



Cristhian Julian Gómez Lizcano

**Controle Fuzzy de um Sistema Auxiliar de
Navegação de um Robô Ambiental Híbrido**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica da PUC-Rio.

Orientadora: Marley Maria Bernardes Rebuzzi Vellasco
Co-orientador: Ricardo Tanscheit

Rio de Janeiro
Setembro de 2015



Cristhian Julian Gómez Lizcano

Controle Fuzzy de um Sistema Auxiliar de Navegação de um Robô Ambiental Híbrido

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Profa. Marley Maria Bernardes Rebuzzi Vellasco
Orientador
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Ricardo Tanscheit
Co-Orientador
Departamento de Engenharia Elétrica – PUC-Rio

Profa. Karla Tereza Figueiredo Leite
Departamento de Engenharia Elétrica – PUC-Rio

Prof. José Franco Machado do Amaral
UERJ

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico- PUC-Rio

Rio de Janeiro, 16 de setembro de 2015

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Cristhian Julian Gómez Lizcano

Graduou-se em Engenharia Mecatrônica pela Universidade Santo Tomas de Aquino - Bucaramanga - Colômbia Ustabuca em 2012.

Ficha Catalográfica

Lizcano, Cristhian Julian Gómez

Controle fuzzy de um sistema auxiliar de navegação de um Robô ambiental híbrido / Cristhian Julian Gómez Lizcano ; orientadora: Marley Maria Bernardes Rebuzzi Vellasco ; co-orientador: Ricardo Tanscheit. – 2015.

101 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2015.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Lógica Fuzzy. 3. Navegação. 4. Robótica. 5. Controle Fuzzy. I. Vellasco, Marley Maria Bernardes Rebuzzi. II. Tanscheit, Ricardo. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

Agradecimentos

Ao CNPq e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

À minha orientadora Professora Marley, pelo apoio e parceria para a realização deste trabalho.

Ao meu co-orientador Professor Ricardo Tanscheit pelo apoio e parceria para a realização deste trabalho.

Aos meus pais, Alma Piedad Lizcano Paez e Jorge Alberto Gomez Piraquive, pela educação, atenção e carinho.

A todos os amigos da PUC-Rio.

A todos os amigos e familiares que de uma forma ou de outra me estimularam e me ajudaram.

Aos professores que participaram da Comissão examinadora.

Resumo

Gomez, Cristhian. Vellasco, Marley Maria Bernardes Rebuzzi. Tanscheit, Ricardo. **Controle Fuzzy de um Sistema Auxiliar de Navegação de um Robô Ambiental Híbrido**. Rio de Janeiro, 2016. 101p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Nas últimas décadas o avanço tecnológico tem atingido altos níveis de desenvolvimento, sendo as técnicas de inteligência computacional um dos principais campos em expansão devido à sua aplicabilidade nas diferentes áreas industriais. Uma das principais técnicas com aplicabilidade no setor da robótica é a Lógica Fuzzy, que permite aumentar as características de autonomia dos robôs. Atualmente a indústria brasileira vem desenvolvendo novos métodos robotizados para as tarefas de monitoramento e inspeção de gasodutos. No caso particular do gasoduto Coari-Manaus, foi desenvolvido o Robô Ambiental Híbrido Médio (RAHM), o qual possui um Sistema Primário de Navegação (SPN), mas não um Sistema Auxiliar de Navegação (SAN) para uso em caso de falha. Este trabalho tem como principal objetivo desenvolver um Sistema de Navegação Auxiliar controlado através da Lógica Fuzzy, de forma a auxiliar a navegação autônoma do robô em caso de falha do SPN. O trabalho envolveu o projeto da eletrônica e de um sistema de inferência fuzzy para oferecer um controle adequado na navegação do robô em casos de emergência. O SAN é avaliado através de um estudo de caso comparativo, confirmando os benefícios que a Lógica Fuzzy oferece para o Sistema Auxiliar de Navegação.

Palavras-Chave

Lógica fuzzy; navegação; robótica; controle fuzzy.

Abstract

Gómez, Cristhian. Vellasco, Marley Maria Bernardes Rebuzzi (Advisor). Tanscheit, Ricardo (Co – Advisor). **Fuzzy Control of an Auxiliary Navigation System for a Hybrid Ambient Robot.** Rio de Janeiro, 2015. 101p. MSc. Dissertation – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Technological progress has reached high levels of development in the last decades and computational intelligence techniques have been one of the main expanding fields due to their applicability in different industrial areas. One of those techniques that can be used in the field of robotics is Fuzzy Logic, which contributes to an increase in the autonomy of robots. Brazilian industry has been developing new robotics methods for monitoring and inspection tasks of pipelines. In the particular case of Coari-Manaus pipeline the Environmental Hybrid Medium Robot (RAHM) has been developed. This robot has a primary navigation system (PNS), but lacks an Auxiliary Navigation System (ANS) to be used in the case of failure. This dissertation has as its main objective developing an Auxiliary Navigation System controlled through Fuzzy Logic to help autonomous navigation in the case of the PNS failure. The work has involved the electronic project and the design of a fuzzy inference system for an adequate control of the robot navigation of Robot in an emergency. The ANS is evaluated through a comparative case study and results confirm the benefits from using Fuzzy Logic in the design of the Auxiliary Navigation System.

Keywords

Fuzzy logic; navigation; robotics; fuzzy control.

Sumário

1. Introdução	13
1.1. Motivação	13
1.2. Objetivos	14
1.2.1. Objetivo Geral	14
1.2.2. Objetivos Específicos	14
1.3. Descrição do Trabalho	14
1.4. Organização da Dissertação	15
2. Robô Ambiental Híbrido	16
3. Projeto Eletrônico do Robô Ambiental Híbrido Médio	19
3.1. Introdução	19
3.2. Descrição Geral	19
3.2.1. Sensores	20
3.2.2. Processador	22
3.2.2.1. Configuração e Cabeamento da Placa Arduino Mega	24
3.2.2.2. Adaptação do Módulo Ethernet no Arduino Mega	27
3.2.3. Módulo de Comunicação	27
3.2.3.1. Seleção e Configuração do Switch	27
3.2.3.2. Rádio Comunicação	29
3.2.4. Módulo de Atuação	32
3.2.4.1. Atuadores EPOS dos Motores DC de Tração	32
3.2.4.2. Motores DC de Tração	34
3.3. Diagrama Geral do Projeto Eletrônico	35
4. Sistema Auxiliar de Navegação do RAHM	36
4.1. Sistema de Inferência Fuzzy Geral	36
4.1.1. Sistema Intermediário	38
4.1.1.1. Variáveis de Entrada do SIF Intermediário	38

4.1.1.2. Variáveis de Saída	40
4.1.1.3. Base de Regras do SIF Intermediário	40
4.1.2. Sistema de Inferência Fuzzy 2 (Sistema de Controle).	40
4.1.2.1. Variáveis de Entrada do Sistema de Controle	41
4.1.2.2. Variáveis de Saída do SIF2 (Sistema de Controle)	41
4.1.2.3. Base de Regras do SIF 2 (Sistema de Controle)	42
4.1.3. Defuzzificação	42
4.2. Implementação do SIF em Visual Studio	42
4.2.1. Biblioteca Dotfuzzy	44
5. Estudo de Casos	45
5.1. Tópicos de Comparação	45
5.1.1. Custo Financeiro	45
5.1.2. Custo Computacional	46
5.1.3. Fluxo de Dados pela Rede	47
5.1.4. Tempo	48
5.1.5. Consumo de Corrente	49
5.2. Testes em Ambientes Controlados	50
5.2.1. Comparação dos Sistemas de Navegação	57
6. Conclusões e Trabalhos Futuros	60
6.1. Conclusões	60
6.2. Trabalhos Futuros	60
7. Referências Bibliográficas	62
Anexo A– Configuração Placa Conversora RS232/Ethernet	65
Anexo B– Programação do SIF com a Biblioteca Dotfuzzy no Visual Studio.	68
Anexo C–Base de Regras dos Sistemas de Inferência Fuzzy	70

Lista de Figuras

Figura 1. Protótipo RAH 2011. Fonte Relatório Ativatec.	16
Figura 2. Versão atual do RAHM.	17
Figura 3. Diagrama Geral Projeto Eletrônico	19
Figura 4. Sensores de ultrassom e angular.	20
Figura 5. Faixa de trabalho do Max Sonar.	21
Figura 6. Localização dos sensores de ultrassom.	21
Figura 7. Distribuição dos sensores na zona frontal.	22
Figura 8. Localização do sensor angular.	22
Figura 9. Placa Arduino Mega.	23
Figura 10. Placa de Redundância Arduino.	24
Figura 11. Esquema de ligações da placa Arduino Mega.	25
Figura 12. Conexão detalhada do sensor com o Arduino.	25
Figura 13. Conexões dos sensores no RAHM.	26
Figura 14. Módulo Ethernet	27
Figura 15. Switch O-ring IGS-1050A	28
Figura 16. Diagrama de conexões do RAHM	29
Figura 17. Radio de comunicação RAJANT da WETSCOM	30
Figura 18. Teste de rádios RAJAN no Bosque da Barra.	31
Figura 19. Teste de rádios RAJAN na Lagoa Rodrigo de Freitas.	32
Figura 20. Atuadores EPOS motores de tração.	33
Figura 21. Modulo RS232/Ethernet Zuchi.	33
Figura 22. Teste serial Ethernet com as controladoras EPOS.	34
Figura 23. Motores de Tração.	34
Figura 24. Esquema Geral do Projeto Eletrônico.	35
Figura 25. Sistema de Inferência Fuzzy geral.	37
Figura 26. Pares de motores.	37
Figura 27. Sistema intermediário.	38
Figura 28. Distância ao Obstáculo.	39
Figura 29. Ângulo de desvio.	39
Figura 30. Sistema de Inferência 2 (Sistema de Controle).	41

Figura 31. Variável de saída.	42
Figura 32. Interface Gráfica de monitoramento.	44
Figura 33. Monitoramento da memória RAM.	47
Figura 34. Monitoramento do fluxo de dados.	48
Figura 35. Experimento 1.	51
Figura 36. Experimento 2.	51
Figura 37. Experimento 3	52
Figura 38. Experimento 4.	53
Figura 39. Experimento 5.	53
Figura 40. Experimento 6.	54
Figura 41. Experimento 7.	55
Figura 42. Experimento 8.	55
Figura 43. Experimento 9.	56
Figura 44. Experimento 10.	57
Figura 45. Software da placa conversora.	65
Figura 46. Ferramenta da placa conversora (Parâmetro Serial).	67
Figura 47. Programação da entrada do SIF com Dotfuzzy.	68
Figura 48. Programação da Saída do SIF com Dotfuzzy.	68
Figura 49. Definição de saídas e entradas.	68
Figura 50. Definição das regras do sistema.	69
Figura 51. Defuzzificação	69

Lista de Tabelas

Tabela 1 Comparação entre o RAHM 2011 e o RAHM 2015.	18
Tabela 2 Características gerais dos motores de tração.	35
Tabela 3 Custo financeiro dos sistemas.	46
Tabela 4 Consumo de corrente SAN.	49
Tabela 5 Consumo de corrente do SPN.	50
Tabela 6 Comparação entre o SAN e o SPN.	58

Glossário

As seguintes são as siglas utilizadas ao longo da presente dissertação :

- SAN: Sistema Auxiliar de Navegação.
- SPN: Sistema Principal de Navegação.
- SIF: Sistema de Inferência Fuzzy.
- RAH: Robô Ambiental Híbrido.
- RAHM: Robô Ambiental Híbrido Médio.
- PM1: Par de motores 1.
- PM2: Par de motores 2.
- SIF1: Sistema de Inferência 1.
- SIF2: Sistema de Inferência 2.
- CE: Computador Embarcado.
- CR: Computador Remoto.

1. Introdução

1.1. Motivação

A robótica é um novo e crescente campo de pesquisa científica que vem atraindo cada vez mais o olhar da indústria global e, em particular, da indústria brasileira. Na área de combustíveis fósseis, por exemplo, as principais indústrias brasileiras vêm utilizando a robótica na obtenção de combustíveis de forma a minimizar os riscos humanos e ambientais (Ferreira et al., 2012).

O gasoduto Coari-Manaus, localizado no estado de Amazonas, necessita de constante monitoramento para avaliação do impacto ambiental que vem produzindo no ambiente circundante. Até recentemente, tal monitoramento era realizado manualmente por um operário, que tinha que enfrentar situações adversas do ambiente selvático (Salvi dos Reis and Góes, 2007). Em 2005, a Petrobras investiu no desenvolvimento de uma nova metodologia de monitoramento do gasoduto que permitisse diminuir ou eliminar o risco humano. O CENPES/PETROBRAS, em parceria com a Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) desenvolveu, de 2005 a 2011, um Robô Ambiental Híbrido (RAH) com as seguintes funcionalidades: identificação e superação de obstáculos, sistema de navegação local, locomoção na água e nos vários tipos de terrenos encontrados na Amazônia (terra firme, pântanos, areia, etc.), e coleta de informações da flora e fauna local. Obtiveram-se resultados satisfatórios nos testes realizados na Amazônia ao longo do período de desenvolvimento do protótipo (Barral et al., 2013).

Em busca da melhora contínua do RAH, o CENPES/PETROBRAS assinou um termo de cooperação tecnológica com a empresa ATIVA Tecnologia & Desenvolvimento para a construção do Robô Ambiental Híbrido Médio (RAHM) como produto (Ferreira, 2010).

O RAHM possui um Sistema Primário de Navegação (SPN) que lhe fornece autonomia total. Entretanto, em caso de falha, não há sistema de navegação de emergência que permita ao robô se locomover de forma autônoma. Esta dissertação tem como finalidade o desenvolvimento de um Sistema Auxiliar de Navegação (SAN), baseado na

Lógica Fuzzy, capaz de oferecer autonomia ao RAHM, consumindo menos energia e com um custo computacional e financeiro mais baixo.

1.2. Objetivos

1.2.1. Objetivo Geral

- Desenvolver um sistema de controle baseado na Lógica Fuzzy para o Sistema Auxiliar de Navegação do Robô Ambiental Híbrido Médio, com as seguintes características principais:
 - Consumo de energia mínimo,
 - Custo financeiro e computacional baixo
 - Simplicidade no cumprimento do objetivo geral.

1.2.2. Objetivos Específicos

- Desenvolver o hardware do Sistema Auxiliar de Navegação (SAN) do Robô Ambiental Híbrido Médio, especificando os equipamentos necessários para o seu bom funcionamento;
- Desenvolver o Sistema de Inferência Fuzzy para o Sistema Auxiliar de Navegação (SAN) do RAHM;
- Desenvolver uma interfase gráfica para o monitoramento do Sistema de Inferência Fuzzy (SIF);
- Avaliar o Sistema de Inferência Fuzzy por meio de estudo de caso comparativo (dez cenários controlados).

1.3. Descrição do Trabalho

Este trabalho foi elaborado seguindo as seguintes etapas:

- Estudo de técnicas de modelagem aplicadas ao controle de navegação de robôs;
- Estudo sobre modelos de Lógica Fuzzy para o controle de navegação de robôs;
- Pesquisa sobre os diferentes equipamentos eletrônicos, mecânicos e de comunicação, necessários para o desenvolvimento do hardware do SAN.

- Programação do SIF no computador remoto, acompanhado da interface gráfica de monitoramento.
- Estudo de casos.

Na primeira etapa do trabalho, foi realizado um levantamento bibliográfico referente às áreas de robótica e de Lógica Fuzzy. A seguir, foi efetuado o projeto eletrônico e foram definidos os diferentes equipamentos a ser instalados no RAHM.

O SIF foi estruturado e desenvolvido para ser implementado em um computador remoto em constante comunicação com o RAHM. Desenvolveu-se também uma interface gráfica para permitir o monitoramento em tempo real de todos os dados de entrada adquiridos pelo hardware e da informação de saída fornecida ao RAHM pelo controlador.

Finalmente, realizou-se um estudo de caso com dez cenários controlados (distribuições de obstáculos) de forma a comparar e avaliar o Sistema Auxiliar de Navegação desenvolvido com o Sistema Principal de Navegação já existente.

1.4. Organização da Dissertação

Esta dissertação está organizada em cinco capítulos adicionais, conforme segue.

O Capítulo 2 apresenta uma introdução geral ao RAHM, fornecendo os conceitos fundamentais para o entendimento do protótipo desenvolvido.

O Capítulo 3 apresenta o desenvolvimento completo do hardware necessário para a coleta de variáveis de entrada do SIF e para a recepção de suas saídas.

O Capítulo 4 apresenta detalhadamente a estrutura do sistema de Inferência Fuzzy (SIF), descrevendo os seus módulos principais: variáveis de entradas e de saída, base de regras e o sistema de inferência propriamente dito.

O Capítulo 5 aborda o estudo de caso, realizado em um ambiente controlado com dez distribuições diferentes de obstáculos, com a finalidade de comparar os dois sistemas de navegação: o principal, já existente, e o auxiliar, baseado na Lógica Fuzzy.

Finalmente, no Capítulo 6 apresentam-se as conclusões e os possíveis trabalhos futuros.

2. Robô Ambiental Híbrido

Devido à necessidade de monitoramento ambiental do gasoduto Coari – Manaus, o Laboratório de Robótica do CENPES/PETROBRAS, em parceria com a Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) desenvolveu de 2005 a 2011 o Robô Ambiental Híbrido (RAH), que deveria obedecer aos requisitos exigidos pelo ambiente amazônico, com capacidade de se movimentar na lama, na água e em terrenos irregulares, além de possuir equipamentos para a coleta de informação do ambiente.

O Robô Ambiental Híbrido (RAH) foi construído de forma a comprovar a viabilidade técnica e operacional da utilização de robôs móveis para o monitoramento do gasoduto Coari-Manaus (Figura 1). Ao longo dos testes realizados na floresta amazônica, O RAH demonstrou ser capaz de se deslocar nos vários terrenos lá encontrados (terra firme, pântano, lama, areia e brejos) e também na água – podendo ser empregado na coleta de amostras da fauna e flora local –, de efetuar medições de parâmetros físico-químicos do meio e ainda de detectar níveis de gás metano em regiões alagadas por construções de hidroelétricas. O RAH é tele-operado: o operador envia comandos remotamente e o veículo robótico executa as missões em campo.

Em função do bom desempenho apresentado pelo protótipo nos testes (Barral et al., 2013), foi firmado um termo de cooperação tecnológica entre o CENPES/PETROBRAS e a empresa ATIVA Tecnologia & Desenvolvimento para a construção de um Robô Ambiental Híbrido como produto (Ferreira, 2010).



Figura 1. Protótipo RAH 2011. Fonte Relatório Ativatec.

O desenvolvimento da nova versão do RAH apresenta melhorias nas partes mecânica e eletrônica e, principalmente, na sua capacidade de navegação autônoma. Esta nova versão, mostrada na Figura 2, é denominada Robô Ambiental Híbrido Médio (RAHM), em que a palavra “Médio” indica que o RAHM é um pouco menor do que o RAH. Ele deve ser capaz de se locomover autonomamente, de se esquivar de obstáculos e de detectar rotas mais adequadas para chegar aos objetivos estabelecidos.

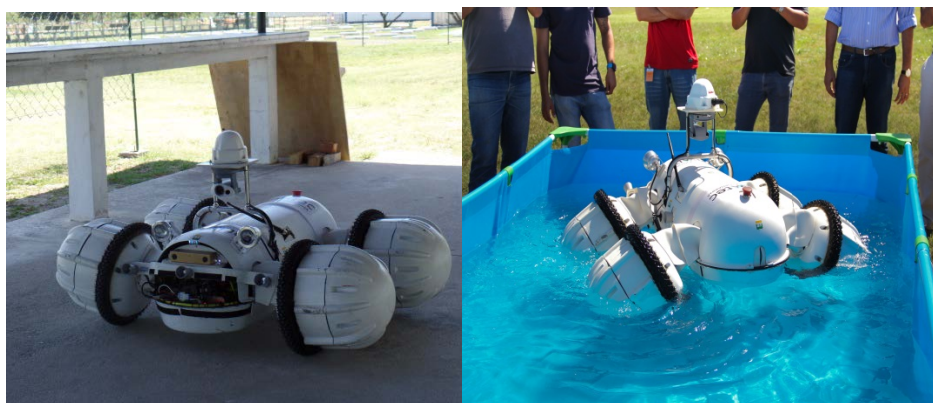


Figura 2. Versão atual do RAHM.

A

Tabela 1 apresenta uma comparação entre as características do RAHM e do RAH (versão desenvolvida até 2011).

RAH (2011)	RAHM
Modo de operação	
Tele-operado via joystick e rádio ponto-a-ponto;	Manual (malha aberta com sistema de segurança para evitar colisões) e navegação autônoma.
Localização	
Via GPS (posição) e unidade de medida inercial (orientação);	Sensor Sistema de Navegação Inercial (INS) com GPS acoplado e utilização de filtro de <i>Kalman</i> interno para calcular a posição e orientação do robô.
Mapeamento	
Sensores de infravermelho e imagens enviadas ao operador;	Sensor de varredura a laser 3D, câmera estéreo, e sensores de ultrassom;
Aquisição de dados físico-químicos	
Sonda e sensor de gás metano	Sonda e sensor de gás metano
Baterias	
Baterias NiMh e sensores de corrente elétrica dos motores e tensão das baterias;	Ion-polímero com alta densidade e capacidade energética. PoE (Power over ethernet)

Equipamento de obtenção de imagens	
Câmera IP com pan-tilt e utilização de um codec para digitalização e compactação (H.264);	Câmera IP com grande campo de visão, zoom óptico, visão noturna, até 60 quadros por segundo, padrão de compactação de vídeo H.264 e áudio embarcado; montada em uma unidade pan-tilt com estabilização inercial;
Equipamentos de Emergência	
Chave para desligamento geral do sistema no robô e watchdog em software para desligamento dos motores, quando da perda do sinal de rádio;	Transmissão de dados via mensagens SMS utilizando rede celular em caso da perda do rádio, Inter travamento de segurança para ligar e desligar os equipamentos do robô, operação manual com acionamento direto dos drivers das rodas em caso de defeito do computador embarcado;
Independência de sistemas mecânicos nas Pernas do RAHM	
Suspensão e sistema de cambagem acoplados;	Desacoplamento entre o movimento da suspensão e cambagem e maior curso da suspensão;
Transmissão de dados	
Sistema de rádio ponto-a-ponto;	Rádio ethernet 900MHZ de longo alcance (suporte a qualidade de serviço) e repetidores, junto com switch gerenciável para rede interna do robô;
Impermeabilidade	
Necessidade de vedação do robô;	Equipamentos com proteção IP67 (permite imersão por 30 minutos a 1 metro) ou encapsulados para vedação;
Sistema de transmissão mecânico	
Utilização de sistema de redução com razão de transmissão constante;	Sistema de transmissão continuamente variável (CVT) para melhorar o funcionamento do robô nos meios encontrados (água e terra);

Tabela 1 Comparação entre o RAHM 2011 e o RAHM 2015.

Quanto à impermeabilidade, a proteção IP67 diz respeito à norma internacional IEC 60529 “Graus de Proteção”. O grau de proteção IP67 indica o seguinte:

- As letras “IP” são as siglas em inglês de Ingress Protection.
- O valor “6” no primeiro dígito descreve o nível de proteção contra o pó.
- O valor “7” no segundo dígito descreve o nível de proteção contra os líquidos

Para o caso do RAHM, todos os equipamentos devem apresentar filtragem total de poeira e imersão completa a um metro de profundidade durante 30 minutos.

3. Projeto Eletrônico do Robô Ambiental Híbrido Médio

3.1. Introdução

Como foi mencionado, o RAHM possui dois sistemas de navegação – o SPN e o SAN –, com o objetivo de se ter um maior grau de segurança em casos de emergência. O SAN é um sistema de controle de navegação baseado na Lógica Fuzzy, implementado no computador remoto, que recebe e envia informações para o robô. Este capítulo apresenta, de forma detalhada, o hardware desenvolvido para o sistema de navegação auxiliar, envolvendo a especificação dos sensores e atuadores necessários para um bom desempenho do sistema.

3.2. Descrição Geral

O diagrama geral do projeto eletrônico é apresentado na Figura 3.

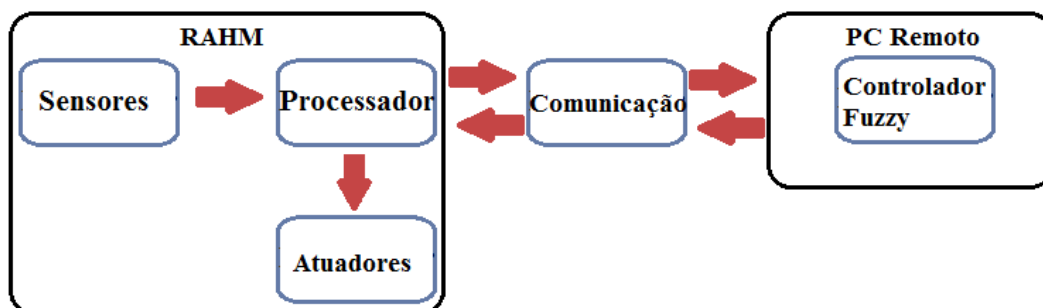


Figura 3. Diagrama Geral Projeto Eletrônico

O projeto é composto por dois módulos gerais: o RAHM, contendo os equipamentos de aquisição e processamento de informação, os atuadores e o computador remoto onde o Controlador Fuzzy é implementado. O módulo de sensores é encarregado de adquirir a informação do entorno do robô; o processador tem a função de codificar e interpretar dessa informação. O módulo de comunicação é responsável pela troca de informações

entre o robô e o computador, de forma que os atuadores comprem com a função de locomover o robô de acordo com a saídas fornecidas pelo Controlador Fuzzy.

3.2.1. Sensores

Para se locomover autonomamente, o robô necessita de sensores de obstáculos, direção, etc. Na detecção de obstáculos, fez-se uso de sensores do tipo HRXL Max Sonar da série WR_MB7360, que é ótimo para identificar obstáculos dentro sua faixa de detecção. Adicionalmente, fez-se uso do sensor angular Blade360, que possui a proteção IP67, para determinar a orientação do robô em relação ao objetivo estabelecido. Ambos os sensores apresentam baixo custo e baixo consumo de corrente. A Figura 4 mostra estes dois sensores: o de ultrassom à esquerda e o angular à direita.



Figura 4. Sensores de ultrassom e angular.

O Max sonar possui um alcance máximo de 5 metros e tem uma zona morta de 30 centímetros. Pode se comunicar através de protocolo serial RS232 ou através de um pino que possui um sinal de tensão de saída analógico de 0 a 5 volts. É alimentado com 5 volts. Observa-se, na Figura 5, a faixa de trabalho do Max Sonar em função da tensão de alimentação.

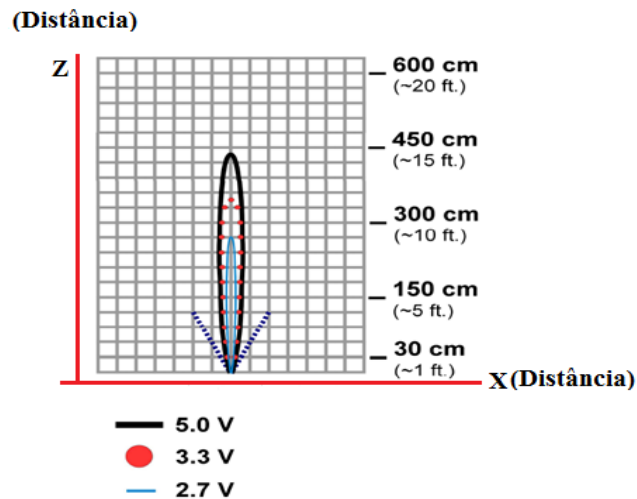


Figura 5. Faixa de trabalho do Max Sonar.

Após a realização de vários testes, com diferentes configurações de número e localizações dos sensores, estabeleceu-se uma configuração com cinco sensores, dos quais três foram colocados na parte frontal do robô, um na lateral esquerda, e o quinto na lateral direita. Enfatizou-se a parte frontal do robô, pois, evidentemente, esta é a área mais importante para a navegação. Testes realizados com um número maior de sensores não foram satisfatórios, pois acarretavam em um maior nível de ruído nos sensores. Além disso, um aumento no número de entradas implica numa maior complexidade do SIF.

A distribuição final dos sensores de ultrassom é apresentada na Figura 6.



Figura 6. Localização dos sensores de ultrassom.

A Figura 7 apresenta, de forma mais clara, o campo visual do robô, de forma a detectar qualquer obstáculo presente à sua frente.

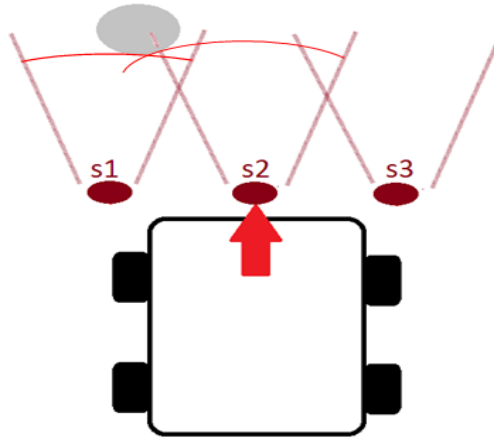


Figura 7. Distribuição dos sensores na zona frontal.

O sensor angular Blade 360 fornece o ângulo de desvio do protótipo em relação a um alvo com uma resolução de até 0,3 graus. O valor medido é uma tensão de 0 a 5 volts, correspondendo a um ângulo entre 0 a 360 graus. O sensor está localizado na parte central do RAHM, conforme mostra a Figura 8.

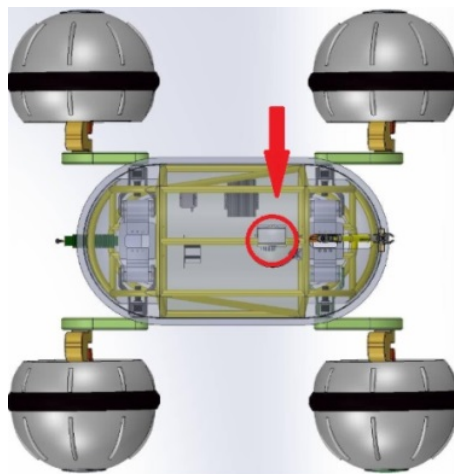


Figura 8. Localização do sensor angular.

3.2.2. Processador

Neste tipo de aplicação, é importante selecionar um processador de baixo custo e de rápido processamento. A informação fornecida pelos sensores deve ser processada na ordem de milissegundos, de forma a se obter um controle em tempo real. Assim, a placa

Arduino Mega foi selecionada para esta tarefa. Esta placa possui um microprocessador ATmega2560, com 54 pinos configuráveis como entradas ou saídas digitais, dos quais 15 podem ser saídas PWM, 16 entradas analógicas e 4 portas seriais UARTs. A placa opera com uma frequência de 16 MHz, permitindo realizar cálculos em questão de microssegundos. Para o desenvolvimento do Sistema Auxiliar de Navegação do RAHM foram utilizadas seis (6) entradas analógicas para a leitura dos dados fornecidos pelos cinco sensores de ultrassom e pelo sensor angular, um canal de comunicação através de um módulo ethernet e a implementação de um programa em linguagem C para traduzir a informação em tempo real.

Apesar de não possuir proteção IP67, a placa Arduino Mega foi colocada em uma caixa projetada com esta proteção, de forma a evitar a interrupção do seu funcionamento em caso de sujeira ou umidade.

A Figura 9 apresenta a placa Arduino Mega, com todos os seus conectores (alimentação de energia, conexão USB com o computador, ligação da placa com diferentes dispositivos de entrada e saída).

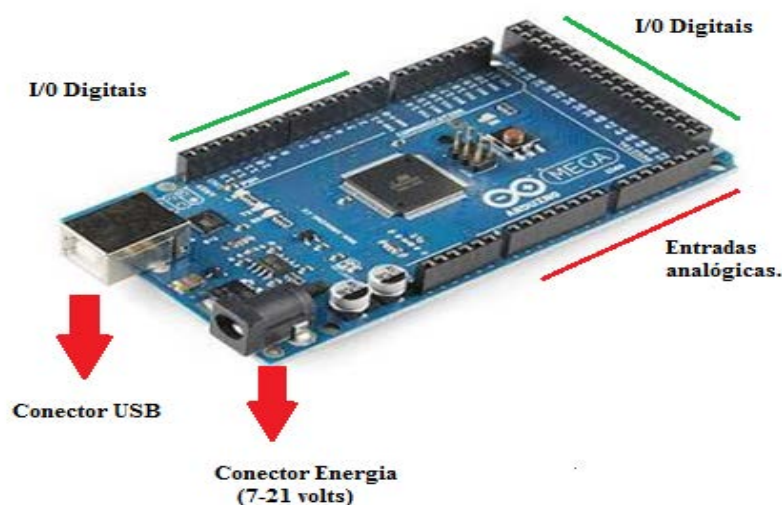


Figura 9. Placa Arduino Mega.

Foi desenvolvida também uma placa (Figura 10) com a função de redundância para aumentar o nível de segurança no sistema. Ao se detectarem eventuais falhas na placa Arduino Mega, as conexões são redirecionadas para outra placa Mega de forma que as tarefas continuem sendo realizadas.

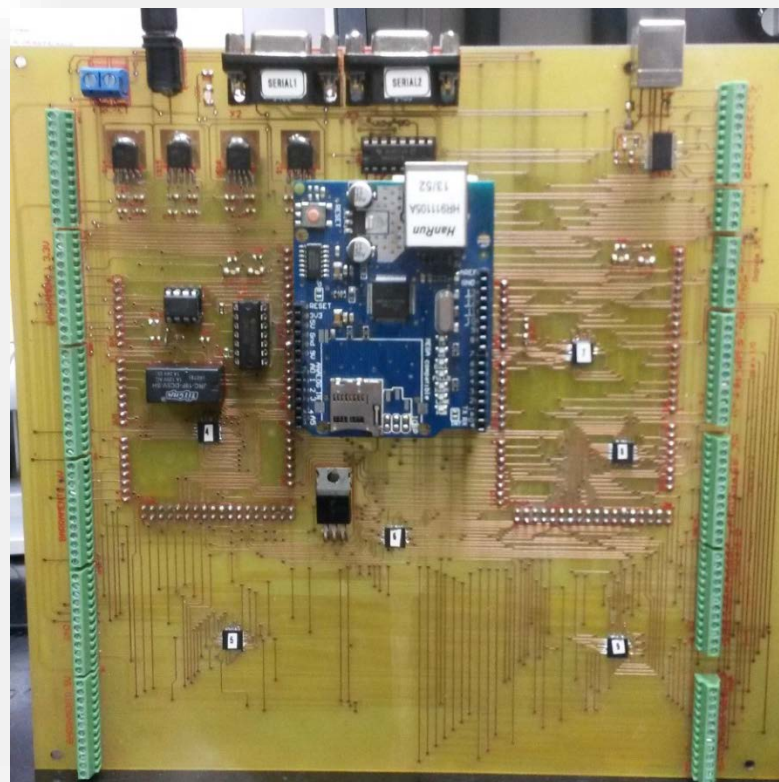


Figura 10. Placa de Redundância Arduino.

3.2.2.1. Configuração e Cabeamento da Placa Arduino Mega

A placa Arduino Mega é o cérebro auxiliar do RAHM e tem como função receber e enviar ordens para outros dispositivos. A Figura 11 mostra o esquema o de conexão geral da placa.

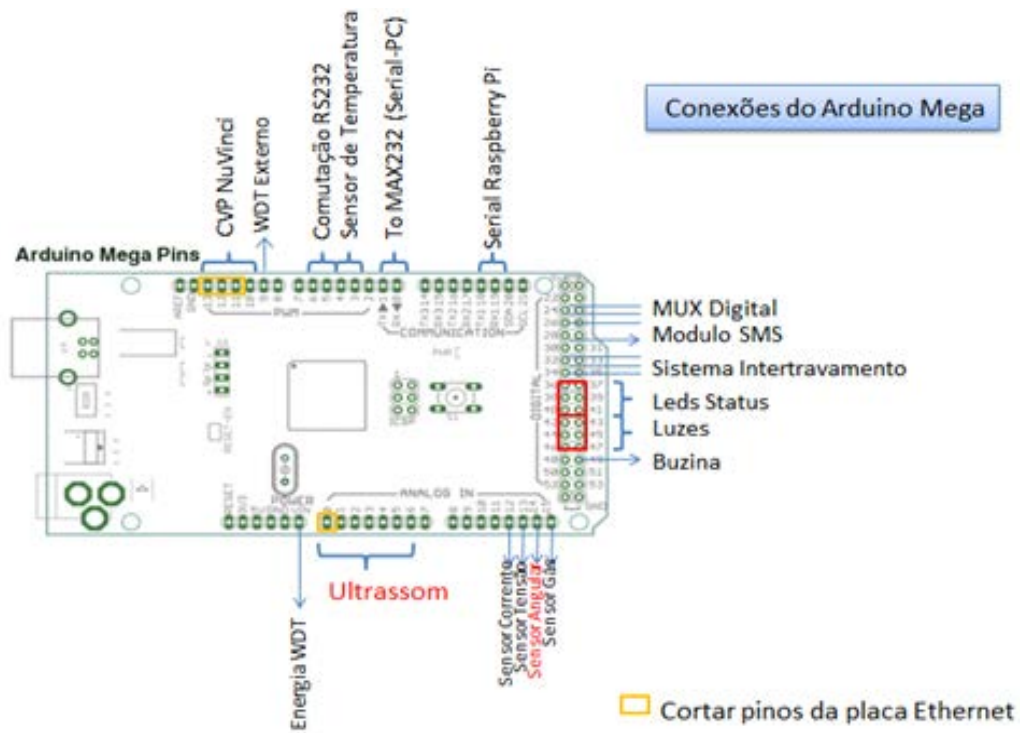


Figura 11. Esquema de ligações da placa Arduino Mega.

A Figura 12 apresenta a ligação detalhada entre o sensor e a placa.

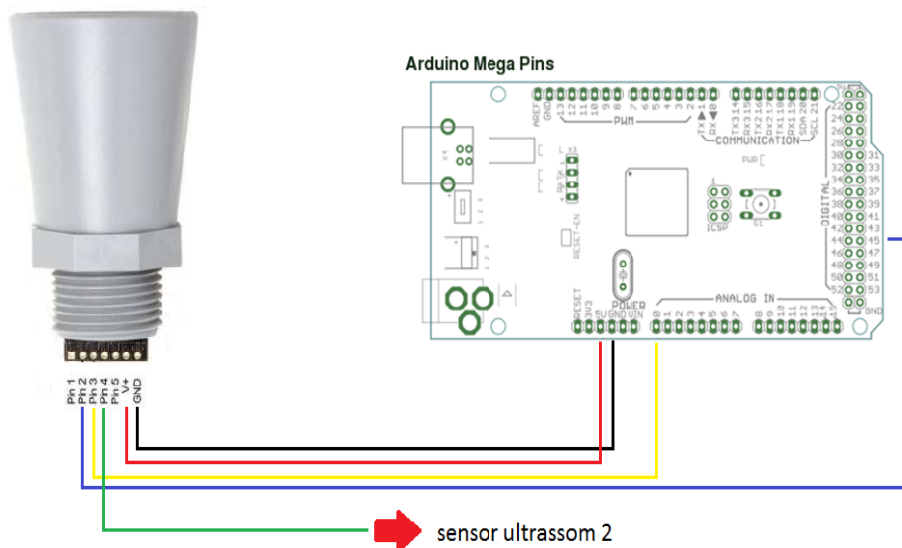


Figura 12. Conexão detalhada do sensor com o Arduino.

- O ponto de terra (linha preta) é um ponto comum para todos os dispositivos do RAHM.

- A alimentação dos sensores é obtida do pino de 5 volts de saída do Arduino (linha vermelha).
- A linha amarela representa a saída analógica fornecida pelo sensor, a qual é conectada à entrada analógica A0 do Arduino.
- A linha azul representa a ligação de uma saída digital do Arduino, que deve ser conectada ao pino dois (2) do sensor, através do qual se envia uma ordem de disparo para o sensor realizar a leitura; este sinal apresenta uma frequência determinada e controlada pelo Arduino (no caso específico deste trabalho, o sinal é enviado a cada 50 milissegundos).
- A linha verde representa a conexão que serve para sincronizar os sensores de ultrassom. Este pino (pino4) é uma saída ligada ao pino dois do sensor seguinte, que fica aguardando o sinal de controle para obter a sua leitura. Esta ligação em série dos sensores permite uma melhor sincronização das leituras, mas tem a desvantagem de impossibilitar a leitura dos sensores seguintes caso algum dos sensores apresente uma falha.

A Figura 13 apresenta as ilustrações dos cabearios feitos no sensor e da distribuição destes cabos ao longo da estrutura do robô, utilizando calhas e abraçadeiras. Foram utilizados conectores de quatro vias para facilitar as ligações e desconexões dos sensores, e também para o caso de manutenção, inspeção ou simplesmente troca de um sensor.



Figura 13. Conexões dos sensores no RAHM.

3.2.2.2. Adaptação do Módulo Ethernet no Arduino Mega

Dentro dos requerimentos do RAHM, foi determinado que as informações advindas dos diferentes equipamentos deveriam ser transmitidas utilizando o mesmo protocolo de comunicação: um módulo desenvolvido pela empresa Arduino, onde todos os sinais são recebidos em um switch através de conectores RJ45 e protocolo ethernet. Este módulo é apresentado na Figura 14.



Figura 14. Módulo Ethernet

Pode se perceber, pela distribuição dos conectores, que a placa foi projetada para se conectar especificamente à placa Arduino Mega. A placa contém o conector ethernet RJ45, o circuito integrado W5100 entrada para memórias micro SD externas.

3.2.3. Módulo de Comunicação

O módulo de comunicação é responsável pela troca de informação entre o robô e o computador remoto. Compreende equipamentos como o switch mencionado na seção anterior e os rádios transmissor (robô) e receptor (computador remoto). Apresentam-se a seguir estes equipamentos, identificando as necessidades que devem cumprir para o funcionamento do projeto eletrônico.

3.2.3.1. Seleção e Configuração do Switch

Este dispositivo deve possuir uma quantidade de portas suficientes para receber todos os sinais, sua velocidade de transmissão deve ser adequada para não retardar o

processamento das diferentes tarefas do RAHM, deve ter proteção IP67 e suas dimensões devem ser adequadas para a sua alocação dentro do robô.

O switch IGS-1050, da empresa Oring, apresentado na Figura 15, cumpre com todos os requisitos acima: tem 18 portas RJ45, suficientes para as conexões dos diferentes equipamentos, possui a propriedade PoE para obter energia a partir da mesma conexão ethernet, facilitando conexões no RAHM, e apresenta a proteção IP67.

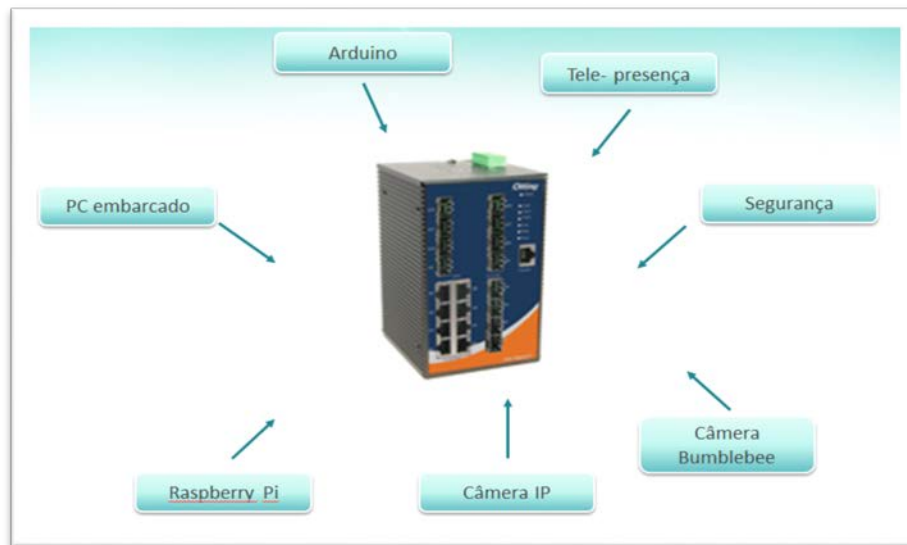


Figura 15. Switch O-ring IGS-1050A

A Figura 16 apresenta um diagrama das conexões gerais do projeto. O código de cores identifica os diferentes protocolos de comunicação entre os dispositivos.

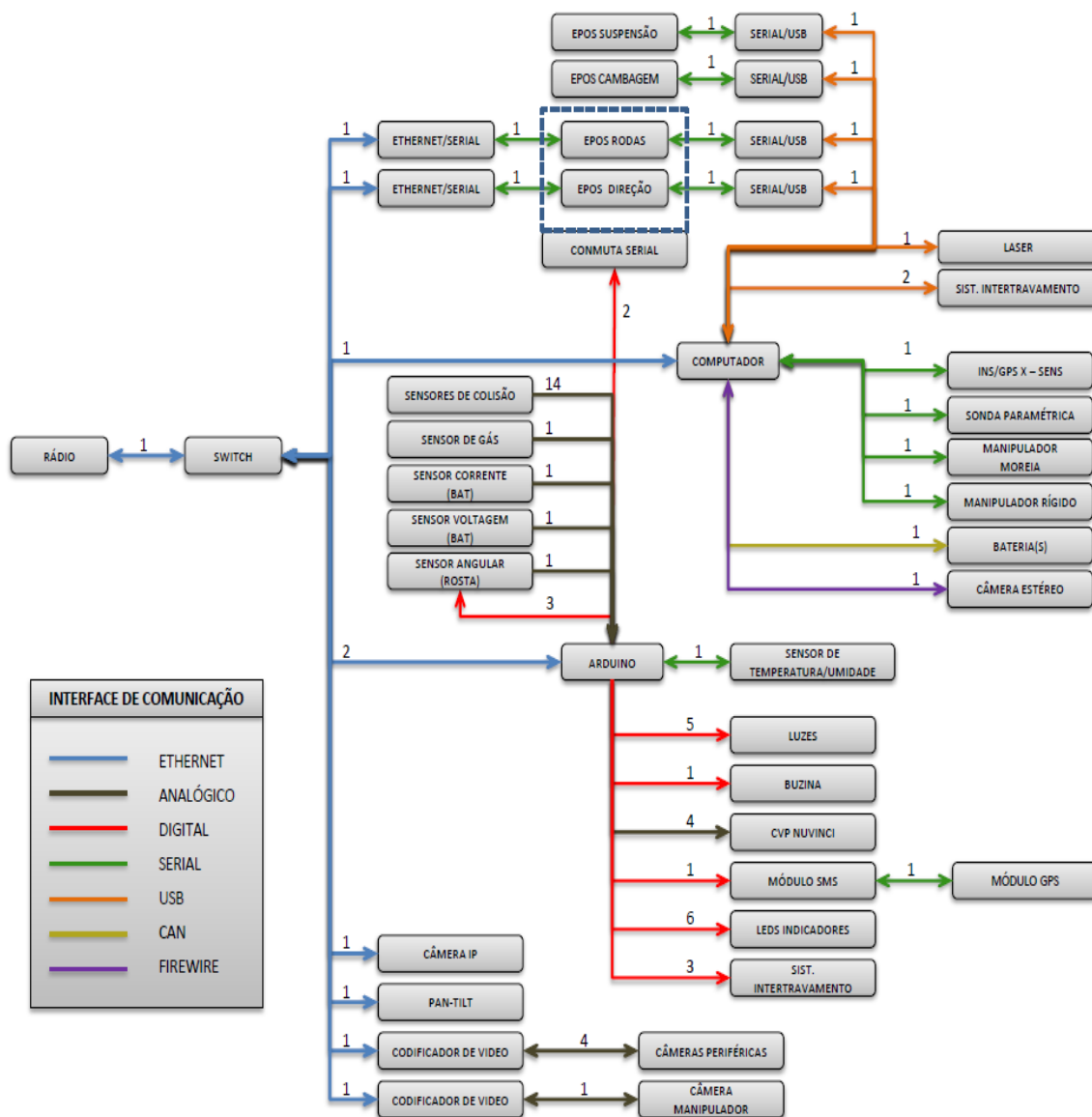


Figura 16. Diagrama de conexões do RAHM

3.2.3.2. Rádio Comunicação

Ao longo da trajetória de 400 km que o RAHM deve percorrer para o monitoramento do ambiente no entorno do gasoduto Coari-Manaus, foram colocados, a cada 1 km, repetidores de radio frequência. Este método de comunicação entre o robô e o computador remoto foi escolhido pelo alcance que esta tecnologia oferece, além de ser o método de comunicação mais apropriado para ambientes externos (confiável e de fácil utilização). O equipamento utilizado, mostrado na Figura 17, apresenta a proteção IP67.



Figura 17. Radio de comunicação RAJANT da WETSCOM

Com a finalidade de conhecer a real cobertura, atraso e taxa de transmissão de dados dos rádios WETSCOM (rádios RAJANT 2,4 GHz e 4,5 GHz), foram realizados testes em locais distintos, mas similares aos encontrados na Amazônia: o bosque da Barra, o parque Lage e a Lagoa Rodrigo de Freitas. Para tal, foi montado um equipamento portátil composto de uma mochila com bateria de 12 volts, rádio portátil e câmera IP. Outro equipamento foi montado na estação base com o notebook, apresentando o vídeo da câmera IP do equipamento portátil. Dois celulares com cronômetros sincronizados também foram utilizados para medir o atraso da imagem transmitida. A imagem do cronômetro, enviada pela câmera, foi comparada com a leitura do cronômetro do celular sincronizado na base.

Com os testes realizados no bosque da Barra (Figura 18) e no parque Lage, verificou-se que o sinal de rádio é influenciado pela presença de árvores, com conseqüente redução na taxa de transmissão de dados e aumento no atraso do sinal. Sem a presença de árvores, a taxa de transmissão de dados da câmera chegou a ser de 2,9 Mbps com um atraso de 0,2 segundos aproximadamente. Na presença de árvores, a taxa de transmissão passou a ser de 1,5 Mbps.

Desta primeira experiência concluiu-se que a comunicação com poucos obstáculos é satisfatória. Além disso, vale ressaltar que nos testes realizados foram transmitidos sinais de vídeo, que contém muito mais informação do que os dados trocados entre o RAHM e o Controlador Fuzzy instalado no computador remoto.

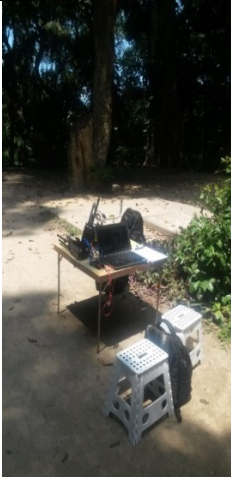
		
Estação base	Equipamento móvel	Local de testes
		
Distância percorrida ainda com cobertura		

Figura 18. Teste de rádios RAJAN no Bosque da Barra.

Um segundo teste foi realizado na Lagoa Rodrigo de Freitas (Figura 19), para se conhecer a distância de cobertura sem obstáculos. Neste caso, a comunicação foi estabelecida até aproximadamente 2 km, com uma taxa de transmissão de dados superior a 2 Mbps e com um atraso inferior a 1 segundo, sem uso de repetidor entre os equipamentos transmissor e receptor.



Figura 19. Teste de rádios RAJAN na Lagoa Rodrigo de Freitas.

3.2.4. Módulo de Atuação

O módulo de atuação é composto pelos equipamentos encarregados de converter a informação de saída recebida do Controlador Fuzzy em movimentos adequados para uma boa navegação do robô. Neste módulo estão os atuadores EPOS, descritos a seguir.

3.2.4.1. Atuadores EPOS dos Motores DC de Tração

Os atuadores EPOS, mostrados na Figura 20, traduzem os valores precisos de saída do Controlador Fuzzy em níveis de tensão para o controle da velocidade angular dos motores. Fornecem também o parâmetro de tempo de desaceleração e aceleração do motor durante o arranque, parada ou inversão do sentido de giro, com o objetivo de não ocasionar mudanças bruscas na velocidade angular.

Estes equipamentos possuem várias formas de comunicação, entre elas a comunicação serial RS-232. Sua tensão de alimentação é de 24 VCC.



Figura 20. Atuadores EPOS motores de tração.

Lembrando que os sinais de saída de todos os equipamentos do RAHM devem obedecer ao protocolo Ethernet, foi necessário compatibilizar os dados enviados via comunicação RS232 com este protocolo.

Foi realizado um teste para controlar os motores DC da MAXON Motors, utilizando duas placas de conversão Serial-Ethernet, apresentadas na Figura 21 (para configuração das placas, vide Anexo A) e o driver controlador dos motores (EPOS). O objetivo deste teste foi realizar uma ponte física entre o protocolo de comunicação das controladoras (EPOS) e o protocolo de comunicação Ethernet. Assim, as EPOS podem ser conectadas na rede Ethernet do RAHM e os motores podem ser acionados via conexão Ethernet (Figura 22).

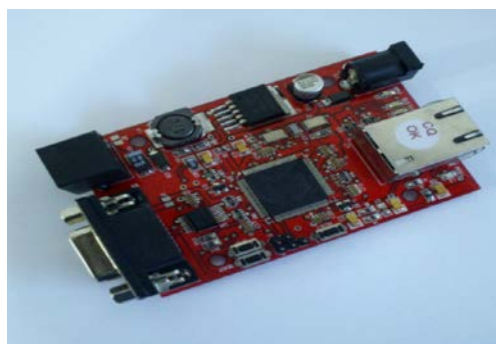


Figura 21. Módulo RS232/Ethernet Zuchi.

Como resultado final, foi possível controlar a velocidade do motor a partir do computador. Recomenda-se utilizar duas placas conversoras serial/Ethernet que sejam iguais (mesma marca e referência) para assegurar estabilidade na comunicação.

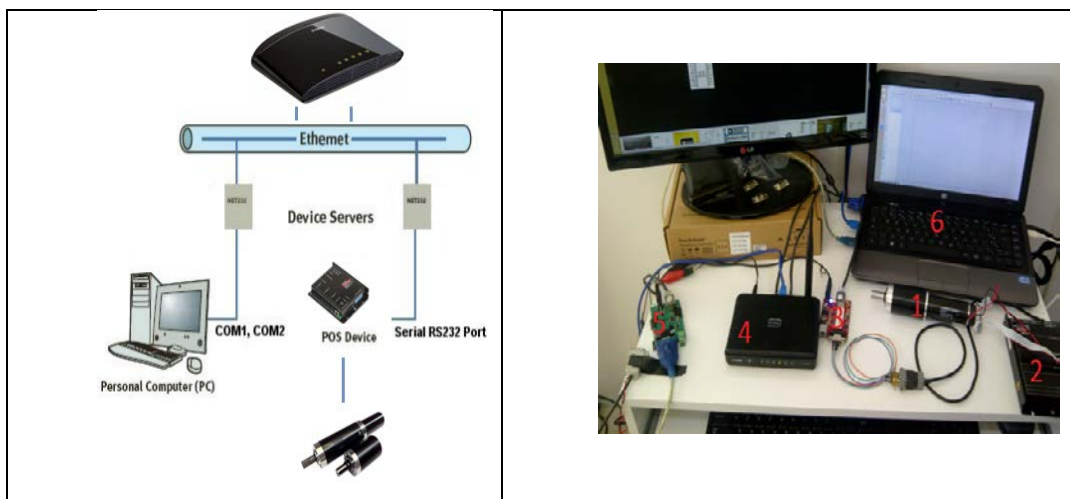


Figura 22. Teste serial Ethernet com as controladoras EPOS.

Na Figura 22, à esquerda, apresenta-se o diagrama do canal de comunicação e controle dos motores exatamente como implementado no RAHM.

3.2.4.2. Motores DC de Tração

Os motores responsáveis pelo deslocamento do RAHM foram selecionados de forma a satisfazer requisitos mecânicos e elétricos, tais como dimensões, torque, velocidade angular, níveis de tensão e corrente, fácil instalação física e comunicação. Como a empresa MAXON fabrica os motores adequados para os diferentes atuadores que desenvolve, os motores foram previamente escolhidos para então escolherem-se os atuadores. A Figura 23 ilustra os motores DC da MAXON utilizados, cujas características principais estão dispostas na Tabela 2.



Figura 23. Motores de Tração.

Característica	Valor
Tensão nominal	24 V
Velocidade em vazio	9640 RPM
Corrente em vazio	29.5 mA
Velocidade nominal	7440 RPM
Par nominal (máx. par em contínuo)	4.81 mNm
Corriente nominal (máx. corrente em contínuo)	0.84 A
Par de arranques	20.5 mNm
Corrente de arranque	3.48 A
Máx. rendimento	83 %

Tabela 2 Características gerais dos motores de tração

3.3. Diagrama Geral do Projeto Eletrônico

A Figura 24 apresenta um esquema com todos os equipamentos utilizados em cada um dos módulos abordados neste capítulo.

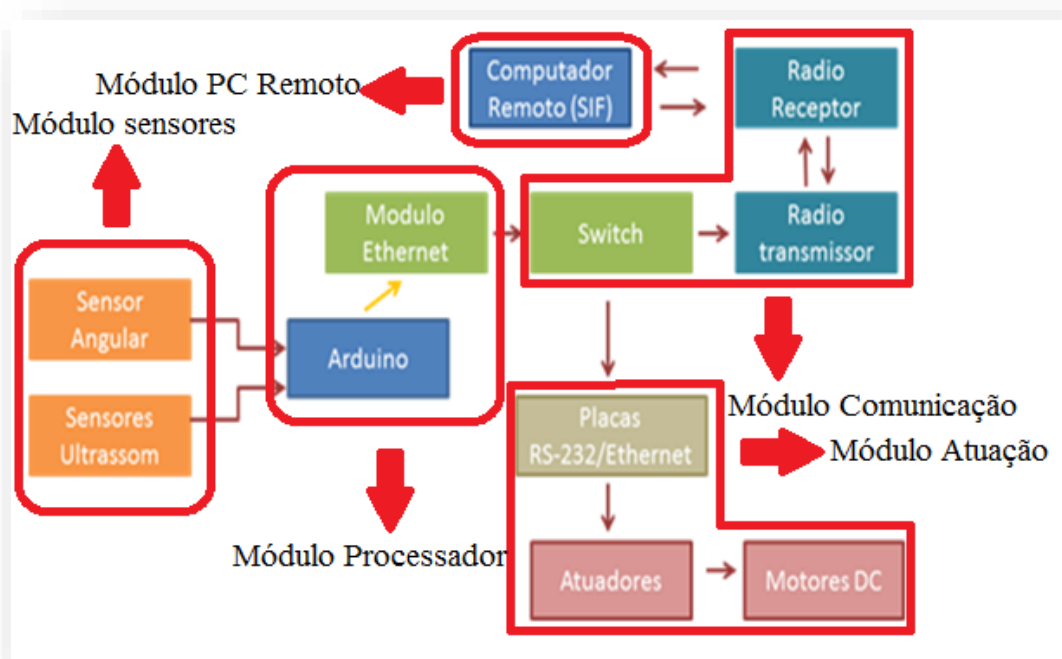


Figura 24. Esquema Geral do Projeto Eletrônico.

4. Sistema Auxiliar de Navegação do RAHM

Conforme mencionado, o RAHM possui três tipos de funcionamento: automático, semiautomático e teleoperado. No modo automático, o RAHM é completamente autônomo, baseando sua navegação em todos os seus sensores. No modo semiautomático, o RAHM desliga alguns de seus equipamentos e a navegação é realizada por meio de instrumentos mais simples, com o objetivo de poupar energia e recursos computacionais, tais como sensores e módulo auxiliar de processamento de informação. Por fim, no modo de teleoperação o RAHM responde somente à tração, câmera IP e canal de comunicação, ou seja, os equipamentos vitais para que o operador possa movimentar o robô até atingir uma zona segura para inspeção e reparação.

Este capítulo apresenta o controlador fuzzy desenvolvido para o controle do robô no modo semiautomático, descrevendo todos os módulos do Sistema de Inferência Fuzzy desenvolvido.

4.1. Sistema de Inferência Fuzzy Geral

Devido ao número de variáveis consideradas no controle do robô em modo semiautomático (seis no total), optou-se pelo desenvolvimento de um sistema de inferência fuzzy hierárquico, onde sub-sistemas de inferência analisam sub-conjuntos das variáveis de entrada, reduzindo, assim, o número total de regras. O controlador Fuzzy é, portanto, composto por dois subsistemas de inferência fuzzy, conforme apresentado na Figura 25: o SIF1, denominado, “Área frontal” (Sistema Intermediário) e o SIF 2, denominado “Sistema de Controle”. O primeiro tem como objetivo obter informação dos obstáculos à frente do robô, incluindo a distância a estes obstáculos e a sua orientação em relação ao objetivo. Já o SIF 2 recebe a informação de saída do sistema intermediário e, juntamente com os sensores que detectam obstáculos nas zonas laterais, fornece os valores de saída para os atuadores EPOS.

Na Figura 25 são apresentadas as entradas e as saídas do sistema geral: seis entradas de cinco sensores de ultrassom e um sensor angular, e duas saídas que controlam os motores de tração.

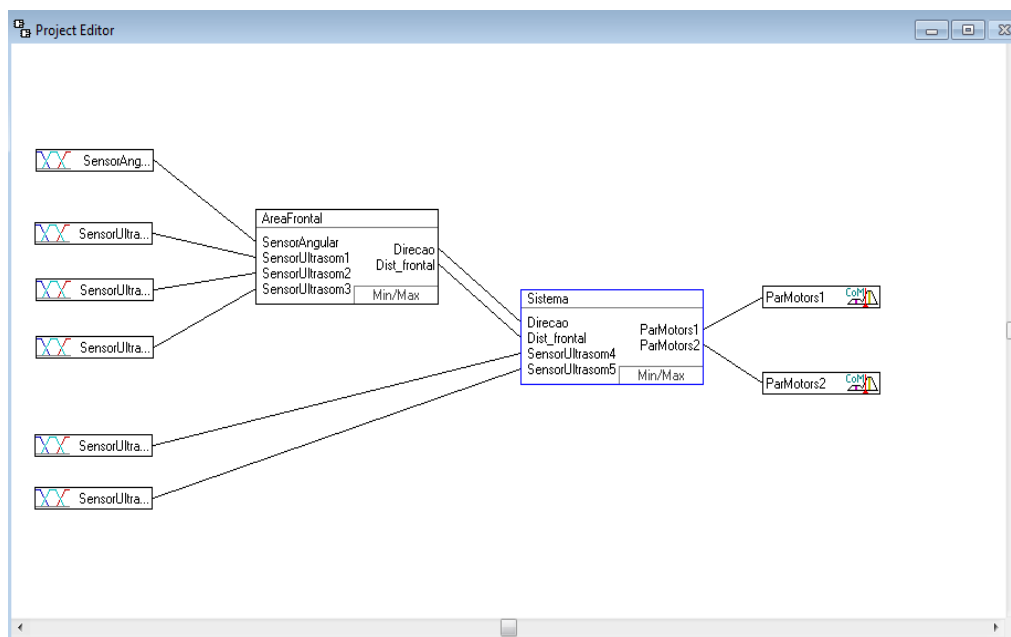


Figura 25. Sistema de Inferência Fuzzy geral.

A direção do robô é dada pela diferença de velocidade angular entre os pares de motores do lado esquerdo em relação com os pares de motores do lado direito. São necessárias, portanto, duas saídas – fornecidas pelo SIF principal. Uma saída envia o mesmo valor para os dois motores do lado esquerdo (PM1) e a outra envia a informação para os dois motores do lado direito (PM2), conforme mostrado na Figura 26. Assim, a curva é determinada pela diferença de velocidades angulares entre PM1 e PM2.

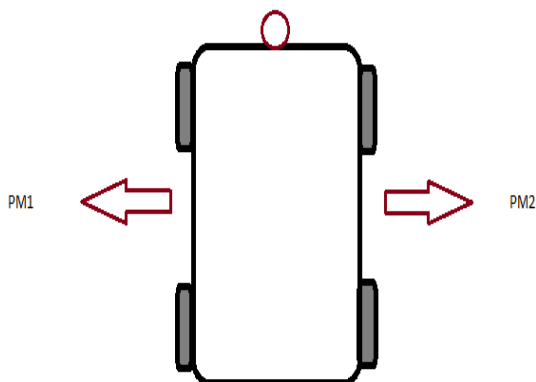


Figura 26. Pares de motores.

4.1.1. Sistema Intermediário

Para a navegação do robô, é vital a informação obtida na sua parte frontal. Assim, três dos cinco sensores de ultrassom foram colocados na frente do robô. Com a informação destes três sensores, juntamente com a do sensor angular, é possível desviar dos obstáculos presentes no caminho do robô e, ao mesmo tempo, enviar ordens para que o protótipo siga a rota até o objetivo preestabelecido. A partir das informações destes quatro sensores, o SIF intermediário (Figura 27) fornece como saídas a orientação a ser seguida pelo robô e a distância média geral dos obstáculos na zona frontal.

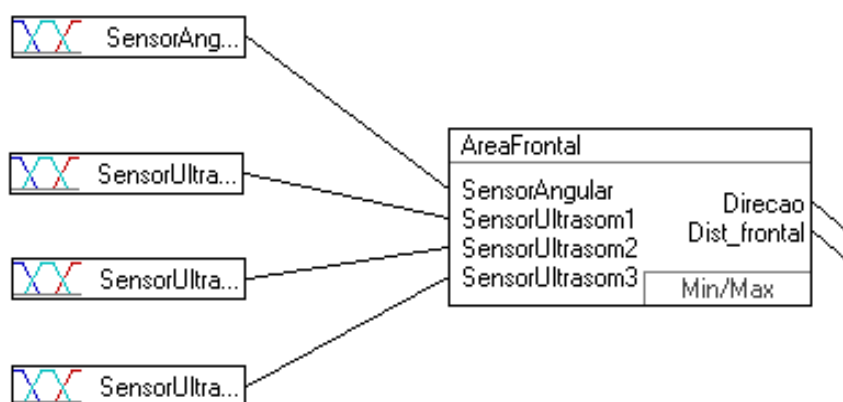


Figura 27. Sistema intermediário.

As duas saídas intermediárias “direção” e “distância frontal” são defuzzificadas e utilizadas como entradas do SIF2 (Sistema de Controle).

4.1.1.1. Variáveis de Entrada do SIF Intermediário

Os sensores de ultrassom oferecem a informação de distância de qualquer obstáculo que seja detectado em uma faixa de 30 cm a 5 metros na linha de visão do robô. O sensor angular fornece o ângulo de desvio em relação a um alvo especificado previamente, ou seja, a orientação da frente do robô em relação ao objetivo. A seguir serão apresentados os termos linguísticos e os conjuntos fuzzy, definidos por funções de pertinência, correspondentes às variáveis de entrada consideradas.

- **Distância ao Obstáculo:** os termos linguísticos e as funções de pertinência para as três entradas do SIF (dados de três sensores) intermediário que dizem respeito à distância do obstáculo estão mostradas na Figura 28.

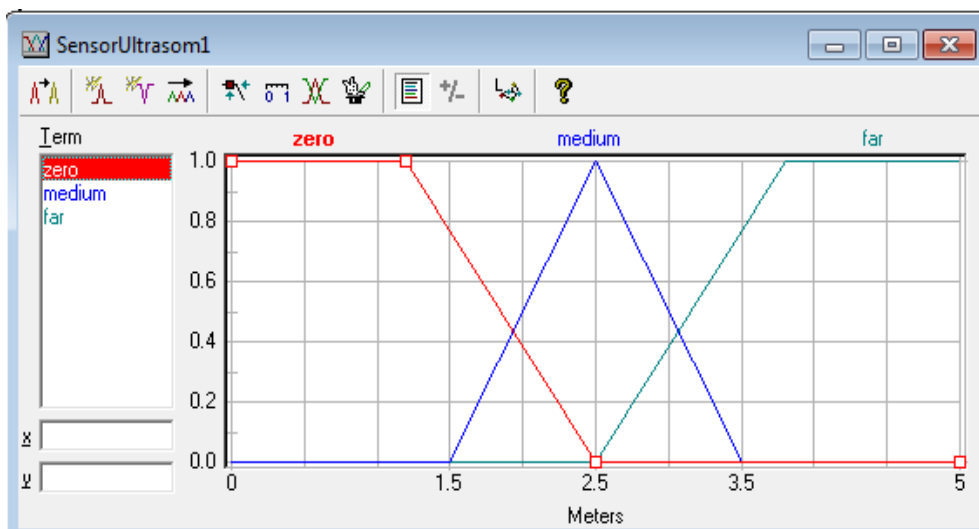


Figura 28. Distância ao Obstáculo.

- **Ângulo de Desvio:** os cinco termos linguísticos e as funções pertinência correspondentes ao ângulo de desvio estão mostradas na Figura 29. A placa Arduino foi utilizada para a conversão do sinal fornecido pelo sensor, de 0 a 360 graus, para o universo de discurso de -180 graus a +180 graus, de forma a se detectar o sentido do desvio para a correção da trajetória do robô.

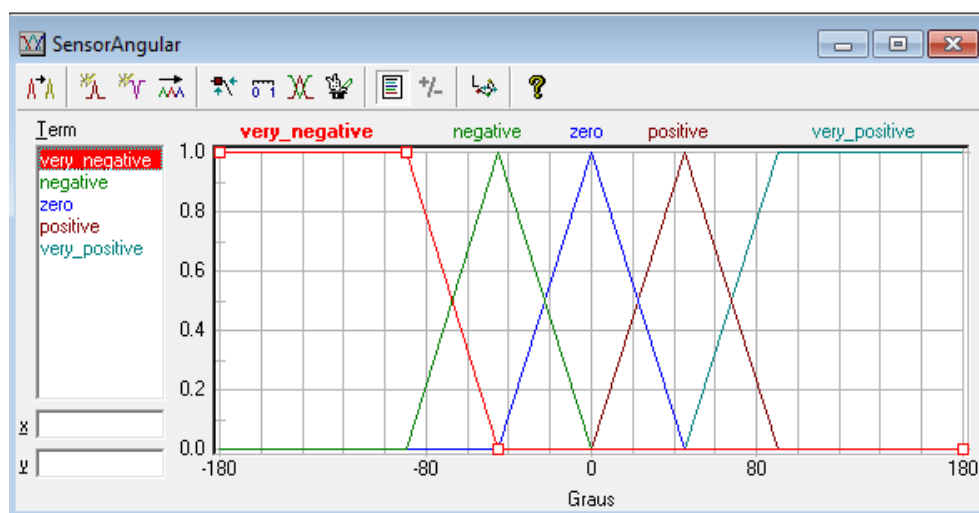


Figura 29. Ângulo de desvio.

4.1.1.2. Variáveis de Saída

As saídas do SIF1 são: distância média em relação ao obstáculo que se encontra na frente do robô (Distância Frontal) e ângulo de direção que o robô deve seguir para manter a sua trajetória até a posição desejada.

Para a saída Distância Frontal, os termos linguísticos e funções de pertinência são idênticos aos utilizados para as entradas relativas à distância (Figura 27). Para a saída Direção, os termos linguísticos e as funções de pertinência são iguais aos mostrados na Figura 28 para a variável de entrada Ângulo de Desvio.

A saída direção apresentada na Figura 27 corresponde às características apresentadas na Figura 29, mesmas características fuzzy da entrada ângulo de desvio, a diferença esta em que esta saída não mostra o ângulo de desvio, mostra o ângulo que tem que corrigir o robô para ir direto ao alvo.

4.1.1.3. Base de Regras do SIF Intermediário

O número de regras possíveis para o SIF1 é 135, já que recebe três informações de distância, com três termos linguísticos cada, e uma do sensor angular, com cinco termos linguísticos. A base de regras completa encontra-se no Anexo C; um exemplo de regra é apresentado abaixo:

Se ultrasom1 é perto e ultrasom2 é médio e ultrasom3 é médio e ângulo é muito negativo **então** a direção é muito_positivo e a distância é media.

4.1.2. Sistema de Inferência Fuzzy 2 (Sistema de Controle).

Este sistema recebe como entradas os valores defuzzificados do SIF1, além das informações dos sensores localizados nas zonas laterais do robô, e fornece na saída a informação para os pares de motores mencionados na seção 4.1. A Figura 30 apresenta a estrutura do SIF2, com suas quatro entradas e duas saídas.

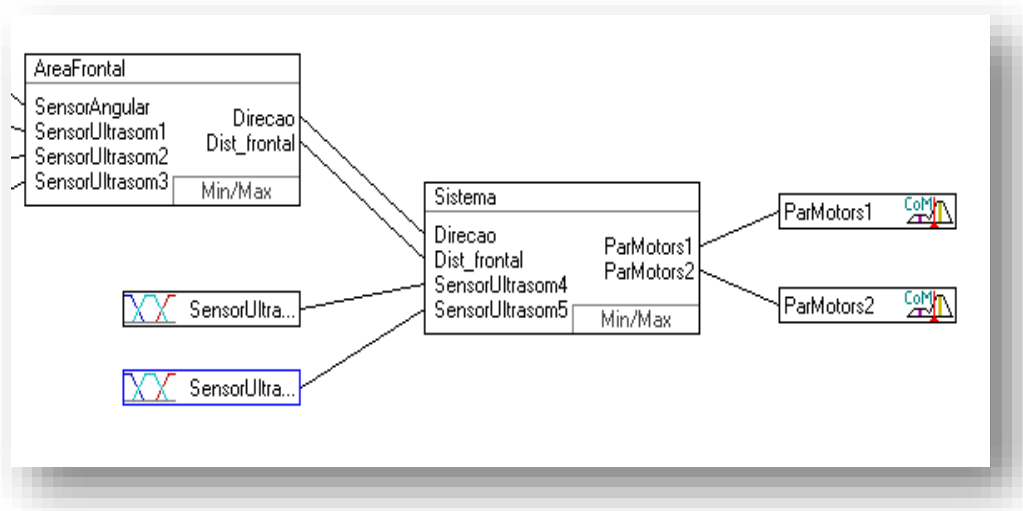


Figura 30. Sistema de Inferência 2 (Sistema de Controle).

4.1.2.1. Variáveis de Entrada do Sistema de Controle

Como mencionado, são quatro as variáveis de entrada deste sistema:

- **Distância Frontal e Direção:** fornecidas pelo sistema intermediário, estas duas entradas possuem as características apresentadas na Figura 28 e Figura 29, respectivamente;
- **Distância ao Obstáculo:** são duas variáveis fornecidas pelos sinais dos sensores de ultrassom localizados nas laterais direita (sensor ultrassom 4) e esquerda (sensor ultrassom 5). Estas entradas possuem as mesmas características dos sensores de ultrassom do SIF1 Intermediário.

4.1.2.2. Variáveis de Saída do SIF2 (Sistema de Controle)

Conforme mencionado, o SIF 2 fornece as duas saídas que serão utilizadas para acionar o par de motores ilustrado na Figura 26.

Os atuadores das rodas de tração (Figura 20), recebem um valor entre -5000 e 5000 que é proporcional ao sinal de tensão fornecido para o motor de tração.

O sinal positivo ou negativo informa o sentido de rotação do motor, que corresponde ao movimento para frente ou para trás do robô. A velocidade do robô é dada pela velocidade angular dos motores, que é proporcional ao sinal recebido nos

atuadores, sendo 5000 a velocidade máxima do robô e 0 a velocidade mínima. A velocidade máxima do robô é de 1m/s.

Na Figura 31 são apresentadas os termos linguísticos e as funções de pertinência da variável do SIF 2. O universo de discurso, determinado pelos atuadores, é de -5000 a +5000. A distribuição assimétrica das funções de pertinência ao longo do universo deve-se à necessidade de se ter um controle mais fino quando o protótipo se movimenta para frente. Quando o robô se encontra totalmente bloqueado por obstáculos à frente e nas laterais, ele deve se deslocar para trás até alcançar uma região mais livre. Neste caso, o controle não necessita ser tão fino.

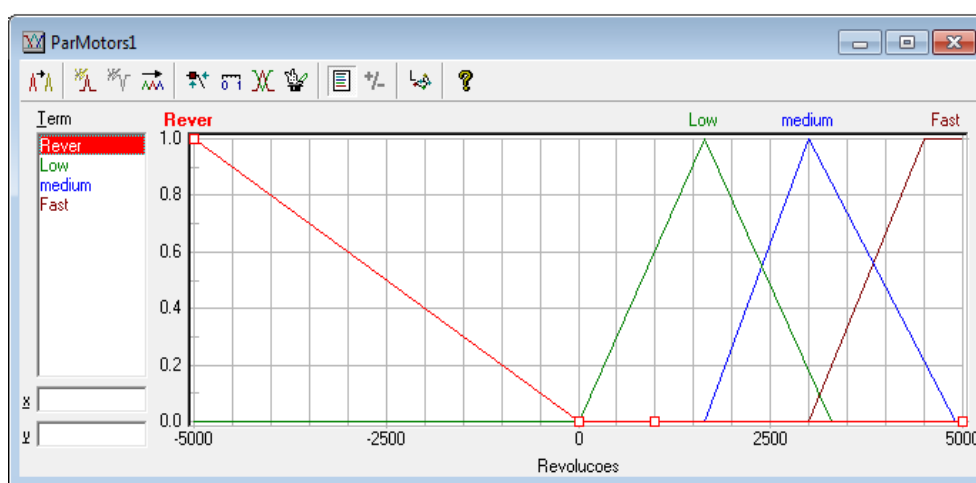


Figura 31. Variável de saída.

4.1.2.3. Base de Regras do SIF 2 (Sistema de Controle)

A exemplo do SIF Intermediário, o número de regras possíveis é 135. A base completa encontra-se no Anexo C e a seguir apresenta-se um exemplo de regra.

Se distância é perto e direção é muito_positivo e ultrasom4 é médio e ultrasom5 é longe então PM1 é Rápido e PM2 é lento.

4.1.3. Defuzzificação

Para os dois SIFs, utilizou-se o método de centro dos máximos.

4.2. Implementação do SIF em Visual Studio

O SIF hierárquico apresentado na seção anterior deve ser programado em uma plataforma que permita a sua conexão com os dispositivos existentes no projeto eletrônico. A ferramenta Visual Studio foi a escolhida, uma vez que apresenta os seguintes requisitos fundamentais para a programação do SIF:

- possui componente para receber dados através da porta ethernet, já que os dados obtidos pelos sensores são enviados por radio frequência ao PC remoto, o qual os recebe pela porta ethernet.
- Possui componente para enviar dados através da porta serial, pois as ordens de controle dadas pelo controlador fuzzy são enviadas aos atuadores das rodas através da placa conversora RS232/Ethernet pela porta serial.
- permite o desenvolvimento de uma interface gráfica para o monitoramento de todas as variáveis de entrada e saída.
- possui biblioteca de Lógica Fuzzy na ferramenta.

De forma a realizar o monitoramento de todas as informações do robô em tempo real, foi desenvolvida uma interface em visual Studio, conforme mostrado na Figura 32. Na parte esquerda da interface tem-se uma imagem do RAHM, com a distribuição dos cinco sensores de ultrassom (três frontais e dois laterais) e seus respectivos valores. Já na parte inferior do lado esquerdo, pode-se visualizar o dado recebido do sensor angular. Na parte central, é possível visualizar os dois valores de saída do SIF intermediário: direção e a distância média da zona frontal.

Por fim, na parte direita, tem-se novamente uma imagem do robô onde são monitorados os dados enviados para os quatro atuadores das rodas de tração. A localização de cada caixa corresponde à localização real do atuador que controla a roda.

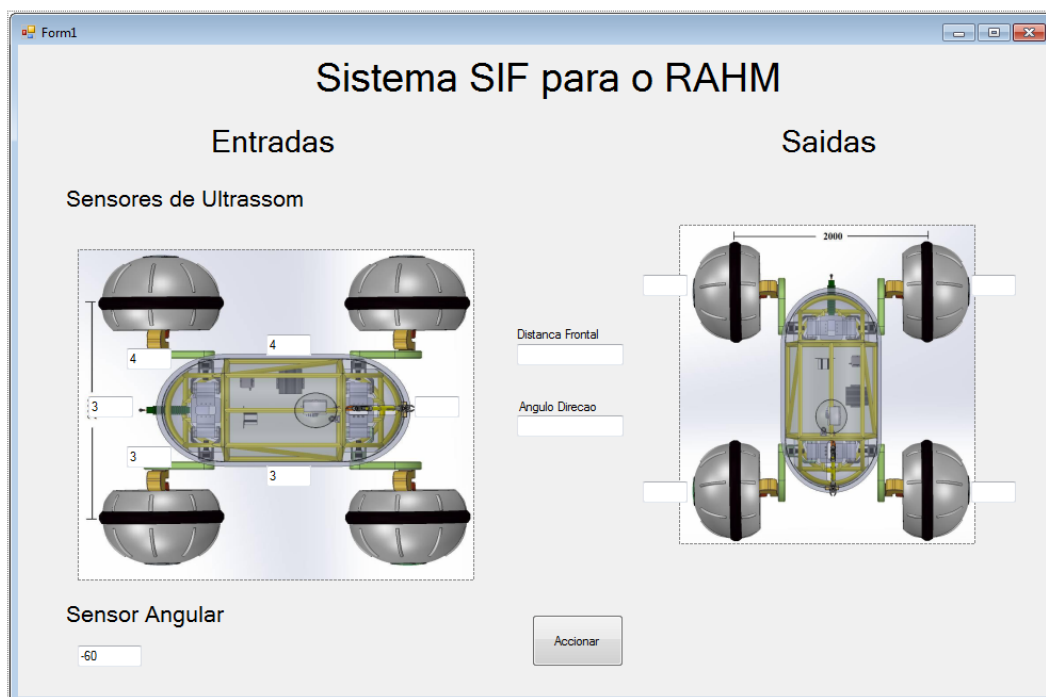


Figura 32. Interface Gráfica de monitoramento.

4.2.1. Biblioteca Dotfuzzy

Visual Studio é uma plataforma de desenvolvimento de software alimentada por diferentes usuários que desenvolvem diversos tipos de aplicações. A biblioteca Dotfuzzy é destinada à programação de Sistemas de Inferência Fuzzy, muito útil para as necessidades deste projeto. O Anexo B traz informações detalhadas a respeito.

5. Estudo de Casos

Para avaliar o comportamento do Sistema de Navegação Auxiliar (baseado na Lógica Fuzzy e implementado no RAHM), em comparação ao do sistema de navegação primário, foram criados dez cenários controlados (distribuições de obstáculos). Esta comparação envolveu itens como custo financeiro, custo computacional, consumo de energia, tempo de percurso e fluxo de dados, de forma a se avaliar a relação custo benefício de cada um deles.

Os dez experimentos foram realizadas em um mesmo ambiente com diferentes configurações de obstáculos. O protótipo deveria completar o trajeto até o ponto final especificado, esquivando-se de todos os obstáculos existentes.

5.1. Tópicos de Comparação

Vários itens foram usados para comparar o sistema de navegação auxiliar, desenvolvido nesta dissertação, com o sistema de navegação primário. As seções a seguir detalham cada uma das métricas de comparação dos dois sistemas de navegação.

5.1.1. Custo Financeiro

Foi realizada uma comparação do custo total de todos os equipamentos utilizados nos dois sistemas. A Tabela 3 apresenta esta informação de forma detalhada.

Sistema Auxiliar de Navegação (SAN) (R\$)	
Sensor de ultrassom	30 *(5)= 150
Arduino Mega	70
Modulo Ethernet Arduino	45
Switch.	160
Radio.	5000
Pc Remoto.	4000
Atuadores Epos dos Motores.	1200*(4)= 4800
Sensor Angular.	250
Motores de tração.	1800*(4)= 7200
Total	R\$ 21675

Sistema Principal de Navegação (SPN)(R\$)	
Sensor Laser LMS 500	75000
Xsens	2500
Computador Embarcado	15000
Switch	160
Radio	5000
Pc Remoto	4000
Atuadores Epos dos Motores	1200(4)=4800
Câmera IP	1200
Mecanismo Pantilt	35000
Câmera Bumblebee	4050
GPS	400
Motores de Tração	1800*(4)= 7200
Total	R\$ 154310

Tabela 3 Custo financeiro dos sistemas.

Pode se verificar na Tabela 3 que o custo total dos equipamentos utilizados no sistema de navegação principal é sete vezes maior do que o dos equipamentos utilizados com o sistema de navegação secundário.

5.1.2. Custo Computacional

Verificou-se a saturação dos processadores utilizados pelos computadores para realizar os cálculos nos dois sistemas por meio de um monitoramento constante (em porcentagem) da saturação da memória RAM dos computadores utilizados. No caso do sistema de navegação secundário, mediu-se a saturação no computador remoto que

executa o controle fuzzy. No caso do sistema principal, tem-se dois computadores: o computador embarcado e o PC remoto.

A Figura 33 apresenta um exemplo de gráfico do Windows para monitoramento em tempo real o consumo da memória RAM.

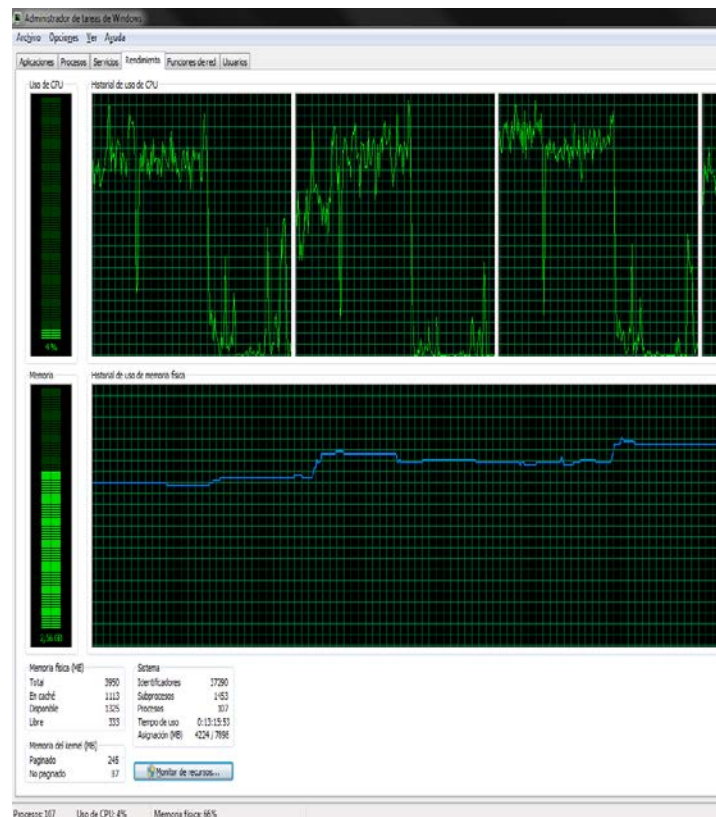


Figura 33. Monitoramento da memória RAM.

Os resultados obtidos por cada sistema de navegação do RAHM durante cada um dos dez experimentos serão apresentados em um quadro comparativo na Seção 5.2.1.

5.1.3. Fluxo de Dados pela Rede

Como explicado no capítulo 3, toda a informação trocada entre o robô e o PC remoto é feita por radio frequência pela porta ethernet do computador remoto. Monitorou-se, em tempo real, a saturação da placa de rede do computador remoto em função dos dados enviados pelos diferentes equipamentos utilizados nos dois sistemas. A placa do PC remoto é a Realtek 10/100Mbps Pci.

O Windows oferece também um gráfico para visualizar a saturação da placa de rede (em porcentagem), como mostrado na Figura 34.

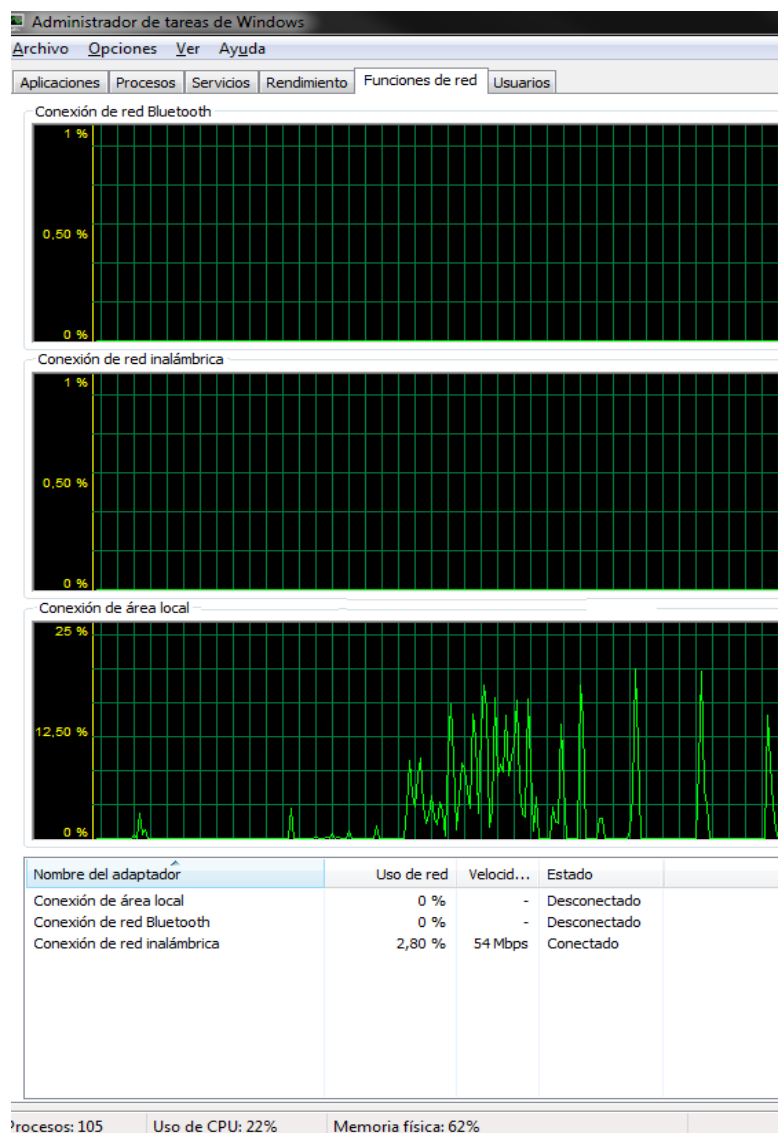


Figura 34. Monitoramento do fluxo de dados.

Os resultados obtidos por cada sistema de navegação do RAHM durante cada um dos dez experimentos serão apresentados no quadro comparativo na Seção 5.2.1.

5.1.4. Tempo

Este item se refere ao tempo que o protótipo leva para percorrer a distância do ponto de partida até o de chegada, esquivando-se dos obstáculos definidos em cada cenário. Para cada sistema de navegação do robô, este tempo foi medido com um cronômetro.

5.1.5. Consumo de Corrente

Além dos diferentes equipamentos e sensores para os sistemas de navegação do robô, foi também instalado um sensor de corrente, que, além de fornecer informação sobre consumo excessivo de algum equipamento em particular, é útil para monitorar o consumo constante e em tempo real dos equipamentos durante a navegação do robô. As tabelas a seguir apresentam os consumos elétricos de cada um dos componentes que compõem tanto o SAN como o SPN.

Sistema Auxiliar de Secundário (SAN)	
Sensores de ultrassom.	2.3 (6)= 13.8
Placa Arduino Mega.	500
Modulo Ethernet Arduino.	100
Switch.	230
Radio.	140
Pc Remoto.	
Atuadores Epos dos Motores.	1150*(4)= 4600
Sensor angular	5
Motores de tração.	840*(4)= 3360
	Consumo(mA): 8948.5

Tabela 4 Consumo de corrente SAN.

Sistema Principal de Navegação (SPN)	
Sensor Laser LMS 500.	2000
Xsens.	80
Computador embarcado.	1100
Switch.	230
Radio.	140
Pc Remoto.	
Atuadores Epos dos Motores.	1150*(4)= 4600
Camera IP	120
Mecanismo pantilt	830

Camera Bumblebee	130
Gps	122
Motores de tração.	840*(4)= 3360
	Consumo (mA): 12712

Tabela 5 Consumo de corrente do SPN.

Pode se concluir que, quando todos os equipamentos do sistema são utilizados, o consumo do SPN é 50% maior do que o do SAN e se concentra nos motores de tração e nos atuadores.

5.2. Testes em Ambientes Controlados

Esta seção apresenta os dez experimentos realizados para avaliar o sistema de navegação auxiliar. Em um ambiente externo plano de 50 m x 30 m foram definidas diferentes distribuições de obstáculos. Tais experimentos são ilustrados nas imagens a seguir, extraídas de vídeos, em que se apresentam os trajetos do robô utilizando o SPN (linha laranja) e o SAN (linha azul).

- **Experimento 1:**

Conforme ilustrado na Figura 35, este experimento considera três obstáculos distribuídos de forma que seja indiferente a escolha, pelo robô, da rota a seguir (pelo lado esquerdo ou direito). O objetivo encontra-se a 50 metros de distância. O robô, quando controlado pelo SAN, logrou chegar ao objetivo, mas sem percorrer a rota mais eficiente.

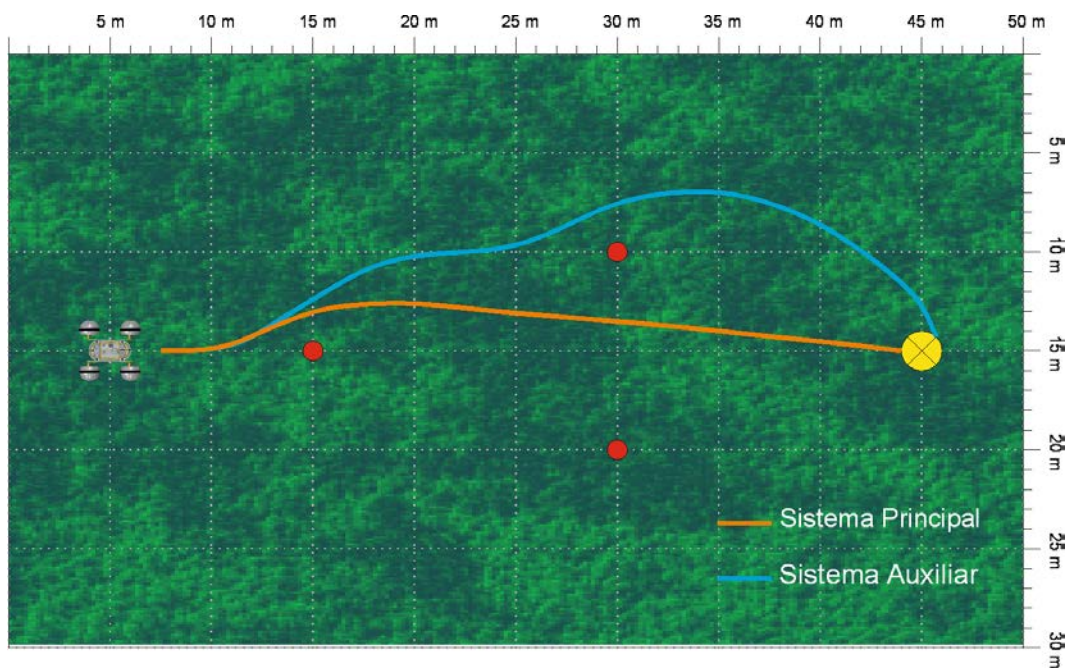


Figura 35. Experimento 1.

- **Experimento 2:**

A distribuição de obstáculos foi idealizada de maneira a forçar um trajeto pelo lado esquerdo. Novamente, pode se verificar a diferença de eficiência entre os sistemas, embora o SAN tenha atingido o objetivo.

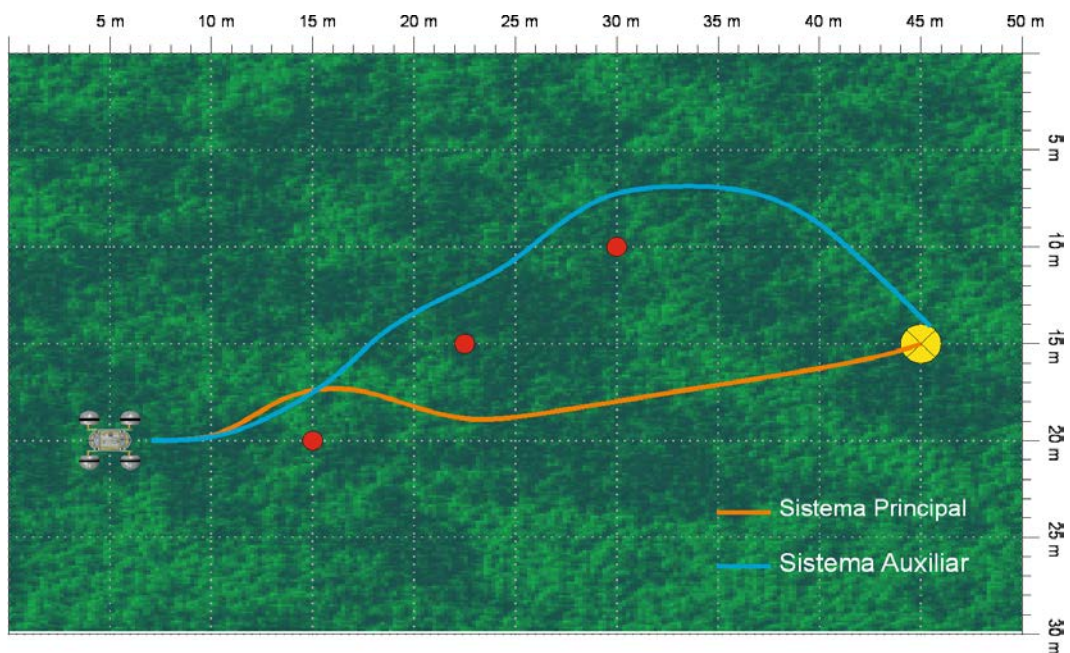


Figura 36. Experimento 2.

- **Experimento 3:**

Este é um espelho do experimento 2: a configuração dos obstáculos induz a uma trajetória ótima pelo lado direito. Os resultados foram similares aos obtidos no experimento 2: o SPN foi mais eficiente que o SAN, mas este conseguiu novamente atingir o objetivo.

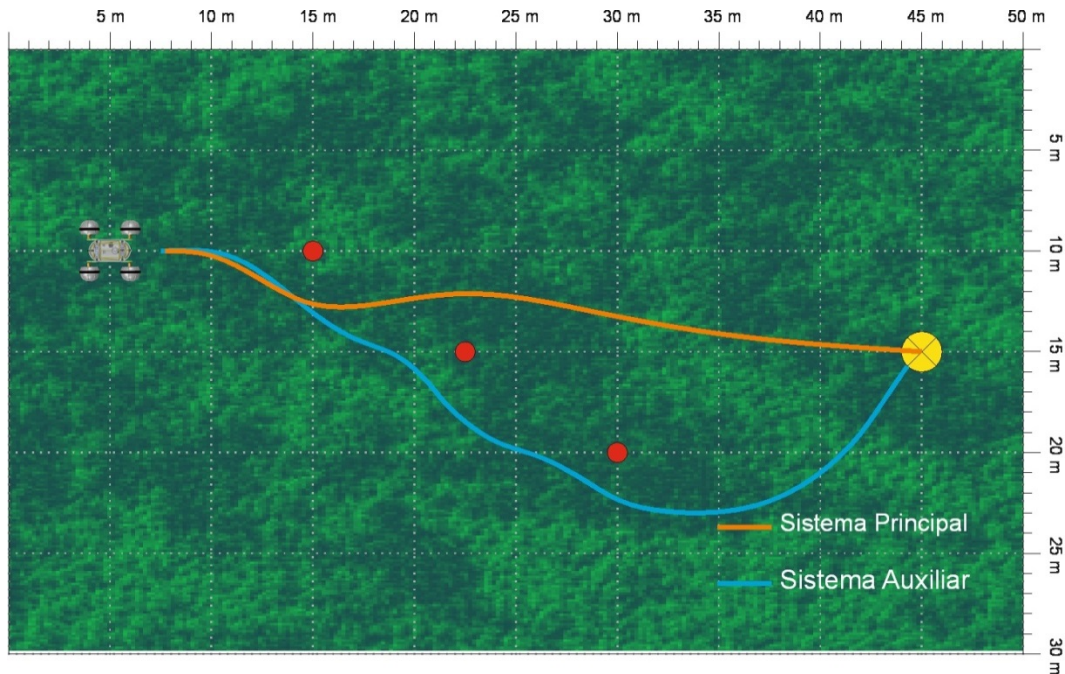


Figura 37. Experimento 3

- **Experimento 4:**

Similar ao experimento 1, não indicando um lado preferencial para desvio dos obstáculos. Conforme apresentado na Figura 38, o SPN foi um pouco mais eficiente que o SAN, uma vez que os equipamentos deste não permitem detectar obstáculos a uma distância maior do que cinco metros. Assim, o robô controlado pelo SAN só consegue se esquivar e construir uma rota alternativa até o objetivo após detectar a presença dos obstáculos

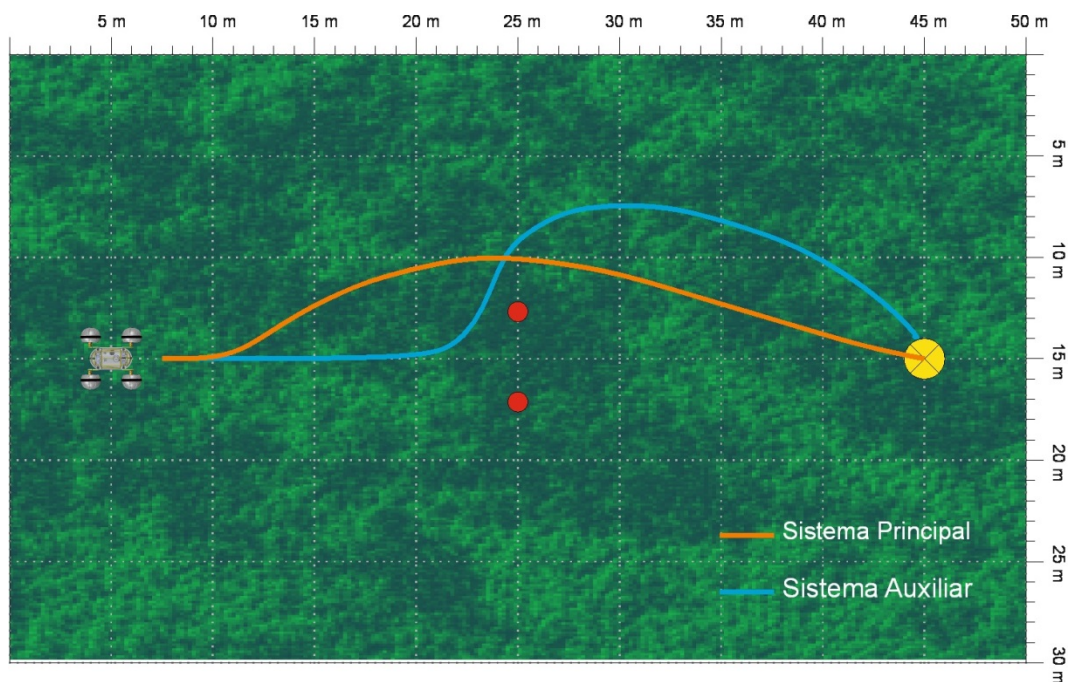


Figura 38. Experimento 4.

- **Experimento 5:**

O objetivo era forçar o robô a girar para a direita e se deslocar rumo ao objetivo. Observa-se que o SPN consegue construir uma melhor rota, ao passo que o SAN (linha azul), devido à limitação de 5 metros para a distância de aquisição de dados, não consegue detectar os outros obstáculos e segue pela esquerda. Conforme os detecta, procura se esquivar e prosseguir rumo ao objetivo.

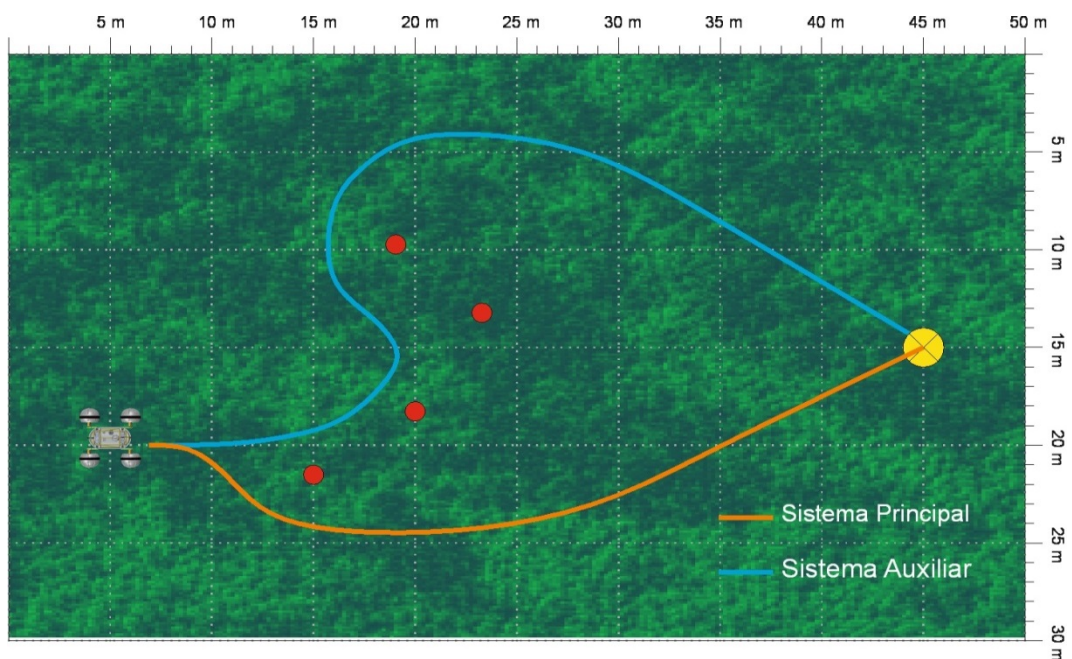


Figura 39. Experimento 5.

- **Experimento 6:**

Nesta distribuição aleatória de obstáculos, o comportamento do SAN foi bastante similar ao do SPN, apenas um pouco menos suave nas proximidades do obstáculo.

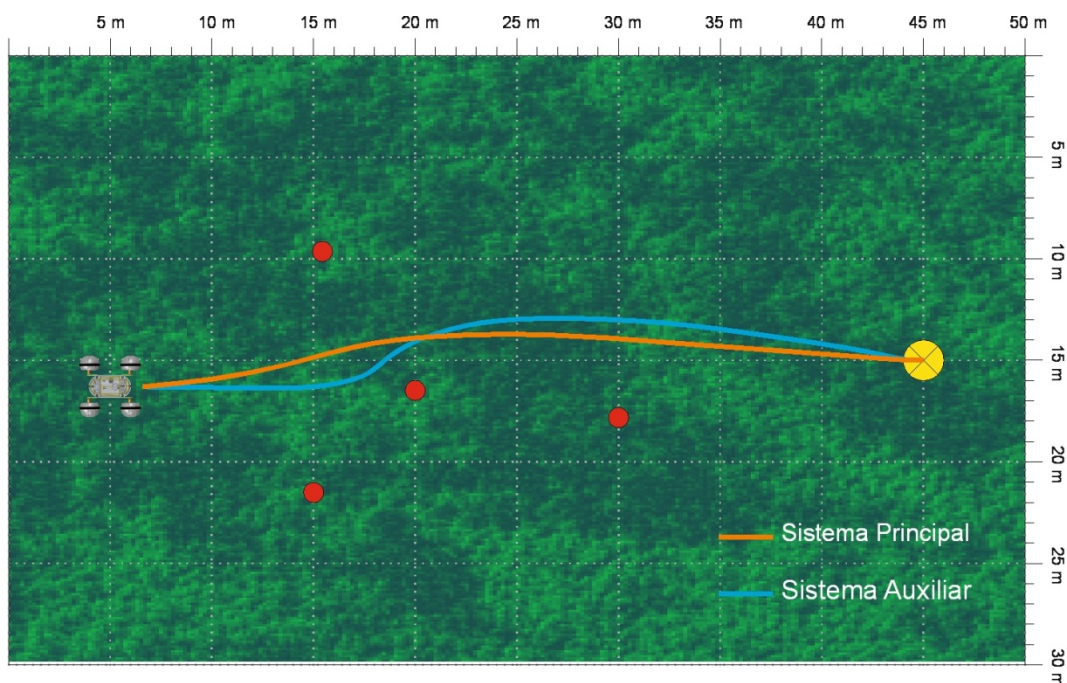


Figura 40. Experimento 6.

- **Experimento 7:**

Quatro obstáculos foram dispostos de modo a formar um corredor estreito que permitiria ao robô atingir o objetivo sem necessidade de se esquivar de obstáculo algum. Conforme apresentado na Figura 41, o SPN produz uma trajetória praticamente retilínea, enquanto que, quando controlado pelo SAN, o robô tende a se esquivar dos obstáculos detectados nas zonas laterais, perdendo um pouco de eficiência no trajeto realizado.

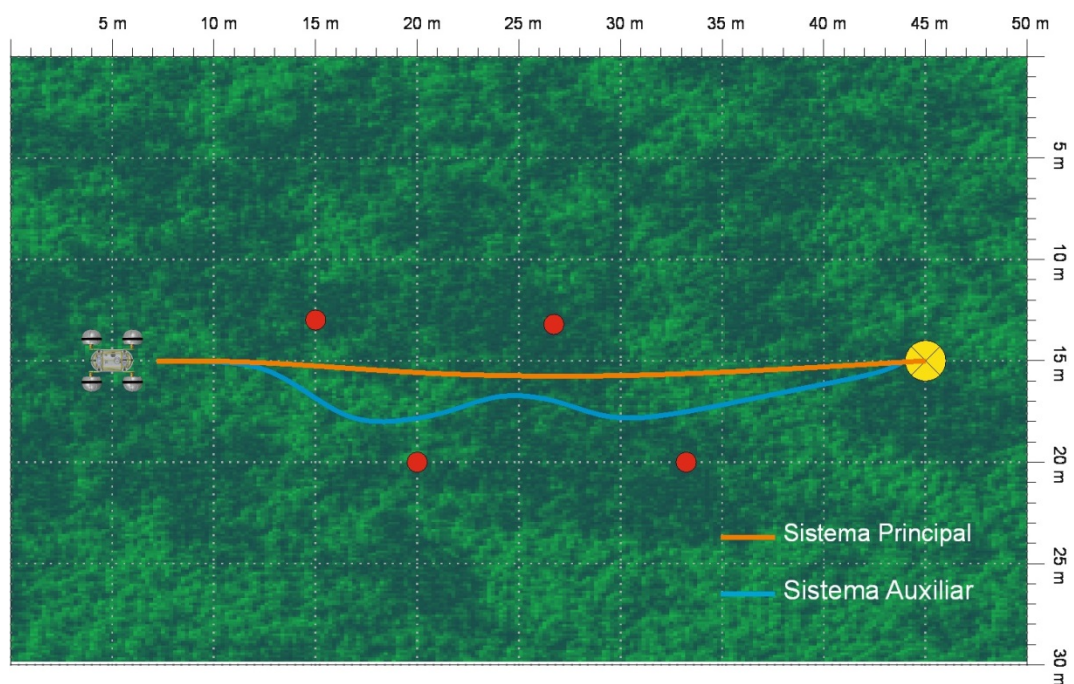


Figura 41. Experimento 7.

- **Experimento 8:**

Esta situação, mostrada na Figura 42, foi a única em que o robô controlado pelo SAN não conseguiu atingir o objetivo. A distribuição dos obstáculos faz com que o robô os detecte já muito próximo do objetivo, fazendo com que sempre se desvie de um deles e acabe por não atingir o objetivo.

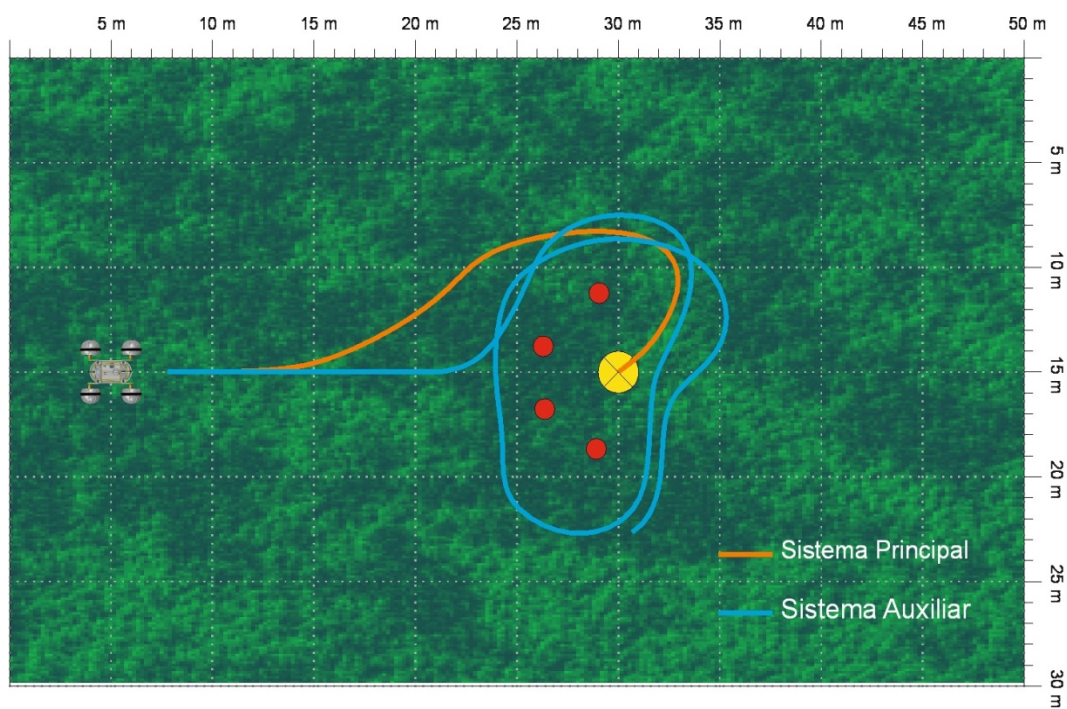


Figura 42. Experimento 8.

- **Experimento 9:**

Os quatro obstáculos forma um retângulo e distância entre os dois primeiros insuficiente para o robô passar entre eles. Os resultados obtidos foram bons tanto para o SAN como para o SPN, uma vez que em nenhum dos casos o RAHM colidiu com os obstáculos. A diferença está na eficiência das trajetórias.

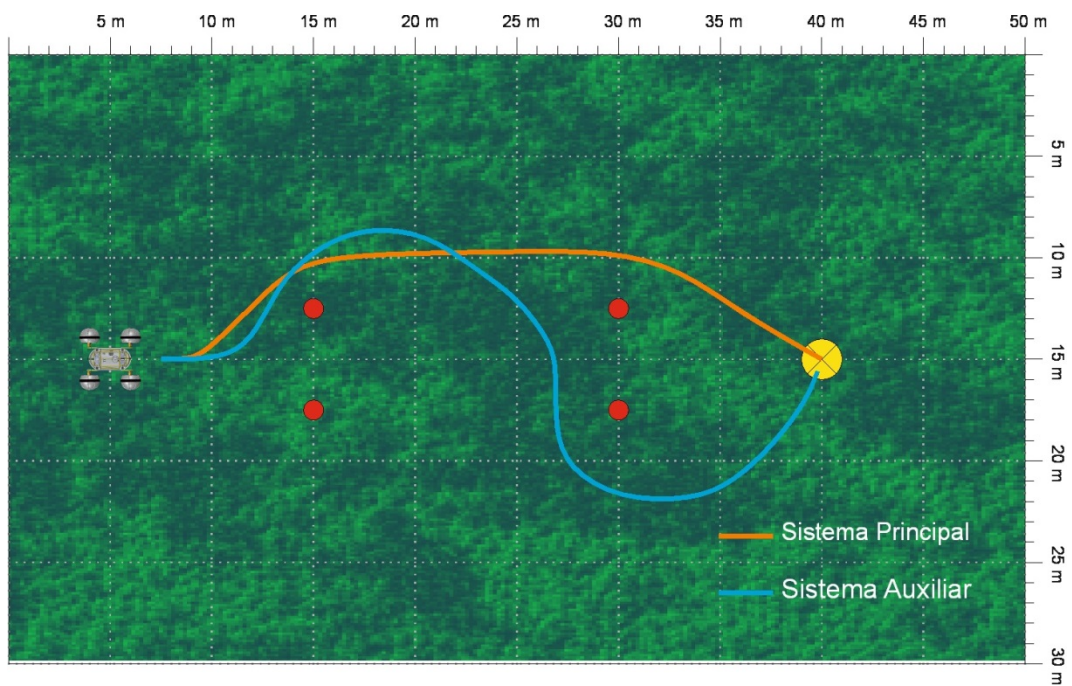


Figura 43. Experimento 9.

- **Experimento 10:**

Objetivo similar ao do experimento 6, com obstáculos distribuídos aleatoriamente. Os resultados foram similares aos do experimento 6, inclusive no que tange à eficiência.

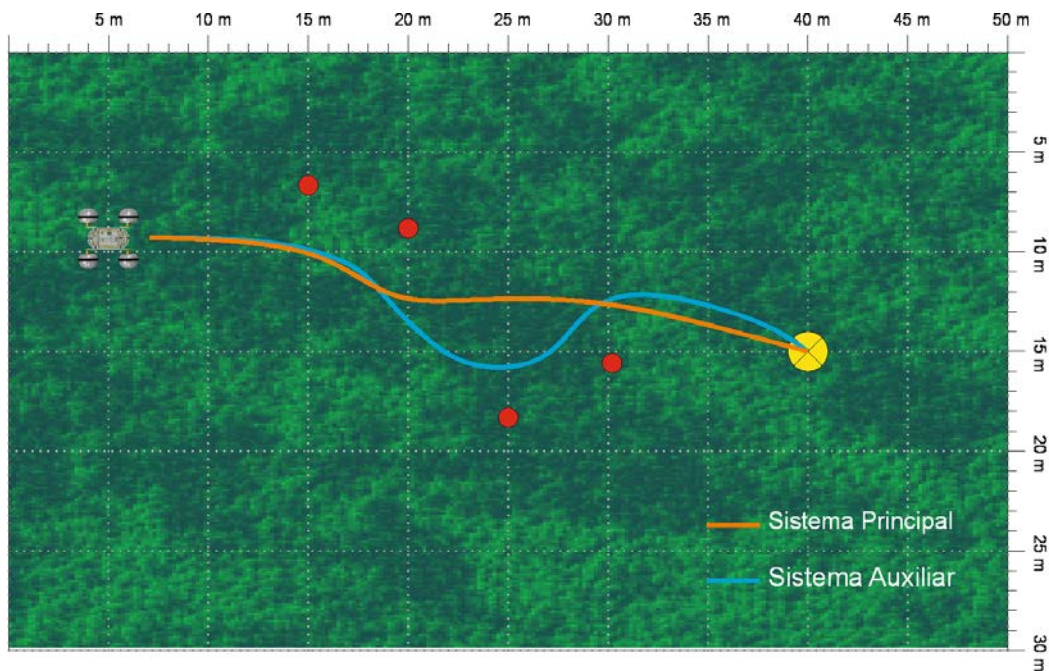


Figura 44. Experimento 10.

É importante ressaltar a diferença existente entre os dois sistemas no tocante aos sensores: o sensor laser, equipamento essencial no SPN, consegue mapear todo o ambiente dentro de um raio de 80 metros à frente, permitindo ao robô construir a rota mais eficiente, já que o ambiente é de apenas 50 metros; o SAN, por outro lado, tem que construir a rota em função das informações obtidas de uma visão máxima de 5 metros. Mesmo com esta desvantagem, o SAN apresentou bons resultados, com um custo financeiro e computacional bastante menor e com um consumo energético baixo.

5.2.1. Comparação dos Sistemas de Navegação

Fazendo uso as métricas descritas anteriormente, a Tabela 6 apresenta os resultados comparativos para cada sistema de navegação.

No caso do SAN, só é monitorada a memória do PC remoto, identificado na tabela com a sigla CR; no caso do SPN, são monitoradas tanto a memória do PC embarcado (CE) como a do PC remoto (CR). Percebe-se que o SAN é um sistema que não satura a placa de rede, pois só se utiliza 15% do seu poder. Além disso, a memória do PC remoto é utilizada em 50%, já levando em conta o custo computacional da execução do

controlador fuzzy programado em Visual Studio. O consumo de corrente do sistema é 6,5 Amperes em média.

No caso do SPN, observa-se um maior consumo de corrente, tempos mais curtos para realizar os diferentes experimentos e saturação bastante alta tanto no computador embarcado como na placa de rede para a comunicação.

Tempo (Seg)		Consumo de corrente (mA)		Corrente X tempo mA/h		Custo Computacional(%)		Dados na Rede (%)	
SAN	SPN	SAN	SPN	SAN	SPN	SAN	SPN	SAN	SPN
1:43	1:32	6468	10415	185,05	266,16	51% PR	76% CE 65% CR	12,8%	76%
2:04	1:30	6820	10115	234,91	252,87	48% PR	81% CE 67% CR	12,5%	82%
2:07	1:28	7430	10815	262,11	225,31	49% PR	80% CE 67% CR	13%	82%
1:27	1:15	7115	10085	171,94	210,1	52% PR	78% CE 65% CR	11,6%	86%
2:10	1:10	6680	10225	241,22	198,81	51% PR	90% CE 64% CR	14,3%	78%
1:03	55	6955	10180	121,71	155,52	51% PR	83% CE 63% CR	12,8%	80%
1:08	1:00	6200	9980	117,11	166,33	46% PR	77% CE 66% CR	11,9%	85%
-----	52	-----	10345	-----	149,42	-----	86% CE 73% CR	-----	82%
1:35	1:12	7050	10280	186,04	205,6	50% PR	82% CE 64% CR	12,7%	78%
1:48	1:10	6650	9720	199,5	189	47% PR	84% CE 64% CR	12,5%	75%

Tabela 6 Comparação entre o SAN e o SPN.

Destacam-se em cor cinza, na Tabela 6, os itens com melhor resultado para os sistemas utilizados. Percebe-se que o SAN é um sistema menos eficiente em relação ao tempo de percurso mas necessita de apenas um pouco mais da metade da energia requerida pelo SPN, além de não saturar nem a rede nem o computador necessário para o controlador fuzzy.

Os dez experimentos, além de avaliar o sistema auxiliar de navegação, foram importantes para testar o RAHM de forma geral, depurar todos os sistemas e encontrar problemas específicos que poderiam ser melhorados. Foi encontrado, por exemplo, um problema mecânico – roda dianteira direita mecanicamente freada – que ocasionava um maior consumo de corrente quando o robô girava para a direita. Outro problema encontrado foi a saturação da placa de rede quando do uso do SPN, configurando-se em um risco que poderia causar perda de comunicação.

6. Conclusões e Trabalhos Futuros

6.1. Conclusões

Esta dissertação teve como objetivo principal desenvolver um sistema de controle fuzzy para o Sistema Auxiliar de Navegação do Robô Ambiental Híbrido Médio, com as seguintes características: baixo consumo de energia, baixo custo financeiro e computacional e simplicidade no projeto. Desenvolveu-se também uma interface para monitorar os dados de entrada e saída, pois o controlador fuzzy é programado e executado remotamente.

Observou-se um bom desempenho do sistema desenvolvido (SAN) em um conjunto de dez experimentos, confirmando os benefícios da Lógica Fuzzy para o Sistema Auxiliar de Navegação do RAHM. Em nove dos experimentos, o SAN conseguiu atingir o objetivo, mostrando-se confiável frente às limitações de equipamento e às características desejadas.

Por não ser dotado de sensor laser – presente apenas no SPN –, que permite o mapeamento de todo o ambiente, o SAN mostrou-se, em geral, menos eficiente do que SPN em termos de trajetórias da origem ao objetivo. Por outro lado, o SAN apresenta custos bastante inferiores ao do SPN, configurando-se em uma excelente alternativa para situações de emergência.

6.2. Trabalhos Futuros

Os seguintes trabalhos futuros podem ser destacados:

- Substituir os sensores de ultrassom por um sensor laser básico, o que não se constituiria em uma diferença financeira muito elevada. Isto aumentaria a faixa de aquisição de dados simplificaria o projeto eletrônico (cabearamento e instalação) e o número de entradas do SIF.

- Utilizar um método de sintonia por aprendizado como o FRMLC (Reference Model Learning Control) (Wesley, 1998), para adaptação do SIF a eventuais mudanças de características do robô, como, por exemplo, peso ou atrito nas rodas.
- Possibilitar a configuração de parâmetros do SIF na interface gráfica, pois a atual não permite efetuar modificações no SIF.

Referências Bibliográficas

A. CHOHRA, A. FARAH, e M. BELLOUCIL. **Neuro-fuzzy expert system E-S-CO-V for the obstacle avoidance of intelligent autonomous vehicles (IAV)**, em Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97, Grenoble, France, págs. 1706-1713.

ARDUINO GENUINO. **Controller Board** Disponível em: < <http://www.Arduino.cc/>>. Acesso em 12/03/2015.

A. SAFFIOTTI. **Fuzzy Logic in Autonomous Navigation**. Springer-Physica Verlag. Pages 3-24. 2001.

A. SAFFIOTTI. **Handling uncertainty in control of Autonomous robots**. Artificial Intelligence Today. Springer-Verlag. Pages 381-408. 1999.

A. SAFFIOTTI. **The Uses of Fuzzy Logic in Autonomous Robot Navigation**. Soft Computing 1 (4):180-197. 1997.

BAJESTANI, N. e S. ZARE, A. **Application of optimized Type 2 fuzzy time series to forecast Taiwan stock index**. 2nd International Conference on Computer, Control and Communication. p. 1-6. 2009.

BERKAN, C., e TRUBATCH, S. **Fuzzy Systems Design Principles**. IEEE Press. 1996.

BING-FEI WE et al. **GPS navigation based autonomous driving system design for intelligent vehicles**, em 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, 2007, págs. 3294-3299.

CHEN, G. e PHAM, T. T. **Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems**, CRC Press, 2000.

CHIU, CHUI-YU; PARK, y CHAN S. **Fuzzy cash flow analysis using present worth criterion**. The Engineering Economist, Vol 39, 113-137, 1994.

CHEN-KUI LEE, CHUN-LIANG LIN, e BING-MIN SHIU. **Autonomous vehicle parking using artificial intelligent approach**, in 2009 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, 2000, págs. 496-501.

C. M. KUMILE e N. S. TLALE. **Intelligent distributed fuzzy logic control system (IDFLCS) of a mecanum wheeled autonomous guided vehicle**, em

IEEE International Conference Mechatronics and Automation, 2005, Niagara Falls, Ont., Canada, págs. 131-137.

COOK, G. **Mobile Robots: Navigation, Control and Remote Sensing**, Wiley-IEEE Press, 324 p, 2011.

COX, EARL. **Fuzzy logic for business and industry**. Massachusetts: Charles River Media Inc., 1995.

C. ROGRIDO y C. ALDO. **Sistema de Guiado para un robot móvil basado en lógica difusa**. XV Congreso de la Asociación Chilena de Control Automático. Octubre de 2002.

DIRECT INDUSTRY. **Inspection Robot** Disponível em: <<http://www.directindustry.es/>>. Acesso em 21/05/2015.

D. DRIANKOV. **A Reminder on Fuzzy Logic**. Springer- Physica Verlag. Pages 25-47.2001.

GOMIDE F., GUDWIN, e R. TANSCHHEIT. (1995). **Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações**. Proc. 6th IFSA Congress -Tutorials, pp. 1-38, São Paulo, Brazil.

J. CORDOBA, L. RIOS, y C. RESTREPO. **Descripción de una plataforma móvil implementada para el estudio de técnicas inteligentes de navegación: PMETIN**. Scientia Et Technica. Ano XI, No.27 Abril de 2005.

JERRY M. MENDEL **Uncertain rule-Based Fuzzy Logic Systems**. Prentice Hall PTR, 6-11, 2001.

J. VELAGIC, B. LACEVIC. e B. PERUNICIC. **A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches**. Robotics and Autonomous Systems 54 Pages 989–1004. 2006.

K. AUFMANN, ARNOLD; GUPTA, e MADAN M. **Fuzzy mathematical models in engineering and management science**. Amsterdam: Elsevier Science Publishers B, 1988.

K. C. CHOEK, G. E. SMID, K. KOAYASHI, J. L. OVERHOLT, e P. LESCOS, **A fuzzy logic intelligent control system architecture for an autonomous leader-following vehicle**, em Proceedings of the 1997 American Control Conference (Cat. No.97CH36041), Albuquerque, NM, USA, págs. 522-526.

MAXON MOTORS. **Controller Systems** Disponível em: <http://www.maxonmotor.es>. Acesso em 02/02/2015.

M. HUANG, R. ZHANG, Y. MA, e Q. Yan, **Research on Autonomous Driving Control Method of Intelligent Vehicle Based on Vision Navigation**, em 2010 International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 2010, págs. 1-7.

M. PASQUIER, C. QUEK, W. L. TUNG, D. CHEN, e T. M. YEP, **Fuzzylot II: a novel soft computing approach to the realization of autonomous driving maneuvers for intelligent vehicles**, in 7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002. Singapore, págs. 746-751.

N. SALVI DOS REIS, E. GOES. **Visita técnica e sobrevôo à faixa de interferência do gasoduto coari- Manaus**. CENPES PETROBRAS, 2009.

N. KIM, G-H. YOON e D. LEE, **Intelligent Navigation and Control of an Autonomous Underwater Vehicle based on Q-Learning and Self-Organizing Control**, em ICROS-SICE International Joint Conference, Fukuoka, Japão, 2009, págs. 630-634.

PEDRYCZ, W. **Fuzzy Control and Fuzzy Systems**. Research Studies Press and John Wiley and Son Inc. London England. 1993.

PINHO, e ALEXANDRE F. **Uma contribuição para a resolução de problemas de programação de operações em sistemas de produção intermitentes flow-shop: A consideração de incertezas**. 1999.

R. SIMARRO, C. RICOLFE, J. NAVARRO, y J. SIMÓ. **Herramienta para la prueba de controladores Inteligentes aplicados a la robótica móvil**. Universidad Politécnica de Valencia. Departamento de Informática de Sistemas y Computadoras. 2002.

ROBOTSIERRA. **PIPE INSPECTION ROBOT**. Disponível em: <<http://www.robot.sierrachile.com//>>. Acesso em 21/05/2015.

REN C. LUO, T.-J. PAN, y C. M. ARAS. **Intelligent Motion Control System for Autonomous Vehicles Operating in a Factory Floor**, in Proceedings of the IEEE International Workshop on Intelligent Motion Control, Istanbul, Turkey, págs. 419-423.

R. FERREIRA, e A. BARRAL, **Relatório de reavaliação do Robô Ambiental Híbrido**. Fevereiro de 2012, Rio de Janeiro, págs. 6-15.

TANAKA, K. **An Introduction to Fuzzy Logic for Practical Applications**. Springer USA. 1997.

VISUAL STUDIO. **Development Applications** Disponível em: <<https://www.visualstudio.com/support>>. Acesso em 10/11/2014.

VON ALTROCK, **Constantin. Fuzzy logic and Neuro Fuzzy applications in business and finance**. New Jersey: Prentice Hall PTR, 1996.

YEN, JOHN; LANGARI, REZA, ZADETH e LOFTI A. **Industrial applications of fuzzy logic and intelligent systems**. New York: IEEE Press, 1994.

Anexo B– Configuração da Placa Conversora RS232/Ethernet

As placas Zuchi RS232/Ethernet são conversoras de protocolos capazes de receber dados enviados por um equipamento serial, empacotá-lo através de um protocolo de rede como TCP/IP ou UDP e transmiti-los a uma aplicação remota via socket e vice-versa.

Características Principais:

- Tamanho reduzido.
- Conexão rápida e simples entre dispositivos Ethernet e Serial.
- Diversos modos de gerenciamento do roteamento de pacotes.
- Dispõe de ferramenta de configuração amigável para facilitar a parametrização da operação do conversor.
- Suporta parametrização via interface Ethernet e via interface Serial.
- Suporta velocidade de 10 ou 100Mbps via Ethernet e máxima de 230Kbps via Serial.
- Suporta os protocolos TCP, UDP, IP, ARP, ICMP e Ethernet MAC.

A seguir apresenta-se o software que a placa possui para configuração de parâmetros.

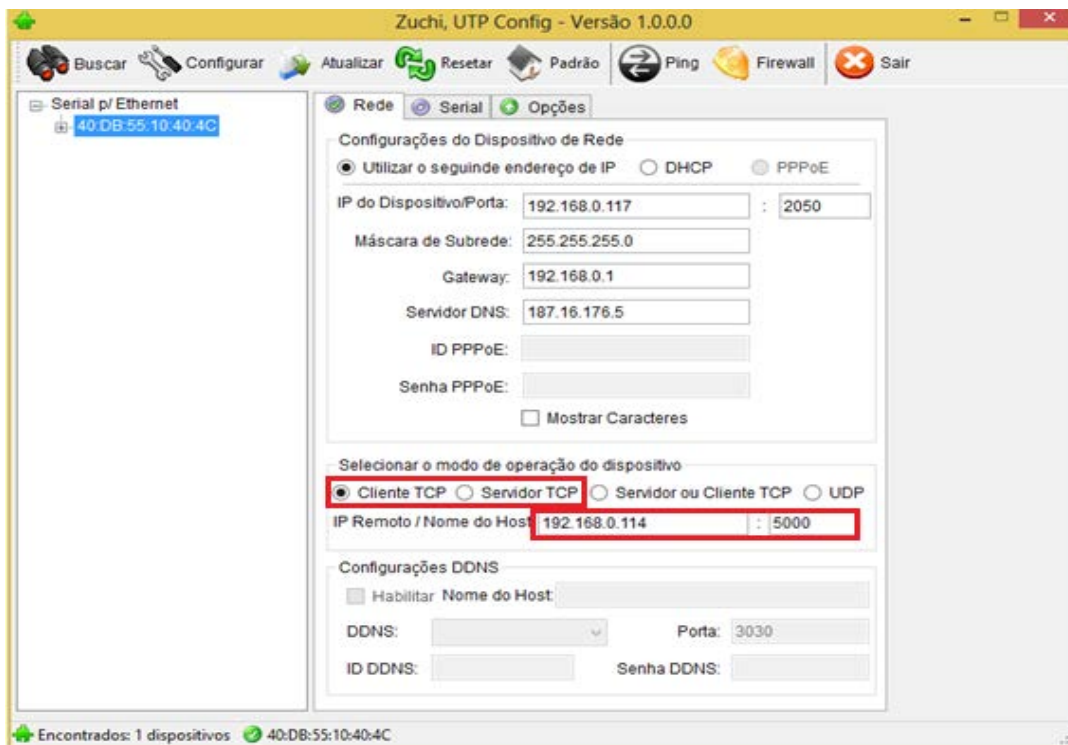


Figura 45. Software da placa conversora (parâmetros de configuração Ethernet).

O primeiro passo é conectar a placa com o PC através de um cabo Ethernet; a seguir, dá-se um click no botão Buscar, como se pode ver na Figura 45, e todos os parâmetros da placa nela configurados anteriormente serão mostrados:

- IP da placa
- Máscara de Subrede.
- Gateway
- Servidor DNS

Os dados das caixas vermelhas, como mostra a Figura 45, são os mais importantes a serem configurados. Na primeira caixa, o usuário configura a placa como cliente ou como servidor (em nosso caso, como cliente), e depois configura os dados:

- **IP remota:** que será a IP da outra placa que trabalhará como SERVIDOR.
- **Porta:** canal de comunicação entre as duas placas (neste caso, “5000”).

Na placa que operará como servidor, o procedimento é bem simples: o usuário deverá configurá-la como servidor e alocar a ela a porta de comunicação 5000.

No caso da comunicação serial, os parâmetros a configurar são a velocidade de transmissão de bits por segundo. Neste caso, a EPOS exige uma taxa de transferência de 115200 bps (como mostrado na Figura 46). Esta velocidade deverá estar configurada em ambas as placas, no computador e na placa controladora EPOS, para que possam se comunicar.

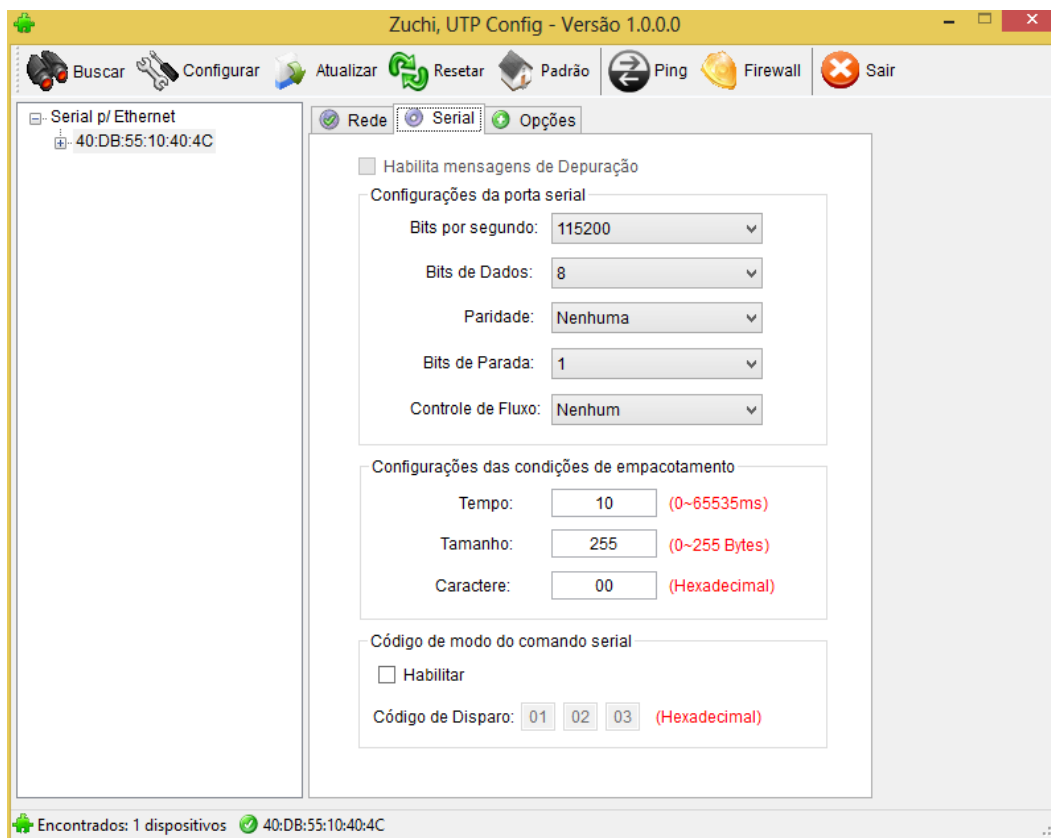


Figura 46. Ferramenta da placa conversora (Parâmetro Serial).

Outros parâmetros importantes são as configurações das condições de empacotamento. É importante que sejam os mesmos nas duas placas: tempo de “10ms”, tamanho do pacote de “255” e caractere hexadecimal “00”.

Anexo B– Programação do SIF com a Biblioteca Dotfuzzy no Visual Studio.

A implementação do Sistema de Inferência Fuzzy com o Dotfuzzy segue o procedimento explicado a seguir. O primeiro passo é a declaração das variáveis de entrada. No caso deste projeto, são seis entradas e um exemplo de configuração de uma delas é apresentado na Figura 47.

```
LinguisticVariable SensorAngular = new LinguisticVariable("SensorAngular");
SensorAngular.MembershipFunctionCollection.Add(new MembershipFunction("High_Left", -180, -180, -90, -45));
SensorAngular.MembershipFunctionCollection.Add(new MembershipFunction("Medium_Left", -90, -45, -45, 0));
SensorAngular.MembershipFunctionCollection.Add(new MembershipFunction("Center", -45, 0, 0, 45));
SensorAngular.MembershipFunctionCollection.Add(new MembershipFunction("Medium_Right", 0, 45, 45, 90));
SensorAngular.MembershipFunctionCollection.Add(new MembershipFunction("High_Right", 45, 90, 180, 180));
```

Figura 47. Programação da entrada do SIF com Dotfuzzy.

Definem-se, as características das variáveis de saída.

```
LinguisticVariable Dist_Frontal = new LinguisticVariable("Dist_Frontal");
Dist_Frontal.MembershipFunctionCollection.Add(new MembershipFunction("Close", 0, 0, 1.25, 2.5));
Dist_Frontal.MembershipFunctionCollection.Add(new MembershipFunction("Medium", 1.5, 2.5, 2.5, 3.5));
Dist_Frontal.MembershipFunctionCollection.Add(new MembershipFunction("Far", 2.5, 3.75, 5, 5));
```

Figura 48. Programação da Saída do SIF com Dotfuzzy.

Conforme mostrado na Figura 49, é necessário indicar a natureza das variáveis: entrada ou saída.

```
//Comando Para executar o Fuzzy
FuzzyEngine fuzzyEngineDist_Frontal = new FuzzyEngine();
FuzzyEngine fuzzyEngineDirecao = new FuzzyEngine();

//Sao definidas as saidas dos sistemas
fuzzyEngineDist_Frontal.Consequent = "Dist_Frontal";
fuzzyEngineDirecao.Consequent = "Direcao";
```

Figura 49. Definição de saídas e entradas.

Apresenta-se a seguir o código para a definição e programação das regras do sistema.

```
//Definicao das Regras do Sistema

fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule
("IF (SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule
("IF (SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close) AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));
```

Figura 50. Definição das regras do sistema.

A Figura 50 apresenta a estrutura de uma regra, com seus antecedentes e consequentes. No Dotfuzzy, as regras estão divididas em duas partes, pois são dois os consequentes.

Por ultimo, a defuzzificação (Figura 51):

```
var salida_Dist_Frontal = fuzzyEngineDist_Frontal.Defuzzify();
var salida_Direcao =fuzzyEngineDirecao.Defuzzify();
```

Figura 51. Defuzzificação

Anexo C– Base de Regras dos Sistemas de Inferência Fuzzy.

Neste anexo serão apresentadas as 270 regras do SIF geral, correspondendo a 135 do SIF1 (Sistema intermediário) e a 135 regras do SIF2 (Sistema de Controle), conforme implementadas no Visual Studio.

//Definição das Regras do SIF1 (Sistema Intermediário)

```

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));

    fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
    fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

```

```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

```



```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Far"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));

```



```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Medium"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Direcao IS Left"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Direcao IS Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Direcao IS Left"));
```



```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Far"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));

```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Medium"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Left) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Close) THEN Direcao IS Very_Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Medium) THEN Direcao IS Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Far) THEN Direcao IS Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Right"));
```

```
fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
```

```
fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Direcao IS Very_Left"));
```

```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

```



```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Far"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close) AND
(Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium) AND
(Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));

```

```

        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium) AND
(Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Center) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium) AND
(Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

```

```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

```



```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Far) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Far) AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Far"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Far) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));

```

```

        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Far) AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Far) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Far) AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS Medium_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

```

```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Close) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Right"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));

```

```

        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Medium) AND (Ultrassom3 IS Far) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Far) THEN Direcao IS Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Dist_Frontal IS Far"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Far) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Close)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Far) THEN Direcao IS Very_Left"));

```

```

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Medium) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS
Close) AND (Ultrassom3 IS Far) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Medium) AND (Ultrassom2 IS Far)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Dist_Frontal IS Close"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Far) AND
(Ultrassom3 IS Close) THEN Direcao IS Very_Left"));

        fuzzyEngineDist_Frontal.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Dist_Frontal IS Medium"));
        fuzzyEngineDirecao.FuzzyRuleCollection.Add(new FuzzyRule("IF
(SensorAngular IS High_Right) AND (Ultrassom1 IS Far) AND (Ultrassom2 IS Medium)
AND (Ultrassom3 IS Medium) THEN Direcao IS Very_Left"));

```

```
//Definição das Regras do SIF2 (Sistema de Controle).
```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM2 IS Rever"));

```



```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM1 IS Rever"));

```

```

        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

```



```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));

```

```

        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS Far)
THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS Far)
THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Close)
THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Close)
THEN PM2 IS Medium"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Left) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));

```

```

        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Medium"));

```



```

        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Center) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));

```



```

        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5 IS
Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Close) THEN PM2 IS Medium"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Medium) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5 IS
Far) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS Far)
THEN PM2 IS Medium"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Close) THEN PM1 IS Fast"));

```

```

        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Close) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND
(Ultrassom5 IS Close) THEN PM1 IS Rever"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND
(Ultrassom5 IS Close) THEN PM2 IS Rever"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND
(Ultrassom5 IS Medium) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND
(Ultrassom5 IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND
(Ultrassom5 IS Far) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Close) AND
(Ultrassom5 IS Far) THEN PM2 IS Fast"));

```

```

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Medium) AND
(Ultrassom5 IS Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Medium) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Medium) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Far) THEN PM1 IS Fast"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Medium) AND (Ultrassom4 IS Far) AND (Ultrassom5
IS Far) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM1 IS Slow"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Close) THEN PM2 IS Slow"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM1 IS Medium"));
        fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Medium) THEN PM2 IS Fast"));

        fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM1 IS Medium"));

```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Close) AND (Ultrassom5
IS Far) THEN PM2 IS Fast"));
```

```
fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM1 IS Fast"));
```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Close) THEN PM2 IS Medium"));
```

```
fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM1 IS Fast"));
```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Medium) THEN PM2 IS Medium"));
```

```
fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM1 IS Fast"));
```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Medium) AND (Ultrassom5
IS Far) THEN PM2 IS Medium"));
```

```
fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM1 IS Fast"));
```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Close) THEN PM2 IS Medium"));
```

```
fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM1 IS Fast"));
```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Medium) THEN PM2 IS Medium"));
```

```
fuzzyEnginePM1.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM1 IS Fast"));
```

```
fuzzyEnginePM2.FuzzyRuleCollection.Add(new FuzzyRule("IF (Direcao IS
Very_Right) AND (Dist_Frontal IS Far) AND (Ultrassom4 IS Far) AND (Ultrassom5 IS
Far) THEN PM2 IS Medium"));
```