Universidade Federal do Rio de Janeiro Instituto de Matemática Núcleo de Computação Eletrônica

Patrick Barbosa Moratori

Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs

> Rio de Janeiro 2006

Patrick Barbosa Moratori

Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs

> Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática / Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Orientadores: Prof. Dr. Josefino Cabral Melo Lima Docteur, Université Pierre et Marie Curie, 1992

Prof. Dr. Adriano Joaquim de Oliveira Cruz Ph.D., University of Southampton, 1988

> Rio de Janeiro 2006

M831

Moratori, Patrick Barbosa.

Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs / Patrick Barbosa Moratori. Rio de Janeiro, 2006.

128 f. il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica, 2006.

Orientadores: Josefino Cabral Melo Lima e Adriano Joaquim de Oliveira Cruz.

 Inteligência Computacional. 2. Sistemas de Controle – Teses. I. Lima, Josefino Cabral Melo (Orient.).
II. Cruz, Adriano Joaquim de Oliveira (Orient.).
III. Universidade Federal do Rio de Janeiro.
Instituto de Matemática. Núcleo de Computação Eletrônica. IV. Título. Patrick Barbosa Moratori

Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática do Instituto de Matemática e do Núcleo de Computação Eletrônica, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do grau de Mestre em Informática.

Rio de Janeiro, 20 de fevereiro de 2006.

Prof. Dr. Josefino Cabral Melo Lima Docteur, Université Pierre et Marie Curie, 1992

Prof. Dr. Adriano Joaquim de Oliveira Cruz Ph.D., University of Southampton, 1988

Prof. Dr. Marco Aurélio Cavalcanti Pacheco Ph.D., University College London, 1991

Prof. Dr. Felipe Maia Galvão França Ph.D., Imperial College London, 1994 Dedico este trabalho à minha família, por tornar possível mais esta fundamental conquista.

AGRADECIMENTOS

Agradeço a minha mãe Ruth, minhas irmãs Filinha e Raquel, meus cunhados Peixoto e Luciano, pela força e apoio incondicionais. Faço também um agradecimento ao Zé, Silvinha, tia Soninha e Katinha, pois mesmo distantes sempre mandaram boas energias.

Agradeço ao prof. Dr. Josefino Cabral Melo Lima, por ter me aceitado no programa de mestrado e ter acreditado no trabalho que desenvolvemos. A sua prontidão para resolver todas as situações tem um valor inestimável.

Faço um agradecimento especial ao prof. Dr. Adriano Joaquim de Oliveira Cruz, pois seus trabalhos em sala de aula motivaram a produção científica de artigos, sendo um momento ímpar para o desenvolvimento desta dissertação. A sua presteza também foi um presente, levando o grupo a tentar realizar trabalhos cada vez melhores.

Agradeço o prof. Dr. Marco Aurélio Cavalcanti Pacheco e o Prof. Dr. Felipe Maia Galvão França por terem aceitado o convite de participação da banca examinadora desta dissertação.

Agradeço aos amigos Emília Barra Ferreira, Márcia Valpassos Pedro e Marcus Vinicius Pereira pela parceria no desenvolvimento do primeiro protótipo do robô. Além das fundamentais contribuições, estiveram sempre presentes nos momentos mais desafiadores.

Agradeço também às amigas Laci Mary Manhães e a Leila Cristina Vasconcelos de Andrade pelo total suporte dado nas mais diversas situações e, principalmente, por me incentivarem a utilizar os materiais publicados como o escopo para o desenvolvimento desta dissertação. Agradeço aos fiéis amigos Raimundo José Macário Costa; George de Souza Alves, Jorge Juan Zavaleta Gavidia, Raquel Defelippo Rodrigues, Maria Teresa Andrade de Gouvêa, José Henrique Gandra e Silva, Gianna Oliveira Bogossian Roque, Austeclyneo Pereira, Pablo Nascimento Mendes, Claudia Cristina Paranhos Cruz, Cyntia Fernanda dos Santos, Sandro de Azambuja, Solange Altoé de Moura; Renata Pinheiro Cardoso, Jorge Fernando Silva de Araújo, José Adolfo Snajdauf de Campos, César Augusto Rangel Bastos, Ilan Chamovitz e Mariano Gomes Pimentel por todas as dicas e conselhos providos. Um obrigado em destaque para a Francine Ferreira Vaz, pelo "abre portas" junto ao grupo de informática na educação.

Agradeço também aos professores Claudia Lage Rebello da Motta, Adriana Benevides Soares, Marcos da Fonseca Elia, Fábio Ferrentini Sampaio, Lígia Alves Barros, Maria Luiza Machado Campos, Carlos Alberto da Silva Franco, Lilian Markenzon, Ângela Rocha dos Santos, Antonio Carlos Gay Thomé, Bruno Feijó, Esteban Walter Gonzalez Clua pelas fundamentais orientações e ajudas prestadas.

Agradeço aos professores José Antonio Aravena Reyes, Fernanda Cláudia Alves Campos e Maria Clicia Stelling de Castro da Universidade Federal de Juiz de Fora por me motivarem a dar continuidade à vida acadêmica.

Agradeço aos colegas Eugênio Silva, Pedro Demasi, Rafael de S. Tenório Cavalcanti, Viviane Soares Rodrigues Silva, Catia Araújo, Renato Torres Raposo pelo suporte em implementações, nas disciplinas e no acesso às maquinas do Laboratório de Inteligência Computacional (LabIC).

Agradeço a Lina Marchese Olivieiro dos Santos, Amélia Abigail Rosário de Almeida, Roselinda Passos Franco, Mônica Menezes Santos, Cláudia Taveres Sampaio, Sonia Barbosa Tupinambá, Adriana Furtado Lima, Oswaldo Vernet, Sérgio Rocha pela boa-vontade em resolver todas as questões administrativas e burocráticas junto ao NCE (Núcleo de Computação Eletrônica).

Agradeço também a "Tia Deise", Zezé, Regina por tornar possível realizar todos os trâmites necessários junto ao departamento de ciência da computação.

Agradeço ao pessoal do MBI (*Master of Business Information Technology*) e do suporte do NCE por liberar salas e prestar toda a ajuda possível para realizar os experimentos deste trabalho.

Agradeço também à professora Carlota da Cunha Parreira pelo auxílio na língua inglesa, além do seu constante incentivo para a escrita desta dissertação.

Agradeço a Márcia Martins Oliveira e a equipe do Colégio Pedro II (Lúcia, Sheila, Luciene e Vera) por apoiarem as investidas que foram realizadas no mestrado.

Agradeço aos amigos Viviane Kawamura, Kele Teixeira Belloze, Luciana Campos, Waldemar Lopes, Tathyane Hofke, Tatiana Viana, Maria Silvia Ferreira, Naline Barros Martins, Alexandre Ferreira, Jacqueline Mariano, Jean Rojas e Camille Serpa, por conseguirem me fazer desgrudar do computador e dar uma relaxada de vez em quando.

Agradeço também a minha querida amiga Lívia Maria Santiago por me motivar, apoiar e sempre estar ao meu lado, mesmo quando não pude dar-lhe toda a atenção merecida.

Finalmente, agradeço a todos que contribuíram de alguma forma para tornar este sonho possível.

RESUMO

MORATORI, Patrick Barbosa. Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs. Rio de Janeiro, 2006. Dissertação (Mestrado em Informática) – Instituto de Matemática/Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

O objetivo principal deste trabalho é realizar um estudo de estabilidade e robustez de controladores nebulosos. Um simulador foi desenvolvido para controlar um robô deslocando-se por um mundo virtual, que contém obstáculos randomicamente distribuídos. Este modelo foi implementado e provou ter êxito no alcance do objetivo nos testes realizados. Regras foram removidas deste sistema para simular situações nas quais o projetista não possui todo o conhecimento sobre um determinado problema, ou controladores em momentos de falha. Os experimentos realizados indicam que os sistemas de controle baseados em lógica nebulosa podem funcionar adequadamente mesmo sob condições adversas.

ABSTRACT

MORATORI, Patrick Barbosa. Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs. Rio de Janeiro, 2006. Dissertação (Mestrado em Informática) – Instituto de Matemática/Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

The main goal of this work is to study the stability and robustness of fuzzy control systems. A simulator has been developed in order to guide a robot through a virtual world populated with randomly placed obstacles. This model was engineered and proved to be successful reaching the goal in all designed tests. Rules were removed from the system to simulate situations when the designer does not have complete knowledge about the problem, or an out of order controller. Our experiments indicate that fuzzy control systems can perform reasonably under adverse conditions.

LISTA DE FIGURAS

Figura 2.1: Conjunto IDOSO na abordagem booleana (a) e nebulosa (b)	8
Figura 2.2: Variável nebulosa idade	9
Figura 2.3: Interseção entre os conjuntos NÃO ALTO e ALTO da variável nebu	losa
altura	10
Figura 2.4: União entre os conjuntos NÃO ALTO e ALTO da variável nebulosa	altura
	11
Figura 2.5: Exemplo de regras de inferência	13
Figura 2.6: Estrutura básica de um controlador nebuloso	14
Figura 2.7: Modelo clássico de Mamdani	16
Figura 2.8: Modelo clássico de Larsen	17
Figura 2.9: Modelo de interpolação de Takagi-Sugeno	18
Figura 2.10: Modelo de interpolação de Tsukamoto	19
Figura 2.11: Representação do caminhão e a área de estacionamento	21
Figura 2.12: Representação dos conjuntos nebulosos do caminhão	23
Figura 2.13: Composição dos conjuntos nebulosos e o valor defuzzyficado pelo)
método do centro de gravidade	25
Figura 3.1: Diagrama do mundo virtual com obstáculos e o robô simulado	33
Figura 3.2: Região de colisão do robô simulado	34
Figura 3.3: Detecção dos obstáculos randômicos	35
Figura 3.4: Cálculo da distância entre o robô e um obstáculo específico	37
Figura 3.5: Interseções da reta direção do robô com a reta delimitadora da pare	ede
para a determinação do ponto x _p	38
Figura 3.6: Estrutura do módulo nebuloso do sistema de controle do robô	42

Figura 3.7: Gráfico das funções de inclusão trapézio (a) e triângulo (b)43	
Figura 3.8: Os conjuntos das variáveis do sistema e as funções de inclusão44	
Figura 3.9: Exemplo 1 de colisão iminente para o robô49	
Figura 3.10: Exemplo 2 de colisão iminente para o robô49	
Figura 3.11: Estrutura atualizada do módulo nebuloso50	
Figura 3.12: Os conjuntos da variável <i>y</i> _r 50	
Figura 3.13: Exemplo 1 com configuração (5, 15, -80°)52	
Figura 3.14: Exemplo 2 com configuração (5, 50, 90°)52	
Figura 3.15: Exemplo 3 com a configuração (5, 10, -80°)52	
Figura 3.16: Exemplo 4 com a configuração (5, 89, 60°)52	
Figura 3.17: Exemplo 5 com configuração (5, 85, -80°)52	
Figura 3.18: Exemplo 6 com configuração (5, 20, 60°)52	
Figura 3.19: Exemplo 7 com configuração (5, 40, -60°)53	
Figura 3.20: Exemplo 8 com configuração (5, 75, -90°)53	
Figura 3.21: Exemplo 9 com configuração (5, 10, -60°)53	
Figura 3.22: Exemplo 10 com configuração (3, 60, -20°)53	
Figura 3.23: Exemplo 11 com configuração (4, 68, -74°)53	
Figura 3.24: Exemplo 12 com configuração (4, 20, 60°)53	
Figura 4.1: Diagrama do mundo virtual com obstáculos58	
Figura 4.2: Colisão do robô com um obstáculo durante a trajetória (a) e o arquétipo	
delimitado para representar o ponto onde tal amostra ficou estagnada (b)60	
Figura 4.3: Posicionamentos de alcance das amostras no sistema com 37 (a), 36 (b) e	Э
35 regras (c)61	
Figura 4.4: Posicionamentos de alcance das amostras no sistema com 29 (a), 28 (b) e	Э
27 regras (c)61	

Figura 4.5: Posicionamentos de alcance das amostras no sistema com 11 (a), 9 (b) e
8 regras (c)61
Figura 4.6: Posicionamentos de alcance das amostras no sistema com 3 (a), 2 (b) e 1
regra (c)61
Figura 4.7: Cálculo do erro de trajetória62
Figura 4.8: Dois exemplos de configurações, identificando os pontos onde as
amostras ficaram estagnadas63
Figura 4.9: Exemplo de uma trajetória do robô no mundo virtual com obstáculos (a) e
variação do ângulo ϕ com a horizontal (b)64
Figura 4.10: Média final do erro de trajetória (a) e o desvio padrão (b)66
Figura 4.11: Média final da quantidade de amostras que atingia o objetivo67
Figura 4.12: Quantidade de amostras que atingia o objetivo segundo um critério
particular de remoção68
Figura 4.13: Quantidade de amostras que atingia o objetivo segundo critérios
específicos de remoção69
Figura 4.14: Reformulação da estrutura do módulo nebuloso70
Figura 4.15: Os conjuntos da variável de saída Δ 70
Figura 4.16: Exemplo de uma trajetória do robô com as novas configurações (a),
variação do ângulo com a horizontal (b) e a variação do deslocamento (c)73
Figura 4.17: Média final do erro de trajetória (a) e o desvio padrão (b)75
Figura 4.18: Média final da quantidade de amostras que atingia o objetivo76
Figura 4.19: Quantidade de amostras que atingia o objetivo segundo critérios
específicos de remoção77
Figura 4.20: Exemplo de uma trajetória do robô na primeira abordagem (a), variação
do ângulo com a horizontal (b) e a variação do deslocamento (c)

Figura 4.21: Exemplo de uma trajetória do robô na segunda abordagem (a), variação
do ângulo com a horizontal (b) e a variação do deslocamento (c)
Figura 4.22: Quantidade de amostras que atingiram o objetivo final para as duas
configurações do controlador81
Figura 4.23: Média da quantidade de passos realizados (a) e do tempo gasto no
processamento de cada amostra (b), para as duas configurações do
controlador82
Figura 4.24: Os conjuntos modificados e as funções de inclusão
Figura 4.25: Exemplo de uma trajetória do robô com as novas diretivas (a), variação
do ângulo com a horizontal (b) e a variação do deslocamento (c)86
Figura 4.25: O mundo virtual e a área de contorno aceitável
Figura 4.26: Média final do erro de trajetória (a) e o desvio padrão (b)88
Figura 4.27: Média final da quantidade de amostras que atingia o objetivo88
Figura 4.28: Quantidade de amostras que atingia o objetivo segundo critérios
específicos de remoção89
Figura 4.29: Exemplo de uma trajetória do robô na primeira abordagem (a), variação
do ângulo com a horizontal (b) e a variação do deslocamento (c)91
Figura 4.30: Exemplo de uma trajetória do robô na segunda abordagem (a), variação
do ângulo com a horizontal (b) e a variação do deslocamento (c)91
Figura 4.31: Exemplo de uma trajetória do robô na terceira abordagem (a), variação
do ângulo com a horizontal (b) e a variação do deslocamento (c)92
Figura 4.32: Quantidade de amostras que atingiram o objetivo final para as três
configurações do controlador94

Figura 4.33: Média da quantidade de passos realizados (a) e do tempo gasto no	
processamento de cada amostra (b), para as três configurações do	
controlador	95

LISTA DE TABELAS

Fabela 2.1: Base de regras do controlador do caminhão	23
Tabela 2.2: Regras ativas no exemplo de iteração	24
Tabela 3.1: Detalhamento das características dos conjuntos nebulosos do sistema	44
Fabela 3.2: Base de regras para o controlador do robô	47
Tabela 3.3: Detalhamento das características dos conjuntos nebulosos de y_r	50
Fabela 4.1: Coordenadas dos obstáculos fixados	58
Tabela 4.2: Seqüência de remoção de regras para o critério seqüencial por	
coluna (a) e para a extração periféricas / centrais (b)	35
Fabela 4.3: Detalhamento das características dos conjuntos nebulosos de Δ	71
Tabela 4.4: Base de regras para o novo sistema	71
Tabela 4.5: Nova seqüência de remoção para o critério seqüencial por coluna	74
Tabela 4.6: Nova seqüência de remoção para o critério periféricas / centrais	74
Fabela 4.7: Análise geral das amostras 8	33
Fabela 4.8: Análise das amostras que atingiram a meta	33
Tabela 4.9: Detalhamento das características dos conjuntos nebulosos modificados	\$
	35
Tabela 4.10: Modificações na base de conhecimento para a nova configuração do	
sistema	35
Fabela 4.11: Análise geral das amostras	96
Fabela 4.12: Análise das amostras que atingiram a meta	96

LISTA DE ABREVIATURAS

ANFIS Adaptive Neuro-fuzzy Inference System

SUMÁRIO

CAPÍTULO 1	1
1. INTRODUÇÃO	1
CAPÍTULO 2	4
2. LÓGICA NEBULOSA	4
2.1 Introdução	4
2.2 Variáveis e conjuntos	7
2.3 Implicações lógicas e regras de inferência	11
2.4 Sistemas nebulosos	13
2.4.1 Interface de fuzzyficação	14
2.4.2 Base de conhecimentos	14
2.4.3 Procedimento de inferência	15
2.4.4 Interface de defuzzificação	19
2.5 Exemplo	20
2.6 Estabilidade de sistemas de controle nebulosos	26
2.7 Projeto de controladores nebulosos	29
2.7 Considerações parciais	32
CAPÍTULO 3	33
3. DESCRIÇÃO DO SISTEMA NEBULOSO DE CONTROLE DO ROBÔ	33
3.1 Visão geral	33
3.2 Sensores do robô	34
3.2.1 Detecção dos obstáculos randômicos	35
3.2.2 Detecção das paredes	37
3.2.3 Detecção do obstáculo mais próximo	40

3.3 Movimentação do robô	40
3.4 Intervalos dos parâmetros do sistema	40
3.5 Módulo fuzzy	41
3.5.1 Estrutura do modelo	42
3.5.2 Conjuntos nebulosos	42
3.5.3 Regras nebulosas	45
3.5.4 Retificação da estrutura, adição de conjuntos e de regras ao módulo	2
nebuloso	50
3.6 Exemplos	51
3.7 Considerações parciais	54
CAPÍTULO 4	56
4. ESTUDO DA ESTABILIDADE E ROBUSTEZ DO SISTEMA	56
4.1 Características gerais	56
4.1.1 Configuração dos parâmetros	57
4.1.2 Critérios de remoção	59
4.1.3 Exemplo de aplicação	60
4.1.4 Características da análise de estabilidade	62
4.2 Experimentos	63
4.2.1 Estudo de estabilidade do modelo original	64
4.2.1.1 Características específicas	65
4.2.1.2 Simulações e resultados	66
4.2.2 Estudo de estabilidade de um controlador com velocidade variável	69
4.2.2.1 Características específicas	73
4.2.2.2 Simulações e resultados	75
4.2.3 Estudo comparativo do desempenho entre os dois sistemas prévios	78

4.2.3.1 Características específicas8	0
4.2.3.2 Simulações e resultados	31
4.2.4 Estudo de estabilidade de um controlador otimizado com velocidade	
variável	34
4.2.4.1 Características específicas	37
4.2.4.2 Simulações e resultados	57
4.2.5 Comparativo geral de desempenho entre as diferentes configurações d	0
sistema)0
4.2.5.1 Características específicas)3
4.2.5.2 Simulações e resultados) 3
4.3 Considerações parciais9)7
CAPÍTULO 5	9
5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	9
5.1 Considerações finais9	9
5.2 Trabalhos futuros10	11
REFERÊNCIAS10)3

1

1. INTRODUÇÃO

As informações presentes no dia-a-dia possuem aspectos como a imprecisão e a incerteza, que são característicos da linguagem humana. Termos como "um pouco", "por volta de", "muito" e outros são comumente utilizados. A teoria clássica dos conjuntos e a teoria da probabilidade conseguem modelar certas situações, porém, nem sempre possibilitam captar as riquezas das afirmações fornecidas pelos seres humanos.

A teoria dos conjuntos nebulosos, desenvolvida por Zadeh (1965), trata estes aspectos vagos da informação. Desta forma, os conceitos clássicos definidos por Aristóteles passam a ser vistos como um caso particular desta abordagem mais flexível. Em 1978, Zadeh também descreveu a teoria das possibilidades para o tratamento da incerteza (ZADEH, 1978), sendo esta menos restritiva que a da probabilidade clássica. Em um discurso usual, esta menor restrição fica perceptível, como, por exemplo, ser mais fácil afirmar que um evento é possível ao invés de ser provável (SANDRI e CORREA, 1999).

A teoria dos conjuntos nebulosos, quando aplicada em um contexto lógico, como os sistemas baseados em conhecimento, é conhecida como lógica nebulosa (difusa ou "*fuzzy*"). Esta constitui uma das tecnologias atuais bem sucedidas para o desenvolvimento de modelos para o gerenciamento de processos sofisticados (CASTRO, 1995), (MENDEL, 1995), (DUTTA, 1993), (DRIANKOV, HELLENDOORN e RENFRANK, 1993), (LEE, 1990). Com sua utilização, requerimentos complexos podem ser desenvolvidos em controladores simples, de fácil manutenção e baixo

custo. Então, quando o modelo matemático está sujeito às incertezas, a aplicação dos controladores nebulosos se torna especialmente interessante (GUERRA, SANDRI e SOUZA, 1997), (WOODARD, 1996), (CONWAY *et al.*, 1994), (WEIL, PROBST e GRAF, 1994), (CHIU e CHAND, 1994), (KARR e GENTRY, 1993). Basicamente, um controlador deste tipo é um sistema que possui um conjunto de regras do tipo "Se <premissas> então <conclusões>", que definem as ações de controle em função dos valores que as variáveis de estado podem assumir (KLIR e FOLGER, 1988), (D'AMORE, SAOTOME e KIENITZ, 1997). Estas informações são mapeadas por conjuntos nebulosos que utilizam os termos lingüísticos para a delimitação das características.

Dentro deste contexto, observa-se que o desenvolvimento de sistemas confiáveis que simulam condições reais caracteriza um processo elaborado tanto na criação de software quanto de hardware. Desta forma, aponta-se, neste trabalho, que a utilização de recursos da lógica nebulosa permite minimizar a implementação, além de preservar a robustez dos modelos, pois um projetista mesmo sem conhecer todas as informações de um contexto específico pode construir um controlador simples, que realize de maneira adequada os tratamentos necessários para o alcance dos objetivos. Esta qualidade fica sugerida, pois um sistema de controle nebuloso foi gerado e testes de estabilidade foram realizados (MORATORI *et col*, 2005c), (MORATORI *et col*, 2005d), (MORATORI *et col*, 2005c), (MORATORI *et col*, 2005d), (MORATORI *et col*, 2005e), observando que êxitos nos resultados ainda eram obtidos, mesmo após a extração de uma quantidade razoável de conhecimentos do módulo de inferência.

Assim, o objetivo principal desta dissertação é realizar um estudo de robustez e estabilidade dos sistemas difusos de controle. A contribuição central consiste em

mostrar a confiabilidade destes controladores, indicando que um funcionamento adequado pode ser obtido mesmo quando condições adversas são configuradas.

O sistema desenvolvido para este trabalho é semelhante ao modelo do caminhão descrito por Kosko (1992), delimitando um modelo clássico do tipo Mamdani. Desta forma, operadores de interseção e de união são utilizados para a manipulação de conjuntos e o método de "centro de gravidade" é aplicado para gerar as ações de controle.

O Capítulo que se segue apresenta uma explanação sobre os conceitos fundamentais relacionados à lógica nebulosa, mostrando as definições básicas da teoria dos conjuntos; a interpretação dos conceitos de interseção, união e complemento; as regras de inferência aplicadas e o processo de desenvolvimento destes controladores. Adicionalmente, também são mostradas as diretivas que são utilizadas para a análise de estabilidade destes sistemas.

No Capítulo 3, é realizada a descrição do controlador implementado, especificando com detalhes todo o processo de desenvolvimento e a definição das estratégias de controle adotadas.

O Capítulo 4 apresenta os experimentos realizados nesta pesquisa, descrevendo os estudos de estabilidade e robustez que foram desenvolvidos para avaliar o desempenho das tomadas de decisão do sistema previamente estabelecido. Análises comparativas e modelagens adicionais foram incrementadas com o intuito de observar os possíveis comportamentos deste controlador frente a diferentes situações.

Finalmente, o Capítulo 5 mostra as considerações finais e as perspectivas para trabalhos futuros.

4

2. LÓGICA NEBULOSA

O objetivo deste Capítulo é realizar uma explanação sobre os conceitos essenciais relacionados à lógica nebulosa. Este aporte teórico fundamenta os principais aspectos e características relacionados ao processo de desenvolvimento dos sistemas de controle nebulosos. Além disso, também são definidas as diretivas que são aplicadas à análise de estabilidade destes controladores.

2.1 INTRODUÇÃO

O homem possui a habilidade de lidar com processos complexos em seu dia-a-dia, que envolvem muitas vezes informações aproximadas. A estratégia adotada pelos operadores humanos para tratar tais situações também é de origem imprecisa, devido ao fato de que as pessoas utilizam comumente termos lingüísticos em suas tomadas de decisão, empregando palavras como "muito", "pouco", "raramente", "perto", "longe" etc (CRUZ, 2004).

A lógica clássica de Aristóteles classifica objetos em categorias bem definidas, onde "Tudo de ser ou não ser, seja no presente ou no futuro". Embora a cultura ocidental, fundamentada nesta lógica binária, resolva de forma extraordinária muitos problemas, é necessário que se preencham os espaços não adequadamente endereçados pelos métodos tradicionais. Neste sentido, a lógica nebulosa (*fuzzy*) permite ver os "graus de verdade" entre o ser e o não ser, como tons de cinza entre o preto e o branco, caracterizando, assim, uma generalização da lógica aristotélica.

A incapacidade da lógica tradicional em lidar com problemas reais foi constatada por pensadores importantes como Bertrand Russel e Albert Einstein. Os seguintes pensamentos são atribuídos a Russel mostrando seu posicionamento: "Toda linguagem é vaga"; "Toda lógica tradicional habitualmente assume que símbolos precisos estão sendo empregados. Portanto, não é aplicável à vida terrestre, mas somente para uma imaginária existência celestial" e "Não se imagina como tudo é vago até que se tente fazê-lo de maneira precisa" (RUSSEL, 1923). A Einstein é atribuída a seguinte afirmação: "Quando as leis da matemática referemse à realidade, elas não estão certas. Quando estas leis estão certas, elas não se referem à realidade" (EINSTEIN, 1954).

Na década de 1920, o matemático polonês Jan Lukasiewicz desenvolveu uma lógica multivalorada (LUKASIEWICZ, 1920), argumentando principalmente sobre a lei da contradição, ou seja, embora contrária à natureza psicológica do homem, é perfeitamente plausível em termos matemáticos uma afirmação do tipo "X e Y são e não são, ao mesmo tempo", desde que os graus de verdade não sejam apenas bivalentes (verdadeiro e falso).

Mas foi em 1965 que a Teoria dos Conjuntos *Fuzzy* foi concebida pelo professor Lofti Zadeh da Universidade de Berkeley na Califórnia. O objetivo foi introduzir uma lógica mais flexível, denominada *Fuzzy Logic* (lógica nebulosa ou difusa), criando um método de tradução de expressões verbais (vagas, imprecisas e/ou qualitativas) em valores numéricos tratáveis (ZADEH, 1965).

O professor Zadeh também formulou em 1973 o princípio da incompatibilidade, afirmando que: "A medida que a complexidade de um sistema aumenta, nossa habilidade para fazer afirmações precisas e que sejam significativas acerca deste sistema diminui até que um limiar é atingido, além do qual a precisão e a significância (ou relevância) tornam-se quase que características mutuamente exclusivas" (ZADEH, 1979). Além disso, observava-se uma incoerência entre a capacidade criativa do homem e as possibilidades proporcionadas por máquinas binárias.

Assim, os conceitos apresentados por Zadeh eliminaram tal restrição, fornecendo um ferramental matemático para o tratamento adequado de informações de caráter impreciso.

Devido à resistência de cientistas, a lógica nebulosa cresceu no mercado comercial para depois se desenvolver nas universidades (TANSCHEIT, 2005). As primeiras aplicações comerciais situaram-se na área de controle, tanto na automação de processos quanto na supervisão. Desde então, verificou-se uma crescente utilização em diversos campos de conhecimento como classificação, planejamento e previsão de séries. mineração de dados, otimização. Alguns exemplos bem sucedidos são: o controle de velocidade, aceleração e frenagem dos trens do Metrô de Sendai (Japão), recarregadores ultra-rápidos de bateria NiCd da Bosh, detectores de fumaça Cerberus (Suiça), ajustes de imagem de televisores da Sony, autofocus para câmera de vídeo da Canon, otimização em elevadores da Hitachi, dentre outros.

Desta forma, a lógica nebulosa torna possível criar modelagens mais realistas, permitindo máquinas binárias trabalhar de maneira próxima ao pensamento humano, o qual é intrinsecamente "nebuloso".

2.2 VARIÁVEIS E CONJUNTOS

O conceito de pertinência de um elemento a um determinado conjunto fica muito bem definido quando se utiliza a lógica clássica aristotélica, ou seja, através do atributo da bivalência é possível delimitar funções de avaliação para identificar se um elemento pertence ou não a um conjunto. Assim, dado um conjunto A em um universo de discurso X, define-se a função característica $f_A(x) = 1$ quando $x \in A$ e $f_A(x) = 0$ quando $x \notin A$, onde x é um elemento de X.

Porém, existe um descompasso entre o mundo real e tal abordagem bivalente. Como definir exatamente quando se está quente ou se uma pessoa é jovem? Deve-se considerar que no mundo real tudo é uma questão de ponto de vista e definições muito rigorosas podem ocasionar perda de informações. Em uma realidade multivalente as possibilidades de interpretação são estendidas, como gradações entre o verdadeiro e o falso. A lógica nebulosa permite capturar tais "graus de verdade" de uma afirmação, trabalhando com a incerteza e a verdade fenômenos naturais. de maneira sistemática parcial de е rigorosa (SHAW e SIMÕES, 1999). Desta forma, a função característica passa a ser definida como um número real pertencente ao intervalo (0,1), eliminando a restrição dos valores serem apenas **0** ou **1**. A função de pertinência $\mu_A(x)$ indica o grau de inclusão (de compatibilidade) de um elemento x ao conjunto A dentro do universo de discurso X, com:

 \checkmark $\mu_A(x) = 1$, quando x é completamente compatível com A;

 $\checkmark \mu_A(x) = 0$, quando x é completamente incompatível com A;

✓ 0 < $\mu_A(x)$ < 1, quando x é parcialmente compatível com A, com grau $\mu_A(x)$.

A figura 2.1 mostra um exemplo comparativo entre a abordagem booleana e a nebulosa para o conjunto IDOSO.



Figura 2.1: Conjunto IDOSO na abordagem booleana (a) e nebulosa (b)

Na abordagem booleana, na figura 2.1(a), as pessoas que atingem 65 anos passam a pertencer ao conjunto específico IDOSO. Porém, tal delimitação fica muito restrita, pois os valores próximos deste limiar mudam abruptamente de NÃO IDOSO para IDOSO.

Na abordagem nebulosa, na figura 2.1(b), os elementos que possuem valores superiores a 55 anos passam a fazer parte do conjunto IDOSO, entretanto com diferentes níveis de inclusão. Observa-se que o grau de pertinência começa nulo no ponto 55 e atinge seu valor máximo em 65, definindo nesta faixa o fator incerteza, que permite identificar os graus de representatividade que um valor pode assumir dentro de um mesmo conjunto.

Para construir uma variável nebulosa é necessário subdividir o universo de discurso em diversos conjuntos nebulosos, os quais deverão possuir rótulos de identificação. A figura 2.2 mostra uma representação gráfica da variável lingüística **idade** com suas respectivas subdivisões.



Figura 2.2: Variável nebulosa idade

Para compor tal variável configurou-se um domínio no intervalo entre 0 e 100 anos, subdividindo-o em 3 partes: JOVEM (entre 0 e 35 anos), MEIA-IDADE (entre 30 e 60) e IDOSO (entre 50 e100). Observe que a forma e o posicionamento de tais conjuntos dependerão da preferência e experiência de cada projetista, o qual considera a complexidade do modelo em questão e os custos computacionais exigidos. Na prática, funções simples como triangulares, trapezoidais e Gaussianas são as mais comumente utilizadas, pois simplificam a computação e produzem bons resultados. É importante ressaltar que, muitas vezes, são necessários alguns ajustes para a modelagem final da variável nebulosa.

Na figura 2.2, uma pessoa com 57 anos pertence tanto ao conjunto MEIA-IDADE quanto ao conjunto IDOSO, onde se observam seus respectivos níveis de inclusão, com **0.2** no primeiro e **0.7** no segundo. Desta forma, um mesmo elemento x pode assumir valores simultâneos em diferentes conjuntos, permitindo configurar os graus de compatibilidade em valores $\mu_A(x)$. A flexibilidade é uma característica das variáveis nebulosas, pois os rótulos não necessariamente serão exclusivos.

Este tipo de definição possibilita também a identificação dos elementos que são mais representativos da idéia geral dos conjuntos, ou seja, quanto mais próximo o

valor $\mu_A(x)$ estiver de **1.0**, maior o grau de representatividade do termo lingüístico aplicado.

As operações com conjuntos nebulosos são definidas com base nas funções de pertinência utilizadas. Segundo Zadeh (1965), a função de inclusão para a união dos conjuntos A e B (U = A \cup B) é definida como $\mu_U(x) = \max(\mu_A(x), \mu_B(x)), x \in X$, sendo X o universo de discurso da variável. A interseção entre estes mesmos conjuntos A e B (I = A \cap B) foi definida como $\mu_I(x) = \min(\mu_A(x), \mu_B(x)), x \in X$. Por fim, a função de inclusão do complemento para o conjunto A é $\mu_C(x) = 1 - \mu_A(x), x \in X$. Tais configurações são equivalentes às operações básicas clássicas, apenas estendendo os valores do conjunto {0,1} para o intervalo (0,1). Outras definições para os operadores de união e interseção foram apresentadas por diferentes pesquisadores (DUBOIS, PRADE e YAGER, 1993), (COX 1994).

Entretanto, a "Lei da Não Contradição" (A $\cap \neg A = \emptyset$) e a "Lei da Exclusão" (A $\cup \neg A = X$) não estão contempladas na abordagem nebulosa. A lógica clássica interpretaria como uma contradição um elemento que pudesse estar, ao mesmo tempo, em um determinado conjunto e no seu complemento. Por exemplo, uma pessoa não poderia ser ALTA e NÃO ALTA simultaneamente. Porém, a figura 2.3 abaixo mostra a variável nebulosa **altura**, identificando que a interseção entre os conjuntos das pessoas ALTAS e das NÃO ALTAS não está vazio. Ou seja, uma pessoa com 1.80 metros é *alta* e *não alta*, apresentando um grau de inclusão 0.5 para os dois conjuntos nebulosos (CRUZ, DEMASI e LIMA, 2003).



Figura 2.3: Interseção entre os conjuntos NÃO ALTO e ALTO da variável nebulosa altura

De forma análoga, tem-se pela figura 2.4 que a união entre os conjuntos não preenche todo o universo de discurso da variável **altura**, pois foi configurado o fator "incerteza" entre os valores **1.7** e **1.9**.



Figura 2.4: União entre os conjuntos NÃO ALTO e ALTO da variável nebulosa altura

2.3 IMPLICAÇÕES LÓGICAS E REGRAS DE INFERÊNCIA

Os seres humanos utilizam comumente implicações lógicas para formular conexões entre causas e efeitos, ou seja, consciente ou inconscientemente são criadas regras de inferência com o seguinte formato:

Se (antecedentes) então (conseqüentes)

Estas regras podem combinar diversos antecedentes (premissas) e conseqüentes (conclusões) através da utilização de operadores lógicos "e" e "ou". A estrutura de uma proposição condicional nebulosa é similar à da lógica booleana, onde sinais como <, > e = podem ser substituídos por termos lingüísticos como BAIXO, GRANDE e EQUIVALENTE. Entretanto, a interpretação de uma regra nebulosa é diferente de uma regra clássica.

Na lógica aristotélica, uma conclusão somente será inferida se a afirmação de seus antecedentes for verdadeira. Por exemplo, uma regra formada apenas por conectivos "e" necessita de que todas as premissas sejam positivas para verificar sua conclusão. Enquanto que em outra que utiliza apenas o "ou", pelo menos um dos antecedentes deverá ser verdadeiro.

Na lógica nebulosa, as premissas podem assumir graus de verdade no intervalo entre o totalmente falso e o inteiramente verdadeiro. Desta forma, as avaliações dos antecedentes podem ser manipuladas através de operações também delimitadas por Zadeh, assumindo que as definições de *max* e *min* constituem representações para os operadores "ou" (união) e "e" (interseção), respectivamente (ZADEH, 1965). Por exemplo, a regra "Se (a é A) e (b é B ou c é C) então (d é D)", com graus de inclusão $\mu_A(a) = 0.7$; $\mu_B(b) = 0.3$ e $\mu_C(c) = 0.5$, geraria a seguinte avaliação $\mu_A(a) \cap (\mu_B(b) \cup \mu_C(c)) = min(0.7, max(0.3, 0.5)) = min(0.7, 0.5) = 0.5$. Este resultado reflete o grau de verdade da resposta, ou seja, o grau de pertinência que o conseqüente possui em relação ao conjunto D. É importante ressaltar que tal resposta também pode ser considerada como uma contribuição parcial ao processo de inferência, sendo possível ocorrerem situações em que mais de uma regra estivesse ativa.

Deve-se enfatizar que muitos sistemas baseados em regras resultam de entrevistas com operadores humanos, os quais possuem uma boa experiência no contexto da aplicação. Assim, a liberdade de um projetista em alterar a estrutura de inferência estará relacionada à compreensão das descrições fornecidas pelo especialista. Por outro lado, muito menos tempo de ajuste é esperado neste tipo de projeto, já que a experiência do operador, que contém o melhor desempenho, estará embutida nas regras.

A combinação de todas as inferências nebulosas constitui a base de regras de um controlador, a qual permitirá realizar todas as ações de controle desejadas sob as condições especificadas. As regras de inferência da figura 2.5 mostram um exemplo de ações de controle delimitadas para um ar condicionado, onde foram combinadas as variáveis de entrada **temperatura** e **umidade** para gerar o nível de potência do ventilador.

Se (temperatura = baixa) e (umidade = baixa) então (potência = muito baixa) Se (temperatura = baixa) e (umidade = alta) então (potência = baixa) Se (temperatura = média) e (umidade = baixa) então (potência = média) Se (temperatura = média) e (umidade = alta) então (potência = média-alta) Se (temperatura = alta) e (umidade = baixa) então (potência = alta) Se (temperatura = alta) e (umidade = alta) então (potência = muito alta) Se (temperatura = alta) e (umidade = alta) então (potência = muito alta) Figura 2.5: Exemplo de regras de inferência

É importante que existam tantas regras quantas forem necessárias para mapear totalmente as combinações das variáveis, isto é, que a base seja completa, garantindo que ao menos uma regra seja disparada para qualquer entrada. Também são essenciais a consistência, onde procura-se evitar a possibilidade de contradições, e a interação entre regras, de modo a contornar situações de ciclo (SANDRI e CORREA, 1999).

Como visto anteriormente, as regras podem possuir múltiplas entradas e saídas. Porém, não se aceitam conectivos "ou" nas conclusões (SANDRI, 1997).

2.4 SISTEMAS NEBULOSOS

A ciência da computação é baseada na bivalência, onde os procedimentos computacionais regulares não possuem a capacidade de interpretar termos lingüísticos comuns na comunicação humana. A lógica nebulosa permite preencher esta lacuna, traduzindo os graus de verdade das afirmações, de maneira que as máquinas possam processar tais informações.

Na literatura, existem diversos tipos de controladores nebulosos. Os modelos clássicos mais conhecidos são o Mamdani e o Larsen. Em uma outra vertente, para

os que utilizam técnicas de interpolação, tem-se o Takagi-Sugeno e o Tsukamoto. Porém, todos eles possuem uma mesma estrutura básica de representação, mostrada abaixo na figura 2.6, a qual é uma adaptação da descrição realizada por (LEE, 1990). Outras variações são propostas de acordo com o objetivo do projeto em questão, porém este é um modelo geral o suficiente para a identificação dos módulos envolvidos, fornecendo uma idéia do fluxo da informação.



Figura 2.6: Estrutura básica de um controlador nebuloso

2.4.1 Interface de fuzzyficação

A *interface de fuzzyficação* avalia todos os valores de entrada em conjuntos que compõem as respectivas variáveis nebulosas. Este processo transforma um valor numérico (*"crip"*) em conjuntos nebulosos (*"fuzzy"*), para que estes possam tornar-se instâncias das variáveis lingüísticas.

2.4.2 Base de conhecimentos

A *base de conhecimentos* armazena todas as informações sobre os conjuntos e operadores do modelo, caracterizando os universos de discurso de cada parâmetro e as respectivas funções de pertinência dos termos nebulosos. Além disso, envolve também a base de regras que permite configurar a estratégia de controle e suas metas.

2.4.3 Procedimento de inferência

O *procedimento de inferência* manipula as regras juntamente com os dados de entrada, o qual gera ações de controle baseado no estado atual do sistema, aplicando o operador de implicação. Este processo, segundo Sandri e Correa (1999), consiste em:

- Verificar o grau de compatibilidade entre os fatos e as cláusulas nas premissas das regras;
- Determinar o grau de compatibilidade global da premissa de cada regra;
- Determinar o valor da conclusão, em função do grau de compatibilidade da regra com os dados e a ação de controle constante na conclusão (precisa ou não);
- Agregar os valores obtidos na conclusão das diversas regras, obtendo uma ação de controle global.

Nos modelos nebulosos clássicos, a conclusão de cada regra especifica um conjunto nebuloso. Assim, para cada regra de inferência é necessário aplicar uma técnica de agregação dos conjuntos antecedentes, com o objetivo de gerar um conjunto conseqüente. No modelo Mamdani, esta associação é realizada aplicando o operador interseção (mínimo), onde o conseqüente sofre um corte horizontal no menor grau de inclusão ativado pelas regras (MAMDANI, 1974). Por exemplo, na figura 2.7 tem-se as entradas x_a^* e x_b^* ativando dois conjuntos nebulosos simultaneamente (A₁ e A₂; B₁ e B₂). As associações das premissas A₁ com B₁ e

 A_2 com B_2 geram, respectivamente, as inferências C_1 e C_2 . Assim, estes conseqüentes obtidos (C_1 e C_2) sofrem cortes no valor mínimo do grau de inclusão presente nas premissas (A_1 com B_1 e A_2 com B_2).

No modelo de Larsen, o operador utilizado é o produto, levando a um achatamento proporcional no seu conseqüente (LARSEN, 1980). A figura 2.8 mostra o mesmo exemplo anterior, utilizando esta abordagem. Neste caso, as inferências obtidas em C₁ e C₂ representam uma diminuição proporcional dos conjuntos que foram inferidos pelas associações das premissas A₁ com B₁ e A₂ com B₂.

Assim, tanto no Mamdani quanto no Larsen, quando N conjuntos antecedentes estiverem ativos serão gerados N conjuntos conseqüentes. Desta forma, todas as inferências geradas ($C_1 \ e \ C_2$, do exemplo) são combinadas criando um único conjunto final de saída (C'), o qual é dado pela aplicação do operador união (máximo). As figuras 2.7 e 2.8 mostram este processo para o modelo do Mamdani e do Larsen, respectivamente.



Figura 2.7: Modelo clássico de Mamdani


Figura 2.8: Modelo clássico de Larsen

Nos modelos nebulosos de interpolação, cada conclusão é dada através de uma função monotônica, usualmente diferente para cada regra. As funções são geradas a partir de um conjunto de dados de treinamento e validação, ajustando os pesos (constantes) até a obtenção de resultados satisfatórios, seguindo o mesmo raciocínio aplicado às redes neurais. No esquema de Takagi-Sugeno, esta função é uma combinação linear das entradas, tendo como parâmetro um conjunto de constantes (TAKAGI e SUGENO, 1985). Na figura 2.9, tem-se a mesma situação mostrada anteriormente aplicando esta abordagem. O valor mínimo do grau de inclusão das premissas também é utilizado, porém ao invés de gerar um conjunto conseqüente nebuloso utiliza-se a função monotônica delimitada para cada regra ativada. Por exemplo, A₁ com B₁ acionou a regra y_1 ' = f_1 ($x_a^* \cdot x_b^*$) = $d_0 + d_1 \cdot x_a^* + d_2 \cdot x_b^*$. Desta forma, quando os valores de $x_a^* e x_b^*$ são substituídos, obtém-se um valor preciso y_1 ', visto que d_0 , $d_1 e d_2$ são constantes previamente definidas. Este mesmo processo ocorre para a regra de A₂ com B₂.

No modelo de Tsukamoto, a função é geralmente não-linear, possuindo como domínio os possíveis graus de compatibilidade entre cada premissa e as entradas

(TSUKAMOTO, 1979). A figura 2.10 mostra o exemplo anterior aplicando este esquema. A referência do grau de pertinência mínimo continua sendo aplicada, mas o conseqüente passa a ser um valor pertencente à função previamente delimitada. Neste exemplo, os valores precisos y_1 ' e y_2 ' foram inferidos a partir da combinação de A_1 com B_1 e A_2 com B_2 .

Nos modelos nebulosos de interpolação, obtém-se para cada regra um único valor para a variável de controle. Finalmente, uma ação de controle global é obtida fazendo-se uma média ponderada dos valores individuais obtidos (y₁' e y₂', do exemplo), onde cada peso é o próprio grau de compatibilidade entre a premissa da regra e as entradas (DRIANKOV, HELLENDOORN, e REINFRANK, 1993). As figuras 2.9 e 2.10 representam as respectivas interpretações gráficas para o método de interpolação de Takagi-Sugeno e Tsukamoto.



Figura 2.9: Modelo de interpolação de Takagi-Sugeno



Figura 2.10: Modelo de interpolação de Tsukamoto

2.4.4 Interface de defuzzificação

Nos controladores nebulosos do tipo clássico, a *interface de defuzzificação* é utilizada para gerar um valor de saída preciso ("*crisp*") a partir do conjunto conseqüente gerado na etapa anterior. Este processo identifica o domínio das variáveis de saída em um correspondente universo de discurso e com a diretiva de controle nebulosa inferida evolui-se uma ação de controle precisa. Os métodos de defuzzificação mais comumente utilizados são:

- Primeiro máximo: o valor de saída é o ponto em que o grau de pertinência da distribuição de ação de controle atinge o primeiro valor máximo;
- Média dos máximos: encontra-se o ponto médio entre os valores que possuem o maior grau de compatibilidade inferido pelas regras;
- Centro de gravidade: é o valor no universo que divide a área inferida pelas regras em duas partes iguais.

A escolha do método está relacionada diretamente com as características do processo controlado e o comportamento de controle necessário. Por exemplo, o método da "média" e do "primeiro máximo" podem produzir ações inadequadas ao modo de operação, pois mudanças bruscas na inferência poderiam gerar danos a equipamentos, considerando os possíveis solavancos produzidos pelas saídas do sistema.

Há ainda outros métodos de defuzzificação com diferentes fatores de velocidade e eficiência (PATEL, 2005), (NURCAHYO *et al*, 2003), (GUERRA, 1998), (SANDRI, 1997), (GOMIDE, GUDWIN e TANSCHEIT, 1995), (DRIANKOV, HELLENDOORN, e REINFRANK, 1993). Porém, as características devem ser avaliadas em conjunto com os requisitos de cada projeto.

É importante ressaltar que os controladores nebulosos do tipo interpolação não necessitam de uma interface de defuzzificação, pois estes obtêm diretamente os valores precisos para os processos controlados.

2.5 EXEMPLO

De modo a verificar o funcionamento de um sistema de inferência nebuloso, considere o problema clássico de estacionar um caminhão (KOSKO, 1992).

Inicialmente, o veículo está em uma posição (x, y) qualquer no plano, com um ângulo ϕ em relação à linha horizontal. O par ordenado (x, y) permite especificar a posição central da traseira do caminhão. O objetivo do simulador é fazer com que o mesmo chegue à posição final de estacionamento (x_f, y_f) com um ângulo ϕ = 90°, considerando que as manobras são feitas somente em marcha ré. A figura 2.11 mostra o caminhão e a área de estacionamento.



Figura 2.11: Representação do caminhão e a área de estacionamento

Desta forma, a cada etapa da simulação, o sistema nebuloso deve produzir um ângulo de rotação θ , permitindo atualizar o posicionamento do volante para o desenvolvimento de um percurso em direção à meta traçada. Este ângulo θ inicialmente é configurado como nulo, simulando que no primeiro momento as rodas estão paralelas à lateral do veículo. Assume-se também que existe um espaço suficiente para o caminhão fazer as suas manobras e que a velocidade "r" exercida será sempre constante. Para descrever o movimento de uma posição (x, y) para (x', y') têm-se as seguintes equações:

$$\phi' = \phi + \theta$$

$$x' = x + r(\cos \phi')$$

$$y' = y + r(\sin \phi')$$

As definições realizadas geram o seguinte comportamento: a cada iteração são fornecidas ao módulo nebuloso as entradas x e ϕ atuais para que seja obtida a saída θ . Vale ressaltar que, no modelo descrito por Kosko (1992), o parâmetro y não é utilizado como referência para as tomadas de decisão nebulosas.

Após a inferência, atualiza-se a posição das rodas do caminhão, que significa incrementar o ângulo atual ϕ com o resultado θ fornecido pelo controlador. Além

disso, o posicionamento geral do veículo é modificado, utilizando o parâmetro de deslocamento fixo "r", permitindo, desta forma, desenvolver movimentações em marcha ré a cada passo da simulação.

Os universos de discurso das variáveis foram definidos nos intervalos descritos abaixo, onde os valores positivos dos ângulos representam rotações de direção no sentido horário e negativos, no anti-horário.

$$0 \le x \le 100$$
$$-90^{\circ} \le \phi \le +270^{\circ}$$
$$-30^{\circ} \le \theta \le +30^{\circ}$$

Neste exemplo, as três variáveis nebulosas configuradas pelo sistema foram subdividas nos seguintes conjuntos lingüísticos:

- Posição x: LE (esquerda), LC (esquerda central), CE (centro), RC (direita central) e RI (direita).
- Ângulo θ: NB (muito negativo), NM (meio negativo), NS (pouco negativo), ZE (zero), PS (pouco positivo), PM (meio positivo) e PB (muito positivo).

A base de regras, que constitui a estratégia de estacionamento do caminhão, está representada por uma matriz na tabela 2.1. Esta estrutura combina todos os conjuntos das duas variáveis de entrada (x e ϕ) para formar em cada célula uma regra do sistema nebuloso, identificando, assim, a ação delimitada pelo conjunto de saída θ .

¢ X	LE	LC	CE	RC	RI
RB	PS	PM	PM	PB	PB
RU	NS	OS	PM	PB	PB
RV	NM	NS	OS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NM	NS	NS	PM
LU	NB	NB	NM	NS	OS
LB	NB	NB	NM	NM	NS

Tabela 2.1: Base de regras do controlador do caminhão

A leitura das regras a partir da matriz é exemplificada para a célula sombreada ZE:

Se (x é CE) e (ϕ é VE) então (θ é ZE)

Neste problema, as interpretações do modelo clássico do Mamdani foram aplicadas, com a utilização do operador interseção (*min*) para combinar os antecedentes de cada regra (dois, neste caso) e o operador união (*max*) para gerar o conjunto final de saída.

Os conjuntos nebulosos correspondentes a cada variável estão representados por suas funções de pertinência na figura 2.12.



Figura 2.12: Representação dos conjuntos nebulosos do caminhão

Como um exemplo de iteração, é mostrada uma situação hipotética descrita por (TANSCHEIT, 2005), onde seriam configurados os valores de entrada x = 65 e ϕ = 113°. Observe que cada parâmetro ativa dois conjuntos nebulosos com diferentes graus de pertinência.

- Para a variável x: RI com 0.2 e RC com 0.7.
- Para a variável ϕ : LV com 0.9 e VE com 0.5.

Desta forma, quatro regras devem ser analisadas, considerando a combinação dos conjuntos anteriores. A tabela 2.2 mostra as células ativadas na base de regras.

¢ X	LE	LC	CE	RC	RI
RB	PS	PM	PM	PB	PB
RU	NS	OS	PM	PB	PB
RV	NM	NS	OS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NM	NS	NS	PM
LU	NB	NB	NM	NS	OS
LB	NB	NB	NM	NM	NS

Tabela 2.2: Regras ativas no exemplo de iteração

De forma explícita, obter-se-ia:

- Se (x é RC) e (φ é VE) então (θ é PM) ou
- Se (x é RC) e (ϕ é LV) então (θ é PS) ou
- Se (x é RI) e (φ é VE) então (θ é PM) ou
- Se (x é RI) e (φ é VE) então (θ é PM).

Os antecedentes devem ser avaliados pelo operador interseção (*min*), seguindo as especificações abaixo:

$$\mu_{\mathsf{B}^{\star}}(\theta) = (\mu_{\mathsf{A1}}(\mathsf{x}) \wedge \mu_{\mathsf{A2}}(\phi) \wedge \mu_{\mathsf{B}}(\theta))$$

onde, A1 simboliza os valores da variável x, A2 os de ϕ e B os de θ . Observe que B* não corresponderá a um rótulo pré-especificado, pois ele será uma combinação de conjuntos. Para cada uma das regras ativadas, tem-se, portanto:

•
$$\mu_{PM^*}(\theta) = (\mu_{RC}(x) \wedge \mu_{VE}(\phi) \wedge \mu_{PM}(\theta)) = (0.7 \wedge 0.5) \wedge \mu_{PM}(\theta) = 0.5 \wedge \mu_{PM}(\theta)$$

•
$$\mu_{PS^*}(\theta) = (\mu_{RC}(x) \wedge \mu_{LV}(\phi) \wedge \mu_{PS}(\theta)) = (0.7 \wedge 0.9) \wedge \mu_{PS}(\theta) = 0.7 \wedge \mu_{PS}(\theta)$$

•
$$\mu_{PM^*}(\theta) = (\mu_{RI}(x) \wedge \mu_{VE}(\phi) \wedge \mu_{PM}(\theta)) = (0.2 \wedge 0.5) \wedge \mu_{PM}(\theta) = 0.2 \wedge \mu_{PM}(\theta)$$

•
$$\mu_{PM^*}(\theta) = (\mu_{RI}(x) \wedge \mu_{LV}(\phi) \wedge \mu_{PM}(\theta)) = (0.2 \wedge 0.9) \wedge \mu_{PM}(\theta) = 0.2 \wedge \mu_{PM}(\theta)$$

Subseqüentemente, é realizada a união (*max*) das regras acima. Verifica-se que o conjunto de saída PM ocorre em três situações. Assim, após a aplicação do operador restarão os seguintes conjuntos nebulosos de saída:

•
$$\mu_{PM^*}(\theta) = 0.5 \wedge \mu_{PM}(\theta)$$

•
$$\mu_{PS^*}(\theta) = 0.7 \wedge \mu_{PS}(\theta)$$

Esta composição também pode ser observada graficamente na figura 2.13.



Figura 2.13: Composição dos conjuntos nebulosos e o valor defuzzyficado pelo método do centro de gravidade

Segue-se aqui o mesmo raciocínio de interpretação gráfica apresentado para a figura 2.7 deste Capítulo. Neste exemplo, quatro conjuntos conseqüentes foram delimitados, levando em consideração que quatro regras estiveram ativas. Para cada inferência, utilizava-se o operador interseção (*min*) para pegar o menor valor do grau de inclusão das premissas, ou seja, na regra "Se (x é RC) e (ϕ é VE) então (θ é PM)", com $\mu_{RC}(x) = 0.7$ e $\mu_{VE}(\phi) = 0.5$, tem-se como resultado o valor $\mu_{PM}(\theta) = 0.5$. Subseqüentemente, os quatro conjuntos gerados necessitavam ser agregados para gerar um único resultado nebuloso, como mostra a forma em destaque na figura 2.13. Para isso, aplicava-se o operador união (*max*) de modo a

agregar os conjuntos subseqüentes. Neste caso, três regras acionaram um mesmo rótulo PM com diferentes graus de pertinência. Assim, apenas o maior valor foi considerado após a aplicação do operador.

Para gerar a inferência final era necessário transformar a saída nebulosa em um resultado preciso, ou seja, a área apresentada na figura 2.13 deveria ser transformada em um valor numérico final. O método de defuzzyficação do "centro de gravidade" permitiu gerar este resultado, considerando que o valor obtido **9.7** é o ponto exato que divide a área gerada em duas partes.

Assim, a cada passo da simulação, os novos dados atualizados do caminhão eram utilizados novamente como parâmetros de entrada do sistema, de forma a gerar uma próxima ação de controle. Portanto, nestas sucessivas tomadas de decisão, desenvolvia-se uma trajetória para estacionar o veículo adequadamente no espaço previsto como meta.

2.6 ESTABILIDADE DE SISTEMAS DE CONTROLE NEBULOSOS

Uma questão fundamental para os desenvolvedores de sistemas de controle referese ao estudo de estabilidade, devido à aplicação prática exercida por tais controladores. Existem técnicas bem fundamentadas para o estudo específico de sistemas de controle em malha fechada, determinando limites de estabilidade para controladores clássicos, digitais e multivariáveis (SHAW e SIMÕES, 1999).

Porém, tais diretivas não abrangem os modelos que aplicam recursos da lógica nebulosa. Desta forma, diversos estudos vêm sendo desenvolvidos por pesquisadores para analisar a estabilidade destes sistemas (PRECUP e PREITL,

2005), (ARACIL e GORDILLO, 2004), (PRECUP, DOBOLI e PREITL, 2000), (ROSENWASSER e YUSUPOV, 2000), (SUGENO, 1999).

Em uma primeira vertente, tem-se que um sistema nebuloso é, em termos matemáticos, um mapeamento de entradas Rⁿ em um espaço de saída Q^m com as seguintes propriedades (ALTROCK, 1995):

- determinístico: as mesmas condições de entrada no modelo sempre resultam nas mesmas condições de saída;
- invariante no tempo: a função que descreve a relação entre entrada-saída não se altera com o tempo;
- não-linear: as variáveis de saída não são combinações lineares das variáveis de entrada.

Os modelos que possuem tais características são classificados pela teoria de controle como controladores não-lineares multivariáveis (ou multibanda). Desta forma, todos os métodos que são aplicáveis a estes tipos de sistema são passíveis de utilização em modelos nebulosos (ALTROCK, 1995).

Entretanto, devido à complexidade de certos processos, à não linearidade e/ou à invariância no tempo, fica quase impossível de se determinar um modelo analítico oportuno (ZHANG, LI e ZHANG, 2001). Além disso, os controladores nebulosos, principalmente os que são somente baseados em regras, não podem ser expressos em termos matemáticos, não permitindo se chegar a uma definição analítica (SHAW e SIMÕES, 1999). É importante enfatizar que mesmo em estudos de estabilidade por métodos analíticos de controladores convencionais, com um grau similar de complexidade do contexto, tem-se uma difícil obtenção de soluções. (ARACIL e GORDILLO, 2004).

Em uma segunda vertente, apresentada por Kosko e Kong (1992), analisa-se a estabilidade destes controladores através da configuração de situações não previstas na modelagem inicial dos sistemas, ou seja, a base de conhecimentos pode sofrer dois tipos de modificação, havendo substituições de regras ou até mesmo a remoção aleatória das mesmas. Esta delimitação caracteriza a possibilidade de um projetista não conhecer todas as informações de um determinado contexto simulado; ou ainda, momentos de falha no funcionamento destes controladores.

No estudo realizado pelos autores (KOSKO e KONG, 1992), foi observado que as perturbações realizadas ocasionavam diferentes comportamentos nas tomadas de decisão. Desta forma, os sistemas eram analisados utilizando como referência as "taxas de erro" provenientes das simulações realizadas. No modelo do caminhão, dois parâmetros específicos foram analisados: (1) o "erro de estacionamento", que era a distância euclidiana entre a posição final alcançada e a meta final desejada e (2) o "erro de trajetória", que era o tamanho do percurso desenvolvido dividido pela distância entre a posição inicial e a final esperada.

Esta abordagem permite avaliar a estabilidade dos controladores em situações práticas, pois é possível mensurar o desempenho dos mesmos frente a situações inesperadas ou de conflito.

Em geral, os controladores nebulosos encontram maior utilidade em sistemas não-lineares, sendo capazes de superar perturbações e plantas com níveis de ruído. Além disso, mesmo em sistemas onde a incerteza se faz presente de maneira intrínseca, estes agregam uma robustez característica. No entanto, provar determinadas propriedades que garantam a robustez é uma tarefa difícil neste tipo de abordagem (SANDRI e CORREA, 1999).

2.7 PROJETO DE CONTROLADORES NEBULOSOS

Segundo Guerra (1998), de uma maneira geral, as tarefas para a construção de um controlador nebuloso podem ser descritas brevemente nas seguintes etapas:

- Definição do modelo e das características operacionais: para estabelecer as particularidades da arquitetura do sistema (como sensores e atuadores) e delimitar as propriedades operacionais, como o tipo do controlador nebuloso, operadores a serem utilizados, métodos de defuzzificação etc;
- Definição dos termos nebulosos de cada variável: para garantir suavidade e estabilidade deve-se permitir que haja sobreposição parcial entre os conjuntos vizinhos;
- Definição do comportamento de controle: para a descrição das regras que associam as variáveis de entrada às propriedades de saída do modelo.

No projeto destes controladores é necessária, portanto, a definição de alguns parâmetros, obtidos a partir da experiência do projetista ou através de experimentos. Alguns destes parâmetros são estruturais e outros de sintonização:

- 2. Parâmetros estruturais:
 - Número de variáveis de entrada e saída;
 - Variáveis lingüísticas;
 - Funções de pertinência parametrizadas;
 - Intervalos de discretização e normalização;
 - Estrutura da base de regras;
 - Conjunto básico de regras;
 - Recursos de operação sobre os dados de entrada.
- 3. Parâmetros de sintonização:
 - Universo de discurso das variáveis;
 - Parâmetros das funções de pertinência (p.ex. núcleo e suporte).

As propriedades da base de regras como a completude, consistência, interação e robustez também devem ser testadas. A robustez do modelo está relacionada com a sensibilidade do controle frente a ruídos ou algum comportamento incomum nãomodelado. Uma possibilidade de análise seria a introdução de um ruído aleatório nas entradas para observar as mudanças nos valores das variáveis de saída.

A sintonização é uma tarefa complexa devido à flexibilidade que decorre da existência de muitos parâmetros, exigindo esforço do projetista na obtenção do melhor desempenho do controlador. Alguns dos parâmetros podem ser alterados por mecanismos automáticos de adaptação e aprendizado. Entretanto, os procedimentos de treinamento e a sintonia são normalmente realizados pelo projetista.

A sintonização pode ser desenvolvida através de métodos de busca, que são atividades típicas em Inteligência Artificial. O processo pode ser ocorrer da seguinte forma:

Desenvolve-se um controlador simples com:

- Conjunto de variáveis mais relevantes;
- Baixo número de variáveis lingüísticas.
- Incrementa-se o conhecimento conforme a experiência resultante do processo:
 - Buscando-se novas variáveis lingüísticas ou físicas para contornar as dificuldades;
 - Ajustando-se as funções de pertinência e os parâmetros do controlador;
 - Adicionando-se regras ou modificando a estrutura de controle.
- Valida-se a coerência do conhecimento incorporado com novas condições de operação para o sistema.

Essas tarefas necessitam de plataformas sofisticadas, com interface poderosa e que permitam uma rápida inferência. Isto é proporcionado pelos pacotes integrados, dedicados especificamente à análise de modelos nebulosos (GOMIDE, GUDWIN e TANSCHEIT, 1995).

2.7 CONSIDERAÇÕES PARCIAIS

O objetivo deste Capítulo foi fazer uma descrição dos principais conceitos relacionados à lógica nebulosa, fundamentando o processo para o desenvolvimento de sistemas de controle nebulosos.

A criação destes sistemas aplica basicamente as teorias formuladas pelo professor Zadeh. A modelagem é simples e rápida, pois as características e inferências a serem configuradas seguem o mesmo padrão das afirmações utilizadas pelos seres humanos em seu dia-a-dia.

Nos sub-tópicos anteriores, foram mostradas as formas para gerar as variáveis, os conjuntos e as regras, identificando também como ocorre o fluxo de informações dentro destes modelos.

Foram apresentadas, também, as duas vertentes para a análise de estabilidade destes controladores. Porém, nesta dissertação utilizar-se-ão como referência os procedimentos aplicados por Kosko e Kong (1992), por ser a alternativa mais adaptável ao modelo desenvolvido.

33

3. DESCRIÇÃO DO SISTEMA NEBULOSO DE CONTROLE DO ROBÔ

O objetivo deste Capítulo é fazer a descrição e a explanação do sistema de controle nebuloso que foi desenvolvido para este trabalho. Posteriormente, este modelo servirá de referência para a realização do estudo de estabilidade e robustez.

3.1 VISÃO GERAL

O controlador nebuloso desenvolvido permitiu guiar um robô simulado simples da parede esquerda para a direita através de um mundo virtual pré-definido. Durante o seu percurso, colisões com obstáculos e com as outras paredes foram evitadas. A figura 3.1 mostra o mundo virtual com obstáculos e o robô simulado.



Figura 3.1: Diagrama do mundo virtual com obstáculos e o robô simulado

O raio do robô é igual a 6 unidades e o de cada obstáculo igual a 3. O mundo virtual corresponde ao plano [0,200] x [0,100].

No início de cada simulação, o robô era colocado junto à parede esquerda. O usuário do sistema configurava o seu posicionamento no eixo y (no intervalo entre 6 e 94) e o seu ângulo ϕ com a horizontal (no intervalo entre -90° e 90°). A quantidade de obstáculos também era configurável, e os mesmos eram fixados em posicionamentos randômicos pelo sistema. Assim, o controlador gerava sucessivos deslocamentos até que a posição $x_f = 200$ (parede direita) fosse atingida, independentemente da coordenada y_f .

3.2 SENSORES DO ROBÔ

Os três parâmetros x_r , $y_r \in \phi$ determinavam a posição exata do robô. Ele possuía sensores que detectavam e mediam as distâncias aos obstáculos que estavam em seu caminho. Se mais de um obstáculo fosse detectado, somente o mais próximo era considerado. A figura 3.2 mostra como foram detectadas as possíveis colisões e a medida final gerada pelo sensor (*dist*_o).



Figura 3.2: Região de colisão do robô simulado

Para elucidar o tratamento realizado na detecção dos obstáculos e paredes, são descritas a seguir as estratégias que foram utilizadas no desenvolvimento do sistema.

3.2.1 Detecção dos obstáculos randômicos

Após a configuração da quantidade de obstáculos que iriam fazer parte do mundo virtual, os mesmos eram fixados em posicionamentos randômicos para que o robô pudesse realizar o percurso em cada simulação.

Para que o controlador pudesse detectar estes obstáculos, foram definidas as seguintes estratégias:

A) Cálculo da distância de cada centro dos obstáculos existentes à reta direção do robô (perpendicularmente). Se o valor desta distância for menor que a soma do raio do robô com a do obstáculo (d ≤ 9), é porque o mesmo está na zona de risco.



Figura 3.3: Detecção dos obstáculos randômicos

Inicialmente, foi necessário definir a equação da reta direção do controlador do robô. A seguir, é mostrada a estrutura padrão para estas equações:

$$y = mx + n$$
, onde $\begin{cases} m \rightarrow \text{coeficiente angular} \\ n \rightarrow \text{coeficiente linear} \end{cases}$

O coeficiente angular (m) e o linear (n) poderiam ser delimitados através do ângulo ϕ e pelas coordenadas do centro do robô (x_r , y_r), que conjuntamente formulariam a expressão da reta em sua forma geral.

A obtenção do coeficiente angular (m) se baseia na seguinte igualdade da geometria analítica:

Porém, para descobrir o valor do coeficiente linear (n) foi necessário substituir as coordenadas do robô (x_r , y_r) na estrutura padrão da equação da reta:

$$y_r = m. x_r + n$$

 $n = y_r - (tg\phi). x_r$

Desta forma, utilizando os respectivos coeficientes, chega-se na expressão geral da reta do robô:

$$y = m. x + n$$

 $y = (tg\phi). x + (y_r - (tg\phi). x_r)$
 $(tg\phi). x - y + (y_r - (tg\phi). x_r) = 0$

O próximo passo consistiu em calcular a distância de cada centro dos obstáculos à reta direção do robô. Deste modo, a seguinte fórmula foi utilizada:

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \text{ onde } \begin{cases} (x_0, y_0) \text{ são as coordenadas do ponto} \\ ax + by + c = 0 \text{ é a equação geral da reta} \end{cases}$$

Da equação do controlador observa-se que:

$$a = (tg\phi)$$
; $b = -1 e c = (y_r - (tg\phi).x_r)$

Fazendo a substituição destes valores e considerando um obstáculo qualquer de coordenadas (x_o , y_o), chegou-se à expressão:

$$d = \frac{\left| (tg\phi).x_o - y_o + (y_r - (tg\phi).x_r) \right|}{\sqrt{tg^2\phi + 1}}$$

Também foi utilizada a identidade trigonométrica $\sec^2\phi = tg^2\phi + 1$, obtendo:

$$d = \frac{\left| (tg\phi) x_o - y_o + (y_r - (tg\phi) x_r) \right|}{sec\phi}$$

Assim, para cada obstáculo presente no mundo virtual as respectivas distâncias à reta direção foram calculadas, determinando quem estava presente na zona de colisão (d \leq raio do robô + raio do obstáculo).

B) Cálculo das distâncias entre o robô e os obstáculos presentes na zona de colisão.

Para cada obstáculo presente na zona de colisão, foi calculada a distância entre os mesmos e o robô simulado (parâmetro *dcc*). A figura 3.4 mostra um exemplo de como o cálculo foi realizado, utilizando a fórmula do Teorema de Pitágoras.



Figura 3.4: Cálculo da distância entre o robô e um obstáculo específico

Logo,
$$dcc = \sqrt{(x_o - x_r)^2 + (y_o - y_r)^2}$$

3.2.2 Detecção das paredes

O robô também identificava as paredes do mundo virtual que lhe ofereciam risco de colisão, como a esquerda, a inferior e a superior. A parede da direita continuava a ser sua meta. Assim as seguintes estratégias foram definidas:

A) Determinação do ponto x_p, que é o lugar onde a reta direção do robô intercepta as retas delimitadoras das paredes. Verificou-se também se esta interseção era pertencente ao plano que configurava mundo virtual.

A figura 3.5 identifica o ponto x_p para dois casos específicos. O primeiro gráfico (a) mostra uma interseção que não pertence ao mundo virtual. O segundo (b) mostra o caso onde o ponto faz parte do plano delimitado.



Figura 3.5: Interseções da reta direção do robô com a reta delimitadora da parede para a determinação do ponto $x_{\rm p}$

A figura 3.5 (b) mostra, também, como o valor do ponto x_p é determinado quando o ângulo ϕ pertencia ao quadrante de 0° a 90°. O triângulo retângulo formado permitiu utilizar a razão tangente, chegando, assim, ao valor final de x_p :

$$tg\phi = \frac{100 - y_r}{x_p - x_r}$$
$$x_p = x_r + \frac{(100 - y_r)}{tg\phi}$$
$$x_p = x_r + (100 - y_r).cotg\phi$$

Procedeu-se de forma análoga para a determinação dos valores de x_p para os outros quadrantes, considerando os intervalos do ângulo ϕ . Em síntese, os valores foram:

Para ϕ entre 0° e 90°: $x_p = x_r + (100 - y_r).cotg (\phi)$ Para ϕ entre 90° e 180°: $x_p = x_r - (100 - y_r).cotg (180° - \phi)$ Para ϕ entre 0° e -90°: $x_p = x_r + (y_r).cotg (-\phi)$ Para ϕ entre -90° e -180°: $x_p = x_r - (y_r).cotg (180° + \phi)$

Vale ressaltar que após o cálculo do ponto x_p , era verificado se o mesmo pertencia às delimitações do mundo virtual.

B) Cálculo da distância entre o robô e a parede que oferece risco.

Foi usada a razão trigonométrica seno juntamente com o ângulo ϕ para se obter a distância do robô à parede (d_p). Baseado na figura 3.5 (b) obteve-se:

$$sen \phi = \frac{100 - y_r}{d_p}$$
$$d_p = \frac{100 - y_r}{sen \phi}$$
$$d_p = (100 - y_r)^* cossec$$

φ

Se o ângulo ϕ fosse igual a 90°, -90° e 180° ou -180°, a distância era obtida através das próprias coordenadas do robô:

$$\phi = 90^{\circ} \Rightarrow d_{p} = 100 - y_{r}$$

$$\phi = -90^{\circ} \Rightarrow d_{p} = y_{r}$$

$$\phi = 180^{\circ} \Rightarrow d_{p} = x_{r}$$

$$\phi = -180^{\circ} \Rightarrow d_{p} = x_{r}$$

Analogamente ao exemplo da figura 3.5(b) foram calculados os valores de d_p para os outros quadrantes. Em síntese, chegou-se às seguintes configurações:

Para ϕ entre 0° e 90°: $d_p = (100 - y_r)^* \operatorname{cossec} \phi$ Para ϕ entre 90° e 180°: $d_p = (100 - y_r)^* \operatorname{cossec} (180° - \phi)$ Para ϕ entre 0° e -90°: $d_p = (y_r)^* \operatorname{cossec} (-\phi)$ Para ϕ entre -90° e -180°: $d_p = (y_r)^* \operatorname{cossec} (180° + \phi)$ Após o cálculo de d_p, adicionou-se a sua referência ao parâmetro *dcc* do sistema, como sendo mais um obstáculo a ser tratado.

3.2.3 Detecção do obstáculo mais próximo

Após o cálculo das distâncias a todos os obstáculos presentes na zona de colisão e a consideração das paredes, foi definido que apenas o obstáculo mais próximo $(dist_o)$ seria tratado a cada passo de iteração do sistema.

3.3 MOVIMENTAÇÃO DO ROBÔ

Para permitir o deslocamento do robô pelo mundo virtual foram utilizadas equações cinemáticas simples similares às descritas por Kosko (1992), para o problema de dar marcha ré no caminhão. As equações que descrevem o movimento são:

$$\phi' = \phi + \theta$$

$$x'_{r} = x_{r} + \Delta \cos (\phi')$$

$$y'_{r} = y_{r} + \Delta \sin (\phi')$$

O parâmetro de deslocamento Δ (velocidade) foi configurado com um valor fixo de 1,3 unidades. Assim, a cada estágio de simulação o robô tinha o seu ângulo atualizado ($\phi'=\phi+\theta$), deslocando-se da posição (x_r, y_r) para (x_r, y_r) com um passo fixo Δ .

3.4 INTERVALOS DOS PARÂMETROS DO SISTEMA

Os intervalos dos parâmetros do sistema foram configurados com as seguintes delimitações:

 $0 \le x_r \le 200$ $0 \le y_r \le 100$ $0 \le dist_o \le 220$ $-180^\circ \le \phi \le +180^\circ$ $-30^\circ \le \theta \le +30^\circ$

Observa-se que o ângulo ϕ inicialmente era configurado pelo usuário com valores pertencentes ao intervalo de -90° a 90°, porém considerando os sucessivos incrementos (positivos ou negativos) na sua direção, foi necessário reconfigurar sua faixa para -180° e 180°.

As variáveis x_r , y_r e *dist*_o foram demarcadas com base nas características da estrutura do mundo virtual pré-definido.

Foi configurado um intervalo pequeno para o ângulo de rotação θ , de forma a construir um modelo de representação que fosse coerente com uma estrutura real, ou seja, quando comparado à roda dianteira de um carro, por exemplo, a variação do ângulo é relativamente pequena.

A convenção trigonométrica habitual foi aplicada; assim, os valores positivos de $\phi \in \theta$ representam o sentido anti-horário de rotação do robô e os valores negativos, o sentido horário.

3.5 MÓDULO FUZZY

O comportamento do robô foi simulado pela aplicação de recursos da lógica nebulosa. Desta forma, foi possível configurar os processos de tomada de decisão que um especialista realizaria para cumprir uma determinada tarefa.

O modelo clássico do Mamdani foi utilizado como referência (MAMDANI, 1974), com a manipulação dos conjuntos pelos operadores interseção (*min* - para combinar os antecedentes) e união (*max* - para gerar o conjunto final de saída), assim como o exemplo previamente apresentado na seção 5 do Capítulo anterior. Aplicou-se, também, o método do "centro de gravidade" para a obtenção das ações de controle.

3.5.1 Estrutura do modelo

A cada estágio de simulação, o robô atualizava sua direção baseado nas seguintes informações: ângulo com a horizontal (ϕ) e a distância ao obstáculo mais próximo (*dist*_o). Desta forma, o módulo nebuloso gerava o ângulo de rotação (θ) que, subseqüentemente, seria incrementado ao valor atual de ϕ . O modelo desenvolvido possui a seguinte estrutura:



Figura 3.6: Estrutura do módulo nebuloso do sistema de controle do robô

3.5.2 Conjuntos nebulosos

O universo de discurso das variáveis pertencentes ao módulo *fuzzy* foi subdividido em conjuntos nebulosos que representavam a semântica de termos lingüísticos na tomada de decisão, ou seja, configurou-se o ambiente com as mesmas informações utilizáveis por um especialista no processo de guiar um robô através de um mundo complexo.

Os conjuntos nebulosos utilizam funções de inclusão que caracterizam um número real entre 0 e 1 para todo elemento *x* no universo de discurso *X*. O número $\mu_{\bar{A}}(x)$ representa o grau para o qual um objeto *x* pertence a um determinado conjunto

nebuloso Ã. Desenvolvedores de sistemas *fuzzy*, dependendo de suas preferências e experiências, têm usado diferentes funções de inclusão (YEN e LANGARI 1999, KOSKO 1997, MAMDANI 1997, DUBOIS e WIDROW 1980). Na prática, as funções triangulares e trapezoidais são as mais usadas, pois simplificam o processamento e produzem resultados satisfatórios (KOSKO e KONG 1992). Ambas podem ser representadas por apenas quatro pontos *a*, *b*, *c* e $d \in \Re$. A função triangular é, na verdade, uma trapezoidal degenerada, com b= c (DEMASI, 2003). Os gráficos destas funções podem ser vistos na figura 3.7.



Figura 3.7: Gráfico das funções de inclusão trapézio (a) e triângulo (b)

Inicialmente, as variáveis ϕ , *dist*_o e θ foram definidas em cinco conjuntos nebulosos cada. Mas após a realização de testes com o controlador, verificou-se a necessidade de uma melhor distribuição dos mesmos, pois o problema necessitava de tomadas de decisão mais específicas. Desta forma, as variáveis ϕ e θ foram modificadas para uma configuração com sete conjuntos nebulosos.

A distribuição dos conjuntos para as variáveis de entrada e saída do controlador e suas respectivas funções de inclusão podem ser observadas na figura 3.8.



Figura 3.8: Os conjuntos das variáveis do sistema e as funções de inclusão

Os intervalos traçados para cada conjunto refletem informações fundamentais para o bom funcionamento do sistema. Assim, um detalhamento mais específico das características acima apresentadas é mostrado na tabela 3.1.

Variáveis	Conjuntos	Descrição	Formato	Intervalo
φ	MH	Muito horário	Triângulo	[-180 -145 -90]
Ť	Н	Horário	Triângulo	[-105 -70 -35]
	PH	Pouco Horário	Triângulo	[-45 -30 -3]
	ZE	Zero	Triângulo	[-5 0 5]
	PAH	Pouco Anti-Horário	Triângulo	[3 30 45]
	AH	Anti-Horário	Triângulo	[35 70 105]
	MAH	Muito Anti-Horário	Triângulo	[90 145 180]
disto	MP	Muito perto	Trapézio	[0 0 10 20]
	Р	Perto	Triângulo	[15 25 35]
	MD	Médio	Triângulo	[30 45 60]
	L	Longe	Triângulo	[55 75 100]
	ML	Muito longe	Trapézio	[90 180 220 220]
θ	GDM	Girar à direita muito	Trapézio	[-30 -30 -25 -15]
	GD	Girar à direita	Triângulo	[-25 -15 -5]
	GDP	Girar à direita pouco	Triângulo	[-6 -4 -2]
	ZE	Zero	Triângulo	[-3 0 3]
	GEP	Girar à esquerda pouco	Triângulo	[2 4 6]
	GE	Girar à esquerda	Triângulo	[5 15 25]
	GEM	Girar à esquerda muito	Trapézio	[15 25 30 30]

Tabela 3.1: Detalhamento das características dos conjuntos nebulosos do sistema

Na variável ϕ , pode-se observar que os conjuntos PH (pouco horário), ZE (zero) e PAH (pouco anti-horário) ficaram mais estreitos que os outros. Este fato ocorre para evitar grandes desvios de direção. Assim, se o robô estivesse próximo à linha horizontal, ele evitaria mudanças bruscas em seu sentido, por já estar na direção correta. Os conjuntos restantes tiveram seus tamanhos gradativamente aumentados para obter um controle mais radical quando o robô não estivesse voltado para a parede esquerda (meta), tentando, assim, realizar uma convergência para o ângulo de 0°.

Os conjuntos MP (muito perto), P (perto) e MD (médio) da variável *dist*_o foram configurados como estreitos para ajudar o controlador a tomar decisões rápidas quando o obstáculo estivesse efetivamente próximo a ele.

Para a variável θ , os conjuntos GDP (girar à direita pouco), ZE (zero) e GEP (girar à esquerda pouco) ficaram mais estreitos para permitir um controle refinado nos sucessivos processos de tomadas de decisão. Evitando também mudanças bruscas no sentido quando o controlador estivesse na direção correta.

3.5.3 Regras nebulosas

O controlador, em condições normais, operou baseado em informações sobre o ângulo ϕ com a horizontal e a distância ao obstáculo mais próximo ou à parede (*dist*_o). Desta forma, foi possível modelar o seu comportamento para as sucessivas decisões sobre o ângulo θ de rotação durante o seu percurso. Assim, a base de conhecimento principal foi composta por 35 regras (combinando os 7 conjuntos de ϕ com os 5 de *dist*_o) que permitiram ao robô girar tanto à esquerda quanto à direita através do mundo virtual. Inicialmente, foram definidas regras simétricas em relação ao conjunto zero (ZE) da variável ϕ , ou seja, para os conjuntos muito horário (MH), horário (H) e pouco horário (PH), o robô deveria sempre girar à esquerda, e para os conjuntos pouco anti-horário (PAH), anti-horário (AH) e muito anti-horário (MAH), ele giraria sempre à direita. Vale ressaltar que havia intensidades diferenciadas para cada rotação, levando em consideração a distância ao possível obstáculo (*dist*_o).

Após alguns testes de validação, verificou-se que quando havia obstáculos na rota do robô, a colisão sempre ocorria e o controlador não conseguia tratar tais situações. Foi observado que, se um obstáculo estivesse no caminho e o ângulo fosse pertencente ao conjunto pouco anti-horário (PAH), por exemplo, a regra fazia com que o robô girasse à direita. Assim, a nova direção do ângulo ϕ ficava pertencente ao conjunto pouco horário (PH), tomando a decisão subseqüente de girar à esquerda. Desta forma, a colisão era inevitável, pois as regras do módulo *fuzzy* eram complementares.

Assim sendo, algumas regras foram modificadas, configurando uma tendência do robô de contornar seus obstáculos sempre pelo lado esquerdo. A tabela 3.2 mostra a nova base de conhecimento com tais modificações, que permitiram ao controlador um funcionamento mais adequado. A idéia principal foi tentar convergir o ângulo \u03c6 para a horizontal, de forma a permitir o alcance da parede direita do mundo virtual (meta) o mais rapidamente possível.

rabela 5.2. Base de regras para o controlador do robo	

∳ dist₀	МН	Н	PH	ZE	PAH	АН	МАН
MP	GEM	GEM	GEM	GE	GEP	GEP	GDM
Р	GEM	GEM	GEM	GE	GEP	GEP	GDM
MD	GEM	GE	GEP	GEP	GDP	GD	GDM
L	GEM	GE	GEP	GDP	GDP	GD	GDM
ML	GEM	GE	GEP	GDP	GDP	GD	GDM

A seguir são dadas as explanações sobre a construção da base de regras apresentada.

Nos casos onde ϕ pertencia aos conjuntos muito horário (MH) ou muito anti-horário (MAH), independentemente da distância *dist_o*, a decisão tomada foi de girar à esquerda muito (GEM) ou à direita muito (GDM), respectivamente. Esta medida foi tomada pois estes conjuntos indicavam que o robô estava na direção oposta do seu objetivo, ou seja, o valor absoluto de ϕ era maior que 90°. Assim, necessitava-se de uma mudança brusca no comportamento para atualizar seu sentido.

Quando o obstáculo não estivesse próximo ao robô (conjuntos médio - MD, longe - L ou muito longe - ML) e o ângulo ϕ apontando para regiões próximas à parede inferior ou superior (horário - H ou anti-horário - AH), configurou-se girar à esquerda ou à direita, a fim de convergir rapidamente o valor de ϕ para uma direção próxima à horizontal.

Esta situação foi tratada de forma análoga à anterior, considerando apenas que a direção do robô não se encontrava tão divergente da horizontal, sendo

assim a decisão foi definida como girar à esquerda pouco (GEP) ou à direita pouco (GDP), dependendo da informação corrente do ângulo ϕ .

Baseado no problema inicial, onde todas as regras eram complementares, foi tomada a decisão de contornar obstáculos sempre girando pela esquerda quando os mesmos estivessem próximos ao robô (conjunto muito perto - MP e perto - P). A intensidade deste giro foi determinada de acordo com o ângulo ϕ , seguindo uma gradação relativa a tal direção.

Configurou-se que quando um obstáculo estivesse a uma distância média (MD) do robô e seu posicionamento próximo à horizontal, ele iria girar à esquerda pouco (GEP), seguindo a mesma idéia da medida tomada previamente. Apesar de a distância ser média, o obstáculo já poderia começar a ser tratado, pois o tamanho estreito definido para os conjuntos muito perto (MP), perto (P) e médio (MD) identificava o obstáculo como uma colisão eminente.

A forma como as regras anteriores foram definidas delimitou uma tendência do controlador de realizar giros pelo lado esquerdo. Estas regras restantes tentaram equilibrar os processos de tomadas de decisão, para simular melhor uma situação real, onde um robô poderia decidir se iria girar para uma direção ou para a outra. Assim, se um obstáculo estivesse longe (L) ou muito longe (ML), girava-se um pouco à direita (GDP).

Existiram situações particulares em que o robô estava próximo às paredes superior ou inferior, sobre as quais as regras previamente mostradas não o guiavam corretamente.

A figura 3.9 mostra um exemplo quando o robô estava próximo à parede inferior, seu ângulo ϕ era trigonometricamente pequeno (conjunto ZE) e o valor de

dist^o era relativamente alto (detecção apenas da parede da direita como muito longe – ML ou longe – L, não havendo nenhum outro obstáculo no caminho). A regra definida configurava girar um pouco à direita (GDP), levando a uma inevitável colisão.



Figura 3.9: Exemplo 1 de colisão iminente para o robô

No outro extremo, quando o robô estava próximo à parede superior, o ângulo apontando para cima (pouco anti-horário - PAH ou anti-horário - AH) e uma distância próxima à parede (muito perto - MP ou perto - P), a ação era girar um pouco à esquerda (GEP), o que também seria inadequado. Esta situação pode ser observada na figura 3.10.



Figura 3.10: Exemplo 2 de colisão iminente para o robô

Desta forma, foi necessário agregar ao módulo *fuzzy* a informação do posicionamento do robô no eixo y, para que fosse possível realizar um tratamento mais adequado para a simulação do controlador.

3.5.4 Retificação da estrutura, adição de conjuntos e de regras ao módulo nebuloso

Para agregar a informação do posicionamento y_r do robô, a estrutura do módulo *fuzzy* foi alterada para a seguinte configuração:



Figura 3.11: Estrutura atualizada do módulo nebuloso

Foram também adicionados os conjuntos para a variável y_r , bem como sua respectiva função de inclusão, como mostrado na figura 3.12.





O detalhamento das características apresentadas acima para os conjuntos nebulosos pode ser observado na tabela a seguir:

Variável	Conjuntos	Descrição	Formato	Intervalo
y r	BAIXO	Baixo	Triângulo	[0 10 12]
-	MBAIXO	Meio Baixo	Triângulo	[8 30 40]
	MEIO	Meio	Triângulo	[35 50 65]
	MALTO	Meio Alto	Triângulo	[60 70 92]
	ALTO	Alto	Triângulo	[88 90 100]

Tabela 3.3: Detalhamento das características dos conjuntos nebulosos de y_r

Observe que o universo de discurso de y_r foi subdividido em cinco conjuntos nebulosos. No modelo presente só houve a necessidade de utilizar duas destas

características: o conjunto BAIXO quando o robô estivesse próximo à parede inferior e ALTO quando próximo à superior.

A base de conhecimento anterior foi mantida e duas novas regras foram agregadas. Estas novas delimitações não ficaram vinculadas aos outros parâmetros de entrada, ou seja, independentemente do ângulo ϕ ou da distância ao obstáculo (*dist*_o), o giro para a mudança de rota sempre seria realizado quando o robô estivesse próximo às paredes. As regras adicionadas foram:

Se (y_r é BAIXO) então (θ é GEM) Se (y_r é ALTO) então (θ é GDM)

Desta forma, o módulo *fuzzy* foi completamente configurado, compondo uma base de conhecimento com 37 regras envolvendo três parâmetros de entrada e um de saída. Com estas definições foi possível simular de maneira satisfatória o comportamento do robô simulado pelo mundo virtual com obstáculos.

O sistema nebuloso de controle não utilizava toda a base de conhecimento a cada etapa de simulação. Na maioria das iterações, apenas uma ou duas inferências ficavam ativas. O valor máximo de cinco regras se configurava quando ocorriam casos do posicionamento do robô junto às paredes superiores e inferiores.

3.6 EXEMPLOS

O sistema de controle nebuloso para robô foi construído, obtendo êxito no alcance do objetivo traçado. Para observar o seu comportamento nas tomadas de decisão e trajetórias realizadas no mundo virtual, são apresentados alguns exemplos gráficos das simulações realizadas no ambiente simulado. No início de cada simulação, o usuário configurava os seguintes dados de entrada para o simulador: (1) quantidade de obstáculos, (2) ordenada y_r do robô e (3) ângulo inicial ϕ com a horizontal. Desta forma, cada uma das figuras apresentadas abaixo tem como referência o seguinte padrão: (número de obstáculos, y_r , ϕ).



Figura 3.13: Exemplo 1 com configuração (5, 15, -80°)



Figura 3.14: Exemplo 2 com configuração (5, 50, 90°)



Figura 3.15: Exemplo 3 com a configuração (5, 10, -80°)



Figura 3.16: Exemplo 4 com a configuração (5, 89, 60°)



Figura 3.17: Exemplo 5 com configuração (5, 85, -80°)



Figura 3.18: Exemplo 6 com configuração (5, 20, 60°)


Figura 3.19: Exemplo 7 com configuração (5, 40, -60°)



Figura 3.20: Exemplo 8 com configuração (5, 75, -90°)



Figura 3.21: Exemplo 9 com configuração (5, 10, -60°)



Figura 3.22: Exemplo 10 com configuração (3, 60, -20°)



Figura 3.23: Exemplo 11 com configuração (4, 68, -74°)



Figura 3.24: Exemplo 12 com configuração (4, 20, 60°)

A randomicidade dos obstáculos nem sempre garantia situações simuladas possíveis de serem tratadas. Por exemplo, se houvesse um posicionamento dos mesmos muito próximos à parede superior (ou fossem formadas barreiras verticais encostadas na mesma parede) a colisão seria inevitável, pois o robô foi configurado para contornar sempre pelo lado esquerdo.

Um outro caso particular seria quando o robô estivesse encostado inicialmente em uma das paredes horizontais, com um ângulo \u00e9 apontando para fora do plano XY definido para o mundo virtual. Neste caso, o tratamento pelo módulo nebuloso até seria possível, mas a característica de um deslocamento fixo para cada etapa da simulação não permitia ao controlador efetuar giros em torno do próprio eixo. Assim, uma falha também era iminente.

Porém, para todos os casos onde era possível realizar um percurso válido no alcance da parede direita, o robô sempre desempenhou com sucesso a sua tarefa de controle.

3.7 CONSIDERAÇÕES PARCIAIS

O objetivo deste Capítulo foi fazer uma descrição e explanação do sistema de controle nebuloso desenvolvido para guiar um robô simulado simples através de um mundo virtual, que continha obstáculos fixados randomicamente.

Foram detalhadas as informações sobre os aspectos gerais e específicos dos tratamentos realizados por este sistema, identificando o funcionamento dos sensores do robô, as movimentações através do mundo virtual, o universo de discurso dos parâmetros, as tomadas de decisão com inferências nebulosas e a apresentação de exemplos de simulação.

Vários testes foram realizados para validar todas as configurações definidas para este modelo. O robô obteve êxito no alcance de seu objetivo final para a grande maioria das simulações realizadas. Ocorreram falhas quando situações desfavoráveis haviam sido inoportunamente aplicadas.

A simplicidade e a facilidade de construção do controlador sugerem que, para algumas tarefas de controle, é possível criar um sistema nebuloso que simule, de forma satisfatória, condições reais.

Este modelo será referenciado no Capítulo subseqüente, quando será mostrado um estudo de estabilidade e robustez para o controlador do robô, frente às sucessivas retiradas de regras da base de conhecimento do sistema.

CAPÍTULO 4

4. ESTUDO DA ESTABILIDADE E ROBUSTEZ DO SISTEMA

O objetivo deste Capítulo é mostrar os estudos de estabilidade e robustez que foram desenvolvidos para avaliar a tarefa de controlar um robô simulado simples através de mundo virtual com obstáculos. Inicialmente, o sistema original foi testado com as configurações mostradas no Capítulo anterior. Análises comparativas e modelagens adicionais foram incrementadas com o intuito de observar os possíveis comportamentos deste mesmo sistema em diferentes situações.

4.1 CARACTERÍSTICAS GERAIS

Para testar a estabilidade e a robustez do sistema, foi definido o critério de retirada sucessiva de regras da base de conhecimento, com a finalidade de monitorar o comportamento do robô durante todas as etapas de simulação nos experimentos realizados.

Esta análise sugere que um projetista está apto a construir fácil e rapidamente um sistema de controle que seja estável, mesmo sem conhecer todo o escopo das possibilidades de um determinado problema, ou seja, ele não precisa necessariamente possuir o conhecimento total das regras envolvidas em um contexto específico. Para tal finalidade, os recursos trazidos pela lógica nebulosa devem ser aplicados, permitindo ao simulador desempenhar de maneira satisfatória condições reais.

4.1.1 Configuração dos parâmetros

A fim de definir os parâmetros adequados para os experimentos desenvolvidos, foram realizados testes de simulação com o controlador nebuloso em diferentes circunstâncias, que permitiram avaliar a sua habilidade em guiar um robô através de um mundo complexo.

Inicialmente, o número de obstáculos variava entre 1 e 5 unidades, sendo que os mesmos eram fixados em posições randomicamente distribuídas. Para cada um destes arranjos testavam-se 10 diferentes pontos de partida no eixo *y* (parâmetro y_r) e 5 ângulos iniciais formados com a horizontal (ϕ), que foram simetricamente distribuídos sobre o universo de discurso de cada variável.

Guiar o robô da parede esquerda para a da direita continuava a ser o objetivo, evitando as possíveis colisões. Foi verificado que o controlador era capaz de desempenhar com êxito sua tarefa de controle, pois, para todas as situações testadas, ele guiava o robô corretamente à meta traçada.

Para garantir a validade das comparações entre as simulações dos experimentos, foi necessário fixar alguns parâmetros. Assim, definiu-se que o número e a posição dos obstáculos seriam constantes, sendo a escolha de suas configurações feita de maneira a assegurar que o robô, independentemente de sua posição inicial, sempre teria pelo menos um obstáculo no seu percurso. Desta forma, o posicionamento foi definido nos 5 pontos de vértice de um pentágono irregular inclinado. As coordenadas são mostradas na tabela 4.1, bem como a sua distribuição final no mundo virtual, como apresentado na figura 4.1.

Xo	Уo
59.1	48.5
78.8	19.4
118.2	77.6
137.9	29.1
157.6	58.2

Tabela 4.1: Coordenadas dos obstáculos fixados



Figura 4.1: Diagrama do mundo virtual com obstáculos

Observe que, aparentemente, as posições próximas às paredes superiores e inferiores não apresentariam dificuldades para o controlador no desenvolvimento de sua trajetória frente aos obstáculos. Porém, as tomadas de decisão definidas no sistema de inferência nebuloso (regras) nunca geram ações para o robô se deslocar exatamente em uma linha horizontal, garantindo, assim, a presença de possíveis colisões para qualquer caso simulado.

Foram definidas 50 disposições iniciais combinando os parâmetros ângulo ϕ com a horizontal (-90°, -45°, 0°, 45°, 90°) e posição *y*_r na parede esquerda (6, 15, 25, 35, 45, 55, 65, 75, 85, 94).

Quando o robô estava encostado na parede superior ou inferior com o seu ângulo direção ϕ apontando para fora do mundo virtual, ou seja, nas combinações (6, -45°), (6, -90°), (94, 45°) e (94, 90°), as colisões eram inevitáveis. Isto ocorria porque o controlador se movimentava a um passo fixo e a atualização do ângulo com a horizontal não era suficientemente grande para mudar o sentido da rota do robô para um deslocamento interno ao mundo virtual, ou seja, mesmo quando o sistema tentava atualizar corretamente o ângulo direção ϕ , o próximo movimento realizado levaria a uma falha. Vale salientar que o controlador pára sua movimentação quando situações de erro acontecem durante o percurso, seja em choques com paredes ou com obstáculos. Assim, estas configurações foram desconsideradas, por apresentarem condições iniciais desfavoráveis ao modelo desenvolvido. As 46 disposições restantes foram fixadas, delimitando as amostras que compunham os experimentos.

4.1.2 Critérios de remoção

Foram definidos cinco diferentes critérios para a retirada de regras do sistema de inferência nebuloso: (1) a remoção aleatória; (2) a seqüencial por coluna; (3) a seqüencial inversa por coluna; (4) as periféricas depois as centrais; (5) e as centrais depois as periféricas.

A idéia foi simular uma quantidade razoável de remoções para obter dados representativos do comportamento do controlador em condições diversas.

O critério aleatório pôde ser aplicado diversas vezes em um mesmo experimento, configurando diferentes modos de remoção. As outras ordens de extração representaram situações particulares. Assim, também foi observada a relevância de certos conjuntos de regras (centrais, periféricas ou pertencentes a alguma coluna) dentro do escopo de um determinado problema.

4.1.3 Exemplo de aplicação

Inicialmente todas as amostras eram processadas com a base de conhecimento completa do ambiente. A cada etapa, uma regra era retirada de acordo com um dos critérios pré-definidos, ocorrendo subseqüentemente o reprocessamento das mesmas amostras. E assim o processo ocorria sucessivamente, até que todas as informações do sistema de inferência fossem esgotadas.

Para elucidar este processo, será mostrado um exemplo com o detalhamento das simulações. Cada uma das amostras percorria por diferentes trajetórias, pelo fato da especificidade de cada configuração ($y_r e \phi$). Assim, se todos os percursos fossem mostrados em um mesmo gráfico, ficaria complicado distinguir as informações geradas. Então, para efeito de explicação, ficou definido que apenas a posição final das mesmas seria representada no exemplo. O arquétipo delimitado pode ser observado na área destacada da figura 4.2 (b) que tem como base a representação de uma amostra no momento quando houve a colisão do robô com um obstáculo no mundo virtual (a).



Figura 4.2: Colisão do robô com um obstáculo durante a trajetória (a) e o arquétipo delimitado para representar o ponto onde tal amostra ficou estagnada (b)

Um exemplo hipotético é mostrado a seguir, onde o critério de remoção aleatória de regras foi aplicado. As figuras representam alguns dos intervalos do

processo realizado, identificando em cada gráfico as posições alcançadas pelas 46 amostras nas retiradas sucessivas de regras.



Figura 4.6: Posicionamentos de alcance das amostras no sistema com 3 (a), 2 (b) e 1 regra (c)

Observe que a quantidade de amostras que conseguia atingir o objetivo final foi diminuindo à medida que as regras haviam sido retiradas da base de conhecimento. As colisões ficavam cada vez mais freqüentes, pois as tomadas de decisão realizadas não guiavam o robô de maneira adequada.

Tais informações permitiam observar o comportamento do sistema de uma maneira geral, pois era possível identificar tanto os momentos de estabilidade como os períodos críticos ocorridos nas simulações.

4.1.4 Características da análise de estabilidade

A estabilidade do sistema foi analisada nas sucessivas retiradas de regras com base em dois parâmetros: erro de trajetória e número absoluto de amostras que alcançavam o objetivo.

O erro de trajetória foi definido como a distância euclidiana da atual posição final do robô à parede direita do mundo virtual. Este cálculo foi realizado para cada uma das amostras processadas nos experimentos. A figura 4.7 mostra dois exemplos de como esta medida foi coletada, com base nas informações do posicionamento onde o robô ficou estagnado.



Figura 4.7: Cálculo do erro de trajetória

Quando a meta do robô era alcançada, este parâmetro ficava com valor nulo. Porém, para os outros casos, a distância gerada significava o grau de acerto do controlador no desenvolvimento do percurso. Delimitando que quanto menor o erro de trajetória, maior o sucesso obtido. A partir dos dados observados na figura 4.8 (seguindo o padrão delimitado para a visualização das posições finais das amostras), é possível afirmar que na média o grau de acerto do exemplo (a) é maior que a do (b), configurando, assim, uma situação mais estável.



Figura 4.8: Dois exemplos de configurações, identificando os pontos onde as amostras ficaram estagnadas

Entretanto, o valor desta média não permitia distinguir um valor absoluto do número de amostras que alcançavam a meta. Desta forma, foi definido o segundo parâmetro para a análise de estabilidade, que permitia inferir diretamente a quantidade de êxitos obtidos pelo controlador nas diversas situações configuradas para os experimentos.

4.2 EXPERIMENTOS

Foram realizados cinco experimentos para avaliar o comportamento do controlador do robô em diferentes condições simuladas. O objetivo foi investigar a estabilidade do modelo para tais configurações, permitindo apontar características sobre a robustez do sistema projetado.

4.2.1 Estudo de estabilidade do modelo original

O sistema de controle nebuloso mostrado no Capítulo 3 foi utilizado como referência para delimitar este primeiro estudo de estabilidade (MORATORI *et col,* 2004). Todas as configurações definidas previamente para o controlador foram mantidas, submetendo o mesmo ao processo de retirada de conhecimento de sua base de regras, nas sucessivas simulações realizadas.

A figura 4.9 (a) mostra um exemplo de uma trajetória completa do robô, partindo da posição $y_r = 7$ com o ângulo $\phi = -30^\circ$. Este caso foi realizado sem retirar regras da base de conhecimento do sistema original. Adicionalmente, é mostrado um gráfico (b) com as informações de variação de ϕ para o trajeto percorrido.



Figura 4.9: Exemplo de uma trajetória do robô no mundo virtual com obstáculos (a) e variação do ângulo ϕ com a horizontal (b)

De forma análoga à apresentada na seção 4.1.3 no exemplo de aplicação, a taxa de acerto do controlador diminuía à medida que as regras eram removidas. Foi definido um total de 14 seqüências diferentes de retirada de regras, sendo 10 aleatórias e 4 seguindo critérios particulares de remoção.

A tabela 4.2 mostra a ordem definida para a retirada seqüencial por coluna (a) e a seqüência de extração periféricas / centrais (b). Observe que as informações da tabela 4.2 (a) e (b) começaram na terceira remoção, pois as regras adicionais do sistema original (controle das paredes) foram primeiramente selecionadas, delimitando a seguinte orientação:

> 1°→ Se (y_r é BAIXO) então (θ é GEM) 2°→ Se (y_r é ALTO) então (θ é GDM)

∳ dist₀	мн	Н	PH	ZE	PAH	AH	MAH
MP	3°	8°	13°	18°	23°	28°	33°
Р	4°	9 °	14°	19°	24°	29°	34°
MD	5°	10°	15°	20°	25°	30°	35°
L	6°	11°	16°	21°	26°	31°	36°
ML	7°	12°	17°	22°	27°	32°	37°
(a)							

∳ dist₀	мн	Н	PH	ZE	PAH	AH	МАН
MP	3°	8°	13°	15°	17°	19°	24°
Р	4°	9°	31°	30°	29°	20°	25°
MD	5°	10°	37°	36°	35°	21°	26°
L	6°	11°	34°	33°	32°	22°	27°
ML	7°	12°	14°	16°	18°	23°	28°
(b)							

Tabela 4.2: Seqüência de remoção de regras para o critério seqüencial por coluna (a) e para a extração periféricas / centrais (b)

Os outros dois casos particulares de extração utilizaram exatamente a seqüência inversa destas configurações.

O detalhamento dos critérios de remoção é importante, pois para os outros experimentos serão utilizadas diferentes seqüências de retirada, permitindo, então, a obtenção de características dos possíveis comportamentos do controlador frente às mudanças.

Neste caso particular foi definido que as amostras que estivessem inicialmente encostadas na parede (superior ou inferior) seriam automaticamente desconsideradas, gerando assim um total de 40 disposições iniciais.

Combinando as 40 posições de saída com a remoção de 37 regras e as 14 diferentes seqüências de extração de conhecimento, foi obtido um total de 20720 execuções do simulador, que permitiu coletar informações para a análise de sua estabilidade.

4.2.1.2 Simulações e resultados

Após o processamento de todas as simulações, os dados de erro de trajetória foram agregados em um único gráfico delimitando a média de acertos do controlador frente às sucessivas extrações de regras. A figura 4.10 (a) mostra tais informações. Adicionalmente, para uma melhor interpretação da análise deste parâmetro, foi calculado o desvio padrão (b) para estimar a dispersão desta média obtida.



Figura 4.10: Média final do erro de trajetória (a) e o desvio padrão (b)

Observou-se que os valores do erro eram incrementados a cada processo de remoção realizado. O comportamento das amostras, nos diferentes critérios de retirada, não foi uniforme, pois o valor do desvio padrão sofreu um crescimento considerável, ou seja, o seu tamanho em relação aos valores da média foi relativamente alto.

O erro de trajetória refletiu a média de sucesso do simulador, porém também foram gerados altos valores para o desvio padrão. Assim, para inferir mais apropriadamente os percentuais de estabilidade geral do modelo escolheu-se referenciar os dados absolutos, que identificavam a quantidade de amostras que atingia o objetivo frente às extrações de conhecimento. A figura 4.11 identifica tais informações.



Figura 4.11: Média final da quantidade de amostras que atingia o objetivo

O gráfico acima mostra que mesmo após a retirada de 10 regras do sistema, 79,64% de todas as amostras ainda eram capazes de atingir o seu objetivo final. Após a remoção de 20 regras, aproximadamente 33,46% das amostras ainda eram bem sucedidas.

Considerando os resultados das simulações realizadas, observou-se que não havia uma predominância de posições favoráveis de saída do robô, confirmando o fato de que todas as configurações de saída apresentavam algum tipo de dificuldade durante o desenvolvimento de seu percurso no mundo virtual.

A figura 4.12 mostra a quantidade de amostras que obteve êxito no alcance do objetivo quando a ordem de remoção inversa por coluna foi aplicada.



Figura 4.12: Quantidade de amostras que atingia o objetivo segundo um critério particular de remoção

21^a Existe uma diminuição acentuada na extração da regra. Este comportamento não se repetiu quando a mesma era retirada segundo outros critérios de remoção. Porém, observou-se que este padrão de decrescimento também acontecia para outras regras em situações particulares. Esta característica sugere que a importância de uma regra pode ser delimitada com base na ordem e na quantidade de informações que foram previamente removidas do ambiente simulado, fazendo, então, um estudo de caso a caso. É interessante notar ainda que, após a próxima remoção (22^a), levou novamente ao aumento do número de resultados positivos. Assim, foi verificado que algumas regras possuíam um efeito inibitório (ou positivo) em algumas condições e após a sua remoção o robô ficava livre (ou impedido) para tomar um conjunto particular de decisões.

Uma outra análise interessante refere-se à ordem de retirada de dois grupos característicos de regras: periféricas e centrais. A figura 4.13 mostra a quantidade de amostras que alcançou a meta quando as regras foram retiradas na seguinte ordem: (a) periféricas/ centrais e (b) centrais /periféricas.



Figura 4.13: Quantidade de amostras que atingia o objetivo segundo critérios específicos de remoção

O estudo comparativo dos gráficos da figura 4.13 (a) e (b) evidencia que para ambas as estratégias apresentadas geraram aproximadamente as mesmas quantidades de êxitos para o controlador. Assim, foi possível apontar que as regras centrais e periféricas possuíam a mesma importância para o sistema nebuloso de inferência.

4.2.2 Estudo de estabilidade de um controlador com velocidade variável

O sistema de controle nebuloso original sofreu algumas modificações para delimitar as características deste segundo experimento (MORATORI *et col*, 2005a). Na tentativa de simular melhor uma situação real, em que um objeto tentava se mover em um mundo complexo, duas formas de tratamento da informação foram necessárias, ou seja, não apenas tomar a decisão do ângulo direção como também configurar a velocidade do controlador, permitindo, assim, um deslocamento mais satisfatório para os percursos realizados.

Esta propriedade não gerou modificações nas equações de movimento que foram descritas previamente. Houve alteração na referência do parâmetro Δ , que neste momento deixou de ter um valor fixo para possuir dados variáveis.

O intervalo definido para os possíveis tamanhos do deslocamento foi:

$$-0.3 \leq \Delta \leq +10$$

Assim, o robô podia se deslocar com passos curtos ou mais alongados dependendo de cada situação encontrada no desenvolvimento de sua trajetória.

Ocorreu uma reformulação da estrutura do módulo nebuloso, já que o mesmo passou a gerar também informações sobre o deslocamento. A figura 4.14 mostra tal evolução.



Figura 4.14: Reformulação da estrutura do módulo nebuloso

Foram agregadas ao sistema as demarcações para os conjuntos nebulosos da variável de saída Δ , identificando os respectivos valores para a função de inclusão dos mesmos, conforme observado na figura 4.15.



O detalhamento das características apresentadas acima para os conjuntos pode ser observado na tabela a seguir:

Variável	Conjuntos	Descrição	Formato	Intervalo
Δ	ZE	Zero	Triângulo	[-0,3 0 0,6]
	ACP	Acelerado Pouco	Triângulo	[0,4 0,9 1]
	NOR	Normal	Triângulo	[0,9 1,3 1,7]
	AC	Acelerado	Triângulo	[1,5 3 4,5]
	ACM	Acelerado Muito	Trapézio	[4 5 10 10]

Tabela 4.3: Detalhamento das características dos conjuntos nebulosos de ${\boldsymbol \Delta}$

O formato de ZE, ACP e NOR ficou mais estreitado que os outros, pela necessidade de um tratamento refinado da velocidade quando colisões no percurso eram iminentes.

Para cada uma das regras originais do módulo *fuzzy* adicionou-se uma inferência para o parâmetro passo.

Em condições reais, um especialista com a tarefa de guiar um carro por uma pista possui tomadas de decisão para a orientação de direção e adequação de velocidade para o veículo, sendo estas medidas realizadas de forma concomitante no desenvolvimento de seu trajeto. Este referencial delineou que os tratamentos para este novo modelo seriam feitos desta mesma forma, gerando a cada etapa de simulação um par ordenado de saídas, combinando o ângulo de rotação e o tamanho do deslocamento (θ , Δ). A tabela 4.4 mostra as regras produzidas.

dist _o	MH	Н	РН	ZE	РАН	AH	MAH
MP	(GEM, ZE)	(GEM, ZE)	(GEM, ZE)	(GE, ZE)	(GEP, ZE)	(GEP, ZE)	(GDM, ZE)
Р	(GEM, ACP)	(GEM, NOR)	(GEM, NOR)	(GE, NOR)	(GEP, NOR)	(GEP, NOR)	(GDM, ACP)
MD	(GEM, NOR)	(GE, AC)	(GEP, AC)	(GEP, ACM)	(GDP, AC)	(GD, AC)	(GDM, NOR)
L	(GEM, AC)	(GE, AC)	(GEP, AC)	(GDP, ACM)	(GDP, AC)	(GD, AC)	(GDM, AC)
ML	(GEM, AC)	(GE, AC)	(GEP, AC)	(GDP, ACM)	(GDP, AC)	(GD, AC)	(GDM, AC)

Tabela 4.4: Base de regras para o novo sistema

Adicionalmente, as delimitações referentes ao posicionamento no eixo y ficaram com a seguinte definição:

Se
$$(y_r \in BAIXO)$$
 então $(\theta \in GEM) e (\Delta \in ZE)$
Se $(y_r \in ALTO)$ então $(\theta \in GDM) e (\Delta \in ZE)$

Basicamente, para construir estas novas inferências para parâmetro Δ , foram considerados dois aspectos das variáveis de entrada: (1) o tamanho da distância ao obstáculo e (2) o quanto o ângulo ϕ estava próximo da linha horizontal. Assim, uma maior velocidade era obtida quando o robô estivesse apontado para a parede direita e em momentos onde não existissem colisões iminentes. As gradações configurando tais características podem ser visualizadas na tabela anterior.

De forma análoga, os tratamentos para as posições no eixo y geraram deslocamentos muito curtos, pois se considerou a pequena distância a um obstáculo (parede superior ou inferior).

Pelo fato do controlador possuir um controle refinado junto aos obstáculos e paredes, havia situações em que o robô se movia lentamente. Assim, um número máximo de 500 movimentações para a realização da trajetória foi delimitado, de forma a evitar tempos excessivos de processamento.

Para o estudo da estabilidade deste modelo foram mantidos os mesmos critérios aplicados no experimento anterior, com as sucessivas remoções de regras da base de conhecimento, seguindo um dos critérios de retirada.

Um exemplo de um percurso completo realizado pelo robô nestas novas configurações pode ser visualizado na figura 4.16 (a). A posição inicial era $y_r = 55$ com ângulo $\phi = 22^{\circ}$. Este caso também foi simulado sem a retirada de regras. Adicionalmente, são mostrados os gráficos com informações de variação do ângulo com a horizontal (b) e do deslocamento (c).



Figura 4.16: Exemplo de uma trajetória do robô com as novas configurações (a), variação do ângulo com a horizontal (b) e a variação do deslocamento (c)



Neste experimento foram agregadas 10 seqüências de extração junto às 14 presentes no estudo anterior, configurando 20 aleatórias e 4 com critérios particulares.

Houve modificações na ordenação para algumas retiradas. As tabelas 4.5 e 4.6 mostram, respectivamente, as novas seqüências para a remoção por coluna e para as periféricas / centrais.

∳ dist₀	МН	Н	PH	ZE	PAH	AH	МАН
MP	3°	13°	23°	33°	28°	18°	8°
Р	4°	14°	24°	34°	29°	19°	9°
MD	5°	15°	25°	35°	30°	20°	10°
L	6°	16°	26°	36°	31°	21°	11°
ML	7°	17°	27°	37°	32°	22°	12°

Tabela 4.5: Nova seqüência de remoção para o critério seqüencial por coluna

As regras adicionais para este critério tiveram a seguinte ordem:

1°→ Se (y_r é BAIXO) então (θ é GEM) 2°→ Se (yr é ALTO) então (θ é GDM)

Tabela 4.6: Nova seqüência de remoção para o critério periféricas / centrais

∳ dist₀	мн	Н	PH	ZE	PAH	AH	MAH
MP	2°	12°	22°	25°	21°	11°	1°
Р	6°	16°	28°	31°	27°	15°	5°
MD	10°	20°	34°	35°	33°	19°	9°
L	8°	18°	30°	32°	29°	17°	7 °
ML	4°	14°	24°	26°	23°	13°	3°

Para o critério periféricas / centrais as definições foram as seguintes:

36°→ Se (y_r é BAIXO) então (θ é GEM) 37°→ Se (yr é ALTO) então (θ é GDM)

Os outros dois casos particulares utilizaram a seqüência inversa destas configurações.

As alterações realizadas tiveram o intuito de formular uma situação diferenciada para observação do comportamento do controlador.

A forma como o sistema foi desenvolvido originalmente delimitou que as tomadas de decisão permitiriam desvios de obstáculos tanto pelo lado direito quanto pelo esquerdo, gerando na base de conhecimento a complementaridade entre determinadas regras. A nova ordem de extração nos casos particulares levou em consideração tais características, retirando as informações de forma complementar. Neste estudo foram geradas 46 disposições iniciais para o controlador, desconsiderando apenas as amostras que inicialmente estavam encostadas na parede com um ângulo ϕ apontando para fora do mundo virtual.

Combinando estas 46 posições de saída com a remoção de 37 regras e as 24 diferentes seqüências de extração de conhecimento, foi obtido um total de 40848 execuções do simulador, que permitiu coletar informações para a análise de sua estabilidade.

4.2.2.2 Simulações e resultados

Os cálculos da média do erro de trajetória e do desvio padrão seguiram a mesma abordagem mostrada no experimento anterior. As informações obtidas podem ser observadas na figura 4.17 (a) e (b), respectivamente. Os comportamentos de dois processos de extração (S02 - critério aleatório e S21 - critério seqüencial por coluna) também estão presentes nos gráficos. Estas curvas fizeram parte da composição final da média apresentada.



Figura 4.17: Média final do erro de trajetória (a) e o desvio padrão (b)

Como esperado, obteve-se um aumento da taxa de erro a cada retirada de regra do sistema. O comportamento das amostras também não foi uniforme, pois o desvio padrão possuiu um valor relativamente alto quando comparado com a média gerada. Os casos particulares seguiram as mesmas características.

A figura 4.18 identifica o valor absoluto da quantidade de amostras que atingia a parede direita do mundo virtual frente às extrações de conhecimento.



Figura 4.18: Média final da quantidade de amostras que atingia o objetivo

Pelas informações apresentadas acima, observa-se que após a remoção de 7 regras do sistema, 71,29% de todas as amostras ainda conseguiam chegar a meta. Na retirada de 22 regras, aproximadamente 28,99% das amostras obtinham êxito.

Confirmou-se novamente a não predominância de posições favoráveis nas disposições iniciais de saída do robô, pois dificuldades no percurso foram encontradas para todos os casos simulados.

A restrição imposta de realizar no máximo 500 movimentações para o desenvolvimento da trajetória gerou um percentual de 0,56% de ocorrência em relação ao número total das amostras processadas. Este valor foi baixo, identificando que o sistema não ficou demasiadamente envolvido no tratamento das colisões, de forma a comprometer o rendimento geral do modelo.

As características que delimitavam a importância de uma determinada regra também foram observadas neste experimento. Tal relevância esteve relacionada com a ordem e quantidade de conhecimentos extraídos para cada caso específico. Assim, uma regra que era fundamental em um momento, poderia ser completamente dispensável em outro, além dos possíveis efeitos inibitórios (ou positivos) que a mesma poderia estabelecer no sistema.

As modificações realizadas para a ordem de remoção das regras periféricas e centrais, seguindo a complementaridade das inferências do módulo nebuloso, geraram um comportamento divergente do apresentado no estudo anterior. A figura 4.19 mostra as seqüências de retirada para periféricas / centrais (a) e centrais / periféricas (b).



Figura 4.19: Quantidade de amostras que atingia o objetivo segundo critérios específicos de remoção

A comparação entre os gráficos (a) e (b) da figura 4.19 evidencia diferenças perceptíveis entre o desempenho das amostras. De forma geral, o grau de sucesso em (a) é maior que em (b). Também foi verificado que em nenhum outro critério de remoção aplicado, houve um comportamento tão abrupto na queda do desempenho, quando apenas 3 regras haviam sido extraídas. Assim, tal análise aponta que para

as novas configurações do sistema neste experimento, existe uma suposta importância do conjunto das regras centrais.

4.2.3 Estudo comparativo do desempenho entre os dois sistemas prévios

Foi realizado um estudo comparativo entre as duas configurações previamente apresentadas para o sistema de controle do robô (MORATORI *et col,* 2005b). Na primeira abordagem, como mostrado anteriormente, as decisões inferidas pelo módulo nebuloso controlavam apenas o ângulo direção a ser desenvolvido nas trajetórias. Enquanto que na segunda, tratava-se também a velocidade dos deslocamentos efetuados.

Para verificar o desempenho, observou-se o contexto de análise de estabilidade do controlador frente às sucessivas extrações de regras nas suas diferentes configurações, examinando os parâmetros de tempo de processamento e a quantidade de movimentações desenvolvidas nos percursos.

O objetivo foi mostrar que resultados mais satisfatórios poderiam ser obtidos quando o sistema incorporava o gerenciamento da velocidade, trazendo processamentos mais adequados, para uma futura transposição para um modelo real de robótica.

Para caracterizar os tratamentos aplicados nas diferentes abordagens, são mostrados a seguir gráficos com as trajetórias efetuadas pelo robô e as variações de $\phi e \Delta$ para as duas configurações apresentadas. A posição de partida foi *y*_r = 53 com o ângulo ϕ = -24°. Este caso também foi realizado sem nenhuma extração de regras da base de conhecimento do sistema.







Figura 4.21: Exemplo de uma trajetória do robô na segunda abordagem (a), variação do ângulo com a horizontal (b) e a variação do deslocamento (c)

É possível observar nas figuras anteriores como houve mudanças no tratamento das informações do ambiente. O passo fixo da primeira abordagem, na figura 4.20 (c), levou a diferentes tomadas de decisão quanto à trajetória a ser desempenhada pelo robô, conforme a visualização dos gráficos de 4.20 (a) e 4.21 (a). Foi possível identificar que na segunda configuração ocorreu uma menor quantidade de deslocamentos para atingir o objetivo final desejado, quando comparados os resultados entre 4.20 (b) e 4.21 (b).

4.2.3.1 Características específicas

As 24 diferentes seqüências de extração de regras aplicadas no segundo experimento foram utilizadas como padrão para realizar este estudo comparativo de desempenho.

Foram também delimitadas as mesmas 46 disposições iniciais para o robô, além da restrição da quantidade máxima de 500 movimentações para a realização do percurso no desenvolvimento das duas abordagens.

O sistema original não possuía as mesmas características para as quantidades de amostras e de seqüências de retirada, além da limitação do número de deslocamentos. Assim, foi necessário reprocessar todas as simulações para fazer uma efetiva comparação. Vale ressaltar que, nos critérios aleatórios, aplicavam-se seqüências idênticas de remoção para as duas configurações.

Combinando as 46 posições de saída com a retirada de 37 regras e as 24 diferentes seqüências de extração de conhecimento para cada um dos experimentos, foi obtido um total de 81696 execuções do simulador, que permitiu coletar informações tanto de estabilidade como de desempenho do controlador.

Foram colhidos para todas as amostras simuladas os dados sobre o tempo de processamento e quantidade de passos realizados. Os testes foram desenvolvidos em um computador equipado com um processador Intel® Pentium® M 710 (1.40 GHz) com 512 MB de memória DDR SDRAM (2 x 256).

4.2.3.2 Simulações e resultados

A estabilidade do controlador foi analisada utilizando como referência apenas o valor absoluto das amostras que alcançavam o objetivo final. A figura 4.22 apresenta tais informações para as duas abordagens definidas, sendo a linha tracejada para a configuração com uma variável de saída e a contínua para duas variáveis.



Figura 4.22: Quantidade de amostras que atingiram o objetivo final para as duas configurações do controlador

A abordagem que utilizou um passo constante (linha tracejada) obteve um melhor desempenho na estabilidade. Era possível fazer modificações no controlador para aperfeiçoar os tratamentos aplicados no ambiente de passo variável. Entretanto, para efeito de comparação, foi necessário criar um sistema que permitisse guiar corretamente todas as amostras (quando a base de regras estivesse completa) para ambas as configurações. Desta forma, otimizações não

puderam ser aplicadas, pois a melhoria voltada a um caso não necessariamente atenderiam às necessidades do outro. Vale ressaltar que a diferença entre os resultados obtidos para as duas abordagens é relativamente pequena.

O desempenho do sistema foi avaliado utilizando os dois critérios: tempo de processamento e quantidade de passos realizados durante a trajetória. Para obter uma visão mais completa, foi observado o comportamento geral das amostras e o desempenho específico apenas daquelas que atingiam a meta.

Na análise geral das amostras, foram registrados os seguintes dados para cada uma das configurações: (1) a média do número de passos realizados em todas as simulações; (2) a média do tempo de processamento de cada amostra; (3) a média do tempo gasto para simular todas as amostras em cada um dos critérios de remoção de regras e (4) o tempo total gasto para efetuar todas as simulações.



Figura 4.23: Média da quantidade de passos realizados (a) e do tempo gasto no processamento de cada amostra (b), para as duas configurações do controlador

A figura 4.23 (a) apresenta, simultaneamente, a média do número de passos realizados pelas duas abordagens. Percebe-se claramente que os valores obtidos para o passo constante (linha pontilhada) são expressivamente maiores durante todo o processo de retirada de regras, refletindo, assim, um pior rendimento. Do mesmo modo na figura 4.23 (b), onde se observa a média de tempo de

processamento (em segundos) por amostra, verifica-se também a menor eficiência desta configuração. Outras características desta análise estão ilustradas na tabela 4.7.

	Característica	Passo constante	Passo variável
Tempo (segundos)	Média de tempo gasto para simular todas as amostras em cada um dos 24 critérios de remoção de regras	373.2	221.3
	Tempo total de execução para todas as simulações	8958.1	5312.4

Tabela 4.7: Análise geral das amostras

No estudo específico das amostras bem sucedidas foram observados os seguintes aspectos: (1) o número médio de passos realizados para atingir o objetivo e seu respectivo menor valor; e (2) a média de tempo gasto para simular uma única amostra e sua menor referência.

	,	3	
	Característica	Passo constante	Passo variável
Passos (unidades)	Média de passos para simular todas as amostras em cada um dos critérios de retirada de regras	154.06	78.51
	Menor quantidade de passos	143	36
Tempo (segundos)	Média de tempo para simular uma única amostra no experimento	0.33	0.18
	Menor tempo gasto por uma amostra	0.31	0.13

Tabela 4.8: Análise das amostras que atingiram a meta

Por essas duas vertentes foi possível observar o melhor desempenho do sistema com deslocamento variável, para todos os casos observados, exceto a pequena diferença na estabilidade apresentada na figura 4.22. De uma maneira geral, a configuração com Δ constante necessita de aproximadamente 49 passos adicionais para cumprir com êxito uma mesma tarefa realizada com Δ variável. O tempo total de execução também gastou 3645.7 segundos a mais para processar integralmente as amostras definidas. Os resultados indicam que é possível incrementar um controle de velocidade no modelo sem comprometer sua eficiência.

4.2.4 Estudo de estabilidade de um controlador otimizado com velocidade variável

Neste experimento, algumas características do controlador foram modificadas, para dar continuidade na análise do comportamento do sistema frente a diferentes configurações (MORATORI *et col,* 2005c). Esta nova abordagem representa uma otimização dos tratamentos previamente aplicados, permitindo ao robô atingir mais rapidamente o seu objetivo através de um controle mais adaptável de sua movimentação pelo mundo virtual.

A estrutura estabelecida manteve o módulo nebuloso de inferência para o controle dos parâmetros velocidade e ângulo direção.

Vários testes de simulação foram realizados, delimitando algumas modificações no tamanho dos conjuntos das variáveis ϕ , $y_r \in \theta$, mostrados nas áreas sombreadas da figura 4.24.



O detalhamento das informações apresentadas acima pode ser observado na

tabela a seguir:

Variáveis	Coniuntos	Descricão	Formato	Intervalo
ά.	MH	Muito horário	Triângulo	[-180 -145 -70]
Ψ	Н	Horário	Triângulo	[-85 -60 -35]
	PH	Pouco Horário	Triângulo	[-45 -30 -3]
	ZE	Zero	Triângulo	[-5 0 5]
	PAH	Pouco Anti-Horário	Triângulo	[3 30 45]
	AH	Anti-Horário	Triângulo	[35 60 85]
	MAH	Muito Anti-Horário	Triângulo	[70 145 180]
y r	BAIXO	Baixo	Triângulo	[0 9 9]
-	MBAIXO	Meio Baixo	Triângulo	[8 30 40]
	MEIO	Meio	Triângulo	[35 50 65]
	MALTO	Meio Alto	Triângulo	[60 70 92]
	ALTO	Alto	Triângulo	[91 91 100]
θ	GDM	Girar à direita muito	Trapézio	[-30 -30 -25 -22]
	GD	Girar à direita	Triângulo	[–25 –15 –5]
	GDP	Girar à direita pouco	Triângulo	[642]
	ZE	Zero	Triângulo	[–3 0 3]
	GEP	Girar à esquerda pouco	Triângulo	[2 4 6]
	GE	Girar à esquerda	Triângulo	[5 15 25]
	GEM	Girar à esquerda muito	Trapézio	[22 25 30 30]

Tabela 4.9: Detalhamento das características dos conjuntos nebulosos modificados

Adicionalmente, a base de conhecimento do controlador também sofreu alterações, com a redefinição de quatro regras do ambiente. As células sombreadas da tabela a 4.9 mostram os novos pares ordenados delineados para as saídas $\theta \in \Delta$.

dist _o	MH	Н	РН	ZE	РАН	AH	MAH
MP	(GEM, ZE)	(GEM, ZE)	(GEM, ZE)	(GE, ZE)	(GE, ZE)	(GE, ZE)	(GDM, ZE)
Р	(GEM, ACP)	(GEM, NOR)	(GEM, NOR)	(GE, NOR)	(GE, NOR)	(GE, NOR)	(GDM, ACP)
MD	(GEM, NOR)	(GE, AC)	(GEP, AC)	(GEP, ACM)	(GDP, AC)	(GD, AC)	(GDM, NOR)
L	(GEM, AC)	(GE, AC)	(GEP, AC)	(GDP, ACM)	(GDP, AC)	(GD, AC)	(GDM, AC)
ML	(GEM, AC)	(GE, AC)	(GEP, AC)	(GDP, ACM)	(GDP, AC)	(GD, AC)	(GDM, AC)

Tabela 4.10: Modificações na base de conhecimento para a nova configuração do sistema

O controle do número máximo de 500 movimentações para a realização da trajetória também foi aplicado neste experimento.

A figura 4.25 (a) mostra um exemplo de um percurso completo realizado pelo robô com as novas diretivas. O ponto de partida foi em y_r = 28 com ângulo ϕ = 87°. Adicionalmente, são mostrados os gráficos com informações de variação do ângulo ϕ (b) e do deslocamento Δ (c). Vale ressaltar que este caso também foi simulado sem a retirada de regras.



Figura 4.25: Exemplo de uma trajetória do robô com as novas diretivas (a), variação do ângulo com a horizontal (b) e a variação do deslocamento (c)

Em algumas situações de partida, o robô se deslocava em direção à parede esquerda do mundo virtual após a extração de determinadas regras do sistema. Assim, foi definido um espaço de contorno na parede inicial, no qual os tratamentos só poderiam ser realizados até um deslocamento máximo de 10 unidades externas ao plano traçado para o mundo virtual. Caso houvesse a extrapolação de tal limite, a simulação era interrompida, gerando uma falha. A figura 4.25 mostra as áreas aceitáveis para o desenvolvimento do percurso.



Figura 4.25: O mundo virtual e a área de contorno aceitável

4.2.4.1 Características específicas

As seqüências de retiradas de regras seguiram as mesmas características do estudo apresentado na seção 4.2.2.1, com 24 diferentes ordens de extração de conhecimento. Da mesma forma, delineou-se a quantidade de amostras a serem processadas, com 46 disposições de partida.

Combinando as 46 posições iniciais com a remoção de 37 regras e as 24 seqüências de retirada, foi obtido um total de 40848 execuções do simulador, que permitiu gerar os dados para a análise da estabilidade.

4.2.4.2 Simulações e resultados

As informações coletadas para o erro de trajetória e o desvio padrão podem ser visualizadas na figura 4.26 (a) e (b). Os comportamentos de dois procedimentos de extração (S11 - aleatório e S21 - seqüencial por coluna) estão também representados nos mesmos gráficos.



Os resultados obtidos para o desvio padrão ficaram novamente altos quando comparados à média de acerto do controlador. Sendo assim, a análise percentual de estabilidade do controlador foi traçada mais uma vez com referência nos dados do número absoluto de amostras que chegava ao objetivo, que pode ser observado na figura a seguir.



Figura 4.27: Média final da quantidade de amostras que atingia o objetivo

Para a retirada de até 7 regras do sistema, 76,09% de todas as amostras conseguiam alcançar a parede direita do mundo virtual. Após a extração de 13 regras, aproximadamente 50% das amostras ainda eram bem sucedidas.

Não houve ocorrência de simulações que foram limitadas pela restrição da quantidade máxima de 500 movimentações no desenvolvimento do percurso, ou seja, o controlador não ficou demasiadamente envolvido nos tratamentos de colisão.
A figura 4.28 mostra as seguintes seqüências de retirada de regras: periféricas / centrais (a) e centrais / periféricas (b), seguindo as mesmas ordens de extração apresentadas no experimento da seção 4.2.2.



Figura 4.28: Quantidade de amostras que atingia o objetivo segundo critérios específicos de remoção

O grau de sucesso do gráfico (a) é maior que o de (b). Evidenciou-se que na aplicação dos outros critérios de remoção não houve um comportamento tão abrupto na queda do desempenho quando apenas 7 regras haviam sido extraídas, como os resultados apresentados na figura 4.28 (b). Assim, tal análise aponta que para esta configuração específica, existe uma suposta importância do conjunto das regras centrais.

É importante salientar que, a relevância de um conjunto de regras deve levar em consideração o contexto presente em um determinado experimento, identificando as quantidades de remoções já realizadas, a ordem desta extração, o caráter inibitório (ou positivo) das regras e as configurações específicas do ambiente, como, por exemplo, o número e o posicionamento dos obstáculos.

A otimização das configurações permitiu tratar de forma mais adaptável a movimentação do robô pelo mundo virtual. Os dados comparativos que atestam tal melhoria são descritos no estudo subseqüente.

4.2.5 Comparativo geral de desempenho entre as diferentes configurações do sistema

Esta análise comparativa foi desenvolvida para identificar as mudanças ocorridas no comportamento do controlador junto às modificações aplicadas no modelo (MORATORI *et col*, 2005d), (MORATORI *et col*, 2005e). Foram utilizadas como referência todas as delimitações traçadas para os estudos apresentados anteriormente. No primeiro momento, somente o ângulo direção era tratado; subseqüentemente, agregou-se o controle da velocidade e, por fim, a aplicação de uma otimização para permitir a realização de deslocamentos mais adaptáveis. Assim, foi possível observar um comportamento mais amplo do sistema frente a diferentes condições.

O desempenho do modelo foi verificado no contexto da análise de estabilidade frente às sucessivas remoções de regras para as diferentes configurações delineadas, observando os fatores de tempo de execução e número de passos produzidos nas trajetórias.

A seguir são mostrados exemplos de percursos efetuados pelo robô para as diferentes abordagens do sistema. Adicionalmente, identificam-se as informações da variação do ângulo com a horizontal e do tamanho do deslocamento. Para todos os casos definiu-se o ponto de partida em $y_r = 45$ com o ângulo $\phi = -50^{\circ}$. Estas simulações também foram realizadas sem extrair regras da base de conhecimento.



Figura 4.29: Exemplo de uma trajetória do robô na primeira abordagem (a), variação do ângulo com a horizontal (b) e a variação do deslocamento (c)



Figura 4.30: Exemplo de uma trajetória do robô na segunda abordagem (a), variação do ângulo com a horizontal (b) e a variação do deslocamento (c)



Figura 4.31: Exemplo de uma trajetória do robô na terceira abordagem (a), variação do ângulo com a horizontal (b) e a variação do deslocamento (c)

As figuras 4.29, 4.30 e 4.31 mostram os comportamentos apresentadas pelo robô nas diferentes abordagens aplicadas. Os resultados identificam um tratamento mais adequado das informações do ambiente para as sucessivas delimitações do controlador nebuloso. Foi observado que a quantidade de passos realizados sofria decréscimos a cada nova configuração, como mostra os dados dos gráficos (b) das figuras 4.29, 4.30 e 4.31. Notou-se também que as tomadas de decisão também foram mais eficientes, pois as trajetórias desempenhadas foram se tornando cada vez mais simples, como ilustrado nos gráficos (a) de cada uma das figuras anteriores. Além disso, as variações do tamanho do passo também sofreram melhorias, como mostram os dados de (c), ou seja, no primeiro caso o deslocamento era constante, levando a uma longa trajetória; no segundo a velocidade era grande na maioria das iterações, delimitando movimentações curtas somente quando havia perigo de colisão; no terceiro, quase não ocorreu momentos de risco e a velocidade

ficou alta pra quase toda a simulação. Desta forma, as sucessivas evoluções apresentadas pelo controlador foram verificadas.

4.2.5.1 Características específicas

Foram mantidas as características das disposições iniciais das amostras, a limitação de um número máximo de movimentações e as seqüências de extração de regras definidas no estudo de estabilidade da seção anterior. O estabelecimento de uma zona de contorno na parede direita do mundo virtual não havia sido configurado para os outros experimentos. Desta forma, aplicou-se tal controle nas diferentes abordagens e todas as simulações foram reprocessadas, permitindo assim gerar dados para uma efetiva comparação.

Combinando as 46 posições de saída com a retirada de 37 regras e as 24 diferentes seqüências de retirada de conhecimento nos três experimentos, obteve-se um total de 122544 execuções do simulador, que permitiu coletar dados da estabilidade e do desempenho do controlador.

Os mesmos hardwares descritos anteriormente foram utilizados.

4.2.5.2 Simulações e resultados

Os dados comparativos de estabilidade e desempenho do controlador seguiram as mesmas diretivas do estudo da seção 4.2.5.

A figura 4.32 identifica a quantidade absoluta de amostras que chegava ao objetivo para as três abordagens definidas, sendo a linha tracejada para a

configuração do passo constante, a pontilhada para o do deslocamento variável e a contínua para o otimizado.



Figura 4.32: Quantidade de amostras que atingiram o objetivo final para as três configurações do controlador

A abordagem otimizada conseguiu obter uma melhoria em relação à configuração do passo variável para a maior parte do processo de remoção de regras do sistema. Porém, a curva que representa a aplicação do deslocamento constante continuou obtendo uma melhor atuação para o fator estabilidade, apesar das diferenças coletadas serem relativamente pequenas.

O gráfico anterior mostra uma distribuição regular dos dados nas diferentes modalidades testadas, definindo um comportamento geral do controlador frente às sucessivas extrações de conhecimento. Assim, numa visão mais generalista, observou-se que após a retirada de 13 regras, aproximadamente 40% das amostras ainda atingiam a meta.

Na avaliação do desempenho foram analisados os aspectos tempo de processamento e quantidade de passos efetuados nos percursos, colhendo informações gerais e específicas das amostras.

No estudo geral, registraram-se para cada abordagem os seguintes dados: (1) a média do número de passos realizados em todas as simulações; (2) a média do tempo de processamento para cada amostra; (3) a média do tempo gasto para simular todas as amostras em cada um dos critérios de remoção de regras, (4) o tempo total consumido para efetuar todas as simulações e (5) a quantidade de amostras estagnadas durante a realização de suas trajetórias.



A figura 4.33 apresenta, simultaneamente, a média do número de passos realizados (a) e o tempo de processamento de cada amostra (b) para as três abordagens do modelo. Os valores obtidos mostram a evolução destes aspectos nas diferentes configurações aplicadas nos experimentos. Expressivamente, houve uma melhoria na transição da característica do deslocamento constante (linha tracejada) para o variável (linha pontilhada). Ocorrendo, posteriormente, um novo ganho quando o sistema foi otimizado.

A tabela 4.11 mostra as outras características gerais coletadas, que também refletem o progresso no desempenho do sistema para os fatores tempo e quantidade de amostras estagnadas.

	Característica	Passo constante	Passo variável	Passo variável otimizado
Tempo (segundos)	Média de tempo gasto para simular todas as amostras em cada um dos 24 critérios de remoção de regras	375,8	214,1	171,8
	Tempo total de execução para todas as simulações	9018,8	5139,5	4124,1
Amostras (unidades)	Quantidade de amostras que estagnaram no percurso pela limitação do número máximo de 500 movimentações	69	53	Ausente

Tabela 4.11: Análise geral das amostras

Na análise específica das amostras bem sucedidas foram observadas as seguintes informações: (1) o número médio de passos realizados para atingir o objetivo e seus respectivos menores e maiores valores nas simulações; e (2) a média de tempo gasto para simular uma única amostra, com sua menor e maior referência.

	Característica	Passo constante	Passo variável	Passo variável otimizado
Passos (unidades)	Quantidade média de deslocamentos realizados pelas amostras	155	79	65
	Menor quantidade de movimentos realizados pelo controlador nas simulações do experimento	143	36	35
	Maior quantidade de movimentos realizados pelo controlador nas simulações do experimento	498	437	298
Tempo (segundos)	Média de tempo gasto para simular uma única amostra no experimento	0,34	0,18	0,15
	Menor tempo gasto por uma amostra	0,18	0,05	0,05
	Maior tempo gasto por uma amostra	1,88	0,65	0,67

Tabela 4.12: Análise das amostras que atingiram a meta

Pelos resultados encontrados foi possível observar consecutivas melhorias no desempenho do sistema, para todas as características observadas. Os dados indicam que a otimização foi efetiva e que um controle mais adequado da velocidade pode ser aplicado sem comprometer a eficiência do modelo estudado.

4.3 CONSIDERAÇÕES PARCIAIS

O objetivo deste Capítulo foi fazer um estudo sobre a estabilidade e a robustez do controlador nebuloso em diferentes situações, baseado no êxito de cumprir a tarefa de guiar um robô simulado simples através de um mundo virtual com obstáculos. Foram desenvolvidos cinco experimentos para avaliar o comportamento geral do sistema, formulando modelagens adicionais e análises comparativas. No total foram processadas 306656 simulações para coletar as informações sobre o modelo.

Nos testes realizados identificou-se que as regras centrais e periféricas possuíam certa importância para dois dos estudos particulares aplicados. Porém, apontou-se que a relevância de um conjunto de regras deve levar em consideração o contexto presente em cada caso, identificando as quantidades de remoções já realizadas, a ordem desta extração, o caráter inibitório (ou positivo) das regras e as configurações específicas dos parâmetros, como, por exemplo, o número e o posicionamento dos obstáculos.

As abordagens definidas mostraram também que é possível incrementar um controle de velocidade no modelo, sem comprometer a sua eficiência. Os tratamentos mais adequados trouxeram resultados mais satisfatórios, podendo futuramente ser transponível para um projeto de robótica.

O controlador nas diferentes modalidades continuava a atingir sua meta mesmo após a retirada de quase 20% das regras da base de conhecimento do módulo de inferência, permitindo que aproximadamente 77% das amostras ainda atingissem a parede direita do mundo virtual. Desta forma, aponta-se que um especialista, utilizando recursos da lógica nebulosa, não necessita de todas as informações do ambiente para construir um modelo eficiente, que realize de maneira satisfatória tarefas de controle para situações reais.

CAPÍTULO 5

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

5.1 CONSIDERAÇÕES FINAIS

A lógica nebulosa constitui uma poderosa ferramenta para o desenvolvimento de sistemas que necessitam controlar processos sofisticados. A sua aplicação tem ganhado cada vez mais espaço tanto em estudos científicos quanto em produtos comerciais, automatizando controles de processo e auxiliando procedimentos de tomadas de decisão.

As estratégias de controle nebulosas são realizadas a partir da experiência do projetista (especialista), ao invés de trabalhar com modelos matemáticos complexos. Portanto, a implementação constitui um processo simples e rápido, pois as características e inferências delimitadas seguem um padrão lingüístico similar às afirmações utilizadas pelos seres humanos em seu dia-a-dia. A estrutura de software é transparente e compreensível, garantindo, desta forma, uma fácil manutenção. Os módulos nebulosos permitem agregar uma grande quantidade de entradas na base de conhecimentos, porém, somente são ativadas as inferências relevantes a cada condição específica, levando, assim, a um baixo custo computacional.

O sistema desenvolvido para este trabalho foi semelhante ao modelo clássico do caminhão (KOSKO, 1992), delimitando um controlador do tipo Mamdani. As tomadas de decisão inferidas foram satisfatórias, pois permitiram reproduzir adequadamente uma situação real. Os operadores de interseção e de união, utilizados para manipular os conjuntos, e o método de "centro de gravidade", aplicado para gerar as ações de controle, foram efetivos no tratamento da tarefa de controle especificada. Vários testes foram realizados para validar todas as configurações definidas e êxitos foram alcançados nas situações observadas. A implementação foi realizada utilizando o software MATLAB ® 7.0 em uma máquina equipada com um processador Intel® Pentium® M 710 (1.40 GHz) com 512 MB de memória DDR SDRAM (2 x 256).

Entretanto, o modelo criado apresentou uma limitação quando o robô estava inicialmente encostado nas paredes superiores e inferiores. Neste caso, uma inevitável colisão ocorria, pois as ações inferidas pelo controlador não giravam um tamanho de ângulo suficientemente grande para permitir o desvio necessário, além da característica do robô de não girar em torno do próprio eixo sem se movimentar. Estas situações geravam configurações iniciais inoportunas, assim, foram desconsideradas no levantamento de dados deste trabalho.

Foram desenvolvidos cinco experimentos para avaliar a robustez e a estabilidade geral do sistema, formulando modelagens adicionais e análises comparativas de desempenho. No total foram processadas 306656 simulações para coletar as informações deste estudo.

Nos testes realizados identificou-se que as regras centrais e periféricas possuíam certa importância para dois dos estudos particulares aplicados. Porém, apontou-se que a relevância de um conjunto de regras deve levar em consideração o contexto presente em cada simulação, identificando as quantidades de remoções já realizadas, a ordem desta extração, o caráter inibitório (ou positivo) das regras e as configurações específicas dos parâmetros, como, por exemplo, o número e o posicionamento dos obstáculos.

Os experimentos mostraram que também é possível incrementar tratamentos de outros parâmetros (como a velocidade) pelo módulo nebuloso, trazendo melhorias no desempenho geral do modelo, sem comprometer sua estabilidade e eficiência.

O controlador nas diferentes modalidades continuava a atingir sua meta mesmo após a retirada de quase 20% das regras da base de conhecimento do módulo de inferência, permitindo que aproximadamente 77% das amostras ainda atingissem a parede direita do mundo virtual. Desta forma, aponta-se a robustez e a estabilidade dos sistemas nebulosos, identificando que um especialista não necessariamente precisa de todas as informações do ambiente para construir um modelo eficiente, que realize de maneira satisfatória tarefas de controle para situações reais.

Além da simplicidade característica, sugere-se, portanto, a confiabilidade de funcionamento dos sistemas nebulosos de controle, pois êxitos continuam a ser obtidos mesmo sob condições adversas.

5.2 TRABALHOS FUTUROS

O contexto apresentado para análise de estabilidade e robustez dos controladores nebulosos poderia ser estendido, aplicando não só a extração de regras da base de conhecimento, mas também, a modificação das inferências presentes por valores errôneos. Esta delimitação permitiria observar o comportamento do modelo frente a uma situação de mau planejamento ou de equívocos cometidos pelo projetista.

Uma sistemática para avaliar a estabilidade poderia ser construída, de forma a sugerir ao desenvolvedor diretivas sobre o sistema nebuloso, realizando indicações

sobre possíveis redundâncias, inferências mais utilizadas, quantidade de regras necessárias e situações onde se observa falta de conhecimento do especialista.

O controlador desenvolvido para este trabalho também poderia ser implementado utilizando as técnicas do modelo de interpolação proposto por Takagi-Sugeno. Estes sistemas *neuro-fuzzy* também são construídos de forma simples, pois o módulo de inferência nebuloso é automaticamente gerado a partir de uma base de dados, que contém os valores dos possíveis parâmetros de entrada e as suas respectivas saídas esperadas. Desta forma, uma metodologia conhecida como ANFIS (*Adaptive Neuro-fuzzy Inference System*) treinaria este conjunto de dados até que o sistema híbrido desempenhasse satisfatoriamente a sua tarefa de controle.

Uma análise comparativa de estabilidade e robustez entre os controladores nebulosos e os híbridos permitiria delimitar as melhores alternativas no desenvolvimento de soluções para condições específicas. Um estudo preliminar está sendo realizado (MORATORI *et col*, 2005f), (MORATORI, CRUZ e LIMA, 2006), mas ainda são necessários maiores esforços de pesquisa e investigações.

Além disso, a transposição destes projetos para a área de robótica também possibilitaria uma análise real dos comportamentos apresentados, permitindo estender considerações para o uso prático destes modelos.

ALTROCK, C. Fuzzy Logic and NeuroFuzzy Applications Explained. Prentice-Hall, Englewood Cliffs, 1995.

ARACIL, J. e GORDILLO, F. Describing function method for stability analysis of PD and PI fuzzy controllers. Fuzzy Sets and Systems 143 (2), 233-249, 2004.

CASTRO, J. *Fuzzy logic controllers are universal approximators*. IEEE Transactions on Systems, Man and Cybernetics, 25(4):629 – 635., Abril 1995.

CHIU, S. e CHAND, S. *Adaptative Traffic Signal Control Using Fuzzy Logic*. The Institute of Eletrical and Eletronics Engineers, Inc., New York, 1994.

CONWAY, D., SPERLING, R., FOLTA, D., RICHON, K. e DEFAZIO, R. Automated Maneuver Planning Using a Fuzzy Logic Algorithm. In: Proc. 1994 GSFC Flight Mechanics/Estimation Theory Symposium. p. 49-62, 1994.

COX, E. The Fuzzy Systems Handbook. Massachussets: Academic Press, 1994.

CRUZ, A. Lógica Nebulosa. Disponível em http://equipe.nce.ufrj.br/adriano/ fuzzy/apostila.pdf. Arquivo consultado em agosto de 2004.

CRUZ, A. J. O., DEMASI, P., LIMA, J. C. M. Jogos Educativos Inteligentes: Ferramentas de Suporte **In:** Simpósio Brasileiro de Informática na Educação: Minicursos.1 ed. Rio de Janeiro : SBC, 2003, v.1, p. 93-126.

D'AMORE, R., SAOTOME, O. e KIENITZ, K. Controlador nebuloso com detecção de regras ativas. *Anais do 3º Simpósio Brasileiro de Automação Inteligente,* p. 313–318, Setembro 1997.

DEMASI, P. *Estratégias Adaptativas e Evolutivas em Tempo Real para Jogos Eletrônicos*. Universidade Federal do Rio de Janeiro, 2003. (Dissertação de Mestrado)

DRIANKOV, D., HELLENDOORN, H. e RENFRANK, M. An Introduction to Fuzzy Control. Springer-Verlag, 1993.

DUBOIS D. e WIDROW B. *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, Orlando, 1980.

DUBOIS, D., PRADE, H., e YAGER, R. *Readings in Fuzzy Sets for Intelligent Systems*, Morgan Kaufmann, 1993.

DUTTA, S. Fuzzy logic applications: Technological and strategic issues. *IEEE Transactions on Engineering Management*, 40(3):237 – 254, Agosto 1993.

EINSTEIN, A. Ideas and Opinions. New York: Wings Books, 1954.

GOMIDE, F., GUDWIN, R. e TANSCHEIT, R. *Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações.* Sixth International Fuzzy Systems Association World Congress/ Tutorials - IFSA95, pages 01 – 38., July 1995.

GUERRA, R. *Projeto e simulação do controle de atitude autônomo de satélites usando lógica nebulosa*. Instituto Nacional de Pesquisas Espaciais - INPE,São José dos Campos - SP, Março 1998. (Dissertação de Mestrado)

GUERRA, R., SANDRI, S. e SOUZA, M. *Dynamics and design of autonomous attitude control of a satellite using fuzzy logic. Anais do COBEM*'97, (COB 1338), Dezembro 1997.

KARR, C. e GENTRY, E. *Fuzzy control of pH using genetic algorithms.* IEEE Transactions on Fuzzy Systems, 1(1):46 – 53, Fevereiro 1993.

KLIR, G. e FOLGER, T. *Fuzzy Sets, Uncertainty, and Information.* Prentice Hall, Englewood Cliffs - New Jersey, 1988.

KOSKO, B. e KONG, S. Comparison of Fuzzy and Neural Truck Backer-Upper Control Systems. Prentice Hall, Englewood Cliffs, 1992.

KOSKO, B. Fuzzy Engineering, Prentice Hall, Upper Saddle River, 1997.

KOSKO, B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. New Jersey: Prentice Hall, 1992.

LARSEN, P. Industrial Applications of Fuzzy Logic Control, International Journal of Man-Machine Studies, 12(1), 1980, pp. 3–10.

LEE, C. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I & Part II*, IEEE Transactions on Systems, Man and Cybernetics, 20(2), pp. 404-435, 1990.

LUKASIEWICZ, J. O *logice trójwartosciowej (*On three-valued logic), Ruch Filozoficzny, 5, 170-171, 1920. In: Borkowski, L. (Ed.) Selected Works (S.1.): North Holland, 1970. p 87-88

MAMDANI, E. *Application of Fuzzy Algoritms for Control of Simple Dynamic Plant.* Proceedings of the IEEE Control and Science, Vol. 121, 298-316, 1974.

MAMDANI, E. *Application of Fuzzy Logic to Approximate Reasoning using Linguistc Synthesis*, IEEE Transactions on Computers, vol. C-26, n° 12, pp. 1182- 1191, 1997.

MENDEL, J. *Fuzzy Logic Systems for Engineering: A Tutorial*, volume 83. Proceedings of the IEEE, Março, 1995.

MORATORI, P., PEDRO, M., FERREIRA, E., CRUZ, A. Analysis of Sensitivity and Robustness to a Fuzzy System developed to guide a Simulated Robot through a Virtual World with Obstacles. In Proceedings of the 2004 IEEE 4th International Conference on Intelligent Systems Design and Applications (ISDA 2004), pp. 283-286, 2004.

MORATORI, P., CRUZ, A., FERREIRA, E., PEDRO, M., MANHAES, M., ANDRADE, L., LIMA, C. *Analysis of the performance of a Fuzzy Controller developed to guide a simulated robot.* In Proceedings of the 2005 IEEE 3rd International Conference on Computational Cybernetics (ICCC 2005), 2005a.

MORATORI, P., CRUZ, A., PEDRO, M., FERREIRA, E., MANHAES, M., ANDRADE, L., LIMA, C. *Analy*sis of stability of a Fuzzy Control System developed to control a simulated robot. In Proceedings of the 2005 IEEE 14th International Conference on Fuzzy Systems (FUZZ-IEEE 2005), pp. 726-730, 2005b.

MORATORI, P., CRUZ, A., FERREIRA, E., PEDRO, M., MANHAES, M., ANDRADE, L., LIMA, C., MORATORI, R. *Análise de Estabilidade e Robustez de um Sistema de Controle Fuzzy* Otimizado desenvolvido para guiar um robô simulado. Anais do

XXXII SEMISH: Seminário Integrado de Software e Hardware, São Leopoldo, RS, Brasil, pp. 1704-1715, 2005c.

MORATORI, P., CRUZ, A., MANHAES, M., FERREIRA, E., PEDRO, M., LIMA, C., ANDRADE, L. Analysis of the Performance of Different Fuzzy System Controllers Developed to Guide a Simulated Robot. In Proceedings of MICAI 2005: Fourth Mexican International Conference on Artificial Intelligence, 2005d.

MORATORI, P., CRUZ, A., MANHAES, M., FERREIRA, E., PEDRO, M., LIMA, C., ANDRADE, L. *Analysis of the Performance of Different Fuzzy System Controllers*. Lecture Notes in Artificial Intelligence / Lecture Notes in Computer Science, vol. 3789, pp. 1113-1123, Springer-Verlag, 2005e.

MORATORI, P., CRUZ, A., ANDRADE, L., M., LIMA, C. *Comparing Sensitivity and Robustness of Fuzzy and Neuro-fuzzy Controllers*. In Proceedings of HIS 2005: The Fifth International Conference on Hybrid Intelligent Systems, Rio de Janeiro, pp. 528-530, 2005f.

MORATORI P., CRUZ, A. e LIMA, C. Comparing Robustness and Performance of Hybrid and Non-Hybrid Fuzzy Controllers aimed to guide a simulated robot. IEEE 15th International Conference on Fuzzy Systems (FUZZ-IEEE 2006), 2006. (em avaliação)

NURCAHYO, G., SHAMSUDDIN, S., ALIAS, R. e SAP, M. "Selection of Defuzzification Method to Obtain Crisp Value for Representing Uncertain Data in a Modified Sweep Algorithm", JCS&T, 3, 22-28, 2003.

PATEL, A. Simplest Fuzzy PI Controllers under Various Defuzzification Methods. International Journal of Computational Cognition (ISSN 1542-5908), Volume 3, Number 1, March 2005, pages 21-34,

PRECUP, R. e PREITL, S. *On a Class of Control Systems with Takagi-Sugeno PI-Fuzzy Controllers*. In Proceedings of the 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, 2005.

PRECUP, R., DOBOLI, S. e PREITL, S. *Stability analysis and development of a class of fuzzy control systems*. Engineering Applications of Artificial Intelligence, vol. 13, 237-247, 2000.

ROSENWASSER, E. e YUSUPOV, R. *Sensitivity of Automatic Control Systems.* CRC Press, Boca Raton, FL, 2000.

RUSSEL, B. Vagueness. Australian Journal of Philosofy, v.1, 1923.

SANDRI, S. e CORREA, C. *Lógica Nebulosa*. V Escola de Rede Neurais. pp. c073c090. São José dos Campos. São Paulo, 1999.

SANDRI, S. *Introdução à lógica "fuzzy".* Minicurso 2, III- SBAI - Simpósio Brasileiro de Automação Inteligente, Vitória ES, 1997.

SHAW, I. e SIMÕES M. *Controle e Modelagem Fuzzy*. FAPESP, Editora Edgard Blücher LTDA, São Paulo, Brasil, 1999.

SUGENO, M. On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents. IEEE Trans. Fuzzy Systems, vol. 7, 201–224, 1999.

TAKAGI, T. e SUGENO, M. *Fuzzy identification of systems and its applications to modeling and control.* IEEE Trans. on Systems, Man and Cybernetics, 15, 1985, 116–132.

TANSCHEIT, R. Sistemas Fuzzy. Disponível em www.ica.ele.puc-rio.br/cursos/ download/LN-Sistemas%20Fuzzy.pdf. Arquivo consultado em maio de 2005.

TSUKAMOTO, T. *An approach to fuzzy reasoning method*. In: Gupta, M., Regade, R., Yager, R. (Eds.), Advances in Fuzzy Set Theory and Applications, North-Holland, Amsterdam, 1979.

WEIL, H., PROBST, G. e GRAF, F. *Fuzzy Expert System for Automatic Transmission Control*. The Institute of Eletrical and Eletronics Enginees, Inc., New York, 1994.

WOODARD, M. *Fuzzy Open-Loop Attitude Control for the FAST Spacecraft*, San Diego -CA. Proceedings of the NASA AIAA, Guidance, Navigation and Control Conference, Julho 1996.

YEN, J. e LANGARI, R. *Fuzzy Logic: Intelligence, Control and Information*, Prentice Hall, Englewod Cliffs, 1999.

ZADEH, L. *A Theory of Approximate Reasoning*. In: Hayes, J., Mitchie, D. e Mikulich, L. (Eds.) Machine Intelligence, Vol. 9, WILEY, New York, 1979, pp. 149–94.

ZADEH, L. *Fuzzy sets as a basis for a theory of possibility.* Fuzzy Sets and Systems, 1:3 – 28, 1978.

ZADEH, L. Fuzzy sets. Fuzzy Sets, Information and Control, 8:338 – 353., 1965.

ZHANG, J., LI, R. e ZHANG, P. Stability analysis and systematic design of fuzzy control systems. Fuzzy Sets and Systems (120), 65-72, 2001.