

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**APRENDIZADO SUPERVISIONADO DE SISTEMAS DE  
INFERÊNCIA *FUZZY* APLICADOS EM VEÍCULOS INTELIGENTES**

Luiz Lenarth Gabriel Vermaas

*Tese submetida ao Programa de Pós-graduação em  
Engenharia Elétrica da UNIFEI como requisito para obtenção  
do título de Doutor em Ciências em Engenharia Elétrica na  
área de Automação e Sistemas Elétricos Industriais.*

Orientador: Prof. Dr. Leonardo de Mello Honório

Co-orientador: Prof. Dr. Luiz Edival de Souza

Itajubá 2010

# UNIVERSIDADE FEDERAL DE ITAJUBÁ

## PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

### APRENDIZADO SUPERVISIONADO DE SISTEMAS DE INFERÊNCIA *FUZZY* APLICADOS EM VEÍCULOS INTELIGENTES

Luiz Lenarth Gabriel Vermaas

#### **Comissão Examinadora:**

Prof. Dr. Takashi Yoneyama – ITA

Prof. Dr. Guilherme Augusto Silva Pereira – UFMG

Prof. Dr. André Luis Marques Marcato – UFJF

Prof. Dr. Luiz Eduardo Borges da Silva – UNIFEI

Prof. Dr. Benedito Isaías Lima – UNIFEI

Prof. Dr. Luiz Edival de Souza – UNIFEI

Prof. Dr. Leonardo de Mello Honório – UNIFEI

#### **Presidente da Comissão Examinadora:**

Prof. Dr. Carlos Henrique Valério de Moraes

Itajubá 2010

À minha esposa Andréa,  
minhas filhas Yasmim e Melissa,  
com muito amor.

## AGRADECIMENTOS

À inteligência suprema, causa primária de todas as coisas: Deus.

Ao Prof. Leonardo de Mello Honório, mais do que um orientador desta tese de doutorado, um grande amigo. Pelo apoio incondicional, pelas brilhantes idéias, pela confiança depositada em mim, pelas experiências adquiridas nos momentos de alegria e dificuldade que passamos durante o desenvolvimento deste trabalho, por me orientar e trabalhar na concretização desta tese.

Ao Prof. Luiz Edival de Souza, co-orientador desta tese, por sua amizade, seu apoio, suas sugestões e incentivo para concluir este trabalho.

Aos meus pais Luiz Vermaas (*in memorian*) e Dynéa Maria Gabriel Vermaas (*in memorian*), exemplos de dignidade, bondade, honestidade, dedicação à família, trabalho, amor ao próximo. Pela educação e amor que me proporcionaram, por seus exemplos de vida...

À minha esposa amada Andréa Costa Vermaas e minhas filhas queridas Yasmim e Melissa, por estarem presentes em todos os momentos, pelo carinho, pela paciência, pela atenção, pelo incentivo, pela compreensão, pelas alegrias que me dão, pelo amor que tenho de vocês.

Aos meus irmãos Luciano Gabriel Vermaas e Dulcenéa Gabriel Vermaas pela presença mesmo que distante, pelos conselhos, pelo incentivo, pela amizade e carinho que ajudam a superar obstáculos.

Aos amigos Carlão, Kleber, Michael, Muriel, Luciano, Guilherme, Rodrigo e Daniele do Grupo de Automação e Tecnologia da Informação - GATI e participantes do Projeto Driving4u.

À Universidade Federal de Itajubá e ao Instituto de Engenharia de Sistemas e Tecnologias da Informação – IESTI.

Aos colegas professores do IESTI pelo suporte durante meu afastamento para capacitação.

À Financiadora de Estudos e Projetos – FINEP e à Fundação de Amparo à Pesquisa de Minas Gerais – FAPEMIG pelo apoio financeiro ao Projeto Driving4u.

**"Aprender é a única coisa de que a mente nunca se cansa,  
nunca tem medo e nunca se arrepende."**

Leonardo da Vinci

## Resumo

Esta tese trata do desenvolvimento de um algoritmo para aprendizado supervisionado capaz de extrair conhecimentos provenientes de tarefas de manobras veiculares realizadas por um motorista humano. Como o objetivo final é determinar o conhecimento do motorista de forma compreensível, este será representado por Sistema de Inferência *Fuzzy* (FIS) baseado no modelo Mamdani, por representar o problema de forma simples por meio da linguagem natural humana.

O desenvolvimento de tal sistema é, na verdade, um problema de otimização inteira mista, onde se deseja mapear o relacionamento entre as entradas provenientes do sistema de aquisição de dados do veículo com as respostas fornecidas por um humano.

A geração deste Sistema *Fuzzy* implica na geração de uma estrutura de regras *fuzzy* e funções de pertinência que representem adequadamente o conjunto de exemplos de treinamento. Assim, o problema é dividido em duas partes: uma responsável pelo aprendizado das regras e a outra responsável pela otimização das funções de pertinência. A implementação do algoritmo para solucionar este problema, aplica os conceitos de Sistema Imunológico Artificial baseado em Gradiente (GbAIS) com duas populações distintas de anticorpos: uma para aprendizagem da estrutura de regras e outra para otimização das funções de pertinência. Por meio do processo de co-evolução das duas populações é possível: trocar informações entre elas, uma vez que as mesmas são interdependentes; evitar o surgimento de ótimos locais; e aumentar o *fitness* do sistema gerado.

Para validar esta proposta, o FIS gerado é utilizado em uma aplicação de veículos inteligentes. O algoritmo foi testado inicialmente em um ambiente de simulação 3D e posteriormente em um veículo de passeio real. Os resultados obtidos para o problema de estacionamento em vaga paralela e navegação em um circuito com *waypoints* comprovaram a eficácia do algoritmo proposto.

As principais contribuições desta tese são: 1) a utilização de técnicas de aprendizado supervisionado para geração automática de sistemas de controle de alto nível em veículos inteligentes, por ser um tema pouco pesquisado neste tipo de aplicação; 2) a proposta da Tabela de Regras Potencias (TRP) para pré-seleção de regras candidatas, conduzindo a redução do espaço de busca; e 3) a aplicação de CGbAIS, uma nova técnica baseada em população.

## **Abstract**

The goal of this thesis is the development of a supervised learning algorithm that extracts driver's knowledge from a data base about the vehicle itself and the environment. As the final goal is to achieve a comprehensible model, it will be represented by a Fuzzy Inference System (FIS) based on Mamdani type, which represents the problem using the natural human language.

The development of this system is a mixed-integer optimization problem. The goal is to map the relationship between inputs from the vehicle data acquisition system and the outputs from the driver's action.

The generation of this Fuzzy System implies that fuzzy rules and membership functions must be created to represent the relationship between data base inputs and outputs correctly. Thus, the problem is divided into two parts: one responsible for learning fuzzy rules and the other one for optimizing membership functions. The algorithm used to solve these problems applies concepts of Gradient-based Artificial Immune Systems (GbAIS) with two different populations of antibodies: one for rule structure learning and the other one for membership functions optimization. By using a co-evolution process of both populations, it is possible to exchange information between them, as they depend on each other, avoid local optima and increase the generated system fitness.

In order to validate this proposal, the generated FIS is used in IV applications. The algorithm was initially tested in a 3D simulation environment and then in a real car. The achieved results of a parallel park problem and driving situation proved the efficacy of the proposed algorithm.

The major contributions of this thesis are: 1) the application of supervised learning techniques to automatic generate a high level control system to intelligent vehicles, as this is not very common in the literature; 2) the proposal of a Potential Rule Table technique in order to pre-select candidate rules, which reduces the search space; and 3) the use of CGbAIS, a new technique when compared with other population based algorithm.

# Sumário

<b>Lista de Figuras</b>	i
<b>Lista de Tabelas</b>	ii
<b>Lista de Abreviaturas</b>	iii
<b>Capítulo 1 – Introdução</b>	
1.1 Considerações Iniciais	1
1.2 Estudo de Casos em Veículos Inteligentes	3
1.3 Motivação e Justificativa	8
1.4 Objetivo Geral desta Proposta	10
1.5 Principais Contribuições	12
1.6 Organização do Documento	14
<b>Capítulo 2 – Aprendizado de Sistemas <i>Fuzzy</i></b>	
2.1– Considerações Gerais	15
2.2– Modelo Geral de um Agente de Aprendizado	17
2.3– Paradigmas de Aprendizado	18
2.3.1 - Aprendizado Supervisionado	18
2.3.2 - Aprendizado por Reforço	20
2.3.3 - Aprendizado Não-supervisionado	20
2.4 – Abordagens Michigan e Pittsburgh para Representação de Regras	21
2.5 – Indução de Sistemas <i>Fuzzy</i>	22
2.5.1 – Sistema <i>Fuzzy</i> Tipo Mamdani	23
2.5.1.1 – Modo FATI	25

2.5.1.2 – Modo FITA	25
2.5.1.3 – Exemplo de Aplicação	26
2.5.2 – Sistema Fuzzy Tipo Takagi-Sugeno-Kang	29
2.6 – O Projeto Automático de Controladores <i>Fuzzy</i>	30
<b>Capítulo 3 – Sistemas Imunológicos Artificiais</b>	
3.1 – Considerações Iniciais	33
3.2 – O Sistema Imunológico Artificial	34
3.3 – Principais Algoritmos	35
3.3.1 – CLONALG	35
3.3.2 – Opt-Ainet	38
3.4 – CGbAIS – Sistema Imunológico Artificial baseado em Gradiente e Cluster	39
3.4.1 – Análise do Gradiente em Otimização	39
3.4.2 – Clusterização e Controle de Maturação em Otimização Contínua	43
3.4.3 – O Algoritmo CGbAIS	46
3.4.4 – Problemas de Otimização Discreta	48
<b>Capítulo 4 – Aprendizado Supervisionado Baseado no algoritmo CGbAIS</b>	
4.1 – Considerações Iniciais	52
4.2 – Geração de um FIS a partir de Banco de Dados Numérico	54
4.3 – Aprendizado da Estrutura da Regra <i>Fuzzy</i>	56
4.3.1 Aplicação do Algoritmo CGbAIS	58
4.4 – Otimização das Funções de Pertinência	59
<b>Capítulo 5 – Aplicação e Resultados Obtidos</b>	
5.1 – Considerações Iniciais	60
5.2 – Estacionamento Paralelo: Resultado de Simulação	61
5.3 – Estacionamento Paralelo: Resultados Reais	69
5.4 – Navegação em um Circuito com <i>Waypoints</i> : Resultados de Simulação	71
5.5 - Navegação em um Circuito com <i>Waypoints</i> : Resultados Reais	75
<b>Capítulo 6 – Conclusões</b>	78

6.1 – Conclusões deste trabalho	78
6.2 – Propostas de Continuidade do Trabalho	80
6.3 – Produção Científica	82
<b>Apêndice A – Veículos Inteligentes</b>	<b>83</b>
A.1 – Estado da Arte em Veículos Inteligente	84
A.2 – Arquitetura de Veículos Inteligentes	87
A.3 – Instrumentação Veicular	90
A.3.1 – Introdução	90
A.3.2 - Sensores, Aplicações e Características	91
A.3.3 - Multi-sensores e Fusão	94
A.3.4 - Comunicação Interveicular e Sistema de Transporte Inteligente	97
A.4 – Sistema de Visão Computacional	99
A.4.1 - Conceitos Básicos	100
A.4.2 - Seleção de Casos de Visão Computacional em IV	102
A.5 – Modelo Cinemático do Veículo	103
A.5.1 – Representação da Posição do Veículo	103
A.5.2 – Modelo Cinemático Aplicado	104
A.6 – O Projeto Driving4u	109
<b>Referências</b>	<b>112</b>

# Lista de Figuras

Figura 1 - (a) Planejamento de rota local utilizando Redes Bayesianas [fonte (YAGI et al., 1995)],	6
Figura 2 - (a) detecção de obstáculos {fonte [(KATO et al., 2002)]}, (b) acompanhamento de carro em comboio {fonte [(BISHOP, 2005)]}, e (c) acompanhando veículos, com identificação de semáforos, situação de parar e seguir e evitar colisão {fonte [(BISHOP, 2005)]}	6
Figura 3 - Interação Motorista - Veículo - Ambiente	11
Figura 4 - Modelo Geral de um Agente de Aprendizado	17
Figura 5 - Estrutura Básica do Controlador Fuzzy Mamdani	23
Figura 6 - Funções de pertinência das variáveis consideradas no exemplo	26
Figura 7 - Modo de Operação do Sistema de Inferência Fuzzy	27
Figura 8 - Defuzzificação Modo FATI com estratégia centro de gravidade	28
Figura 9 - Defuzzificação modo FITA	29
Figura 10 - Processo de Hipermutação - Vetor tangente (a,b,c) e Utilizando indivíduos aleatórios (d,e,f)	43
Figura 11- Processo de agrupamento, clusterização e controle de maturação.	45
Figura 12 - Algoritmo CGbAIS	46
Figura 13 - Otimização Combinatória Utilizando o Algoritmo CGbAIS	50
Figura 14 - Processo Co-evolucionário	55
Figura 15 - Divisão do espaço de entradas e saídas em regiões fuzzy	57
Figura 16 - CGbAIS aplicado em otimização discreta	59
Figura 17 - Coordenadas da Função de Pertinência	59
Figura 18 - Variáveis consideradas no problema de estacionamento paralelo	61
Figura 19 - Diagrama de Estados para Estacionamento Paralelo	62
Figura 21 - Saída Original da base de dados (Exemplo de Treinamento), resultado do algoritmo proposto (Exemplo de Teste) e resultado do método Wang & Mendel	65
Figura 22 - Gráfico $X_A \times Y_D$ da Manobra de Estacionamento Paralelo - Simulação	66
Figura 23 - Tela ANFISEDIT do Matlab para treinamento do FIS para baliza.	67
Figura 24 - Estrutura de Regras geradas pelo ANFIS - 27 regras geradas	67
Figura 25 - Funções de Pertinência no ANFIS	68
Figura 26 - Seqüência de manobras no simulador para estacionamento em vaga paralela	68
Figura 27. Exemplos de Treinamento:Estacionamento Paralelo -	69
Figura 28 - Funções de Pertinência para Controlador Fuzzy de Estacionamento	70
Figura 29 - Estacionamento paralelo executado de forma autônoma utilizando o FIS aprendido com o motorista humano.	71
Figura 30. Vetores ângulo do carro, da trajetória e Erro_angular	72
Figura 31 - Trajetória Mapeada e Executada - Resultado de Simulação	74
Figura 32 - Ângulo da roda do veículo durante trajetória executada.	74
Figura 33. Circuito utilizado para treinamento do veículo	75
Figura 34. Funções de Pertinência para Controlador Fuzzy de Navegação	76

Figura 35. Circuito de 800m de extensão utilizado para teste do FIS gerado. ....	76
Figura 36. Comparação das Trajetórias Mapeada e Executada .....	77
Figura 37 - Ângulo da Roda durante Trajetória Executada .....	77
Figura 38 - Boss, veículo vencedor do Darpa Urban Challenge 2007 .....	85
Figura 39 - Stanley, veículo vencedor do Darpa Grand Challenge 2005 .....	86
Figura 40 - Arquitetura Típica de Veículos Inteligentes .....	88
Figura 41 - Comboio e diagrama em blocos da fusão de sensores .....	95
Figura 42 - Diagrama de um Filtro de Kalman para integração INS/GPS .....	96
Figura 43 - Configuração de um Sistema PUA e PVUA .....	97
Figura 46 - Representação da Posição do Veículo.....	104
Figura 47 - - Modelo Cinemático do Veículo .....	105
Figura 48 - Veículo Inteligente - Projeto Driving4u.....	109
Figura 49 - Diagrama de sensores e atuadores no veículo.....	110
Figura 50 - Diagrama de Dados do Veículo.....	111

## Lista de Tabelas

<i>Tabela 1 - Resultado das simulações do CGBAIS e variações.....</i>	<i>48</i>
<i>Tabela 2 – Tabela de Regras Potenciais - TRP .....</i>	<i>58</i>
<i>Tabela 3 – Tabela de Regras Potenciais – Estacionamento Paralelo.....</i>	<i>64</i>
<i>Tabela 5 – Regras obtidas por meio do algoritmo proposto.....</i>	<i>64</i>
<i>Tabela 6 – Regras obtidas para estacionamento paralelo .....</i>	<i>70</i>
<i>Tabela 7 - Regras obtidas para navegação no simulador .....</i>	<i>73</i>
<i>Tabela 8 - Regras obtidas para navegação .....</i>	<i>75</i>
<i>Tabela 9 - Tipos de Sensores, Aplicações e Características.....</i>	<i>92</i>

## Lista de Abreviaturas

Para alguns termos técnicos de língua inglesa, adotou-se a abreviatura do termo em inglês.

Ab:	Anticorpo
ABS:	Sistemas de Frenagem Antitravamento
AF:	Ajuste para Frente
AG:	Algoritmo Genético
AI:	Inteligência Artificial
AIS:	Sistema Imunológico Artificial
AM:	Aprendizado de Máquina
ANN:	Rede Neural Artificial
AT:	Ajuste para Trás
BDI:	Banco de Dados Inicial
CAN:	<i>Controller Area Network</i>
CE:	Carro Estacionado
CGbAIS:	Sistema Imunológico Artificial baseado em Gradiente e Clusterização
CLONALG:	Algoritmo de Seleção Clonal
CNE:	Carro não Estacionado
DARPA:	<i>Defense Advanced Research Projects Agency</i>
DGPS:	Sistema de Posicionamento Global Diferencial
EV:	Entrada na Vaga
FATI:	Primeiro Agrega, depois Infere
FIS:	Sistema de Inferência <i>Fuzzy</i>
FITA:	Primeiro Infere, depois Agrega
FP:	Funções de Pertinência
GA:	Algoritmo Genético
GbAIS:	Sistema Imunológico Artificial baseado em Gradiente
GPS:	Sistema de Posicionamento Global
INS:	Sistema de Navegação Inercial

ITS:	Sistema de Transporte Inteligente
IV:	Veículos Inteligentes
MMD:	Método da Distância MaxMin
Ne:	Número de amostras no banco de dados
NH:	Número de Hipermutações
NM:	Número de Maturações
NR:	Número de Receptor de Edições
PUA	Arquitetura com atualização da posição
PVUA	Arquitetura com atualização da posição e velocidade
TRP:	Tabela de Regras Potenciais
TSK:	Takagi-Sugeno-Kang
VT:	Vetor Tangente
W-LAN:	Rede Local sem fio

# Capítulo 1

## Introdução

### 1.1 – Considerações Iniciais

Veículos Inteligentes (IV) não são mais tecnologias visionárias que poderiam estar presentes em situações reais em um futuro muito distante. Desde aplicações mais simples de auxílio ao motorista, tais como o controle eletrônico de velocidade ou sistemas de frenagem antitravamento (ABS), até as mais complexas, tais como sistemas militares autônomos de alta tecnologia ou sistemas automáticos de estacionamento de veículos, estas tecnologias estão se tornando cada vez mais comuns na atualidade (BISHOP, 2000).

O termo “Veículo Inteligente” (IV) é de certa forma, muito genérico. Veículos podem ser terrestres, aéreos ou aquáticos, assim como a designação inteligente pode ser empregada com diferentes significados (BISHOP, 2005). Nesta proposta o termo IV será utilizado para designar veículos terrestres não-holonômicos (tipo carro de passeio) com capacidade de perceber o ambiente onde ele está, por meio de sensores, e tomar decisões estratégicas autônomas de navegação e de controle de direção.

No meio acadêmico e industrial há muita pesquisa sendo realizada nesta área visando tanto o desenvolvimento de *hardware* quanto de algoritmos baseados em técnicas de Inteligência Artificial (AI). Este crescente interesse pode ser observado por meio dos diversos *workshops*, conferências, publicações e desafios promovidos pelo IEEE, IEE e outras organizações relevantes.

A análise das publicações nesta área mostra que os principais campos de pesquisa em IV são:

1. **Instrumentação de Alta Precisão** – Atualmente, a instrumentação disponível para aplicações automotivas é de alta precisão. Os Sistemas de Posicionamento Global Diferencial (DGPS), por exemplo, atingem precisão sub-métrica (BISHOP, 2005), podendo ser empregados para detectar com precisão a posição do veículo. Esta precisão e a confiabilidade dos dados fornecidos pelos sensores aumentam quando se utiliza técnicas de registro simultâneo e fusão de sensores (LI; LEUNG, 2004; ROGERS, 1996; EL-SHEIMY et al., 2006; STAMPFLE; BECKER, 2005; BEVLY et al., 2006) temas muito pesquisados e em desenvolvimento.
2. **Técnicas de Inteligência Artificial** – Poderosos algoritmos utilizando técnicas de AI são capazes de lidar com dados imprecisos e incompletos, permitindo a modelagem do sistema de uma maneira muito confiável (THRUN et al., 2001, FUJIMORI et al., 1997, KROGH; FENG, 1989; YEN; PFLUGER, 1995; LI et al., 2004). Por meio da utilização destes algoritmos é possível receber dados de diferentes sensores, identificar claramente a situação atual, planejar determinadas decisões para executar uma tarefa, fazer re-planejamento caso uma situação inesperada ocorra, prever certas condições evitando situações desastrosas, entre outras.
3. **Visão Computacional** – Trata-se de umas das áreas mais pesquisadas em AI. A visão computacional integra a utilização de *hardware* de alto desempenho com *software* de tempo real. Pesquisas neste campo (MACCALL; TRIVEDI, 2006; KUTAMI et al., 1990; TSUGAWA, 1994; THORPE et al., 1988) oferecem

um largo espectro de características, tais como, visão *stereo*, movimento, aparência, identificação de bordas, entre outras.

Os resultados já alcançados nesta área somado ao interesse em impulsionar ainda mais as pesquisas sobre IV, incentivaram o Departamento de Defesa dos EUA a criar o *DARPA Grand Challenge*, um desafio que consistiu em desenvolver um veículo totalmente autônomo capaz de navegar 212km em meio ao deserto nos EUA. Na primeira competição, realizada em 2004, nenhum dos inscritos conseguiu completar o percurso. Porém, em 2005, a equipe *Stanford Racing Team*, da Universidade de Stanford, desenvolveu o veículo denominado *Stanley* que completou o percurso em primeiro lugar (DARPA, 2007), recebendo por esta conquista o maior prêmio já distribuído na história da robótica: dois milhões de dólares. Em 2007, a terceira versão do desafio foi lançada, porém tendo como enfoque o meio urbano, conduzindo a novos e diferentes desafios. O *DARPA Urban Challenge* (DARPA, 2008) teve a equipe *Tartan Racing*, de Pittsburgh (Carnegie Mellon), sagrando-se vencedora com o veículo denominado “*Boss*”, e o veículo “*Junior*” de Stanford em segundo.

A análise dos documentos técnicos (DARPA, 2007, 2008; BAKER; DOLAN, 2008; THRUN et al., 2006; OZGUNER et al., 2007) dos participantes das duas competições promovidas pelo *Darpa* em 2005 e 2007, mostra que a utilização de DGPS, sistemas de navegação inercial, câmeras, e sensores laser são pontos comuns entre eles, variando em quantidade, local de instalação, especificações técnicas e fabricante. Um aspecto importante a ser considerado são os sistemas de controle, ou a “*inteligência*” do veículo. Os sistemas de controle de alto nível destes veículos utilizam algoritmos baseados em sistemas de controle clássico ou em técnicas de AI programados por um especialista.

## **1.2 – Estudo de Casos em Veículos Inteligentes**

O objetivo principal de um IV é navegar de forma segura de sua posição inicial até uma determinada localização. Esta não é uma tarefa trivial, mesmo que o ambiente fosse completamente estático e conhecido. Portanto, em aplicações reais, uma série de situações inesperadas pode acontecer e várias ações distintas podem ser necessárias, demandando a capacidade de mudanças de estratégias de controle. Por exemplo, se o IV

está navegando utilizando uma estratégia de controle baseada em detecção da pista e a possibilidade de colisão com um obstáculo é identificada, a estratégia atual de controle (seguir marcações de pista) deve ser modificada para evitar colisão. Assim, para veículos completamente autônomos um sistema de tomada de decisão é fundamental. Para cada situação identificada no ambiente, uma ou mais estratégias devem ser executadas. Para determinar qual estratégia é mais adequada em um momento específico, diversos projetos utilizam máquinas de estado finito em suas implementações (BISHOP, 2000).

Um sistema de auxílio ao motorista é apresentado em Huang (1998), de forma a requisitar ao motorista assumir o controle do veículo quando situações de perigo são identificadas. Em Ozguner et al. (2007), no entanto, é apresentado um sistema completamente autônomo de um veículo participante do *Darpa Grand Challenge*, onde todas as decisões são tomadas pelo próprio veículo sem a intervenção humana.

As estratégias de controle de alto nível mais comuns são: navegação global e local, seguir pista, navegação em comboio, manobras de ultrapassagem, mudança de pista, parar e continuar navegar, evitar colisão e manobras de estacionamento. Estes tópicos são apresentados a seguir.

### **1.2.1 - Localização Autônoma**

Um tema importante nas técnicas de navegação autônoma é a localização. Como os veículos autônomos estão migrando de ambientes de laboratório “bem comportados” para cenários do mundo real, as técnicas de localização devem ser aprimoradas. Embora pontos de referência, guias ou outras técnicas semelhantes continuem sendo usadas em ambientes externos, elas são insuficientes. Portanto, conforme apresentado na Seção A.3, um conjunto de sensores é utilizado para cumprir a tarefa de localização do veículo.

O posicionamento local não demanda apenas a determinação do posicionamento relativo do veículo em relação a um ponto de referência, mas também a localização de objetos, obstáculos, semáforos, sinais de trânsito, etc.. Esta detecção pode estar baseada em identificação sensorial de alguns elementos no ambiente, podendo processar esta

informação de duas maneiras diferentes: identificando a posição precisa, se a localização do elemento identificado é bem conhecida, ou por meio de Densidade Gaussiana Multivariável que utilizam filtros de Kalman para limpar os dados iniciais e solucionar o problema utilizando uma abordagem de incertezas.

Para localização global a discretização de Markov é a técnica mais aplicada. Esta técnica quebra o espaço de estados mapeando a topologia do ambiente. Embora seja uma metodologia poderosa, tem a desvantagem de consumir muito tempo de processamento em sua implementação. Para evitar um esforço computacional extra, Thrun et al. (2001) afirma que é possível representar o grau de certeza na localização por meio de amostras aleatórias da posição do veículo e restrições baseadas nas variáveis observadas. Além disso, Monte Carlo e condensação são outros métodos capazes de indentificar eficientemente a posição global.

### **1.2.2 – Navegação Local e Global**

Navegação global é um tema relativamente antigo e fácil, que envolve o conhecimento de dados geoespaciais das áreas de interesse e um algoritmo baseado em busca gráfica. O mais comum e também confiável algoritmo é o A\*, que busca uma árvore de soluções para um caminho otimizado baseado em uma função heurística, como por exemplo: menor distância, tempo gasto, segurança, etc.. Exemplos detalhados de aplicação do algoritmo A\* podem ser encontrados em Ozguner et al. (2007).

Enquanto a navegação global é um processo mais simples, o mesmo não ocorre com o planejamento de movimentação local. Nesta situação, o veículo não conhece antecipadamente o ambiente. Assim, ele deve ser capaz de simultaneamente executar em tempo real, técnicas de aprendizado, raciocínio e resolução de problemas.

O campo de potencial artificial e suas variantes é uma das técnicas mais conhecidas para navegação local (FUJIMORI et al., 1997). A literatura também apresenta técnicas baseadas em Redes Bayesianas (KORB; NICHOLSON, 2004), Lógica *Fuzzy* (YEN; PFLUGER, 1995) e outras baseadas em estimação por vídeo (LI et al., 2004; MCCALL;

TRIVEDI, 2006; KUTAMI et al., 1990; TSUGAWA, 1994). A Figura 1 mostra a detecção local de uma pista usando técnicas diferentes de visão computacional (LI et al., 2004).



Figura 1 - (a) Planejamento de rota local utilizando Redes Bayesianas [fonte (YAGI et al., 1995)], (b) identificação da pista por visão computacional [fonte (THORPE et al., 1988)].

### 1.2.3 – Comboio, Ultrapassagem, Parar e Sair e Evitar Colisão

Em ambientes reais é comum para o veículo autônomo estar em uma situação onde a navegação local é restringida por fatores externos, conforme já descrito. Assim, uma vez identificados, um algoritmo específico deve ser executado para controlar o veículo de forma segura.

A Figura 2 mostra três diferentes situações presentes em ambientes reais: (a) identificação de obstáculos, (b) sistema de navegação em comboio e (c) situação de parada, com identificação de obstáculos e semáforos.



Figura 2 - (a) detecção de obstáculos {fonte [(KATO et al., 2002)]}, (b) acompanhamento de carro em comboio {fonte [(BISHOP, 2005)]}, e (c) acompanhando veículos, com identificação de semáforos, situação de parar e seguir e evitar colisão {fonte [(BISHOP, 2005)]}.

### **1.2.4 – Estacionamento Paralelo**

Uma das mais atraentes situações de direção autônoma, inclusive comercialmente, refere-se à manobra de estacionamento paralelo ou baliza. Esta tarefa específica vem atraindo a atenção de pesquisadores e da indústria, levando a uma variedade de metodologias.

As abordagens utilizadas para solução deste problema podem ser classificadas em três grupos: 1) Controle da direção baseado em realimentação de estado; 2) Planejamento de uma trajetória viável baseado no modelo cinemático do veículo; e 3) Combinação de Realimentação de Estados e Planejamento (PAROMTCHIK; LAUGIER, 1996; LEE; TSAI, 2007).

Na abordagem de controle baseado em realimentação de estados é necessário que o sistema de percepção forneça, com exatidão, a localização do veículo durante o movimento para que a manobra seja realizada satisfatoriamente. Já na abordagem que aplica planejamento é necessário utilizar o modelo cinemático e/ou dinâmico. No entanto, alguns parâmetros do veículo são incertos ou desconhecidos, o que dificulta a obtenção de modelos que representem o veículo com exatidão. Visando minimizar os problemas típicos de cada abordagem, gerando um sistema mais robusto, algumas técnicas combinam realimentação de estados e planejamento. A idéia principal desta abordagem é realizar a movimentação do veículo aplicando um procedimento repetitivo e interativo de localização, planejamento e execução até que a posição objetivo seja atingida.

### **1.2.5 – Execução do Controle**

A execução das decisões neste nível de controle requer resposta altamente precisa e a integração com a camada de controle de baixo nível. É necessário identificar corretamente uma situação, definir as estratégias de controle e saber quando determinada situação foi atingida.

Esta função é mais desafiadora quando computadores distintos são responsáveis por cada estratégia de controle. Isto leva a uma situação onde diversos sistemas devem gerar dados e informações em tempo real, estratégias e controle para atuar corretamente no veículo. Para lidar com este desafio utilizam-se conceitos de computação distribuída em tempo real e comunicação de dados via barramento CAN disponível em grande parte dos carros comerciais .

### **1.3 – Motivação e Justificativa**

Sistemas de Inferência *Fuzzy* (FIS) são largamente empregados em tarefas de classificação, em ferramentas de suporte a tomada de decisão, simulação e controle de processos. As principais vantagens que fazem com que os FIS sejam tão aplicados, estão relacionadas a sua capacidade de lidar com conceitos ou termos linguísticos, sua robustez e por serem aproximadores universais capazes de fazer o mapeamento não-linear entre entradas e saídas de um sistema.

Os primeiros tipos de FIS possuíam regras baseadas no conhecimento do especialista e eram chamados sistemas *fuzzy* especialistas ou controladores *fuzzy*, dependendo de sua aplicação final. A acuidade deste tipo de sistema está diretamente relacionada com o conhecimento do especialista e como este conhecimento é representado. Portanto, para sistemas complexos ou para os quais não se tenha conhecimento suficiente do especialista, o projeto do controlador *fuzzy* pode ter baixa acuidade ou até mesmo não ser possível.

Outra classe de FIS são aqueles gerados automaticamente a partir de um banco de dados de entradas e saídas do sistema. Esta abordagem utiliza técnicas de aprendizado de máquina, mais especificamente de aprendizado supervisionado, onde o algoritmo de aprendizado recebe um conjunto de exemplos de treinamento para os quais os rótulos (entradas) da classe associada (saídas) são conhecidos.

Esta abordagem é útil quando o projetista não possui conhecimento total sobre o ambiente onde o agente está e neste caso o aprendizado é a única forma do agente

descobrir o que é preciso fazer. Além disto, esta também é uma boa forma de construir sistemas de alto desempenho, dando a capacidade de aprendizado pela experiência no domínio da aplicação.

Neste caso, o Aprendizado de Máquina é um processo onde os parâmetros do modelo de um sistema são ajustados para que seu comportamento se torne similar ao comportamento do sistema modelado. Desde que este comportamento possa ser caracterizado por pares entrada-saída, o objetivo do processo de aprendizagem é fazer com que o modelo do sistema gere saídas similares às saídas do sistema real quando são aplicadas as mesmas entradas.

O problema de gerar o FIS a partir de exemplos de treinamento pode ser decomposto em duas fases: 1) a indução e otimização da estrutura da base de regras e 2) a otimização das funções de pertinência. A literatura apresenta uma quantidade grande de métodos, técnicas e algoritmos para solução deste problema, que se baseiam em algoritmos de agrupamento (*clustering*), algoritmos genéticos, métodos baseados em gradiente e redes neurais (LI et al., 2002, TRIANTAPHYLLON; FELICE, 2006; WITTEN; FRANK, 2005; MITCHELL, 1997).

No que diz respeito a estrutura das regras, a modelagem *fuzzy* pode ser feita com base em duas abordagens distintas. A primeira abordagem privilegia a interpretabilidade do modelo *fuzzy* pelo ser humano, quando tarefas de extração de conhecimento e análise de dados são importantes. Nesta abordagem geralmente é adotada a estrutura de regras tipo Mamdani para representação do conhecimento. A segunda abordagem privilegia a exatidão na aproximação da função, onde a estrutura de regras tipo Takagi-Sugeno-Kang (TSK) é aplicada. Na prática, ter boa interpretabilidade de regras e alta exatidão do modelo são exigências contraditórias. Em geral uma prevalece sobre a outra, dependendo do objetivo da modelagem e da estrutura de regras utilizada (CASILLAS et al., 2003).

Neste aspecto, o algoritmo proposto nesta tese procura equilibrar as exigências de interpretabilidade e acuidade, propondo o projeto automático de um FIS baseado no modelo tipo Mandani, aplicando uma heurística para geração de regras pré-candidatas denominada Tabela de Regras Potenciais (TRP) e aplicando Sistemas Imunológicos

Artificiais baseado em Gradiente e Clusterização, uma nova técnica baseada em população.

## 1.4 – Objetivo Geral desta Proposta

Conforme apresentado na Seção 1.1, diversas abordagens já foram propostas para controlar a navegação de veículos inteligentes. Algumas das principais são: Sistemas de Controle Clássico, Controladores *Fuzzy*, Redes Neurais Artificiais, Planejamento, Sistemas Classificadores, além de técnicas de localização e rastreamento por visão computacional e instrumentação avançada (técnicas utilizadas isoladamente ou de maneira combinada). No entanto, em todas as referências pesquisadas (citadas no Apêndice A) que tratam dos algoritmos de controle de alto nível aplicados em IV, a abordagem mais comum é a utilização de técnicas de IA em algoritmos programados pelo especialista.

O objetivo desta proposta é explorar um tema pouco pesquisado nas aplicações de Veículos Inteligentes: um algoritmo que permita que o veículo aprenda manobras de dirigibilidade com um motorista humano. A idéia central deste algoritmo é gerar automaticamente um Sistema de Inferência *Fuzzy* (FIS) a partir de um banco de dados numérico que contenha exemplos de treinamento do sistema de controle a ser modelado.

A opção por FIS para aplicação em IV é justificada por se tratar de uma técnica simples e aplicável em controle de sistemas não-lineares sem modelos matemáticos. Além disto, é uma técnica robusta que não requer entradas precisas, podendo ser facilmente compreendida e modificada por utilizar termos linguísticos. Outra característica importante para aplicações em IV, em relação a outras técnicas de controle diz respeito, a facilidade de implementação em sistemas microprocessados e a sua baixa demanda computacional, aspectos relevantes em sistemas embarcados.

O projeto automático de FIS tem sido amplamente pesquisado (Cordón; Herrera, 1999; Jang, 1993; Jin, 2000; Wang; Mendel, 1992). As pesquisas nesta área abordam diferentes estratégias para geração automática da estrutura das regras *fuzzy* e para o

processo de otimização das funções de pertinência que integram a base de conhecimento do FIS.

Por se tratar de uma aplicação de controle, o projeto automático de um FIS para IV deverá obrigatoriamente levar em consideração a exatidão do sistema gerado. No entanto, para preservar umas das características naturais dos sistemas *fuzzy*, a interpretabilidade também será considerada. A estrutura de regras adotada neste trabalho baseia-se no modelo Mamdani, por gerar resultados satisfatórios utilizando uma estrutura simples, intuitiva e de fácil interpretação.

A metodologia aplicada neste trabalho é organizada em 3 fases: geração de um banco de dados de manobras, indução de um FIS a partir deste banco de dados e aplicação do FIS no IV para navegação e realização de manobras. Ao final deste processo deseja-se substituir o controlador humano (motorista) da Figura 3 por um controlador *fuzzy* que aprendeu a “dirigir” com o motorista e que seja capaz de executar suas funções.

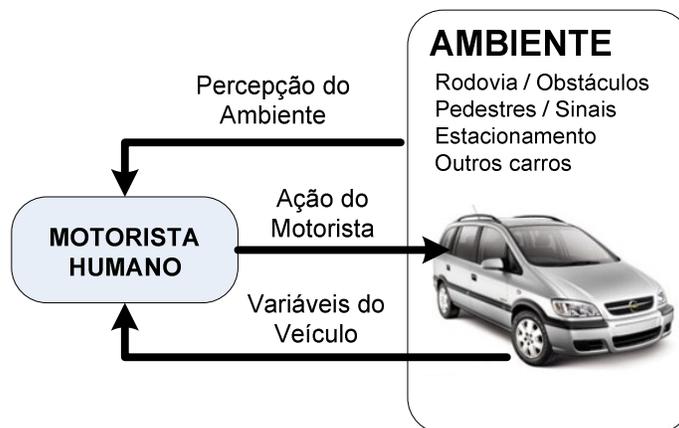


Figura 3 - Interação Motorista – Veículo - Ambiente

Para gerar um banco de dados de manobras é necessário equipar o veículo com um conjunto de sensores capaz de perceber o ambiente externo. É necessário também registrar variáveis internas do próprio veículo, tais como velocidade, ângulo da roda, distância percorrida e ações do motorista: acelerar, frear, virar direção. Desta forma será possível observar o comportamento do motorista em diversos cenários e

armazenar estas informações em um banco de dados para utilização futura. Importante ressaltar que este banco de dados deverá conter exemplos de treinamento considerados válidos, ou seja, que represente correta e satisfatoriamente o sistema a ser modelado.

A segunda fase consiste na extração do conhecimento do motorista contido na base de dados por meio de um FIS. A geração deste FIS implica na geração de uma base de regras e funções de pertinência que representem adequadamente o conjunto de exemplos de treinamento. Assim, o problema é dividido em duas partes: uma responsável pelo aprendizado da estrutura das regras e a outra responsável pela otimização das funções de pertinência. A implementação do algoritmo para solucionar este problema, aplica os conceitos de Sistema Imunológico Artificial baseado em Gradiente e Clusterização (CGbAIS) com duas populações distintas de anticorpos: uma para aprendizagem e outra para otimização.

A fase final é a aplicação do Controlador *Fuzzy* no veículo substituindo o motorista humano. Para validar esta proposta, o algoritmo foi aplicado inicialmente em um ambiente de simulação 3D capaz de reproduzir o modelo cinemático do veículo e posteriormente em um veículo de passeio real.

O detalhamento da implementação de cada fase do algoritmo proposto é apresentado no Capítulo 4 deste documento.

## **1.5 – Principais Contribuições**

Em geral, um método de representação do conhecimento visa fazer um mapeamento entre as entradas e saídas de um sistema real. Independente do tipo destas entradas e saídas (se são contínuas ou discretas), em um FIS elas são representadas por um conjunto de funções de pertinência que requer otimização contínua. Entretanto, a extração da base de regras é um problema de otimização discreta, uma vez que uma regra pode ou não fazer parte do sistema de solução. Desta forma, desenvolver um FIS é, na verdade, resolver um problema de otimização envolvendo variáveis discretas e contínuas. Sendo assim, os problemas referentes à

escalabilidade, ótimos locais, confiabilidade e tempo de processamento devem ser considerados.

Dentre os problemas mencionados, o primeiro tratado neste trabalho é a escalabilidade. Em um problema de otimização, um dos principais fatores que determinam a complexidade para encontrar o ótimo global ou sub-ótimo é o número de variáveis discretas. Quanto maior for este número, maior é o espaço de soluções, aumentando a exigência sobre o sistema. Entretanto, como o problema abordado é real existem combinações de variáveis que nunca irão ocorrer e, desta forma, não necessitam ser exploradas. Para reduzir o espaço de soluções, os dados adquiridos pelo sistema de percepção do ambiente externo (mundo exterior) e interno (variáveis do carro) em conjunto com as ações do motorista são analisados por meio de uma técnica proposta nesta tese denominada de Tabela de Regras Potenciais (TRP). Esta técnica visa pré-selecionar regras candidatas, reduzindo assim o espaço de busca.

A segunda contribuição deste trabalho refere-se ao modelo e métodos de otimização utilizados. A construção de um FIS implica na geração de uma estrutura de regras e funções de pertinência que representem adequadamente o conhecimento e o sistema presentes nos dados adquiridos. Como dito anteriormente, um dos objetivos de ter um conjunto de regras que representam o conhecimento humano em relação ao problema de manobras e direção é ter a capacidade de análise e inferência sobre as características que levaram àquele conjunto de regras e possíveis diferenças existentes entre motoristas. Entretanto isso não pode ser obtido se o conjunto de regras for muito extenso ou se a quantidade de funções de pertinência também for muito grande. Partindo do pressuposto que seja improvável que um motorista utilize uma quantidade muito grande de regras para executar uma manobra, a meta é representar o conhecimento, dentro de uma faixa de exatidão aceitável, porém com um pequeno número de regras.

Com o espaço de solução reduzido e com a estratégia de otimização definida a próxima contribuição deste trabalho é o algoritmo de otimização. O algoritmo de otimização é dividido em duas partes: uma responsável pelo aprendizado das regras (discreto) e a outra responsável pela otimização das funções de pertinência (contínuo). A implementação do algoritmo para solucionar esta topologia, aplica os conceitos de

Sistema Imunológico Artificial baseado em Cluster e Gradiente com duas populações distintas de anticorpos, uma para cada parte. Por meio do processo de co-evolução das duas populações é possível trocar informações entre elas, uma vez que as mesmas são interdependentes. As características do CGbAIS permitem a busca simultânea de vários ótimos locais e reduzir a população sem perda de qualidade de solução por meio da eliminação de indivíduos redundantes.

## 1.6 – Organização do Documento

Esta tese está organizada em seis capítulos, contendo um apêndice. Este capítulo apresenta considerações gerais sobre veículos inteligentes, aplicação que deu origem ao tema proposto nesta tese, motivação, justificativa, objetivo geral e principais contribuições desta tese. O capítulo 2 apresenta conceitos importantes de Aprendizado de Máquina e FIS aplicados nesta tese, assim como uma revisão bibliográfica sobre o projeto automático de FIS. O capítulo 3 apresenta conceitos básicos de Sistemas Imunológicos Artificiais, assim como o algoritmo de seleção clonal CLONALG, base para o desenvolvimento do algoritmo CGbAIS proposto nesta tese. O capítulo 4 detalha o algoritmo para aprendizado supervisionado do FIS baseado em CGbAIS a partir de banco de dados numérico. O algoritmo é validado em aplicações de veículos inteligentes em casos reais e simulados cujos resultados são apresentados no capítulo 5. O capítulo 6 apresenta conclusões, assim como propostas de continuidade deste trabalho. O apêndice A apresenta um *survey* sobre Veículos Inteligentes, destacando sua arquitetura típica, o sistema de percepção e as principais estratégias de controle de alto nível aplicadas nesta área.

## Capítulo 2

# Aprendizado de Sistemas *Fuzzy*

### 2.1 – Considerações Iniciais

Aprendizado de Máquina (AM) ou *Machine Learning* é uma sub-área de Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais relacionadas ao aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática (TRIANANTAPHYLLON; FELICE, 2006). De forma sucinta pode-se afirmar que AM é a área relacionada a algoritmos que aprendem com experiências.

Genericamente existem dois tipos de aprendizado: indutivo, que extrai regras e padrões de grandes conjuntos de dados (também denominados casos ou exemplos), e dedutivo, inferência que parte do universal para o particular. O aprendizado indutivo, alvo de interesse desta proposta, está intimamente ligado à área de Mineração de Dados (*Data Mining*) por compartilharem diversas práticas, métodos e objetivos (WITTEN; FRANK, 2005).

Segundo (MITCHELL, 1997, pag. 2) “um programa aprende a partir da experiência **E**, em relação a uma classe de tarefas **T**, com medida de desempenho **P**, se seu desempenho em **T**, medido por **P**, melhora com **E**.”

Visando exemplificar este conceito e ao mesmo tempo direcionar este estudo para a proposta apresentada nesta tese, consideremos uma aplicação de estacionamento de um IV em uma vaga paralela, manobra conhecida como baliza.

### **Manobra de Baliza**

- Tarefa **T**: estacionar um automóvel em uma vaga paralela utilizando sensores inerciais e de posição.
- Medida de Desempenho **P**: distância e posicionamento final do automóvel em relação à vaga (critério definido pelo ser humano).
- Experiência de Treinamento **E**: um banco de dados com informações dos sensores e comandos de direção registrados por meio de um sistema de aquisição de dados.

Dentre as vantagens da aplicação de técnicas de AM, pode-se destacar que o aprendizado é a única forma do agente descobrir o que é preciso saber, quando o projetista não possui conhecimento total sobre o ambiente onde o agente está ou quando este conhecimento não é suficiente. Além disto, estas técnicas representam excelentes ferramentas para construção de sistemas de alto desempenho, por viabilizar a capacidade de aprendizado pela experiência no domínio da aplicação.

Apesar do conceito e das técnicas de aprendizado poderem variar de uma simples experiência de memorização até a criação de complexas teorias científicas, este trabalho restringe-se ao aprendizado indutivo, ou seja, a construção de uma função que descreve um sistema a partir de um conjunto de exemplos de entradas e saídas do mesmo. Esta restrição é feita, por ser o aprendizado indutivo adequado a proposta de projeto de tese apresentada neste documento.

## 2.2 - Modelo Geral de um Agente de Aprendizado

Um agente de aprendizado pode ser modelado como sendo constituído de quatro componentes conceituais (RUSSELL; NORVIG, 1995), conforme apresentado na Figura 4.

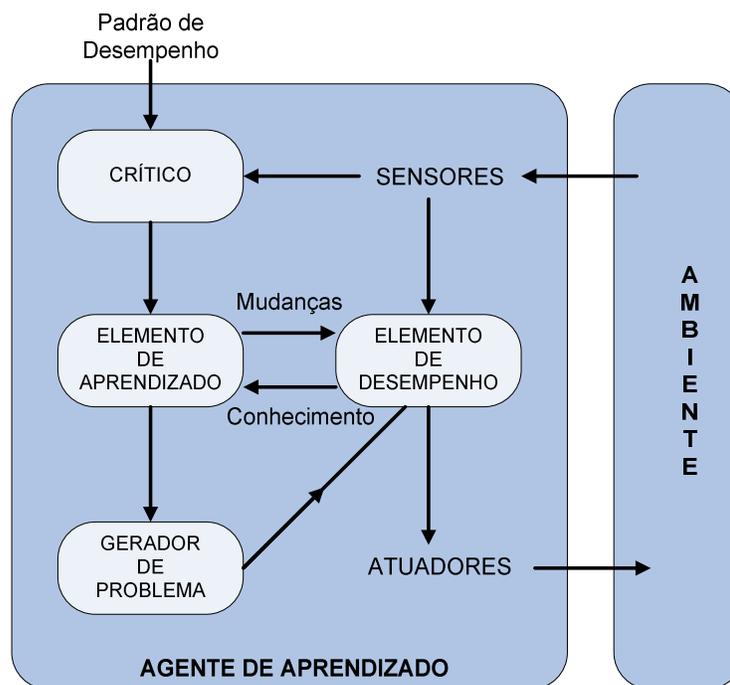


Figura 4 - Modelo Geral de um Agente de Aprendizado

As funções de cada um destes componentes podem ser resumidas como:

1. Elemento de Aprendizado – responsável por fazer melhorias, aperfeiçoamento ou progresso. Este elemento recebe realimentação sobre como o agente está atuando no ambiente e determina que modificações devam ser feitas para que o elemento de desempenho execute melhor suas funções.
2. Elemento de Desempenho – responsável por selecionar ações externas. Recebe sinais de sensores (percepção) e decide que ações devem ser tomadas.
3. Crítico – projetado para avaliar o elemento de aprendizado, a partir de um padrão de desempenho.
4. Gerador de Problemas – responsável por sugerir ações que levarão a novas experiências, como por exemplo, sugerir uma ação exploratória.

No caso específico deste trabalho, o ambiente de aprendizado possui as seguintes características: acessível, não-determinístico, dinâmico, não-linear e possui variáveis discretas e contínuas. Apesar de classificado como acessível, pois o sistema de percepção instalado no veículo é capaz de detectar o ambiente, esta não é uma tarefa trivial. Devido a diversidade de situações a serem detectadas e falta de padronização no ambiente, o acesso a informações do ambiente para tomada de decisões requer a utilização de vários tipos de sensores diferentes. Outro desafio relacionado ao ambiente diz respeito às características não-determinística e dinâmica do ambiente.

## **2.3 - Paradigmas de Aprendizado**

As diversas técnicas de AM podem ser organizadas, de maneira geral, segundo o paradigma empregado pelas mesmas na análise dos dados. Pode-se dizer que a maneira pela qual o ambiente influencia o agente em seu aprendizado define este paradigma, ou seja, o modo como se dará o seu aprendizado por meio dos dados. Os três paradigmas (RUSSELL; NORVIG, 1995) atualmente empregados são:

- Aprendizado supervisionado;
- Aprendizado por reforço;
- Aprendizado não-supervisionado.

Os subitens seguintes descrevem estes três paradigmas de aprendizado, com ênfase no tipo supervisionado, por ser o que está relacionado com a proposta aqui apresentada.

### **2.3.1 - Aprendizado Supervisionado**

No aprendizado supervisionado o algoritmo de aprendizado recebe um conjunto de exemplos de treinamento para os quais os rótulos (entradas) da classe associada (saídas) são conhecidos. Cada exemplo (instância ou padrão) é descrito por um vetor de valores (atributos) e pelo rótulo da classe associada. O objetivo do agente de aprendizado é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados (WITTEN; FRANK, 2005). A principal

característica do aprendizado supervisionado é que o aprendiz aprende por meio de exemplos recebidos de um supervisor sábio (especialista).

Se uma função  $\varphi(\mathbf{x})$  denota a representação desejada do sistema a ser modelado e  $f(\mathbf{x}, \mathbf{p})$  denota o modelo empregando uma aproximação do sistema, onde  $\mathbf{x} \in X$  é um vetor de entradas e  $\mathbf{p}$  é um vetor de parâmetros ajustáveis, a exigência de que o modelo aprendido gere saídas similares às saídas do sistema real, pode ser expressa como:

$$\forall x \in X: \varphi(x) \approx f(x, p)$$

No caso supervisionado o aprendizado acontece utilizando um conjunto de amostras de treinamento (pares de entrada-saída) que devem representar suficientemente o sistema a ser modelado. Se o número de amostras é  $m$ , a entrada da amostra  $i$  é representada por  $\mathbf{x}_i$ , a saída desejada é  $\mathbf{d}_i = \varphi(\mathbf{x}_i)$  e a saída do modelo é  $\mathbf{y}_i = f(\mathbf{x}_i, \mathbf{p})$ , então a fórmula seguinte pode ser utilizada:

$$\forall i \in [1, m]: d_i \approx y_i$$

O erro ( $\varepsilon$ ) mostra quão similares são o modelo e o sistema real. Uma definição amplamente utilizada para o cálculo do erro é a somatória do erro quadrático:

$$\varepsilon = \sum_{i=1}^m (d_i - y_i)^2$$

Naturalmente, o objetivo é minimizar esta função e isto pode ser feito por meio de diversos algoritmos de otimização.

Alguns exemplos de problemas de engenharia que podem ser apresentados na forma de um problema de aprendizado supervisionado incluem:

1. classificação e reconhecimento de padrões;
2. predição (ou previsão) de séries temporais;
3. identificação de sistemas;
4. controle de processos;

## 5. projeto de filtros em processamento de sinais.

Nesta proposta de tese, técnicas de aprendizado de máquina supervisionado serão utilizadas para induzir regras de decisão para controle de um processo. Mais especificamente, deseja-se induzir regras *fuzzy* para controle de navegação de um automóvel a partir de exemplos de treinamento gerados por um motorista humano.

### **2.3.2 - Aprendizado por Reforço**

No aprendizado por reforço não existe uma interação direta com um supervisor ou modelo explícito do ambiente (GULLAPALLI, 1991). Geralmente, as únicas informações disponíveis são um valor que indica a qualidade do desempenho do agente de aprendizado e um objetivo a ser alcançado. Durante o processo de aprendizagem, o agente “tenta” algumas ações (saídas) e recebe um sinal de reforço (estímulo) que permite avaliar a qualidade de sua ação. O sistema em aprendizagem seletivamente retém as ações que levam a uma maximização dos sinais de reforço.

Uma diferença entre o aprendizado supervisionado e o aprendizado por reforço é que no último não existem tabelas de dados relacionando as variáveis percebidas no ambiente. Estas tabelas devem ser construídas implicitamente a partir da experiência adquirida por meio da interação com o ambiente, de modo que se encontre uma função  $f(x_i)$  que maximize o ganho ao longo do tempo.

### **2.3.3 - Aprendizado Não-Supervisionado**

No paradigma não-supervisionado ou auto-organizado não existe supervisor para avaliar o desempenho do agente de aprendizado em relação aos dados de entrada. Nenhuma medida de erro é utilizada para realimentar o agente. Os dados são ditos não-rotulados, no sentido de que as classes às quais eles pertencem ou as saídas desejadas são desconhecidas. O modelo se adapta a regularidades estatísticas nos dados de entrada, desenvolvendo uma capacidade de criar representações internas que codificam as características dos dados de entrada, tornando-se, portanto capaz de identificar a quais classes novos padrões pertencem (VACHKKOV; ISHIHARA, 2008).

Estas técnicas de aprendizado são utilizadas quando o objetivo for encontrar em um conjunto de dados padrões ou tendências (aglomerados) que auxiliem o entendimento desses dados.

## **2.4 - Abordagens Michigan e Pittsburgh para Representação de Regras**

Um aspecto importante a ser considerado nas técnicas de aprendizado para indução de regras que utilizam algoritmos baseados em população é a forma de representação de cada indivíduo e como a população de regras é tratada. Existem duas abordagens gerais para este problema, que receberam os nomes das universidades de Michigan e Pittsburgh onde foram desenvolvidas.

A abordagem Michigan, proveniente dos trabalhos de Holland e Reitman na década de 70 na Universidade de Michigan, emprega uma forma de representação em que cada indivíduo da população representa uma única regra, isto é uma parte da solução candidata composta por todas as regras (população) (MICHALEWICZ, 1997). Na abordagem Michigan, após vários ciclos a solução é dada, em geral, por uma única regra evoluída que representa o melhor indivíduo da população. Entretanto, nem sempre uma única regra é capaz de representar toda a solução de um problema. Existem formas de co-evoluir os indivíduos de forma que toda a população faça parte da solução (SMITH, 2002).

Ainda, outra forma é repetir sucessivas vezes o ciclo evolutivo do AG de forma a obter-se várias regras. Entretanto, esse é um processo lento que demanda um grande esforço computacional.

Por outro lado, os trabalhos desenvolvidos por De Jong e Smith na Universidade de Pittsburgh, empregam outra abordagem na representação dos indivíduos, a qual ficou conhecida como *Pittsburgh* ou simplesmente *Pitt*.

Diferentemente da abordagem Michigan, nessa abordagem cada indivíduo da população representa um conjunto de regras candidatas à solução do problema. Dessa forma, a população contém vários conjuntos de regras, sendo que cada indivíduo (conjunto de regras) representa uma solução homogênea do problema. Comparativamente com a abordagem Michigan, a abordagem Pitt requer um esforço computacional menor para obter a solução, embora o cálculo da aptidão dos indivíduos seja mais complexo que na outra abordagem (FREITAS, 2002).

## 2.5 – Indução de Sistemas *Fuzzy*

Em 1965, Zadeh introduziu no meio acadêmico os conceitos relativos à lógica *Fuzzy*, também chamada de lógica nebulosa ou difusa, proporcionando novos paradigmas conceituais relacionados à teoria dos conjuntos matemáticos. (ZADEH, 1965, 1975)

A lógica *Fuzzy* é uma técnica comparativamente simples e de vasto espectro de aplicabilidade, em particular a problemas de controle e de decisão. O controle executado pela lógica *Fuzzy* utiliza um sistema de inferência baseado em regras ao invés de um controle explicitamente restrito a modelos matemáticos como equações diferenciais. O objetivo do controlador *Fuzzy* é gerar uma saída para o processo a partir de um conjunto de entradas não precisas, com ruídos ou até mesmo faltantes.

De um modo geral, o projeto de um controlador *Fuzzy* consiste em:

1. Definir os universos de discurso das variáveis linguísticas do sistema e variação da saída do controlador ;
2. Definição do número de graus de pertinência dos conjuntos difusos que representam cada termo;
3. Determinação das regras que formam o algoritmo de controle;
4. Definição de parâmetros de projeto, como método de inferência, lógica a ser empregada, forma de defuzzificação e atuação do controlador.

A lógica *Fuzzy* apresenta as seguintes características em relação a outras técnicas de controle:

- Robusta porque não requer entradas precisas.
- Modificada facilmente pois é baseada em regras.
- Sem modelo matemático.
- Solução mais rápida e com menor custo em alguns casos.
- Implementável facilmente em microprocessadores.

Segundo (CORDÓN et al., 2001), em se tratando de métodos de inferência, existem dois tipos principais de Sistemas de Inferência *Fuzzy* (FIS): tipo Mamdani e Takagi-Sugeno-Kang (TSK) abordados detalhadamente nas próximas sub-seções.

### 2.5.1 - Sistema *Fuzzy* Tipo Mamdani

O primeiro tipo de sistema *fuzzy* aplicado em problemas de controle com entradas e saídas reais, foi proposto por Mamdani (1974). Este sistema, também comumente referenciado como Controlador *Fuzzy* Mamdani, é apresentado na Figura 5.

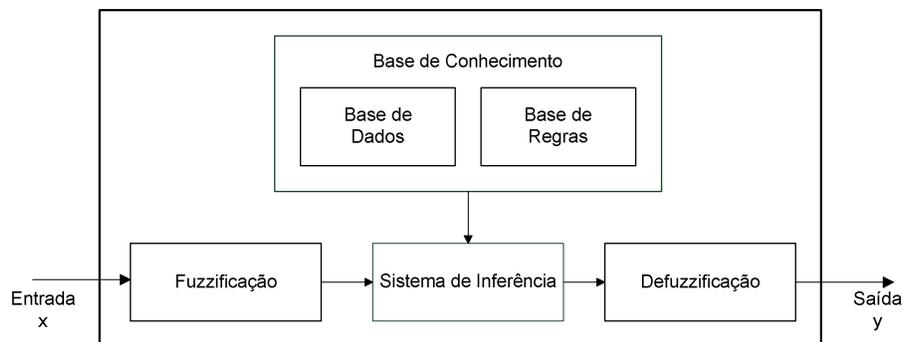


Figura 5 – Estrutura Básica do Controlador *Fuzzy* Mamdani

O bloco denominado Base de Conhecimento armazena o conhecimento disponível sobre o problema na forma de regras *fuzzy* “SE-ENTÃO”, enquanto que os blocos Fuzzificação, Sistema de Inferência e Defuzzificação, compõem a máquina de inferência *fuzzy*. O bloco de Fuzzificação estabelece um mapeamento entre valores de

entradas reais e conjuntos *fuzzy* no mesmo universo de discurso, sendo que o bloco de defuzzificação realiza a operação inversa. O sistema de inferência é o componente que gera as saídas *fuzzy* a partir do conjunto de entradas *fuzzy* de acordo com a relação definida na Base de Conhecimento.

Importante observar que a Base de Conhecimento possui dois níveis diferentes de informação, isto é, a semântica das regras *fuzzy* na forma de conjuntos *fuzzy* e a representação das regras linguísticas contendo o conhecimento do especialista. Esta distinção conceitual é representada na Figura 5 por meio de dois blocos separados: a Base de Dados e a Base de Regras.

A estrutura de regras mais comum no Controlador *Fuzzy* Mamdani envolve variáveis linguísticas na forma:

$$SE (X_1 \text{ é } A_1) \text{ e } \dots \text{ e } (X_n \text{ é } A_n) \text{ ENTÃO } (Y \text{ é } B_1)$$

Onde:  $X_i$  – variável linguística de entrada

$Y$  – variável linguística de saída

$A_i$  e  $B_i$  – valores linguísticos associados a conjuntos *fuzzy* que definem seus significados

A principal diferença entre os sistemas *fuzzy* tipo Mamdani e Takagi-Sugeno-Kang (TSK), a ser descrito na próxima seção, diz respeito ao método de defuzzificação empregado.

Como o processo de inferência do sistema *fuzzy* tipo Mamdani trabalha com um conjunto de regras individuais, é possível que para um determinado valor de entrada, várias regras sejam ativadas individualmente, gerando assim um conjunto de saídas *fuzzy*. Desta forma, a interface de defuzzificação deverá agregar a informação destas várias saídas *fuzzy* para convertê-las em um valor de saída *crisp* (preciso ou não-*fuzzy*). Esta agregação pode ser feita de duas formas distintas: FATI (*First Aggregate, then infer* – Primeiro Agrega, então infere) ou FITA (*First infer, then aggregate* – Primeiro Infere, então agrega), apresentadas nas próximas subseções.

### 2.5.1.1 - Modo FATI

Neste caso o processo de defuzzificação ocorre da seguinte forma:

- Agregação dos conjuntos *fuzzy* de saída em um único conjunto por meio de um operador de agregação mínimo ou máximo, por exemplo;
- Aplicação de um método de defuzzificação, como por exemplo: “centro de gravidade” ou “média dos máximos” para transformar o conjunto *fuzzy* em um valor de saída *crisp*.

No método de defuzzificação “centro de gravidade” ou “centróide”: a saída *crisp* ( $y_0$ ) é o valor no universo que corresponde ao centro de gravidade do conjunto *fuzzy* de saída ( $B$ ), calculado conforme a Equação 2.1.

$$y_0 = \frac{\int y \cdot \mu_B(y) dy}{\int \mu_B(y) dy} \quad (2.1)$$

Já no método “média dos máximos”, a saída precisa é obtida tomando-se a média entre os dois elementos extremos no universo de discurso que correspondem aos maiores valores da função de pertinência do conjunto *fuzzy* de saída ( $B$ ).

### 2.5.1.2 - Modo FITA

Neste modo de defuzzificação a contribuição de cada conjunto *fuzzy* é considerada separadamente e o valor final preciso é obtido por meio da média ponderada dos valores precisos obtidos individualmente em cada regra *fuzzy*.

O cálculo do valor *crisp* da saída, neste caso, é feito por meio da Equação 2.2:

$$y_0 = \frac{\sum_{i=1}^m h_i \times y_i}{\sum_{i=1}^m h_i} \quad (2.2)$$

Onde:

- $y_i$  é o valor *crisp* de saída considerando cada regra *fuzzy* individualmente.

-  $h_i$  é o grau de pertinência obtido ao aplicarem-se os valores de entrada no antecedente das regras.

A vantagem deste modo de defuzzificação é a redução da complexidade computacional comparada com o modo FATI, uma vez que nesta abordagem a inferência é feita antes da agregação.

### 2.5.1.3 - Exemplo de Aplicação

Para ilustrar os conceitos apresentados anteriormente, um exemplo simples de um problema com duas variáveis de entrada (ENT1 e ENT2) e uma saída (SAÍDA) é apresentado a seguir. As funções de pertinência  $\mu$  das variáveis deste problema são apresentadas na Figura 6.

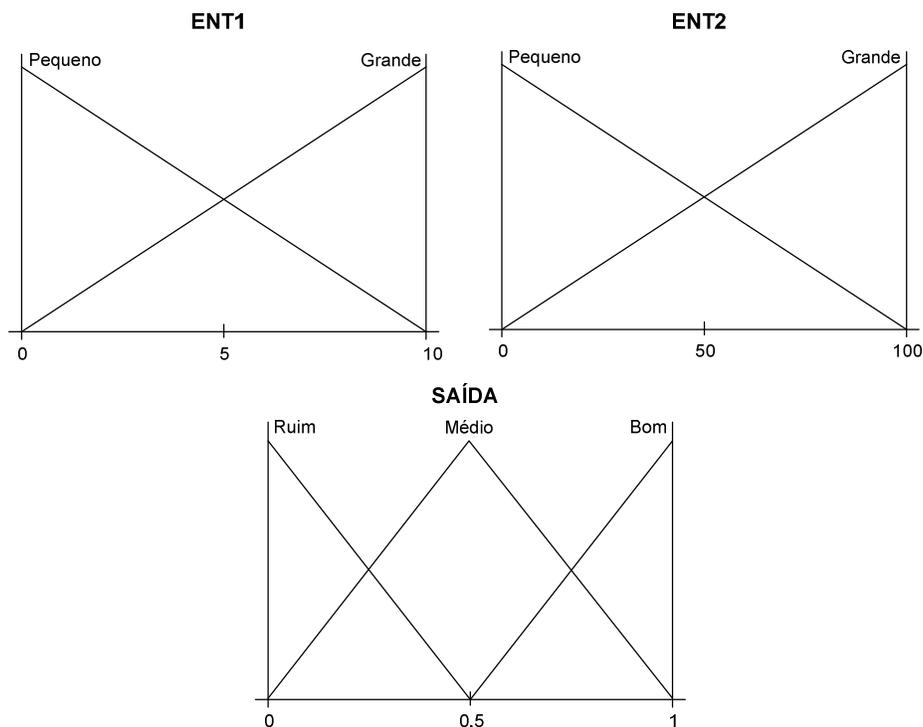


Figura 6 – Funções de pertinência das variáveis consideradas no exemplo

A Base de Regras deste exemplo é composta pelas seguintes regras linguísticas:

R1: SE (ENT1 é pequeno) e (ENT2 é pequeno) ENTÃO (SAÍDA é ruim)

R2: SE (ENT1 é pequeno) e (ENT2 é grande) ENTÃO (SAÍDA é médio)

R3: SE (ENT1 é grande) e (ENT2 é pequeno) ENTÃO (SAÍDA é médio)

R4: SE (ENT1 é grande) e (ENT2 é grande) ENTÃO (SAÍDA é bom)

Neste exemplo considera-se que a interface de defuzzificação utiliza o Modo FATI e o método de defuzzificação centro de gravidade. As variáveis de entrada ENT1 e ENT2, a título de exemplificação, assumem respectivamente os valores precisos 2 e 25, conforme ilustrado na Figura 7.

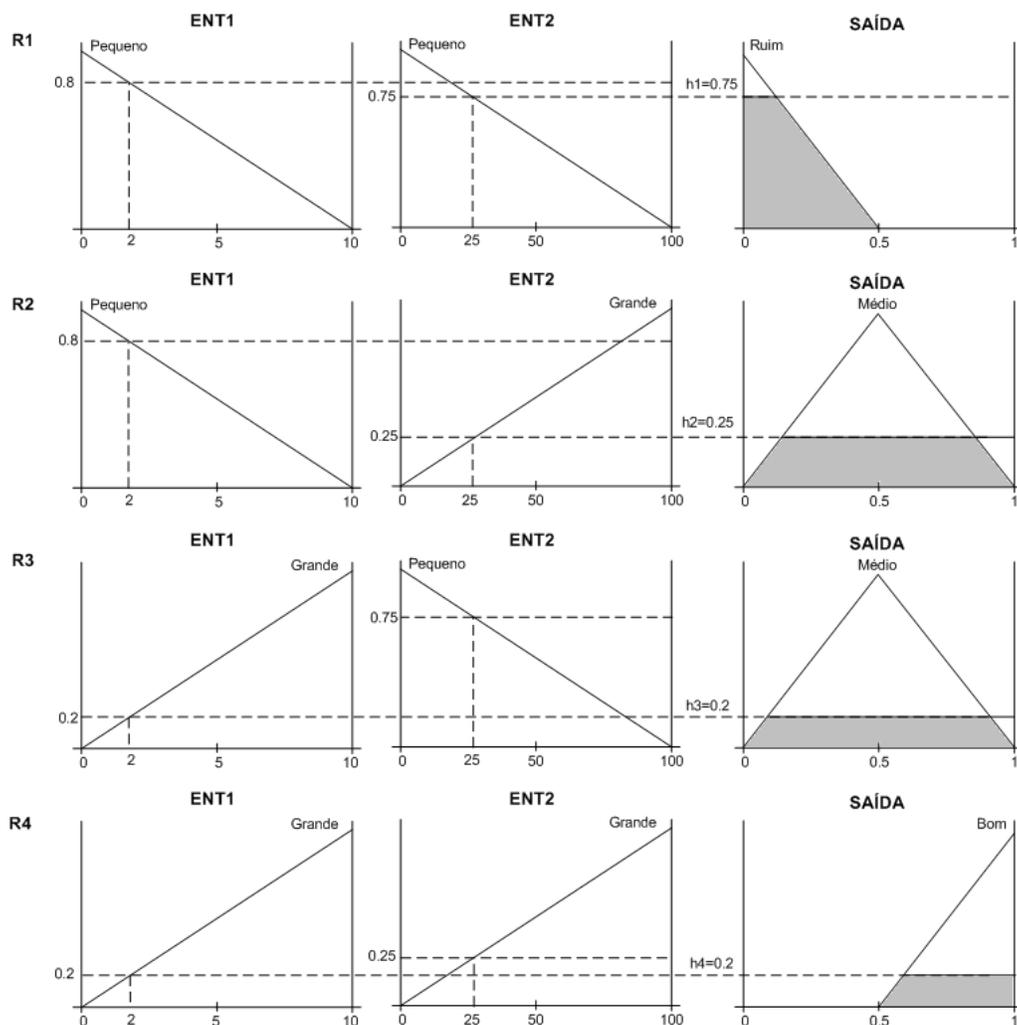


Figura 7 – Modo de Operação do Sistema de Inferência Fuzzy

Estes valores são aplicados ao antecedente de cada uma das regras, conforme representado na Figura 7, para se obter o grau de pertinência  $h_i$ :

$$R1 : h1 = \min(\mu_{\text{pequeno}}(2), \mu_{\text{pequeno}}(25)) = \min(0.8, 0.75) = 0.75$$

$$R2 : h2 = \min(\mu_{\text{pequeno}}(2), \mu_{\text{grande}}(25)) = \min(0.8, 0.25) = 0.25$$

$$R3 : h3 = \min(\mu_{\text{grande}}(2), \mu_{\text{pequeno}}(25)) = \min(0.2, 0.75) = 0.2$$

$$R4 : h4 = \min(\mu_{\text{grande}}(2), \mu_{\text{grande}}(25)) = \min(0.2, 0.25) = 0.2$$

Posteriormente, o valor  $h_i$  é aplicado ao conseqüente de cada regra para se obter a saída *fuzzy* inferida, conforme ilustrado graficamente por meio da área sombreada na Figura 7.

Finalmente, a interface de defuzzificação agrega as quatro saídas *fuzzy* individuais por meio do operador máximo e calcula o valor preciso  $y_0$  de saída, por meio do método centro de gravidade. Este processo é graficamente representado na Figura 8.

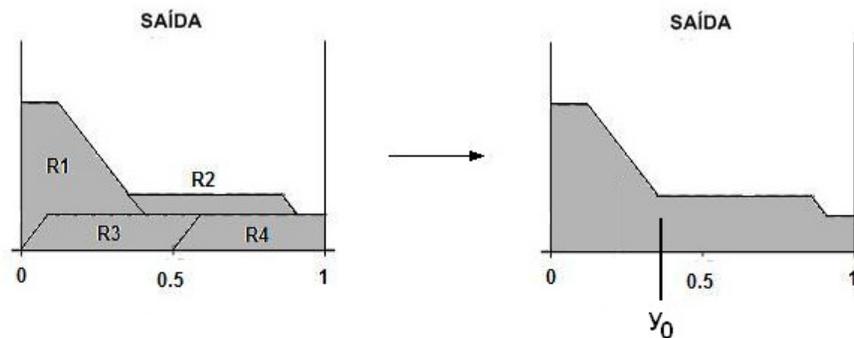


Figura 8 – Defuzzificação Modo FATI com estratégia centro de gravidade

Por outro lado, a defuzzificação utilizando o modo FITA aplicada neste mesmo exemplo é ilustrada na Figura 9. Observa-se pela ilustração que o cálculo do CG de cada área sombreada individualmente ( $y_i$ ) é mais simples do que quando as saídas são agregadas conforme ilustrado na Figura 8.

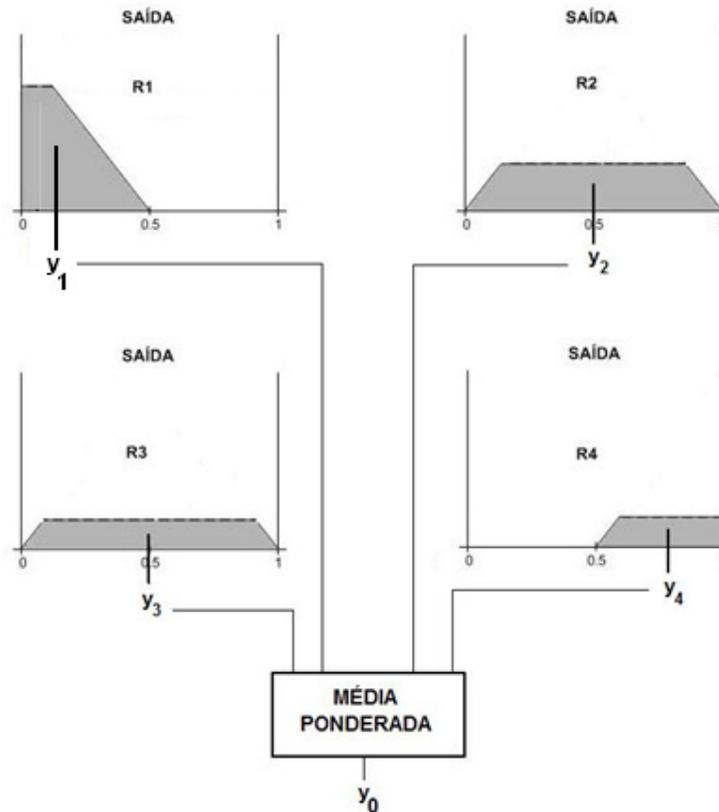


Figura 9 – Defuzzificação modo FITA

## 2.5.2 - Sistema *Fuzzy* Tipo Takagi-Sugeno-Kang

O sistema de inferência *fuzzy* tipo Takagi-Sugeno-Kang (TSK) diferencia-se do sistema Mamdani descrito na seção anterior, basicamente pela representação do conseqüente das regras (CORDÓN et al., 2001). No modelo TSK as regras possuem antecedentes compostos de variáveis linguísticas semelhantes ao modelo Mamdani, sendo o conseqüente representado por uma função das variáveis de entrada. A base de regras típica do modelo TSK tem a forma:

$$R_i: SE (X_1 \text{ é } A_1) e \dots e (X_n \text{ é } A_n) ENTÃO (Y_i = p_1 \cdot X_1 + \dots + p_n \cdot X_n + p_0)$$

Onde:  $X_i$  – variável linguística de entrada

$Y_i$  – variável de saída

$A_i$  – valores linguísticos associados a conjuntos *fuzzy* que definem seus significados

$\vec{p} = (p_0, p_1, \dots, p_n)$  - vetor de parâmetros reais

Para se obter a resposta final, considerando-se a contribuição de cada uma das regras ativadas, deve-se calcular a média ponderada de todos os resultados de saída, considerando os graus de pertinência de cada regra ativada, conforme a expressão:

$$\frac{\sum_{i=1}^m \mu_i \cdot Y_i}{\sum_{i=1}^m \mu_i}$$

Onde:  $m$  - número total de regras

$\mu_i$  - grau de pertinência de cada regra ativada

$Y_i$  - valor de saída de cada regra ativada

## 2.6 – O Projeto Automático de Controladores *Fuzzy*

O projeto não automatizado de controladores *fuzzy* é baseado na experiência pessoal de um especialista, que deverá determinar o mapeamento entre entradas e saídas do sistema a ser controlado, assim como ajustar as funções de pertinência de cada variável do sistema. No entanto, em algumas aplicações, o conhecimento do especialista pode ser impreciso ou não ser suficiente para o projeto destes controladores. Em cenários como este, os conceitos de Aprendizado de Máquina e Mineração de Dados podem ser utilizados para analisar um banco de dados de variáveis que representem o comportamento do sistema a ser controlado e gerar automaticamente as regras *fuzzy* e as funções de pertinência.

A literatura apresenta diversas metodologias para extrair regras *fuzzy* de banco de dados numéricos (PAL; CHAKRABORTY, 2001; QUINLAN, 1993; JONIC et al., 1999; ALVES et al., 2004; WANG; MENDEL, 1992). Uma abordagem comumente utilizada é a extração de regras *fuzzy* utilizando árvores de decisão construídas a partir de amostras de dados. Os algoritmos mais populares nesta categoria são o ID3 (PAL; CHAKRABORTY, 2001) e o C4.5 (QUINLAN, 1993).

O método de aprendizado de regras *fuzzy* baseado em grau de certeza são aqueles que visam identificar uma configuração preliminar (em alguns casos, definitiva) da base

de regras, atribuindo um peso (grau de certeza) para cada regra obtida nos exemplos de treinamento. Este processo é realizado fazendo-se uma única varredura do banco de dados.

Duas estratégias distintas podem ser aplicadas neste caso: 1) estratégia orientada a espaço de entrada (ISS – do inglês *Input Space oriented Strategy*), que visa identificar a melhor regra examinando cada região *fuzzy* (Ishibuchi et al., 1992); e 2) estratégia orientada a exemplos (ES – do inglês *Example oriented Strategy*), que visa selecionar a melhor transformação de exemplos de treinamento em regras *fuzzy* (Wang; Mendel, 1992).

Carmona et al. (2004) analisa o comportamento e as características de duas estratégias para identificar regras *fuzzy* baseadas em grau de certeza. As estratégias ISS e ES são aplicadas em problemas com nível de complexidade diferentes. Os resultados mostraram que nenhuma estratégia possui desempenho totalmente superior a outra. Ambas possuem vantagens e desvantagens dependendo das características do problema. Carmona et al. (2004) propõe um método híbrido que combina as vantagens dos dois métodos. Apesar da simplicidade do método proposto, da baixa demanda computacional e da melhoria na exatidão do sistema *fuzzy* gerado quando comparado com os métodos originais, a otimização das funções de pertinência não é abordada. Esta abordagem poderia conduzir a melhorias na exatidão.

Em Alves et al. (1993) um algoritmo de descoberta de regras de classificação integrando Sistemas Imunológicos Artificiais (AIS) e Sistemas *Fuzzy* é proposto. Este algoritmo possui duas partes distintas: procedimento de cobertura seqüencial e o procedimento de evolução de regras. Cada anticorpo (candidato à solução) corresponde a uma regra de classificação. A classificação de novos exemplos (antígenos) considera não apenas o *fitness* da regra *fuzzy* baseada em todo o conjunto de treinamento, mas também a afinidade entre a regra e o novo exemplo. Embora todos estes métodos apresentem resultados muito bons, eles se baseiam no princípio de que cada regra no processo de inferência deve ser construída pelo aprendizado individual de cada parte do antecedente. Esta estratégia gera um espaço de solução muito grande, além de consumir muito tempo computacional para encontrar a solução para o problema.

Uma outra abordagem, que reduz significativamente o espaço de soluções, é apresentada em Wang e Mendel (1992). A partir de funções de pertinência padronizadas e pré-estabelecidas, o algoritmo utiliza uma lógica simples para gerar regras capazes de representar um determinado sistema a partir de um banco de dados numérico de suas variáveis. Esta lógica baseia-se na associação de entrada e saída, levando-se em consideração a maior função de pertinência de cada variável, onde cada amostra do banco de dados original pode gerar uma regra *fuzzy*. Ao final desta etapa, tem-se um conjunto de regras *fuzzy*, que serão tratadas a fim de eliminar inconsistências. Este método também apresenta bons resultados e é capaz de lidar com banco de dados grandes. No entanto, algumas de suas desvantagens são: a utilização de funções de pertinência padronizadas e a possibilidade de geração de um conjunto de regras inexpressivas devido às técnicas utilizadas para evitar inconsistências e redundâncias.

Danyadi et al. (2010) compara a eficiência de diversos algoritmos evolucionários aplicados em aprendizado de sistemas de inferência *fuzzy* (FIS). Sete algoritmos evolucionários foram utilizados em três problemas clássicos visando otimizar a base de regras de um FIS. A comparação foi realizada com base em critérios de convergência do algoritmo e precisão do modelo gerado. Os algoritmos baseados em clonagem, como o CLONALG, apresentaram melhor desempenho relacionados a convergência do que os algoritmos baseados em *crossover* dos pais, como o Algoritmo Genético. Na comparação da precisão do modelo gerado, com base no critério do erro médio quadrático, o algoritmo CLONALG foi classificado entre os dois melhores algoritmos nos três problemas estudados.

Considerando a análise feita nesta seção e a estratégia de otimização aplicando Sistemas Imunológicos Artificiais baseado em Gradiente e Clusterização apresentada no Capítulo 3, a proposta de um algoritmo capaz de aprender manobras de dirigibilidade a partir de exemplos de treinamento gerados por um motorista será apresentada no Capítulo 4.

## Capítulo 3

# Sistemas Imunológicos Artificiais

### 3.1 – Considerações Iniciais

Diversos são os métodos de otimização existentes inspirados em processos naturais e baseados em população. Dentre estes, pode-se destacar: Algoritmos Genéticos (AG), Enxame de Partículas (Swarm), Colônia de Formigas (Ant Colony) e Sistemas Imunológicos Artificiais (AIS).

A seleção do método de Sistemas Imunológicos Artificiais baseou-se nas características do problema a ser solucionado, ou seja, a necessidade de otimização dinâmica, implementação de busca local e global, otimização contínua e discreta, além da característica de memorização.

A analogia entre sistemas biológicos e sistemas computacionais é cada vez mais explorada por pesquisadores que, motivados pela alta sofisticação e eficácia dos sistemas biológicos utiliza-os como fonte de inspiração para as mais diversas aplicações.

A metodologia do sistema imunológico artificial (AIS) é uma destas técnicas. Tem como princípio o sistema imunológico natural. As características de um sistema imunológico podem ser utilizadas principalmente para reconhecimento de padrões, otimização e aprendizado.

Este capítulo tem como objetivo apresentar os conceitos básicos de AIS, assim como seus principais algoritmos.

### **3.2 – O Sistema Imunológico Artificial**

O sistema imunológico artificial (AIS) é baseado no princípio do sistema imunológico natural. As características de um sistema imunológico podem ser utilizadas tanto para aprendizado como para otimização (CASTRO; ZUBEN, 1999, 2002). Para problemas de otimização três tópicos são interessantes: proliferação, mutação e seleção. Enquanto a proliferação é a capacidade de gerar novos indivíduos, mantendo a diversificação e fazendo o processo dinâmico de otimização, mutação é a habilidade de pesquisar dentro de um espaço de solução por pontos sub-ótimos. A mutação ocorre com duas características diferentes: hipermutação, responsável pela pesquisa local, e o receptor de edição, responsável pela pesquisa global. O processo de seleção é responsável pela eliminação das células com baixa afinidade. Estas características fazem de AIS uma poderosa ferramenta que possibilita a pesquisa para vários ótimos locais.

Existem várias metodologias de AIS disponíveis na literatura para algoritmos de otimização. Os autores Castro e Zuben (2002) e Honório et al. (2007) mostram uma estratégia interessante encontrada em alguns algoritmos evolutivos denominada agrupamento, onde indivíduos são direcionados a pontos mais promissores no espaço de solução. Apesar destes algoritmos apresentarem bons resultados, o número de indivíduos usados no processo é alto, inadequado para problemas em grande escala e não-lineares.

Para aprimorar os algoritmos originais CLONALG (CASTRO; ZUBEN, 2002) e Opt-Ainet (TIMMIS; EDMONDS, 2004), descritos ainda neste capítulo, e melhorar a dinâmica de problemas não-lineares e em grande escala serão realizadas duas modificações.

Na primeira, informações numéricas são inseridas ao gradiente durante o processo de hipermutação (HONORIO et al., 2007). Desta forma é possível guiar este procedimento (no algoritmo original de AIS a geração dos melhores indivíduos é totalmente aleatória), logo o processo de otimização será mais rápido e dotado eventualmente de maior probabilidade de sucesso. Outra vantagem está na possibilidade de criar um critério de parada bem definido para cada anticorpo, desde que um gradiente nulo signifique um ponto estacionário.

Apesar deste método ter uma convergência mais rápida, o número de indivíduos definidos no início do processo é o mesmo. Como na metodologia AIS baseada em agrupamento todos os anticorpos são influenciados pelo mesmo atrator (um ótimo local) os indivíduos poderão convergir para um único ponto, gerando redundância. Baseado nesta informação a segunda mudança está na adição do controle de maturação ao algoritmo.

Com o uso destes conceitos pretende-se identificar os clusters de indivíduos influenciados pelo mesmo atrator e então, durante o processo de evolução eliminar os redundantes, exceto o melhor, acelerando a convergência.

Embora o conceito de clusterização tenha sido introduzido em (HONORIO et al., 2007) a estratégia adotada aqui é outra. A principal diferença está na redução da população durante o processo de convergência, diminuindo o esforço computacional.

### **3.3 - Principais Algoritmos**

Dentre os algoritmos bem sucedidos, baseados em sistemas imunológicos, serão citados o CLONALG e Opt\_AiNet.

#### **3.3.1 - CLONALG**

O algoritmo de seleção clonal (CLONALG) foi proposto por Castro e Zuben (2002). Originalmente, foi desenvolvido para resolver problemas de aprendizagem de máquina

e reconhecimento de padrões. Possui uma população aleatória de anticorpos que devem aprender e reconhecer um conjunto de antígenos. Devido às características adaptativas, foi estendido para aplicações de problemas de otimização.

O algoritmo visa representar a resposta da imunização adaptativa de um organismo na presença de um antígeno Ag, reproduzindo aqueles indivíduos com altas afinidades e selecionando seus melhores clones. Por meio destes conceitos é possível gerar soluções de alta qualidade para problemas complexos. A utilização do operador mutação é feita como critério de diversificação e o operador clonagem como critério de intensificação.

A seguir a descrição genérica e simplificada do algoritmo de acordo com Castro e Zuben (2002):

### **Algoritmo CLONALG**

1. Gere um conjunto (P) de candidatos a solução, composto pelo subconjunto de células de memória (M) mais o restante ( $P\{r\}$ ) da população ( $P = P\{r\} \cup M$ );
2. Determine (processo de seleção) os n melhores indivíduos ( $P\{n\}$ ) da população (P), baseado em uma medida de afinidade;
3. Reproduza (processo de clonagem) estes n melhores indivíduos, gerando uma população temporária de clones (C). A quantidade de filhos (clones) de cada indivíduo é diretamente proporcional à sua afinidade;
4. Submeta a população de clones a um esquema de hipermutação em que a taxa de mutação é proporcional à afinidade do anticorpo. Uma população de anticorpos maduros é gerada ( $C^*$ );
5. Re-selecione os melhores indivíduos de  $C^*$  para compor o conjunto de memória M;
6. Substitua d anticorpos por novos indivíduos (diversidade). Os anticorpos com menores afinidades possuem maiores probabilidades de serem substituídos.

Neste algoritmo, se  $n = N$ , ou seja, se todos os indivíduos da população forem selecionados para reprodução, cada candidato à solução será visto localmente, resultando em um algoritmo capaz de executar uma busca multimodal dentro do espaço S.

As etapas 2 e 5 do CLONALG podem ser feitas de forma probabilística, ou seja, aqueles indivíduos com maiores afinidades terão maiores probabilidades de serem selecionados.

A literatura dispõe de outros algoritmos baseado na teoria da seleção clonal para as mais variadas aplicações.

### 3.3.2 - Opt-Ainet

O Opt-Ainet é um algoritmo inspirado no princípio da seleção clonal e teoria da rede imunológica que foi desenvolvido para solucionar problemas de otimização multimodal. É considerado um procedimento de busca evolutivo baseado na mutação e população com tamanho dinâmico. Apresenta várias características interessantes, como a possibilidade de ajustar o tamanho da população dinamicamente, exploração no espaço de busca para soluções novas e melhores, capacidade de localizar múltiplos ótimos, definir critérios de parada, dentre outros.

O algoritmo pode ser considerado uma extensão do CLONALG com passos envolvendo as interações anticorpos-anticorpo e não apenas interações antígenos-anticorpos. Possui um ciclo de maturação por afinidade e é repetido até atingir a estabilidade. No caso do Opt-Ainet a estabilidade da rede é medida pela diferença entre a afinidade média da população atual com a anterior.

O algoritmo inicia com geração aleatória e avaliação da população inicial. O critério de parada é guiado pela variação do tamanho da população sendo interrompido quando o tamanho da mesma não varia entre duas iterações consecutivas.

Segundo Timmis e Edmonds (2004) o algoritmo segue os passos:

#### **Algoritmo Opt-AiNet**

1. Gerar uma população inicial (o número inicial não é relevante)
2. Enquanto não [população constante da memória], faça
3. Calcule *fitness* (valor da função objetivo) e gere clones de cada célula da rede.
4. Mudar clones proporcionalmente ao *fitness* e determinar o *fitness* novamente.
5. Calcular a média *fitness*.
6. Se média *fitness* não variar, então continue. Senão, retorne ao passo 2
7. Calcula a afinidade entre as células e eliminar todas, com exceção das que a afinidade sejam maiores que o limite  $s$ . Determinar o número após a eliminação.
8. Introduzir a porcentagem de células geradas aleatoriamente e retornar ao passo 2
9. Fim Enquanto.

Este algoritmo gera um número fixo de clones de cada célula (anticorpo) da rede, sendo que este número não é proporcional a afinidade da célula. Posteriormente ocorre a hipermutação, geralmente aplicada a taxas inversamente proporcionais a afinidade de cada clone.

Os clones de uma célula formam uma sub-população, e a melhor célula é selecionada. Caso sua afinidade seja maior, esta substitui uma célula pai.

Quando a estabilidade é alcançada, ocorre uma etapa de supressão, onde se tenta eliminar a redundância de células na população.

### **3.4 - CGbAIS : Sistema Imunológico Artificial baseado em Gradiente e Cluster**

No método AIS, o conjunto de variáveis de entrada ou de controle representa os anticorpos (Ab's). Em sistemas não-lineares o gradiente representa a sensibilidade do sistema, dada uma pequena perturbação ao redor de determinado ponto de operação. Estas sensibilidades são avaliadas para cada variável de controle e o resultado final define um vetor que indica a direção mais provável para alcançar o objetivo da otimização. Os incrementos aplicados geram os clones hipermutados.

#### **3.4.1 - Análise do Gradiente em Otimização Contínua**

Existem vários caminhos para avaliar o gradiente. A escolha depende das características do sistema. Por exemplo, se um problema não-linear contínuo é utilizado, o vetor tangente (VT) podem ser aplicados. O VT trata a função objetivo como uma "caixa preta", onde pequenas perturbações são individualmente aplicadas sob cada dimensão do vetor de entrada, gerando o vetor indicado na Equação 3.1.

$$VT = \begin{bmatrix} \frac{f(x_1 + \Delta x_1, \dots, x_n) - f(x_1, \dots, x_n)}{|\Delta x_1|} \\ \vdots \\ \frac{f(x_1, \dots, x_n + \Delta x_n) - f(x_1, \dots, x_n)}{|\Delta x_n|} \end{bmatrix} \quad (3.1)$$

onde:  $n$  é o número de variáveis de controle ou de entrada,  $f(\cdot)$  é a função objetivo a ser otimizada;  $x_1, \dots, x_n$  são as variáveis de entrada; e  $\Delta x_k$  é o incremento aleatório limitado para  $x_k$ .

No caso de VT, o número de clones hipermutados ( $nc$ ) deve ser igual ao número de variáveis de entrada, por exemplo,  $n$ , desde que este número seja suficiente para assegurar que a sensibilidade de todo o espaço seja capturada apropriadamente.

Como exemplo, considere a função apresentada em Castro e Zuben (2002), e exibida na Equação 3.2

$$Max f(x_1, x_2) = x_1 \sin(4\pi x_1) - x_2 \sin(4\pi x_2 + \pi) + 1 \quad (3.2)$$

O vetor tangente (VT) é apresentado pela Equação 3.3:

$$VT = \begin{bmatrix} \frac{f(x_1 + \Delta x_1, x_2) - f(x_1, x_2)}{|\Delta x_1|} \\ \frac{f(x_1, x_2 + \Delta x_2) - f(x_1, x_2)}{|\Delta x_2|} \end{bmatrix} \quad (3.3)$$

Aplicar algum destes vetores no processo de hipermutação significa gerar o seguinte clone:

$$cAb_i = Ab_i + \alpha \times rand \times Gr \quad (3.4)$$

onde,  $cAb_i$  é o clone do elemento  $i$ ,  $\alpha$  é a mutação dada pela direção do gradiente,  $rand$  é um número aleatório entre de 0 e 1, e  $Gr$  é o vetor gradiente, calculado pelo VT. Na Equação 3.4, " $\alpha \times rand \times Gr$ ", representa o passo de mutação empregado.

Como o gradiente tende a ser nulo perto de um ponto estacionário, o valor médio da expressão diminui, a precisão numérica aumenta e saltos ao redor do ótimo local são evitados. Porém, para aumentar ainda mais a precisão numérica e possibilitar um critério de parada bem definido, o algoritmo GbHipermutação é utilizado e descrito pelo algoritmo a seguir.

A seguir a descrição da terminologia é apresentada.

$Ab\{nps\}\{n\}$  estrutura representando a população composta por  $nps$  anticorpos e  $n$  variáveis de entrada.

$Ab\{i\}\{j\}$  campo representando a variável de entrada  $j$  do anticorpo  $i$ ;

$Ab\{i\}.gr$  campo de dimensão  $n$ , representando o vetor gradiente de  $Ab\{i\}$ ;

$Ab\{i\}.open$  Campo booleano, representando se o anticorpo ainda está aberto para mutações ou não;

$Ab\{i\}.\alpha$  fator de mutação;

$Ab\{i\}.fit$  valor da função de avaliação;

$F(Ab\{i\})$  função de avaliação;

$cAb\{i\}$  clone de  $Ab\{i\}$ , herda todos os seus campos;

$\beta$  parâmetro para controlar a mutação em caso de falha na melhoria;

$\alpha_{min}$  comprimento de passo mínimo.

**Algoritmo** Hipermutação com entrada  $Ab_{\{nps\}\{n\}}$ 

```

FOR EACH  $Ab_{\{i\}}$  IN  $Ab_{\{nps\}}$ 
  IF1  $Ab_{\{i\}}.open == TRUE$ 
     $Ab_{\{i\}}.gr = GRDVECTOR(Ab_{\{i\}});$ 
     $cAb_{\{i\}} = Ab_{\{i\}} + Ab_{\{i\}} \cdot \alpha * rand * Ab_{\{i\}}.gr;$ 
     $cAb_{\{i\}}.fit = F(cAb_{\{i\}});$ 
    IF2 ( $cAb_{\{i\}}.fit > Ab_{\{i\}}.fit$ )
       $Ab_{\{i\}}.fit = cAb_{\{i\}}.fit;$ 
    ELSE2
       $Ab_{\{i\}}.\alpha = \beta * Ab_{\{i\}}.\alpha;$ 
      IF3 ( $Ab_{\{i\}}.\alpha < \alpha_{\{min\}}$  )
         $Ab_{\{i\}}.open = FALSE;$ 
      END IF3
    END IF2
  END IF1
END FOR EACH

```

A função  $GRDVECTOR(.)$  pode ser calculada pelo vetor tangente. É possível concluir que VT precisa de 3 clones (um para cada dimensão e outro representando a direção conjugada final).

Baseado no problema dado pela Equação 3.2, uma análise sobre a eficiência do algoritmo é ilustrada pela Figura 10, onde todos os  $Ab$ 's com todos os clones são apresentados. A Figura 10 (a), (b) e (c) exibem resultados obtidos com a técnica VT e, a Figura 10 (d), (e) e (f) o método utilizado pelo CLONALG e opt-AiNET. É importante salientar que estes dois algoritmos possuem basicamente o mesmo processo de hipermutação, logo, os resultados são bem parecidos.

Analisando os resultados obtidos pelo CLONALG, com as melhorias fornecidas pela adição da informação do gradiente ao processo de convergência, é possível concluir que a alteração trouxe grandes vantagens ao processo evolutivo. Prosseguindo com a análise e considerando que cada ponto da figura é uma avaliação do problema, pode-se concluir também que sistemas com alta dimensão teriam uma velocidade maior no processo de convergência, conforme a Figura 10.

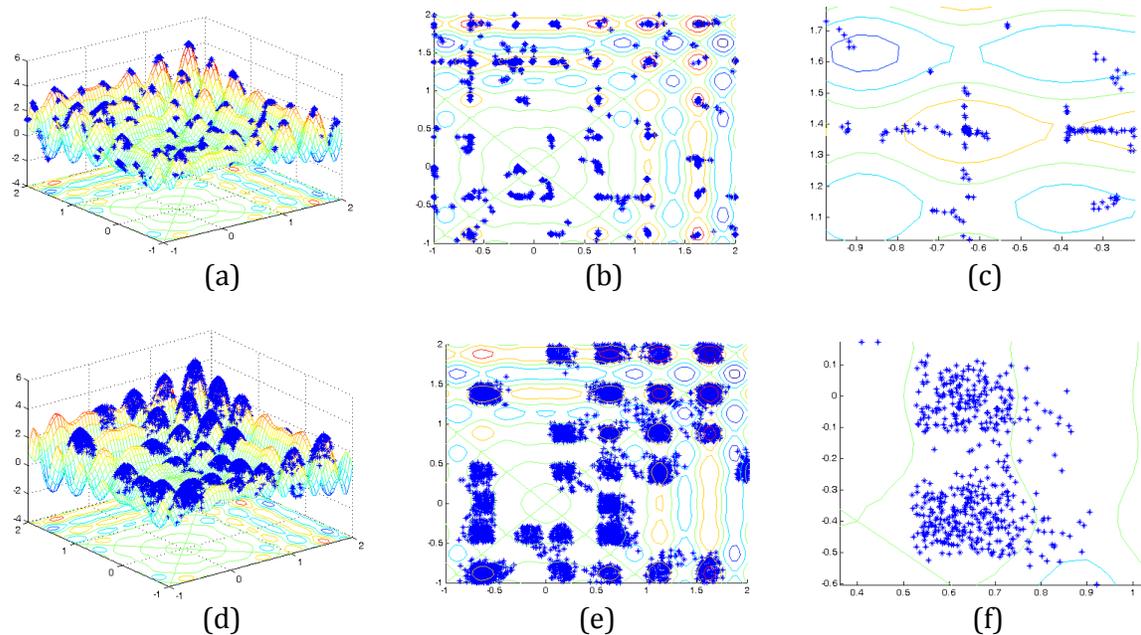


Figura 10 - Processo de Hipermutação – Vetor tangente (a,b,c) e Utilizando indivíduos aleatórios (d,e,f)

### 3.4.2 - Clusterização e Controle de Maturação em Otimização Contínua

Embora a adição do vetor gradiente tenha contribuído para uma maior velocidade ao processo de convergência, o algoritmo não utilizou uma característica importante presente no sistema imunológico natural, conhecida como controle de maturação.

O controle de maturação é utilizado na estratégia de agrupamentos no algoritmo de AIS. Esta estratégia permite encontrar simultaneamente vários ótimos locais.

Considere que em um dado sistema vários Ab's convirjam para ótimos locais. Após algumas iterações é possível gerar clusters representados por pontos que foram influenciados por um mesmo atrator. Uma vez que estes clusters estejam bem definidos, é possível aplicar o controle de maturação. Neste processo, em cada cluster são eliminados todos os Ab's, exceto o melhor. Esta estratégia pode reduzir

consideravelmente a população inicial, além de melhorar a resposta computacional. Entretanto, é fundamental que os clusters estejam bem definidos.

A literatura mostra vários algoritmos para clusterizar conjuntos de dados. Neste trabalho foi utilizado o método da distância MaxMin (MMD) devido a duas vantagens: a) o algoritmo estima automaticamente o número de clusters (o que é muito importante, já que não se conhece previamente o número de ótimos locais no sistema) e b) o algoritmo necessita somente de um parâmetro, o qual pode ser ajustado deterministicamente. O algoritmo segue os passos:

### **Algoritmo** *MAXMIN*

- Passo 0** Definir o parâmetro de distância mínimo ( $D_p$ )
- Passo 1** Definir o  $A_b$  de maior afinidade como centro do cluster 1 e definir  $N_{cluster} = 1$
- Passo 2** Montar o vetor  $VD$ , onde  $VD_i$  é a distância mínima entre  $A_{b_i}$  e todos os clusters centrais  $C_j$  ( $j=1, N_{cluster}$ ). Também armazena  $IDX_i = j$ , onde  $j$  é o índice do cluster com a distância mínima para  $VD_i$ ;
- Passo 3** Se o maior  $VD_i$  for maior que  $D_p$  então vá para 4, caso contrário, vá para 5.
- Passo 4** Definir o  $A_{b_i}$  associado ao maior  $VD$  como sendo o centro de um cluster. Ir para 2.
- Passo 5** Finalizar os centro dos clusters estão armazenados em  $IDX$ .

Aplicando estes conceitos, a Figura 11 ilustra a evolução da população distribuída no espaço de solução, onde: Figura 11(a) mostra a população inicial, Figura 11(b) e (c) exibem a primeira e a segunda geração, respectivamente, Figura 11(d) e (e) mostram a população depois do controle de maturação e Figura 11(f) uma projeção da equação. O número inicial de indivíduos para esta simulação foi de 100 anticorpos. E terminou com 82.

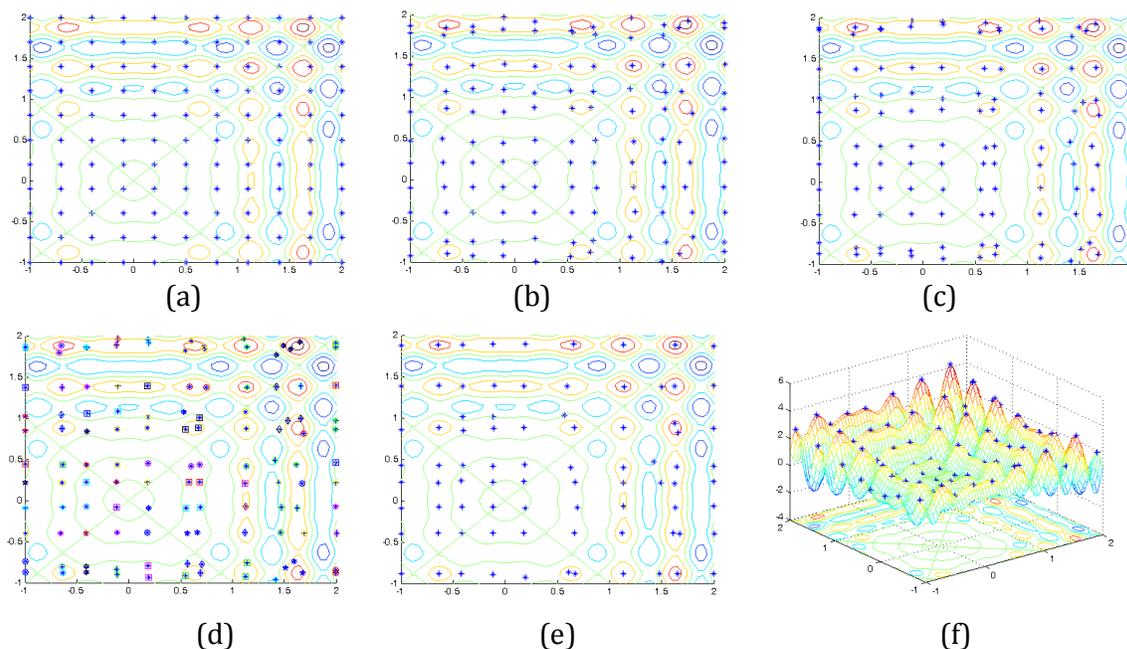


Figura 11- Processo de agrupamento, clusterização e controle de maturação.

De acordo com os resultados, algumas considerações são necessárias:

A primeira refere-se ao número de vezes que a clusterização ocorre. Como mostra a Figura 11, a partir da inicialização os anticorpos movem rapidamente para seus atratores. Realizar o controle de maturação pode produzir uma população escassa, mesmo nos primeiros estágios de evolução. Por esta razão, o processo é executado somente uma vez, na terceira geração.

A segunda está relacionada com o índice de distância, responsável pela determinação da construção do cluster. Se a população é gerada usando uma função de densidade com distribuição uniforme, a distância média entre os indivíduos é a medida do tamanho do espaço de solução pelo número de elementos.

Ainda que a idéia de aplicar a técnica de *clusterização* tenha sido implementada no Opt-AiNET e melhorada posteriormente, os conceitos adotados por estes algoritmos são diferentes dos propostos aqui. As principais diferenças são:

a) No algoritmo opt-AiNET o processo de clusterização acontece depois da primeira simulação, quando o critério de parada é encontrado, enquanto que o CGbAIS realiza este procedimento após poucas gerações.

b) O maior impacto do opt-AiNet em realizar esta clusterização é a redução de memória. Já no CGbAIS o propósito é o ganho de tempo computacional durante o processo de convergência.

### 3.4.3 - O Algoritmo CGbAIS

Analisando o conteúdo apresentado até este ponto é possível concluir que a informação do Gradiente e o controle de maturação podem elevar consideravelmente o esforço computacional e a precisão aos métodos tradicionais baseados no AIS.

A proposta do algoritmo CGbAIS é unificar estes conceitos e melhorar os resultados. A Figura 12 apresenta o diagrama de blocos do método.

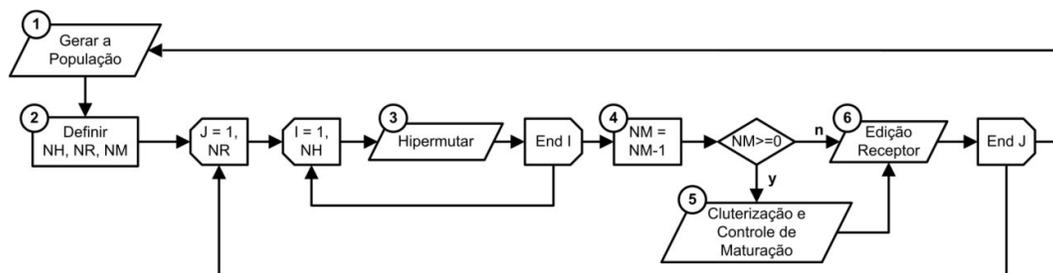


Figura 12 – Algoritmo CGbAIS

onde:

1. Gerar a população aleatoriamente.

2. Definir o número de hipermutações (NH) realizadas antes do passo de clusterização, o número de vezes que o controle de maturação (NM) será executado, e finalmente o do receptor de edições (NR). É interessante frisar que com a utilização VT, a população

tende rapidamente a se agrupar ao redor de um ponto estacionário, assim o NH pode ser muito menor que o NR.

3. Realizar o processo de hipermutação de acordo com a lista 1.
4. Diminuir NM para controlar o número de vezes que o controle de maturação ocorre. Conforme observações empíricas, este valor pode ser fixado em 1 ou 2. Após isto, a redução no número de indivíduos não é significativa.
5. Aplicar o Algoritmo (MMD) para clusterizar todos os indivíduos que convergem para um único atrator. Para cada cluster, eliminar todos os indivíduos, exceto o melhor.
6. Os melhores indivíduos dentro da população original são selecionados para a próxima geração. Os indivíduos restantes são substituídos por novos, gerados aleatoriamente. Este processo simula a edição de receptores e ajuda na busca por melhores soluções em áreas diferentes.

Como o algoritmo proposto consiste de duas estratégias (vetor gradiente e clusterização) é interessante identificar a relevância de cada estratégia separada e o algoritmo final. Quatro simulações foram realizadas. Para efeito de comparação a primeira foi realizada com o Clonalg (um dos principais algoritmos de AIS). Na segunda somente o vetor gradiente é considerado. A terceira somente o processo de clusterização, e quarta (CGbAIS) mostra todas as características (gradiente e cluster).

Conforme mostra a Tabela 1, somente com o gradiente, a *taxa de sucesso* é melhor para todos os casos, o *tempo* é melhor em alguns. Analisando somente com clusterização, a *taxa de sucesso* é menor para todos os casos, o *tempo* também menor. O tempo foi melhor nesta simulação já que foram eliminados indivíduos da população, mas como os indivíduos não foram guiados pelo gradiente a taxa de sucesso foi menor.

Com as duas técnicas juntas a *taxa de sucesso* é semelhante a do gradiente e o *tempo* semelhante ao cluster. A *taxa de sucesso* foi melhor para todos os casos e o *tempo*

foi melhor para quase todos (neste caso existe uma diferença devido a população ser muito pequena, assim o processo de clusterização não eliminou muitos indivíduos).

Com o algoritmo proposto quanto mais indivíduos melhor o tempo computacional comparado ao CLONALG.

Tabela 1 - Resultado das simulações do CGBAIS e variações

Pop	Gen	Pop.Fixa	CLONALG		GRADIENT ON		CLUSTER ON		CGBAIS	
			Taxa Suces%	T. Med(s)						
20	10	15	42	0.09	59	0.11	39	0.08	61	0.09
40	10	20	88	0.25	91	0.2	83	0.17	93	0.1
60	10	30	97	0.47	100	0.29	91	0.23	100	0.1

### 3.4.4 Problemas de Otimização Discreta

Para adaptar os princípios do CGBAIS na otimização discreta, algumas considerações são necessárias:

a) Um anticorpo é representado por um caminho parcial em uma árvore de busca. O nível começa no nível 1, conhecendo apenas um nó, e de acordo com a evolução, novos nós são adicionados. Isto significa que a população inicial pode pesquisar em todo o espaço de solução, e durante o processo de evolução são direcionadas a áreas interessantes.

b) Em cenários discretos, qualquer mudança nas variáveis de controle (neste caso representado por um caminho na árvore) pode levar o sistema a um estado operacional completamente diferente, tornando incorreta a idéia de pequenas mudanças no estado corrente. A estratégia adotada durante a hipermutação foi de manter os nlocalbest clones, ao invés de apenas um. Assim, o algoritmo permite que um determinado anticorpo que não foi bem classificado inicialmente, possa evoluir e tornar-se melhor. Este conceito é o mesmo de outros algoritmos bem sucedidos como o

*Simulate Annealing* ou busca Tabu. Para evitar uma explosão combinatória, a cada iteração, todos os clones são comparados e somente os melhores nGlobalBest são mantidos pelo controle de maturação.

c) Em qualquer aplicação, a condição de um elemento de um dado cluster deve seguir algum indicador de similaridade. No caso da otimização contínua, a população inicial de anticorpos evolui e os clusters são encontrados utilizando o algoritmo de distância MAXMIN. Na otimização combinatória, o trajeto na pesquisa da árvore contém informações mais valiosas que o resultado final, o que indica que o caminho que o elemento evoluiu pode ser usado como medida de similaridade. Desta forma, se um elemento tem os mesmos n-primeiros nós que outro, eles podem ser clusterizados junto no nível-n.

d) O processo de hipermutação ocorre em cada anticorpo onde os clones nLocalBest são mantidos. Como declarado anteriormente, o controle de maturação compara os nLocalBest de todos Ab.s e somente mantêm os nGlobalBest. Neste processo todos os clones de um dado anticorpo podem ser eliminados, o que representa a eliminação de ramificações na árvore de busca. Desta forma, uma boa estratégia é definir o valor nLocalBest menor que o de nGlobalBest (o que evitaria uma eliminação prematura de possíveis caminhos de busca que não são bem classificados em um primeiro momento, mas são promissores). Uma boa estratégia de ajuste permite explorar um bom número de caminhos sem eliminar muitos filhos.

e) Os clones são empregados na análise de um dado nó. A população expande os nós dos seus pais e as informações numéricas obtidas por este processo indicam a probabilidade de encontrar a melhor solução em relação a um determinado caminho. Porém, o sucesso da idéia depende da exatidão da informação numérica, da qual apresenta dois grandes problemas:

- 1) Como evoluir corretamente em cada ramo. Se somente o valor mínimo de um clone for considerado, significa que todos os outros são negligenciados e informações valiosas podem ser perdidas. Para evitar o problema, a função de avaliação utilizada no algoritmo é mostrada a seguir:

$$TV = v_i \times \Phi\{v_1, \dots, v_{nLocalbest}\} \quad (3.5)$$

A função utiliza o valor médio de  $\Phi$  fornecido pelo  $nLocalbest$  individual de cada ramo, e o multiplica por cada clone individual de valor  $v_i$ .

- 2) Se na árvore de buscar tiver dois ou mais ramos com taxas de sucesso parecidas, a probabilidade de não encontrar a melhor no final da simulação vai depender da proximidade destes e do número de clones utilizados para obter as informações necessárias. Neste caso, a solução final pode ser próxima ao melhor global e melhorada facilmente por uma busca local (hipermutação).

A Figura 13 ilustra o processo de como a busca na árvore é realizada. O principal objetivo é expandir um nível da árvore a cada geração. O processo começa no estado inicial 0 e deste, uma população de tamanho  $n$  é gerada, expandindo o primeiro nível da árvore. O processo de clonagem é responsável por definir qual ramo é mais interessante para a próxima pesquisa. Para cada indivíduo, que representa um caminho parcial da árvore, é selecionado um caminho aleatoriamente e avaliado de acordo com a Equação 3.5. Os  $nLocalBest$  clones de cada anticorpo serão selecionados e armazenados em uma lista comum.

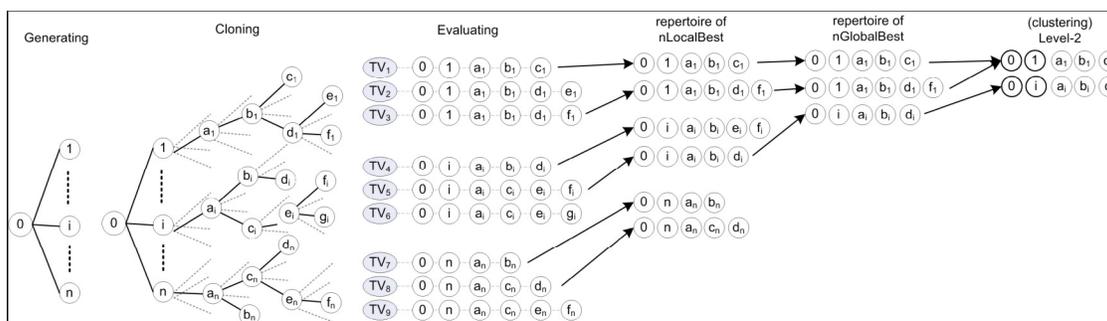


Figura 13 – Otimização Combinatória Utilizando o Algoritmo CGbAIS

Uma observação importante é que os clones formam caminhos completos sob a árvore, e podem ter tamanhos diferentes. Os  $nGlobalBest$  clones são selecionados, gerando anticorpos com um nível a mais que o indivíduo original, representado neste

exemplo pelo nível 2. *Clusterizar* estes indivíduos significa encontrar os indivíduos repetidos e eliminá-los, com exceção de um. Na Figura 13, o clone [0-1-a1-b1-c1] e o [0-1-a1-b1-d1-f1] tem os mesmos dois primeiros nós, [0-1], representando o nível-2 de similaridade, logo eles podem ser *clusterizados*. Note que os anticorpos da população final estão apenas um nível a mais de profundidade na árvore que os iniciais, o que significa que crescem um nó a cada geração.

## Capítulo 4

# Aprendizado Supervisionado baseado no algoritmo CGbAIS

### 4.1 – Considerações Iniciais

Conforme detalhado na Seção A.2 do Apêndice desta tese, tipicamente a arquitetura de veículos inteligentes inclui sistemas de controle de alto nível, ou seja, algoritmos baseados em Lógica *Fuzzy*, Algoritmos Genéticos, Redes Neurais, Aprendizado de Máquina, Planejamento e outras técnicas de Inteligência Artificial.

A abordagem proposta nesta tese é o desenvolvimento e a aplicação de um algoritmo que permita que um veículo aprenda regras de dirigibilidade com um motorista humano. Como estas regras serão executadas por um Sistema de Inferência *Fuzzy*, é necessário, portanto, gerar as Regras *Fuzzy* e as Funções de Pertinência deste sistema. Isto será feito por meio da observação do comportamento de um motorista humano em diversos cenários formando, desta forma, uma base de conhecimento capaz de lidar com uma grande quantidade de incertezas. A metodologia adotada consiste em 3 fases.

A primeira é a criação de um banco de dados de ações. Para isso é necessário a definição e implementação do sistema de percepção do veículo em relação ao ambiente externo (obstáculos e objetivos) e interno (ações do motorista). Com esta definição é possível observar as ações do motorista em diversas situações e armazenar todas as informações externas e internas provenientes dos sensores para futura análise.

A segunda fase consiste na extração do conhecimento do motorista contido na base de dados por meio da indução de um sistema de lógica difusa. A geração deste Sistema *Fuzzy* implica na geração de regras *fuzzy* e funções de pertinência que representem adequadamente o conjunto de exemplos de treinamento. Assim, o problema é dividido em duas partes: uma responsável pelo aprendizado das regras e a outra responsável pela otimização das funções de pertinência. A implementação do algoritmo para solucionar este problema, aplica os conceitos de Sistema Imunológico Artificial baseado em Gradiente com duas populações distintas de anticorpos: uma para aprendizagem e outra para otimização. Por meio do processo de co-evolução das duas populações é possível: a) trocar informações entre elas, uma vez que as mesmas são interdependentes, b) evitar o surgimento de ótimos locais e c) aumentar o *fitness* do sistema gerado.

O processo de otimização inicia com a geração de uma população de anticorpos, onde cada indivíduo é responsável por guardar informações sobre a posição e formato de todas as funções de pertinência associadas com as variáveis do sistema. Já o processo de aprendizado inicia com a análise do Banco de Dados Inicial (BDI) e a construção de uma Tabela de Regras Potenciais (TRP). A TRP é construída utilizando-se a idéia essencial apresentada em Wang e Mendel (1992), mas sem a abordagem relacionada a inconsistência das regras. Portanto, esta tabela conterá todas as possíveis regras necessárias para representar o banco de dados inicial. Continuando este processo, a segunda população de anticorpos deve decidir que regras da TRP podem ser usadas para melhor representação do BDI. Estas duas populações co-evoluem até o final do processo de geração de regras *fuzzy*.

A fase final é a aplicação do conjunto *fuzzy* no veículo autônomo. Visando validar o algoritmo proposto, o mesmo foi aplicado em manobras de veículos não-holonômicos

em estacionamento paralelo (baliza) e navegação, com o objetivo de aprender o sistema de controle *fuzzy* e realizar estas manobras de forma autônoma, a partir de um banco de dados de treinamento.

## 4.2 - Geração de um FIS a partir de um Banco de Dados Numérico

Considere a existência de um banco de dados com entradas e saídas numéricas que representam o funcionamento de uma determinada aplicação. O objetivo é gerar automaticamente um sistema *fuzzy* capaz de representar a relação entre as entradas e saídas desta aplicação. Tal sistema é composto por dois tipos de elementos: funções de pertinência e regras. As funções de pertinência transformam variáveis numéricas de/para variáveis *fuzzy* com um determinado grau de pertinência, enquanto que as regras, geralmente representadas no formato “SE ***Antecedente*** ENTÃO ***Consequente***”, expressam o domínio de conhecimento da aplicação. As funções de pertinência podem ser representadas de maneiras diferentes, variando em número, formato e posição. Encontrar sua melhor configuração depende do banco de dados e do banco de regras do sistema.

A literatura apresenta diversas propostas para geração de sistemas *fuzzy* a partir de banco de dados (LI, et al., 2002; PAL; CHAKRABORTY, 2001; QUINLAN, 1993; ALVES et al., 2004; WANG; MENDEL, 1992; PEI, 2006; ABE; LAN, 1995; ZHAO et al., 2003), que podem ser organizadas de acordo com três tipos de abordagens. Uma abordagem leva em consideração que as regras *fuzzy* são obtidas por meio de um especialista e utiliza algoritmos baseados em população para otimizar a função de pertinência (ZHAO et al., 2003). A segunda abordagem leva em consideração o pré-estabelecimento de funções de pertinência e o algoritmo gera as regras *fuzzy*. E finalmente, em uma terceira abordagem, alguns algoritmos utilizam duas populações distintas visando aprender tanto as regras quanto as funções de pertinência (LI, et al., 2002)

Como a otimização da funções de pertinência e o processo de inferência de regras são dependentes entre si, eles precisam ser modelados juntos para se atingir a melhor representação possível. Adotando este princípio, a proposta deste trabalho é o

desenvolvimento de um algoritmo baseado em duas populações que co-evoluem visando encontrar a melhor representação *fuzzy* para uma aplicação real. Uma população é responsável pela otimização da função de pertinência enquanto a outra é responsável por aprender as regras *fuzzy*.

Conforme já mencionado, regras *fuzzy* estão diretamente relacionadas e dependem fortemente das funções de pertinência (FP). É impossível extrair qualquer regra de uma base de dados se estas funções não tiverem sido definidas. Assim, o primeiro passo para a construção de um sistema *fuzzy* é a geração de uma população de funções de pertinência  $fmPop = \{fAb_1, \dots, fAb_n\}$ . Para cada indivíduo  $fAb_i$ , uma nova população  $rPop_i = \{r_iAb_1, \dots, r_iAb_m\}$  responsável pelo aprendizado das regras de inferência é criada e evolui utilizando Sistemas Imunológicos Artificiais. Após o aprendizado das regras, cada indivíduo  $fAb_i$  da primeira população terá sua função de pertinência otimizada. Este processo é mostrado na Figura 14 e continua até o final da simulação.

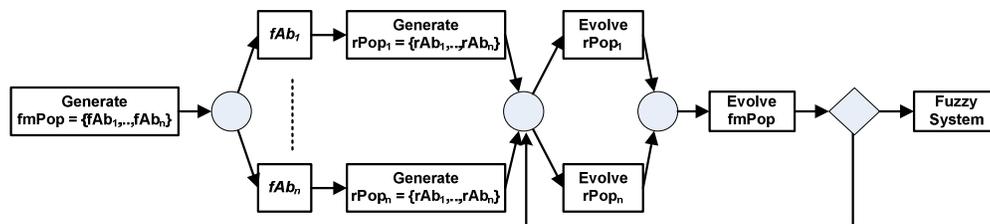


Figura 14 – Processo Co-evolucionário

Como o objetivo principal deste processo é representar de forma precisa um sistema real a partir de um banco de dados, o fitness do anticorpo  $fAb_i$  é dado por:

$$fitness(fAb_i) = \sum_{a=1}^{Ne} (fAb_i(ipDS_a) - opDS_a)^2 + Penalty(ipDS_a) \quad (4.1)$$

Onde:

- $Ne$  é o número de amostras do banco de dados,
- $ipDS_a$  representa o valor de saída da amostra  $a$ ,

-  $Penalty(.)$  é a uma função que retorna um valor muito alto (isto é  $10^{99}$ ) se  $fAb_i$  não possui nenhuma regra associada a  $ipDs_a$ .

A descrição de como cada população evolui individualmente é mostrada nas próximas seções, sendo que a Seção 4.3 apresenta a metodologia para o aprendizado da estrutura das regras *fuzzy* e a Seção 4.4 apresenta o detalhamento sobre a otimização das funções de pertinência.

### 4.3 - Aprendizado da Estrutura da Regra *Fuzzy*

Visando facilitar e enfatizar a explicação do método proposto será levado em consideração um banco de dados numérico com duas variáveis de entrada ( $x_1$  e  $x_2$ ) e uma saída ( $y$ ). As diversas amostras deste banco de dados são representadas como:

$$(x_1^1, x_2^1; y^1), (x_1^2, x_2^2; y^2), \dots, (x_1^n, x_2^n; y^n) \quad (4.2)$$

O primeiro passo no algoritmo proposto consiste em gerar uma Tabela de Regras Potenciais (TRP). Esta tabela deverá conter todas as possíveis regras *fuzzy* geradas a partir do BDI a partir de um conjunto de funções de pertinência. A TRP é gerada a partir da divisão dos valores numéricos de entrada e saída em regiões *fuzzy*, da mesma forma que a proposta feita em Wang e Mendel (1992). Cada intervalo de domínio, isto é, a faixa compreendida entre os valores mínimo e máximo das variáveis do BDI, deve ser dividido em  $2N+1$  regiões, onde  $N$  pode ser diferente para cada variável.

A Figura 14 mostra a divisão do intervalo de domínio em regiões *fuzzy*. Neste caso, o intervalo de domínio da variável de entrada  $x_1$  é dividido em três regiões ( $N=1$ ), a variável de entrada  $x_2$  em cinco regiões ( $N=2$ ) e a variável de saída  $y$  também em cinco regiões ( $N=2$ ). Após esta etapa, de acordo com Wang e Mendel (1992), cada amostra do BDI poderá gerar uma única regra *fuzzy* baseado no maior grau de pertinência de cada variável. Por exemplo, a Figura 14 mostra que  $x_1^1$  tem grau de pertinência de 0.6 em S1, 0.4 em M e zero nas demais regiões. Da mesma forma,  $x_2^1$  tem grau 1 em M e zero nas

demais regiões, enquanto  $y^1$  tem grau 0.8 em  $B_1$  e 0.2 em  $M$ . Assim, esta amostra gera a seguinte regra:

$$(x_1^1, x_2^1; y^1) \Rightarrow \text{IF } (x_1 \text{ is } S_1) \text{ AND } (x_2 \text{ is } M), \text{ THEN } (y \text{ is } B_1) \Rightarrow \quad (4.3)$$

Rule 1

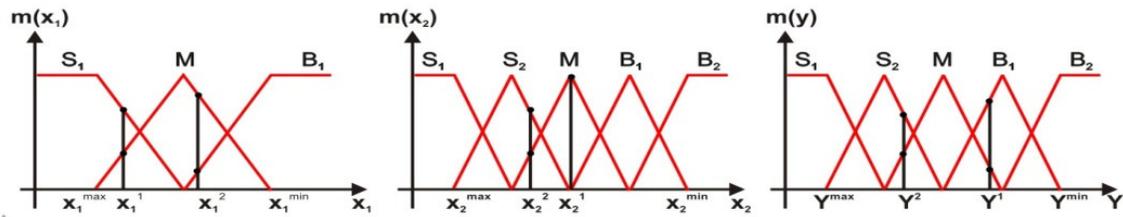


Figura 15 - Divisão do espaço de entradas e saídas em regiões *fuzzy*

É sabido que o valor numérico de saída de um sistema *fuzzy* depende das funções de pertinência ativadas com seu respectivo grau de pertinência. É importante destacar que todas as funções de pertinência ativadas, independentemente do grau de pertinência, contribuem no cálculo do valor de saída. Portanto, gerar regras *fuzzy* baseadas em grau de pertinência, como apresentado anteriormente, implica em fazer uma simplificação do problema. Ao invés de gerar regras baseadas em grau de pertinência, nossa proposta é gerar um conjunto de regras combinando todas as regiões *fuzzy* que possuem funções de pertinência ativadas, qualquer que seja o grau de pertinência. Assim, ao invés de gerar uma única regra *fuzzy* conforme mostrado na Equação 4.3, a amostra do banco de dados  $(x_1^1, x_2^1, y^1)$  vai gerar:

$(x_1^1, x_2^1; y^1) \Rightarrow \text{IF } x_1 \text{ is } S_1 \text{ AND } x_2 \text{ is } M, \text{ THEN } y \text{ is } B_1 \Rightarrow \text{Rule 1}$	(4.4)
$(x_1^1, x_2^1; y^1) \Rightarrow \text{IF } x_1 \text{ is } S_1 \text{ AND } x_2 \text{ is } M, \text{ THEN } y \text{ is } M \Rightarrow \text{Rule 2}$	
$(x_1^1, x_2^1; y^1) \Rightarrow \text{IF } x_1 \text{ is } M \text{ AND } x_2 \text{ is } M, \text{ THEN } y \text{ is } B_1 \Rightarrow \text{Rule 3}$	
$(x_1^1, x_2^1; y^1) \Rightarrow \text{IF } x_1 \text{ is } M \text{ AND } x_2 \text{ is } M, \text{ THEN } y \text{ is } M \Rightarrow \text{Rule 4}$	

A princípio, diversas regras conflitantes, isto é, regras que possuem o mesmo **Antecedente** (parte IF) mas, diferentes **Consequentes** (parte THEN), são geradas quando este procedimento é adotado. Para resolver este problema, a TRP é construída de forma que cada linha representa um único **Antecedente** e cada coluna da parte

**Consequente** represente a função de pertinência de saída. Cada **Antecedente** será associado a certo número de funções de pertinência ativas, sinalizadas com o dígito “1” na TRP.

Como exemplo, a Tabela 2 mostra a TRP geradas a partir das amostras  $(x_1^1, x_2^1; y^1)$  e  $(x_1^2, x_2^2; y^2)$  obtidas da Figura 19. Neste caso, a variável de saída  $y$  é representada por cinco colunas, S2, S1, M, B1, B2, que correspondem a suas funções de pertinência. Aqui, apenas duas amostras do BDI geraram onze regras potenciais, enquanto no método baseado em grau de pertinência apenas duas regras potenciais são geradas. É importante notar que nem todas as regras potenciais farão parte da solução final. A solução final será definida por meio de otimização combinatória utilizando o algoritmo CGbAIS apresentado no Capítulo 3 desta tese.

Tabela 2 – Tabela de Regras Potenciais - TRP

Antecedent	Consequent – Output $y$				
	S2	S1	M	B1	B2
<b>IF</b> $x_1$ is S1 <b>AND</b> $x_2$ is M <b>THEN</b>	0	0	1	1	0
<b>IF</b> $x_1$ is M <b>AND</b> $x_2$ is M <b>THEN</b>	0	1	1	1	0
<b>IF</b> $x_1$ is M <b>AND</b> $x_2$ is S1 <b>THEN</b>	0	1	1	0	0
<b>IF</b> $x_1$ is B1 <b>AND</b> $x_2$ is S1 <b>THEN</b>	0	1	1	0	0
<b>IF</b> $x_1$ is B1 <b>AND</b> $x_2$ is M <b>THEN</b>	0	1	1	0	0

#### 4.3.1 – Aplicação do algoritmo CGbAIS

Uma das características do algoritmo proposto é a criação da TRP, onde cada coluna representa a parte **Consequente** da regra e cada linha a parte **Antecedente**, levando-se em consideração todas as funções de pertinência ativadas independentemente do seu grau de pertinência. Esta abordagem melhora o algoritmo original em dois aspectos: trabalha com um número de regras capaz de melhor representar o sistema e evita regras inconsistentes por meio da implementação das saídas em colunas na TRP. Esta última afirmação deve-se ao fato de apenas haver inconsistência de regras quando, na TRP, a parte **Antecedente** apresenta mais de um **Consequente** associado. Como o algoritmo de otimização será responsável por selecionar regras da TRP, torna-se fácil prevenir situações de inconsistência. A utilização do CGbAIS neste domínio é mostrado na Figura 16 e explicada a seguir.

O primeiro passo, dada a TRP com  $nr$  regras potenciais, é selecionar cada uma delas, gerando  $nr$  anticorpos. O segundo passo é executar o processo de hipermutação, e para isto é necessário calcular o Vetor Tangente (VT). Um número aleatório de regras será selecionado, também aleatoriamente, da TRP, adicionado ao clone hipermutado (isto é  $hCab1_1$ ) e evoluído. O processo de seleção começa e, utilizando os valores de  $nLocalbest$  clones de um determinado anticorpo, o VT evolui de acordo com a Equação 4.5. Então, os  $mGlobalbest$  clones vão gerar uma nova população. O processo continua de acordo com a Figura 16.

$$TV = v_i \times \Phi\{v_1, \dots, v_{nLocalbest}\} \tag{4.5}$$



Figura 16 – CGbAIS aplicado em otimização discreta

### 4.4 - Otimização das Funções de Pertinência

Na otimização das funções de pertinência, cada anticorpo representa um conjunto de vetores contendo as coordenadas das funções de pertinência. Um exemplo é apresentado na Figura 17, onde  $Ab = \{A_1, B_1, C_1, A_2, B_2, C_2, A_3, B_3, C_3\}$ . O processo de otimização também é feito com o algoritmo CGbAIS e o resultado é o melhor ajuste de cada posição do vetor. Considerações sobre os limites de cada variável também podem ser observadas na Figura 17.

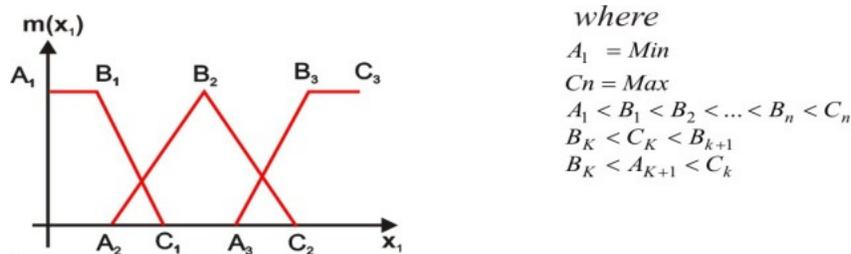


Figura 17 – Coordenadas da Função de Pertinência

## Capítulo 5

# Aplicação e Resultados Obtidos

### 5.1 – Considerações Iniciais

Visando validar o algoritmo proposto nesta tese, este capítulo apresenta resultados reais e de simulação para dois tipos distintos de problemas: 1) estacionamento em vaga paralela ou baliza; e 2) navegação em um circuito mapeado com *waypoints*.

Inicialmente o algoritmo foi testado em um ambiente de simulação 3D (HONÓRIO, 2005) que permite reproduzir a cinemática do veículo conforme descrito no Apêndice A desta tese. Alguns dispositivos, tais como, GPS, sensores ultrasônicos e laser foram modelados matematicamente para perceber o ambiente em torno do veículo. No ambiente de simulação, o motorista virtual realiza manobras por meio de um *joystick*, enquanto o sistema de percepção armazena em um banco de dados informação dos sensores e das ações do motorista, tais como: movimento da direção, aceleração e frenagem. Aplicando-se o algoritmo proposto, o FIS para controle do veículo é gerado automaticamente. Este sistema *fuzzy* é posteriormente aplicado para controle do veículo no ambiente de simulação.

Após a validação da proposta de aprendizado supervisionado no simulador, o mesmo procedimento foi repetido em um veículo real, descrito na Seção A.6 do Apêndice A desta tese. Em ambos os casos o objetivo é aprender as regras de direção com o motorista humano, baseado em exemplos de treinamento.

## 5.2 – Estacionamento Paralelo: Resultados de Simulação

A Figura 18 ilustra o ambiente de simulação utilizado, assim como as variáveis de entrada que foram consideradas no problema de estacionamento paralelo ou baliza.

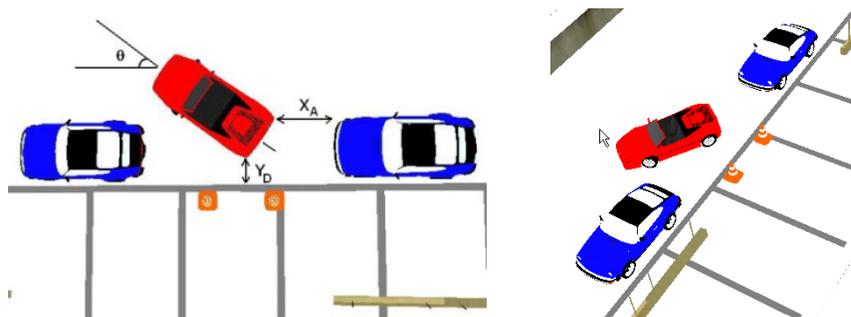


Figura 18 –Variáveis consideradas no problema de estacionamento paralelo

Este tipo de manobra realizada pelo motorista humano leva em consideração a distância do veículo em relação aos limites da vaga e o ângulo do próprio veículo conforme ilustrado na Figura 18. O problema pode ser descrito resumidamente como: movimentar o veículo de ré virando a direção a partir uma posição inicial paralela a vaga até atingir a posição dentro da vaga paralela a posição inicial.

O procedimento de estacionamento paralelo foi dividido em três etapas, descritas a seguir:

1. Procedimento de entrada na vaga (EV): o carro movimenta-se de ré, em velocidade baixa e constante, entrando na vaga até que esteja muito próximo de um de seus limites;
2. Procedimento de ajuste dentro da vaga para frente (AF): estando dentro da vaga, o carro movimenta-se para frente em velocidade baixa e constante até que atinja a posição final desejada ou um dos limites da vaga;
3. Procedimento de ajuste dentro da vaga para trás (AT): estando dentro da vaga, o carro movimenta-se para trás em velocidade baixa e constante até que atinja a posição final desejada ou um dos limites da vaga.

Os procedimentos 2 e 3 são repetidos até que o carro esteja corretamente estacionado (CE) ou até que um determinado número de manobras tenha sido realizado (CNE). O diagrama de estados apresentado na Figura 19 ilustra esta estratégia. Os procedimentos EV, AF e AT são implementados por controladores *fuzzy*. As variáveis de entrada dos controladores *fuzzy* são  $X_A$ ,  $Y_D$ , que correspondem à posição relativa da traseira do carro em relação a um dos limites da vaga (Figura 18) e o ângulo do carro representado por  $\theta$ . A saída do controlador *fuzzy* atua na direção do veículo, determinando o ângulo das rodas do carro durante a manobra.

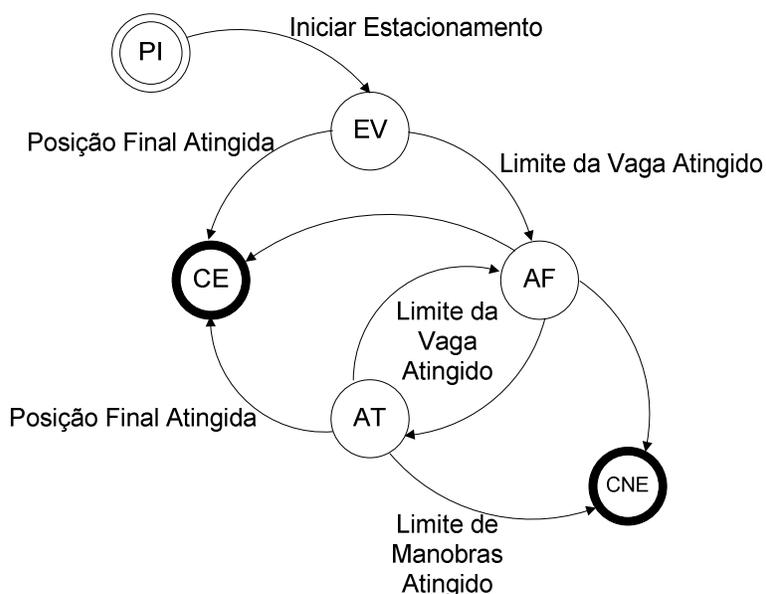


Figura 19 –Diagrama de Estados para Estacionamento Paralelo

Para construir o banco de dados com amostras das variáveis envolvidas neste sistema de controle, a manobra de estacionamento paralelo foi realizada por um operador no simulador utilizando-se um *joystick*. Durante este processo, as variáveis de entrada  $X_A$ ,  $Y_D$  e  $\theta$ , assim como o valor de saída (Direção) obtido por meio de um encoder virtual conectado na barra de direção são armazenados em um banco de dados, gerando os exemplos de treinamento.

Para induzir um sistema *fuzzy* capaz de representar o controle humano sobre o veículo o algoritmo proposto foi utilizado. O primeiro passo é gerar uma população inicial de funções de pertinência e para cada uma gerar a TRP.

Para ilustrar este processo, a Figura 20(a) mostra a configuração inicial das funções de pertinência de um anticorpo. Para se ter parâmetros de comparação, as funções de pertinência foram geradas conforme proposto em Wang e Mendel (1992). Para este anticorpo, a TRP é mostrada na Tabela 3, onde 22 regras potenciais foram consideradas. A utilização da metodologia proposta por Wang e Mendel (1992) levaria a geração de apenas 6 regras potenciais, conforme mostrado na Tabela 5. Apesar do número menor de regras, elas não são suficientes para representar adequadamente o sistema.

À medida que a população evolui, as funções de pertinência e as regras assumem novos formatos apresentando melhores resultados. A Figura 20(b) mostra o resultado da evolução do algoritmo após a primeira geração, onde é possível observar que a função de pertinência  $M_{XA}$  assume um formato trapezoidal. A Figura 20(c) e a Tabela 5 mostram o resultado final do aprendizado das funções de pertinência e das regras obtidos após três gerações.

É importante notar que o número final de regras obtido com a aplicação deste algoritmo foi de apenas cinco. Embora seja apenas uma regra a menos em relação ao número de regras obtido em Wang e Mendel (1992), as regras são diferentes e mais precisas, como pode ser visto na Figura 21, onde o eixo  $y$  representa o valor do *encoder* virtual e o eixo  $x$  as diversas amostras do BDI. Uma das explicações para este resultado pode ser encontrada na comparação entre os valores **Conseqüentes** das Tabelas 4 e 5. Note que o método Wang&Mendel utiliza apenas duas funções de pertinência (S e B)

enquanto o algoritmo proposto utiliza três (S, M e B), provendo uma transição mais suave entre os estados do sistema.

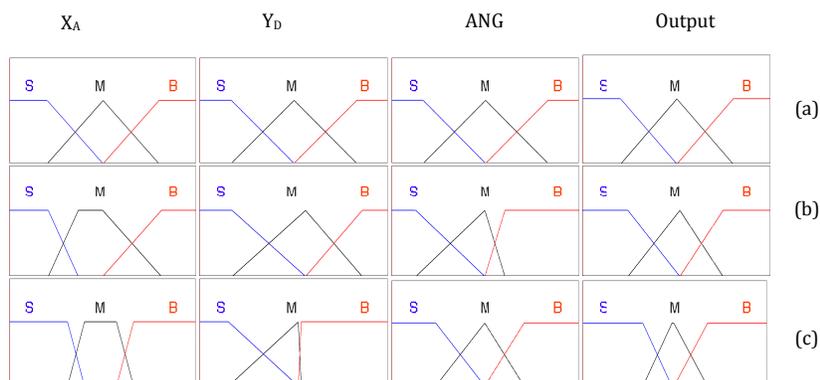


Figura 20 - (a) Funções de Pertinência (FP) sem otimização (b) FP otimizada (c) FP otimizadas após co-evolução.

Tabela 3 – Tabela de Regras Potenciais – Estacionamento Paralelo

Antecedente							Consequente		
							E	C	D
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = S$	<b>ENTÃO</b>	0	1	1
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = M$	<b>ENTÃO</b>	0	1	1
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	0	0	1
<b>SE</b>	$X_A = M$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	1	1	1
<b>SE</b>	$X_A = M$	<b>E</b>	$Y_D = M$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	1	1	1
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = M$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	0	0	1
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = M$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	1	1	1
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	1	1	1
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = S$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	1	1	0
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = S$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	1	1	0

Tabela 4 – Regras obtidas pelo método Wang e Mendel

Antecedente							Consequente
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = M$	<b>ENTÃO</b>	DIREÇÃO = D
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = D
<b>SE</b>	$X_A = M$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = D
<b>SE</b>	$X_A = M$	<b>E</b>	$Y_D = M$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = D
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = M$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = E
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = S$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = E

Tabela 5 – Regras obtidas por meio do algoritmo proposto

Antecedente							Consequente
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = S$	<b>ENTÃO</b>	DIREÇÃO = C
<b>SE</b>	$X_A = B$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = D
<b>SE</b>	$X_A = M$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = D
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = B$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = C
<b>SE</b>	$X_A = S$	<b>E</b>	$Y_D = S$	<b>E</b>	$ANG = B$	<b>ENTÃO</b>	DIREÇÃO = E

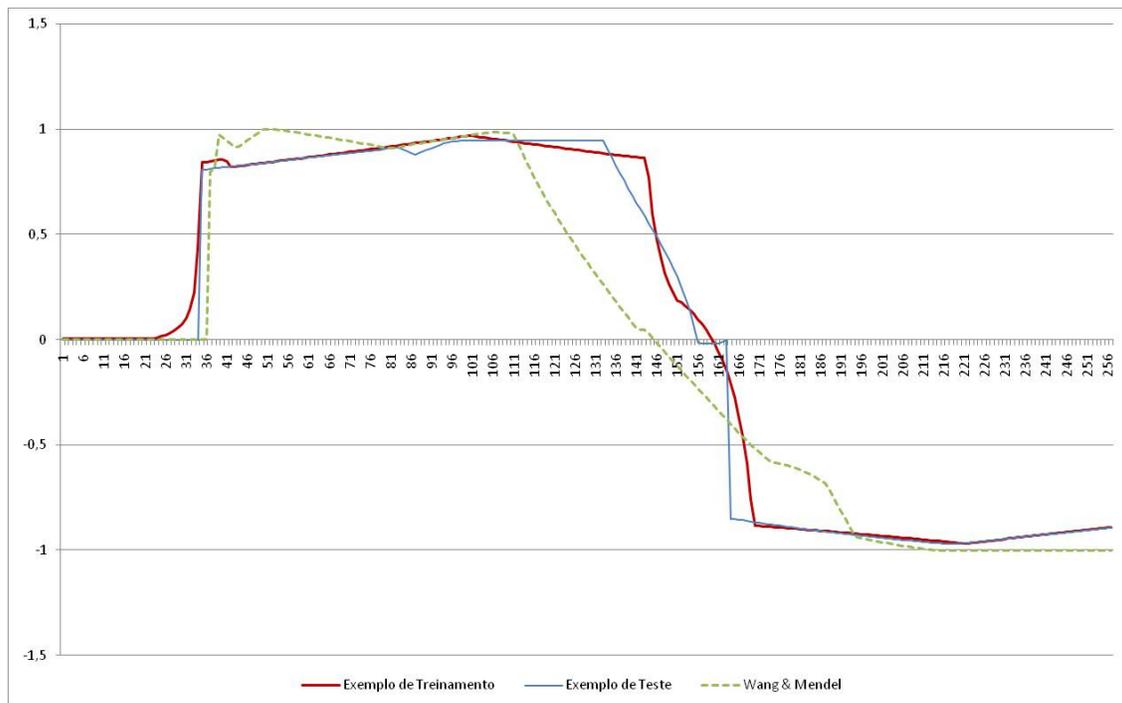


Figura 21 – Saída Original da base de dados (Exemplo de Treinamento), resultado do algoritmo proposto (Exemplo de Teste) e resultado do método Wang & Mendel.

O erro médio quadrático entre as funções “Exemplo de Teste” gerada pelo algoritmo proposto e “Exemplo de Treinamento” é igual a 0,038, enquanto que este erro equivale a 0,102 para a função gerada pelo método Wang e Mendel (1992).

O gráfico apresentado na Figura 22 ilustra o deslocamento do veículo durante a manobra de estacionamento utilizando o FIS gerado pelo algoritmo proposto nesta tese. As coordenadas locais X e Y são representadas em metros, sendo a posição inicial do veículo o ponto (5.6 , 2.0) e a posição final (0.0 , 0.0).

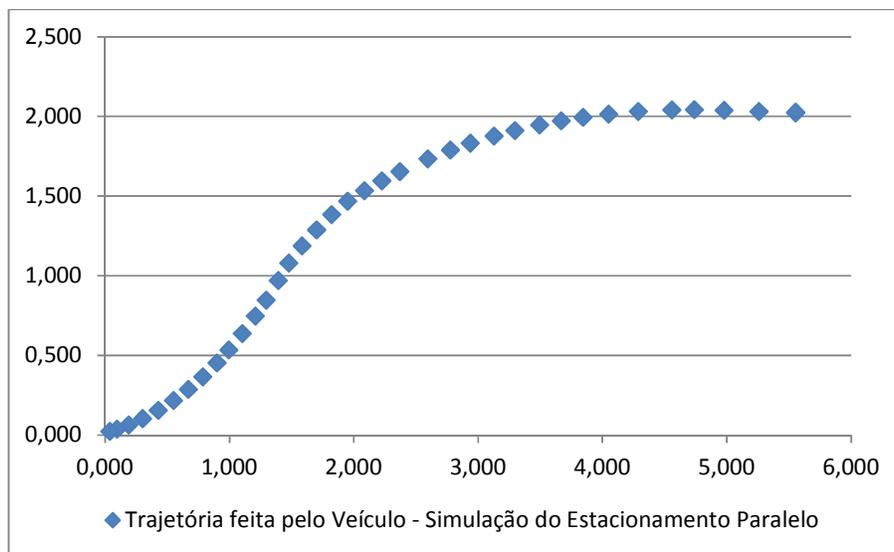


Figura 22 – Gráfico  $X_A \times Y_D$  da Manobra de Estacionamento Paralelo – Simulação  
Unidade: metro

Visando comparar os resultados obtidos aplicando-se o algoritmo proposto nesta tese com uma proposta neuro-nebulosa bastante difundida na literatura, o treinamento de um novo FIS foi treinado com o ANFIS. A função ANFISEEDIT do software MATLAB 6.0 (MATLAB, 2010) foi então utilizada par treinar um sistema neuro-*fuzzy* tipo TSK usando os mesmos dados de treinamento do caso anterior. Foi utilizada a facilidade da função ANFISEEDIT onde o treinamento é realizado até que o número de interações ou o erro mínimo sobre o conjunto de teste seja atingido.

A Figura 23 ilustra a tela ANFISEEDIT do Matlab com os gráficos dos dados de treinamento e saída do FIS treinado pelo ANFIS. O número de *funções de pertinência* ajustado para o treinamento com o ANFIS foi o mesmo ajustado para o algoritmo proposto nesta tese, ou seja, três *funções de pertinência*. O erro médio quadrático obtido neste caso foi de 0,018. Por se tratar de uma abordagem que privilegia a acuidade dos resultados, era esperado que este erro fosse mais baixo. No entanto, os erros do algoritmo proposto e do ANFIS representam 1,9% e 0,9%, respectivamente, da faixa de variação possível dos valores de saída (direção). Tais valores são considerados mais do que satisfatórios para o tipo de aplicação. Em contrapartida, a melhor acuidade do ANFIS é obtida com um FIS cuja interpretabilidade é muito pequena, por possuir uma estrutura composta de 27 regras conforme ilustrado na Figura 24 e utilizar o modelo TSK.

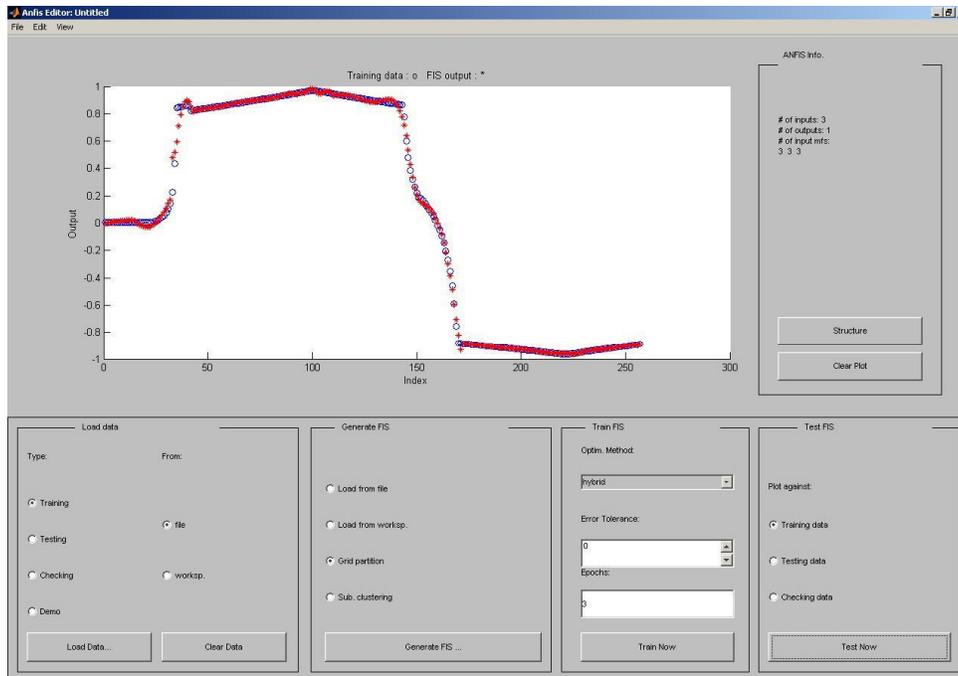


Figura 23 – Tela ANFISEDIT do Matlab para treinamento do FIS para baliza.

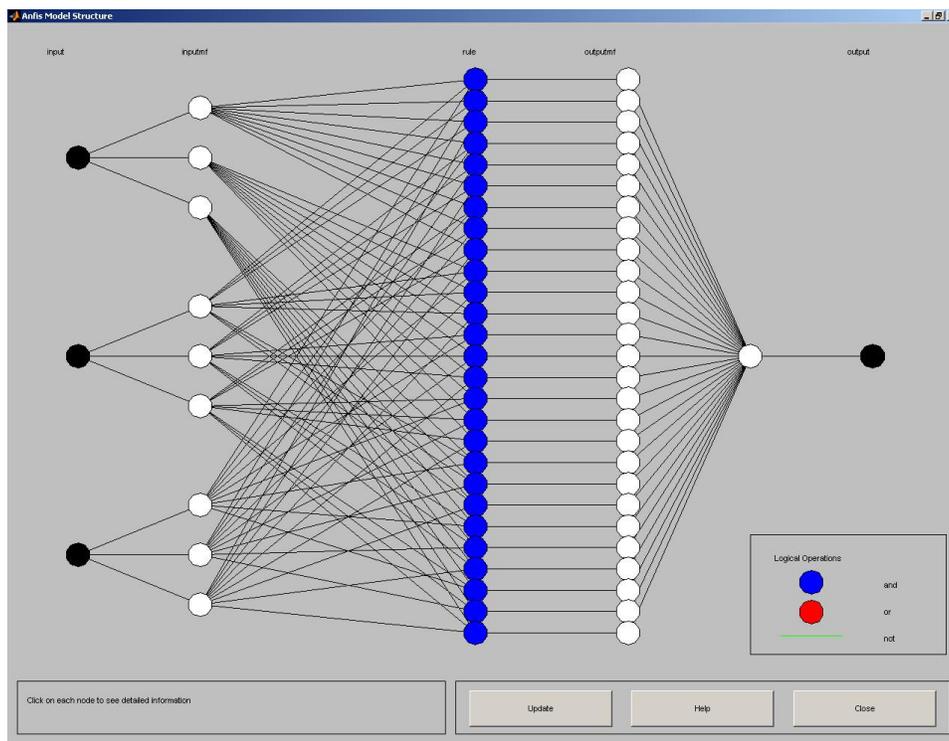


Figura 24 – Estrutura de Regras geradas pelo ANFIS – 27 regras geradas

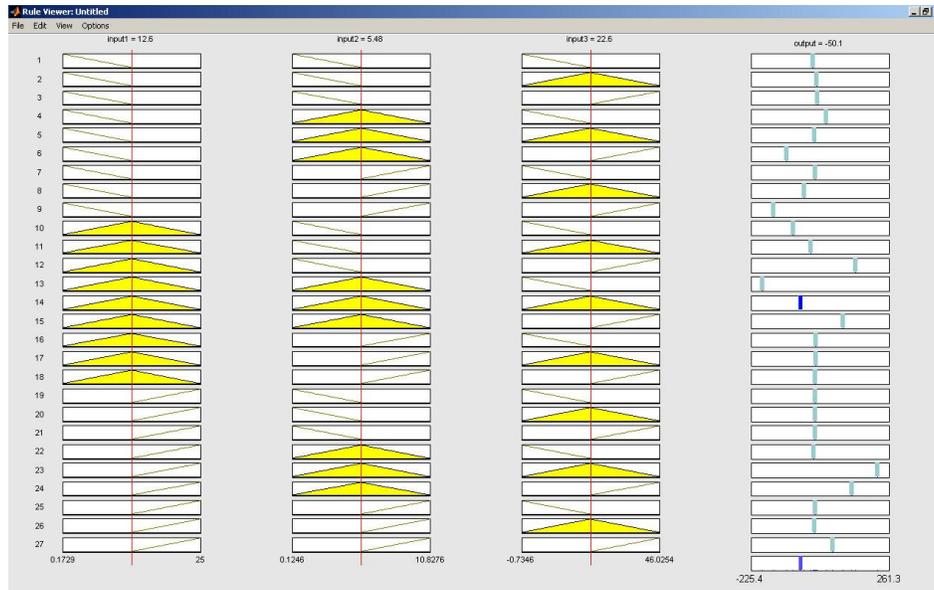


Figura 25 – Funções de Pertinência no ANFIS

Conforme já mencionado, uma das premissas desta proposta é o desenvolvimento de um algoritmo para geração automática de FIS com boa interpretabilidade e acuidade que atenda os requisitos da aplicação. Os resultados de simulação comprovaram estes requisitos.

A Figura 26 ilustra a sequência de manobras de estacionamento realizadas automaticamente no simulador 3D aplicando o FIS gerado pelo algoritmo proposto nesta tese.

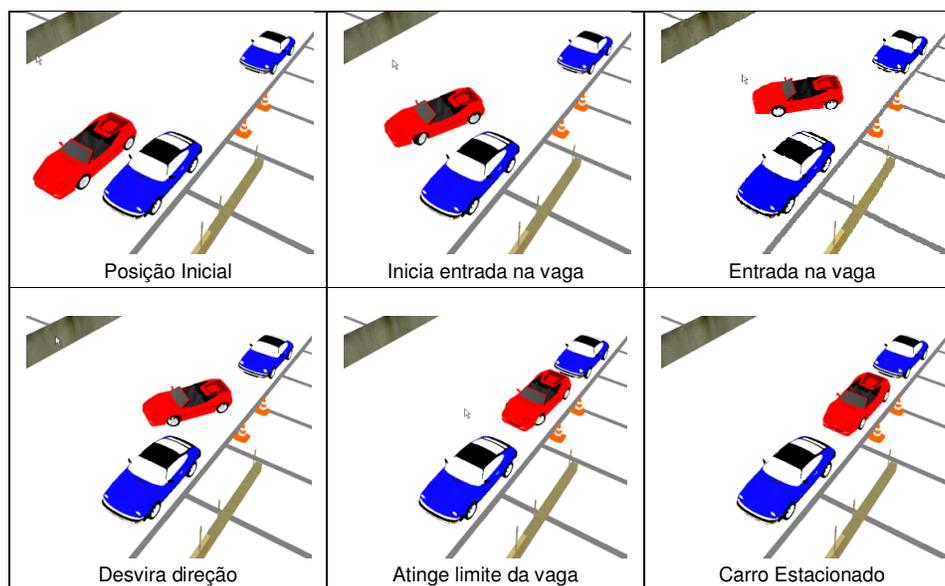


Figura 26 - Sequência de manobras no simulador para estacionamento em vaga paralela.

### 5.3 – Estacionamento Paralelo: Resultados Reais

Procedimento semelhante ao descrito na Seção 5.2 também foi aplicado em um veículo real, uma Zafira – GM automática, equipada conforme descrição detalhada na Seção A.6 do Apêndice A desta tese. Devido a algumas restrições relacionadas aos equipamentos disponíveis para as implementações práticas, mais especificamente o sistema de percepção do veículo, foram realizadas pequenas modificações na estratégia adotada no problema de estacionamento no simulador. As variáveis  $X_A$ ,  $Y_D$  e  $\theta$  utilizadas como entrada do FIS, são estimadas utilizando o modelo cinemático do veículo descrito na Seção A.5 do Apêndice A. Apesar da incerteza na estimativa destas variáveis, natural em aplicações que utiliza o modelo de um sistema real, os testes práticos mostraram que os erros medidos no deslocamento do veículo durante a manobra de estacionamento paralelo são inferiores a 0,5%, valor satisfatório para a aplicação em questão.

O banco de dados para treinamento deste controlador *fuzzy* foi gerado a partir de diversas manobras de estacionamento bem sucedidas realizadas por um motorista humano. Desta forma, os exemplos de treinamento consideram possíveis variações nesta manobra. A Figura 27 apresenta um gráfico com os exemplos de treinamento utilizados para aprendizado do controlador *fuzzy* responsável pelo estacionamento paralelo.

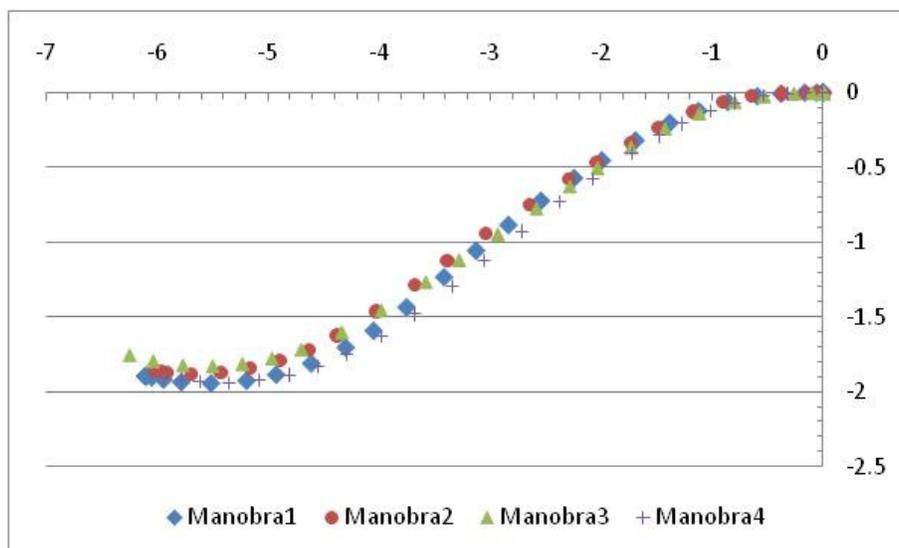


Figura 27. Exemplos de Treinamento:Estacionamento Paralelo -

A estrutura de regras obtida a partir dos exemplos de treinamento apresentados na Figura 27 é apresentada na Tabela 6 e as funções de pertinência ilustradas na Figura 28.

Tabela 6 – Regras obtidas para estacionamento paralelo

Antecedente	Consequente
SE $X_A = G$ E $Y_D = G$ E $\theta = P$ ENTÃO	Direção = C
SE $X_A = G$ E $Y_D = G$ E $\theta = G$ ENTÃO	Direção = D
SE $X_A = M$ E $Y_D = G$ E $\theta = G$ ENTÃO	Direção = D
SE $X_A = P$ E $Y_D = G$ E $\theta = G$ ENTÃO	Direção = C
SE $X_A = P$ E $Y_D = P$ E $\theta = G$ ENTÃO	Direção = E

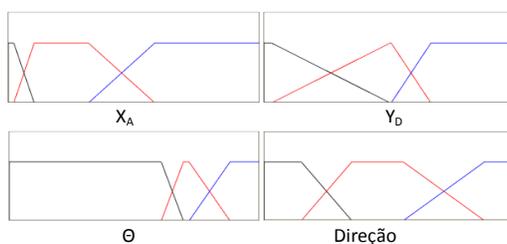


Figura 28 - Funções de Pertinência para Controlador *Fuzzy* de Estacionamento

A aplicação deste controlador *fuzzy* em uma situação real de estacionamento paralelo mostrou a eficácia do sistema *fuzzy* gerado por meio do algoritmo proposto nesta tese. O vídeo *online*<sup>1</sup> gravado durante os testes de estacionamento paralelo autônomo mostra esta manobra sendo realizada pelo IV. A Figura 29 ilustra algumas etapas desta manobra.

<sup>1</sup> <http://dl.dropbox.com/u/14518596/Estacionamento.wmv>

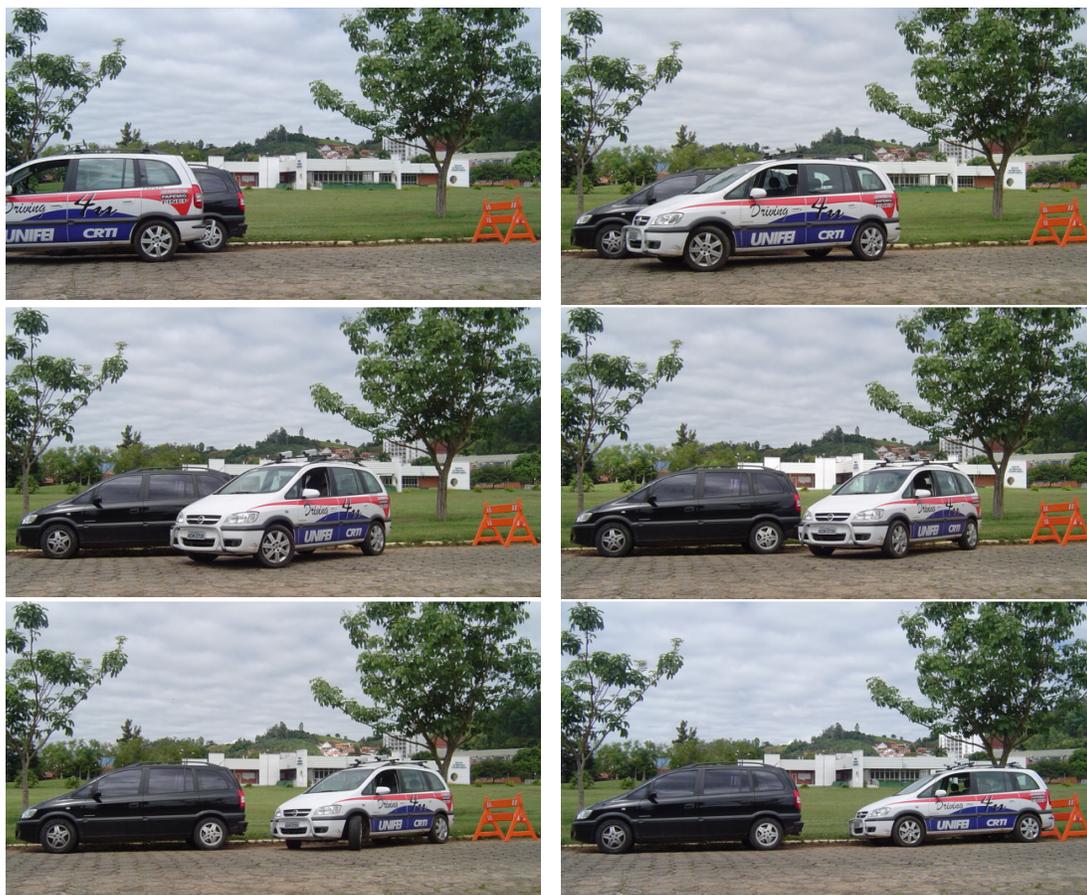


Figura 29 - Estacionamento paralelo executado de forma autônoma utilizando o FIS aprendido com o motorista humano.

## 5.4 – Navegação em um Circuito com *Waypoints*: Resultados de Simulação

Outra manobra utilizada para validar o algoritmo é a navegação em um circuito com *waypoints* utilizando um sistema de controle de direção. O objetivo do Sistema de Controle de Direção é permitir que o IV seja capaz de seguir uma trajetória de uma posição de origem a outra de destino. Esta trajetória é determinada por pontos, chamados *Waypoints*, que representam coordenadas globais válidas e alcançáveis. O IV deverá ser capaz de trafegar de uma posição global inicial para uma posição global objetivo seguindo os *Waypoints*.

Para modelar o desvio lateral e angular percebido pelo motorista humano em relação à rota a ser seguida, utilizou-se a variável *fuzzy* **Erro\_Angular**. Esta variável consiste na diferença de ângulo entre dois vetores: o vetor ângulo do carro, representado por  $\alpha_{carro}$  e o vetor ângulo da trajetória  $\beta_{trajetória}$ . O **Erro\_Angular** pode assumir os valores linguísticos **Esquerda** ou **Direita**.

A Figura 30 ilustra os vetores ângulo do carro, ângulo da trajetória e a variável **Erro\_Angular**.

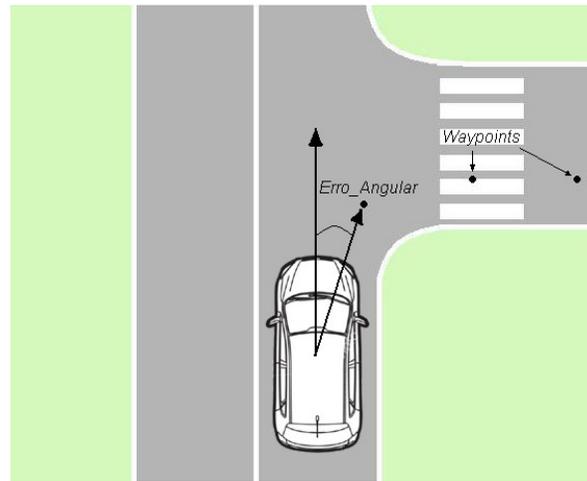


Figura 30. Vetores ângulo do carro, da trajetória e Erro\_angular

O cálculo destes ângulos utiliza as seguintes equações:

$$\alpha_{carro} = \arctg \left( \frac{Yc_{atual} - Yc_{anterior}}{Xc_{atual} - Xc_{anterior}} \right) \quad (2)$$

$$\beta_{trajetória} = \arctg \left( \frac{Yt_{próximo} - Yc_{atual}}{Xt_{próximo} - Xc_{atual}} \right) \quad (3)$$

Onde:

$Xc_{atual}$ ,  $Yc_{atual}$  = coordenadas da posição atual do carro medidas pelo DGPS;

$Xc_{anterior}$ ,  $Yc_{anterior}$  = coordenadas da posição anterior do carro medidas pelo DGPS ;

$Xt_{próximo}$ ,  $Yt_{próximo}$  = coordenadas do próximo ponto da trajetória;

Os ângulos são corrigidos de forma que eles assumam apenas valores positivos. Desta forma, um vetor apontando para o norte aponta para  $0^{\circ}$ , aumentando gradativamente em sentido anti-horário. Finalmente, o **Erro\_Angular**, é calculado por:

$$Erro\_Angular = \beta_{trajetória} - \alpha_{carro} \quad (4)$$

Foram adotados valores positivos representando que o carro deva virar à esquerda e valores negativos caso deva virar à direita.

As variáveis do sistema de controle de navegação são computadas utilizando dados de um DGPS e do mapa contendo os *Waypoints* da trajetória a ser seguida. A variável de saída do controlador *fuzzy* é **Direção** e indica a direção que o volante do veículo deve ser virado para corrigir o erro de entrada. Da mesma forma, a variável **Direção** assume os valores linguísticos **Esquerda** ou **Direita**, indicando respectivamente que o volante deve virar no sentido anti-horário ou horário.

Para gerar o banco de dados de treinamento de navegação, o veículo foi dirigido no ambiente de simulação conforme trajetória apresentada na Figura 31. Este circuito fechado em formato de “oito” possui exemplos de treinamento com duas retas e curvas à esquerda e à direita. Esta diversidade é importante para o aprendizado do sistema *fuzzy*.

O banco de dados de treinamento gerado neste circuito é utilizado para gerar o FIS para controle de navegação do IV aplicando-se o algoritmo proposto nesta tese.

Tabela 7 - Regras obtidas para navegação no simulador

Antecedente	Consequente
SE Erro_Angular = E ENTÃO	Direção = D
SE Erro_Angular = D ENTÃO	Direção = E
SE Erro_Angular = C ENTÃO	Direção = C

A comparação da Figura 31 e Figura 32 que mostra respectivamente a trajetória executada pelo IV e o ângulo da roda do veículo permite identificar as ações do FIS durante o seu deslocamento em cada trecho do circuito.

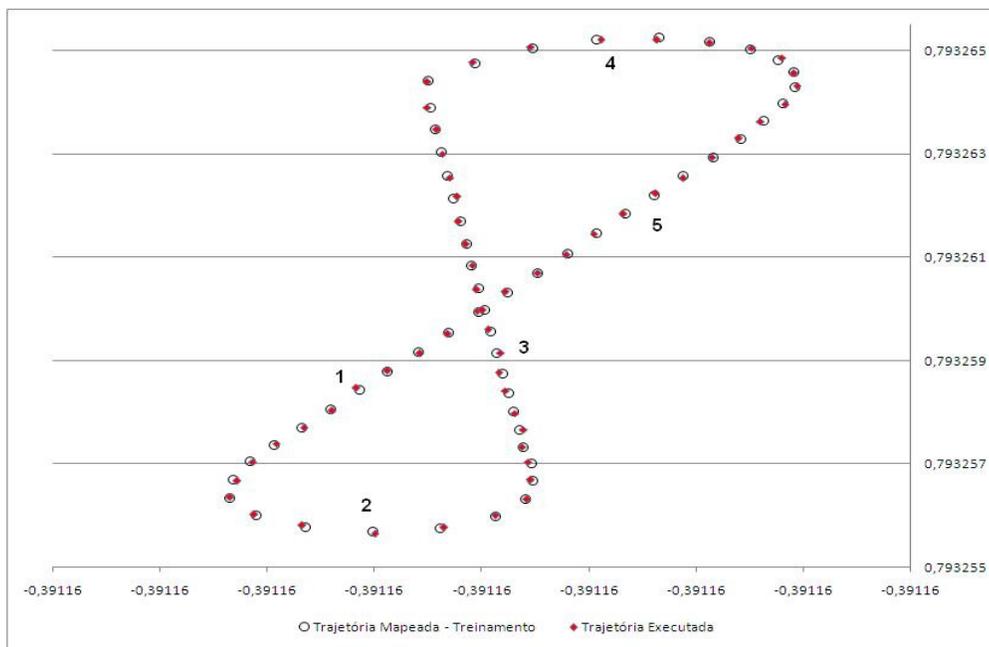


Figura 31 – Trajetória Mapeada e Executada - Resultado de Simulação

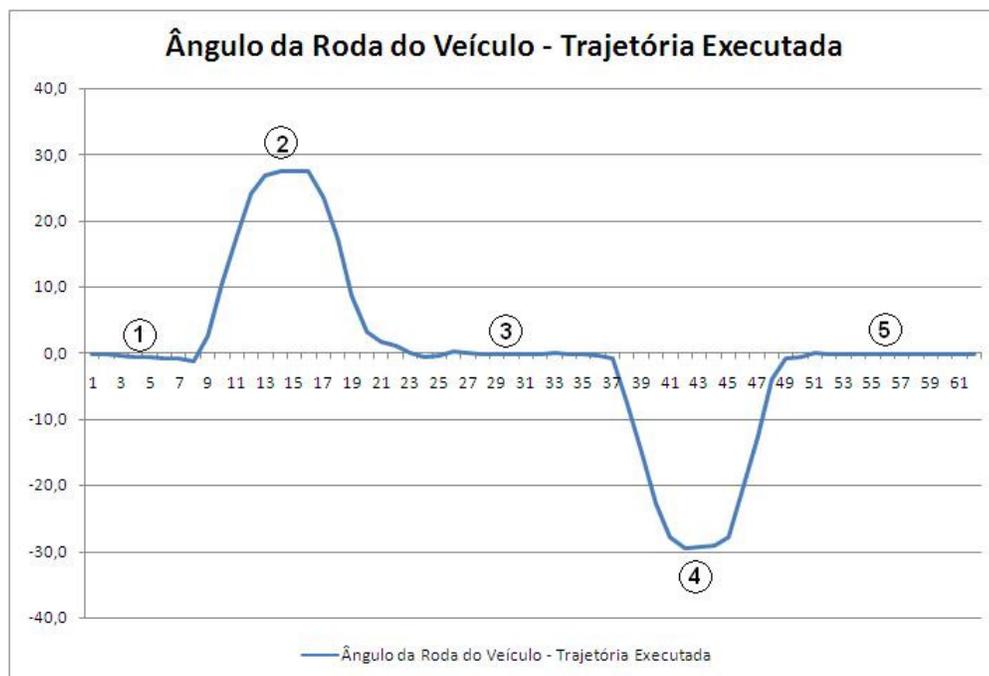


Figura 32 – Ângulo da roda do veículo durante trajetória executada.

## 5.5 - Navegação em Circuito com *Waypoints*: Resultados Reais

Aplicando-se o mesmo procedimento utilizado no simulador, descrito na Seção 5.4, o veículo real foi dirigido em uma pista de teste no Campus da UNIFEI para obtenção dos exemplos de treinamento do FIS. A trajetória feita pelo motorista para geração do banco de dados de treinamento, neste caso, é ilustrada na Figura 33. Este circuito apresenta exemplos de treinamento com retas, curvas acentuadas e curvas leves à esquerda e à direita.



Figura 33. Circuito utilizado para treinamento do veículo

As regras *fuzzy* obtidas a partir destes exemplos de treinamento são apresentadas na Tabela 8 e as funções de pertinência ilustradas na Figura 34.

Tabela 8 - Regras obtidas para navegação

Antecedente		Consequente
SE Erro_Angular = E	ENTÃO	Direção = D
SE Erro_Angular = D	ENTÃO	Direção = E
SE Erro_Angular = C	ENTÃO	Direção = C

O vídeo *online*<sup>2</sup> mostra o IV executando a navegação autônoma no circuito mapeado no Campus da UNIFEI.

<sup>2</sup> [http://dl.dropbox.com/u/14518596/Driving4u\\_Navegacao.wmv](http://dl.dropbox.com/u/14518596/Driving4u_Navegacao.wmv)

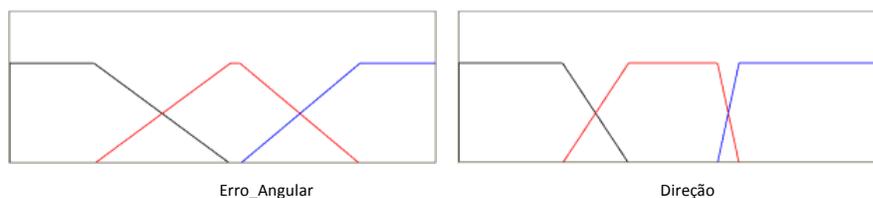


Figura 34. Funções de Pertinência para Controlador *Fuzzy* de Navegação

O FIS gerado para navegação do IV, utilizando as regras e funções de pertinência apresentados na Tabela 8 e na Figura 34, respectivamente, foi aplicado em um circuito diferente do utilizado para o treinamento. A Figura 35 apresenta uma foto do circuito de 800m no Campus da UNIFEI, identificado pelos *waypoints* que definem a trajetória a ser seguida pelo IV.

Durante o percurso as coordenadas GPS da trajetória executada automaticamente pelo veículo foram registradas. A comparação das trajetórias mapeada e executada pelo veículo é apresentada na Figura 36. Durante este percurso a velocidade média do veículo foi de 15 km/h.



Figura 35. Circuito de 800m de extensão utilizado para teste do FIS gerado.

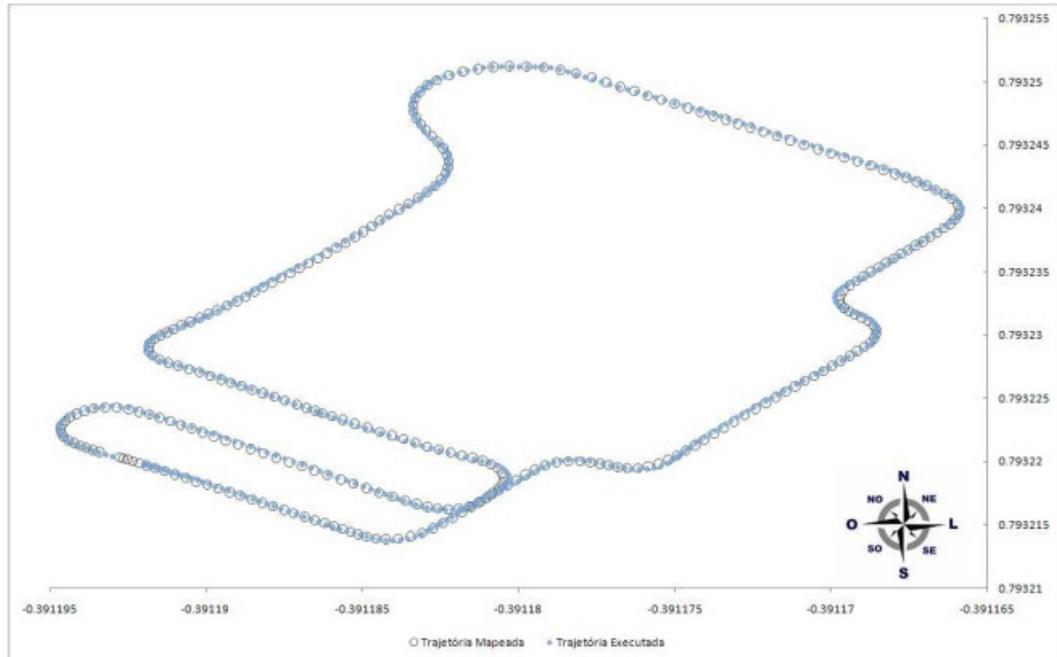


Figura 36. Comparação das Trajetórias Mapeada e Executada

A comparação dos trechos indicados no circuito da Figura 35 e o ângulo da roda mostrado na Figura 37, possibilita uma análise das ações do IV em todo o percurso.

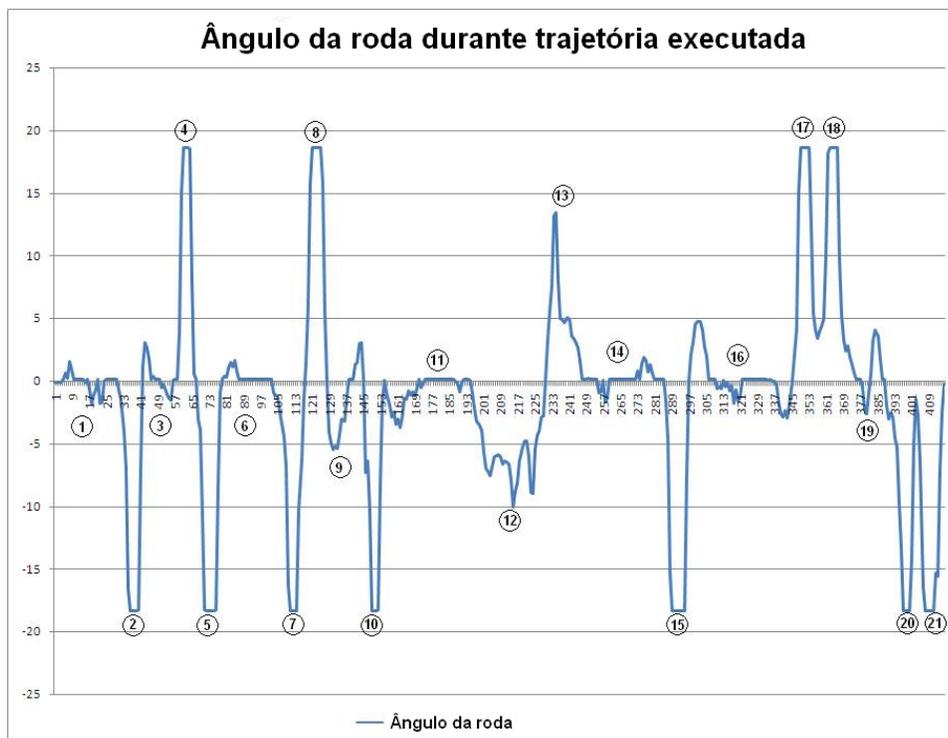


Figura 37 - Ângulo da Roda durante Trajetória Executada

## Capítulo 6

# Conclusões

### 6.1 - Conclusões deste Trabalho

O desenvolvimento do projeto de pesquisa “Técnicas de Automação de Manufatura Industrial com a Integração de Fornecedores e Clientes por meio de um Sistema de Veículos Não Tripulados”, no âmbito do Grupo de Automação e Tecnologias da Informação da Universidade Federal de Itajubá, conduziu à identificação do tema proposto nesta tese de doutorado. A revisão bibliográfica na área de Veículos Não Tripulados, também tratados na literatura como Veículos Autônomos ou Veículos Inteligentes, mostrou que muito pouco tem sido pesquisado sobre técnicas de aprendizado de máquina no controle de alto nível destes veículos. Os algoritmos de controle para funções de localização, navegação, ultrapassagem, mudança de pista, detecção de colisão e estacionamento, entre outras manobras, empregam técnicas de controle clássico ou de inteligência artificial programadas por um especialista.

Esta tese propôs um algoritmo para aprendizado supervisionado de um sistema de inferência *fuzzy* (FIS) aplicado a veículos inteligentes. O algoritmo possibilita o aprendizado do controlador de alto nível baseado no comportamento do motorista humano. A metodologia proposta consistiu na obtenção de um banco de dados de treinamento por meio do sistema de percepção do veículo, a geração do FIS a partir deste banco de dados e a aplicação do FIS gerado automaticamente para controlar o veículo de forma autônoma. A validação do algoritmo foi feita em um ambiente de simulação 3D e em um veículo de passeio real. Em ambos os casos foram considerados dois problemas distintos: navegação autônoma do veículo em um circuito com *waypoints* e estacionamento paralelo.

O problema de aprendizado automático de um FIS a partir de um banco de dados numérico consiste resumidamente na geração da estrutura de regras *fuzzy* e no ajuste das funções de pertinência. Este problema foi tratado como um problema de busca de regras em um espaço de soluções e otimização das funções de pertinência, implementado com um algoritmo que aplica os conceitos de Sistema Imunológico Artificial baseado em Gradiente e Clusterização.

Em geral, no processo de geração da base de regras *fuzzy* é explorado um espaço de busca que possui todas as regras possíveis. Entende-se por regras possíveis, neste contexto, todas as regras geradas pela combinação dos termos linguísticos definidos para as variáveis do problema abordado. Visando restringir o espaço de busca e, conseqüentemente, reduzir o tempo de convergência do algoritmo, foi proposta uma heurística de pré-seleção de regras candidatas para a geração automática da base de regras *fuzzy*, que foi denominada Tabela de Regras Potenciais.

O algoritmo proposto nesta tese foi validado por meio de resultados reais e de simulação, que demonstraram a funcionalidade e a eficácia do aprendizado. O FIS gerado pelo método proposto nesta tese possui como características principais: 1) A estrutura das regras é simples e a demanda computacional para geração das mesmas é baixa; 2) o sistema *fuzzy* é facilmente interpretável pelo ser humano; e 3) a dependência entre as entradas e a saída é claramente identificada.

Embora o algoritmo de aprendizado supervisionado proposto nesta tese tenha sido desenvolvido e testado em aplicações de veículos inteligentes, ele não se restringe a este caso particular. Genericamente, o algoritmo pode ser aplicado para geração automática de FIS a partir de banco de dados numéricos em outras aplicações.

## **6.2 - Propostas de Continuidade do Trabalho**

Com base na experiência adquirida durante o desenvolvimento desta tese, algumas propostas para trabalhos futuros podem ser apresentadas:

- Conforme já mencionado, o algoritmo proposto nesta tese para aprendizado supervisionado de FIS foi concebido para aplicação em veículos inteligentes, no entanto pode ser aplicado em outros casos. Uma sugestão imediata de continuidade deste trabalho é a análise do algoritmo proposto utilizando bancos de dados públicos disponíveis em repositórios e sua comparação com outros algoritmos que utilizam os mesmos bancos de dados;
- Visando aumentar a autonomia do projeto automático do FIS, ou seja, reduzir a intervenção do usuário ou projetista neste processo, propõe-se o ajuste automático do número de funções de pertinência de cada variável, que no algoritmo proposto é um parâmetro definido pelo usuário. Este ajuste pode ser feito por meio de um heurística que leve em consideração o equilíbrio entre os requisitos de interpretabilidade e acuidade.
- A opção pelo aprendizado supervisionado em uma aplicação real de veículos inteligentes se justifica por não ser viável que o aprendizado seja feito a partir da experiência adquirida por meio da interação com o ambiente. Esta estratégia poderia provocar danos ao veículo durante o processo de aprendizagem. No entanto, é possível combinar o algoritmo proposto nesta tese com técnicas de aprendizado por reforço

visando maximizar, em modo *online*, o desempenho do FIS gerado no aprendizado supervisionado.

- No algoritmo proposto nesta tese, as variáveis de entrada do sistema de controle são determinadas com base no conhecimento do especialista. Uma outra abordagem a ser pesquisada refere-se à identificação automática destas variáveis de entrada a partir de um banco de dados contendo toda a informação gerada pelo sistema de percepção.

- Em relação à aplicação, o desempenho de navegação em um circuito com waypoints pode ser melhorado se a velocidade do veículo também for considerada na geração do banco de dados de treinamento.

### 6.3 - Produção Científica

No âmbito desta tese houve o envolvimento no projeto de pesquisa **Técnicas de Automação de Manufatura Industrial com a Integração de Fornecedores e Clientes através de um Sistema de Veículos Não Tripulados** com apoio da FINEP (CT-INFRA - PROINFA 01/2005), o que gerou a publicação de 1 capítulo de livro, 1 artigo completo publicado em periódico e 5 trabalhos completos publicados em anais de congressos, listados a seguir:

#### **Capítulo de Livro:**

VERMAAS, L. L. G. ; HONORIO, L. M. ; JESUS, E. ; FREIRE, M. ; ALCANTARA, D. . Intelligent Vehicle Survey and Applications. In: Germando Lambert-Torres; Jair Minoro Abe; João Inácio da Silva Filho; Helga Gonzaga Martins. (Org.). ADVANCES IN TECHNOLOGICAL APPLICATIONS OF LOGICAL AND INTELLIGENT SYSTEMS. : IOS PRESS, 2008, v. 185, p. 205-235.

#### **Trabalhos Completos Publicados em Anais de Congressos:**

VERMAAS, L. L. G.; HONORIO, L. M. ; FREIRE, M. ; BARBOSA, D. A. . Learning Fuzzy Systems by a Co-Evolutionary Artificial-Immune-Based Algorithm. Lecture Notes in Computer Science (LNCS) **JCR**, v. 5571, p. 312-319, 2009.

VERMAAS, L. L. G.; HONORIO, L. M.; VIDIGAL, M. ; GONÇALVES, L. . Uma metodologia para aprendizado supervisionado aplicada em veículos inteligentes. In: XVIII Congresso Brasileiro de Automática - CBA 2010, 2010, Bonito, MS. CBA 2010, 2010.

HONORIO, L. M. ; VERMAAS, L. L. G. ; JESUS, E. . Survey of Intelligent Vehicles Part I: Architecture and Methodologies. In: VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007, Santos. VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007.

VERMAAS, L. L. G. ; HONORIO, L. M. ; JESUS, E. . Survey of Intelligent Vehicles Part II: Sensors and Communication for Perception. In: VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007, Santos. VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007.

JESUS, E. ; HONORIO, L. M. ; VERMAAS, L. L. G. . Survey of Intelligent Vehicle Part III: Visual Control. In: VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007, Santos. VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007.

HONORIO, L. M. ; VERMAAS, L. L. G. ; FREIRE, M. ; ALCANTARA, D. . Ajuste automático de regras Fuzzy para Estacionamento Paralelo utilizando Sistemas Imunológicos Artificiais. In: VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007, Santos. VI LAPTEC 2007 - Congress of Logic Applied to Technology, 2007.

## Apêndice A

# Veículos Inteligentes

Este apêndice apresenta uma revisão bibliográfica sobre IV, destacando as aplicações de veículos autônomos terrestres. Quatro áreas principais serão discutidas: topologia do sistema, instrumentação para percepção do ambiente, controle de alto nível e visão computacional. A Seção A.2 apresentará a arquitetura mais comum em veículos inteligentes. A instrumentação usualmente empregada, tais como Sistema de Posicionamento Global Diferencial (do inglês *Differential Global Positioning System* - DGPS), sistema de navegação inercial (INS), radares, sensores laser e infravermelhos, assim como as técnicas utilizadas para registro simultâneo e fusão de sensores, serão apresentadas na Seção A.3 deste capítulo. Na Seção A.4 os conceitos básicos e uma visão geral de algumas técnicas e métodos usados para controle visual de veículos autônomos serão discutidos. Na Seção A.5 é apresentado o modelo cinemático adotado

neste projeto e, finalmente, na Seção A.6 é apresentado o Projeto *Driving4u* da Universidade Federal de Itajubá.

## **A.1 – Estado da Arte em Veículos Inteligentes**

Um bom ponto de partida para a identificação do estado da arte em IV é analisar as características dos veículos participantes do *Darpa Grand Challenge* e *Darpa Urban Challenge* referidos na seção anterior. O envolvimento de diversas universidades, principalmente dos EUA, em parceria com empresas do setor automobilístico, faz desta competição uma vitrine para o que existe de mais desenvolvido em IV. Outros projetos igualmente importantes realizados na Europa e Ásia são também analisados nesta seção.

O vencedor do *Urban Challenge 2007*, denominado *Boss*, é uma *Chevy Tahoe* que utiliza um sistema de percepção, *software* de planejamento e controle para tomar decisões sobre como navegar de forma segura em condições de tráfego em ambientes urbanos (BAKER; DOLAN, 2008). A Figura 38 mostra uma foto do veículo equipado com mais de doze sensores do tipo laser, radares e câmeras. Um algoritmo de alto nível para planejamento de rota determina o melhor trajeto em uma malha rodoviária. Além disso, o controle de navegação é capaz de lidar com obstáculos estáticos e dinâmicos; detectar pista, limites da rodovia, vaga de estacionamento, sinais de parada e limites de velocidade. Toda esta tecnologia permite ao *Boss* navegar de forma autônoma, obedecer a regras de trânsito, seguir outros veículos mantendo distância segura, estacionar em uma vaga e reagir a condições dinâmicas, tais como rodovias interditadas, passagem de pedestres ou demais obstáculos.



Figura 38 - Boss, veículo vencedor do *Darpa Urban Challenge 2007*

Já o veículo *Stanley* (THRUN et al., 2006), vencedor do *Grand Challenge 2005*, é um *Touareg R5* da Volkswagen. O veículo, ilustrado na Figura 39, é atuado por meio de um sistema *drive-by-wire*, incorporando medições de DGPS e sistema de navegação inercial. Quando em movimento, o ambiente é percebido por meio de 4 *scanners laser*, sistema de radar, um par de câmeras *stereo* e um sistema de visão monocular. As taxas de amostragem de dados relativos ao ambiente externo estão na faixa de 10 a 100 Hz. Dados do veículo, tais como velocidade e ângulo da direção, são obtidos por meio da interface de comunicação *Controller Area Network – CAN* do próprio veículo. Todo o processamento é feito em 6 computadores *Pentium* alimentados por um sistema auxiliar de baterias. Uma característica importante deste projeto é a possibilidade de ativar e desativar o sistema autônomo, permitindo que o motorista a qualquer momento, mesmo com o veículo em movimento, possa assumir o controle de direção.



Figura 39 - Stanley, veículo vencedor do *Darpa Grand Challenge 2005*

Apesar da grande visibilidade alcançada pelos desafios propostos pelo DARPA nos EUA, diversos outros grupos ao redor do mundo já se propuseram, na última década, a desenvolver um projeto rentável e seguro de direção autônoma. Alguns programas de grande porte na Comunidade Européia, tais como o “*Program for European Traffic with Highest Efficiency and Unprecedented Safety – PROMETEUS*” (FIGUEIREDO et al., 2001). Este programa conta com parcerias tais como, *Daimler-Benz, BMW, Fiat, Renault, Volkswagen*, Ministério dos Transportes da Holanda, entre outros.

Ainda na Europa o programa AUTOPIA (NARANJO et al., 2006), conduzido por pesquisadores espanhóis, do Instituto de Automática Industrial, também estão focadas na direção autônoma cooperativa. Estes pesquisadores equiparam três automóveis comerciais com o objetivo de dirigirem automaticamente em um circuito fechado. A pista, com 1 km de comprimento total, permite reproduzir a maioria das situações de tráfego, tais como: virar, parar e sair, ultrapassagem, dirigir em comboio, estacionar, etc. Além do GPS instalado nos três carros, o primeiro usa também um scanner a laser para detecção de obstáculos, o segundo utiliza um sistema de visão artificial para identificação da pista e o terceiro utiliza um sistema de visão *stereo* usado para detecção de pedestres e obstáculos na pista. Um recurso especial utilizado neste

projeto foi uma rede local sem fio (W-LAN) para transmitir o dado de correção diferencial do GPS base para os GPS receptores instalados nos carros.

Outros projetos de destaque na área de IV também são realizados no Japão. Estes projetos tiveram início na década de 70, podendo-se destacar três importantes contribuições : o projeto denominado “*Intelligent Vehicle*” (TSUGAWA et al., 1991) , o projeto “*Personal Vehicle System – PVS*” (GRAEFE; KUHNERT, 1988) desenvolvido no final da década de 80 e o projeto “*Automated Highway Vehicle System – AHVS*” (SUZUKI et al., 1992) desenvolvido pela Toyota. Além da Toyota outras montadoras de automóveis no Japão vêm conduzindo pesquisas em IV, focadas principalmente em sistemas de visão computacional. Assim como acontece na Europa, o avanço tecnológico nesta área é tão grande que as pesquisas não se restringem apenas ao IV isoladamente, mas também ao ITS (OHTSUKA, 2008) com apoio governamental.

## **A.2 – Arquitetura de Veículos Inteligentes**

O número de camadas na arquitetura de veículos inteligentes pode variar de duas a cinco, dependendo da complexidade da aplicação. A Figura 40 ilustra a arquitetura típica de veículos inteligentes, sendo cada camada descrita posteriormente.

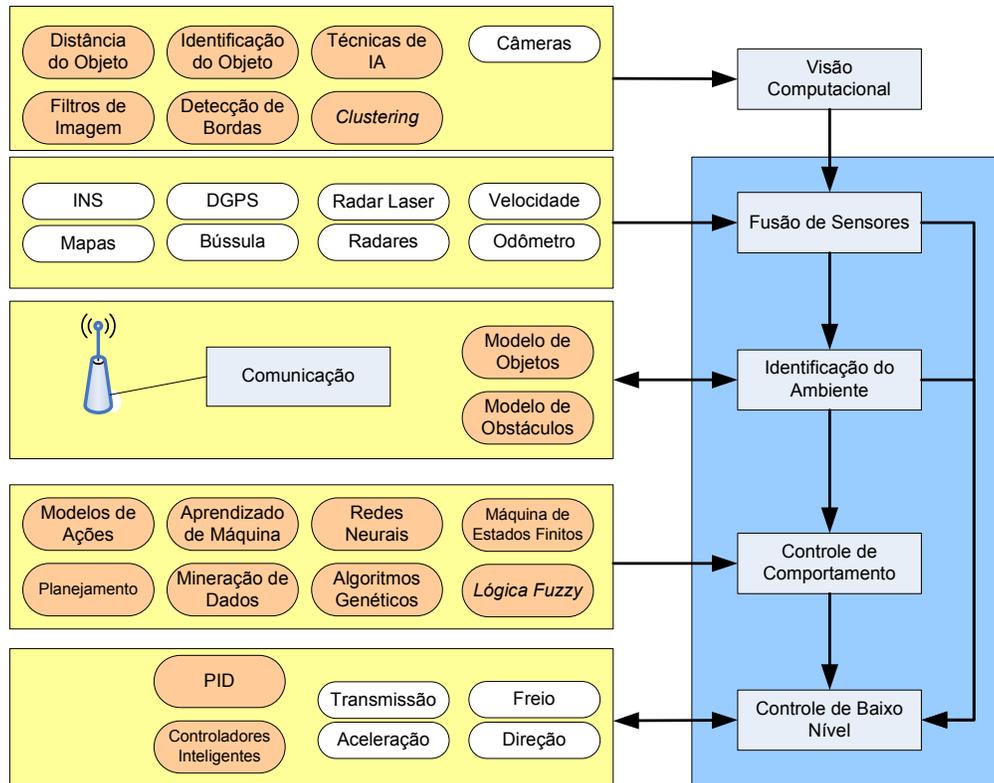


Figura 40 - Arquitetura Típica de Veículos Inteligentes

a) Fusão de Sensores – esta camada caracteriza-se por um conjunto de sensores, tais como câmeras, DGPS, radares, sensores e *scanners* do tipo laser, sensores de velocidade, INS, etc, responsáveis pela percepção do ambiente onde o veículo está inserido. Devido à diversidade de variáveis a serem medidas e à confiabilidade exigida nesta aplicação, um único sensor não é suficiente para permitir a navegação de maneira segura. Portanto, sistemas de multi-sensores empregam estratégias de fusão para aumentar o nível de segurança. O principal desafio nesta camada é transformar dados provenientes de diversos sensores em informação confiável.

b) Identificação do Ambiente – Responsável por obter o entendimento de um cenário por meio da construção de conhecimento de uma rede de sensores. Esta camada deve prover informações sobre obstáculos, possíveis colisões, outros veículos, pedestres, sinais de trânsito, semáforos, pistas, etc. O veículo pode conseguir esta identificação a partir de seu próprio sistema ou trocando informações com outros veículos inteligentes ou a própria rodovia, conseqüentemente expandindo seu

conhecimento sobre o ambiente. Esta camada representa a capacidade humana de reconhecer elementos de um determinado cenário. Embora a cena possa ser corretamente identificada, isto não garante o conhecimento do que esta cena significa. Por exemplo, um IV pode reconhecer a aproximação de outro carro, no entanto pode ser que não consiga identificar uma situação de possível colisão.

c) Controle de Comportamento e Situação – Apenas a descrição obtida pela camada de Identificação do Ambiente ainda não é suficiente para as aplicações de navegação autônoma. É necessário reconhecer a situação que o veículo está envolvido e definir uma ação ou conjunto de ações a ser executado. Para este domínio de aplicação as regras de controle devem representar a capacidade humana de identificar e solucionar problemas. Portanto, esta camada deve prover:

- Modelos de ações e táticas (evitar colisão, manter-se na pista, manobras de estacionamento, ajustes de navegação, e outros),
- Técnicas de AI (lógica *fuzzy*, redes neurais artificiais – ANN (do inglês *Artificial Neural Networks*), algoritmo genético – GA (do inglês *Genetic Algorithm*), planejamento, aprendizado de máquina, mineração de dados, entre outras)
- Máquinas de estado finito responsáveis por definir a mudança de táticas e ações.

No exemplo anterior, após a identificação correta de uma situação de colisão, esta camada estabelece um comportamento para o veículo.

d) Controle de baixo nível – Este nível reúne o modelo cinemático de todos os sistemas do veículo, tais como, direção, transmissão, aceleração, velocidade das rodas e frenagem. Neste nível os controladores podem usar técnicas de IA ou não.

Para aplicações envolvendo baixo nível de inteligência, apenas duas camadas são necessárias. Por exemplo, em freios ABS, os sensores são conectados diretamente no controle de baixo nível que por sua vez atua diretamente nos freios. As outras camadas

desta arquitetura não são necessárias nesta aplicação. À medida que a inteligência do veículo aumenta, o número de camadas tende a aumentar também. Soluções envolvendo veículos completamente autônomos (OZGUNER et al., 2007) apresentam todas as camadas desta arquitetura.

## **A.3 - Instrumentação Veicular**

### **A.3.1 - Introdução**

Uma parte importante de todo tipo de sistema autônomo é a percepção e aquisição de conhecimento sobre o ambiente onde este sistema está inserido. O elemento primário para esta percepção é o sensor. Atualmente, existe uma variedade grande de tipos de sensores disponíveis para aplicação em veículos inteligentes.

As principais fontes de informação do sistema de percepção em veículos inteligentes são baseadas em medições inerciais, laser, DGPS, radar e visão computacional. A Navegação Inercial, por exemplo, baseia-se em medições das acelerações e velocidades angulares do corpo em movimento em relação a determinado sistema de referência. Tais medições são obtidas por instrumentos que empregam as leis newtonianas de movimento, tais como giroscópios e acelerômetros. Por meio de processos de integração, torna-se possível obter velocidade e posição, bem como determinar a orientação do corpo.

De acordo com a literatura o primeiro IV totalmente autônomo foi desenvolvido no Japão em 1977 (TSUGAWA et al., 1991). O projeto pioneiro do grupo coordenado pelo Professor Sadayuki Tsugawa, utilizou um motor acoplado a barra de direção, visão artificial e um sistema de controle proporcional. Apesar da simplicidade do projeto primitivo, ele demonstrou a viabilidade técnica dos veículos autônomos.

Os veículos autônomos capazes de dirigir sem a intervenção humana já são atualmente uma realidade tecnológica. Alguns grupos de pesquisa no mundo já

desenvolveram sistemas de controle automático em veículos reais, como por exemplo, o Projeto *Autopia* (NARANJO et al., 2006), *Partners for Advanced Transit and Highways - PATH* (SHLADOVER, 2006), *DARPA's Navlab* (THORPE et al., 1988), *Martin Marietta's Alvim* (TURK et al., 1988), *Lanelok* (KENUE, 1989), a plataforma avançada para navegação visual de veículos autônomos - VAMP (FRANKE et al., 1998), veículo *ARGO* (BERTOZZI et al., 2000), *Praxitele* (LAUGIER et al., 1999) e Veículo com Sistema Inteligente para Automação de Transporte - VISTA (WANG et al., 2002) entre outros.

Diversas montadoras de automóveis já incorporaram nos carros comerciais uma parte da tecnologia existente, especialmente para assistência ao motorista. Desde 1995, sistemas de controle de cruzeiro adaptativo, utilizando sensores laser ou radar para medir a distância, a velocidade e o ângulo em relação ao carro da frente, são utilizados em carros comerciais. Alguns automóveis comerciais equipados também com sistemas de visão noturna utilizando câmera com infravermelho auxiliam o motorista em condições ruins de visibilidade. Outros empregam câmeras de vídeo para auxiliar o motorista a manter a faixa de direção ou detectar outro automóvel na faixa lateral fora de sua zona de visão. Sistemas que auxiliam o estacionamento em vagas paralelas e que até mesmo estacionam o veículo de forma autônoma já são também uma realidade em alguns automóveis produzidos comercialmente.

### **A.3.2 - Sensores, Aplicações e Características**

Projetar um veículo que seja capaz de navegar sem a intervenção humana em rodovias, em áreas urbanas ou em qualquer ambiente externo é significativamente diferente de projetar um veículo para navegar em ambientes internos, relativamente controlados.

Para perceber o mundo real, o veículo precisa ser equipado com diversos tipos de sensores que podem ser agrupados funcionalmente da seguinte forma:

- a) Sensores para reconhecimento de rota – são em geral sensores de efeito eletromagnético, laser, *transponders* ou visão computacional para identificar marcações na pista.

- b) Sensores para reconhecimento de obstáculos – são comumente scanners a laser, ultra-som, radar, sensor infravermelho ou visão estereoscópica que permitem identificar e evitar colisões com outros veículos, pedestres os demais obstáculos.
- c) Sensores para navegação – são DGPS, acelerômetro, giroscópio, bússula, sensores de velocidade, ou odômetro utilizados para garantir que o veículo trafegue de forma segura e controlada pela rota traçada.

Tabela 9 - Tipos de Sensores, Aplicações e Características

<i>Sensor</i>	<i>Usado para</i>	<i>Custo<sup>1</sup></i>	<i>Características Importantes</i>
DGPS	Navegação	Elevado	Cobertura global e posição absoluta; Posicionamento preciso: ~ 1cm; Taxa de atualização dados baixa: ~ 10 Hz; Sujeito a períodos de indisponibilidade do sinal;
INS	Navegação	Baixo	Posição Relativa; Taxa de atualização dos dados maior que 100Hz; Não depende de sinais externos; Sujeito a erros cumulativos;
Camera	Detecção de Pista e Obstáculos	Médio	Apresenta problemas em condições extremas de clima e estradas <i>off-road</i> ;
Radar	Detecção de Obstáculos	Médio	Permite medir distância e velocidade de veículos próximos; Faixa de medida: ~ 100 m; Taxa de atualização: ~ 10Hz
Laser	Detecção de Pista e Obstáculos	Baixo a Alto	Útil em áreas rurais para detecção de limites de pista;

<sup>1</sup> Foi considerado o custo relativo entre sensores.

Para utilização em aplicação automotiva a instrumentação deverá atender ainda os seguintes requisitos: baixo custo, tamanho reduzido, robustez, eficaz em diversas condições ambientais, detecção em tempo real e capacidade de integração com outros sistemas.

Uma pesquisa sobre os tipos de sensores utilizados em veículos inteligentes mostrou que existem desde projetos simples e de baixo custo até projetos como o *Boss* (BAKER; DOLAN, 2008), veículo autônomo vencedor do *DARPA Urban Challenge - 2007*. Este veículo utilizou 12 sensores laser para medir distância, radares, câmera *stereo* e câmera monocular. Para inferir sua posição, o veículo utilizou ainda receptores DGPS, INS, bússula digital, além de obter a velocidade e a distância percorrida por meio da própria rede CAN (*Controller Area Network*) do automóvel.

Ozguner (2007) propõe um sistema de visão com diferentes sensores capaz de perceber o ambiente 360º em torno do automóvel. Nas laterais e na parte traseira são instalados oito sensores ultrasônicos com alcance de 4m cada. Na parte da frente foi instalado um radar com alcance de até 100m, um par de câmeras *stereo* que permitem visão num raio de 35m e três *scanners* laser de 180º com alcance de até 50m.

Hatipoglo (2003) utiliza um veículo já equipado com *drive by wire* e possibilidade de controle de aceleração e freio, assim como aquisição de diversas medições do estado do veículo por meio de um barramento de comunicação de dados instalado no veículo. O carro foi equipado com uma câmera de vídeo monocromática do tipo CCD de baixo custo instalada na posição do espelho retrovisor central, um radar e um *scanner* laser instalados no pára-choque dianteiro.

No projeto MARVEYE (GREGOR et al., 2002) um sistema de visão multifocal denominado *EMS-Vision* é utilizado. Ao contrário das implementações mais comuns, que trabalham com configurações estáticas e otimizadas para tarefas ou domínios específicos, o *EMS-Vision* é um sistema flexível capaz de se auto-configurar dinamicamente durante a operação, de maneira semelhante aos olhos humanos que se ajustam para visão de perto ou de longe. Enquanto este sistema de visão é utilizado para perceber o ambiente futuro que o veículo vai encontrar, sensores INS provêm informações correntes (atuais) sobre a interação do veículo e o ambiente.

O projeto espanhol AUTOPIA (NARANJO, et al., 2006) equipou três automóveis comerciais com o objetivo de dirigirem automaticamente em um circuito fechado. A pista, com 1 km de comprimento total, permite reproduzir a maioria das situações de tráfego, tais como: virar, parar e sair, ultrapassagem, dirigir em comboio, estacionar, etc. Além do GPS instalado nos três carros, o primeiro usa também um scanner a laser para detecção de obstáculos, o segundo utiliza um sistema de visão artificial para identificação da pista e o terceiro utiliza um sistema de visão *stereo* usado para detecção de pedestres e obstáculos na pista. Um recurso especial utilizado neste projeto foi uma rede local sem fio (W-LAN) para transmitir o dado de correção diferencial do GPS base para os GPS receptores instalados nos carros.

O projeto de um IV cujo objetivo principal é o desenvolvimento de um sistema de alerta de segurança e assistência ao motorista é apresentado em (LI et al., 2004). O veículo, denominado Springrobot, utiliza visão computacional, sensores infravermelho, radar, DGPS, acelerômetro e giroscópio para perceber as condições de tráfego, tomar decisões e planejar a trajetória para trafegar em áreas rurais ou urbanas.

### **A.3.3 - Multi-sensores e Fusão**

A Seção A.3 deste documento abordou a instrumentação comumente utilizada nos projetos de veículos inteligentes. É sabido que um único tipo de sensor não é capaz de fornecer todas as informações necessárias para este tipo de aplicação. Cada sensor possui suas características, vantagens e desvantagens. O DGPS, por exemplo, é capaz de informar a posição com precisão de até 1 cm, mas em alguns lugares o sinal do satélite pode ser bloqueado ou ter sua precisão reduzida. Outro sensor largamente utilizado é o INS, que tem a vantagem de ser pouco sensível a ruídos de alta frequência e intempéries externas. No entanto, o erro medido do INS se acumula se ele não é calibrado *online*. Diversas técnicas de fusão de sensores têm sido pesquisadas e aplicadas com o objetivo de integrar e melhor explorar a complementaridade entre sensores com diferentes características, tais como escala de medida, sensibilidade e confiabilidade. A fusão da informação de diferentes sensores também permite que o

sistema continue funcionando no caso de falha ou perda de um dos sensores: redundância.

Um algoritmo (UFK) baseado em filtro de Kalman é proposto em (LI; LEUNG, 2004), para registrar e fazer a fusão simultaneamente de um GPS, INS, radar e uma câmera para um sistema de direção colaborativa. A Figura 41 ilustra um comboio de veículos em uma rodovia de duas pistas e o diagrama em blocos da instrumentação e do sistema de fusão dos sensores. Primeiramente, para registrar o desvio do sistema e as medições dos diferentes sensores feitas em tempos distintos, foi desenvolvido um modelo de registro temporal-espacial. Após este registro a fusão das informações dos sensores é feita por meio do UFK.

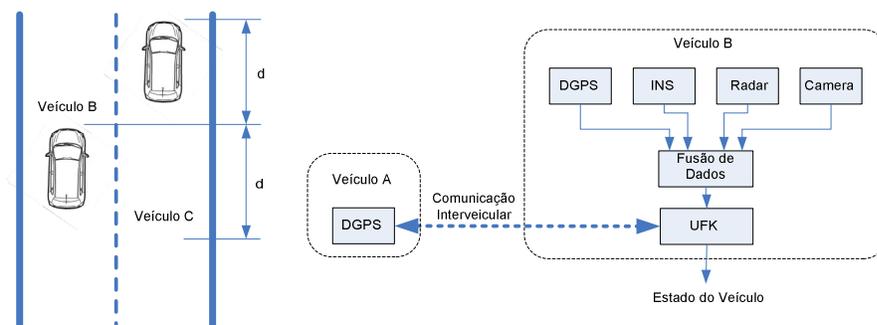


Figura 41 - Comboio e diagrama em blocos da fusão de sensores

A integração GPS/INS tem sido largamente pesquisada (ROGERS, 1996; EL-SHEIMY et al., 2006; STAMPFLE; BECKER, 2005) nas últimas décadas. Esta implementação é essencialmente baseada em técnicas que usam filtro de Kalman ou alguma de suas variantes, como por exemplo, o filtro de Kalman estendido. No entanto, se algum parâmetro desconhecido aparecer no modelo do sistema ou, se o modelo muda com o ambiente, como no caso de Sistemas de Transporte Inteligente (ITS), a precisão estimada diminui. A Figura 42 apresenta um esquema de integração de sensores utilizando Filtro de Kalman.

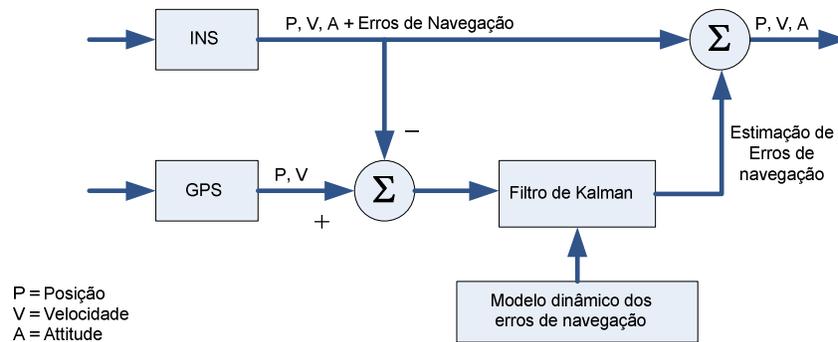
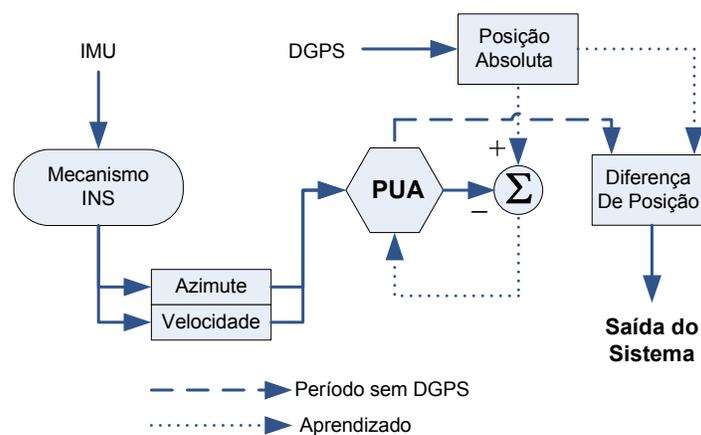


Figura 42 - Diagrama de um Filtro de Kalman para integração INS/GPS

Uma abordagem diferente para o mesmo problema é apresentada em (EL-SHEIMY et al., 2006), onde o método de integração GPS/INS baseia-se em redes neurais artificiais. O método proposto sugere duas arquiteturas diferentes: arquitetura de atualização de posição (PUA) e arquitetura de atualização de posição e velocidade (PVUA). Diversos testes práticos demonstraram a melhor performance destes métodos quando comparadas com as técnicas convencionais de integração GPS/INS. A Figura 43 apresenta a configuração utilizando as redes neurais PUA e PVUA.



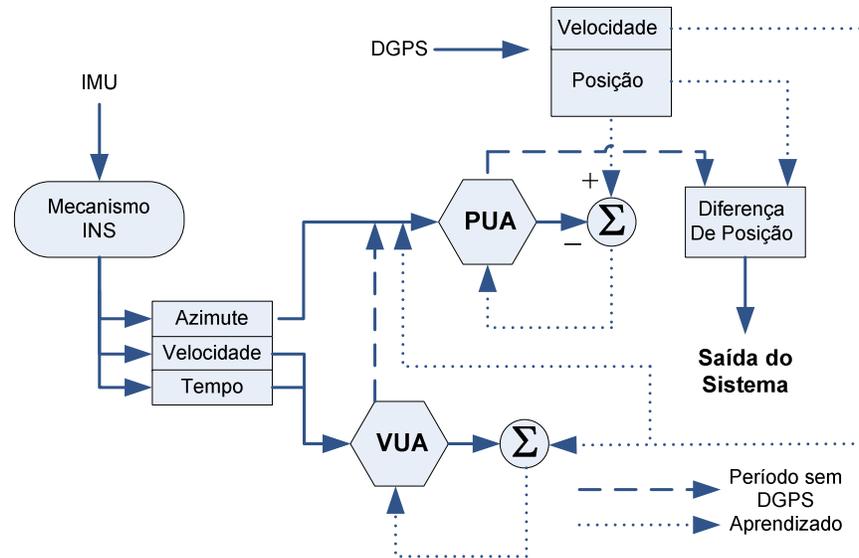


Figura 43 - Configuração de um Sistema PUA e PVUA

### A.3.4 - Comunicação Interveicular e Sistemas de Transporte Inteligente

Para permitir que diversos veículos autônomos trafeguem simultaneamente, existem duas maneiras de garantir certo grau de segurança. A primeira é utilizar sensores, como câmera de vídeo, radar, etc, para perceber os outros carros. Este tipo de solução imita o comportamento do ser humano ao dirigir. A segunda maneira é utilizar comunicação sem fio para obter informações de navegação de cada carro e da própria rodovia e distribuir estes dados entre todos os carros.

Portanto, o principal objetivo ao se utilizar um sistema de comunicação é transmitir informações, tais como: posição, velocidade, direção e decisões de navegação do veículo entre carros que circulam em certa área de cobertura. Adicionalmente esta rede de comunicação permite o controle e a supervisão remotos e também o envio de informações úteis para um host externo se for necessário.

Este sistema de comunicação deverá ainda atender aos seguintes requisitos: meio de transmissão sem fio, operar em tempo real, conectar transmissores e receptores em movimento, possuir baixas taxas de erro e ter capacidade para um grande número de usuários.

Várias tecnologias de comunicação sem fio são consideradas para aplicação em ITS. As soluções que permitem comunicação em duas vias envolvem infravermelho, rádio frequência, serviço de comunicação via satélite, telefonia celular e outras redes de comunicação de dados. Uma comparação detalhada entre as diversas tecnologias sem fio para aplicação em ITS é apresentada em (KAMALI, 1996; KIRSON, 1991; TSUGAWA, 2002).

Shladover et al. (1991), pesquisadores do projeto PATH, analisam as possíveis formas de comunicação interveicular em comboio e propõem a utilização de um link infravermelho para esta aplicação. Utilizando um transmissor ótico na parte traseira e um receptor na frente de cada veículo, cada receptor é capaz de receber informações do veículo a sua frente. Em (MISENER; SHLADOVER, 2006), o mesmo grupo de pesquisadores implementa uma solução baseada em comunicação sem fio. A tecnologia sem fio utilizada é o padrão IEEE 802.11b, escolhido por ser uma solução pronta disponível comercialmente. Com este sistema de comunicação o veículo comunica periodicamente em intervalos de 50 ms, à distância de até 1,5 km trafegando a uma velocidade de até 90 km/h.

Não existe ainda um padrão definido para a arquitetura de comunicação em ITS. Embora diversas propostas venham sendo apresentadas definir este padrão não é uma tarefa tão simples. Assim como observado em outras aplicações, como por exemplo, em redes de comunicação industrial, diversos sistemas proprietários são desenvolvidos antes da definição de um padrão de comunicação. Como resultado, fabricantes diferentes desenvolvem sistemas de comunicação que não possuem interoperabilidade.

Nos Estados Unidos, o *Federal Communication Commission* (FCC) alocou em 1999 a faixa de frequência de 5.850 - 5.925 GHz para aplicações em ITS. Para garantir

interoperabilidade, as comunicações a curta distância seguirão os padrões de camada física e de controle de acesso ao meio do ANSI/IEEE 802.11a, com pequenas modificações (YANG; WANG, 2007). Este padrão será utilizado para funções de telecomunicação e troca de informação entre os sistemas do veículo e da rodovia.

#### **A.4 - Sistemas de Visão Computacional**

O principal objetivo da utilização de visão computacional em veículos inteligentes é garantir a eficiente navegação autônoma. Em aplicações como esta, os sistemas de visão computacional devem ser capazes de reconhecer e operar em condições variadas de tempo, luminosidade, chuva, nebulosidade ou sol. Outras características importantes destes sistemas são velocidade e consistência da informação.

### A.4.1 – Conceitos Básicos

Um controle baseado em visão computacional começa com uma imagem digitalizada obtida por meio de uma única câmera (sistema monocular), ou por meio de duas ou mais câmeras (sistema *stereo*) instalados no veículo a ser controlado.

A próxima etapa é encontrar na imagem obtida do ambiente os limites para localização da rota. Isto pode ser feito utilizando um dos seguintes métodos: detecção de extremidades (*edge detection*) ou análise de intensidade de *pixels* (ROMAN et al.; 1998). Em ambos os casos, é necessário: encontrar e localizar vias; eliminar ou ignorar ruídos e reflexos (principalmente aquelas causadas pela luz do sol); reduzir detecções incorretas e rejeitar obstáculos que possam ser confundidos com os limites da pista.

Para identificar a pista em uma imagem, é necessário aplicar métodos de reconhecimento, também conhecidos como segmentação, que podem utilizar o conceito de *threshold*. Métodos *threshold* podem operar de duas formas distintas: modo global ou local. No primeiro caso a imagem é dividida utilizando apenas um valor de *threshold*. No segundo caso, sub-imagens possuem seus próprios valores de *threshold*. O método de *threshold* global é recomendado para imagens com distribuição uniforme de luz, sendo a forma mais simples computacionalmente de realizar a segmentação da imagem. No entanto, o método de *threshold* multi-nível deve ser usado quando o processamento é feito em diversas regiões distintas da imagem, sendo portanto mais complexo. Neste procedimento é necessário selecionar um valor para o *threshold*. Em um ambiente controlado este valor pode ser obtido estatisticamente por meio de diversas medições feitas no ambiente. No entanto, em ambientes externos com iluminação natural e sujeitos a variações, tais como neblina ou chuva, o valor *threshold* é mais difícil de ser obtido. Neste caso, métodos dinâmicos são empregados para seleção da intensidade de *threshold*.

Algumas etapas devem ser seguidas para se conseguir reconhecer características do ambiente por meio da análise por visão computacional. Estas etapas são: pré-processamento da imagem, segmentação e identificação do ambiente.

A maioria dos procedimentos de pré-processamento lida com imagens em tons de cinza, que são mais fáceis de trabalhar, consomem menos memória e tempo computacional, além de já existirem diversos métodos disponíveis a serem utilizados. No caso de extração de características do ambiente em aplicações de IV, existem casos onde é importante detectar diferentes cores, como, por exemplo, para identificar sinais de trânsito, semáforos ou marcações na pista. Nestes casos, devem-se aplicar técnicas de processamento de imagem colorida.

Ainda nesta fase, é necessário também eliminar ruídos geralmente provenientes de iluminação ruim ou do próprio processo de aquisição de imagem. Dependendo das condições do ambiente, estes ruídos podem ser muito significativos e devem, portanto receber uma atenção especial na fase de pré-processamento da imagem.

Na próxima etapa, segmentação da imagem, a imagem do ambiente é explorada por meio de métodos computacionais com o objetivo de identificar partes de interesse, tais como pista, limites da rodovia, outros carros, sinais de trânsito, etc.. O principal objetivo é dividir a imagem em área de navegação e obstáculos.

O sucesso da segmentação de imagem depende fortemente de uma clara diferenciação entre os elementos que se deseja identificar. A identificação da pista pode ser feita por meio de métodos simples caso as marcações da mesma estejam bem definidas e distintas em relação ao restante da imagem, o que nem sempre ocorre.

Finalmente, na etapa de identificação do ambiente, todos os elementos da imagem são reconhecidos, como o próprio veículo e demais obstáculos. Este é a etapa mais difícil, por não haver um método único que solucione todos os tipos de problemas.

#### A.4.2 – Seleção de Casos de Visão Computacional em IV

O veículo “Junior” (DARPA, 2008), da Universidade de Stanford, é muito bem equipado no que compete a sensores, radares e unidades de processamento. As informações obtidas pelos sensores do robô são transmitidas por meio de suas interfaces de planejamento e controle, de forma paralela em todos os processadores envolvidos, o que despense um tempo máximo de 300 ms entre a captura de dados e o seu tratamento. Existem seis câmeras de vídeo instaladas no automóvel. Elas são calibradas de modo a projetar as imagens sobre um modelo esférico, a fim de prover, aos sistemas de percepção do veículo, uma imagem panorâmica completa do ambiente (inclusive da superfície da estrada).

As principais funções desse sistema panorâmico de câmeras são a detecção de marcas da pista e a localização de precisão. Para tal, foi desenvolvido um completo *pipeline* de processamento (técnica de *hardware* que permite que a CPU faça desvios e procure por outras instruções além da próxima a ser executada, de modo a acelerar o seu processamento), operando em tempo real por meio de uma GPU (Unidade de Processamento Gráfico) *onboard*.

O veículo vencedor do DARPA Urban Challenge 2007 possui uma abordagem na qual navegar por uma faixa de trânsito em um ambiente urbano depende das informações passadas pelo subsistema de percepção, que identifica os limites da pista em tempo real. Então, é traçada uma curva que representa o centro da pista, devendo ser, aquela, o caminho a ser seguido pelo veículo. Para tal, foi desenvolvido um sistema de planejamento local da movimentação, que gera um conjunto de pequenas trajetórias. Estas se dispõem a uma pequena distância à frente do veículo e a pequenas distâncias laterais em relação ao centro da pista.

Posteriormente, é usado um algoritmo de geração de trajetórias, baseado em simulações de movimento frontais e em um fiel modelo computadorizado de um automóvel, de forma a predizer o comportamento do veículo após a execução de

determinados comandos de movimento. Em seguida, uma série de etapas de otimização segue curso, de modo a diminuir o erro entre a posição simulada no modelo e a posição real do veículo, por meio de mudanças nos comandos de movimentação.

Os resultados obtidos dessa etapa de otimização são comparados com as informações acerca dos obstáculos existentes na pista, de forma a fazer o veículo selecionar a melhor trajetória a ser seguida. Como a movimentação já foi simulada em um modelo altamente fiel a um veículo real, os controladores do automóvel podem imediatamente executar o percurso da rota selecionada.

## **A.5 – Modelo Cinemático do Veículo**

O modelo cinemático do veículo oferece uma descrição matemática do movimento sem considerar as forças que afetam o veículo. As equações de movimento são baseadas puramente em relações geométricas que regem o sistema (RAJAMANI, 2006).

### **A.5.1 - Representação da Posição do Veículo**

Nesta tese, o ambiente em que o veículo está inserido é considerado bidimensional, onde existe apenas movimentação no plano horizontal representado pelos eixos  $X$  e  $Y$ , como mostra a Figura 44. Para descrever o movimento do veículo é necessário um sistema de coordenadas globais onde  $(X, Y)$  representa a localização do centro de gravidade do veículo no ambiente, enquanto  $\psi$  descreve a orientação do veículo.

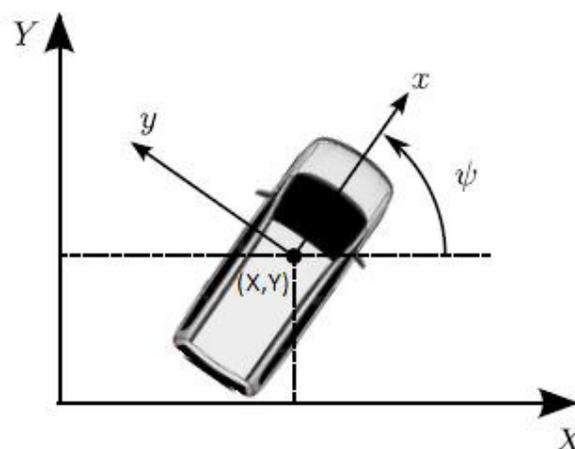


Figura 44 - Representação da Posição do Veículo

Para a orientação do veículo em relação a esse plano é necessário um sistema de coordenadas locais onde o eixo  $x$  representa a movimentação longitudinal do veículo, o eixo  $y$  representa sua movimentação lateral e a origem do sistema de coordenadas é o centro de gravidade do veículo.

### A.5.2 - Modelo Cinemático Aplicado

Considere um modelo biciclo do veículo, como mostrado na Figura 45 . No modelo bicíclico, as duas rodas dianteiras, esquerda e direita, são representadas por uma única roda no ponto  $A$ . Do mesmo modo as rodas traseiras são representadas por uma roda traseira central no ponto  $B$ . O ângulo da direção para a roda dianteira é representada por  $\delta$ . O centro de gravidade do veículo está no ponto  $C$ . A distância dos pontos  $A$  e  $B$  ao centro de gravidade do veículo são  $l_f$  e  $l_r$ , respectivamente e a distância entre eixos do veículo é  $L = l_f + l_r$ .

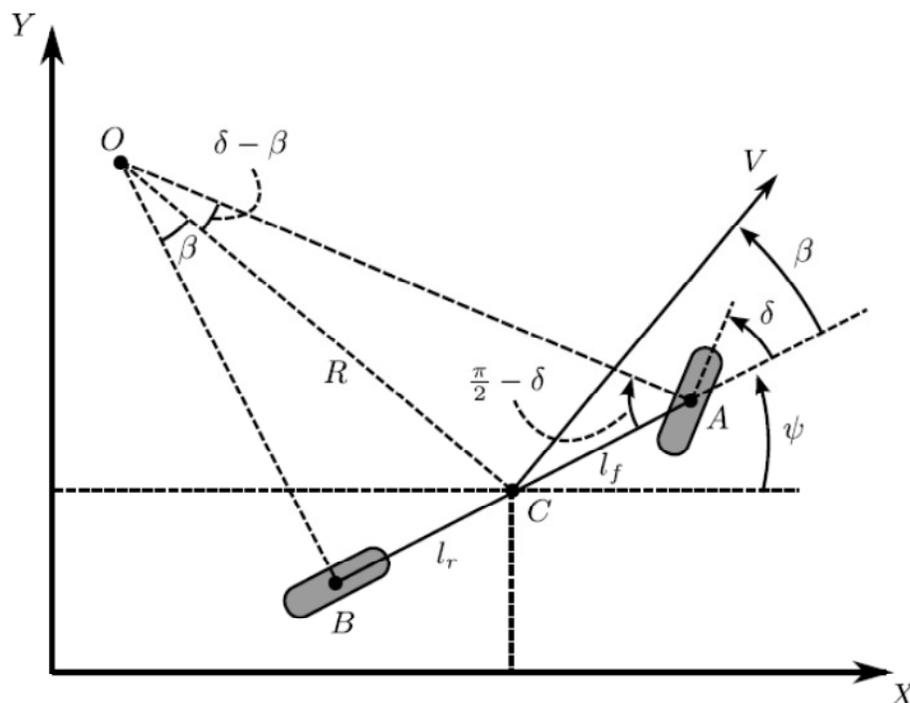


Figura 45 - - Modelo Cinemático do Veículo

Como dito anteriormente,  $(X, Y)$  representa a localização do centro de gravidade do veículo, enquanto  $\psi$  descreve a orientação do veículo. A velocidade no centro de gravidade do veículo é denotada por  $V$  e faz um ângulo de  $\beta$  com o eixo longitudinal do veículo. O ângulo  $\beta$  é chamado de ângulo de deslizamento do veículo.

A principal consideração feita no desenvolvimento do modelo cinemático é que os vetores velocidade nos pontos  $A$  e  $B$  estão na direção da orientação das rodas dianteiras e traseiras, respectivamente. Em outras palavras, o vetor velocidade na roda dianteira faz um ângulo  $\delta$  com o eixo longitudinal do veículo. Isso é equivalente a assumir que os ângulos de escorregamento em ambas as rodas são zero. O ângulo de escorregamento numa roda é o ângulo entre a direção em que a roda aponta e a direção real em que a roda se desloca. Esta é uma suposição razoável para o movimento de baixa velocidade do veículo, por exemplo, para velocidades inferiores a 5 m/s. À baixas velocidades, a força lateral gerada sobre o pneu é pequena (RAJAMANI, 2006).

Para um pequeno período de tempo a velocidade pode ser considerada constante e o ângulo das rodas dianteiras do veículo se mantém fixos. Nessas condições o veículo realiza um movimento circular com um raio de curvatura constante, chamado de curva de estado estacionário (ABE, 2009). O ponto  $O$  é o centro instantâneo de curvatura do veículo e é definido pela interseção das linhas  $AO$  e  $BO$ , que são perpendiculares à orientação das duas rodas.

O raio da trajetória do veículo  $R$  é definido pelo comprimento da linha  $OC$  que liga o centro de gravidade  $C$  ao centro instantâneo de curvatura  $O$ . A velocidade no centro de gravidade é perpendicular a linha  $OC$ .

Aplicando a regra dos senos no triângulo  $OCA$ , temos:

$$\frac{\text{sen}(\delta - \beta)}{l_f} = \frac{\text{sen}(\frac{\pi}{2} - \delta)}{R} \quad (\text{A.1})$$

Da Equação A.1, temos:

$$\frac{\text{sen}(\delta)\cos(\beta) - \text{sen}(\beta)\cos(\delta)}{l_f} = \frac{\cos(\delta)}{R} \quad (\text{A.2})$$

Multiplicando-se os dois lados da Equação A.2 por:  $\frac{l_f}{\cos(\delta)}$

$$\tan(\delta)\cos(\beta) - \text{sen}(\beta) = \frac{l_f}{R} \quad (\text{A.3})$$

Aplicando-se a regra dos senos no triângulo  $OCB$ , temos:

$$\frac{\text{sen}(\beta)}{l_r} = \frac{\text{sen}(\frac{\pi}{2})}{R} \quad (\text{A.4})$$

Logo,

$$\text{sen}(\beta) = \frac{l_r}{R} \quad (\text{A.5})$$

Aplicando-se a Equações A.5 em A.3, temos:

$$\tan(\delta)\cos(\beta) = \frac{l_f + l_r}{R} \quad (\text{A.6})$$

Assumindo-se que o raio da trajetória do veículo muda lentamente, devido à baixa velocidade do veículo, então a taxa de mudança da orientação do veículo deve ser igual à sua velocidade angular (RAJAMANI, 2006).

Portanto,

$$\dot{\psi} = \frac{V}{R} \quad (\text{A.7})$$

Aplicando a Equação A.7 e Equação A.6 pode ser reescrita como:

$$\dot{\psi} = \frac{V}{l_f + l_r} \cos(\beta) \tan(\delta) \quad (\text{A.8})$$

As equações gerais do movimento são, portanto, dadas por:

$$\dot{X} = V \cos(\psi + \beta) \quad (\text{A.9})$$

$$\dot{Y} = V \text{sen}(\psi + \beta) \quad (\text{A.10})$$

$$\dot{\psi} = \frac{V}{l_f + l_r} \cos(\beta) \tan(\delta) \quad (\text{A.11})$$

O ângulo  $\beta$  pode ser obtido das Equações A.3 e A.5:

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r} \tan(\delta)\right) \quad (\text{A.12})$$

As equações obtidas anteriormente representam o modelo cinemático do veículo e são utilizadas para descrever o movimento no ambiente de simulação e na estimativa feita no algoritmo de controle.

## A.6 – O Projeto Driving4u

A proposta apresentada neste Projeto de Tese de Doutorado é parte de um projeto de pesquisa mais abrangente do Grupo de Automação e Tecnologia da Informação da UNIFEI, que recebeu a denominação Driving4u. O projeto Driving4u tem por objetivo a estruturação de uma plataforma de trabalho para pesquisa e desenvolvimento na área de IV. Esta plataforma consiste de uma pista para testes e um veículo equipado com sistemas de percepção e de atuação eletrônico, assim como microcomputadores para aquisição, tratamento de dados e processamento dos algoritmos de controle utilizando técnicas de IA.

A pista de testes é um circuito fechado que permite a realização das principais manobras em situações de trânsito, tais como: parar e sair em cruzamentos, ultrapassagem de veículos, manter distância segura em relação a outros veículos, estacionar, realizar curvas com ângulos diferentes, entre outras.

A Figura 46 apresenta uma foto do veículo do projeto *Driving4u*, uma Zafira – GM automática, utilizada nos testes reais.



Figura 46 - Veículo Inteligente – Projeto *Driving4u*

O projeto conceitual do veículo foi concebido prevendo que todos os sistemas instalados não interfiram nas características originais do veículo, ou seja, sistemas podem ser acoplados ou desacoplados permitindo que o veículo seja dirigido normalmente ou controlado pelo sistema inteligente. Portanto o veículo pode ser operado de duas formas: manual, onde todos os dados obtidos com a percepção do ambiente externo e das ações do motorista são armazenados em um banco de dados; ou automático, onde dada uma tarefa de navegação o veículo recorre ao conhecimento previamente construído para executá-la.

O sistema de percepção proposto consiste de DGPS, acelerômetro, giroscópio, sensores ultrasônicos e laser, assim como medição de velocidade do veículo por meio dos sensores já instalados nas rodas. Em uma etapa posterior, o projeto prevê a utilização de técnicas de visão computacional para navegação e detecção de obstáculos. A Figura 47 mostra os principais componentes instalados/previstos no projeto.

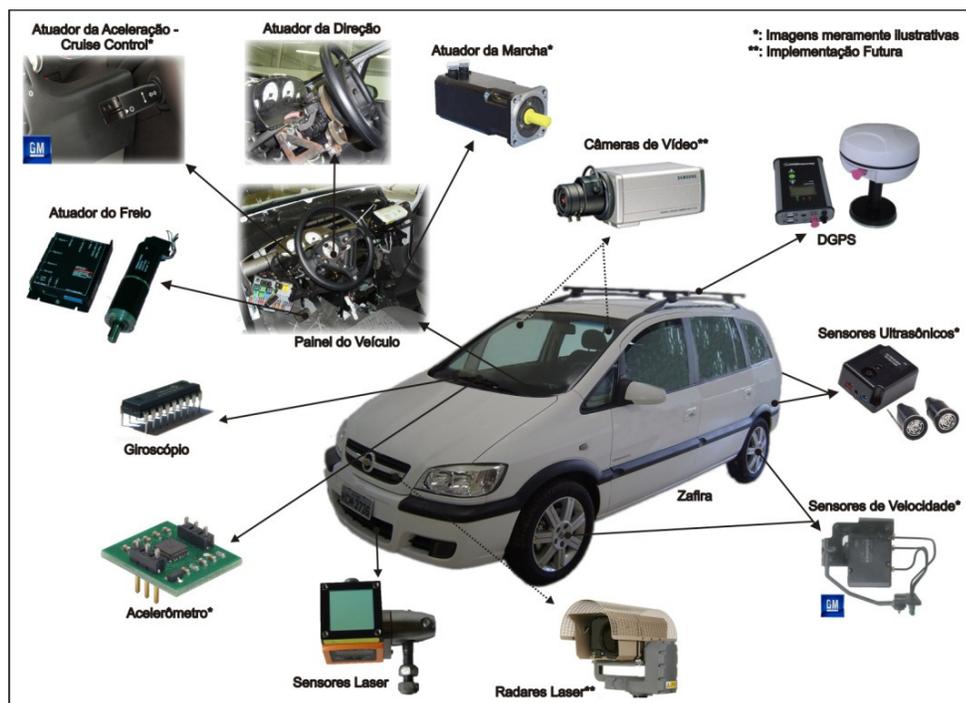


Figura 47 – Diagrama de sensores e atuadores no veículo.

A atuação na direção do veículo é realizada por meio da instalação de um servomotor na barra de direção permitindo o acionamento eletrônico dos mesmos.

Este servomotor pode ser facilmente acoplado ou desacoplado da barra de direção. A mudança na alavanca de transmissão automática para as posições “Estacionamento”, “Dirigir” e “Ré” é implementada por meio de atuador linear elétrico instalado no mecanismo da alavanca. O freio também é acionado por um atuador linear elétrico acoplado ao pedal de forma a permitir o acionamento pelo motorista ou de forma autônoma. Finalmente, a aceleração é realizada eletronicamente por meio do sistema *Cruise Control* do próprio veículo que atua diretamente no módulo de controle do motor.

A diversidade destes elementos junto com o tempo de resposta de cada dispositivo cria uma dificuldade para o sincronismo das informações. Para melhorar a compatibilidade os sensores foram divididos em *clusters* de acordo com a finalidade, importância e a necessidade de resposta. A Figura 48 mostra o diagrama de dados do veículo onde os itens em pontilhado definem o que ainda será implementado no projeto.

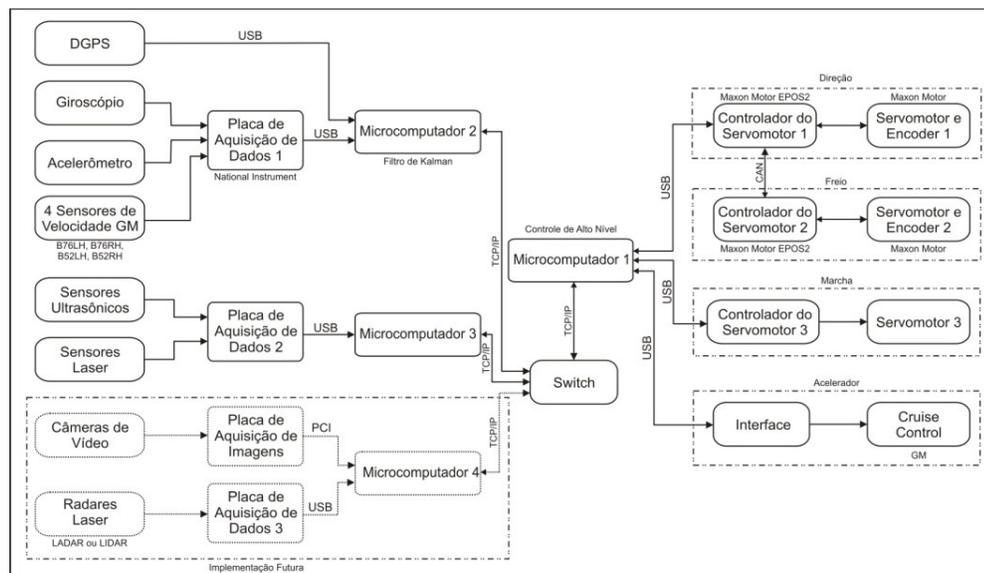


Figura 48 – Diagrama de Dados do Veículo.

# Referências

- Abe, M. **Vehicle Handling Dynamics: Theory and Application**. Elsevier, 2009.
- Abe, S., Lan, M. **Fuzzy Rules Extraction Directly from Numerical Data for Function Approximation** *IEEE Transactions on Systems, Man, and Cybernetics* 01, vol. 25, pp. 119-129, 1995.
- Alves, R. T.; Delgado, R.M.; Lopes, H.S.; Freitas, A.A. **An Artificial Immune System for Fuzzy-Rule Induction in Data Mining**. In: Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervós, J.J. (eds.) *Parallel Problem Solving from Nature - PPSN VIII*. LNCS, vol. 3242, pp. 1011-1020 Springer, Heidelberg, 2004.
- Baker, C.R.; Dolan, J.M. **Traffic interaction in the urban challenge: Putting boss on its best behavior**. *IEEE International Conference on Intelligent Robots and Systems*, pp.22-26, Sept. 2008.
- Bertozzi, M.; Broggi, A.; Fascioli, A.; Nichele, S. **Stereo vision-based vehicle detection**. *IEEE Proc. Intell. Vehicle Symp.*, pp. 39-44, 2000.
- Bevly D.; Ryu, J.;Gerdes, J. **Integrating INS Sensors with GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering Stiffness**. *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 483 – 493, Dec. 2006.
- Bishop, R. **Intelligent Vehicle Applications Worldwide**. *IEEE Intelligent Systems and Their Applications*, vol. 15, no 1, pp. 78-81, Jan-Feb 2000.
- Bishop, R. **Intelligent Vehicle Technology and trends**. Norwood: Artech House Publishers, 2005.
- Carmona, P.; Castro, J.L.; Zurita, J.M. **Strategies to Identify Fuzzy Rules Directly From Certainty Degrees: A Comparison and a Proposal** *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, October 2004.
- Casillas, J.; Cerdón, O.; Herrera, F.; Magdalena, L., **Interpretability Issues in Fuzzy Modeling**. Berlin: Springer-Verlag, 2003.
- Castro, L.; Zuben, F.V. **Artificial Immune Systems: Part I - Basic Theory and Applications**. Relatório técnico, Universidade Estadual de Campinas, Dezembro de, 1999.
- Castro, L.; Zuben, F.V. **Learning and optimization using the clonal selection principle**. *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 3, pp. 239. 251, 2002.
- Cerdón, O.; Herrera, F. **A two-stage evolutionary process for designing TSK fuzzy rule-based systems**. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 29, N. 6, pp 703–714, 1999.
- Cerdón, O.; Herrera, F.; Hoffmann, F; Magdalena, L. **Genetic Fuzzy Systems – Evolutionary Tuning and Learning of Fuzzy Knowledge Bases**. World Scientific, 2001.
- DARPA Grande Challenge 2005, <<http://www.darpa.mil/grandchallenge05>> Acesso em: 20 Mai 2007.

- DARPA Urban Challenge 2007, <<http://www.darpa.mil/grandchallenge/index.asp>> Acesso em 08 Nov 2008.
- Danyadi, Z.; Balazs, K.; Koczy, T. **A comparative study of various evolutionary algorithms used for fuzzy rule-based inference and learning systems.** 2010 International Joint Conference on Computational Cybernetics and Technical Informatics, pp. 49 – 54, 27-29 May 2010.
- El-Sheimy, N.; Chiang, K.; Noureldin, A. **The Utilization of Artificial Neural Networks for Multisensor System Integration in Navigation and Positioning Instruments.** *IEEE Trans. Intell. Transp. Syst.*, vol. 55, no. 5, pp. 1606 – 1615, Oct. 2006.
- Figueiredo, L.; Jesus, I.; Machado, J.A. **Towards the development of intelligent transportation systems.** *IEEE Proc. of Intelligent Transportation Systems*, pp. 1206-1211, Ag. 2001.
- Franke, U.; Gavrila, D.; Gorzig, S.; Lindner, F., **Autonomous Driving Goes Downtown.** *IEEE Intell. Syst. Applic.*, vol. 13, no. 6, pp. 40 – 48, Nov. – Dec. 1998.
- Freitas, A.A. **Data Mining and Knowledge Discovery with Evolutionary Algorithms,** Springer, 2002.
- Fujimori, A.; Nikiforuk, P.N; Madan M. Gupta, M.M. **Adaptative Navigation of Mobile Robots with Obstacle Avoidance.** *IEEE Transactions on Robotics and Automation* Vol.13, 596-602, 1997.
- Graefe, V.; Kuhnert, K. **A high speed image processing system utilized in autonomous vehicle guidance.** in *Proc. IAPR Workshop Comput. Vision*, Tokyo, Japan, pp. 10-13, 1988.
- Gregor, R.; Lutzeler, M.; Pellkofer, M.; Siedersberger, K. **EMS-Vision: A Perceptual System for Autonomous Vehicles.** *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 48 – 59, Mar. 2002.
- Gullapalli, V. **A comparison of supervised and reinforcement learning methods on a reinforcement learning task.** *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pp 394-399, Aug., 1991.
- Hatipoglu, C.; Ozguner, U.; Redmill, K.A. **Automated Lane Change Controller Design.** *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 1, pp. 13 – 22, March 2003.
- Honorio, L M.: **Virtual Manufacture Software.** Information Technology Institute/UNIFEI. Itajubá, Brazil (2005), [Online]. Disponível em: <[www.virtualmanufacturing.unifei.edu.br](http://www.virtualmanufacturing.unifei.edu.br)>
- Honorio, L.M.; Silva, A.L.; Barbosa, D. **A Gradient-Based Immune System Applied to Optimal Power Flow Problems,** *Lecture Notes in Computer Science*, vol. 4628, pp. 1.12, 2007.
- Huang, S. **Safety, Comfort and Optimal Tracking Control in AHS Applications.** *IEEE Control Systems*, 50-64, 1998.
- Ishibuchi, H.; Nozaki, K.; Tanaka, H. **Distributed representation of fuzzy rules and its application to pattern classification.** *Fuzzy Sets Syst.*, vol. 52, no. 1, pp. 21–32, 1992.
- Jang, J.-S. **ANFIS: Adaptive-Network-based Fuzzy Inference Systems.** *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, N. 3, pp 665–685, 1993.

- Jin, Y. **Fuzzy Modeling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement.** *IEEE Transactions on Fuzzy Systems*, Vol. 8, N. 2, pp 212–221, 2000.
- Jonic, S.; Jankovic, T.; Gajic, V.; Popvic, D. **Three machine learning techniques for automatic determination of rules to control locomotion.** *IEEE Transactions on Biomedical Engineering*, vol 46, issue3, pp. 300-310, Março 1999.
- Kamali, B. **Some applications of wireless communications in intelligent vehicle highway systems.** *IEEE Aerosp. Electron. Syst. Mag.*, vol. 11, no. 11, pp. 8 – 12, Nov. 1996.
- Kato, S.; Tsugawa, S.; Tokuda, K.; Matsui, T.; Fuji, H. **Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications.** *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 155 – 161, Sept. 2002.
- Kirson, A.M. **RF Data Communications Considerations in Advanced Driver Information Systems,** *IEEE Trans. Vehic. Tech.*, vol. 40, no. 1, pp. 51 – 55, Feb. 1991.
- Krogh, B.H.; Feng, D. **Dynamic generation of subgoals for autonomous mobile robots using local feedback information.** *IEEE Trans. Automat. Contr.*, vol. 34, pp. 483–493, May 1989.
- Kutami A.; Maruya, Y.; Takahashi, H.; Okuno, A. **Visual Navigation of Autonomous On-Road Vehicle.** *IEEE International Workshop on Intelligent Robots and Systems, IROS '90*, 175-180, 1990.
- Laugier, C.; Paromtchik, I.; Parent, M. **Developing autonomous maneuvering capabilities for future cars.** in *IEEE Proc. Intell. Transp. Syst.*, pp 68 – 73, 1999.
- Lee, T.; Tsai, C. **Fast Parking Control of Mobile Robots: A Motion Planning Approach With Experimental Validation,** *IEEE Transactions on Control Systems Technology* , vol. 12, 661-676, 2007.
- Li, Q.; Zheng, N.; Cheng, H. **Springrobot: A Prototype Autonomous Vehicle and Its Algorithms for Lane Detection.** *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 300 – 308, Dec. 2004.
- Li, W.; Leung, H. **Simultaneous registration and fusion of Multiple Dissimilar Sensors for Cooperative Driving.** *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 84 – 98, Jun. 2004.
- Li, Y.; Ha, M.; Wang, X. **Principle and Design of Fuzzy Controller Based on Fuzzy Learning from Examples.** In: *Proc. of the 1st Int. Con. on Machine Learning and Cybernetics 3*, 1441-1446 (2002)
- Mamdani, E.A. **Application of Fuzzy Algorithms for Control Simple Dynamic Plant.** *Proc. IEE*, 121, 12, pág. 1585-1588, 1974.
- Matlab. **Fuzzy Logic Toolbox.** <<http://www.mathworks.com/help/toolbox/fuzzy/>> Acesso em 05 de Março de 2010.
- McCall, J.C.; Trivedi, M.M. **Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation.** *IEEE Trans. on Intell. Transp. Systems*, vol. 7, 20-37, 2006.
- Michalewicz, Z.; Michalewicz, M. **Evolutionary computation techniques and their applications.** *IEEE International Conference on Intelligent Processing Systems*, pp 14-25, vol. 1, Oct. 1997.

- Misener, J.; Shladover, S. **PATH investigation in Vehicle-Roadside Cooperation and Safety: A Foundation for Safety and Vehicle-Infrastructure Integration Research.** *IEEE Proc. Intell. Transp. Syst.*, pp. 9 – 16, 2006.
- Mitchell, T.M. **Machine Learning**, McGraw Hill, 1997.
- Naranjo, J.; Gonzalez, C.; Pedro, T.; Garcia, R.; Alonso, J.; Sotelo, M. **AUTOPIA Architecture for Automatic Driving and Maneuvering.** *IEEE Proc. Intell. Transp. Syst.*, pp. 1220-1225, 2006.
- Ohtsuka, E. **Communications policy and ITS in Japan**, *8<sup>th</sup> International Conference on ITS Telecommunications*, pp.24-24, Oct. 2008.
- Ozguner, U.; Stiller, C.; Redmill, K. **Systems for Safety and Autonomous Behavior in Cars: The DARPA Grand Challenge Experience.** *Proc. IEEE*, vol. 95, no. 2, pp. 397 – 412, Feb. 2007.
- Pal, N. R.; Chakraborty, S. **Fuzzy Rule Extraction from ID3-Type Decision Trees for Real Data.** *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics* 05, vol. 31, pp. 745-754, 2001
- Paromtchik, I.E.; Laugier, C. **Motion Generation and Control for Parking an Autonomous Vehicle.** *Proceeding of the IEEE International Conference on Robotics and Automation*, 3117-3122, 1996.
- Pei, Z. **A Formalism to Extract Fuzzy If-Then Rules from Numerical Data Using Genetic Algorithms.** In: *Int. Symposium on Evolving Fuzzy Systems*, pp. 143-147, 2006.
- Quinlan, J. R. **C4.5: Programs for Machine Learning.** San Mateo, Inc. MorganKaufmann, 1993.
- Rajamani, R. **Vehicle Dynamics and Control.** Springer, 2006.
- Rogers, R. **Integrated INU/DGPS for Autonomous Vehicle Navigation.** *IEEE Local. Navig. Symp.*, pp. 471 – 476, Apr. 1996.
- Roman, C.; Reinholtz, C.; Tech, V. **Robust Course-Boundary Extraction Algorithms for Autonomous Vehicles.** *IEEE Vision – Based Driving Assistance*, 32-39, 1998.
- Russell, S.; Norvig, P. **Artificial Intelligence – A Modern Approach.** Prentice Hall Series in Artificial Intelligence, New Jersey, 1995.
- Shladover, S. **Path at 20 – History and Major Milestones.** in *Proc. IEEE Intell. Transp. Syst.*, pp. 22 – 29, 2006.
- Shladover, S.; Desoer, C.; Hedrick, J.; Tomizuka, M. **Automated Vehicle Control Developments in the PATH Program.** *IEEE Trans. Vehicular Tech.*, vol. 40, no. 1, pp. 114 – 130, Feb. 1991.
- Smith, J.F. **Co-evolutionary Data Mining to Discovery Rules for Fuzzy Resource Management Lecture Notes in Computer Science**, Springer, 2002.
- Stampfle, D. ; Becker, J. **Performance Evaluation of Automotive Sensor Data Fusion.** *IEEE Proc. Intell. Trans. Syst.*, pp. 50-55, Sept. 2005.
- Suzuki, T.; Aoki, K.; Tachibana, A.; Moribe, H. **An automated highway vehicle system using computer vision-Recognition of white guidelines.** in *1992 JSAE Autumn Convention Proc. 924*, vol. 1, pp. 161-164, Oct. 1992.

- Thorpe, C.; Hebert, M.; Kanade, T.; Shafer, S. **Vision and Navigation for the Carnegie-Mellon Navlab.** *IEEE Trans. Patt. Analysis Mach. Intell.*, vol. 10, no. 3, pp. 362 – 373, Dec. 1988.
- Thrun S.; Montemerlo M.; Dahlkamp H. **Stanley: The robot that won the DARPA Grand Challenge.** *Journal of Field Robotics*, vol 2, no. 9, pp. 661-692, Sept. 2006.
- Thrun, S.; Fox, D.; Dellaert, F.; Burgard, W. **Particle filters for mobile robot localization.** A. Doucet, N. de Freitas and N. Gordon eds. in *Sequential Monte Carlo Methods in Practice*. New York Springer Verlag, New York, 2001
- Timmis, J.; Edmonds, C. **A Comment on Opt-AiNET: An Immune Network Algorithm for Optimisation.** *Lecture Notes in Computer Science*, pp. 308.317, 2004.
- Triantaphyllon, E.; Felice, G. **Data Mining and Knowledge Discovery Approaches based on Rule Induction Techniques.** Springer, New York, 2006.
- Tsugawa, S. **Inter-Vehicle Communications and their Applications to Intelligent Vehicle: an Overview.** *IEEE Intell. Vehicle Symp.*, vol. 2 pp. 17 – 21, June 2002.
- Tsugawa, S. **Vision-Based Vehicles in Japan: Machine Vision Systems and Driving Control Systems.** *IEEE Transactions on Industrial Electronics*, vol. 41, 398-730, 1994.
- Tsugawa, S.; Hirose, T.; Yatabe, T. **Studies on the intelligent vehicle.** *Rep. Mechanical Eng. Lab.*, no. 156 Nov. 1991.
- Turk, M.; Morgenthaler, D.; Gremban, K.; Marra, M. **VITS – A vision system for autonomous land vehicle navigation.** *IEEE Trans. Patt. Analysis Mach. Intell.*, vol. 10, no. 3, pp. 342 – 361, May 1988.
- Vachkkov, G.; Ishihara, H. **Unsupervised learning algorithms for comparison and analysis of images.** *IEEE International Conference on Mechatronics and Automation*, pp 415-420, Aug. 2008.
- Wang, F.; Mirchandani, P.; Wang, Z. **The VISTA Project and ITS Applications.** *IEEE Intell. Syst.*, vol. 17, no. 6, pp. 72 – 75, Nov. – Dec. 2002.
- Wang, L.; Mendel, J.M. **Generating Fuzzy Rules by Learning from Examples.** *IEEE Transactions on Systems, Man, and Cybernetics* 6, vol. 22, pp. 1414-1427, 1992.
- Witten, I.H.; Frank, E. **Data Mining – Practical Machine Learning Tools and Techniques.** Second Edition, Elsevier, San Francisco, 2005.
- Yagi, Y.; Nishizawa, Y.; Yachida, M. **Map-Based Navigation for a Mobile Robot with Omnidirectional Image Sensor COPIS.** *IEEE Transactions on Robotics and Automation*, vol. 11, 634-647, 1995.
- Yang, L.; Wang, F. **Driving into Intelligent Spaces with pervasive Communications.** *IEEE Intell. Syst.*, vol. 22, no. 1, pp. 12 – 15, Jan. – Feb. 2007.
- Yen, J.; Pfluger, N. **A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation.** *IEEE Systems, Man, and Cybernetics* vol.25, 971 – 978, 1995.
- Zadeh, L. A. **Fuzzy Sets.** *Information and Control*, Vol. 8, pp 338–352, 1965.

Zadeh, L. A. **The Concept of Linguistic Variable and its Application to Approximate Reasoning.** *Information Sciences*, Vol. 8, pp 199-249 (part I), 301-357 (partII), 1975.

Zhao, Y.; Collins, E.G.; Dunlap, D. **Design of genetic fuzzy parallel parking control systems.**  
In: *Proc. American Control Conference*, vol. 5, 4107-4112, 2003.