00	12.00	251.00	14.16	6.77
14.00	7.00	203.00	14.10	6.67
14.00	7.00	336.00	12.88	11.16
13.00	11.00	470.00	8.93	13.76
9.00	14.00	523.00	6.67	13.95
7.00	14.00	521.00	6.67	14.10
7.00	14.00	435.00	10.49	13.23
10.00	13.00	283.00	13.72	9.10

Tabela E.3: Seguidor de Linha - Sistema ANFIS

Vel. Entrada Esq.	Vel. Entrada Dir	Cor	Vel. Saida Esq.	Vel. Saida Dir.
5.00	5.00	580.00	5.36	13.50
5.00	13.00	582.00	5.37	13.55
5.00	14.00	589.00	5.40	14.16
5.00	14.00	587.00	5.39	14.17
5.00	14.00	584.00	5.37	14.17
5.00	14.00	589.00	5.40	14.16
5.00	14.00	588.00	5.40	14.16
5.00	14.00	579.00	5.36	14.19
5.00	14.00	535.00	6.24	14.29
6.00	14.00	368.00	15.99	12.38
16.00	12.00	194.00	17.38	6.21
17.00	6.00	202.00	25.10	6.08
25.00	6.00	191.00	22.10	6.10
22.00	6.00	195.00	15.83	6.09
16.00	6.00	317.00	15.44	11.30
15.00	11.00	486.00	8.62	14.08
9.00	14.00	536.00	6.34	14.29
6.00	14.00	482.00	10.12	14.06
10.00	14.00	222.00	14.64	6.39
15.00	6.00	211.00	16.92	6.09
17.00	6.00	198.00	25.13	6.08
25.00	6.00	351.00	18.43	12.78
18.00	13.00	576.00	5.94	13.56
6.00	14.00	570.00	6.01	14.22
6.00	14.00	580.00	6.11	14.19
6.00	14.00	583.00	6.15	14.18
6.00	14.00	578.00	6.09	14.19
6.00	14.00	583.00	6.15	14.18
6.00	14.00	584.00	6.16	14.17
6.00	14.00	577.00	6.08	14.20
6.00	14.00	581.00	6.12	14.18
6.00	14.00	580.00	6.11	14.19
6.00	14.00	588.00	6.22	14.16

6.00	14.00	587.00	6.21	14.17
6.00	14.00	587.00	6.21	14.17
6.00	14.00	589.00	6.23	14.16
6.00	14.00	588.00	6.22	14.16
6.00	14.00	583.00	6.15	14.18
6.00	14.00	589.00	6.23	14.16
6.00	14.00	586.00	6.19	14.17
6.00	14.00	583.00	6.15	14.18
6.00	14.00	573.00	6.04	14.21
6.00	14.00	535.00	6.23	14.29
6.00	14.00	540.00	6.12	14.28
6.00	14.00	452.00	12.66	13.69
13.00	14.00	318.00	12.90	11.04
13.00	11.00	588.00	6.51	15.16
7.00	15.00	589.00	6.76	15.62
7.00	16.00	591.00	6.80	16.69
7.00	17.00	590.00	6.78	26.76
7.00	27.00	590.00	6.78	25.97
7.00	26.00	592.00	6.82	26.13
7.00	26.00	518.00	6.59	25.47
7.00	25.00	475.00	9.03	20.03
9.00	20.00	493.00	7.20	27.17
7.00	27.00	507.00	7.04	25.30
7.00	25.00	525.00	6.40	25.95
6.00	26.00	580.00	6.11	26.32
6.00	26.00	589.00	6.23	26.18
6.00	26.00	584.00	6.16	26.26
6.00	26.00	591.00	6.27	26.15
6.00	26.00	581.00	6.12	26.30
6.00	26.00	584.00	6.16	26.26
6.00	26.00	579.00	6.10	26.33
6.00	26.00	582.00	6.14	26.29
6.00	26.00	588.00	6.22	26.20
6.00	26.00	580.00	6.11	26.32
6.00	26.00	581.00	6.12	26.30
6.00	26.00	481.00	10.22	21.92
10.00	22.00	203.00	14.73	0.00

15.00	0.00	582.00	6.13	0.00
6.00	0.00	579.00	6.10	0.00
6.00	0.00	587.00	6.21	0.00
6.00	0.00	584.00	6.16	0.00
6.00	0.00	586.00	6.19	0.00
6.00	0.00	586.00	6.19	0.00
6.00	0.00	534.00	6.26	0.00
6.00	0.00	427.00	14.02	16.05
14.00	16.00	269.00	14.54	9.61
15.00	10.00	189.00	16.69	6.85
17.00	7.00	498.00	6.32	17.43
6.00	17.00	592.00	6.28	26.78
6.00	27.00	581.00	6.12	26.14
6.00	26.00	588.00	6.22	26.20
6.00	26.00	586.00	6.19	26.23
6.00	26.00	588.00	6.22	26.20
6.00	26.00	586.00	6.19	26.23
6.00	26.00	582.00	6.14	26.29
6.00	26.00	579.00	6.10	26.33
6.00	26.00	583.00	6.15	26.27
6.00	26.00	582.00	6.14	26.29
6.00	26.00	578.00	6.09	26.34
6.00	26.00	195.00	17.63	6.01
18.00	6.00	277.00	18.63	8.52
19.00	9.00	280.00	17.75	8.41
18.00	8.00	302.00	14.19	9.70
14.00	10.00	215.00	14.85	6.80
15.00	7.00	211.00	16.92	6.80
17.00	7.00	197.00	25.14	6.83
25.00	7.00	211.00	22.71	6.80

Apêndice F

Resultados Sistemas de Desvio

Tabela F.1: Desvio - Distância Inicial 30 centímetros

30,00	30,00	15,00	15,00	37,50	67,10
30,00	30,00	67,00	37,00	37,50	67,10
26,00	25,00	67,00	37,00	27,50	47,50
25,00	21,00	47,00	27,00	27,50	47,50
255,00	19,00	47,00	27,00	50,00	50,00
255,00	255,00	50,00	50,00	78,78	78,78
255,00	255,00	78,00	78,00	78,78	78,78
214,00	255,00	78,00	78,00	78,78	78,78
14,00	255,00	78,00	78,00	78,46	26,26
6,00	22,00	26,00	78,00	39,29	12,86

Tabela F.2: Desvio - Distância Inicial 50 centímetros

52,00	51,00	15,00	15,00	71,15	77,43
51,00	51,00	77,00	71,00	69,69	77,43
43,00	45,00	77,00	69,00	47,50	78,78
32,00	36,00	78,00	47,00	47,50	77,78
24,00	28,00	77,00	47,00	40,60	56,52
22,00	19,00	56,00	40,00	30,71	33,01
118,00	17,00	33,00	30,00	50,00	50,00
255,00	255,00	50,00	50,00	78,78	78,78
222,00	255,00	78,00	78,00	78,78	78,78
216,00	112,00	78,00	78,00	78,78	78,78
7,00	20,00	78,00	78,00	34,56	16,67

Referências Bibliográficas

- [1] Amazon. Preço do robô no site da amazon. <https://www.amazon.com/LEGO-4494799-Mindstorms-NXT/dp/B000E4FDAE>. Acessado em 17/06/2018.
- [2] Michelle Andrade and Maria Alice Prudêncio Jaques. Estudo comparativo de controladores de mamdani e sugeno para controle de tráfego em interseções isoladas. *TRANSPORTES*, 16(2):24–31, 2008.
- [3] Leonardo de Paula Batista Benevides. Robótica móvel inteligente.
- [4] Bluetooth. Protocol specifications. <https://www.bluetooth.com/specifications/protocol-specifications>. Acessado em 13/06/2018.
- [5] Matthew Claudel and Carlo Ratti. Full speed ahead: How the driverless car could transform cities. *McKinsey & Company, August*, 14, 2015.
- [6] Olimpíada Brasileira de Robótica. O que é a obr. <http://www.obr.org.br/o-que-e-a-obr>. Acessado em 27/06/2018.
- [7] Arthur Rodrigues Stilben et al. Variation of a car speed using fuzzy system. 2016.
- [8] Pierre Yves Glorennec. Fuzzy logic-based navigation for an autonomous robot. *IFAC - Intelligent Components for Autonomous and Semi-Autonomous Vehicles*, 1995.
- [9] R. R. Gudwin. Novas fronteiras na inteligência artificial e na robótica. *4.º Congresso Temático de Dinâmica, Controle e Aplicações*, 2005.
- [10] LEGO. History of lego robotics. <https://www.lego.com/fi-fi/mindstorms/history>. Acessado em 13/06/2018.
- [11] LEGO. Preço do robô no site da lego. <https://shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547>. Acessado em 17/06/2018.
- [12] Ebrahim H Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the institution of electrical engineers*, volume 121, pages 1585–1588. IET, 1974.
- [13] MathWorks. anfis. <https://www.mathworks.com/help/fuzzy/anfis.html>. Acessado em 29/06/2018.
- [14] MATLAB. Site do matlab. <https://www.mathworks.com/products/matlab.html>. Acessado em 17/06/2018.

- [15] Scott Pawlikowski. Development of a fuzzy logic speed and steering control system for an autonomous vehicle. *MSc Research Project, Department of Mechanical Engineering, University of Cincinnati*, 1999.
- [16] RWTH. Rwth - mindstorms nxt toolbox for matlab. <http://www.mindstorms.rwth-aachen.de/>. Acessado em 13/06/2018.
- [17] Gabriel Drummond de M Simonsen. Resgate utilizando robô real. *São Paulo*, 2011.
- [18] Rodrigo Siqueira-Batista, Camila Ribeiro Souza, Polyana Mendes Maia, and Savio Lana Siqueira. Robotic surgery: bioethical aspects. *ABCD. Arquivos Brasileiros de Cirurgia Digestiva (São Paulo)*, 29(4):287–290, 2016.
- [19] Tomohiro Takagi and Michio Sugeno. Derivation of fuzzy control rules from human operator’s control actions. *IFAC Proceedings Volumes*, 16(13):55–60, 1983.
- [20] UOL. Acompanhe a greve de caminhoneiros e seus reflexos. <https://economia.uol.com.br/ao-vivo/2018/05/25/greve-de-caminhoneiros.htm>. Acessado em 25/06/2018.
- [21] C. et al. Valadão. Analysis of the use of a robot to improve social skills in children with autism spectrum disorder. *Research on Biomedical Engineering*, 32(2):161–175, 2016.
- [22] Athanasios Vourvopoulos and Fotis Liarokapis. Brain-controlled nxt robot: Tele-operating a robot through brain electrical activity. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2011 Third International Conference on*, pages 140–143. IEEE, 2011.
- [23] L. A. Zadeh. Fuzzy sets. *Information and control*, (8):338 – 353, 1965.