UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA

RAFAEL VIEIRA DE SOUSA

ROBÔ AGRÍCOLA MÓVEL (RAM): UMA ARQUITETURA BASEADA EM COMPORTAMENTOS HIERÁRQUICOS E DIFUSOS PARA SISTEMAS AUTÔNOMOS DE GUIAGEM E NAVEGAÇÃO

São Carlos
2007

RAFAEL VIEIRA DE SOUSA

ROBÔ AGRÍCOLA MÓVEL (RAM): UMA ARQUITETURA BASEADA EM COMPORTAMENTOS HIERÁRQUICOS E DIFUSOS PARA SISTEMAS AUTÔNOMOS DE GUIAGEM E NAVEGAÇÃO

Tese apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Doutor em Engenharia Mecânica.

Área de concentração: Engenharia Mecânica
Orientador: Prof. Dr. Arthur José Vieira Porto

São Carlos
2007

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

S725r

Sousa, Rafael Vieira de
Robô agrícola móvel (RAM) : uma arquitetura baseada em comportamentos hierárquicos e difusos para sistemas autônomos de guiagem e navegação / Rafael Vieira de Sousa ; orientador Arthur José Vieira Porto. -- São Carlos, 2007.

Tese (Doutorado) - Programa de Pós-Graduação e Área de Concentração em Engenharia Mecânica -- Escola de Engenharia de São Carlos da Universidade de São Paulo.

1. Arquitetura de comportamento. 2. Automação de máquinas agrícolas. 3. Inteligência computacional. 4. Lógica difusa. 5. Robótica. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **RAFAEL VIEIRA DE SOUSA**

Tese defendida e julgada em 10/08/2007 perante a Comissão Julgadora:

APROVADO

Prof. Titular **ARTHUR JOSÉ VIEIRA PORTO (Orientador)**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Dr. **RICARDO YASSUSHI INAMASU**
(Empresa Brasileira de Pesquisa Agropecuária/EMBRAPA/CNPDIA)

APROVADO

Prof. Dr. **MARIO LUIZ TRONCO**
(Universidade Estadual Paulista "Julio de Mesquita Filho"/UNESP-Campus de São José do Rio Preto)

aprovado

Prof. Associado **JUN OKAMOTO JUNIOR**
(Escola Politécnica/USP)

APROVADO

Prof. Associado **CARLOS EDUARDO CUGNASCA**
(Escola Politécnica/USP)

Prof. Associado **JONAS DE CARVALHO**
Coordenador do Programa de Pós-Graduação
em Engenharia Mecânica

Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Presidente da Comissão da Pós-Graduação da EESC

À Julieta, minha adorável esposa, e ao Leonardo Vinicius, meu querido filho.

AGRADECIMENTOS

Ao Professor Arthur José Vieira Porto pelo companheirismo, pelo incentivo e pela orientação para desenvolvimento deste trabalho e para meu desenvolvimento profissional.

Ao Professor Ricardo Yassushi Inamasu pela amizade e pelas inúmeras e valiosas contribuições.

A todos os colegas, professores e funcionários do Laboratório de Simulação e Controle do Departamento de Engenharia Mecânica da EESC/USP pela colaboração.

Aos Professores Daniel Toal e Colin Flanagan, e a todos os colegas do Grupo de Sistemas de Controle e Instrumentação da Universidade de Limerick (Limerick, República da Irlanda – EU) pelo suporte ao desenvolvimento deste trabalho.

A todos os colegas, pesquisadores e funcionários da Embrapa Instrumentação Agropecuária pela colaboração.

À Fundação de Amparo a Pesquisa do Estado de São Paulo pela bolsa de estudo concedida.

"Nunca ande apenas pelos caminhos traçados, pois eles conduzem somente até aonde os outros já foram."

Alexander Graham Bell

RESUMO

SOUSA, R.V. **Robô Agrícola Móvel (RAM): uma arquitetura baseada em comportamentos hierárquicos e difusos para sistemas autônomos de guiagem e navegação.** 2007, 194p. Tese (Doutorado em Engenharia Mecânica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, S.P. - 2007.

Avanços positivos nas pesquisas em Veículos Agrícolas Autônomos (VAA's) e de Robôs Agrícolas Móveis (RAM's) têm sido conquistados nos últimos anos. Entretanto, um número limitado de trabalhos tem proposto sistemas robustos baseados em arquiteturas robóticas capazes de realizar operações múltiplas e independentes, bem como adaptar-se às mudanças ambientais no campo. Por outro lado, em outras áreas de pesquisa, um número representativo de arquiteturas baseadas em comportamentos tem sido proposto para guiagem e para navegação autônomas de robôs móveis em ambientes não estruturados e/ou não explorados.

No presente trabalho, uma arquitetura robótica baseada em comportamentos é desenvolvida para guiagem e navegação de RAM's e VAA's. Regras difusas são utilizadas para compor e coordenar comportamentos primitivos e comportamentos complexos. O desenvolvimento inclui: a implementação e a simulação da arquitetura em um mini-robô; a avaliação e a caracterização de sensores para o módulo perceptivo da arquitetura; e a aplicação de um método de análise baseado em um modelo matemático para auxiliar a composição de uma rede de comunicação digital baseada no protocolo CAN para sistemas de controle robóticos.

Experimentos foram realizados para avaliar os comportamentos implementados e para avaliar a capacidade de operação da plataforma robótica em um ambiente agrícola simulado. Os resultados mostraram a viabilidade da abordagem proposta. A modularidade da arquitetura com a utilização de controladores difusos descentralizados simplificou a implementação da arquitetura. O processo de arbitragem difuso mostrou-se um método simples e viável para implementar a coordenação de comportamentos e para compor comportamentos complexos.

A avaliação e caracterização de sensores, em especial de um sensor ultrassônico, permitiram a determinação de condições operacionais para a utilização desses sensores em VAA's ou em RAM's. A aplicação do modelo matemático para a rede permitiu a análise de desempenho da rede CAN para diferentes equipamentos e sob diferentes parâmetros de configuração para aplicação em um RAM.

Palavras-chave: Arquitetura de Comportamentos, Automação de Máquinas Agrícolas, Inteligência Computacional, Lógica Difusa, Robótica.

ABSTRACT

Positive advances on AAV (Agricultural Autonomous Vehicle) and MAR (Mobile Agricultural Robot) research are noticed in recent years. However, a limited number of works have proposed reliable systems based on a robotic architectures that are able to perform multiple and independent operations, as well as to self-adapt under changing environmental conditions in the field. In other hand, in other research areas a considerable number of behavior-based architectures have been proposed for mobile robot for autonomous guidance and navigation in unstructured and/or in unexplored environments.

At this work, a robotic behavior-based architecture is developed for guidance and navigation of AAV and MAR. Fuzzy rules are used to compose and coordinate the primitive and the complex behaviors. The development includes: the implementation and the simulation of the proposed architecture on a mini-robot; the evaluation and characterization of sensors for the perceptive module of the architecture; and the application of an analysis method based on a mathematical model to assist the composition of a digital communication networks based on the CAN protocol for robot control systems.

Experiments have been performed to evaluate the implemented behaviors and to evaluate the operation ability of the robotic platform on a simulated agricultural environment. The results show the feasibility of the proposed approach. The modularity of the architecture by using decentralized fuzzy controllers simplifies the implementation of the robotic architecture. The fuzzy arbitration process is an easy and a feasible method to implement the behavior coordination and to compose complex behaviors.

The sensors evaluation and characterization, in particular of an ultrasonic sensor, have allowed establishing operational conditions for using them in AAV or in a MAR. The application of the network mathematical model has allowed the performance analysis of the CAN-based networks under differentiated equipment and configuration parameters for applications in a MAR.

Keywords: Agricultural Machinery Automation, Behaviors Architecture, Computational Intelligence, Fuzzy Logic, Robotic.

LISTA DE FIGURAS

Figura 1 - Ilustração dos módulos componentes de uma arquitetura robótica segundo a abordagem deliberativa. Fonte: adaptado de Brooks (1998). 15

Figura 2 - Ilustração dos módulos componentes de uma arquitetura robótica segundo a abordagem reativa. Fonte: adaptado de Brooks (1986). 17

Figura 3 - Ilustração dos módulos componentes de uma arquitetura robótica híbrida. 19

Figura 4 - Exemplo de uma arquitetura híbrida robótica baseada em comportamentos. Fonte: adaptado de Murphy (2000). 21

Figura 5 - Exemplo de funções de pertinência para variável lingüística *Distância*. 31

Figura 6 - Método de inferência difusa max-min. 38

Figura 7 - Módulos de um sistema de controle difuso. Fonte: adaptado de Arkin (1998). 41

Figura 8 - Controlador difuso baseado o método *Context-Dependent Blending*. Fonte: adaptado de Saffiotti, Ruspini e Konolige (1999) 43

Figura 9 - Arquitetura robótica DAMN aplicada a um veículo submarino autônomo. Fonte: adaptado de Williams et al. (2001) e de Rosenblatt, Williams e Durrant-Whyte (2002) 48

Figura 10 - Arquitetura robótica para navegação em terrenos irregulares. Fonte: adaptado de Seraji e Howard (2002) 52

Figura 11 - Arquitetura robótica baseada em sistemas difuso-genéticos. Fonte: adaptado de Hagras, Callaghan e Colley (2001). 55

Figura 12 - :Ilustração de sistemas de coordenadas para sistema de guiagem e navegação. Fonte: adaptado de Noguchi, Ishii e Terão (1997) 66

Figura 13 - Ilustração de uma rede ISO 11783. 81

Figura 14 - Arquitetura robótica híbrida baseada em comportamentos para a guiagem e para a navegação autônomas de RAM's e VAA's. 87

Figura 15 - Exemplo de composição e integração do módulo reativo e do módulo de arbitragem da arquitetura híbrida proposta. 91

Figura 16 - Vistas lateral de superior do módulo básico do mini-robô Khepera. Fonte: adaptado de K-TEAM (2007). 94

Figura 17 - Plataforma robótica baseada no mini-robô Khepera. 96

Figura 18 - RDL's montados no mini-robô Khepera para seguir linhas. 99

Figura 19 - Variáveis linguísticas utilizadas pelos comportamentos *Evitar Obstáculo* e *Beira Linhas* 102

Figura 20 - Variáveis linguísticas utilizadas pelo comportamento *Seguir Linha.* 104

Figura 21 - Variáveis linguísticas relacionadas com o acionamento dos motos-redutores do mini-robô 105

Figura 22 - Base de conhecimento para o comportamento *evitar obstáculo*. 107

Figura 23 - Bases de conhecimento para o comportamento *beirar linhas*. 108

Figura 24 - Base de conhecimento para o comportamento *seguir linha*. 109

Figura 25 - Variáveis linguísticas relacionadas com o controlador difuso para arbitragem de comportamentos. 111

Figura 26 - Variável lingüística de saída relacionada com o controlador difuso para arbitragem de comportamentos. 112

Figura 27 - Base de conhecimento do árbitro difuso. 112

Figura 28 - Composição e integração do módulo reativo e do módulo de arbitragem. 115

Figura 29 - Filtro de Kalman indireto para fusão de sensores entre os odômetros e a bússola eletrônica. 118

Figura 30 - Aplicativo com interface gráfica para experimentos com odometria. 119

Figura 31 - Curvas experimentais para ajuste do Filtro de Kalman indireto para fusão de sensores entre os odômetros e a bússola eletrônica. 121

Figura 32 - Planta do labirinto de madeira utilizados nos experimentos com a arquitetura robótica híbrida. 122

Figura 33 - Percurso fechado realizado pelo robô seguindo linha pintada no piso. 125

Figura 34 - Tarefa de exploração de ambiente para avaliação dos comportamentos e do processo de arbitragem. 126

Figura 35 - Detalhamento da trajetória do robô e da ativação de comportamentos em tarefa para exploração de ambiente. 128

Figura 36 - Experimento realizado em um ambiente simulado com três linhas com plantas (foto do robô durante no experimento e gráfico com trajetória coletada por odometria). 131

Figura 37 - Detalhamento da trajetória do robô e da ativação de comportamentos em tarefa de navegação entre linhas de plantas. 133

Figura 38 - Sub-sistema Laboratório/Estação Base e Sub-sistema de Campo componentes da plataforma robótica agrícola experimental. 140

Figura 39 - Vista isométrica da estrutura mecânica do RAM: (a) vista tridimensional com ilustração dos módulos principais; (b) vista superior com detalhamento das medidas principais; (c) vista frontal com detalhamento da altura da estrutura e do mecanismo de redimensionamento; e (d) vista tridimensional com ilustração de um MGP e do suporte para baterias em um CHI. 144

Figura 40 - Vistas do MGP: (a) vista isométrica; e (b) vistas lateral e frontal com detalhamento das principais partes componentes. 145

Figura 41 - Fotos do RAM: (a) estrutura mecânica completa; e (b) MGP acoplado a CHI. 146

Figura 42 - Vistas da estrutura do RAM ilustrando os dispositivos principais interligados pela rede CAN. 147

Figura 43 - Rede CAN composta por nó controlador de motor e computador embarcável. 148

Figura 44 - Diagrama da interface CAN. 151

Figura 45 - Fluxograma do programa de controle do transdutor do sensor Polaroid 6500 e interface gráfica do programa para PC para controle das medições. 154

Figura 46 - Medições de distâncias (*d*) com o sensor ultrassônico em relação a um anteparo vegetal. 156

Figura 47 - Medições de referência realizadas em relação ao anteparo plano de madeira: médias (a) e erros relativos (b) 158

Figura 48 - Média das medições em relação ao anteparo vegetal: (a) anteparo parado e (b) anteparo em movimento. 159

Figura 49 - Erro relativo das medições em relação ao anteparo vegetal: (a) anteparo parado e (b) anteparo em movimento. 161

Figura 50 - Telas de entrada de dados do programa de simulação de rede CAN: (a) Entrada de Dados 1; (b) Entrada de Dados 2. 164

Figura 51 - Tela de saída de dados do programa de simulação de rede CAN 165

Figura 52 - Fluxograma baseado a metodologia de análise CAN. Fonte: adaptado de Godoy (2007). 168

LISTA DE TABELAS

Tabela 1 - Características de arquiteturas robóticas baseadas em sistemas difusos. 57

Tabela 2 - Dados para a utilização do modelo matemático da rede CAN. 167

Tabela 3 – Dispositivos para a rede CAN referentes à arquitetura híbrida proposta. 169

Tabela 4 – Tempo de transmissão para o conjunto de mensagens da rede CAN. 172

Tabela 5 – Resultados obtidos para a rede CAN na primeira simulação. 173

Tabela 6 – Resultados obtidos para a rede CAN na segunda simulação. 173

Tabela 7 – Resultados obtidos para a rede CAN na terceira simulação. 174

LISTA DE ABREVIATURAS E SIGLAS

VAA - Veículo Agrícola Autônomo
RAM - Robôs Agrícolas Móveis
GPS - Global Positioning System
DGPS - Differential Global Positioning System
INS – Inertial Navigation System:
GDS - Geomagnetic Direction Sensor
FK - Filtro de Kalman
FKE - Filtro de Kalman Estendido
IMU - Inertial Measurement Unit
RTK-GPS - Real Time Kinematic GPS
CCD - Charged Coupled Device
RGB - Red, Green and Blue
PC - Personal Computer
FPS - Frames per Second
TH - Transformada de Hough
DIN - Deutsches Institut für Normung
CAN- Controller Area Network
ECU - Electronic Control Unit
ISO - International Organization for Standardization
ASCII - American Standard Code for Information Interchange
RDL - Light Dependant Resistor
OSI - Open Systems Interconnection
PCI - Peripheral Component Interconnect
PCMCIA - Personal Computer Memory Card International Association
IHM- Interface Homem-Máquina
USART - Universal Synchronous/Asynchronous Receiver Transmitter
EEPROM - Electrically Erasable Programmable Read-Only Memory
SIG - Sistema de Informação Geográfico
VRT - Variable Rate Technology

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

1. INTRODUÇÃO 1

1.1. Objetivo 6

1.2. Estrutura do trabalho 7

2. ROBÔS MÓVEIS 9

2.1. Arquiteturas para robôs móveis 9

2.2. Evolução histórica dos paradigmas deliberativo e reativo 14

2.3. Arquiteturas híbridas 19

3. ARQUITETURAS ROBÓTICAS BASEADAS EM COMPORTAMENTOS DIFUSOS 28

3.1. Fundamentos da lógica difusa 29

3.1.1 Função de pertinência e fuzzificação 29

3.1.2 Variáveis lingüísticas 30

3.1.3 Operadores difusos 33

3.1.4 Relação difusa 35

3.1.5 Base de regas 36

3.1.6 Inferência difusa 37

3.1.7 Defuzzificação 39

3.2. Controladores e comportamentos difusos 40

3.3. Controladores difusos para navegação em ambientes pouco estruturados e dinâmicos 46

4. SISTEMAS AGRÍCOLAS AUTÔNOMOS 62

4.1. Abordagem deliberativa com enfoque no sistema de percepção 64

4.1.1 Filtro de kalman em sistemas agrícolas autônomos 65

4.1.2 Sistemas agrícolas autônomos com enfoque no sistema de percepção 69

4.2. Abordagem híbrida baseada em comportamentos 73

4.3. Sistemas distribuídos 79

5. ARQUITETURA ROBÓTICA HÍBRIDA BASEADA EM COMPORTAMENTOS PARA GUIAGEM E NAVEGAÇÃO AUTÔNOMAS 83

5.1. Proposta da arquitetura híbrida baseada em comportamentos difusos 85

5.2. Plataforma robótica para simulação e análise da arquitetura proposta 91

5.2.1 Plataforma khepera 94
5.3. Módulo reativo – comportamentos difusos 98
5.3.1 Variáveis lingüísticas associadas aos comportamentos 100
5.3.2 Bases de regras componentes dos comportamentos 106
5.4. Módulo de arbitragem – árbitro difuso 110
5.5. Módulo de mapeamento e controle de tarefas 116
5.6. Simulação e análise da arquitetura proposta - resultados 122
5.6.1 Guiagem e mapeamento sobre uma trajetória 124
5.6.2 Exploração e navegação segura 126
5.6.3 Guiagem e navegação entre linhas de plantas 131
5.7. Considerações gerais sobre a arquitetura híbrida proposta 136
6. ROBÔ AGRÍCOLA MÓVEL - APONTAMENTOS SOBRE A IMPLEMENTAÇÃO DA ARQUITETURA PROPOSTA 139
6.1. Estrutura mecatrônica do RAM 143
6.2. Caracterização e avaliação do comportamento de sensor ultrassônico para sistemas agrícolas autônomos 152
6.2.1 Materiais e métodos aplicados para caracterização do sensor ultrassônico 153
6.2.2 Medições de distância com o sensor ultrassônico 157
6.3. Simulação e análise de rede CAN com dispositivos para a arquitetura robótica proposta 162
6.3.1 Programa de simulação para rede CAN 164
6.3.2 Metodologia de análise para rede do RAM 167
6.3.3 Resultados da simulação para rede do RAM 170
6.4. Considerações gerais sobre a implementação da arquitetura robótica no RAM 175
7. CONCLUSÕES 179
7.1. Principais contribuições 179
7.2. Trabalhos futuros 182
8. REFERÊNCIAS BIBLIOGRÁFICAS 184

1. INTRODUÇÃO

O panorama Mundial da agricultura tem mudado acentuadamente nos últimos anos. A necessidade de incremento da produtividade, evitando a ampliação da área de cultivo e buscando minimizar impactos ambientais, posicionou a Automação dos Processos Agrícolas como um fator essencial para o desenvolvimento da agricultura, em especial para suas especializações, a Agroindústria e o Agronegócio. Esses apontamentos foram reafirmados na recente transição de milênio e de século por cientistas (KUTZBACH, 2000; AUERNHAMMER, 2001; KASSLER, 2001; EARL, THOMAS e BLACKMORE, 2000) e por organizações nacionais e internacionais (FAO, 2007; TECA, 2007), que os destacam como essenciais ao desenvolvimento sustentável da agricultura.

Atualmente, a Biotecnologia e a Automação de Processos Agrícolas tornaram-se elementos fundamentais para inserção ou manutenção da competitividade comercial no mercado globalizado de países com vocação agrícola. O baixo nível de aplicação dessas tecnologias é fator de exclusão comercial, principalmente para alguns países, como o Brasil, que tem grande potencial para desenvolvimento da agroindústria e do agronegócio, e cujo papel da agricultura na economia tornou-se tão importante quanto o papel de outras atividades econômicas. A constatação da gravidade desse fato tem levado os governos e organizações buscarem formas de fomento a pesquisa e a disseminação de conhecimento para promover a competitividade comercial e a sustentabilidade da agricultura (FAO, 2007; TECA, 2007, passim).

Entre os conceitos atuais que buscam a automação como suporte para as atividades agrícolas, destaca-se a Agricultura de Precisão, que neste trabalho é considerada como a utilização de tecnologias para inferir a variabilidade espacial e temporal de áreas de cultivo, e para definir manejo diferenciado de solo, insumos e das próprias culturas (ADRIAN, NORWOOD E MASKC, 2006, passim; MOLIN, 2004, passim).

As práticas relacionadas com a Agricultura de Precisão ou que buscam a otimização na utilização dos insumos agrícolas e o uso racional da terra trouxeram com as suas evoluções uma série de desafios para a pesquisa. A escala de amostragem das variáveis envolvidas e a precisão na execução das tarefas para essas práticas, muitas vezes, são maiores que as exigidas pela agricultura tradicional, demandando tecnologias para veículos agrícolas que as viabilizem em custo e eficiência (EARL, THOMAS e BLACKMORE, 2000; KUTZBACH, 2000; AUERNHAMMER, 2001; REID, 2004, PEDERSEN et al., 2005).

A pesquisa em tecnologias para veículos agrícolas e busca por inovações para atender as necessidades das novas práticas culminou em alguns produtos comerciais que, atualmente, constituem o estado-da-arte das tecnologias para automação de máquinas agrícolas. Dentre essas tecnologias e respectivas metodologias destacam-se:

- *VRT (Variable Rate Technology*: tecnologia de Aplicação à Taxa Variável): tecnologias aplicadas a máquinas agrícolas para permitir a aplicação controlada de insumos (fertilização e pulverização) ou para permitir o controle de plantio (espaçamento e quantidade de sementes). Normalmente necessitam de controle de velocidade mais preciso para a máquina agrícola e de um sistema

computacional de apoio prévio para estudo e geração de mapa de recomendação;

- *Tecnologias On-The-Go*: sensoriamento, processamento (tomada de decisão) e atuação durante o movimento da máquina. Frequentemente demandam controle mais preciso de navegação e redução no tempo de resposta do sistema;

- *Piloto Automático*: tecnologia aplicada a máquinas agrícolas para permitir a navegação mais precisa através de sistemas semi-autônomos ou autônomos. Utilizados normalmente para viabilização de trabalho noturno, plantio, aplicação precisa de insumos e suporte para as técnicas citadas nos destaques anteriores.

Se as tecnologias descritas são frutos de pesquisa e de desenvolvimentos realizados por grupos de pesquisa em máquinas agrícolas nas ultimas duas décadas, atualmente, um dos focos de pesquisa nessa área tem sido a busca por sistemas robóticos que tornem viável a aplicação de Veículos Agrícolas Autônomos (VAA's) e de Robôs Agrícolas Móveis (RAM's). A importância da pesquisa nesses temas pode ser constatada pelo número expressivo de publicações em revistas científicas nacionais e internacionais. Na mudança de milênio, a aplicação de VAA's e de RAM's na Europa (KEICHER & SEUFERT, 2000) e em países como EUA (REID et. al., 2000) e Japão (TORII, 2000) já era vista como uma forte tendência, verificando-se consideráveis avanços em pesquisas associados a esses temas.

Além da necessidade de inovações em máquinas agrícolas para atender às novas práticas agrícolas, outros fatores, como o altíssimo custo de mão de obra, o envelhecimento da população rural sem perspectivas de renovação, a necessidade de minimizar a exposição dos operadores a atividades insalubres e a redução dos

custos da eletrônica necessária para construção de tais sistemas, têm incentivado e justificado pesquisas em VAA's e RAM's por empresas como AGCO, John Deer, CNH e Yanmar, e por grupos, como os apresentados em Keicher e Seufert (2000) Reid et al. (2000) e Torii (2000). Nos últimos 5 anos, trabalhos como Hague, Southall, e Tillett (2002), Southall et al. (2002), Åstrand e Baerveldt (2002), Bak e Jakobson (2004) e Blackmore et al. (2004) têm apresentado soluções viáveis para o desenvolvimento de sistemas robóticos para máquinas agrícolas semi-autônomas ou autônomas que possibilitam operações mais precisas para reduzir custos e minimizar o impacto ambiental de tarefas agrícolas. Porém, como é relatado por Reid et al. (2000) e reafirmado por Reid (2004) e por Pedersen et al. (2005) há um número limitado de trabalhos direcionados ao desenvolvimento de sistemas com funções autônomas diversificadas, com comportamentos robóticos capazes de atuarem em ambientes menos estruturados, adaptando-se às condições do campo para realização de tarefas diversas.

Por outro lado, atualmente diversos trabalhos de pesquisa em outras áreas além da área agrícola têm apresentado soluções notáveis para construção arquiteturas robóticas para robôs móveis que atuam em ambientes pouco estruturados e dinâmicos (BRADSHAW, 2002). Entre essas soluções, um número expressivo de grupos de pesquisa tem conseguido resultados positivos com o desenvolvimento de arquiteturas robóticas baseadas em comportamentos que utilizam Lógica Difusa (*Fuzzy Logic*). Entre as propostas mais notáveis podem ser destacadas as apresentadas em Hagras, Callaghan e Colley (2001), em Rosenblatt, Williams e Durrant-Whyte (2002) e em Seraji e Howard (2002). Trabalhos atuais utilizam tais métodos para composição de comportamentos robóticos, assim como a composição de módulos de gerenciamento para as arquiteturas, como, por exemplo,

módulos de coordenação de comportamentos (BLACKMORE et. al., 2004; HAGRAS, CALLAGHAN e COLLEY, 2004; IP, RAD e WONG, 2004; YANG, MOALLEM e PATEL, 2005; PEDERSEN et al., 2005; MAALOUFA, SAADA e SALIAHB, 2006; REMONDINI e SAFFIOTTI, 2006).

A realidade brasileira da pesquisa em robôs para sistemas agrícolas é muito diferente e a aplicação desse tipo de tecnologia parece estar mais distante. Pelo Diretório de Grupos de Pesquisa no Brasil do CNPq (PLATAFORMA LATTES, 2002) verifica-se que em 2002 no Brasil não havia grupo desenvolvendo tecnologias neste tema para auxiliar a resolver os problemas do novo cenário da área agrícola. Essa situação permanece e coloca o País numa posição cada vez mais de consumidor de tecnologia, que não raro, é inadequada para o tipo de que agricultura necessita.

Alguns esforços recentes podem ser identificados no Brasil para reverter essa situação. Nos anais do Congresso Brasileiro de Agricultura de Precisão de 2004, realizado em Piracicaba-SP, na Escola Superior de Agricultura Luiz de Queiroz (ESALQ-USP), coordenado pelo Departamento de Engenharia Rural, no período de 17 a 19 de maio de 2004 (ConBAP 2004), pode-se verificar um considerável número de trabalhos nessas áreas em relação a eventos anteriores, e, inclusive, uma sessão especifica sobre guiagem e navegação autônomas com palestras e debates sobre o tema entre representantes de empresas e pesquisadores. Em 2006, o mesmo evento realizado em São Pedro-SP, no período de 04 a 07 de junho de 2006 (ConBAP 2006), contou com uma sessão temática sobre robôs agrícolas e com a presença do Professor Simon Blackmore (BLACKMORE et al., 2004; BLACKMORE e GRIEPENTRONG, 2006), respeitado pesquisador por suas contribuições ao desenvolvimento da Agricultura de Precisão na Europa e por sua contribuições para a pesquisa de robôs agrícolas. Além dessas sessões temáticas em eventos, em

2006, foi realizado o Workshop em Tecnologia de Robótica para Agricultura, promovido pelo Laboratório de Simulação e Controle do Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos da USP (SEM/EESC-USP) e pela Embrapa Instrumentação Agropecuária de São Carlos (Embrapa Cnpdia), com o objetivo de iniciar trabalhos interdisciplinares, bem como iniciar ações que integrem grupos multidisciplinares e multi-institucional, promovendo e fomentando o desenvolvimento das referidas tecnologias no país .

Posto isso, há motivação suficiente para o desenvolvimento de grupos nacionais de pesquisa nessa área e para o desenvolvimento de sistemas que possibilitam a construção e utilização de VAA's e RAM's com características próprias para as culturas nacionais, como café, cana-de-açucar, soja e citrus.

1.1. Objetivo

O objetivo maior do trabalho é contribuir para o desenvolvimento de uma arquitetura robótica híbrida robusta e flexível para viabilizar a construção de VAA's e RAM's. A contribuição é obtida pelo objetivo específico do trabalho em questão, que é a proposta e avaliação através de simulação de uma arquitetura robótica híbrida baseada em comportamentos difusos para sistemas de guiagem e navegação autônomas de VAA´s e RAM´s.

As características principais almejadas para a arquitetura são:

- Potencializar a utilização de estruturas próprias do ambiente agrícola como as linhas de plantio e permitir a navegação autônoma em diferentes tipos de cultura e fases das culturas agrícolas;

- Possuir flexibilidade para que sub-sistemas de percepção, controle e atuação sejam inseridos ou removidos em acordo com as necessidades de cada operação agrícola em campo ou para possibilitar o teste de novos dispositivos e metodologias relacionadas com controle de navegação;

- Permitir a utilização de sistemas distribuídos baseados em redes de campo para comunicação de dados e para controle via rede, além de permitir a integração da arquitetura e dos dispositivos a esta relacionados, em máquinas agrícolas convencionais.

Faz parte da proposta a caracterização de dispositivos para aplicação junto a arquitetura e também a avaliação do desempenho da arquitetura para sua utilização em sistemas de controle via rede.

1.2. Estrutura do trabalho

O Capítulo 2 e o Capítulo 3 a seguir apresentam o estado-da-arte das metodologias e tecnologias aplicadas a robôs móveis em outras áreas que não a área agrícola. Em especial, no Capítulo 3 são descritas as principais metodologias e tecnologias baseadas em comportamentos e a utilização da lógica difusa em sistemas robóticos, com destaque para utilização de controladores difusos para navegação em ambientes pouco estruturados e dinâmicos. No Capítulo 4 é descrito

o estado-da-arte das metodologias e tecnologias para máquinas agrícolas autônomas, incluindo as pesquisas recentes em robôs agrícolas móveis. No capítulo 5 é proposta um uma arquitetura baseada em comportamentos hierárquicos e difusos para sistemas autônomos de guiagem e navegação de VAA's e RAM's. No mesmo Capítulo 5 são descritos a implementação e a simulação da arquitetura em um mini-robô, assim como os principais experimentos realizados. O Capítulo 6 apresenta apontamentos sobre a implementação da arquitetura híbrida proposta, destacando-se: a estrutura mecatrônica do RAM que está em desenvolvimento; a caracterização e avaliação do comportamento de sensor ultrassônico para aplicações em sistema de guiagem e navegação autônomas; e a simulação e a análise da rede CAN com dispositivos para a arquitetura híbrida proposta. No Capítulo 7 são apresentas as principais conclusões do trabalho destacando-se as suas principais contribuições e as possibilidades para trabalhos futuros.

2. ROBÔS MÓVEIS

As pesquisas em robótica têm buscado desenvolver sistemas inteligentes para viabilizar a construção de sistemas robóticos com capacidade de tomar de decisões, ou seja, sistemas que permitam ao robô extrair informações do ambiente em que ele está imerso e utilizar estas informações na busca de soluções para execução de uma determinada tarefa (RIBEIRO, REALI-COSTA e ROMERO, 2001). Os conceitos relacionados com tais sistemas inteligentes estão contidos no campo da Inteligência Artificial (IA) dentro da teoria de Agentes Artificiais (RUSSEL e NORVIG, 1995).

2.1. Arquiteturas para robôs móveis

As conceituações de robô móvel e agente estão relacionadas. Entretanto, a definição precisa de o que é um *agente* constitui um tema polêmico dentro da teoria de IA. Franklin e Graesser (1996) apresentam uma revisão sobre as definições adotadas por diversos pesquisadores. Dentre essas definições, adota-se para este trabalho a definição proposta em Russel e Norvig (1995) por estar relacionada diretamente com a definição de robô: “agente é tudo que é capaz de perceber seu ambiente por meio de sensores e agir sobre esse ambiente através de atuadores” (RUSSEL e NORVIG, 1995, p. 33). Assim, em harmonia com a definição de agente adotada, um robô móvel pode ser visto como um agente artificial inserido em um

ambiente real, possuindo capacidade de locomoção e com funções que associam estímulos sensoriais a ações (RIBEIRO, REALI-COSTA e ROMERO, 2001).

Junto aos conceitos de agente artificial e de robô móvel, outros dois conceitos são importantes para os propósitos deste trabalho: *arquitetura de controle de robô* (arquitetura robótica) e *comportamento robótico*. A definição do que é uma arquitetura robótica também é um tema divergente entre os autores de pesquisa em IA e robótica. De forma geral, em diversos trabalhos de pesquisa pode-se verificar que uma arquitetura é um conjunto de componentes que interagem entre si e permitem a atuação do sistema inteligente ou robô (ARKIN, 1998, passim; MURPHY, 2000, passim, RUSSEL e NORVIG, 1995, passim).

As divergências sobre o conceito de arquitetura robótica aparecem na forma de construção dos seus componentes e nas interações entre esses componentes. Alguns autores preferem definir uma arquitetura do ponto de vista de programas computacionais ou algoritmos (ARKIN, 1998; RUSSEL e NORVIG, 1995). Há autores que incluem os circuitos e os dispositivos elétricos associados aos programas computacionais ou algoritmos como componentes das arquiteturas. Outros autores adotam definições mais gerais para arquitetura como, por exemplo, para Mataric (1992) uma arquitetura provê princípios de organização de um sistema de controle; em Hayes-Roth et al. (1995) apud Arkin (1998) uma arquitetura é um conjunto estrutural de componentes com percepção, raciocínio e atuação; para Ribeiro, Reali-Costa e Romero (2001) as arquiteturas definem como um robô é projetado a partir de blocos fundamentais com funções que incluem módulos de inibição e de ativação desses blocos e estruturas para planejamento de tarefas.

O desenvolvimento de arquiteturas robóticas demanda esforços conjuntos e multidisciplinares para aplicar técnicas de inteligência artificial em sistemas robóticos (ARKIN, 1998). Nesse contexto, desde a formalização sobre arquiteturas robóticas apresentada em Brooks (1986), as diversas linhas de pesquisa em arquiteturas para sistema inteligentes, principalmente em robótica, podem ser agrupadas em duas abordagens: a Deliberativa e a Reativa (ARKIN, 1998; MURPHY, 2000). A abordagem deliberativa, sendo a mais antiga, é também chamada de abordagem clássica. Nessa abordagem a etapa de processamento (tomada de decisão), entre a percepção e a atuação, usa um modelo interno do ambiente associado a um estágio de planejamento para determinação a atuação a ser realizada (MURPHY, 2000).

Como uma alternativa à abordagem deliberativa consolidou-se com o trabalho Brooks (1986) a abordagem denominada reativa. Nessa abordagem o modelo interno do ambiente não existe e a atuação é baseada em uma resposta direta a um estímulo. Esse elemento de resposta é denominado Comportamento e é através da coordenação de diversos comportamentos que são implementadas as tarefas do sistema inteligente ou do robô. Arkin (1998) relata que parte dos pesquisadores inspirou-se na natureza do comportamento animal para desenvolver sistemas reativos capazes de permitir o máximo de autonomia a robôs móveis na execução de tarefas.

O termo Comportamento está presente em muitos trabalhos de pesquisa em robótica móvel, sendo que, sua definição exata não é um consenso mesmo depois de duas décadas transcorridas desde os trabalhos seminais de Brooks (1986), Payton (1986) e Arkin (1987). Arkin (1998) refere-se a comportamento robótico como sendo uma resposta a um estímulo e apresenta o conceito de Comportamento

Emergente como sendo uma conseqüência da interação de comportamentos individuais com o ambiente de atuação do robô.

Trabalhos mais recentes sobre arquiteturas robóticas baseadas comportamentos adotam definições mais amplas para comportamento do que as formas puramente reativas. Murphy (2000) utiliza a definição de que os comportamentos são entidades que compõem uma arquitetura de controle e relacionam de diferentes formas entradas de sensores à saídas de atuação de acordo com as tarefas a serem executadas e o contexto apresentado pelo meio de atuação do robô em cada instante. Segundo Murphy (2000), os comportamentos podem competir ou cooperar para realização de uma tarefa, além disso, um comportamento pode utilizar mais de um sensor e diferentes comportamentos podem utilizar sensores em comum através de fusão sensorial. Nesse mesmo sentido, Ribeiro, Reali-Costa e Romero (2001) definem comportamento como uma função que relaciona estímulos sensoriais a atuações efetuadas pelo robô. Em ambas as conceituações, os comportamentos possuem uma gradação entre comportamentos puramente reativos, que associam um estímulo diretamente a uma reação, e comportamentos mais deliberativos, nos quais um modelo interno do ambiente associado a um estágio de planejamento é utilizado para determinar a atuação ou para determinar as interações entre os comportamentos.

Junto às conceituações mais recentes sobre comportamento começaram a surgir diversas propostas de sistemas que integrassem a abordagem deliberativa e a reativa, de forma a somar as vantagens e superar deficiências dessas abordagens (ARKIN, 1998; MURPHY, 2000). Nesse sentido, diversos trabalhos têm buscado viabilizar o desenvolvimento de sistemas inteligentes, incluído arquiteturas de controle para robôs móveis, que utilizam benefícios das duas abordagens gerando

assim as arquiteturas denominadas Híbridas (MURPHY, 2000). Detalhes sobre as duas abordagens e a abordagem híbrida serão descritas posteriormente (itens 2.2 e 2.3).

Embora na última década trabalhos de pesquisa tenham proposto diversas arquiteturas de controle para robôs móveis, o cerne para construção dessas arquiteturas, que é a definição das associações entre os diversos estímulos com ações desejadas, ainda constitui um grande desafio para o desenvolvimento de robôs móveis que atuam em ambientes pouco estruturados. A identificação desse desafio e as propostas para superá-los podem ser identificadas em trabalhos de expressão como, por exemplo: Saffioti, Ruspini e Konolige (1999); Rosenblatt, Williams e Durrant-Whyte (2000); Stoytchev e Arkin (2001); Seraji e Howard (2002); Hagras, Colley e Callaghan (2003); Shell e Mataric (2005); e Yang, Moallem e Patel (2005).

Ambientes pouco estruturados podem ser habitados ou ocupados com frequência por seres humanos como, por exemplo, casas, escritórios ou empresas (SAFFIOTI, RUSPINI e KONOLIGE, 1999; STOYTCHEV e ARKIN, 2001; SHELL e MATARIC, 2005), assim como podem ser ambientes não habitados ou pouco freqüentados por seres humanos como, por exemplo, ambientes subaquáticos (ROSENBLATT, WILLIAMS e DURRANT-WHYTE, 2000), agrícolas (GARCIA-ALEGRE et al, 2001; HAGRAS, COLLEY e CALLAGHAN, 2003; PEDERSEN et al, 2005) ou de outros planetas (SERAJI e HOWARD, 2002). Tais ambientes de atuação das arquiteturas propostas por esses autores para seus robôs móveis, muitas vezes denominados dinâmicos ou abertos, caracterizam-se pela falta de conhecimento a priori do ambiente, seja pela frequência com que alguns parâmetros

físicos mudam, tornando difícil seu mapeamento, ou mesmo porque não podem ser mapeados pela dificuldade de acesso.

Associada à imprevisibilidade natural dos ambientes pouco estruturados, a variação temporal e espacial de certos parâmetros físicos como umidade, luminosidade e temperatura, principalmente nos ambientes abertos, elevam a imprecisão natural nas medições dos sensores conduzindo a elevados graus de incerteza como relatado pelos autores Saffioti, Ruspini e Konolige (1999), Rosenblatt, Williams e Durrant-Whyte (2000), Stoytchev. e Arkin (2001), Seraji e Howard (2002), Shell e Mataric (2005) e Yang, Moallem e Patel (2005). Na área agrícola esses desafios são prementes como descrito em trabalhos como, por exemplo, Åstrand e Baerveldt (1999), Garcia-Alegre et al (2001), Hague, Southall e Tillett (2002) e Hagras, Colley e Callaghan (2003), Bak e Jakobson (2004). Sob essa perspectiva, a pesquisa referente à tese em questão teve as atividades focadas na busca de soluções para esses desafios, através de propostas que viabilizam arquiteturas robóticas híbridas baseadas em comportamentos para aplicações agrícolas.

2.2. Evolução histórica dos paradigmas deliberativo e reativo

Como mencionado anteriormente (item 2.1), os sistemas robóticos baseados na abordagem deliberativa utilizam um modelo interno de mundo que é construído a partir de um conhecimento *a priori* sobre o ambiente e que pode ser atualizado com informações adquiridas pelos sensores ao tempo da atuação do robô (ARKIN, 1998). Como relatado por Brooks (1986), tais sistemas são caracterizados como sistemas

verticais por definirem módulos ou estágios totalmente dependentes, não sendo possível para um estágio realizar uma tarefa específica de controle do robô. A Figura 1 mostra a decomposição do sistema de controle de um robô em estágios verticais segundo a abordagem deliberativa.

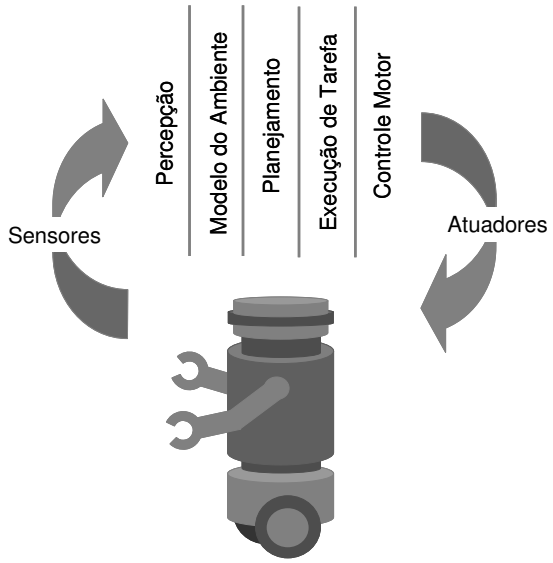


Figura 1 - Ilustração dos módulos componentes de uma arquitetura robótica segundo a abordagem deliberativa. Fonte: adaptado de Brooks (1998).

Como pode ser visto na Figura 1, o módulo de percepção organiza as informações dos sensores e abastece o módulo que contém o modelo do ambiente (modelo interno de mundo). Por sua vez, o modelo do ambiente gera ações que são analisadas pelo módulo de planejamento que decide sobre quais ações efetivamente serão realizadas para realização das tarefas ou objetivos almejados. Normalmente, esse tipo de arquitetura possui um módulo de planejamento subdividido em sub-módulos que obedecem a uma determinada hierarquia, sendo por isso, muitas vezes, são denominadas como Arquiteturas Hierárquicas (ARKIN, 1998). Esses sub-módulos são organizados em níveis de competência e originam uma estrutura que

contém planos mais globais em seus níveis mais altos até planos mais específicos nos níveis inferiores. Detalhes sobre outras arquiteturas deliberativas desenvolvidas podem ser verificados nos trabalhos Giralt, Chatila e Vaisset (1984), Nilsson (1984), Moravec e Cho (1989), Veloso e Carbonell (1993), Meystel (1993).

As arquiteturas deliberativas ou hierárquicas são adequadas para ambientes mais estruturados e estáticos, ou seja, com condições de contorno previsíveis para permitir a construção de modelos internos e módulos de planejamento sem grande complexidade (MATARIC, 1992; MURPHY, 2000). Como relatado em Arkin (1998), quanto menos estruturado e mais dinâmico é o ambiente de operação do robô, mais complicados se tornam o modelo do ambiente e o módulo de planejamento, tornando o projeto e a implementação da arquitetura robótica mais intricados. Assim, tais arquiteturas podem apresentar sub-sistemas eletrônicos e programas computacionais complexos, dependendo do modelo interno do ambiente exigido para a atuação do robô. Em robôs móveis que navegam em ambientes abertos (não estruturados), como os ambientes agrícolas, este problema é ainda maior, devido à quantidade de fatores a serem previstos e às características aleatórias de certos fatores, exigindo modelos deliberativos muitas vezes infactíveis devido ao elevado nível de complexidade (GARCIA-ALEGRE et al, 2001; HAGUE, SOUTHALL e TILLET 2002; HAGRAS, COLLEY e CALLAGHAN, 2003).

Seguindo outra linha, as pesquisas envolvendo arquiteturas robóticas reativas geraram um novo paradigma e trouxeram uma série possibilidades para solucionar alguns problemas do paradigma deliberativo. Como mencionado anteriormente (item 2.1), os sistemas robóticos puramente reativos não possuem um modelo interno de mundo e a percepção é relacionada diretamente a uma atuação, o que possibilitou o surgimento de sistemas de controle de robôs mais simples e modulares, além de

permitirem respostas mais rápida e uma habilidade maior para lidar com as mudanças do ambientes pouco estruturados (BROOKS, 1998; MATARIC, 1992; ARKIN, 1998; MURPHY, 2000).

Brooks (1986) caracteriza os sistemas robóticos baseados na abordagem reativa como sistemas horizontais por permitirem que cada estágio (camada ou módulo) possa controlar a realização de uma tarefa específica. Com a publicação Brooks (1986) firma-se uma nova linha para pesquisa em robótica associada ao paradigma reativo. Nesse trabalho é proposta uma arquitetura robótica puramente reativa denominada Subsunção (*Subsumption*). A Figura 2 mostra a decomposição do sistema de controle de um robô em estágios horizontais segundo a abordagem reativa.

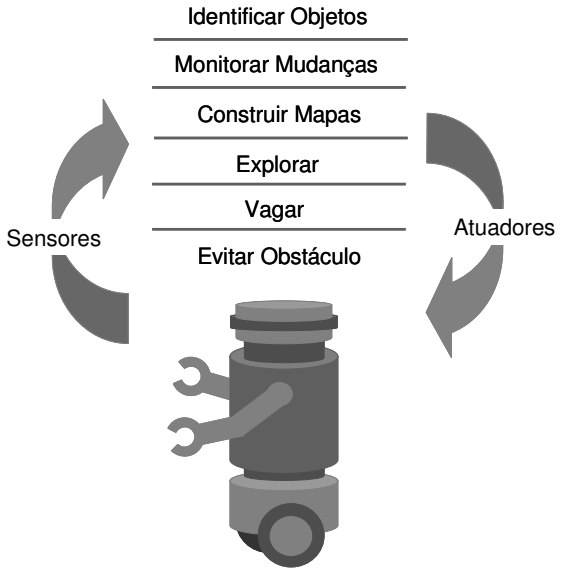


Figura 2 - Ilustração dos módulos componentes de uma arquitetura robótica segundo a abordagem reativa. Fonte: adaptado de Brooks (1986).

Segundo ilustrado na Figura 2, os diversos estágios ou camadas da arquitetura recebem informações dos sensores e geram recomendações diversas. Muitas vezes esses estágios ou camadas são tratados como comportamentos, sendo que a forma de tratar ou coordenar as recomendações geradas por eles é tema de diversas pesquisas como pode ser verificado em Murphy (2000) e Ribeiro, Reali-Costa e Romero (2001), dentre outros. A coordenação de comportamentos será descrita em mais detalhes posteriormente (item 3.3). Além de Brooks (1986), outras informações sobre arquiteturas reativas desenvolvidas podem ser obtidas em Rosenschein e Kaelbling (1986), Maes (1989) e Connell (1990).

Trabalhos de pesquisa posteriores a Brooks (1986) verificam que as arquiteturas puramente reativas constituem soluções viáveis para construção de sistemas robóticos para atuar em ambientes dinâmicos e que demandam respostas rápidas, porém tornam-se ineficientes quando existe a necessidade de planejamento mais complexo de tarefas ou missões. Mataric (1992) afirma que as arquiteturas reativas possuem pouca capacidade de assimilar novas informações tornando difícil melhorar o desempenho do sistema robótico em operação. Payton, Rosenblat e Kersey (1990) afirmam que sistemas robóticos puramente reativos sub-utilizam as informações do sistema sensorial pela falta de cooperação entre seus estágios, assim como, pela ausência de um módulo de planejamento deliberativo. Como constata Murphy (2000), nos sistemas robóticos puramente reativos há pouco compartilhamento de informações entre as camadas ou comportamentos, e, associado a isso, esses sistemas possuem menor capacidade para implementação de planejamento, já que não se pode contar com um modelo interno do ambiente e um módulo de planejamento como na abordagem deliberativa.

Grassi-Junior (2006) destaca e descreve algumas arquiteturas robóticas deliberativas e reativas desenvolvidas. Além disso, descreve as características gerais e sintetiza através de uma tabela comparativa as diferenças principais entre sistemas baseados nesses paradigmas. Também, destaca que as arquiteturas híbridas predominam nos dias atuais.

2.3. Arquiteturas híbridas

Grassi-Junior (2006) em sua revisão sobre arquiteturas robóticas observa que, de forma geral, as arquiteturas híbridas desenvolvidas buscam combinar a capacidade deliberativa de formulação de planos de ação para atingir objetivos específicos, com a capacidade reativa, normalmente estruturada com comportamentos, para execução de ações em tempo real em ambientes dinâmicos. A Figura 3 ilustra a estrutura geral de uma arquitetura híbrida.

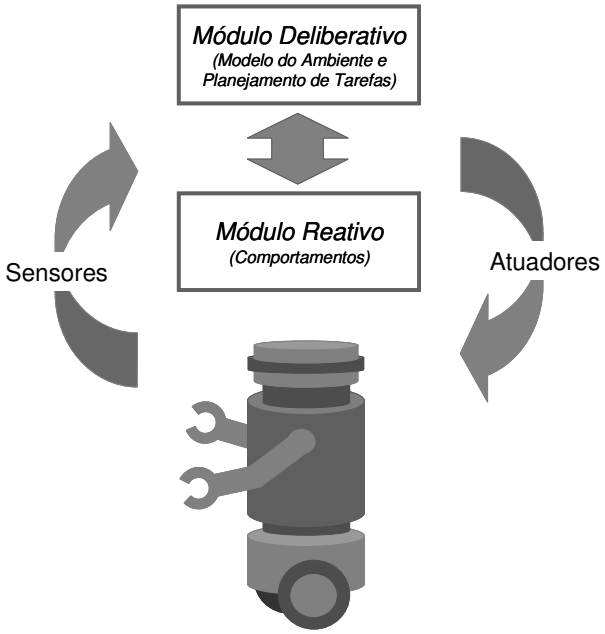


Figura 3 - Ilustração dos módulos componentes de uma arquitetura robótica híbrida.

A camada ou o módulo deliberativo, ilustrado na Figura 3, normalmente é responsável pelo planejamento da missão ou tarefa principal do robô, a partir de informações extraídas do ambiente e aplicadas a um modelo interno de mundo. Murphy (2000) relata que o módulo deliberativo normalmente é sub-dividido. Também, Murphy (2000) utiliza algumas características semelhantes encontradas em trabalhos de pesquisa para apresentar cinco sub-módulos referentes às sub-divisões para o módulo deliberativo: seqüenciador, gerenciador de recursos, cartógrafo, planejador de missões e monitor de problemas. Esses sub-módulos possuem as seguintes funções e características:

- *Seqüenciador*: define os comportamentos robóticos que serão utilizados para realização de uma seqüência de ações. Junto a isso, define uma seqüência de ativação e/ou os parâmetros de ativação para estabelecer a mais eficiente seqüência de ações;

- *Gerenciador de recursos*: tem a função de alocar entre os recursos disponíveis como, por exemplo, sensores ou atuadores, os mais apropriados para atender a uma tarefa ou para a realização de uma determinada seqüência de ações;

- *Cartógrafo*: relaciona-se com o modelo interno de mundo, sendo responsável pela criação, armazenamento, manutenção e acesso aos dados de um mapa com informações do ambiente sob o qual o robô atua;

- *Planejador de missões*: sub-módulo de mais alto nível na relação com o ser humano, constitui a interface com o ser humano para construção de planos de uma missão interpretados pelo robô através de uma linguagem específica;

- *Monitor de desempenho*: responsável pela avaliação do cumprimento de ações, além de indicar ou optar por ações alternativas para garantir a continuidade do plano pré-estabelecido.

O módulo reativo das arquiteturas híbridas baseiam-se em comportamentos, entretanto, nessa arquitetura esses comportamentos podem assumir formas mais complexas e menos reflexivas em relação aos comportamentos das arquiteturas puramente reativas (MURPHY, 2000). A Figura 4 apresenta um exemplo de estrutura de uma arquitetura híbrida baseada em comportamentos robóticos.

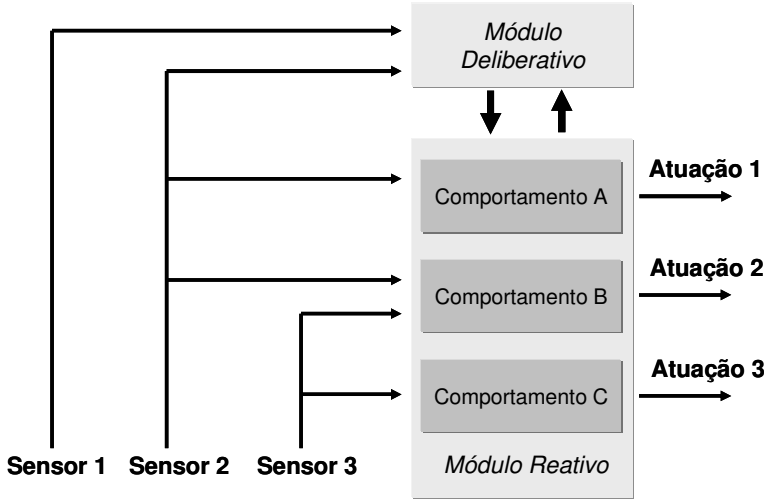


Figura 4 - Exemplo de uma arquitetura híbrida robótica baseada em comportamentos. Fonte: adaptado de Murphy (2000).

Como ilustrado na Figura 4, o módulo deliberativo interage como o módulo reativo. Essa interação permite que através da orientação do módulo deliberativo, os diversos comportamentos possam ser ativados e desativados para realização de uma seqüência de ações que atendam a uma determinada missão ou tarefa definida pelo módulo deliberativo. Junto a isso, os comportamentos ou as suas saídas podem

ser combinados para se obter uma recomendação de atuação mais completa, ou seja, uma recomendação baseada na integração de diferentes informações (ARKIN, 1998, MURPHY, 2000).

Os mecanismos de interação e de coordenação entre os módulos deliberativo e reativo, assim como entre os comportamentos, são elementos fundamentais na concepção de uma arquitetura híbrida (STOYTCHEV e ARKIN, 2001). Brooks (1986) utiliza um mecanismo hierárquico que estabelece prioridade para ativação dos comportamentos, sendo que a cada instante um comportamento está ativo. Em pesquisas posteriores, outros autores apresentam soluções que buscam aproveitar as informações geradas por cada comportamento de forma a se obter comandos de atuação baseados na combinação das recomendações de cada comportamento. São exemplos de propostas que seguem essa linha os seguintes métodos: Campo Potencial (*Potential-Field*; KHATIP, 1986), Vetor Sumário (*Vector Summation*; ARKIN,1989), Combinação Dependente de Contexto (*Context-Dependent Blending*; SAFFOTTI, KONOLIGE e RUSPINI, 1995) e Mecanismo de Voto (*Voting Mechanism*; ROSENBLATT, 1997). Pirjanian e Mataric (2000) apresentam uma sistematização das principais técnicas precursoras que embasam as pesquisas atuais nesse tema.

Junto à busca por mecanismos de interação e de coordenação em arquitetura híbridas, observa-se uma evolução, já notada por Arkin (1998) e Murphy (2000), em que algumas arquiteturas híbridas desenvolvidas utilizam comportamentos que possuem um grau maior de deliberatividade e que executam funções que normalmente eram de responsabilidade do módulo deliberativo. Além disso, em algumas arquiteturas certos comportamentos, denominados Comportamentos Abstratos, podem ser construído a partir da combinação de um conjunto de

comportamentos primitivos. Assim, devido a esse foco na estrutura dos comportamentos e em suas interações, muitos trabalhos de pesquisa utilizam a denominação Arquiteturas Baseadas em Comportamentos ou Arquiteturas Comportamentais (*Behavior-Based Architectures*) para se referir às arquiteturas desenvolvidas sob esse enfoque.

Mataric (1992) apresenta as arquiteturas baseadas em comportamentos em uma classificação independente das arquiteturas deliberativas, reativas ou hibridas. Arkin (1998) coloca as arquiteturas baseadas em comportamentos dentro do contexto das abordagens reativas ou hibridas, portanto, separadamente da abordagem deliberativa. Já Ribeiro, Reali-Costa e Romero (2001) dividem a abordagem baseada em comportamentos em três tipos: reativa, deliberativa e hibrida. Assim, entre esses e outros trabalhos não há um enquadramento e/ou uma definição comum para as arquiteturas robóticas baseadas em comportamentos, sendo estes dependem do enfoque de cada trabalho. Dessa forma, considera-se nesse trabalho que uma arquitetura baseada em comportamentos é uma arquitetura que contem, em pelo menos um de seus módulos, comportamentos que podem ser puramente reativos ou podem possuir certa deliberatividade. Além disso, tais comportamentos podem interagir entre si e/ou serem coordenados segundo algum método proposto para a arquitetura.

Entre as diversas arquiteturas híbridas propostas, destacam-se algumas que se tornaram clássicas por introduzirem, cada uma delas, novos conceitos que são explorados até hoje pelas modernas arquiteturas hibridas para robôs móveis. Dentre elas citas-se:

- *AuRa - Autonomous Robot Architecture* (ARKIN, 1998): considerada a primeira arquitetura hibrida proposta. A partir dos trabalhos seminais Arkin (1986) apud Arkin (1998) e Arkin (1987) apud Arkin (1998) foi sendo aprimorada em diversos trabalhos seguintes. Segundo Murphy (2000), esta arquitetura apresenta cinco subsistemas, sendo que dois destes estão postos na camada deliberativa, o Cartógrafo e o Planejador (possuem funções e características similares às descritas anteriormente neste mesmo item). Outros dois subsistemas estão postos na camada reativa, o Sensor e o Motor. Os subsistemas Sensor e Motor possuem coleções de Esquemas *(Schema)*, que são sub-elementos de um comportamento robótico que expressam percepção e controle motor. Os esquemas de percepção estão presentes no subsistema Sensor e os esquemas de controle motor estão presentes e o subsistema Motor. O subsistema Motor possui um Gerenciador de Esquemas que associa um esquema de percepção com um esquema de controle motor, tomados na coleção de esquemas, formando um comportamento. O quinto subsistema é denominado Controle Homeostático e atua entre a camada deliberativa e a camada reativa como um componente de ajuste dos comportamentos que são instanciados de acordo com avaliações baseadas em funções de custo aplicadas no transcorrer da missão.

- *ATLANTIS* (GAT, 1991; GAT, 1992): é uma arquitetura composta por três componentes: o Deliberador, o Seqüenciador e o Controlador. O Deliberador contém um modelo interno de mundo e um módulo de planejamento, sendo, portanto, um componente deliberativo. O Seqüenciador é a interface entre componente deliberativo e o componente reativo (Controlador) e possui um sistema operacional dedicado que controla o início e o término de atividades primitivas, que são processos geralmente reativos (reflexivos) que relacionam

percepção e reação. O Controlador contém um conjunto de processos que relacionam percepção e reação;

- *Saphira* (SAFFIOTTI, RUSPINI e KONOLIGE, 1993; SAFFIOTTI, KONOLIGE e RUSPINI, 1995; KONOLIGE et al. 1997): apresentada pela primeira vez em 1993, essa arquitetura evoluiu através de diversos trabalhos de pesquisa e permitiu o desenvolvimento de diversos robôs (Shaykey, Flakey e Erratic), incluindo robôs comerciais (MOBILEROBOTS, 2007). Possui dois módulos deliberativos básicos denominados LPS (*Local Perceptual Space* – Espaço Perceptual Local) e PRS-lite (*Procedural Reasoning System-lite* – Sistema de Raciocínio de Procedimento - lite). Segundo Murphy (2000), o primeiro módulo contém um mapa do ambiente e possui funções e características similares ao cartógrafo descrito anteriormente neste mesmo item (Item 2.3). O segundo módulo acumula funções e características dos sub-módulos gerenciador de recursos, planejador de missões e monitor de desempenho. A parte reativa contém comportamentos implementados através de Lógica Difusa ou *Fuzzy Logic* (ZADEH, 1965; ZADEH, 1973; YAGER e FILEV, 1994). Além disso, as regras difusas também são aplicadas para combinar as recomendações de cada comportamento. Descrições mais detalhadas das técnicas que utilizam lógica difusa serão apresentadas em itens posteriores, em especial nos itens 3 e 3.3;

- *DANM - Distributed Architecture for Mobile Navigation* (ROSENBLATT, 1997; WILLIAMS et al., 2001; ROSENBLATT, WILLIAMS e DURRANT-WHYTE, 2002): constituída por um conjunto de módulos independentes que colaboram para determinar as ações robóticas. Pode conter comportamentos reativos e/ou deliberativos que se comunicam com um ou mais módulos de arbitragem, os

Árbitro*s*, enviando votos em favor de determinadas ações. Um Árbitro é implementado através de regras difusas (detalhado no Capítulo 3) e é responsável por combinar os votos e gerar ações enviadas aos sistemas de atuação do robô. Um módulo denominado Controlador de Tarefas é responsável pela ativação ou desativação de comportamentos de acordo com a relevância de cada um deles durante a execução de um plano definido por um usuário;

Outros exemplos de arquiteturas podem ser encontrados em Arkin (1998), Murphy (2000), Yavuz e Bradshaw (2002) e Grassi-Junior (2006). De forma mais sistematizada, Yavuz e Bradshaw (2002) e Grassi-Junior (2006) destacam e descrevem algumas arquiteturas robóticas deliberativas, reativas e híbridas desenvolvidas, apontando as suas características gerais e sintetizando, através em tabelas comparativas, as diferenças principais entre os sistemas baseados nessas arquiteturas.

Observam-se entre os diversos trabalhos relatados (SAFFIOTTI, RUSPINI e KONOLIGE, 1993; ARKIN, 1998; KONOLIGE et al. 1997, MURPHY, 2000; STOYTCHEV e ARKIN, 2001; YAVUZ e BRADSHAW, 2002; ROSENBLATT, WILLIAMS e DURRANT-WHYTE, 2002), propostas de arquiteturas híbridas promissoras que permitem operação de robôs móveis em ambientes pouco estruturados e dinâmicos. Em particular, verificam-se entre esses trabalhos, diversos métodos desenvolvidos e aplicados para implementar arquiteturas híbridas baseadas em comportamentos que utilizam lógica difusa. A aplicação desses métodos origina um campo de estudo denominado Controle Baseado em Comportamentos Difusos (*Fuzzy Behavior-Based Control*) ou Controle Difuso (*Fuzzy Control*), que permitem tanto para composição de comportamentos robóticos, assim como a composição de módulos de gerenciamento para as arquiteturas, como, por

exemplo, módulos de coordenação de comportamentos. São exemplos de aplicação desses métodos as arquiteturas Saphira (composição e coordenação de comportamentos) e DANM (coordenação de comportamentos) descritas anteriormente nesse mesmo item.

Os primeiros trabalhos que buscavam utilizar algoritmos baseados em lógica difusa foram propostos para aplicações reativas, como nos trabalhos seminais Sugeno e Nishida (1985) e Takeuchi, Nagai e Enomoto (1988), para guiagem autônoma de um carro e para controle de um robô móvel respectivamente. Posteriormente, outros trabalhos trouxeram contribuições significativas que colaboraram para formar os fundamentos da aplicação de técnicas baseadas em comportamentos difusos em arquiteturas para robôs móveis. Uma descrição detalhada desses desenvolvimentos será apresentada no item seguinte (Item 3.3).

3. ARQUITETURAS ROBÓTICAS BASEADAS EM COMPORTAMENTOS DIFUSOS

As bases da teoria da Lógica Difusa (*Fuzzy Logic*), também nomeada como Lógica Nebulosa, estão presentes nos trabalhos de Lofti A. Zadeh (ZADEH, 1965; ZADEH, 1973). Sob o ponto de vista da engenharia de sistemas de controle, incluindo os sistemas robóticos, a teoria de lógica difusa provê uma metodologia formal para transferir experiências humanas para os sistemas de controle. Nessa metodologia, variáveis lingüísticas são utilizadas para compor de forma simples e intuitiva estruturas de decisão (conjunto de regras). Em arquiteturas robóticas, essas regras heurísticas associam percepções a ações ou a recomendações gerando, dessa forma, comportamentos ou elementos de deliberações para compor as arquiteturas robóticas de controle. Tais sistemas baseados em lógica difusa, também denominados Sistemas de Inferência Difusa, permitem tanto a geração como a coordenação de comportamento para a realização de tarefas autônomas e inteligentes (SERAJI & HOWARD, 2002).

Os sistemas de inferência difusa são baseados em três processos: Fuzzificação[1] (*Fuzzification*), Inferência Difusa (*Fuzzy Inference*) e Defuzzificação[2] (Defuzzification). Fundamentos sobre a teoria sobre lógica difusa são apresentados

---

[1] Esta palavra também é interpretada na literatura científica em Língua Portuguesa como Difusificação

[2] Esta palavra também é interpretada na literatura científica em Língua Portuguesa como Dedifusificação

no item a seguir (item 3.1) baseando-se nos trabalhos Zadeh (1965), Zadeh (1973), Zimmermann (1991), Yager e Filev (1994) e Ross (1995).

3.1. Fundamentos da lógica difusa

A Teoria de Conjuntos Difusos apresentada por Zadeh (1965), tem como objetivo fornecer um ferramental matemático para o tratamento de informações de caráter impreciso ou vago. Em contraposição à lógica booleana (lógica tradicional) que trata o mundo real como tendo apenas duas classes, verdadeiro ou falso, a lógica difusa atribui às variáveis reais, classes de conjuntos associados a termos lingüísticos, como por exemplo, *rápido, alto, distante, quente, lento, baixo, próximo, frio*, etc. Assim, um elemento pode pertencer com certo grau a um dado conjunto, originando-se o que se costuma chamar Grau de Pertinência (*Grade of Membership*).

3.1.1 Função de pertinência e fuzzificação

O conceito de grau de pertinência é a base da teoria de conjuntos difusos, na qual a lógica difusa é fundamentada. A função que define o grau de pertinência do elemento a um dado conjunto difuso é denominada Função de Pertinência ou Função de Associação *(Membership)*. Cada elemento pode pertencer a mais de um conjunto com um determinado grau de pertinência em cada um deles. Uma função de pertinência pode ser representada por:

$$\mu_A(x) : U \rightarrow [0,1] \tag{1}$$

Pela representação anterior em (1) entende-se que um conjunto difuso *A* definido no universo de discurso *U* é caracterizado por uma função de pertinência μ_A, a qual mapeia os elementos *x* de *U* para o intervalo [0 , 1].

O processo de mapear um valor real (exato) em um conjunto difuso é denominado Fuzzificação. Nesse processo as funções de pertinência são elementos fundamentais. Essas funções podem ser definidas por intermédio de funções analíticas (contínuas) como, por exemplo, Gaussianas, Singletons e Sigmóides ou podem ser definidas pela composição de segmentos contínuos lineares (descontínuas ou discretas) resultando em formas triangulares e/ou trapezoidais como ilustrado na Figura 5.

3.1.2 Variáveis lingüísticas

Um conceito importante relacionado com conjuntos difusos é o de Variável Lingüística. Sendo uma variável um identificador que pode assumir um entre vários valores, uma variável lingüística pode assumir um valor lingüístico entre vários outros em um conjunto de termos lingüísticos. Os termos lingüísticos descritos por intermédio de conjuntos difusos e representados por funções de pertinência são denominados Valores da Variável Lingüística ou Valores Lingüísticos. Na teoria do conjuntos difusos as variáveis reais (exatas) pertencentes ao conjunto clássicos (ordinários) são denominadas Variáveis Crisp. A Figura 5 mostra um exemplo que

envolve os conceitos e denominações sobre função pertinência, variável lingüística e variável crisp.

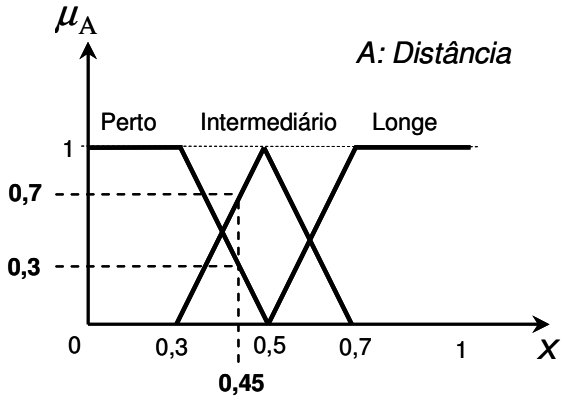


Figura 5 - Exemplo de funções de pertinência para variável lingüística *Distância*.

No exemplo da Figura 5, a variável lingüística *Distância* (conjunto difuso *A*) é composta pelos conjuntos difusos *Perto, Intermediário* e *Longe* (valores lingüísticos) que são descritos por funções de pertinência que associam valores crisp aos valores lingüísticos. Assim, por exemplo, uma leitura crisp igual a 0,45, realizada em um determinado instante por um sensor que mede a distância (leituras podem estar entre 0 e 1 no caso ilustrado), pertence ao conjunto difuso *Distância* com pertinência igual a 0,7 para o valor *Intermediário* da função de pertinência $\mu_{Intermediário}$ e 0,3 para o valor *Perto* da função de pertinência μ_{Perto}.

Como visto, a variável lingüística é uma variável cujos valores são nomes de conjuntos difusos, sendo que essa pode ser formada por um ou mais termos lingüísticos com os respectivos graus de pertinência. Formalmente, uma variável lingüística é caracterizada pela quíntupla {X, T(X), U, G, M}, onde (ZIMMERMANN, 1991):

- X é o nome da variável lingüística (nome do conjunto);
- T(X) é um conjunto de termos lingüísticos (atributos, adjetivos);
- U é o universo de discurso (faixa, intervalo);
- G é uma regra sintática para composição dos termos lingüísticos;
- M é uma regra semântica que associa o cada termo lingüístico de G a um conjunto difuso.

Aplicando-se essa definição ao exemplo da Figura 5 tem-se:

- X: Distância;
- T(Distância): {Perto, Intermediário, Longe};
- U: [0,1];
- G: Perto [X: 0,0 a 0,5], Intermediário [X: 0,3 a 0,7] e Longe [X: 0,7 a 0,1];
- M: associação dos termos de G aos conjuntos difusos definidos por μ_{Perto} , $\mu_{Intermediário}$ e μ_{Longe}.

Variáveis lingüísticas podem também conter modificadores (também lingüísticos) que alteram seu valor. Exemplos de modificadores são: *muito*, *pouco*, *não muito*, *mais*, *menos*, l*evemente* e *extremamente*. Assim, podem-se criar novos termos lingüísticos com as respectivas funções de pertinência a partir da combinação entre termos primários (por exemplo: *perto*, *intermediário* e *longe*) e os modificadores, resultado em novos termos para compor a variável lingüística como, por exemplo, *muito perto*, *pouco intermediário* e *extremamente longe*.

3.1.3 Operadores difusos

A teoria de conjuntos difusos apresenta uma série de definições e operações que podem permitir a criação de novos e complexos conjuntos difusos pela combinação de conjuntos mais simples (ZADEH, 1965; YAGER e FILEV, 1994). Entre, as diversas operações definidas, destacam-se três: Complemento, União e Intersecção.

A função de pertinência relativa ao complemento *A'* de um conjunto difuso *A* é definido pela relação:

$$\mu_{A'}(x) = 1 - \mu_A(x) \qquad \forall x \in X \qquad ...(2)$$

O complemento muitas vezes é associado ao operador *not* (não) simbolizado por $\neg$. Dessa forma, o complemento de um conjunto difuso *A* pode ser simbolizado por $\neg A$.

A união de dois conjuntos difusos *A* e *B* em um universo X, denotada por $A \cup B$, contém todos os elementos pertencentes a *A* ou a *B*. A intersecção dos mesmos conjuntos, denotada por $A \cap B$, contém todos os elementos pertencentes a *A* e a *B*. Os operadores booleanos baseadas nos conectivos lógicos *and* (e) e *or* (ou) são equivalentes as operações de união e intersecção de conjuntos difusos, entretanto, na teoria de conjuntos difusos para essas operações são definidos os operadores *maximum* (max ou $\vee$) e *minimum* (min ou $\wedge$) respectivamente. Assim, a função de pertinência relativa a união de dois conjuntos difusos, *A* e *B*, é definida pela relação:

$$\mu_{A\cup B}(x) = \max\left[\mu_A(x), \mu_B(x)\right] \qquad \forall x \in X \tag{3}$$

A função de pertinência relativa a intersecção dos dois conjuntos *A* e *B* é definida pela relação:

$$\mu_{A\cap B}(x) = \min\left[\mu_A(x), \mu_B(x)\right] \qquad \forall x \in X \tag{4}$$

Ambas as funções de pertinência relativas a união e a intersecção também podem ser representadas respectivamente por:

$$\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x) \qquad \forall x \in X \tag{5}$$

$$\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x) \qquad \forall x \in X \tag{6}$$

Os operados são utilizados em processo denominado Composição ou Agregação para obter-se um grau de consenso entre as informações disponíveis, calculando-se um valor difuso final. Esse processo permite que vários conjuntos difusos (n ≥ 2) sejam combinados de uma forma desejável, para produzirem um único conjunto. As propriedades definidas para os conjuntos difusos (ZADEH, 1965; ZADEH, 1973), similares às propriedades algébricas de conjuntos ordinários como, por exemplo, comutatividade, associatividade, distributividade e absorção, permitem que os operadores de união e intersecção forneçam um mecanismo para extensão de suas definições para qualquer número de argumentos (conjuntos).

3.1.4 Relação difusa

Os operadores de união e intersecção aplicam-se também a conjuntos difusos pertencentes a universos de discurso diferentes. Para tanto, é importante a definição de Relação Difusa (*Fuzzy Relation*) da teoria de conjuntos difusos (ZADEH, 1965). Uma relação difusa representa o grau da associação entre dois ou mais conjuntos difusos pertencentes a universos de discursos diferentes. Formalmente, dados dois universos *U* e *W* representados por:

$$\mu_A(x) : U \rightarrow [0,1] \qquad x \in U \tag{7}$$

$$\mu_B(y) : W \rightarrow [0,1] \qquad y \in W \tag{8}$$

A relação difusa *R* é um conjunto difuso em *U x W*, caracterizada pela função de pertinência:

$$\mu_R(x, y) \in [0,1] \quad , \quad x \in U \quad e \quad y \in W \tag{9}$$

A teoria de conjuntos difusos define formas de aplicação de operadores, como os operadores de união e intersecção, utilizados no processo de composição entre relações difusas, que podem permitir a criação de novos e complexos conjuntos difusos pela combinação de conjuntos mais simples pertencentes a universos de discurso diferentes (ZADEH, 1965; YAGER e FILEV, 1994).

3.1.5 Base de regas

Além dos operadores descritos anteriormente neste item (item 3.1), o operador *se ... então* (*if ... than*), também conhecido como Declaração Condicional Difusa, estabelece a dependência do valor de uma variável lingüística em relação ao valor de outra. Em muitas aplicações da lógica difusa essas declarações condicionais são simplesmente denominadas Regras Lingüísticas (*Fuzzy Rules*). As regras lingüísticas são constituídas por frases do tipo *se x é A então y é B* (*if x is A than y is B*) denominada Implicação, que pode ser expressa pela relação:

$$\mu_{A\to B}(x,y) = f_{\to}\left(\mu_A(x), \mu_B(y)\right) \tag{7}$$

Onde: $f_{\to}$ é um operador de implicação.

Assim, a função de pertinência $\mu_{A\to B}(x,y)$ mede o grau de verdade da relação entre *x* e *y* referente a implicação *se x é A então y é B* (*if is A than y is B*). A variável "*x é A"* é dita Antecedente e a variável "*y é B"* é dita Conseqüente. Uma implicação pode conter um ou mais antecedentes associados pelo processo de composição formando uma Expressão Lingüística.

Normalmente, os sistemas de inferência difusa possuem conjuntos de regras lingüísticas criadas por seus projetistas, denominado Base de Regras. A base de regras nesses sistemas expressa o conhecimento empírico e heurístico humano na sua própria linguagem de comunicação. Em projetos de sistemas de inferência difusa, a idéia básica consiste-se em incorporar a experiência ou conhecimento de um processo no projeto de um controlador difuso através da formulação de

implicações que irão constituir a base de regras do sistema. Dessa maneira, podem ser construídas regras como, por exemplo:

- **se** *Distância é Muito Perto* **e** *Velocidade é Grande* **então** *Propulsão é Pare*
- **se** *Distância é Muito Longe* **e** *Velocidade é Grande* **então** *Propulsão é Frente*
- **se** *Distância é Extremamente Perto* **e** *Velocidade é Muito Pequena* **então** *Propulsão é Trás*

Através do conjunto de regras lingüísticas, um algoritmo de controle pode ser construído, com seus termos definidos como conjuntos difusos. Dessa forma, a lógica difusa permite criar controladores difusos que simulam o pensamento humano, incorporando sua inerente falta de precisão na observação de um sistema físico (YAGER e FILEV, 1994).

3.1.6 Inferência difusa

A associação entre as variáveis lingüísticas e a base de regras de um sistema difuso permite deduzir novos fatos a partir de informações de entrada, sendo que esse processo é denominado Inferência Difusa. Quando uma base de regras é avaliada para um conjunto de valores de variáveis de entrada, encontram-se, em geral, mais de uma regra aplicável. Neste caso, as conseqüências obtidas pela inferência das implicações devem ser combinadas ou agregadas para produzir uma resposta única do sistema para cada variável de saída.

Na literatura pesquisada são definidos diversos métodos de inferência que possuem a mesma finalidade de se obter um novo conjunto difuso (função de pertinência) que expressa uma informação deduzida (inferida) a partir das informações de entrada. Um método bastante comum é *max-min* que é baseado nas operações de união e intersecção. A Figura 6 ilustra tal método aplicado a duas regras (k) diferentes.

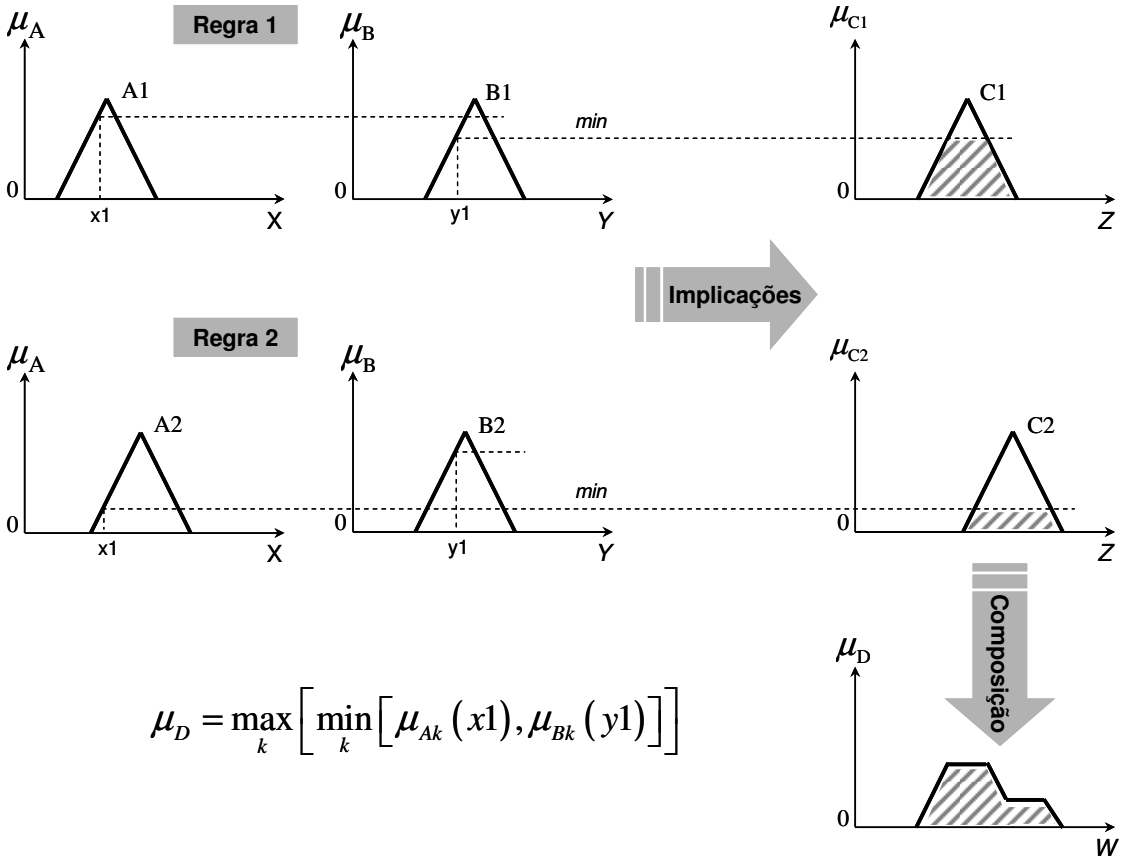


Figura 6 - Método de inferência difusa max-min.

O exemplo da Figura 6 ilustra duas variáveis lingüísticas, *A* e *B*, que possuem conjuntos difusos *A1*, *A2*, pertencentes a variável A, B1 e *B2*, pertencentes a variável B, com suas respectivas funções de pertinência, $\mu_A(x)$ (A1 e A2) e

$\mu_B(y)$ (B1 e B2). Através da avaliação pelo método <u>*min*</u> (intersecção) da implicação relativa a *regra 1*, que relaciona os conjuntos *A1* e *B1* diante das entradas *x1* e *y1* apresentadas, obtém-se como conseqüência um sub-conjunto difuso do conjunto difuso *C1* pertencente a variável de lingüística C. Sob o mesmo procedimento, a avaliação pelo método <u>*min*</u> da implicação relativa a *regra 2*, que relaciona os conjuntos *A2* e *B2* diante das mesmas entradas *x1* e *y1* apresentadas, obtém-se como conseqüência um sub-conjunto difuso do conjunto difuso *C2* pertencente a variável de lingüística *C*. Posteriormente, a composição pelo método *max* (união) dos sub-conjuntos de *C1* e *C2* gera um conjunto difuso de saída *D* com função de pertinência $\mu_D(w)$.

3.1.7 Defuzzificação

Em sistemas de inferência difusa, uma vez obtido o conjunto difuso de saída através do processo de inferência, como o ilustrado na Figura 6 (μ_D), é necessário efetuar uma interpretação dessa informação, representada por uma quantidade exata, através de um processo de Defuzzificação. Isto se faz necessário, pois em aplicações práticas geralmente são requeridas saídas exatas. Na literatura pesquisada existem diversos métodos de defuzzificação, sendo que os dois mais empregados são: o Centro de Gravidade e a Média dos Máximos. O método do centro de gravidade, também denominado centróide, é representado por:

$$w^* = \frac{\int \mu_D . w \quad dw}{\int \mu_D \quad dw} \tag{8}$$

A equação (8) aplicada ao exemplo da Figura 6 permite determinar o valor exato (nítido), ω^*, através do centróide da área referente á função de pertinência μ_D. Em um controlador difuso, esse valor exato constitui um comando de atuação ou uma informação para ativar um comando de atuação.

3.2. Controladores e comportamentos difusos

A tendência de aplicação da lógica difusa aos propósitos da robótica móvel tem sido justificada pela sua eficácia em lidar com diferentes graus de incerteza e de traduzir a relação robô-ambiente para estruturas com expressões lingüísticas mais intuitivas para o ser humano. Além disso, a lógica difusa também permite uma suavização na transição entre os comportamentos robóticos em seqüência (SAFFIOTTI, 1997; AGUIRRE; GONZÁLEZ, 2000; YANG, MOALLEM e PATEL, 2005).

Segundo sistematiza Arkin (1998), um sistema de controle fundamentado em lógica difusa é composto por quatro elementos: o fuzzificador (*fuzzifier*), a base de regra difusa (*fuzzy rule base*), a máquina de inferência difusa (*fuzzy inference engine*) e o defuzzificador (*defuzzifier*).

A Figura 7 ilustra os módulos de um sistema de controle baseada em lógica difusa. Como pode ser notado pela ilustração, os valores das variáveis lidas pelos sensores do robô (denomindados não difusos ou crisp) são associados à valores difusos das variáveis lingüísticas pelo módulo Fuzzificador. Esses valores difusos alimentam o módulo denominado Máquina de Inferência, que avalia esses valores através de implicações definidas pela base de regras, deduzindo assim, novos fatos ou novos valores a partir de informações de entrada. Esses novos fatos ou valores difusos constituem recomendações que são associadas a valores reais através de um método pré-definido de defuzzificação pelo módulo denominado Defuzzificador.

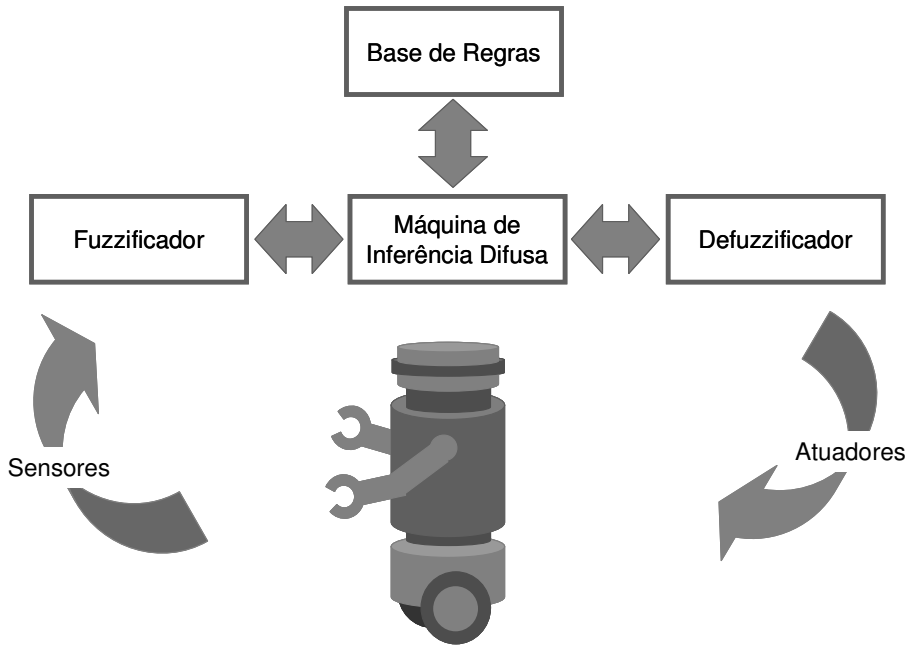


Figura 7 - Módulos de um sistema de controle difuso. Fonte: adaptado de Arkin (1998).

Após os trabalhos seminais de Sugeno e Nishida (1985) para guiagem autônoma de um carro e de Takeuchi, Nagai e Enomoto (1988) para controle de um

robô móvel, diferentes métodos baseados em lógica difusa foram propostos para aplicações em arquiteturas híbridas.

Os autores Ruspini (1990), Saffiotti, Ruspini e Konolige (1993) e Saffiotti, Konolige e Ruspini (1995), como descritos anteriormente (item 2.3), apresentam as bases de uma arquitetura baseada em comportamentos difusos denominada Saphira que evoluiu através de diversos trabalhos de pesquisa posteriores (KONOLIGE et al., 1997; SAFFIOTTI, RUSPINI e KONOLIGE, 1999; REMONDINI e SAFFIOTTI, 2006) e permitiu o desenvolvimento de diversos robôs para pesquisa (Shaykey, Flakey e Erratic) e robôs comerciais (MOBILEROBOTS, 2007). Nos trabalhos desses autores, além de proposição de uma forma para a composição de comportamentos difusos, descreve-se um método para coordenação dos comportamentos baseado em lógica difusa. Como descrito por Remondini e Saffiotti (2006), por esse método um comportamento difuso é composto por dois tipos de conjuntos de regras difusas: *ground rules* (regras basais) e *meta rules* (meta-regras). O primeiro conjunto relaciona informações dos sensores (variáveis lingüísticas relacionadas com informações de percepção) com comandos para controle (variáveis lingüísticas relacionadas com informações de atuação). O segundo conjunto relaciona contextos de aplicação dos comportamentos (variáveis lingüísticas relacionadas com informações de percepção) com a ativação de comportamentos. Um processo de inferência e defuzzificação, denominado pelos autores *Context-Dependent Blending* (CDB; Combinação Dependente do Contexto), é utilizado para combinar as informações geradas pelos diferentes conjuntos. A Figura 8 ilustra a estrutura do controlador baseado em comportamentos difusos do método descrito.

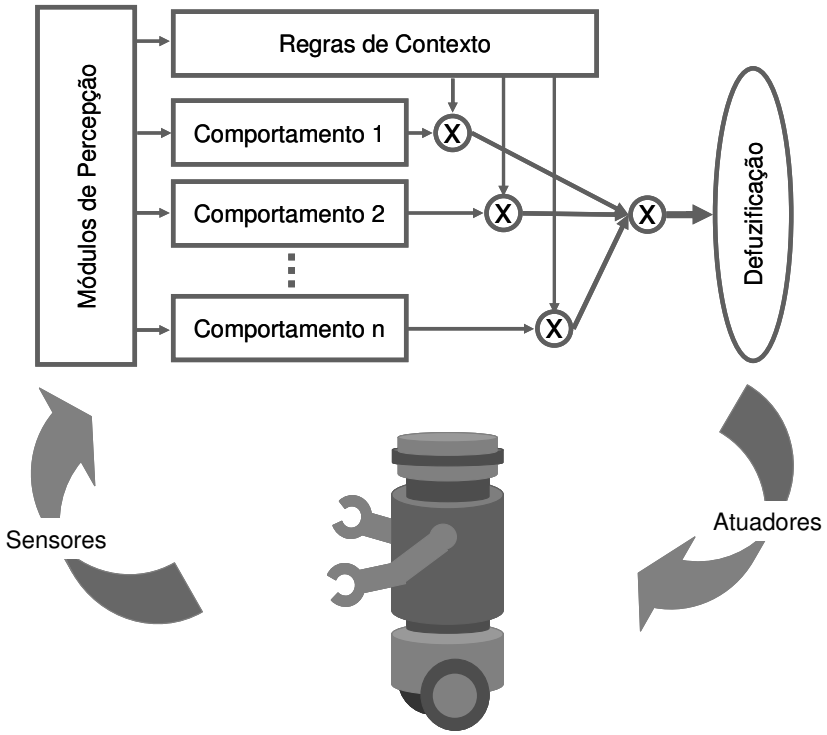


Figura 8 - Controlador difuso baseado o método *Context-Dependent Blending*. Fonte: adaptado de Saffiotti, Ruspini e Konolige (1999)

Como podem ser verificadas na Figura 8, as informações geradas por cada comportamento e as informações geradas pelo conjunto de regras de contexto são combinadas no processo de inferência difusa através de operados como, por exemplo, união, intersecção e complemento, gerando conjuntos difusos de saída. Por fim, os valores de comando para os atuadores de robô são obtidos através da defuzzificação aplicada a esses conjuntos difusos de saída.

O método aplicado na arquitetura Saphira define a ativação dos comportamentos adequados para realização de tarefas específicas a cada momento, assim como, uma forma de combinar as recomendações geradas pelos comportamentos ativos, originando uma estrutura descentralizada, com flexibilidade para a sua expansão e com integração entre seus comportamentos. Entretanto, Goodridge e Luo (1994), Goodridge, Kay e Luo (1996) e Goodridge, Kay e Luo

(1997) mostram que em sistemas com muitas entradas o número de regras difusas para os conjuntos difusos torna-se elevado, com conseqüente aumento da complexidade para construção dos controladores difusos e seus respectivos comportamentos. Diante disso, os mesmos autores propõem um método para composição de comportamento robótico difuso utilizando um conjunto de controladores difusos ou comportamentos difusos mais simples.

O método introduzido por Goodridge e Luo (1994) baseia-se na aplicação de duas técnicas: a divisão de um comportamento difuso em mais de um conjunto de regras e a aplicação da defuzzificação utilizando o método do centróide baseado em funções de pertinência singleton para os conjuntos difusos de saída. Os autores descrevem os experimentos realizados com o robô MARGE com uma arquitetura robótica composta por comportamentos individuais, como, por exemplo, *goal-seeking* (*perseguir meta*), *obstacle avoidance* (*evitar obstáculo*), *barrier following* (*identificar barreira*) e *object docking* (*transportar objeto*), sendo que cada comportamento é composto por um conjunto de controladores difusos. Essa abordagem conduz a uma simplificação na elaboração dos comportamentos difusos, pois em vez de se ter um comportamento complexo baseado em um grande número de entradas, pode-se construir comportamentos mais simples, com menor número de entradas e menor número de regras difusas, e combinar suas saídas para formação do comportamento complexo. A recomendação de cada comportamento é combinada com um peso fornecido por um controlador difuso associado ao comportamento, assim é realizada a coordenação dos comportamentos pelo método.

Técnicas semelhantes à proposta em Goodridge e Luo (1994) de dividir um comportamento difuso em mais de um conjunto de regras são exploradas em

trabalhos posteriores. Nesse sentido, a definição de métodos de coordenação de comportamentos difusos mais simples, ou primitivos, para composição de comportamentos complexos, constitui o cerne para o desenvolvimento de arquiteturas baseadas em comportamentos robóticos difusos. Alguns trabalhos recentes denominam tais métodos como abordagem baseada em Comportamentos Difusos Hierárquicos ou, também, abordagem baseada em Comportamentos Difusos Multiníveis (TUNSTEL, LIPPINCOTT e JAMSHIDI, 1997; TUNSTEL, OLIVEIRA e BERMAN, 2002; TAO LI e CHUN LI, 2005; REMONDINI e SAFFIOTTI, 2006).

Junto às propostas promissoras apresentadas em Sugeno e Nishida (1985), Takeuchi, Nagai e Enomoto (1988), Goodridge e Luo (1994) e Saffiotti, Ruspini e Konolige (1993), outros trabalhos desenvolvidos e publicados na ultima década do milênio passado trouxeram contribuições significativas que colaboraram para formar os fundamentos da aplicação de técnicas baseadas em comportamentos difusos em arquiteturas para robôs móveis. Entre essas publicações destacam-se: Yen e Pfluger (1991), Gasós, García-Alegre e García (1991), Pin e Watanabe (1994), Yen e Pfluger (1995), Bonarini (1997), Lin e Wang (1997) e Tunstel, Lippincott e Jamshidi (1997). Na área agrícola desta-se a pesquisa seminal apresentada por Garcia-Alegre, Bustos e Guinea (1995) para desenvolvimento de um robô agrícola móvel.

Um robô móvel pode ser utilizado em tarefas de exploração com coleta de dados e atuação em diversos ambientes, entretanto a ação fundamental para realização dessas tarefas é a locomoção pelo ambiente de operação. Muitas vezes, um sistema automático pode viabilizar as tarefas citadas, não demandando tomada de decisão ou adaptação. Porém, a tarefa de locomoção, principalmente em ambientes pouco estruturados e dinâmicos, muitas vezes demanda inteligência computacional para tomada de decisão ou adaptação diante das incertezas e

imprecisões características desses tipos de ambientes. Recentemente, um número expressivo de grupos de pesquisa busca soluções baseadas em controladores difusos para navegação em ambientes pouco estruturados, dinâmicos e até mesmo desconhecidos. Entre as propostas mais notáveis podem ser destacadas as apresentadas em Hagras, Callaghan e Colley (2001), Rosenblatt, Williams e Durrant-Whyte (2002) e Seraji e Howard (2002), cujos resultados são referências para trabalhos atuais (BLACKMORE et. al., 2004; HAGRAS, CALLAGHAN e COLLEY, 2004; IP, RAD e WONG, 2004; YANG, MOALLEM e PATEL, 2005; PEDERSEN et al., 2005; MAALOUFA, SAADA e SALIAHB, 2006; REMONDINI e SAFFIOTTI, 2006). Uma descrição detalhada desses desenvolvimentos é apresentada no item a seguir (Item 3.3).

3.3. Controladores difusos para navegação em ambientes pouco estruturados e dinâmicos

Rosenblatt e Payton (1989) e Payton, Rosenblatt e Keirsey (1990) apresentam métodos para integração de informações de comportamento para aprimoramento abordagem de Brooks (1986), com o argumento de que nessa abordagem as informações do sistema sensorial do robô são sub-utilizadas. Como uma evolução dessas pesquisas, é apresentada a arquitetura baseada em comportamentos denominada DANM (descrita no item 2.3), que foi desenvolvida para aplicação em sistemas robóticos de veículos terrestres autônomos (ROSENBLATT, 1997). Posteriormente, essa arquitetura é aplicada em sistemas robóticos de veículos submarinos autônomos *(AUV – Autonomous Underwater*

Vehicles) (WILLIAMS et al., 2001; ROSENBLATT, WILLIAMS e DURRANT-WHYTE, 2002).

A arquitetura DANM não utiliza prioritariamente comportamentos difusos, mas permite que diferentes comportamentos mais ou menos complexos, construídos a partir da aplicação de diferentes métodos, possam compartilhar o controle robótico. A lógica difusa é aplicada nessa arquitetura para construção de módulos de arbitragem, os Árbitros, que realizam a coordenação dos comportamentos existentes através da avaliação de votos gerados pelos próprios comportamentos em prol ou contra a realização de uma determinada ação. Esse tipo de abordagem foi adotada e apresentada pelos pesquisadores em Williams et al. (2001) e Rosenblatt, Williams e Durrant-Whyte (2002), sendo que, anteriormente, os Árbitros utilizavam um método de avaliação dos votos por interpolação (ROSENBLATT, 1997). A Figura 9 mostra esquematicamente a arquitetura DAMN aplicada ao sistema de navegação de um veículo submarino autônomo.

Como ilustrado na Figura 9, o módulo controlador de tarefas ativa ou desativa os comportamentos mais adequados para realização das tarefas em cada momento ao longo de uma missão. Esse planejamento é realizado anteriormente a cada missão, sendo que um operador humano define as tarefas e os comportamentos a serem ativados, assim como conjunto de condições que são avaliadas ao longo da missão.

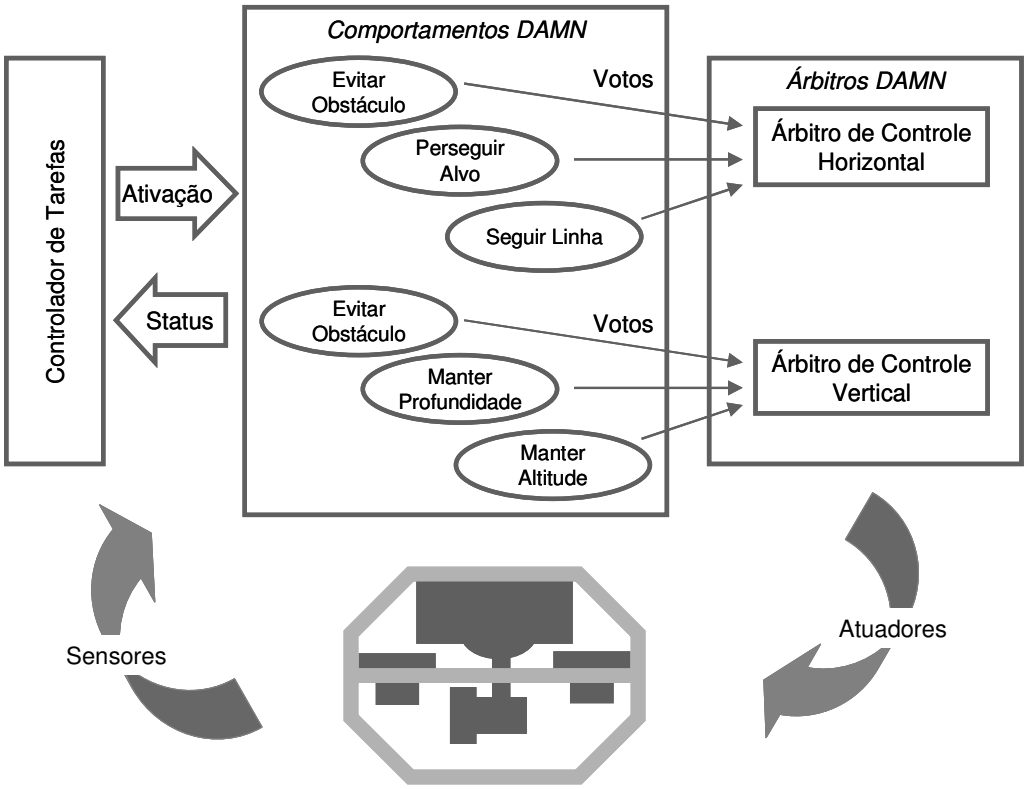


Figura 9 - Arquitetura robótica DAMN aplicada a um veículo submarino autônomo. Fonte: adaptado de Williams et al. (2001) e de Rosenblatt, Williams e Durrant-Whyte (2002)

Na arquitetura DAMN, os comportamentos são independentes e atuam paralelamente gerando votos quando ativados. Cada comportamento tem uma entrada sensorial independente, sendo que um sensor pode alimentar um ou mais comportamentos ao mesmo tempo. Junto a isso, como pode ser visto na Figura 9, cada árbitro é responsável por receber e avaliar os votos de um grupo de comportamentos e gerar um comando para um determinado sistema de atuação. Os árbitros são implementados através de um controlador difuso que utiliza a inferência difusa para gerar a partir dos votos recebidos, conjuntos difusos submetidos ao processo de defuzzificação pelo método dos máximos (ROSENBLATT, WILLIAMS e DURRANT-WHYTE, 2002). Esse mecanismo que atua sobre a saída dos comportamentos buscando coordenar as suas recomendações é classificado pelos autores como Fusão de Comandos, em oposição aos mecanismos de combinação

de informações de sensores identificados na literatura como Fusão de Sensores (ROSENBLATT, 1995; CHUNG, OJEDA e BORENSTEIN, 2001; BORENSTEIN e FENG,1996).

Em Williams et al. (2001) e Rosenblatt, Williams e Durrant-Whyte (2002) os experimentos são realizados com o submersível autônomo denominado Oberon que contém como elementos de percepção dois sonares, uma câmera com sensor CCD (*Charged Coupled Device*) colorida, um sensor de profundidade batimétrico, um giroscópio de fibra óptica e uma bússola eletrônica. Para atender a navegação com três graus de liberdades foram utilizados dois árbitros como pode ser observado na Figura 9. O árbitro de controle horizontal (velocidade e direção no plano horizontal) utiliza os comportamentos *evitar obstáculos* (*avoid obstacle), perseguir alvos* (*follow targets*) e *seguir traçado* (*follow rope*) e o árbitro de controle vertical (profundidade) utiliza os comportamentos *evitar obstáculos* (*avoid obstacles*), *manter altitude* (*maintain altitude*) e *manter profundidade* (*maintain depth*).

Os experimentos realizados com o submersível Oberon (ROSENBLATT, WILLIAMS e DURRANT-WHYTE, 2002) mostram que os comportamentos e os árbitros na arquitetura DAMN podem atuar de forma descentralizada, permitindo a construção de um sistema com processamento distribuído e com flexibilidade para inserção ou remoção de novos comportamentos e árbitros. Dessa forma, são utilizados três computadores interconectados por uma rede padrão *Ethernet* (IEEE 802.3), sendo um embarcado no submersível e dois na superfície. Um dos computadores da superfície é utilizado exclusivamente para processamento de imagens da câmera embarcada no submersível.

Como uma evolução dos trabalhos Seraji (2000) e Howard e Seraji (2001), e baseando-se nos trabalhos Saffiotti (1997) e Tunstel, Lippincott e Jamshidi (1997), Seraji e Howard (2002) apresentam a construção e os experimentos realizados com uma arquitetura baseada em comportamentos difusos para navegação autônoma de robôs em terrenos irregulares em ambientes abertos, dinâmicos e pouco estruturados. A arquitetura é desenvolvida por pesquisadores do Laboratório de Tecnologias de Propulsão da Agencia Espacial Norte Americana (NASA-JPL) para utilização nos robôs da família *Sojourner* para exploração do planeta Marte (MATTHIES et al. 1995 apud SERAJI e HOWARD, 2002).

A arquitetura proposta e testada em Seraji e Howard (2002) possui comportamentos robóticos compostos por controladores difusos para navegação que contêm bases de regras e variáveis lingüísticas associadas à navegação, assim como controladores difusos para pesos que contêm bases de regras e variáveis lingüísticas que geram pesos aplicados na defuzzificação dos controladores difusos para navegação. Assim existem dois tipos de regras, um tipo para navegação e outro para pesos. As regras lingüísticas de navegação possuem o seguinte formato:

$$if \quad C \quad than \quad A \tag{9}$$

Onde: "*C*" é uma <u>condição</u> representada por uma expressão lingüística composta por uma ou mais variáveis antecedentes associadas pelo processo de composição, através de operados como, por exemplo, união, intersecção e complemento. "*A*" é uma <u>ação</u> representada por uma variável conseqüente (detalhes teóricos são postos no item 3.1).

Após o processo de fuzzificação, os controladores difusos de navegação avaliam as leituras do sistema sensorial do robô diante das condições "*C*", para então, associar essas leituras a ações "*A*". Posteriormente, essas ações formadas por conjuntos difusos de saída são submetidas a um processo de defuzzificação através do método centro de gravidade modificado para incluir valores exatos dos pesos fornecidos pelos controladores difusos de pesos (SERAJI e HOWARD, 2002). As regras lingüísticas de peso possuem o seguinte formato:

$$if \quad S \quad than \quad W \tag{10}$$

Onde: *S* é uma situação representada por uma expressão lingüística composta por uma ou mais variáveis antecedentes associadas pelo processo de composição, através de operados como, por exemplo, união, intersecção e complemento. *W* é um fator de peso representado por uma variável conseqüente (detalhes teóricos são postos no item 3.1).

Após o processo de fuzzificação, os controladores difusos de peso avaliam as leituras do sistema sensorial do robô diante das situações *S*, para então, associar essas leituras a fatores de peso *W*. Posteriormente, esses fatores formados por conjuntos difusos de saída são submetidos a um processo de defuzzificação através do método centro de gravidade que gera os valores exatos dos pesos utilizados no processo de defuzzificação dos controladores de navegação (SERAJI e HOWARD, 2002).

A arquitetura robótica baseada em comportamentos difusos apresentada em Seraji e Howard (2002) é ilustrada na Figura 10. Nessa ilustração são representados

os comportamentos robóticos compostos pelos dois tipos de controladores e as bases de regras descritas anteriormente nesse mesmo item.

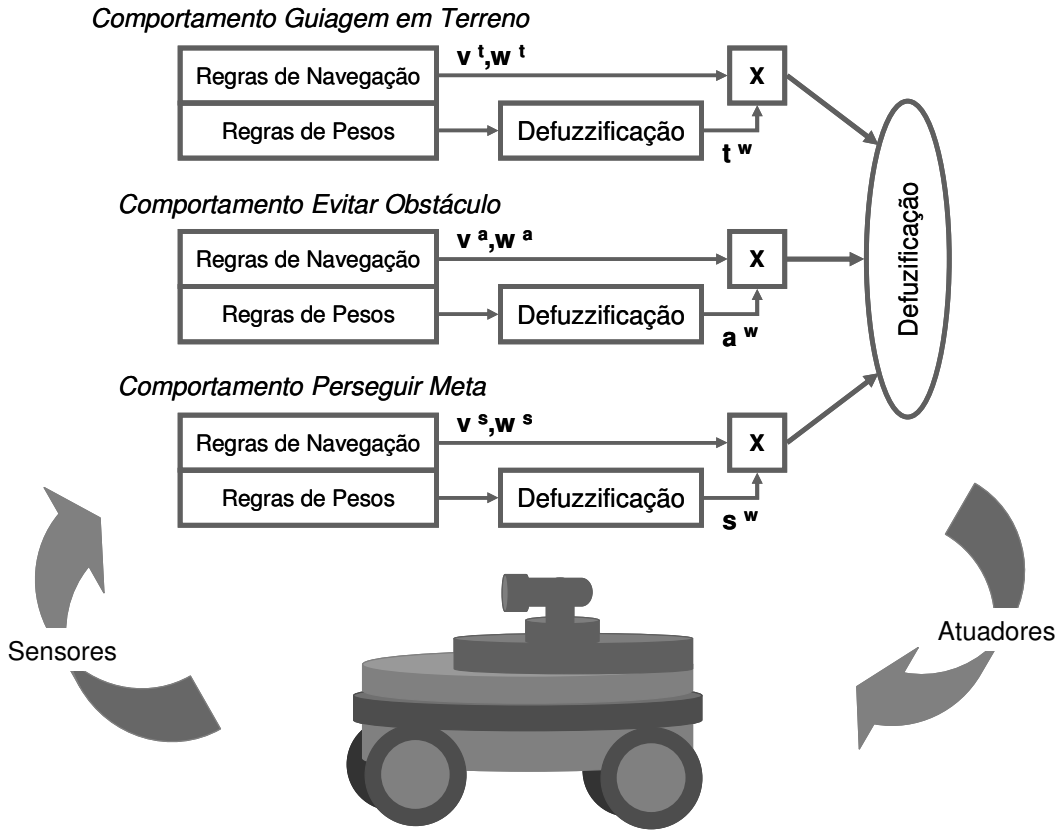


Figura 10 - Arquitetura robótica para navegação em terrenos irregulares. Fonte: adaptado de Seraji e Howard (2002)

Como pode ser visto na Figura 10, a arquitetura ilustrada possui três comportamentos e cada comportamento possui um conjunto de regras para navegação e outro para pesos. Cada conjunto de regras de navegação é dividido em dois subconjuntos, sendo um subconjunto associado com o movimento translação que tem como conjunto difuso de saída a velocidade v, e outro subconjunto associado com o movimento de rotação do robô que tem como conjunto difuso de saída a rotação w. Dessa forma, os conjuntos difusos de saída (v^t, w^t) e o peso

crisp t^w contêm as recomendações do comportamento travessia de terreno (*traverse-terrain behavior*), os conjuntos difusos de saída (v^a, w^a) e o peso crisp a^w contêm as recomendações do comportamento *evitar-obstáculo* (a*void-obstacle behavior*) e os conjuntos difusos de saída (v^s, w^s) e o peso crisp s^w contêm as recomendações do comportamento *perseguir-meta* (*seek-goal behavior*).

Como ilustrado na Figura 10, os valores crisp de velocidade e rotação (velocidade angular) para o sistema de atuação do robô são obtidos pelo processo de defuzzificação aplicado aos subconjuntos difusos de saída associados aos valores crisp dos pesos. Nesse processo, as recomendações de cada comportamento são combinadas de acordo com os pesos associados, criando um processo dinâmico, em que em cada iteração os pesos são ajustados dependendo da situação momentânea encontrada pelo robô no ambiente.

As arquiteturas robóticas baseadas em comportamentos difusos permitem que o conhecimento humano seja embarcado nos sistemas robóticos tanto para composição de comportamentos assim como para coordenação desses comportamentos como visto em item anterior (Item 3.2) e neste item (Item 3.3). Entretanto, devido às próprias características da Lógica Difusa, as arquiteturas descritas nesses itens não permitem ao robô aprender a partir das experiências de sucesso e de erro ocorridas durante a realização das tarefas robóticas em seu ambiente de atuação.

Baseando-se nos trabalhos Bonarini (1997), Saffiotti (1997) e Tunstel, Lippincott e Jamshidi (1997), Hagras, Callaghan e Colley (2001) propõem uma arquitetura robótica baseada em comportamentos difusos hierárquicos associados a

um módulo que permite ao robô aprender e adaptar-se a partir das experiências de sucesso e de erro ocorridas ao longo de sua operação. Esse tipo de processo de aprendizado utilizado pelos autores denomina-se Aprendizado por Reforço (*Reinforcement Learning*), sendo que, em alguns trabalhos utiliza-se a ferramenta matemática e computacional denominada Algoritmo Genético (*Genetic Algorithm*) para sua construção. Essa ferramenta algorítmica imita o processo de evolução genético da natureza, baseado nos mecanismos de seleção, recombinação e mutação, para modificar e re-configurar os parâmetros de um sistema computacional.

A associação de Lógica Difusa e Algoritmo Genético em sistemas de controle denominados Sistemas Difuso-Genéticos (*Fuzzy-Genetic Systems*) não constituí um inovação desenvolvida por Hagras, Callaghan e Colley (2001). Entretanto os mesmos autores afirmam que a aplicação de uma arquitetura robótica de comportamentos com sistema difuso-genético para aprendizado e adaptação em ambientes abertos e não estruturados não havia sido realizada. Os autores mencionam que as técnicas propostas anteriormente ao seu trabalho utilizam simulação e espaço fixo de soluções para desenvolver o aprendizado (HAGRAS, CALLAGHAN e COLLEY, 2004).

O sistema difuso-genético descrito em Hagras, Callaghan e Colley (2001) possui patente Britânica com número 99-10539.7. Nesse sistema, os comportamentos são construídos a partir de regras difusas. As recomendações dos comportamentos são combinadas por pesos definidos por um controlador difuso específico. Dessa forma, a combinação dos comportamentos é realizada após a defuzzificação através da combinação dos comandos crisp de cada comportamento associado a um peso. O módulo que permite ao robô aprender e adaptar-se,

denominado Motor de Experiência Associativa (*Associative Experience Engine*), atua sobre as funções de pertinência dos conjuntos de entrada, sobre as regras difusas de cada comportamento e sobre o controlador difuso gerador dos pesos. O módulo Motor de Experiência Associativa modifica os elementos citados de acordo com mecanismos da ferramenta Algoritmo Genético. A Figura 11 ilustra a arquitetura robótica proposta por Hagras, Callaghan e Colley (2001) e patenteada no Reino Unido.

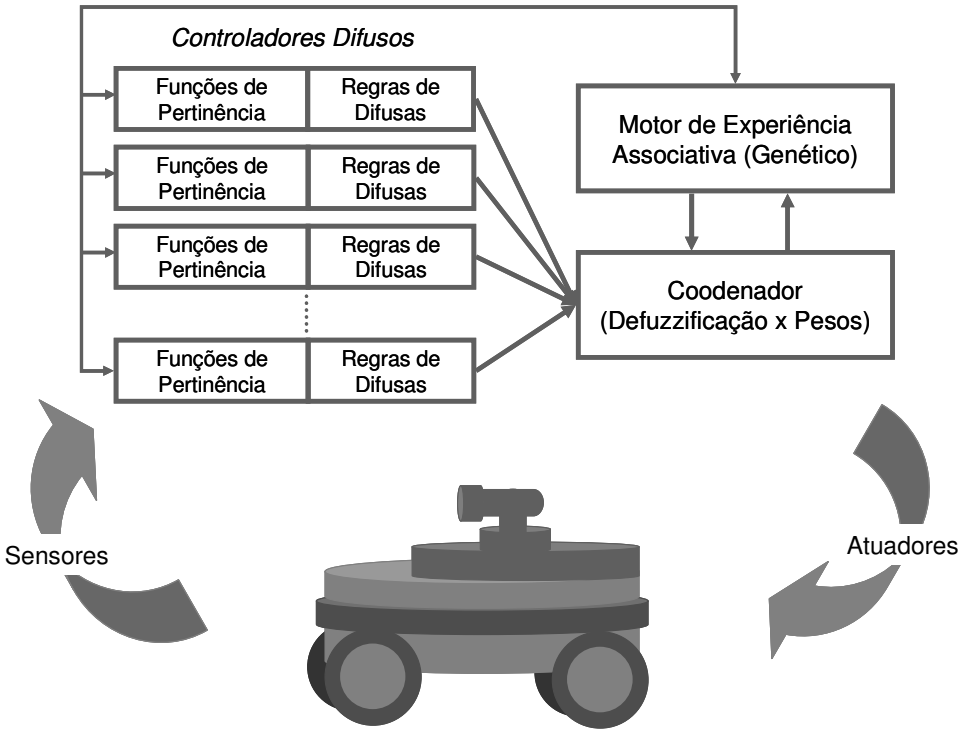


Figura 11 - Arquitetura robótica baseada em sistemas difuso-genéticos. Fonte: adaptado de Hagras, Callaghan e Colley (2001).

Hagras, Callaghan e Colley (2004) apresentam a aplicação da arquitetura ilustrada na Figura 11 em um robô móvel para navegação em ambientes dinâmicos e pouco estruturados. São implementados quatro comportamentos: *perseguir meta* (*goal-seeking*), *evitar obstáculo* (*obstacle-avoidance*), *seguir barreira à direita* (*right*

edge-following), *seguir barreira à esquerda* (*left edge-following*). Os comportamentos e o módulo de coordenação são representados por conjuntos difusos de entrada e bases de regras específicas para cada um desses elementos segundo proposto por Saffiotti (1997). Como citado anteriormente neste mesmo item (Item 3.3), o processo de aprendizado e adaptação utiliza algoritmo genético para ajustar as funções de pertinência dos conjuntos de entrada, as regras difusas de cada comportamento e o controlador difuso gerador dos pesos. Esse processo de aprendizagem e adaptação é realizado através de tentativas iniciadas a partir da identificação de um erro, sendo executado durante a operação normal do robô em seu ambiente de atuação.

Como citado anteriormente, os trabalhos Hagras, Callaghan e Colley (2001), Rosenblatt, Williams e Durrant-Whyte (2002), Seraji e Howard (2002) são referências para trabalhos atuais que utilizam lógica difusa para construção de comportamentos ou para coordenação de suas recomendações em arquiteturas de robôs móveis que atuam em ambientes pouco estruturados e dinâmicos. Junto a esses trabalhos, Saffiotti, Konolige e Ruspini (1995) e Tunstel, Lippincott e Jamshidi (1997) formam um conjunto de referências de citação freqüente e que mostram a evolução dos trabalhos de pesquisa desses mesmos grupos, desde a proposição até a validação funcional das arquiteturas robóticas desenvolvidas. A Tabela 1 sistematiza as principais características das arquiteturas robóticas baseadas em controladores difusos desses trabalhos de pesquisa.

Tabela 1 - Características de arquiteturas robóticas baseadas em sistemas difusos

Especificações da Arquitetura / ***Referências***		**Saffiotti, Konolige e Ruspini (1995)**	**Tunstel, Lippincott e Jamshidi (1997)**	**Hagras, Callaghan e Colley (2001)**	**Rosenblatt, Williams e Durrant-Whyte (2002)**	**Seraji e Howard (2002)**
Processamento	Centralizado	x	x	x		x
	Distribuído				x	
Método de Implementação dos Comportamentos	Difusos	x	x	x		x
	Difusos Hierárquicos		x	x		
	Diversos				x	
Método de Coordenação	Arbitragem				x	
	Fusão de comandos	x	x	x		x
	Seleção de Comportamentos	x	x	x	x	x
	Controlador Difuso	x	x	x	x	x
Ambiente dos Experimentos	Simulação física		x	x		x
	Fechado e controlado		x			
	Fechado e dinâmico	x				
	Aberto e controlado			x		
	Aberto, dinâmico e pouco estruturado				x	x

Como se pode observar pela Tabela 1, as características comuns das arquiteturas apresentadas são os métodos de coordenação de comportamentos baseados na seleção de comportamentos e a utilização de controlador difuso para a coordenação. A fusão de comandos também é utilizada pela maioria dos autores, com exceção dos Rosenblatt, Williams e Durrant-Whyte (2002) que utilizam um método de arbitragem descrito anteriormente nesse mesmo item (Item 3.3). Embora, os autores apontem que a fusão de comandos (antes da defuzzificação) é mais eficiente no aproveitamento das informações geradas pelos comportamentos

(recomendações), no método de arbitragem os comportamentos realizam o processamento de forma independente e permitem que este seja distribuído.

Quanto à implementação dos comportamentos, observa-se pela Tabela 1 que todas as arquiteturas utilizam comportamentos difusos, sendo que a arquitetura DAMN apresentada por Rosenblatt, Williams e Durrant-Whyte (2002) tem maior flexibilidade nesse parâmetro, pois permite a utilização de comportamentos construídos por outros métodos. Ainda com relação à implementação de comportamentos, as arquiteturas propostas por Tunstel, Lippincott e Jamshidi (1997) e por Hagras, Callaghan e Colley (2001) utilizam comportamentos hierárquicos, ou seja, comportamentos mais simples são utilizados para construção de comportamentos mais complexos, o que facilita o processo de desenvolvimento e de implementação da arquitetura como descrito no Item 3.2.

Todos os trabalhos apresentados na Tabela 1 buscam soluções de controle para aplicação de robôs em ambientes dinâmicos e pouco estruturados. Porém somente Saffiotti, Konolige e Ruspini (1995) e Rosenblatt, Williams e Durrant-Whyte (2002) mostram resultados obtidos com a aplicação do robô e de sua arquitetura de navegação em seu ambiente real de operação. Ressalta-se que, embora Seraji e Howard (2002) e Hagras, Callaghan e Colley (2001) não tenham realizado experimentos com o robô em seu ambiente real de operação, foram realizadas simulações de operação em ambientes abertos similares aos reais.

Os autores nomeados na Tabela 1 concordam que a implementação de comportamentos difusos é viável e traz uma série de benefícios aos sistemas robóticos, além de não constituir mais um desafio, principalmente para navegação em ambientes em que incertezas e imprecisões são características naturais. Os

autores também afirmam que o desafio está na coordenação dos comportamentos e na sua integração com outros módulos robóticos. Nesse sentido, há uma profusão de pesquisas atuais em arquiteturas robóticas baseadas em sistemas difusos que buscam soluções para aprimorar a coordenação comportamentos difusos e para desenvolvimento e integração de camadas ou módulos com funcionalidades específicas para cada aplicação da arquitetura. Por exemplo, os autores Yang, Moallem e Patel (2005) apresentam uma arquitetura baseada em duas camadas com comportamentos difusos semelhantes aos definidos por Seraji e Howard (2002). Uma camada é responsável por determinar pontos intermediários de navegação para atingir o alvo final. Outra camada utiliza comportamentos reativos para guiar o robô entre os pontos intermediários.

Sob o mesmo foco de Yang, Moallem e Patel (2005), Remondini e Saffiotti (2006) utilizam os resultados de Saffiotti, Konolige e Ruspini (1995) e propõem um comportamento de alto nível responsável por executar um plano de navegação através da avaliação e seleção de sub-comportamentos. Ip, Rad e Wong (2004) desenvolvem uma arquitetura hierárquica de comportamentos difusos que utiliza um algoritmo fundamentado na Teoria Bayesiana (Teorema de Bayes) para determinar espaços livres em um ambiente desconhecido e, através de similaridade entre parâmetros de navegação adquiridos em treinamento prévio, selecionar comportamentos durante a operação do robô no ambiente. Maaloufa, Saada e Saliahb (2006) desenvolvem um controlador difuso que associa entradas geradas por um modelo cinemático de um robô à saídas de comando para guiar o robô entre pontos intermediários até um ponto final desejado. Watanabe et al. (2005), seguindo a mesma linha de Hagras, Callaghan e Colley (2001), propõe e compara dois métodos de coordenação de comportamentos baseados em algoritmos genéticos.

Malhotra e Sarkar (2005) propõem um método de combinação das recomendações de comportamentos dos difusos *evitar obstáculo* (*obstacle avoidance*) e *perseguir meta* (*goal acquisition*), associados ao método de campos potencias desenvolvido por Khatip (1986) e comentado anteriormente no Item 2.3. Gaonkar, DelSorbo e Rattan (2005) apresentam uma arquitetura robótica de navegação baseada em quatro camadas com comportamentos difusos, cujas recomendações são submetidas a um processo de combinação e seleção por pesos. Parhi (2005) utilizam um módulo de seleção de tarefas baseado na ferramenta de modelagem e simulação Redes de Petri para coordenar as recomendações dos comportamentos difusos, *evitar obstáculo* (*obstacle avoidance*) e *perseguir alvo* (*target seeking*), para navegação de um robô móvel.

Além da busca por métodos de coordenação e de desenvolvimento e integração de camadas, a definição correta dos comportamentos difusos ou dos conjuntos de comportamentos difusos constitui outro foco de pesquisas atuais em arquiteturas robóticas baseadas em sistemas difusos. Nos trabalhos Rosenblatt, Williams e Durrant-Whyte (2002) e Seraji e Howard (2002), descritos anteriormente nesse mesmo item (Item 3.3), constata-se a definição de comportamentos específicos para a tarefa de navegação nos respectivos ambientes de atuação dos robôs, o ambiente submarino e a superfície irregular e rochosa de planetas. Na área agrícola, os autores Garcia-Alegre, Bustos e Guinea (1995) e Garcia-Alegre e Guinea (1997) já apontavam requisitos para definição de comportamentos robóticos para tarefas agrícolas. Recentemente, os autores Blackmore, Have Fountas (2002), Blackmore et al. (2004) e Pedersen et al. (2005) discutem e apresentam especificações e métodos para definição de comportamentos para compor arquiteturas robóticas relacionadas com as tarefas agrícolas executadas atualmente,

principalmente sob o contexto da Agricultura de Precisão. Nesse sentido, o Capítulo 4 apresenta uma sistematização de informações sobre os principais sistemas pesquisados e utilizados em tarefas agrícolas praticadas atualmente, principalmente para navegação e guiagem autônomas.

4. SISTEMAS AGRÍCOLAS AUTÔNOMOS

Os sistemas autônomos para veículos autônomos agrícolas (VAA's) e robôs agrícolas móveis (RAM's) encontrados nos trabalhos de pesquisa atuais podem ser divididos em três estágios: percepção, computação e atuação. Entre as linhas de pesquisa atuais nesses temas, um dos destaques é a busca por soluções para navegação e guiagem autônomas ou semi-autônoma de máquinas agrícolas, principalmente para tratores (GUO, ZHANG e FENG, 2003; MIZUSHIMA, et al., 2003; MIZUSHIMA, NOGUCHI e ISHII, 2005; OKSANEN, LINJA e VISALA, 2005; BLACKMORE e GRIEPENTRONG, 2006; ROVIRA-MÁS e HAN, 2006). Essas pesquisas procuram atender as necessidades das práticas agrícolas atuais que requerem controle mais preciso de posicionamento e de velocidade como destacado no Capítulo 1 (VRT, Tecnologias *On-The-Go* e Piloto Automático).

Como pode ser notado em trabalhos de pesquisa atuais, como, por exemplo, em Blackmore e Griepentrong (2006) e em Rovira-Más e Han (2006), existem esforços para utilizar na área agrícola as tecnologias e metodologias estabelecidas em outras áreas de aplicação da guiagem e da navegação autônomas para veículos autônomos e robôs móveis. Entretanto, um dos desafios identificados é a adaptação dessas tecnologias e metodologias para o ambiente agrícola, que possui características próprias e distintas dos outros ambientes.

Observa-se que o foco das pesquisas em guiagem e em navegação autônomas tem sido no estágio de percepção. Desde os trabalhos de pesquisa

desenvolvidos e publicados na ultima década do milênio passado, como por exemplo, Le Bars et al. (1997); Stombaugh et al. (1997), Benson e Hummel (1998), Benson et al. (1998), Sanchiz et al. (1998) e Åstrand e Baerveldt (1999), constata-se que os esforços de pesquisa têm se concentrado nas busca por aplicações de novos sensores e na integração de sensores diversos, que aproveitam algumas características estruturais específicas do ambiente, como as linhas de plantio. Esse fato também pode ser constatado nos trabalhos de revisão Keicher e Seufert (2000), Reid et. al. (2000) e Torii (2000) sobre esses temas.

Nos trabalhos sobre tendências da automação agrícola do novo milênio, os autores Auernhammer (2001) e Michael Kassler (2001) apontam a aplicação de robôs na agricultura como uma das tendências promissoras e inevitáveis. Entretanto, embora a abordagem híbrida e a abordagem baseada em comportamentos hierárquicos e difusos tenham suas bases estabelecidas para aplicações em robô móveis há aproximadamente dez anos, a aplicação dessas abordagens em sistemas agrícolas autônomos só tomou impulso nos últimos cinco anos. Antes desse período, as soluções para sistemas de navegação e guiagem autônomas de máquinas agrícolas foram focadas essencialmente no estágio de percepção, baseando-se em sistemas deliberativos. Uma exceção são os trabalhos de pesquisa em arquiteturas baseadas em comportamentos para robôs agrícolas com resultados apresentados pelo grupo de pesquisadores Garcia-Alegre et al. (1995), Garcia-Alegre e Guinea (1997) e Garcia-Alegre et al. (2001).

Atualmente, é crescente o número de trabalhos de pesquisa que buscam aprimorar o estágio de computação desses sistemas, capacitando-os à realização de tarefas autônomas (BLACKMORE e GRIEPENTRONG, 2006). Assim, podem-se estabelecer dois panoramas atuais para as pesquisas em sistemas agrícolas

autônomos. O primeiro é constituído por trabalhos de pesquisa que buscam desenvolver sistemas autônomos fundamentados na abordagem deliberativa com enfoque principal no estágio de percepção. O outro se constitui de trabalhos recentes que buscam por soluções fundamentadas na abordagem híbrida baseada em comportamentos. Esses dois panoramas são detalhados nos itens a seguir (Itens 4.1 e 4.2)

4.1. Abordagem deliberativa com enfoque no sistema de percepção

Nos trabalhos de revisão Keicher e Seufert (2000), Reid et. al. (2000) e Torii (2000) e nos trabalhos de pesquisa atuais Guo, Zhang e Feng (2003), Mizushima et al. (2003), Mizushima, Noguchi e Ishii (2005), Oksanen, Linja e Visala (2005), Blackmore e Griepentrong (2006) e Rovira-Más e Han (2006), sobre navegação e guiagem autônomas de veículos autônomos e de robôs móveis, observa-se que os principais dispositivos utilizados no estágio de percepção desses sistemas são:

- Câmeras (Visão Computacional ou Visão de Máquina);
- Receptor DGPS (*Differential Global System Navigation*: Sistema de Posicionamento Global Diferencial) ou receptor RTK-GPS (*Real Time Kinematic GSP*: GPS Cinemático);
- Varredor Laser (*Scanner Laser*);
- Sistema Inercial (*INS – Inertial Navigation System*: Sistema de Navegação Inercial) baseado em giroscópio, acelerômetro e/ou GDS (*Geomagnetic Direction*

Sensor: Sensor Geomagnético de Direção) e em odômetros para medição de velocidade (intensidade e direção).

4.1.1 Filtro de kalman em sistemas agrícolas autônomos

Nos mesmos trabalhos citados anteriormente no início deste mesmo item (Capítulo 4), pode-se constatar que uma solução bastante utilizada para integração ou para fusão das informações dos sensores (Fusão de Sensores) é a ferramenta matemática conhecida Filtro de Kalman (FK). O FK desenvolvido por Kalman (1960 apud WELCH e BISHOP, 2002) é um estimador utilizado para prever o estado de um sistema dinâmico linear perturbado por um Ruído Branco Gaussiano, a partir de medidas que são funções lineares do estado do sistema, mas corrompidas pelo Ruído Branco Gaussiano aditivo.

No processo de guiagem dos VAA's e RAM's, os dados de sensores combinados através do FK são aplicados a um modelo matemático do sistema móvel. O modelo gera saídas que são normalmente o ângulo de guiagem e o deslocamento lateral que alimentam o estágio de atuação. A Figura 12 ilustra sistema de coordenadas para o sistema com as definições para ângulo de guiagem (g) e deslocamento lateral (d), como tem sido utilizado na maioria dos trabalhos pesquisados.

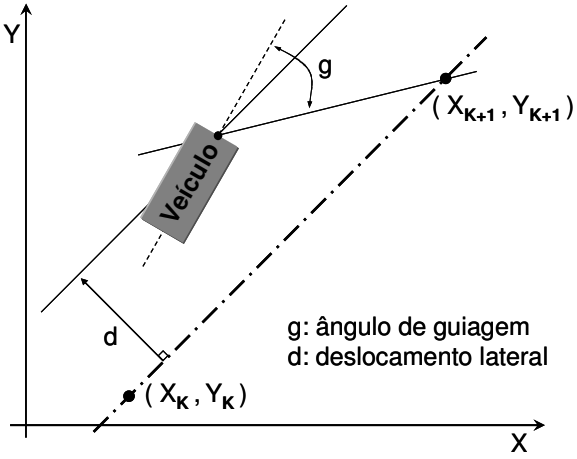


Figura 12 - :Ilustração de sistemas de coordenadas para sistema de guiagem e navegação. Fonte: adaptado de Noguchi, Ishii e Terão (1997)

Como descreve Welch e Bishop (2002), a operação de um FK discreto estima um processo utilizando uma forma realimentada de controle. O filtro estima o estado de um processo em um instante de tempo e, então, obtém a realimentação na forma de uma medição com ruído. Dessa forma, as equações do FK são agrupadas em dois conjuntos: equações de atualização no tempo e equações de atualização das medições. As equações de atualização no tempo projetam o estado do sistema à frente no tempo e as equações de atualização das medições são responsáveis pela incorporação das medidas e da covariância do ruído de medição. Considerando a equação de estado de um sistema (11) e a equação da medição (12):

$$x_k = A.x_{k-1} + B.u_k + w_{k-1} \tag{11}$$

$$z_k = H.x_k + v_k \tag{12}$$

Onde: x_k é o estado *a posteriori* estimado no instante de amostragem k dada um medição z_k; x_{k-1} é o estado *a posteriori* estimado no instante de amostragem k-1;

u_k é uma entrada externa ao sistema; A, B, H são as matrizes das equações do sistema; e w_{k-1} e v_k são variáveis que representam os ruídos do processo e da medição.

As distribuições de probabilidade do ruído do processo e da medição são representadas pelas relações (13) e (14):

$$p(w) \sim N(0, Q) \tag{13}$$

$$p(v) \sim N(0, R) \tag{14}$$

Onde: Q, R são as covariâncias do ruído do processo e do ruído da medição relativos à distribuição normal de probabilidade.

Considerando as relações (11), (12), (13) e (14), as relações (15) e (16) compõem o conjunto de equações de atualização no tempo e as relações (17), (18) e (19) compõem o conjunto de equações de atualização das medições. O procedimento de aplicação do FK é realizado segundo a seguinte seqüência:

1º. Projeção do estado do sistema à frente: $x_k^- = A.x_{k-1} + B.u_k$ (15)

2º. Projeção da covariância do erro a frente: $P_k^- = A.P_{k-1}.A^T + Q$ (16)

3º. Cálculo do ganho do FK: $K_k = P_k^-.H^T.(H.P_k^-.H^T + R)^{-1}$ (17)

4º. Atualização do estado a partir da medição: $x_k = x_k^- + K_k.(z_k - H.x_k^-)$ (18)

5º. Atualização da covariância do erro: $P_k = (I - K_k.H).P_k^-$ (19)

Onde: x_k^- é o estado *a priori* estimado no instante de amostragem k dado; x_{k-1}: é o estado *a posteriori* estimado no instante de amostragem k-1; P_k^-, P_{k-1} são estimações das covariâncias do erro *a priori* e *a posteriori* respectivamente; P_k é estimação da covariância do erro *a posteriori* atualizado; e K_k é ganho do FK.

Modelos matemáticos derivados de FK constituem representações razoáveis para muitos problemas práticos, incluindo problemas de controle e de estimação. O Filtro de Kalman Estendido (FKE) é uma aplicação do FK a problemas não lineares, através da linearização do sistema em torno de um ponto (WELCH e BISHOP, 2002). A utilização do FK em técnicas para fusão de sensores e compensação de erros de medição, especialmente para posicionamento e localização de robôs móveis, é um procedimento comum e com inúmeras aplicações (MURATA e HIROSE, 1993; BORENSTEIN e FENG, 1996; SANDI, HEMERLY e LAGES, 1997; ROUMELIOTIS, SUKHATME e BEKEY, 1999).

Dois parâmetros que influenciam fortemente a elaboração de um sistema de percepção e consequentemente todos os sistema robóticos são custo e precisão. Em geral quanto mais preciso, mais caro será o sensor e mais caro será o sistema. Assim o que se tem buscado são sistemas que potencializam as boas características dos sensores utilizados e minimizam a influências de suas deficiências, para que se possa utilizar sensores menos precisos e mais baratos (HAGUE, MARCHANT & TILLETT, 2000). Como será descrito a seguir, é foco de pesquisas atuais a busca por ferramentas matemáticas, como o FK, que realizam a integração das medições de cada sensor compensando as deficiências de cada um e gerando um valor final mais preciso.

4.1.2 Sistemas agrícolas autônomos com enfoque no sistema de percepção

Em pesquisas de sistemas para guiagem autônoma para máquinas agrícolas, Stombaugh, Benson e Hummel (1998) utilizam um RTK-GPS para navegação baseada em mapa de uma área de gerado *a priori*. Com este sistema de percepção, o erro obtido na tarefa de seguir as linhas definidas a partir dos dados do mapa foi 0,160 m em uma velocidade de operação de 4,5 m/s. Pesquisadores do mesmo grupo obtiveram um erro de 0,084 m utilizando um sistema de percepção com RTK-GPS e GDS integrados através de fusão de sensores utilizando método baseado em FKE (NOGUCHI et al, 1998 apud REID et al, 2000).

Benson et al (1998) mostram que com um sistema com GDS e um receptor GPS com precisão inferior a um receptor RTK-GPS, integrados por fusão de sensores baseada em FKE, pode-se obter um erro menor que o erro de 0,160 m obtido com o sistema de percepção baseado somente em um RTK-GPS. Este trabalho, dependendo da aplicação e precisão desejadas, pode ser uma interessante referência, pois um inconveniente em utilizar um sistema RTK-GPS é o seu alto custo. Já um sistema com um receptor DGPS mais simples (precisão menor) e um GDS terá custo inferior.

Le Bars et al (1997) apresentam uma comparação entre estágio de percepção composto por DGPS, outro composto por radar *doppler* (medição de velocidade pelo Efeito Doppler), GDS e receptor DGPS e outro composto por radar doppler, giroscópio (medição de aceleração) e receptor DGPS. Nos dois sistemas com diferentes sensores é utilizado FKE para fusão de sensores e os resultados permitem verificar que, sob as condições em que os testes foram realizados, a

combinação entre radar *doppler*, GDS e receptor DGPS apresenta maior precisão na tarefa de guiagem e navegação autônoma.

Pesquisas recentes realizam experimentos com sistemas inerciais e receptores GPS de baixo custo combinados através FK na busca por sistemas de custo menor, aumentando a precisão dos sistemas com RTK-GPS (OKSANEN, LINJA e VISALA, 2005). Guo, Zhang e Feng (2003) mostram que um sistema inercial tipo IMU (*Inertial Measurement Unit*) de baixo custo contendo um giroscópio, um acelerômetro e um GDS, combinado através de FK com um receptor DGPS, melhora sensivelmente a precisão do sistema de posicionamento e suporta a navegação durante falhas momentâneas na recepção do sinal do GPS. Mizushima, Noguchi e Ishii (2005) utilizam um sistema semelhante com 3 giroscópios (giroscópio 3D) e dois inclinômetros combinados com um receptor DGPS através de FK, e mostram que esse sistema tem precisão semelhante a de um receptor GPS-RTK. Rovira-Más e Han (2006) apresentam um método para combinar através de FKE dados de um sistema de visão com câmera digital monocular e um receptor DGPS.

A visão computacional constitui uma das tecnologias pesquisadas para aplicação em sistemas autônomos de VAA's e RAM's. Em ambientes abertos, como o ambiente agrícola, há uma série de desafios a serem superados para a implementação de um sistema de visão, em relação aos sistemas autônomos que operam em ambiente fechados. No ambiente agrícola, a luminosidade apresenta uma variabilidade espacial e temporal acentuada e também apresenta barreiras físicas que são menos previsíveis que ambientes fechados. Dependendo das condições climáticas e do horário do dia a luminosidade se altera alterando os perfis de sobras. Também, embora a estrutura de linhas de plantio seja relativamente estruturada, falhas nessas linhas, finais de linhas ou corpos estranhos não são

incomuns. Assim, fatores como estes têm levado à adoção de câmeras especiais, mais apropriadas para aplicação em ambientes abertos. Também é comum a utilização de computadores dedicados ou até mesmo a utilização de placas de aquisição e processamento de imagens especiais como as *framegrabbers*, para se obter sistemas visão com capacidade de processamento para operar em tempo real (SANCHIZ et al, 1998, SOUTHALL et al, 2002; HAGUE, SOUTHALL e TILLETT, 2002).

No trabalho de revisão de Keicher e Seufert (2000) sobre veículos agrícolas autônomos na Europa é apresentado o trabalho de Keicher et al. (1998) em que foi desenvolvido um sistema de visão para navegação autônoma de um trator. Este sistema era constituído de um computador PC (*Personal Computer* – Computador Pessoal) com freqüência do processador de 300 MHz dedicado, uma placa *framegrabber* e uma câmera CCD RGB (modelo de cores *Red-Green-Blue*). O sistema de controle em tempo real operando a 180 MHz era obtido com o sistema de visão operando a uma taxa de aquisição de imagem de 50 FPS (50 *frames per second* – 50 quadros por segundo ou 50 Hz) e resolução de 768x572 pixels, para se ter uma precisão de 0,050 m com o trator deslocando-se a uma velocidade de 3,333 m/s (12 km/h).

Em Astrand & Baerveldt (1999) é relatado o desenvolvimento de um robô móvel para atuar em plantações de beterraba. Neste trabalho o sistema de visão utilizado era composto de um PC dedicado com freqüência do processador de 333 MHz, uma placa *framegrabber* e uma câmera CCD. O sistema de visão operava de 8 a 12 FPS para atender ao sistema de controle de navegação que permitia ao veículo atingir uma velocidade de 0,556 m/s (2 km/h) e com precisão de 0,450 m.

Como se pode observar o desempenho do sistema implementado por Astrand e Baerveldt (1999) (0,556 m/s e 0,450 m) é inferior ao sistema apresentado por Keicher et al. (1998) apud Keicher e Seufert (2000) (3,333 m/s e 0,050 m). Isto acontece porque a velocidade de operação do veículo e a precisão na navegação estão relacionados com a capacidade de resposta do sistema, que nos dois casos dependem do sistema de visão. Quanto menor o tempo de resposta melhor a precisão e/ou mais rápido o veículo pode operar e vice-versa.

No trabalho de Astrand e Baerveldt (1999) foi utilizado um artifício que se tornou comum em aplicações de visão computacional em máquinas agrícolas. Foi feita a adaptação de uma câmera monocromática para a faixa do infravermelho, através da remoção do filtro de infravermelho da câmera e inserção de uma lente de filtragem da banda visível, permitindo assim utilizar o sistema de visão na escala de infravermelho próximo. A vantagem desse artifício é que a diferença entre a vegetação e o solo se acentua e os efeitos negativos das variações de luminosidades causados por sombras diminuem na faixa do infravermelho próximo, permitindo a identificação mais clara das linhas de plantio.

A adaptação de câmeras para a faixa de infravermelho próximo tem sido muito aplicada em trabalhos que utilizam visão para guiagem e navegação autônomas de máquinas agrícolas. O grupo de pesquisa em Veículos Autônomos Agrícolas do Departamento de Engenharia Agrícola da Universidade de Ilinois-EUA, foi um dos pioneiros na utilização desta técnica associada à utilização de algoritmos baseados na Transformada de Hough (TH) para detecção de linhas de plantio que passou a ser utilizada por outros grupos de pesquisa posteriormente (REID et al., 2000).

A TH é uma ferramenta utilizada para detectar características analiticamente representáveis em imagens binarizadas, assim como linhas, círculos e elipses. Em geral, a transformada é aplicada após a imagem sofrer um pré-processamento, usualmente a detecção de bordas (ASTRAND e BAERVELDT, 1999; KEICHER e SEUFERT, 2000; REID et al., 2000). Recentemente outras técnicas e algoritmos para sistemas de visão computacional têm sido utilizados na área agrícola, buscando reconhecer e representar virtualmente as linhas de plantio ou as ruas entre elas para que os sistemas autônomos de guiagem e navegação utilizem essa informações (SOUTHALL et al, 2002; HAGUE, SOUTHALL e TILLETT, 2002; BENSON, REID e ZHANG, 2003; ROVIRA-MÁS e HAN, 2006).

4.2. Abordagem híbrida baseada em comportamentos

Dentre os trabalhos pesquisados na área agrícola destacam-se os trabalhos do grupo de pesquisa do Instituto de Automática Industrial Arganda Del Rey, Madri, Espanha, por ser o grupo pioneiro no desenvolvimento de uma arquitetura híbrida baseada em comportamentos para robôs agrícolas. Garcia-Alegre e Recio (1998) e Garcia-Alegre et al (2001) descrevem uma arquitetura híbrida baseada em comportamentos robóticos, denominada AMARA, que teve seus fundamentos apresentados em Garcia-Alegre et al. (1995), e que vem sendo aperfeiçoada pelo grupo. A arquitetura é baseada em uma hierarquia de comportamentos para atuação em ambiente agrícola. Os comportamentos são agrupados em níveis hierárquicos segundo seus níveis de abstração para realização de determinadas tarefas sob coordenação de um módulo deliberativo. Por exemplo, para tarefas de navegação

relacionadas com um nível de abstração menor são definidos comportamentos como: *início-fim* (*start-stop*), *virar a esquerda* (*turn-left*), *virar a direita* (*turn-right*), *seguir em frente* (*forward*) e *retornar* (*backward*). Para tarefas com um nível de abstração intermediário são definidos comportamentos como: *ir para posição* (*go-to-position*), *evitar* (*avoid*) e *alinhar* (*align*). Para níveis de abstração superiores são definidos comportamentos como: *retornar para a base* (*go-home*) e *ir para as oliveiras* (*go-to-olives*).

A arquitetura híbrida AMARA foi implementada pela primeira vez em um pequeno trator-robô denominado Rojo. Garcia-Alegre et al (2001) descrevem os experimentos com a execução de planos de navegação baseados em mapa construído externamente e enviados através de enlace de rádio ao robô. Utiliza-se para isso um receptor DGPS, odômetros e um GDS no estágio de percepção do robô e um sistema eletro-pneumático desenvolvido para o estágio de atuação. O robô executa a tarefa com erro inferior a 0,500 m.

Hagras et al (2002) realizam experimentos com a arquitetura baseada em sistemas difuso-genéticos proposta em Hagras, Callaghan e Colley (2001) e descrita anteriormente no Item 3.3, para aplicação em robôs agrícolas. Os autores utilizam a arquitetura robótica ilustrada na Figura 11 do Item 3.3 com três sonares frontais para o comportamento *evitar obstáculo* (*obstacle-avoidance*), dois sonares laterais posicionados um em cada lado do robô para os comportamentos *seguir barreira à direita* (*right edge-following*) e *seguir barreira à esquerda* (*left edge-following*), e um conjunto formado por um receptor GPS e um GDS para o comportamento *perseguir meta* (*goal-seeking*).

Os experimentos descritos em Hagras et al (2002) são realizados em ambiente aberto e controlado, sendo que algumas situações possíveis de se encontrar no ambiente agrícola são impostas ao robô, como, por exemplo, terrenos irregulares com e sem cobertura vegetal, linhas (barreiras) de plantas irregulares e condições de ambientais adversas (chuva, sol, dia, noite, etc.). Durante cada experimento, quando o robô encontra uma situação adversa que não pode ser resolvida com as regras difusas de cada comportamento naquele momento, inicia-se o processo de aprendizagem e adaptação. Dessa forma, através de tentativas coordenadas pelo módulo Motor de Experiência Associativa, descrito no Item 3.3, as funções de pertinência dos conjuntos de entrada, as regras difusas de cada comportamento e o controlador difuso gerador dos pesos são ajustados de acordo com mecanismos da ferramenta Algoritmo Genético, permitindo ao robô aprender e adaptar-se. Os experimentos mostraram que o robô leva entre duas e quatro horas para adaptar-se e aprender a executar a guiagem e navegação desde o ponto de origem até um ponto final com precisão menor que 0,100 m em relação a um caminho definido como ideal.

Em Bak e Jakobsen (2004) é apresentado o desenvolvimento de uma plataforma agrícola robótica na qual o sistema de controle baseia-se numa arquitetura híbrida simples. A parte reativa é composta por controladores convencionais (malha fechada) que são responsáveis pelo controle dos motores de propulsão e direção para guiagem autônoma da plataforma. O módulo deliberativo é responsável pelo planejamento da excussão da missão através da definição de tarefas e da geração de rotas, assim como a do controlador a atuar em cada momento. Os primeiros testes mostram que a plataforma é capaz de operar

autonomamente em plantações com espaçamento de linhas entre 0,250 m e 0,500 m.

Recentemente, os autores, Blackmore, Have e Fountas (2002), Blackmore et al. (2004) e Pedersen et al. (2005) discutem e apresentam especificações e métodos para definição de comportamentos para compor arquiteturas robóticas relacionadas com as tarefas agrícolas executadas atualmente, principalmente sob o contexto da Agricultura de Precisão. Blackmore, Have e Fountas (2002) definem algumas características desejáveis para VAA´s e RAM´s e que influenciam diretamente na estrutura de uma arquitetura robótica híbrida baseada em comportamentos para esses sistemas. Entre essas características destacam-se:

- *Pequeno tamanho e baixo peso:* máquinas menores e mais leves que os tratores atuais podem atuar em áreas menores promovendo o monitoramento e o controle mais preciso de parâmetros de solo e de saúde das plantas. A redução do peso permite a utilização fontes de energia mais eficientes que os motores a combustão e também diminuem a compactação do solo, que é um problema crônico nas práticas agrícolas atuais. Obviamente, em certas tarefas a utilização de máquinas menores é inviável, como, por exemplo, na colheita ou na aplicação de insumos em áreas extensas. Entretanto, mesmo nessas áreas, RAM's podem realizar atividades de monitoramento com aquisição e processamento de grandes volumes de dados por área;

- *Diversidade de comportamentos autônomos:* o sistema de inteligência computacional deve permitir a máquina adaptar-se a diferentes tarefas e realizá-las de forma segura. Tal sistema deve viabilizar a navegação autônoma baseada em um SIG (Sistema de Informação Geográfico) através de trajetórias pré-

definidas, que são cada vez mais comuns nas práticas agrícolas, mas também deve viabilizar a navegação e exploração em áreas não mapeadas. O sistema deve permitir a realização das tarefas autônomas por longos períodos e de forma não assistida;

- *Capacidade para receber instruções e comunicar informações:* os sistemas internos da máquina devem permitir a programação de diferentes tarefas de acordo com planejamento realizado externamente. Assim, esses sistemas necessitam de interfaces flexíveis para comunicação externa de dados segundo padrões atuais para troca de informações com sistemas de gerenciamento e programação. Além disso, é necessária uma comunicação interna eficiente entre os diversos sub-sistemas da máquina para possibilitar a configuração de diferentes tarefas. Entre os padrões atuais utilizados em máquinas agrícolas, é destacada a norma ISO 11783, que define não só protocolos de comunicação e programação de tarefas internos para a máquina como protocolos para comunicação com sistemas externos. Esse padrão será discutido posteriormente no Item 4.3;

- *Cooperação com outras máquinas:* cada RAM ou VAA deve ter a capacidade para atuar individualmente na execução de tarefas, mas também necessita poder atuar de maneira autônoma, colaborativa e segura com grupos de máquinas para somar esforços, de forma coordenada, na realização de tarefas.

Blackmore et al. (2004) a Bak e Jakobsen (2004) apresentam um método para definir e construir comportamentos para RAM´s. Por esse método, as possíveis operações de uma máquina agrícola autônoma são particionadas em tarefas específicas. Essas tarefas são classificadas em dois grupos, tarefas determinísticas

e tarefas reativas. As tarefas determinísticas são planejadas no início de cada operação e demandam um conhecimento prévio do ambiente. As tarefas reativas utilizam comportamentos reativos baseados em respostas a estímulos e a contextos não determinísticos. Nesse trabalho, a metodologia proposta é simulada em um pequeno robô comercial (ATRV-jr da empresa iRobot) através de uma operação de exploração (mapeamento) ao longo de um percurso fechado pré-definido em ambiente aberto, controlado e com obstáculos. Essa operação denominada *navegar* (*navigate*) é formada por uma tarefa determinística denominada *ir para* (*goto*) e por três tarefas reativas denominadas *seguir contorno* (*follow boundary*), *perto* (*too close*) e *muito perto* (*much too close*). A tarefa determinística constitui-se da navegação com velocidade constante passando por quatro pontos georeferenciados definidos antes do início da operação. As tarefas reativas são utilizadas para evitar e contornar obstáculos e empregam, de acordo com o contexto, um dos cinco comportamentos denominados *seguir em frente* (*straight ahead*), *parar* (*stop*), *retornar* (*back u*p), *virar a esquerda* (*turn left*) e *virar a direita* (*turn right*).

O mecanismo de tomada de decisão do método proposto Blackmore et al. (2004) utiliza estruturas condicionais do tipo *se ... senão* (*if ... else*) baseadas em lógica booleana. Nesse método, os valores exatos fornecidos pelos sensores são avaliados por um conjunto encadeado de estruturas condicionais para seleção da tarefa e do comportamento atuantes em cada instante de amostragem.

Como pode ser observado nos trabalhos descritos neste item (Item 4.2), as arquiteturas robóticas propostas utilizam comportamentos primitivos similares aos desenvolvidos para robôs móveis para ambientes dinâmicos e pouco estruturados apresentados no Item 3.3. Entretanto, com relação aos comportamentos de nível de abstração mais alto, pode-se constatar a busca pela definição e pelo

desenvolvimento de comportamentos robóticos específicos para aplicações agrícolas. Com exceção de Hagras, Colley and Callaghan (2002), ainda é restrita a aplicação de mecanismos de tomada de decisão e de coordenação de comportamentos baseados em algoritmos de inteligência computacional. Isso fica evidente se comparado com as arquiteturas para robôs móveis aplicados em outras áreas, como as arquiteturas descritas no Item 3.3 e listadas na Tabela 1. Uma comparação mais detalhada entre estes trabalhos é apresentada no Capítulo 5.

4.3. Sistemas distribuídos

A capacidade de processamento exigida pelo sistema computacional de um robô móvel autônomo e os diversos sensores, atuadores e controladores necessários para as tarefas navegação e aquisição de dados, podem originar um sistema centralizado bastante complexo. Assim o desenvolvimento de um sistema de controle distribuído ou semi-distribuído constitui uma solução, senão necessária, muito bem vinda.

Uma tecnologia que tem sido adotada para promover a interconexão entre dispositivos de automação e controle em máquinas agrícolas e que permite a implementação de sistemas distribuídos é o protocolo de comunicação digital serial CAN – *Controller Area Network* (AUERNHAMMER, 2004). Sousa (2002) reafirma a importância do protocolo CAN para a área agrícola e apresenta uma sistematização do padrão internacional da ISO (*International Organization for Standardization*) baseado no protocolo CAN para aplicações em máquinas e implementos agrícolas denominado ISO 11783, e também conhecido atualmente como ISOBUS.

Atualmente, a busca pela padronização do protocolo CAN, com a implementação da ISOBUS, constitui um dos principais focos de desenvolvimento em sistemas embarcados para máquinas e implementos agrícolas (BENNEWEISS, 2005). Além dos esforços de instituições de pesquisa e associações de normas, observa-se o esforço por parte de fabricantes de máquinas, implementos e outros equipamentos para fomentar a implementação do ISOBUS. São exemplos dessas parcerias entre fabricantes e associações de normas a *Association of Equipment Manufacturers*, que congrega os fabricantes de máquinas e implementos norte-americanos, representada pelo *North American ISOBUS Implementation Task Force*, dos Estados Unidos, e a *Federation of Engineering Industry*, que congrega os fabricantes de máquinas e implementos europeus, representada pelo *Implementation Group ISOBUS*, da Comunidade Européia. No Brasil um grupo de trabalho formado por pesquisadores e representantes da indústria de tratores está buscando congregar a comunidade de fabricantes de máquinas, implementos agrícolas e componentes e sistemas eletrônicos aplicados à área, com o objetivo principal de estruturar a organização denominada Força Tarefa ISOBUS Brasil (FTI, 2007).

O propósito da norma ISO 11783 é prover um padrão aberto para interconexão de sistemas eletrônicos embarcáveis através de um barramento que, por sua vez, é um conjunto formado por fios, conectores e dispositivos de potência para promover a interconexão de dispositivos e permitir a comunicação de dados entre estes. O sistema eletrônico que promove a interconexão de um dispositivo ao barramento é denominado *Electronic Control Unit* (ECU) ou Unidade Eletrônica de Controle. Uma única ECU pode ser responsável pela conexão de um ou mais dispositivos a um barramento de comunicação. Também, um dispositivo pode ser

conectado a um barramento por uma ou mais ECU's. O conjunto formado por ECU e dispositivo constitui um nó CAN (SOUSA, 2002).

A Figura 13 ilustra um exemplo de uma rede segundo o padrão da ISO, disposto em um trator com um implemento conectado.

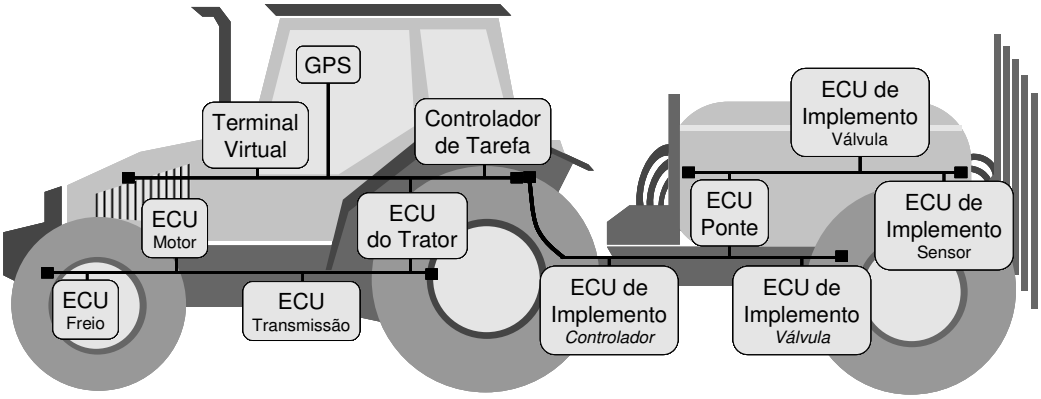


Figura 13 - Ilustração de uma rede ISO 11783.

O potencial de utilização do ISOBUS em sistemas de controle distribuído para guiagem e navegação para veículos agrícolas autônomos é relatado em Jahns (1997). Em Keicher & Seufert (2000) é reafirmado este potencial e é constatada a necessidade do desenvolvimento de soluções para integrar sistemas de controle para navegação autônoma a estes padrões. Hagras et al (2002) utiliza uma rede baseada no protocolo CAN para integração dos dispositivos de percepção e atuação e de um computador central onde está programa arquitetura difuso-genética baseada em comportamentos descrita anteriormente nos Itens 3.3 e 4.2. No trabalho de Bak e Jakobsen (2004) descrito no Item 4.2, a plataforma robótica agrícola desenvolvida utiliza uma rede baseada no protocolo CAN para integrar motores de propulsão e direção do sistema autônomo de guiagem, e uma extensão da rede é

disponibilizada para integração de implementos com sistemas baseados no protocolo CAN.

A utilização do ISOBUS em sistemas distribuídos para VAA's e RAM's é notado como uma forte tendência por Darr, Stombaugh e Shearer (2005), que apresentam o desenvolvimento de um sistema de sistema para guiagem e navegação autônomas adaptado a um trator. Godoy (2007) destaca que apesar das vantagens da utilização sistemas controle distribuído, o projeto e análise de tais sistemas são mais complicados e, muitas vezes, demandam modelos de análise e ferramentas computacionais para o estudo do seu comportamento e para simulação de operação. Nesse contexto, Godoy (2007) propõe um modelo de rede baseada no protocolo CAN e utiliza esse modelo para simulação e análise de um sistema de controle via rede (NCS – *Network Control System*) aplicado a um RAM. Os resultados dessa pesquisa mostram que o modelo desenvolvido para rede pode ser utilizado para simulação e análise como suporte ao projeto de tais sistemas. A aplicação desse modelo para avaliação do desempenho de uma rede CAN junto a arquitetura robótica proposta nesse trabalho (Capítulo 5) é descrita em detalhes no Item 6.3.

5. ARQUITETURA ROBÓTICA HÍBRIDA BASEADA EM COMPORTAMENTOS PARA GUIAGEM E NAVEGAÇÃO AUTÔNOMAS

Nos Itens 3 e 4 foram descritas e discutidas as principais metodologias e tecnologias atuais para construção de arquiteturas para robôs móveis compostas por comportamentos para aplicações em ambientes pouco estruturados e dinâmicos. No Item 3.3 são destacas as arquiteturas com comportamentos construídos com controladores difusos, assim como métodos de coordenação desses comportamentos que utilizam controladores difusos. No processo de coordenação é comum a aplicação de métodos de fusão de comandos (antes da defuzzificação) e a aplicação de métodos de arbitragem. Os métodos baseados em fusão de comando aproveitam melhor as informações geradas pelos comportamentos (recomendações), já os métodos de arbitragem permitem que o processamento de dos comportamentos seja realizado de forma independente e possibilita a utilização de sistemas distribuídos, como, por exemplo, sistemas de controle via rede.

No Capítulo 4 observa-se que o foco das pesquisas em guiagem e em navegação autônomas tem sido no estágio de percepção, através da busca por aplicações de novos sensores e na integração de sensores diversos, e que aproveitam algumas características estruturais específicas do ambiente, como as linhas de plantio. Porém, pode se notar nos trabalhos de pesquisa atuais existe um esforço para utilizar tecnologias e metodologias estabelecidas em outras áreas de aplicação da navegação e da guiagem autônomas de veículos autônomos e de

robôs móveis. Constata-se que um dos desafios atuais é a adaptação dessas tecnologias e metodologias para o ambiente agrícola, que possui características próprias e distintas dos outros ambientes. Como se pode observar nos trabalhos descritos no Item 4.2, as arquiteturas robóticas propostas utilizam comportamentos primitivos similares aos desenvolvidos para robôs móveis para ambientes dinâmicos e pouco estruturados apresentados no Item 3.3. Porém, ainda é restrita a aplicação de mecanismos de tomada de decisão e de coordenação de comportamentos baseados em algoritmos de inteligência computacional. Isso fica evidente se comparado com as arquiteturas para robôs móveis aplicados em outras áreas, como as arquiteturas descritas no Item 3.3 e listadas na Tabela 1.

Em comum, nos Itens 3.3 e 4.2, nota-se a utilização de comportamentos hierárquicos, ou seja, comportamentos mais simples são utilizados para construção de comportamentos mais complexos. Além disso, em algumas arquiteturas o método de coordenação permite a utilização de comportamentos construídos por outros métodos, além do método baseado em lógica difusa descrito no Item 3.2.

Fundamentado na identificação das tendências descritas e nas necessidades destacadas, é proposta e simulada nos sub-itens deste capítulo (Capítulo 5) uma arquitetura robótica híbrida baseada em comportamentos difusos e em coordenação de comportamentos por arbitragem para sistemas de guiagem e de navegação autônomas de VAA’s e RAM’s.

5.1. Proposta da arquitetura híbrida baseada em comportamentos difusos

Baseando-se na revisão de literatura apresentada nos Itens 3 e 4, na discussão apresentada no início deste capítulo (Capítulo 5), e especificamente na discussão apresentada no Item 3.3 referente às características de arquitetura híbridas sistematizadas na Tabela 1, são definidas as seguintes características para a arquitetura híbrida baseada em comportamentos para guiagem e navegação autônomas de RAM´s e VAA´s:

- *Comportamentos difusos hierárquicos*: podem-se construir comportamentos mais simples, com menor número de entradas e menor número de regras difusas, e combinar suas saídas para formação de comportamentos mais complexos. Possibilidade de implementação de comportamentos não difusos;

- *Método de coordenação por arbitragem com seleção de comportamentos realizada por controlador difuso*: um ou mais árbitros constituídos por controladores difusos selecionam em cada instante de amostragem um ou mais comportamentos de acordo com contextos de aplicação e com tarefas pré-definidas. Possibilidade de operação integrada com árbitros não difusos.

- *Adaptação ao ambiente agrícola e a tecnologias próprias para sistemas agrícolas*: utilização de estruturas próprias do ambiente agrícola como as linhas de plantio. Flexibilidade para navegação autônoma em diferentes tipos de cultura e fases de cultura, permitindo que sub-sistemas de percepção, controle e atuação sejam inseridos ou removidos em acordo com as necessidades de cada operação ou para possibilitar o teste de novos dispositivos e metodologias

relacionadas com controle de navegação. Adequação à operação sobre rede CAN segundo a norma ISO 11783.

Para atender essas necessidades listadas, é proposta uma arquitetura híbrida para guiagem e navegação autônomas de RAM's e VAA's composta pelos seguintes módulos:

- *Módulo de perceptivo:* extrai informações internas do robô e informações do ambiente através de sensores e alimenta outros módulos da arquitetura;

- *Módulo reativo:* é composto por comportamentos reflexivos difusos ou não que possuem uma hierarquia pré-definida. As recomendações de comportamentos mais simples são submetidas a árbitros que coordenam a ativação desses comportamentos e a composição de comportamentos mais complexos;

- *Módulo de arbitragem*: constituído por um ou mais árbitros baseados em controladores difusos ou não que são alimentados pelo módulo perceptivo para coordenar, em cada instante de amostragem, a ativação de comportamentos segundo contextos de aplicação;

- *Módulo de mapeamento:* utiliza um modelo do robô e os dados provenientes do módulo perceptivo e processados por um Filtro de Kalman para disponibilizar a posição do robô em relação a um sistema de posicionamento em cada instante;

- *Módulo deliberativo:* possui instruções referentes às tarefas pré-definidas para realização de operações. Pode configurar os outros módulos e monitorar a execução de cada tarefa ou experimento. Esse módulo possui funções de

controle de tarefas, assim como ações deliberativas que são em grande parte definidas por operadores humanos.

A Figura 14 ilustra a arquitetura proposta composta pelos módulos descritos.

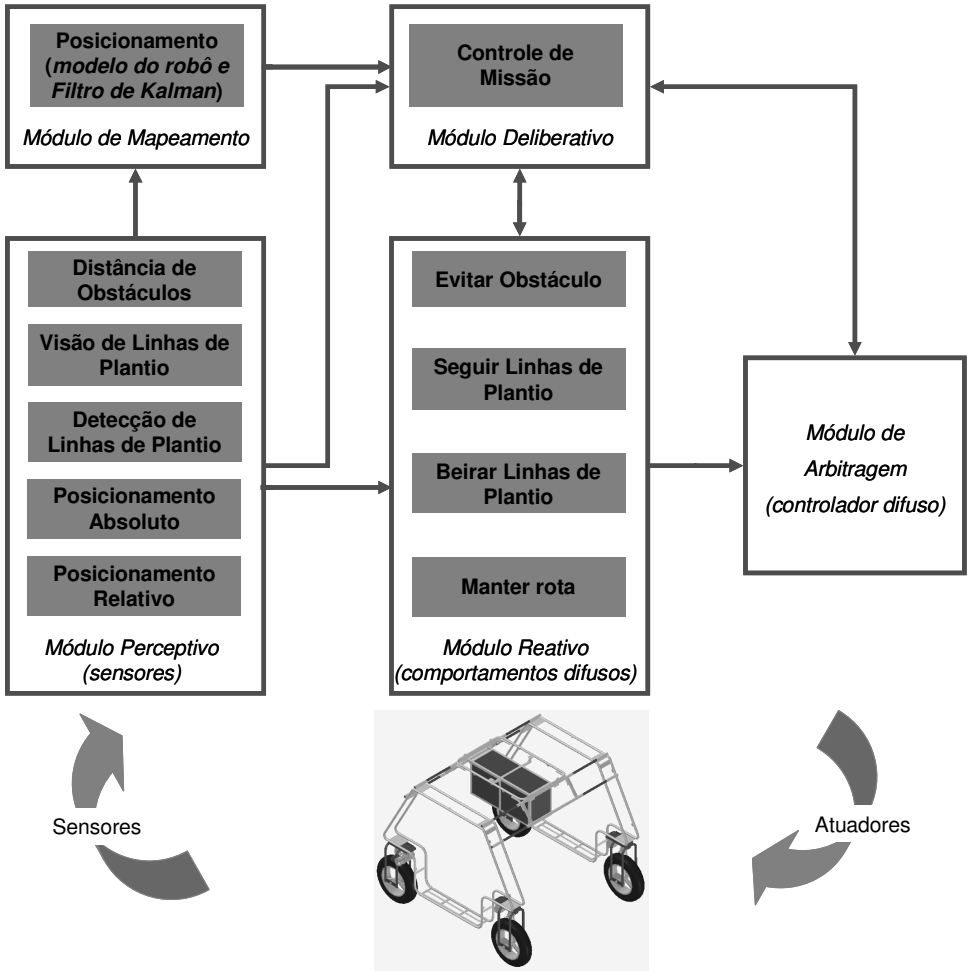


Figura 14 - Arquitetura robótica híbrida baseada em comportamentos para a guiagem e para a navegação autônomas de RAM's e VAA's.

Considerando a pesquisa bibliográfica apresentada no Capítulo 4, em especial a discussão apresentada no Item 4.1 referente aos principais dispositivos de percepção utilizados em máquinas agrícolas, são definidos os seguintes componentes para o módulo de perceptivo da arquitetura híbrida:

- *Distância de obstáculos*: sistema para identificar a proximidade de obstáculos no caminho do robô. Exemplo de sensores mais comuns nos trabalhos pesquisados: sonar e laser;

- *Visão de linhas de plantio*: sistema para identificar e representar virtualmente em um sistema de coordenadas as linhas de plantio para que se possam definir trajetórias para o robô nas ruas entre as linhas de plantio. Exemplo de sensores mais comuns nos trabalhos pesquisados: scanner laser e câmeras. Esse módulo também pode utilizar um mapa previamente construído com coordenadas determinadas por um sistema de posicionamento. Dessa forma, as ruas entre as linhas de plantio estão representadas virtualmente e o robô pode seguí-las através da comparação das coordenadas do mapa com coordenadas tomadas durante a navegação por um sistema de posicionamento como, por exemplo, o GPS;

- *Detecção de linhas de plantio*: sistema para identificar a distância lateral das linhas de plantio ao longo das ruas da plantação. Exemplo de sensores mais comuns nos trabalhos pesquisados: sonar e laser;

- *Posicionamento relativo*: sistema baseado em dispositivos de posicionamento com referência externa. Exemplo de sensores mais comuns nos trabalhos pesquisados: GSP, GDS e sistema triangulação a laser;

- *Posicionamento absoluto (também denominado Dead-Reckoning)*: dispositivos de medidas internas de deslocamento que associadas ao modelo cinemático do robô compõem um sistema de posicionamento Exemplo de sensores mais

comuns nos trabalhos pesquisados: contadores de pulsos ou de giros (odômetros ou *encoders*), acelerômetros e giroscópio.

Baseando-se na revisão de trabalhos científicos apresentada nos Itens 3 e 4, especificamente nas discussões apresentadas nos Itens 3.3 e 4.2 referentes às características de arquitetura híbridas sistematizadas na Tabela 1, são definidos os seguintes comportamentos difusos para a arquitetura híbrida para guiagem e navegação autônomas de RAM's e VAA's:

- *Evitar obstáculos*: comportamento que utiliza sistema medida de distância de obstáculos para identificar obstáculos ou interromper missão em casos em que não encontrar caminhos alternativos;

- *Beirar linhas de plantio*: comportamento que utiliza sistema de detecção de linhas de plantio para navegar nas ruas entre linhas de plantio orientando-se lateralmente pelas próprias linhas;

- *Seguir linhas de plantio*: comportamento que utiliza um sistema de identificação e representação virtualmente em um sistema de coordenadas das linhas de plantio para definir trajetórias nas ruas entre linhas de plantação;

- *Manter rota*: comportamento responsável por manter o robô deslocando-se em frente ou em uma trajetória entre dois pontos definidos.

Para a arquitetura proposta e ilustrada na Figura 14, define-se que os comportamentos robóticos são compostos por controladores difusos para navegação que contêm bases de regras e variáveis lingüísticas associadas à guiagem e navegação. De forma similar aos trabalhos listados na Tabela 1, especialmente

Seraji e Howard (2002), as regras lingüísticas de navegação possuem o formato definido pela relação 9 no Item 3.3. Segundo essa relação, após o processo de fuzzificação, os controladores difusos de navegação avaliam as leituras do sistema sensorial do robô diante das condições "*C*", para então, associar essas leituras a ações "*A*". Posteriormente, as ações "*A*" formadas por conjuntos difusos de saída são submetidas ao processo de defuzzificação através do método centro de gravidade, e dessa forma, cada comportamento gera uma recomendação de atuação.

A ativação de cada comportamento é definida pelo módulo de arbitragem ilustrado na Figura 14, que pode conter um ou mais árbitros. Os árbitros difusos componentes desse módulo utilizam as regras lingüísticas com o formato definido pela relação 10 no Item 3.3. Esses árbitros, após o processo de fuzzificação, avaliam as leituras do sistema sensorial do robô diante das situações "*S*", para então, associar essas leituras a fatores de peso "*W*". Posteriormente, esses fatores formados por conjuntos difusos de saída são submetidos a processo de defuzzificação através do método centro de gravidade, que seleciona um dos comportamentos em cada instante de arbitragem.

A Figura 15 exemplifica a relação entre o módulo reativo e o módulo de arbitragem para coordenação de comportamentos.

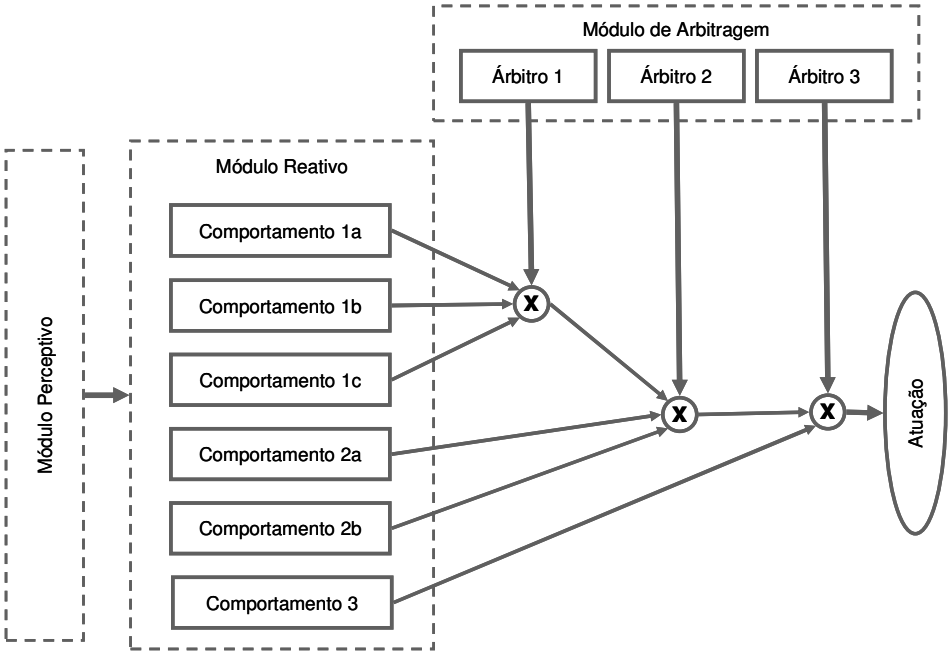


Figura 15 - Exemplo de composição e integração do módulo reativo e do módulo de arbitragem da arquitetura híbrida proposta.

No exemplo ilustrado na Figura 15, o módulo reativo contém seis comportamentos e o módulo de arbitragem contém três árbitros. O árbitro 1 executa a coordenação dos comportamentos 1a, 1b e 1c originando um comportamento de complexidade intermediária. O árbitro 2 executa a coordenação desse comportamento intermediário com os comportamentos 2a e 2b gerando um comportamento complexo. Esse comportamento complexo e o comportamento 3 são coordenados pelo árbitro 3 que define os comandos para os sistemas de atuação.

5.2. Plataforma robótica para simulação e análise da arquitetura proposta

Na literatura científica pesquisa e discutida anteriormente no Capítulo 4, pode-se encontrar trabalhos que buscam adaptar máquinas agrícolas comerciais para

torná-las plataformas agrícolas autônomas (veículos autônomos ou robôs móveis) e, também, trabalhos em que plataformas são construídas especificamente para serem veículos autônomos ou robôs agrícolas. Além disso, como pode ser constatado nos Itens 3.3 e 4.2, há trabalhos de pesquisa em que robôs menores são utilizados como ferramentas de simulação para a pesquisa de novas arquiteturas baseadas em algoritmos de inteligência computacional, incluído arquiteturas baseadas em comportamento difuso.

Com a finalidade de experimentar a arquitetura híbrida proposta, esta foi implementada e testada em uma plataforma robótica baseada no mini-robô comercial denominado Khepera (K-TEAM, 2007). Dessa forma, foi possível realizar simulações para análise e validação de algumas características e de alguns dos requisitos almejados para a arquitetura, explicitados anteriormente nesse mesmo Capítulo 5.

Conforme a proposta para a arquitetura robótica híbrida para guiagem e navegação autônomas (Item 5.1 e Figura 14), foram implementados os módulos propostos com as seguintes características:

- *Módulo de perceptivo:* utilizaram-se os sensores do robô, que são: contadores de pulsos (*encoders*) para medida de deslocamento e sensores de infravermelhos para medida de distância de objetos. Também, foram montados um par de resistores dependentes de luz e uma bússola eletrônica com saída digital;

- *Módulo reativo baseado em comportamentos difusos:* foram implementados os comportamentos difusos definidos para a arquitetura. Os comportamentos *evitar obstáculo* e *beirar linhas* utilizam dados relacionados com os sensores de

infravermelho. O comportamento *seguir linha* utiliza dados relacionados com o par de resistores dependentes de luz, sendo que esses sensores são utilizados para simular um sistema identificação das ruas entre linhas de plantio para seguí-las. O comportamento *manter rota* comportamento natural, ativado quando os outros comportamentos não estão atuando;

- *Módulo de arbitragem*: constituído por um controlador difuso alimentado pelo sistema de percepção para coordenar em cada instante de amostragem a ativação de comportamentos segundo contextos de aplicação;

- *Módulo de mapeamento:* foi desenvolvido para posicionamento absoluto utilizando um modelo do robô e os dados provenientes de um Filtro de Kalman. Esse módulo é alimentado por dados dos sensores de odometria (encoders) e da bússola eletrônica adaptada ao robô;

- *Módulo deliberativo:* foi implementado um módulo simples que permite definir alguns parâmetros para os experimentos realizados, configurar os outros módulos e monitorar a execução de cada experimento. Esse módulo possui funções de controle de tarefas, assim as ações deliberativas são em grande parte tomadas pelo operador.

Detalhes sobre a plataforma robótica construída e sobre a implementação da arquitetura são apresentados nos subitens a seguir.

5.2.1 Plataforma khepera

A arquitetura híbrida proposta anteriormente no Item 5.1 foi implementada e testada em uma plataforma robótica baseada no mini-robô comercial denominado Khepera (K-TEAM, 2007). Esse mini-robô contém um módulo básico com duas rodas com propulsão fornecida por dois motores de corrente contínua acoplados a cada roda por engrenagens redutoras com redução 25:1. No eixo de cada motor é acoplado um sensor contador de pulsos (*encoder*) incremental de 24 pulsos por revolução, que totalizam 600 pulsos por revolução de cada roda e 12 pulsos por milímetro de deslocamento linear. Além do encoder, o módulo básico contém oito sensores de infravermelho (IR) ativos (emissor-receptor), modelo SFH900-2 Siemens, sendo dois frontais, quatro laterais e dois traseiros. Os sinais dos sensores IR são digitalizados e disponibilizados por conversores analógico-digital (AD) de 10 bits. A Figura 16 mostra as vistas lateral e superior do módulo básico do mini-robô.

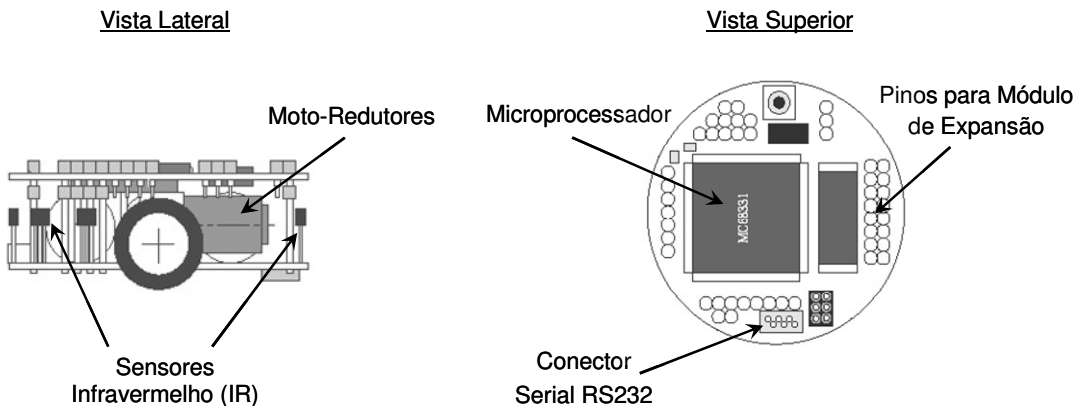


Figura 16 - Vistas lateral de superior do módulo básico do mini-robô Khepera. Fonte: adaptado de K-TEAM (2007).

Como pode ser visto na Figura 16, o mini-robô possui um microprocessador embarcado e uma interface de comunicação padrão RS232. Esse microprocessador faz parte de um sistema computacional com um sistema operacional proprietário que disponibiliza comandos para comunicação com todos os módulos sensores e atuadores do robô. Dessa forma, além da possibilidade de se embarcar o programa computacional de controle do robô, também é possível utilizar a interface RS232 para comunicação entre o robô (sistema operacional do robô) e um computador externo com programa computacional para controle do robô. A comunicação de dados é realizada por mensagens formadas por caracteres que obedecem ao padrão de codificação ASCII (*American Standard Code for Information Interchange*).

O mini-robô possui barramento com pinos para extensão da estrutura através da incorporação de outras placas de circuitos padrão disponibilizadas pelo fabricante. A plataforma montada para a simulação utilizou um módulo de extensão com interfaces de entrada e saída digitais, além de entradas analógicas (conversor AD e amplificadores ajustáveis) e comparadores. A esse módulo, denominado *Khepera IO Turret*, foram conectados outros dois tipos de sensores. Foram adaptados ao robô dois sensores resistores dependentes de luz (RDL - *Light Dependant Resistor*), modelo VT935G, montados na parte frontal inferior do robô a uma distância aproximada de 0,004 m do solo. Além disso, foi conectada uma bússola eletrônica, modelo Vector 2X Compass, fabricada pela empresa Precision Navigation Inc. (PNI, 2007). A bússola foi montada em uma placa padrão (placa mil furos) e conectada ao módulo com entradas e saídas digitais e entradas analógicas (módulo *Khepera IO Turret*). A Figura 17 mostra uma ilustração da plataforma robótica montada e um foto desse conjunto.

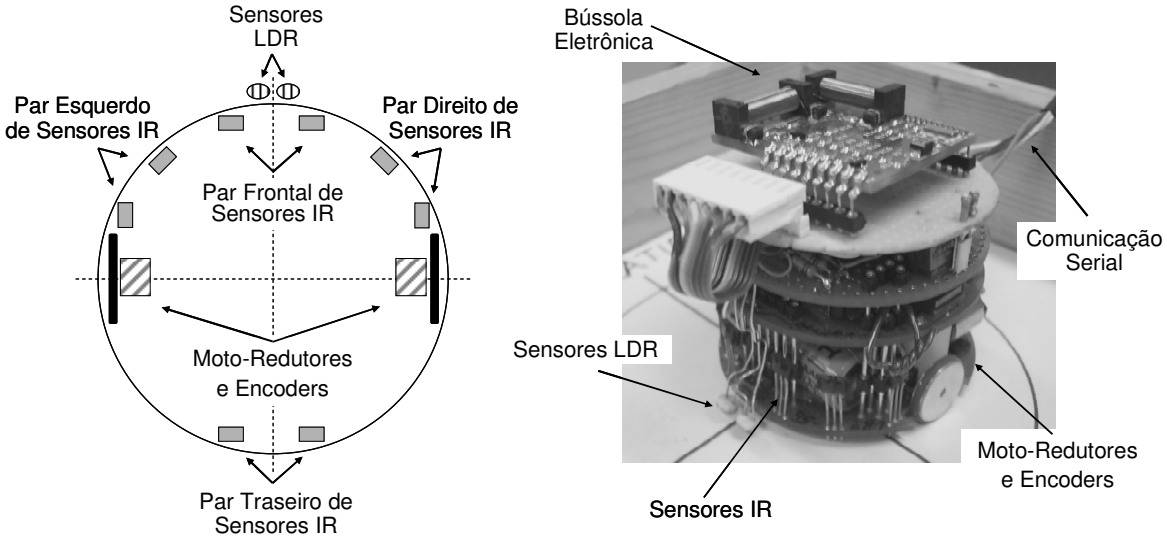


Figura 17 - Plataforma robótica baseada no mini-robô Khepera.

Todas as rotinas de programa foram desenvolvidas e executadas em um computador externo conectado à plataforma robótica pela interface serial operando a 38400 bit/s. As rotinas de programa foram desenvolvidas no ambiente de desenvolvimento de programa Labview da empresa National Instruments (NI, 2007). Além do módulo básico do programa Labview, foi utilizada uma biblioteca complementar denominada *Fuzzy Logic Toolkit*, com funções (Instrumentos Virtuais ou VI's – *Virtual Instruments*) específicas para desenvolvimento de aplicações com lógica difusa. Também, foram utilizadas funções disponibilizadas pelo fabricante do mini-robô Khepera para comunicação com troca de mensagens de baixo nível para operação de todos os sensores e atuadores.

Como citado anteriormente neste mesmo item (Item 5.2), foi desenvolvido um modelo cinemático do mini-robô de duas rodas utilizado (SIEGWART e NOURBAKHSH, 2004). O modelo é representado pelas equações 20 a 24.

$$x(\mathrm{k+1}) = x(k) + ds(k) \cdot \cos\left(\theta(k) + \left(\frac{d\theta(k)}{2}\right)\right) \tag{20}$$

$$y(\mathrm{k+1}) = y(k) + ds(k) \cdot \sin\left(\theta(k) + \left(\frac{d\theta(k)}{2}\right)\right) \tag{21}$$

$$\theta(k+1) = \theta(k) + d\theta(k) \tag{22}$$

$$dD(k) = \frac{dD_D(k) + dD_E(k)}{2} \tag{23}$$

$$d\theta(k) = \frac{dD_D(k) - dD_E(k)}{w} \tag{24}$$

Onde: *k* é o instante de amostragem; *x* , *y* é a posição do robô no sistema de coordenadas plano definido para o modelo (eixos XY); θ é o ângulo entre o eixo de simetria do robô e o eixo X; $d\theta$ é a variação do ângulo θ entre os instantes k e k+1 (período de amostragem); *ds* é o deslocamento do robô entre os instantes k e k+1; dD_D *e* dD_E são os deslocamentos lineares das rodas direita e esquerda respectivamente entre os instantes k e k+1; *w* é a distância entre as duas rodas.

Os dados dos odômetros alimentam o modelo, que por sua vez fornece a cada iteração a posição no plano (x,y) tangente às rodas e a orientação em relação à posição inicial do mini-robô.

5.3. Módulo reativo – comportamentos difusos

Como citado anteriormente nesse mesmo Capítulo 5, o módulo reativo é composto por quatro comportamentos primitivos construídos utilizando Lógica Difusa: *evitar obstáculo*, *seguir linha*, *beirar linhas* e *manter rota*. Esses comportamentos, combinados pelo módulo de arbitragem constituem um comportamento complexo de navegação. De acordo com a proposta, regras difusas foram utilizadas para compor os comportamentos da arquitetura híbrida implementada no mini-robô Khepera. Os comportamentos foram definidos considerando sua aplicação em um RAM para navegar por ruas de plantação com orientação pelas linhas de plantio.

Os comportamentos *evitar obstáculo* e *beirar linha* utilizam os dados lidos com os sensores de IR. Os seis sensores frontais foram agrupados em três pares: par frontal, par esquerdo e par direito. Em cada instante de amostragem o par que possuir leitura de objeto indicando proximidade maior é escolhido como entrada dos controladores difusos que implementam os comportamentos. Esses comportamentos são utilizados para navegação do mini-robô evitando choque com obstáculos frontais (*evitar obstáculo*) e para manter uma distância segura em relação a objetos laterais (*beirar linhas*).

O comportamento *seguir linha* utiliza dados dos dois sensores RDL para simular um sistema identificação e representação virtual das ruas entre as linhas de plantio. Como constatado pela literatura científica e descrito no Item 4.1, os sistemas de visão são utilizados em sistema de navegação de máquinas agrícolas, fornecem o deslocamento lateral (d) e o ângulo de guiagem (g), obtidos da representação das

ruas entre as linhas de plantio em relação a um sistema de coordenada fixo no robô ou veículo autônomo, similar ao mostrado anteriormente na Figura 12. Os RDL's foram utilizados para o mini-robô móvel seguir linhas pintadas no piso do labirinto fornecendo variações de resistência segundo a intensidade de luz refletida pelo piso com linha pintada ou sem linha. Com os dois sensores RDL's é possível obter a diferença de intensidade entre leituras e o valor mínimo das leituras a uma determinada taxa de aquisição de dados. A Figura 18 ilustra o par de RDL's montados no mini-robô e a utilização desses sensores para seguir linhas.

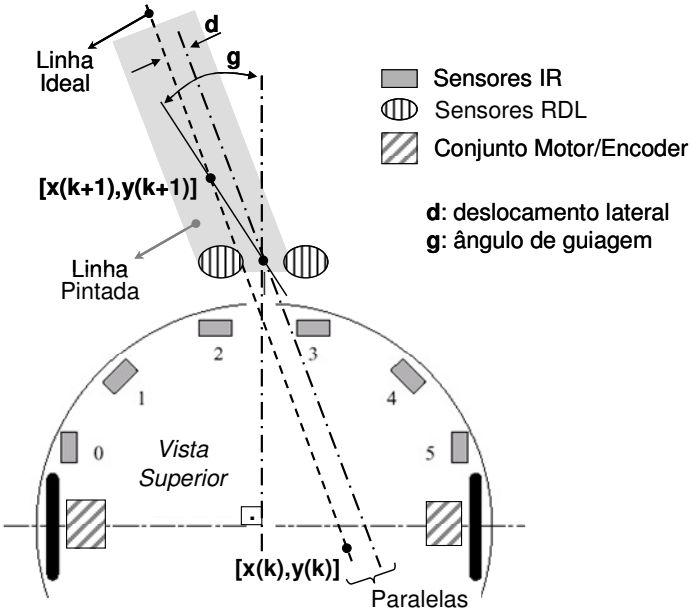


Figura 18 - RDL's montados no mini-robô Khepera para seguir linhas.

Como ilustrado na Figura 18, a diferença de intensidade entre a leitura dos sensores relaciona-se com a distância ou deslocamento em relação à linha pintada que está sendo seguida; o valor mínimo de leitura fornece a direção do robô em relação à linha. Dessa forma, o deslocamento lateral (d) fornecido é simulado pela

diferença de intensidade dos RDL's e o ângulo de guiagem (g) é simulado pelo valor mínimo das leituras dos RDL's.

O comportamento *manter rota* é o comportamento ativado enquanto outros comportamentos não estão atuando, e que simplesmente mantém o robô navegando a uma velocidade constante em linha reta.

Para compor os comportamentos foi utilizado o conjunto de ferramentas para desenvolvimento de controladores com Lógica Difusa disponíveis em um dos módulos de desenvolvimento do Labview denominado *Real Time Toolset* que contém o *Fuzzy Logic Tookit*.

5.3.1 Variáveis lingüísticas associadas aos comportamentos

Considerando os sensores disponíveis e os comportamentos propostos para a arquitetura robótica, foram definidos os valores crisp normalizados de entrada e saída, os termos linguísticos (termos difusos) relacionados os valores crisp, as variáveis lingüísticas e as respectivas funções de pertinência dessas variáveis. Formas triangulares e trapezoidais foram utilizadas para compor as funções de pertinência que foram ajustadas durante os experimentos que serão descritos no Item 5.6.

Para cada par de IR, duas entradas crisp (valores reais de entrada de um sistema difuso) são definidas: $DIST_G$ e DIF_G (G é o índice que representa o par frontal, esquerdo e direito). A variável DIF_G é determinada pela diferença instantânea entre as leituras normalizadas dos sensores componentes de cada par. A variável

$DIST_G$ é determinada pelo maior valor das leituras normalizadas dos sensores componentes de cada par. Dessa forma, os valores crisp relacionados com as leituras dos sensores de IR são determinados pelas equações 25 e 26.

$$DIST_G = \max\left(IR_E^G, IR_D^G\right) \tag{25}$$

$$DIF_G = IR_E^G - IR_D^G \tag{26}$$

Onde: IR_i são as leituras normalizadas dos sensores de infravermelho, sendo que: *G* é o índice do par de IR (frontal, direito e esquerdo); *E* e *D* são os índices que representam o sensor esquerdo e o sensor direito do par G.

O valor $DIST_G$ indica a distância do objeto detectado e qual par de sensores (G: frontal, esquerdo e direito) está mais próximo do objeto (intensidade e direção). O valor DIF_G denota qual sensor do par está mais perto do obstáculo, ou seja, a direção dos espaços livres e ocupados. Os comportamentos *evitar obstáculo* e *beirar linhas* utilizam as entradas crisp, DIF_G e $DIST_G$ e são relacionadas respectivamente a três termos linguísticos (termos difusos): longe (LONGE), médio (MED) e perto (PERTO) para $DIST_G$; e negativo (NEG), zero (ZERO) e positivo (POS) para DIF_G. A partir dos valores crisp e dos termos lingüísticos associados foram definidas quatro variáveis lingüísticas: *IR Direito-DIF*, *IR Direito-DIST*, *IR Esquerdo-DIF* e *IR Esquerdo-DIST*. As funções de pertinência dessas variáveis lingüísticas são apresentadas na Figura 19.

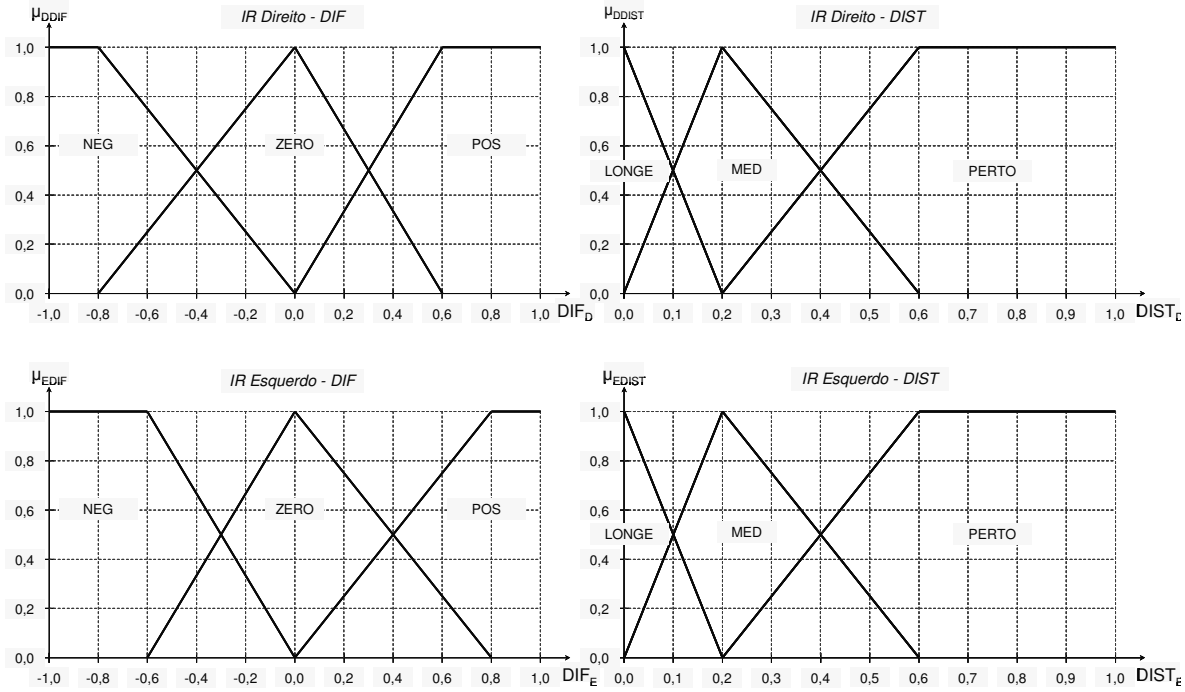


Figura 19 - Variáveis linguísticas utilizadas pelos comportamentos *Evitar Obstáculo* e *Beira Linhas*

As variáveis linguísticas *IR Direito-DIF* e *IR Direito-DIST* mostradas na Figura 19, referem-se às leituras do par quando o sensor direito do par está mais perto do obstáculo. Por outro lado, as variáveis linguísticas *IR Esquerdo-DIF* e *IR Esquerdo-DIST* referem-se às leituras do par quando o sensor esquerdo do par está mais perto do obstáculo. As funções de pertinência dessas variáveis foram ajustadas durante os experimentos que serão descritos no Item 5.6.

Para o par de LDR, duas entradas crisp (valores reais de entrada de um sistema difuso) são definidas: $DIST_{LDR}$ e DIF_{LDR}. A variável DIF_{LDR} é determinada pela diferença instantânea entre as leituras normalizadas dos sensores componentes do par. A variável $DIST_{LDR}$ é determinada pelo menor valor normalizado das leituras dos sensores componentes do par. Dessa forma, os valores crisp relacionados com as leituras dos sensores LDR são determinados pelas equações 27 e 28.

$$DIST=\min\left(LDR_E, LDR_D\right) \tag{27}$$

$$DIF=LDR_E-LDR_D \tag{28}$$

Onde: $LDR_{E,D}$ são as leituras normalizadas dos sensores LDR, que *E* e *D* são os índices que representam o sensor esquerdo e o sensor direito do par.

Para determinação dos valores normalizados LDR_E e LDR_D é realizado um processo de calibração que busca também compensar os efeitos de variação de luminosidade ambiental. O processo de calibração é realizado antes da operação normal do robô. Nesse processo ele é colocado por alguns segundos sobre o solo em que ele deve operar e sob as condições de iluminação do ambiente de operação. Nesse processo são tomadas as leituras mínimas (sobre a linha pintada a ser seguida) e máximas (sem a linha pintada) dos sensores durante um determinado intervalo de tempo. Calcula-se então, a média dessas leituras que são utilizadas para normalização. Através desse processo, as variações de luminosidade referentes ao tipo de material do solo e da iluminação do ambiente são compensadas. Os valores LDR_E e LDR_D são calculados pela normalização das leituras dos sensores LDR de acordo com a equação 29.

$$LDR_i=\frac{R^i-R^i_{MIN}}{R^i_{MAX}-R^i_{MIN}} \tag{29}$$

Onde: *i* é o índice que corresponde a E para o sensor esquerdo e D para o sensor direito; LDR_i é leitura normalizada do sensor *i*; R^i são as leituras do sensor *i* durante a navegação do mini-robô; R^i_{MAX} é a maior leitura do sensor *i* (ou a média) durante o

processo de calibração (linha pintada); e R^i_{MIN} é a menor leitura do sensor *i* (ou a média) durante o processo de calibração (solo sem linha).

O valor $DIST_{LDR}$ indica quão distante o par LDR está distante da linha (intensidade), e o valor DIF_{LDR} denota qual dos sensores do par LDR está mais distante da linha (direção). As entradas crisp para o comportamento *seguir linha* são relacionadas respectivamente com três termos linguísticos: longe (LONGE), médio (MED) e perto (PERTO) para $DIST_{LDR}$, e negativo (NEG), zero (ZERO) e positivo (POS) para DIF_{LDR}. A partir dos valores crisp e dos termos lingüísticos associados foram definidas duas variáveis lingüísticas: *LDR-DIF* e *LDR-DIST*. As funções de pertinência dessas variáveis lingüísticas são apresentadas na Figura 20.

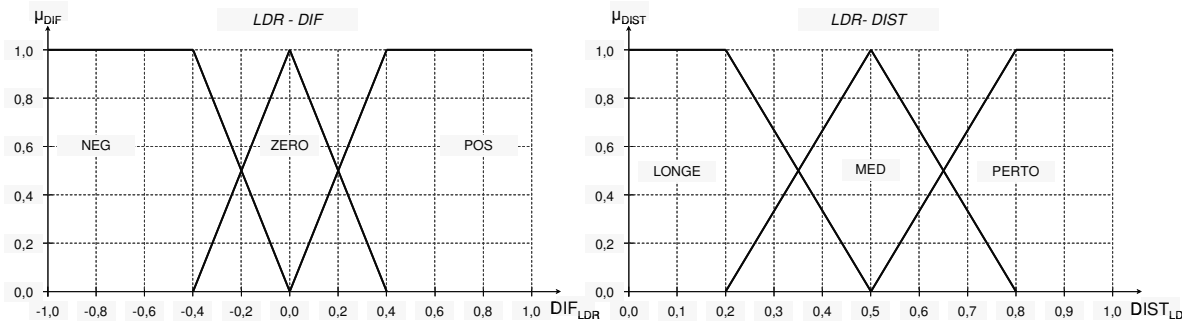


Figura 20 - Variáveis linguísticas utilizadas pelo comportamento *Seguir Linha.*

Os valores C_{MRD} e C_{MRE} referem-se aos comandos de velocidade (valores crisp normalizados) para os dois motos-redutores da plataforma Khepera, sendo *MRD* para o moto-redutor direito e *MRE* para o moto-redutor esquerdo. Esses valores crisp são relacionados a três termos linguísticos utilizados para descrever as saídas: frente (F_i), parado (P_i) e trás (T_i), sendo que *i* é o índice que corresponde a E para moto-redutor esquerdo e D para moto-redutor direito. A partir dessas dos

valores crisp e dos termos lingüísticos associados foram definidas duas variáveis lingüísticas: *Moto-Redutor Direito* e *Moto-Redutor Esquerdo*. As funções de pertinência dessas variáveis linguísticas são apresentadas na Figura 21.

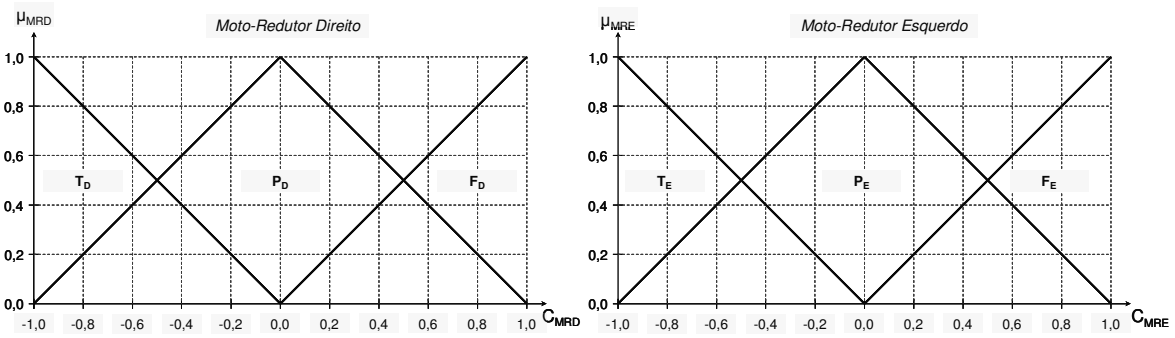


Figura 21 - Variáveis linguísticas relacionadas com o acionamento dos motos-redutores do mini-robô

Observa-se pela Figura 21 que os valores crisp de saída normalizada variam entre -1,0 e 1,0, sendo que $-1,0 \leq C_i < 0$ corresponde a rotação para trás, $0 < C_i \leq 1,0$ corresponde a rotação para frente e $C_i = 0$ corresponde ao estado parado (sendo i: MRE para moto-redutor esquerdo e MRD para moto-redutor direito). Dessa forma, o controle de guiagem é realizado pelo acionamento diferencial das duas rodas (SIEGWART e NOURBAKHSH, 2004), ou seja, as duas rodas podem ser acionadas com a mesma velocidade para que o robô desloque-se em trajetória retilínea para frente ou para trás, ou as duas rodas podem ser acionadas com velocidades diferentes para que o robô desloque-se em trajetória curvilínea para frente ou para trás.

5.3.2 Bases de regras componentes dos comportamentos

Como descrito no Item 3.2 e ilustrado na Figura 7, os valores das variáveis exatas lidas pelos sensores do robô (valores crisp) são associados a valores difusos referentes às variáveis linguísticas do módulo fuzzificador. Esses valores difusos alimentam uma máquina de inferência, que avalia esses valores através de implicações definidas pela base de regras, deduzindo assim, novos fatos ou novos valores difusos a partir de informações de entrada.

Cada um dos comportamentos difusos implementados contém um conjunto de controladores difusos com saídas que comanda a guiagem e navegação do mini-robô. Os controladores difusos possuem regras difusas que foram implementadas de acordo com a relação 9 apresentada no Item 3.3 e definida por Seraji e Howard (2002). Essas regras constituem regras de guiagem e navegação que associam uma condição a uma ação, sendo que ambas são representadas por expressões linguística compostas por uma ou mais variáveis associadas pelo processo de composição, através de operados como, por exemplo, união, intersecção e complemento (detalhes teóricos são postos no Item 3.1).

As bases de regras para cada comportamento são compostas por dois grupos de regras, um para comando do moto-redutor esquerdo e outro para o moto-redutor direito. Esses dois grupos formam um conjunto denominado *Base de Conhecimento*. Assim, um comportamento é formado por uma ou mais bases de conhecimento, que por suas vez, são formados por dois grupos de regras.

O comportamento *evitar obstáculo* contém uma base de conhecimento que relaciona, por meio do operador de união, os conjuntos difusos LONGE, MED e

PERTO e os conjuntos difusos NEG, ZERO e POS referentes às entradas crisp, DIF_G e $DIST_G$, com os conjuntos difusos F_i, P_i e T_i referentes às saídas C_{MRE} e C_{MRD}. A Figura 22 mostra a base de conhecimento do comportamento *evitar obstáculo*.

Evitar Obstáculo

		$DIST_G$		
		LONGE	MED	PERTO
DIF_G	NEG	F_EF_D	P_EF_D	T_EF_D
	ZERO	F_EF_D	P_EP_D	T_ET_D
	POS	F_EF_D	F_EP_D	T_EF_D

Figura 22 - Base de conhecimento para o comportamento *evitar obstáculo*.

Como exemplo de interpretação da base de conhecimento mostrada Figura 22, considera-se a célula correspondente a interseção da linha NEG com a coluna PERTO. Com esses elementos têm-se duas regras:

- **se** *DIST é PERTO* **e** *DIF é NEG* **então** C_{MRE} *é* T_E (sob os conceitos da teoria de conjuntos difusos pode-se interpretar essa regra como: “se distância do objeto é pequena e o objeto está mais à direita então a velocidade é da roda esquerda para trás”)

- **se** *DIST é PERTO* **e** *DIF é NEG* **então** C_{MRD} *é* F_D (sob os conceitos da teoria de conjuntos difusos pode-se interpretar essa regra como: “se distância do objeto é pequena e o objeto está mais à direita então a velocidade da roda direita é para frente”)

No caso descrito pelas regras anteriores, o comportamento de *evitar obstáculo* estaria guiando o robô com rotação acentuada para a esquerda para desviar de um objeto próximo à direita.

O comportamento *beirar linhas* contém duas bases de conhecimento que relacionam, por meio do operador de união, os conjuntos difusos LONGE, MED e PERTO e os conjuntos difusos NEG, ZERO e POS referentes às entradas crisp, DIF_G e $DIST_G$, com os conjuntos difusos F_i, P_i e T_i referentes às saídas C_{MRE} e C_{MRD}. Cada uma das bases de conhecimento correspondem a um sub-comportamento do comportamento *beirar linhas*, sendo um denominado *beirar linha esquerda* e o outro denominado *beirar linha direita*. A Figura 23 mostra as bases de conhecimento do comportamento *beirar linhas*.

Beirar Linha Esquerda

		$DIST_G$		
		LONGE	MED	PERTO
DIF_G	NEG	F_EF_D	F_EP_D	P_ET_D
	ZERO	F_EF_D	F_EP_D	F_ET_D
	POS	F_EF_D	F_EF_D	F_EP_D

Beirar Linha Direita

		$DIST_G$		
		LONGE	MED	PERTO
DIF_G	NEG	F_EF_D	F_EF_D	P_EF_D
	ZERO	F_EF_D	P_EF_D	T_EF_D
	POS	F_EF_D	P_EF_D	F_ET_D

Figura 23 - Bases de conhecimento para o comportamento *beirar linhas*.

O comportamento *evitar obstáculo* e os sub-comportamentos *beirar linha esquerda* e *beirar linha direita* utilizam os pares de sensores de IR e competem pelo comando dos conjuntos motos-redutores. O par frontal de sensores é utilizado pelo comportamento *evitar obstáculo* e os pares esquerdo e direito são utilizados pelos comportamentos *beirar linha esquerda* e *beirar linha direita* respectivamente. Em cada instante de amostragem o par que possuir leitura de objeto (linha ou obstáculo)

indicando proximidade maior é escolhido como entrada dos controladores difusos e define qual dos comportamentos é ativado. Assim, constitui-se um processo de arbitragem simples de comparação e escolha pelo maior valor de leitura baseado na equação 25, que determina qual dos pares de sensores de IR é utilizado em cada instante de amostragem, e, consequentemente, qual dos comportamentos. Dessa forma, os três comportamentos originam um comportamento mais complexo denominado *identificar objeto* que será detalhado posteriormente no Item 5.4.

O comportamento *seguir linha* contém uma base de conhecimento que relaciona, por meio do operador de união, os conjuntos difusos LONGE, MED e PERTO e os conjuntos difusos NEG, ZERO e POS referentes às entradas crisp, DIF_{LDR} e $DIST_{LDR}$, com os conjuntos difusos F_i, P_i e T_i referentes às saídas C_{MRE} e C_{MRD}. A Figura 22 mostra a base de conhecimento do comportamento *seguir linha*.

Seguir Linha

		$DIST_{LDR}$		
		LONGE	MED	PERTO
DIF_{LDR}	NEG	F_ET_D	F_EP_D	F_EF_D
	ZERO	P_EP_D	F_EF_D	F_EF_D
	POS	T_EF_D	P_EF_D	F_EF_D

Figura 24 - Base de conhecimento para o comportamento *seguir linha*.

Após o processo de fuzzificação, a máquina de inferência utiliza o método *max-min* descrito no Item 3.1 e ilustrado Figura 6 para avaliar as regras de guiagem e navegação e deduzir novos valores difusos a partir de informações de entrada. Posteriormente, esses conjuntos difusos resultantes são submetidos ao processo de defuzzificação.

5.4. Módulo de arbitragem – árbitro difuso

Para realizar a coordenação de comportamentos é utilizado um controlador difuso. Esse controlador funciona como um árbitro que em cada instante de amostragem avalia o contexto de aplicação de cada comportamento nas situações de competência de cada um deles: presença de obstáculo, presença de linha e caminho livre. Formas triangulares e trapezoidais foram utilizadas para compor as funções de pertinência e o método centróide é utilizado para a defuzzificação.

As equações 25 e 27 apresentadas no Item 5.3.2 são utilizadas como entrada para o árbitro difuso para avaliar o nível de aplicabilidade de cada comportamento. A equação 25 permite determinar o valor $DIST_G$ que indica a distância do objeto detectado e a equação 27 permite determinar o valor $DIST_{LDR}$ que indica quão distante o par LDR está distante da linha (intensidade). Assim, esses valores crisp são relacionados a três termos linguísticos utilizados para descrever os comportamentos implementados: objeto longe ($LONGE_O$), objeto médio (MED_O) e objeto perto ($PERTO_O$) para $DIST_G$ e linha longe ($LONGE_L$), linha médio (MED_L) e linha perto ($PERTO_L$) para $DIST_{LDR}$. A partir dos valores crisp e dos termos lingüísticos associados foram definidas duas variáveis linguísticas: *Objeto-DIST e Linha-DIST*. As funções de pertinência dessas variáveis linguísticas são apresentadas na Figura 25.

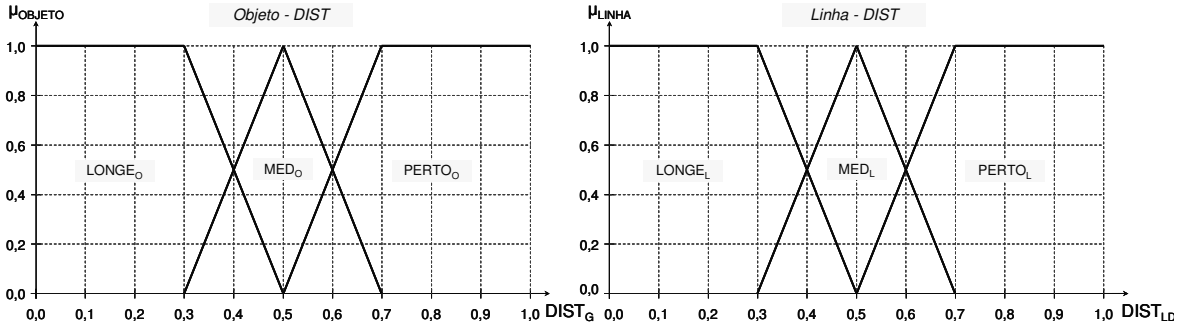


Figura 25 - Variáveis linguísticas relacionadas com o controlador difuso para arbitragem de comportamentos.

O árbitro difuso utiliza a inferência difusa para selecionar os comportamentos em cada instante de aplicação de acordo com o contexto surgidos da interação do robô com o ambiente. Para isso, além variáveis lingüísticas de entrada é definida uma variável de saída que tem como valor crisp o valor S, que indica qual dos comportamentos deve ser ativado em cada instante de arbitragem. Esse valor crisp é relacionado a três termos linguísticos utilizados para descrever a saída: seguir linha (SEGUIR LINHA), manter rota (MANTER ROTA) e identificar objeto (IDENTIFICAR OBJETO). A partir do valor crisp S e dos termos linguísticos associados foi definida uma variável linguística denominada *Comportamento-S*. A função de pertinência dessa variável linguística é apresentada na Figura 26.

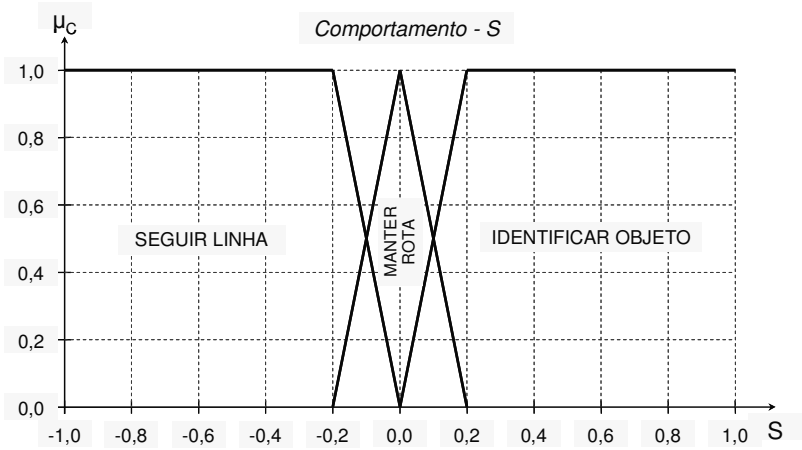


Figura 26 - Variável lingüística de saída relacionada com o controlador difuso para arbitragem de comportamentos.

O árbitro difuso contém uma base de conhecimento que relaciona, por meio do operador de união, os conjuntos difusos $LONGE_O$, MED_O e $PERTO_O$ e os conjuntos difusos $LONGE_L$, MED_L e $PERTO_L$ referentes às entradas crisp, $DIST_G$ e $DIST_L$, com os conjuntos difusos SEGUIR LINHA, MANTER ROTA e IDENTIFICAR OBJETO referentes à saída S. A Figura 27 mostra a base de conhecimento do comportamento do árbitro difuso.

Árbitro Difuso

		$DIST_G$		
		$LONGE_O$	MED_O	$PERTO_O$
$DIST_L$	$LONGE_L$	MANTER ROTA	IDENTIFICAR OBJETO	IDENTIFICAR OBJETO
	MED_L	MANTER ROTA	IDENTIFICAR OBJETO	IDENTIFICAR OBJETO
	$PERTO_L$	SEGUIR LINHA	SEGUIR LINHA	IDENTIFICAR OBJETO

Figura 27 - Base de conhecimento do árbitro difuso.

Como exemplo de interpretação da base de conhecimento mostrada Figura 27, considera-se a célula correspondente a interseção da linha $PERTO_L$ com a coluna $LONGE_O$. Com esses elementos têm-se duas regras:

- **se** *$DIST_L$ é $PERTO_L$* **e** *$DIST_G$ é $LONGE_O$* **então** *S é SEGUIR LINHA* (sob os conceitos da teoria de conjuntos difusos pode-se interpretar essa regra como: "se distância da linha é pequena e o objeto está distante então o comportamento que se apresenta é *seguir linha*").

No caso descrito pela regra anterior, o árbitro identifica que há uma linha a ser seguida e não há nenhum objeto próximo. Assim, diante do contexto apresentado, o comportamento *seguir linha* apresenta-se como a opção mais viável e é recomendada a sua ativação. Nesse caso, o comportamento *manter rota* e o comportamento *identificar objeto* permanecem inativos.

Como outro exemplo de interpretação da base de conhecimento mostrada Figura 27, considera-se a célula correspondente a interseção da linha $PERTO_L$ com a coluna $PERTO_O$. Com esses elementos têm-se a regra:

- **se** *$DIST_L$ é $PERTO_L$* **e** *$DIST_G$ é $PERTO_O$* **então** *S é IDENTIFICAR OBJETO* (sob os conceitos da teoria de conjuntos difusos pode-se interpretar essa regra como: "se distância da linha é pequena e o objeto está próximo então o comportamento que se apresenta é *identificar objeto*, que por sua vez pode ser *evitar obstáculo* ou *beirar linhas*").

Neste outro caso descrito pela regra anterior, o árbitro identifica que há uma linha a ser seguida, porém também há um objeto próximo. Nesse contexto que se apresenta, o comportamento *identificar objeto* apresenta-se como a opção mais

viável e é recomendada a sua ativação. Nesse caso, o comportamento *manter rota* e o comportamento *seguir linha* permanecem inativos.

Os exemplos de interpretação das regras evidenciam que a base de conhecimento apresentada na Figura 27 define uma hierarquia entre os comportamentos, na qual o comportamento *identificar objeto* tem a maior prioridade, o comportamento *seguir linha* tem prioridade intermediária e o comportamento *manter rota* tem a prioridade menor. Essa estrutura hierárquica de controle define a aplicação dos comportamentos em cada instante de amostragem, gerando assim um comportamento mais complexo de guiagem e navegação.

Como descrito anteriormente no Item 5.3.2, o comportamento *identificar objeto* é um comportamento composto por três controladores difusos: *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*. Esses controladores podem ser entendidos como sub-comportamentos, comportamentos primitivos ou comportamentos de baixo nível, que em conjunto compõem um comportamento mais complexo ou um comportamento de alto nível. Como descrito no Item 3.3, essa abordagem (comportamentos hierárquicos) em que comportamentos mais simples são utilizados para construção de comportamentos mais complexos facilita o processo de desenvolvimento e de implementação da arquitetura, principalmente pela significativa diminuição do número de regras (TUNSTEL, LIPPINCOTT e JAMSHIDI, 1997; HAGRAS, CALLAGHAN e COLLEY, 2001).

Após o processo de fuzzificação, a máquina de inferência utiliza o método *max-min* descrito no Item 3.1 e ilustrado na Figura 6 para avaliar as regras componentes do árbitro difuso a partir das informações de entrada. Posteriormente, esse conjunto difuso resultante é submetido ao processo de defuzzificação pelo

método do centróide. O valor crisp de saída indica qual dos comportamentos deve ser ativado, ou qual a saída, entre as saídas dos comportamentos, deve ser utilizada no instante de amostragem para a guiagem do mini-robô.

Como pode ser observado nas descrições deste item (Item 5.4), existem dois árbitros na arquitetura implementada no mini-robô, agora nomeados de árbitro-α e árbitro-β. O árbitro-α define em cada instante sobre a ativação de um dos comportamentos que utilizam os pares de sensores de IR, *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*, através de um processo simples de encolha baseado na maior leitura entre os sensores de IR. O árbitro-β é o árbitro difuso que, através das leituras dos sensores IR e LDR, utiliza uma base de conhecimento para ponderar e decidir sobre qual dos comportamentos deve realizar a guiagem em cada instante no contexto que se apresenta. A Figura 28 mostra um diagrama que ilustra o módulo reativo e o módulo de arbitragem, assim como a integração entre eles.

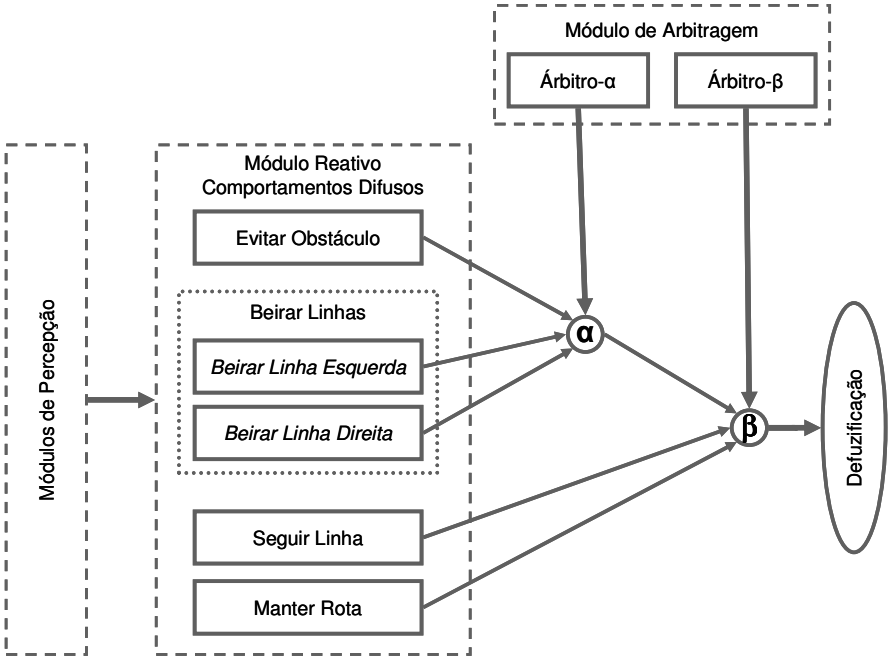


Figura 28 - Composição e integração do módulo reativo e do módulo de arbitragem.

Como pode ser visto na Figura 28, o árbitro-α realiza a primeira seleção (seleção-α) de comportamento, sendo que esse conjunto composto pelos comportamentos *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita* e o árbitro formam um comportamento mais complexo denominado *identificar objeto*. Após essa primeira seleção, o árbitro-β, que é um controlador difuso, seleciona de acordo com o contexto de aplicação (seleção-β), entre os comportamentos *identificar objeto*, *seguir linha* e *manter rota*, o qual destes efetivamente será responsável pela guiagem do robô no instante da amostragem.

O comportamento escolhido no procedimento da Figura 28 é composto por dois controladores difusos, com a estrutura similar à descrita no Item 3.2 e apresentada na Figura 7, sendo um para comando do moto-redutor esquerdo e outro para comando do moto-redutor direito.

5.5. Módulo de mapeamento e controle de tarefas

O módulo de mapeamento utiliza dados dos sensores de odometria (encoders) e da bússola eletrônica adaptada ao robô. Esses dados alimentam o modelo cinemático do mini-robô descrito no Item 5.2 e representado equações 20 a 24 (Item 5.2.1). Esse conjunto fornece a cada iteração (instante de amostragem) a posição no plano (x,y) tangente às rodas e a orientação (θ) em relação a posição inicial do mini-robô.

O processo de posicionamento absoluto baseado em odômetros é também conhecido como odometria. São conhecidos os problemas em tais sistemas relativos a patinagem, a derrapagem e a imperfeições dos odômetros que conduzem a erros cumulativos no processo estimação da posição dos robôs móveis ou veículos autônomos durante a navegação (BORESNTEIN e FENG, 1996). Para minimizar tais problemas muitos trabalhos propõem métodos baseados em fusão de sensores.

Uma das metodologias mais comuns para fusão de sensores é a ferramenta matemática Filtro de Kalman (FK) descrita no Item 4.1. No processo de mapeamento, os dados de sensores combinados através do FK são aplicados ao modelo matemático do sistema móvel. Desta foram, realiza-se a integração das medições de cada sensor compensando as deficiências de cada um e gerando valores finais mais precisos.

Foi implementado um FK para integrar os dados dos odômetros com dados da bússola eletrônica. Detalhes sobre a integração da bússola no mini-robô são apresentados no Item 5.2. O FK foi implementado de forma indireta, ou seja, em vez de se utilizar os valores da bússola e dos odômetros diretamente foram utilizados os valores de erro entre as leituras desses sensores aplicados ao FK. Tal implementação apresenta vantagens em relação à implementação direta como relato em Roumeliotis, Sukhatme e Bekey (1999). Assim, utilizou-se o modelo de erro para odometria apresentado por Sandi, Hemerly e Lages (1997) para obter o erro de orientação dado pela equação 30, que também constitui a equação de estado para o FK.

$$\delta\theta(k+1) = \delta\theta(k) + \omega(k) \tag{30}$$

Onde: δθ(k) e δθ(k+1) são os erros de orientação nos instantes de amostragem k e k+1; ω(k): ruído no processo no instante de amostragem k.

No processo para determinação das coordenadas (x,y) e a orientação (θ), a cada iteração, é calculada a diferença entre a leitura de orientação da bússola ($d\theta_c(k)$) e o valor de orientação calculado pela a equação 24 apresentada no Item 5.2.1 ($d\theta_e(k)$). Esse valor é utilizado pelo FK (equações 15 a 19) aplicado à equação de estado 30 para determinação do valor de correção da orientação ($\hat{\delta\theta}_k$). O diagrama ilustrativo do processo é apresentado na Figura 29.

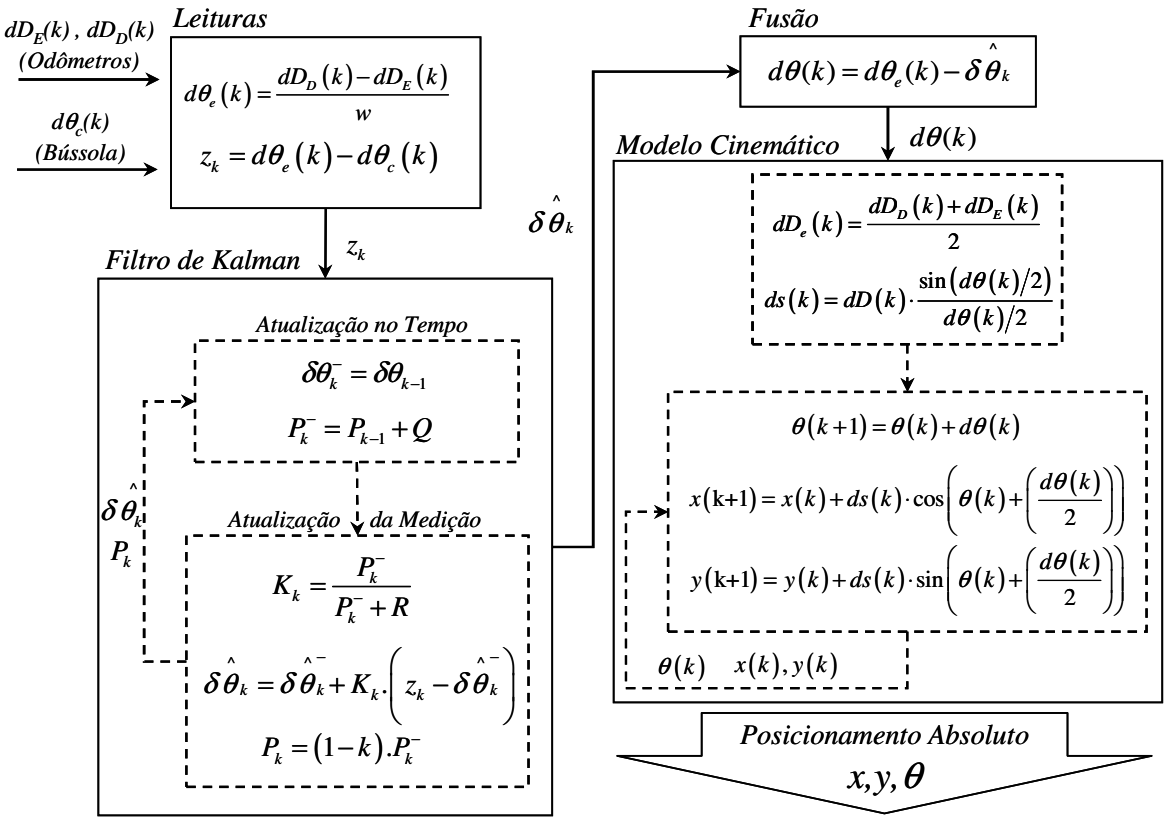


Figura 29 - Filtro de Kalman indireto para fusão de sensores entre os odômetros e a bússola eletrônica.

Como pode ser visto no diagrama da Figura 29, o valor corrigido de orientação (fusão: $d\theta(k)$) é aplicado às equações do modelo cinemático (equações 20 a 24 do Item 5.2.1) para cálculos dos valores finais das coordenadas ($x(k), y(k)$) e a orientação ($\theta(k)$).

Para testes do sistema de posicionamento e experimentos com a arquitetura robótica foi construído um programa em Labview com uma interface gráfica que permite comparar o sistema de posicionamento baseado somente nos odômetros com o sistema baseado na fusão de dados do odômetros e da bússola eletrônica. A Figura 30 mostra a interface gráfica construída.

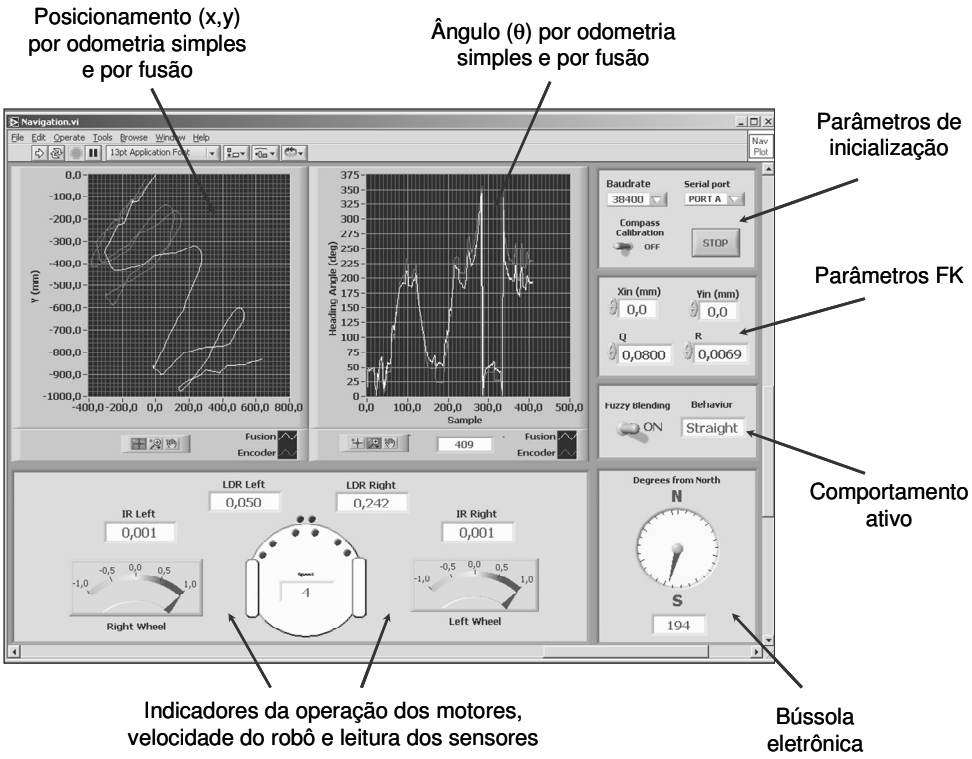


Figura 30 - Aplicativo com interface gráfica para experimentos com odometria.

O aplicativo com interface gráfica apresentada na Figura 30 permite acompanhar a trajetória e a direção de navegação calculada pelo módulo de mapeamento apenas com as leituras dos odômetros (encoders), assim como a trajetória e a direção de navegação calculada pelo módulo de mapeamento utilizando o algoritmo de fusão de sensores pelo FK. Durante a operação do mini-robô é possível monitorar os valores instantâneos de todos os sensores e dos comandos gerados para os motos-redutores, assim como acompanhar a ativação dos comportamentos em cada instante. O aplicativo permite salvar os dados gerados durante os experimentos.

O aplicativo desenvolvido constitui um controlador de tarefas e foi implementado para permitir a definição de alguns parâmetros para os experimentos realizados, configurar os outros módulos da arquitetura e monitorar a execução de cada experimento. Assim as ações deliberativas são em grande parte tomadas pelo operador, não sendo, portanto, funções robóticas da arquitetura.

Inicialmente, o aplicativo foi utilizado em testes para ajustar os parâmetros Q e R do FK, que são respectivamente a covariância do ruído do processo e a covariância do ruído da medição relativo à distribuição normal de probabilidade (equações 16 e 17 do Item 4.1.1). Em testes similares aos realizados em Oksanen, Markus e Visala (2005) e Rovira-Más e Han (2006), o mini-robô foi programado inicialmente para executar uma trajetória semicircular e foram realizados diversos ensaios com ajuste manual de Q e R. Ao mini-robô foi adaptado um pincel para marcar sua trajetória sobre um papel milimetrado colocado sobre a superfície de teste. Durante cada ensaio, a trajetória real do mini-robô foi adquirida a partir do traçado sobre o papel milimetrado e comparada com os valores calculados pelo

módulo de mapeamento baseado somente nos odômetros (encoders) e pelo módulo de mapeamento baseado na fusão de dados do odômetros (encoders) e da bússola eletrônica com o FK. A Figura 31 apresenta um dos gráficos comparativos com a trajetória real e as trajetórias calculadas.

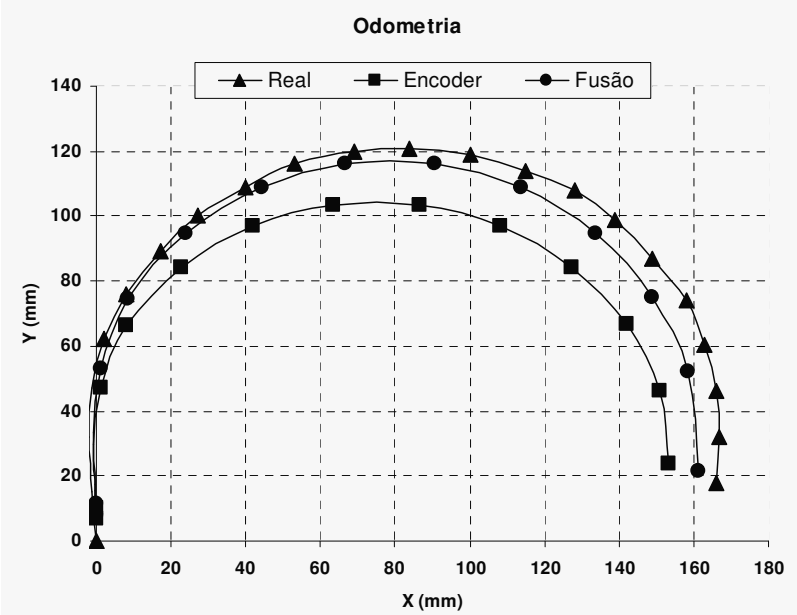


Figura 31 - Curvas experimentais para ajuste do Filtro de Kalman indireto para fusão de sensores entre os odômetros e a bússola eletrônica.

A partir da análise dos gráficos gerados similares ao apresentado na Figura 31, determinou-se os valores para Q e R. Como os experimentos realizados com o mini-robô foram executados em ambientes com pouca variabilidade, os valores de Q e R foram assumidos constantes para todos os experimentos realizados. Uma abordagem mais precisa poderia ser adotada para ajuste dos valores das covariâncias, inclusive assumindo-as variáveis. Porém, para os experimentos realizados com o mini-robô o procedimento adotado foi suficiente para permitir a avaliação da arquitetura robótica proposta como será apresentado nos itens

seguintes. Detalhamentos maiores sobre a influência das covariâncias em sistemas de posicionamento absoluto podem ser obtidos em Boresntein e Feng (1996), Roumeliotis, Sukhatme e Bekey (1999), Sandi, Hemerly e Lages (1997), Welch e Bishop (2002), Oksanen, Markus e Visala (2005) e Rovira-Más e Han (2006).

5.6. Simulação e análise da arquitetura proposta - resultados

Diferentes experimentos foram conduzidos em um labirinto de madeira para avaliação das capacidades do robô de exploração, evitar obstáculo, identificar linhas e seguir linhas. A planta do labirinto é mostrada na Figura 32.

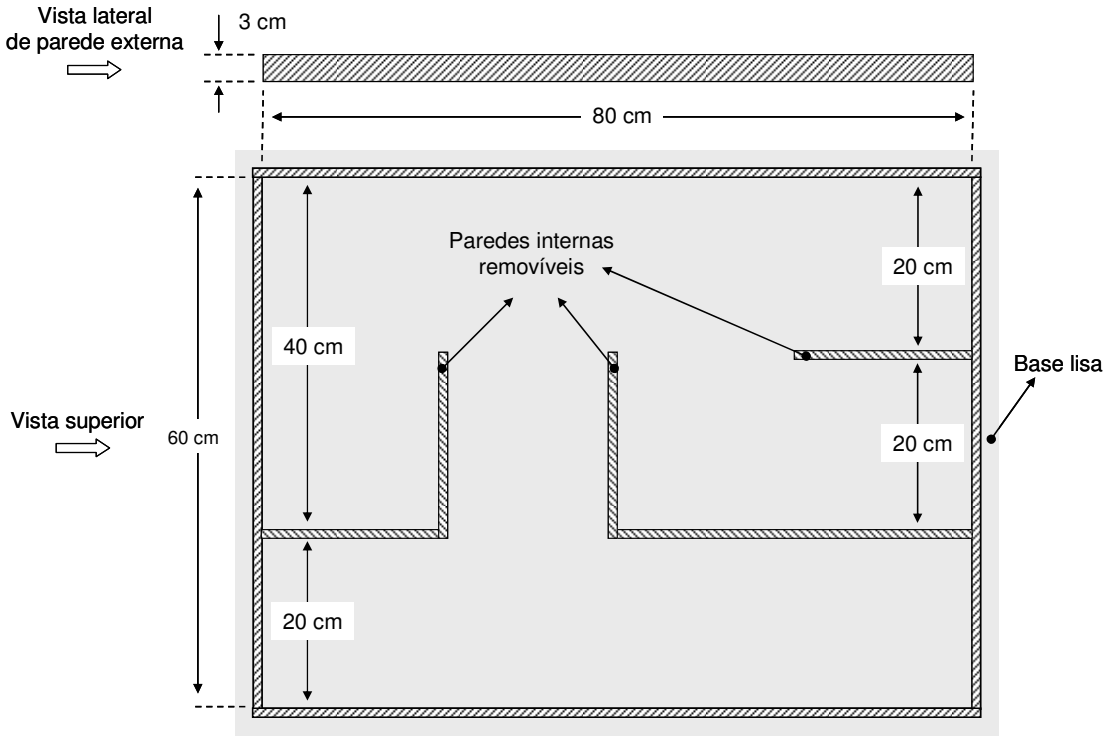


Figura 32 - Planta do labirinto de madeira utilizados nos experimentos com a arquitetura robótica híbrida.

Sob o piso de madeira do labirinto é possível fixar diferentes folhas de papel de acordo com as necessidades de cada experimento. Para os experimentos realizados foi utilizado folha de papel milimetrado e folha de papel tipo cartão (cartolina).

Foram efetuados três tipos de experimentos para avaliação da arquitetura robótica híbrida proposta. Tais tipos de experimentos foram:

- *Seguir linha pintada no piso em um percurso fechado*: esse tipo de experimento foi conduzido não só para avaliar o comportamento *seguir linha* como também para avaliar o módulo de mapeamento (odometria). Um percurso fechado foi pintado sobre o piso com papel milimetrado e alguns pontos com coordenadas em relação à origem foram tomados para mapeamento manual desse percurso e posterior comparação com as coordenadas tomadas por odometria;

- *Exploração de ambiente*: esse tipo de experimento foi conduzido para avaliação dos comportamentos (módulo reativo) e do processo de arbitragem (módulo de arbitragem). Sob o mesmo percurso pintado sobre o piso com papel milimetrado foram inseridos obstáculos (paredes). Neste caso, também foi realizado mapeamento manual do percurso e da posição dos obstáculos para posterior comparação com as coordenadas tomadas por odometria;

- *Navegação em rua entre linhas de plantas*: esse tipo de experimento foi conduzido para avaliar a arquitetura robótica proposta para guiagem e navegação autônomas de um robô agrícola móvel. Três linhas de plantas foram fixadas sobre um papel cartão branco (cartolina) formando dois corredores de entre plantas (ruas). Além disso, alguns trechos de linha foram pintados nesses

corredores sobre o piso, simulando pontos em que dados de um sistema de visão ou de um mapa obtido posteriormente estariam disponíveis.

Diversas rodadas de cada um dos tipos experimentos foram executadas. Os procedimentos e os resultados principais são descritos a seguir.

5.6.1 Guiagem e mapeamento sobre uma trajetória

Para o experimento de guiagem e mapeamento sobre uma trajetória fechada (percurso fechado), foi pintada uma linha em cor preta no piso do labirinto sem barreiras e objetos. O caminho foi pintado com aproximadamente 8 milímetros de largura sobre um papel milimetrado com escala de 1 milímetro. O percurso foi mapeado manualmente, tomando-se 55 (cinqüenta e cinco) do centro da linha pintada para comparação com as coordenadas calculadas por odometria e fusão sensorial através do FK.

A Figura 33 mostra a execução de uma volta, na qual o robô iniciou a navegação na posição (0,0) e seguiu a linha pintada até completar o percurso fechado. Nesse percurso foram tomados 2030 (dois mil e trinta) pontos por odometria pelo módulo de mapeamento.

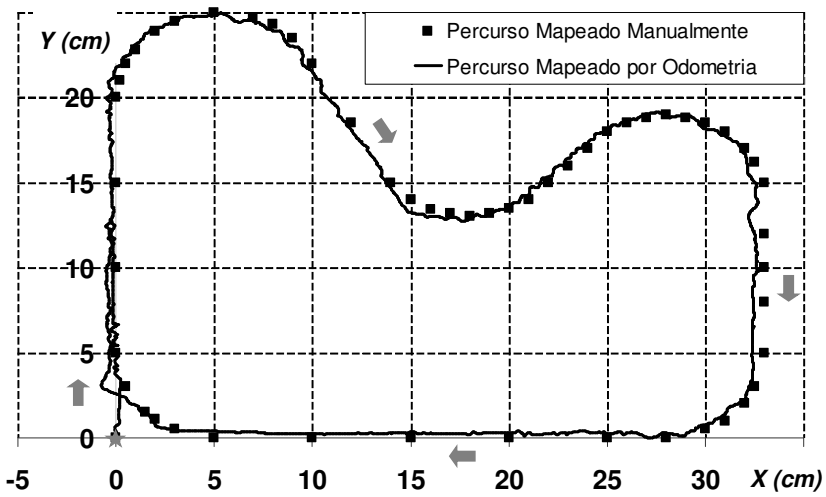


Figura 33 - Percurso fechado realizado pelo robô seguindo linha pintada no piso.

Na Figura 33 são expostos os pontos coletados manualmente e o pontos calculados por odometria em um sistema de coordenadas. O percurso composto pelos pontos discretos corresponde aos pontos coletados manualmente sobre o papel milimetrados. O percurso composto pela linha continua, em verdade, é formada pela interligação dos 2030 pontos referentes às coordenadas obtidas pelo cálculo realizado pelo módulo de mapeamento a partir das leituras realizadas por cada odômetros.

A tarefa de guiagem e mapeamento sobre a trajetória mostrada na Figura 33 utiliza o comportamento *seguir linhas*, sendo que as coordenadas dos pontos amostrados são disponibilizadas pelo módulo de mapeamento. Diversos experimentos desse tipo foram executados com voltas executadas no sentido horário e no sentido anti-horário, incluindo voltas sucessivas realizadas pelo robô no mesmo experimento. Em ambos os casos, a arquitetura proposta permite ao robô navegar sobre uma trajetória pré-definida.

5.6.2 Exploração e navegação segura

Outro tipo de experimento realizado é mostrado na Figura 34. Nesse tipo experimento, o piso com a mesma linha pintada foi utilizado sobre o labirinto. Entretanto, foi inserido um obstáculo (parede interna) no labirinto, interrompendo a continuidade do percurso sobre a linha pintada sobre o piso.

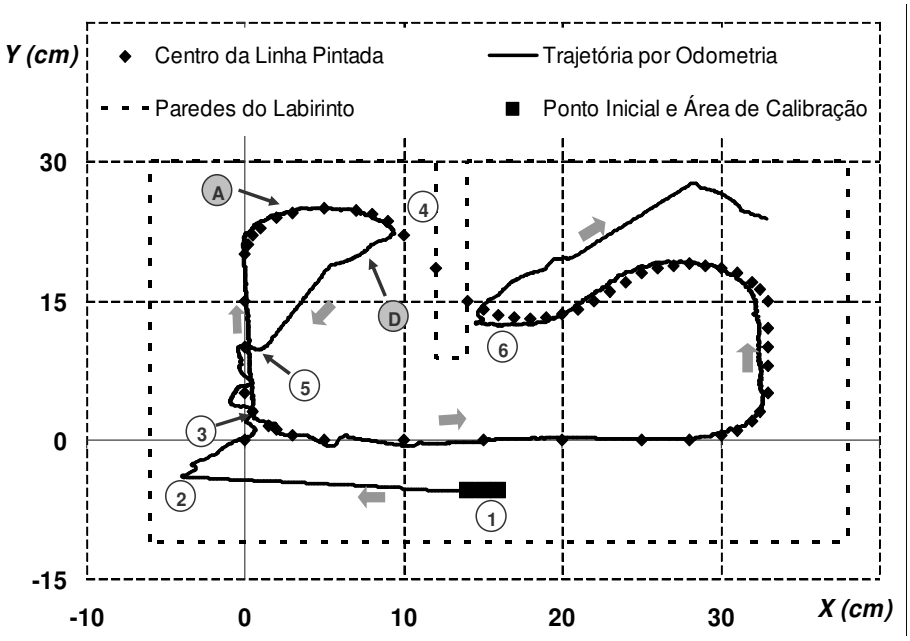


Figura 34 - Tarefa de exploração de ambiente para avaliação dos comportamentos e do processo de arbitragem.

Na Figura 34, a linha tracejada em preto representa o lado interno das paredes do labirinto, incluindo a parede interna inserida como obstáculo. Os pontos discretos correspondem aos 55 pontos coletados manualmente sobre o papel milimetrados referente ao percurso fechado. A linha continua é formada pela interligação de 2500 (dois mil e quinhentos) pontos referentes às coordenadas obtidas pelo cálculo realizado pelo módulo de mapeamento a partir das leituras

realizadas por cada odômetro. As setas em cinza claro indicam o sentido da trajetória do robô.

A Figura 34 ilustra seis regiões identificadas por números de 1 a 6, onde diferentes comportamentos da arquitetura proposta são ativados de acordo com diferentes contextos de aplicação. A região 1 é o local de partida do robô para exploração do ambiente e é o local onde se realiza o processo de calibração detalhado no Item 5.3.1. Entre as regiões 1 e 2, o robô utiliza prioritariamente o comportamento *manter rota*, observando-se que na região 2, o robô defronta-se com uma das paredes do labirinto e, sob determinação do árbitro-β, passa a utilizar o comportamento *identificar objeto*. Na região 2, o robô desvia do obstáculo e mantém uma rota retilínea até encontrar na região 3 um ponto da linha pintada sobre o piso. Entre a região 3 e a região 4, o robô utiliza o comportamento *seguir linha* para seguir em sentido horário uma trajetória pré-definida, baseada na linha pintada sobre o piso, até encontrar uma parede interna na região 4. A partir da região 4, o robô abandona a linha pintada para evitar uma colisão com a parede interna do labirinto utilizando o comportamento *identificar objeto*. Após desviar da parede na região 4, o robô mantém a trajetória utilizando o comportamento *manter rota* até encontrar novamente, em um ponto da região 5, a linha pintada sobre o piso. A partir da região 5, o robô passa a seguir a linha pintada em sentido anti-horário até encontrar com a parede interna do labirinto na região 6 e evitar a colisão com esta.

Entre as regiões 3 e 5, ilustradas na Figura 34 e descritas anteriormente, são destacadas através das letras A e D duas regiões. A guiagem e a ativação de comportamentos entre essas duas regiões são detalhadas na Figura 35. Nesse detalhamento são identificados, além das regiões A e D, as regiões B e C.

Na Figura 35 apresenta um gráfico com a trajetória do robô entre as regiões A e D, ilustrada também na Figura 34, e apresenta um diagrama que mostra em cada instante de amostragem qual dos comportamentos está ativo na referida trajetória. No gráfico com a trajetória, os pontos discretos maiores correspondem a pontos coletados manualmente sobre o papel milimetrados e os pontos discretos menores formam uma trajetória composta por 360 (trezentos e sessenta) pontos com as coordenadas obtidas por odometria.

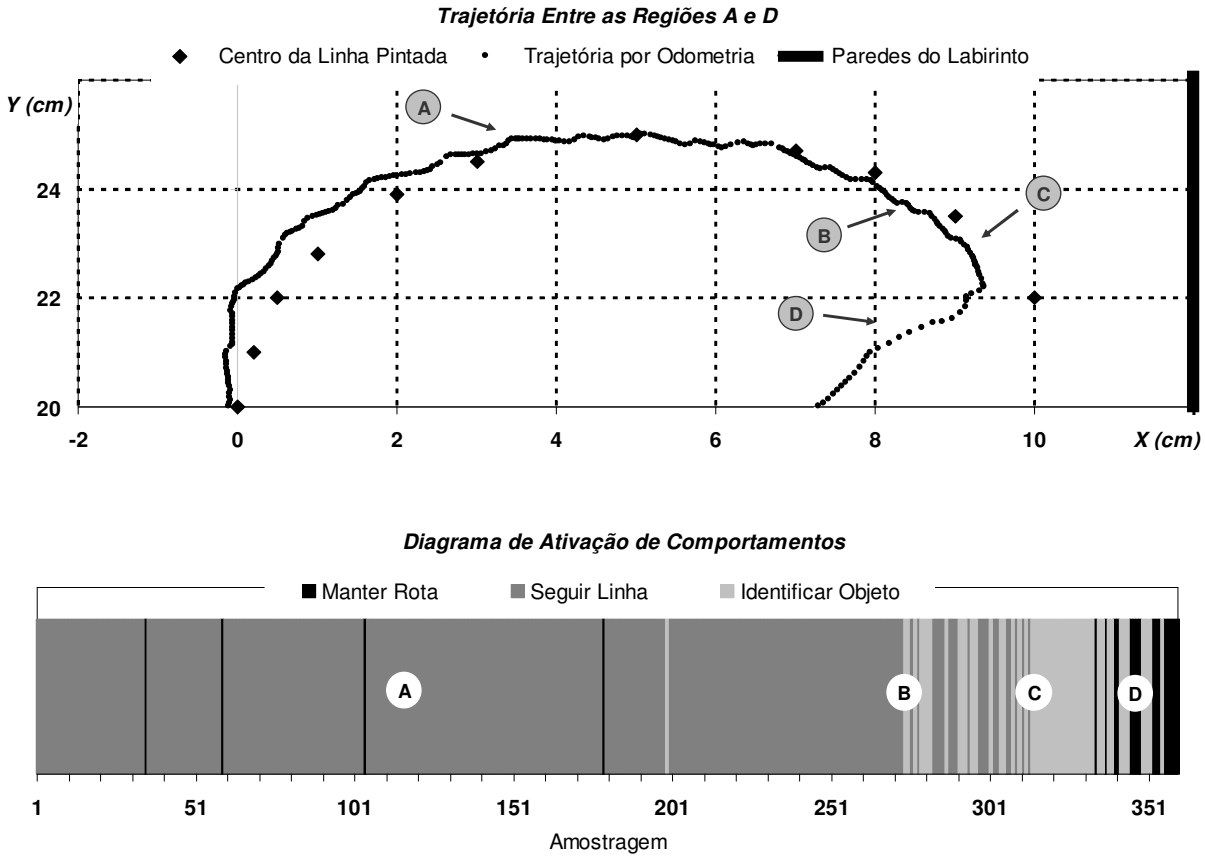


Figura 35 - Detalhamento da trajetória do robô e da ativação de comportamentos em tarefa para exploração de ambiente.

Na região A no gráfico da Figura 35, o robô navega seguindo a linha pintada no piso e, como pode ser observado no diagrama de ativação na mesma Figura 35,

o comportamento *seguir linha* é utilizado na guiagem do robô em praticamente todo o período de amostragem até a região B. Em quatro pontos de amostragem na região A do diagrama é notada a ativação de outros comportamentos, sendo que em três pontos foi ativado o comportamento *manter rota* e em uma vez foi ativado o comportamento *identificar objeto*. A ativação desses comportamentos pelo árbitro-β não condiz com o contexto de aplicação, na medida em que não haviam descontinuidades na linha pintada e não haviam obstáculos nessa região. Essas ativações são atribuídas a erros aleatórios do processo de medição, que, entretanto, não comprometeram a navegação.

A região B no gráfico da Figura 35 é uma região de transição em que a proximidade de um obstáculo (parede interna) introduz um contexto em que há uma linha para ser seguida e um objeto é identificado. Entretanto, como pode ser visto no diagrama de ativação da Figura 35, a distância do objeto ainda não é crítica e os comportamentos *seguir linha* e *identificar objeto* alternam-se na guiagem do robô até a região C.

Na região C no gráfico da Figura 35, a distância do obstáculo passa a ser considerada crítica pelo árbitro-β, que então ativa o comportamento *identificar objeto*. Assim, como pode ser observado pelo diagrama de ativação, o comportamento *identificar objeto*, que possui maior prioridade em relação aos outros dois comportamentos, permanece ativo até a região D buscando evitar uma colisão com o obstáculo. Na região C, com a predominância do comportamento *identificar objeto*, o árbitro-α define em cada instante sobre a ativação de um dos sub-comportamentos que utilizam os pares de sensores de IR, *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*, através de um processo simples de encolha baseado na maior leitura entre os sensores de IR, como descrito no Item 5.4. Como

pode ser observado no Item 5.3.2, na Figura 22 e na Figura 23, os sub-comportamentos *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita* são repulsivos, ou seja, promovem o acionamento dos motos-redutores no sentido de afastar o robô do objeto identificado. Dessa forma, como o robô aproxima-se do objeto com uma inclinação que privilegia as leituras dos pares de sensores de IR frontal e lateral esquerdo, os comportamentos *evitar obstáculo* e *beirar linha esquerda* alternam-se na ativação e conduzem o robô a uma rotação no sentido horário que o afasta da parede interna do labirinto.

Na região D no gráfico da Figura 35, o robô assume gradativamente uma posição segura ao afastar-se da parede. Assim, como pode ser visto no diagrama de ativação da Figura 35, o árbitro-β alterna a ativação dos comportamentos *identificar objeto* e *manter rota*, até que a distância do objeto (parede interna) torna-se segura e o comportamento *manter rota* prevalece na guiagem do robô.

Diversas execuções desse tipo de experimento foram realizadas com a navegação iniciada em pontos diversos do labirinto. Da mesma forma ao experimento ilustrado na Figura 34 e na Figura 35 deste item (Item 5.6.2), verificou-se que a arquitetura proposta permite ao robô navegar sem tocar as paredes do labirinto e os obstáculos adicionados a esse. Quando o robô encontra a linha pintada sobre o piso, este passa a seguí-la até que seja identificado um objeto ou o final da linha. Nessa situação o robô inicia a exploração até que encontre um outro trecho da linha novamente.

5.6.3 Guiagem e navegação entre linhas de plantas

O terceiro tipo de experimento foi viabilizado a partir da construção de um ambiente simulado com três linhas com plantas (anteparos vegetais). As linhas de plantas foram fixadas paralelamente uma das outras sobre o piso do labirinto, a uma distância aproximada de 15 centímetros. Dessa forma, foram formadas duas ruas entre as linhas de plantas. Uma das linhas laterais à linha central foi posicionada a uma distância de 20 centímetros em relação a uma das paredes laterais do labirinto, formando uma terceira rua com plantas de um lado e a parede de madeira do outro. Ao final das ruas foram pintadas linhas em preto sobre o piso de cartolina branca com aproximadamente 10 milímetros de largura, formando percursos para orientar o robô à rua seguinte (manobra de final de linha). A Figura 36 ilustra através de uma foto o ambiente construído e mostra um gráfico com os dados obtidos por odometria com o FK em uma das rodadas de experimento.

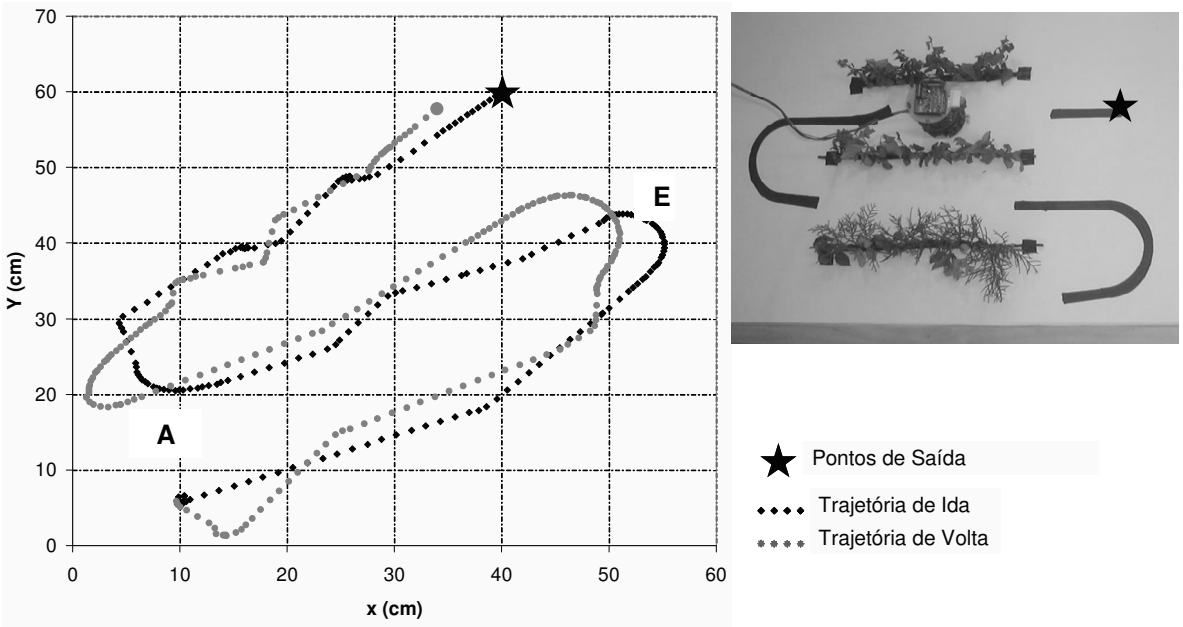


Figura 36 - Experimento realizado em um ambiente simulado com três linhas com plantas (foto do robô durante no experimento e gráfico com trajetória coletada por odometria).

O gráfico da Figura 36 mostra a trajetória de ida e de volta executada pelo robô em navegação orientada pelas linhas de plantas e pelas linhas pintadas sobre o piso. No gráfico, os pontos discretos em preto correspondem aos pontos coletados por odometria no percurso de ida, sendo que o ponto de partida é demarcado por uma estrela. Os pontos discretos em cinza correspondem ao percurso de volta. Os pontos dos percursos de ida e de volta totalizam juntos 410 pontos com coordenadas obtidas pelo cálculo realizado pelo módulo de mapeamento a partir das leituras realizadas por cada um dos odômetros do robô.

Pelo gráfico da Figura 36 observa-se que a arquitetura proposta permite ao robô navegar nas ruas entre as linhas sem a orientação de linhas pintadas. Dessa forma no experimento relatado, enquanto o robô está navegando pelas ruas, a guiagem fica restrita a atuação dos comportamentos *manter rota* e *identificar objeto* selecionados pelo árbitro-β. Além disso, quando o comportamento *identificar objeto* o árbitro-α define em cada instante sobre a ativação de um dos sub-comportamentos que utilizam os pares de sensores de IR, *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*, como descrito no Item 5.4.

Como pode ser observado no Item 5.3.2, na Figura 22 e na Figura 23, os sub-comportamentos *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita* são repulsivos, ou seja, promovem o acionamento dos motos-redutores no sentido de afastar o robô do objeto identificado. Assim, o comportamento *beirar linhas* é implementado pela ativação dos sub-comportamentos *beirar linha esquerda* e *beirar linha direita* que posicionam o robô a uma distância segura entre as linhas de plantas. A Figura 37 mostra detalhadamente a guiagem e a ativação de comportamentos entre as regiões A e E da rua central do experimento mostrado

também na Figura 36. Nesse detalhamento são identificados, além das regiões A e E, as regiões B, C e D.

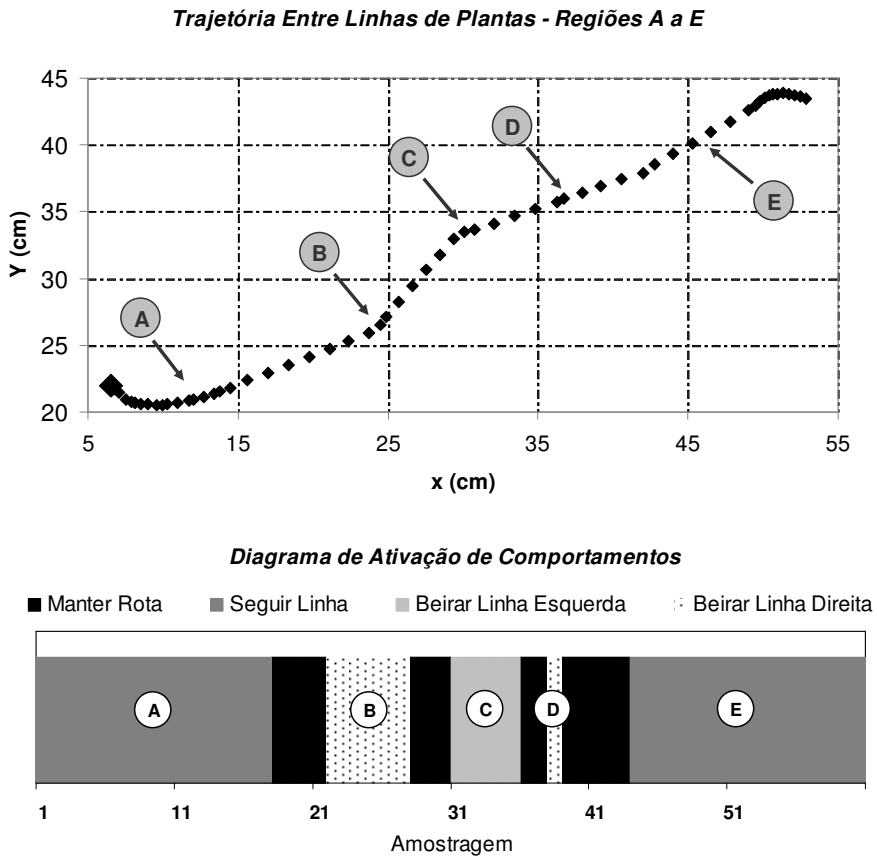


Figura 37 - Detalhamento da trajetória do robô e da ativação de comportamentos em tarefa de navegação entre linhas de plantas.

A Figura 37 apresenta um gráfico com a trajetória percorrida pelo robô entre as regiões A e E da segunda rua na etapa de ida. Juntamente ao gráfico é apresentado um diagrama que mostra em cada instante de amostragem qual dos comportamentos está ativo na referida trajetória. No gráfico, os pontos discretos formam uma trajetória composta por 60 (sessenta) pontos com as coordenadas obtidas por odometria. O diagrama de ativação apresentado busca detalhar a ativação do comportamento *identificar objeto* através identificação de qual de seus

sub-comportamentos atua durante a sua ativação. Assim, nas regiões B, C e D, o comportamento *identificar objeto* está ativado, porém o digrama de ativação mostra qual dos seus sub-comportamentos componentes é utilizado em cada instante da amostragem, *evitar obstáculo*, *beirar linha esquerda* ou *beirar linha direita*.

Na região A no gráfico da Figura 37, o robô navega seguindo a linha pintada no piso que o conduz da primeira rua até a segunda rua entre as linhas de plantas. Como pode ser observado no diagrama de ativação na mesma Figura 37, o comportamento *seguir linha* é utilizado na guiagem do robô até o final da linha pintada sobre o piso e, posteriormente, o comportamento *manter rota* é ativado até a região B, em que é identificada como próxima a linha de plantas à direita do robô. Nesse instante, o árbitro-β desativa o comportamento *manter rota* e ativa o comportamento *identificar objeto*, sendo que nesse mesmo momento o árbitro-α define pela ativação do sub-comportamento *beirar linha direita*.

O sub-comportamento *beirar linha direita* atua na região B da trajetória mostrada no gráfico da Figura 37, promovendo uma rotação suave à esquerda e posicionado o robô em uma região segura. Após o posicionamento seguro, o comportamento *manter rota* é ativado novamente até que é identificada como próxima a linha de plantas à esquerda do robô. Nesse momento, o árbitro-β desativa o comportamento *manter rota* e ativa o comportamento *identificar objeto*, sendo que o árbitro-α define pela ativação do sub-comportamento *beirar linha esquerda*. Esse ultimo sub-comportamento atua na região C da trajetória promovendo uma rotação suave à direita e posicionado o robô em uma região segura.

Após o posicionamento executado pelo sub-comportamento *beirar linha esquerda* na região C, o comportamento *manter rota* é ativado conduzindo o robô

até a região D como pode ser observado no diagrama de ativação na Figura 37. Nessa região o árbitro-β ativa o comportamento *identificar objeto* e no mesmo momento o árbitro-α ativa o sub-comportamento *beirar linha esquerda* para posicionar o robô a um ponto seguro. Após esse posicionamento, o robô é conduzido pelo comportamento *manter rota* até a região E, onde a linha pintada é identificada no final da rua, e o robô é conduzido pelo comportamento *seguir linha* até a próxima rua.

Outras execuções desse tipo de experimento ilustrado na Figura 36 e na Figura 37 deste item (Item 5.6.3) foram realizadas. Da mesma maneira que o experimento ilustrado, verificou-se que a arquitetura proposta permite ao robô navegar entre as ruas baseando-se apenas nos comportamentos *manter rota* e *identificar objeto*, selecionados pelo árbitro-β de acordo com os contextos de aplicação apresentados ao robô durante a navegação. Com o auxílio do comportamento *seguir linha* para realizar manobras no final das linhas, os experimentos mostraram a capacidade da arquitetura de operar em diferentes contextos, permitindo ao robô percorrer trajetórias de ida e volta entre as linhas de plantas. Esses resultados positivos foram atingidos para velocidades até 25% da velocidade máxima da plataforma robótica Khepera. Em velocidades acima desse valor, o tempo de resposta da arquitetura de controle não permite a rápida tomada de decisão, acarretando perda do caminho ou colisões. Esse atraso não está relacionado com a estrutura da arquitetura, mas sim com os atrasos limitados pela capacidade de processamento (externo ao robô) e do meio de comunicação entre o robô e o computador externo.

5.7. Considerações gerais sobre a arquitetura híbrida proposta

Os experimentos realizados mostraram que os três comportamentos complexos *manter rota*, *seguir linha* e *identificar objeto* são independentes e atuam paralelamente gerando comandos de atuação para guiagem e para navegação autônomas. Cada comportamento tem entradas sensoriais independentes, sendo que um sensor pode alimentar um ou mais comportamentos ao mesmo tempo. No primeiro experimento (Item 5.6.1), observou-se a que o comportamento *seguir linha*, que possui prioridade intermediária, foi capaz de guiar o mini-robô sobre um percurso fechado atuando independentemente dos outros comportamentos e viabilizando o mapeamento do percurso pelo módulo de mapeamento.

No segundo experimento (Item 5.6.2), os três comportamentos foram utilizados de forma independente para a exploração segura do ambiente pelo mini-robô. Nesse segundo experimento, a ativação de um ou outro comportamento é definida pelo árbitro-β de forma progressiva, ou seja, entre duas regiões onde apenas um comportamento prevalece há uma região de transição em que os comportamentos alternam a ativação até que o novo contexto seja identificado claramente. Essa característica introduz robustez à arquitetura na medida em que a alternância entre comportamentos na região de transição não prejudica a navegação e promove a imunidade a ruídos aleatórios que poderiam causar algum procedimento incorreto pela identificação de um contexto não existente. Também, o árbitro-α define em cada instante sobre a ativação de um dos sub-comportamentos, *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*.

Os sub-comportamentos *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita* são repulsivos, ou seja, promovem o acionamento dos atuadores no sentido de afastar o robô do objeto identificado. Porém, no terceiro experimento (Item 5.6.3) os sub-comportamentos *beirar linha esquerda* e *beirar linha direita* utilizam essa característica repulsiva para conduzir o mini-robô pelas linhas de planta, mantendo uma distância segura de ambos os lados. Os três sub-comportamentos mais simples, com menor número de entradas e menor número de regras difusas, têm suas recomendações selecionadas pelo árbitro-α compondo através desse processo o comportamento mais complexo *identificar objeto*. Nesse experimento fica evidenciada a característica hierárquica na implementação dos comportamentos, e, além disso, destaca-se a possibilidade de um ou mais árbitros constituídos ou não por controladores difusos serem empregados para coordenação dos comportamentos.

A arquitetura proposta é similar à arquitetura DANM descrita por Williams et al. (2001) e Rosenblatt, Williams e Durrant-Whyte (2002), na medida em que essa arquitetura possibilita que diferentes comportamentos mais ou menos complexos, construídos a partir da aplicação de diferentes métodos, possam compartilhar o controle robótico. Além disso, a arquitetura DANM utiliza um ou mais árbitros difusos para construção de módulos de arbitragem que realizam a coordenação dos comportamentos existentes através de um processo de arbitragem baseado na avaliação de votos gerados pelos próprios comportamentos.

Porém, diferente da arquitetura DANM, a arquitetura proposta permite utilização de árbitros não difusos. Além disso, semelhante aos trabalhos de Seraji e Howard (2002) e de Hagras, Callaghan e Colley (2001), a arquitetura se beneficia das características da lógica difusa devido a sua eficácia em lidar com diferentes

graus de incerteza e de traduzir a relação robô-ambiente para estruturas com expressões lingüísticas mais intuitivas para o ser humano. Além disso, a lógica difusa também permite uma suavização na transição entre os comportamentos robóticos.

A utilização de controladores difusos também destaca a arquitetura proposta em relação às arquiteturas propostas por Blackmore et al. (2004) e por Bak e Jakobsen (2004). Entretanto, a arquitetura proposta assemelha-se às arquiteturas propostas por Blackmore et al. (2004) e por Bak e Jakobsen (2004) na medida em que, essas utilizam comportamentos primitivos similares aos desenvolvidos para robôs móveis para ambientes dinâmicos e pouco estruturados apresentados no Item 3.3, e comportamentos de nível de abstração mais alto específicos para aplicações agrícolas. Também, semelhante às arquiteturas propostas para robôs agrícolas em Blackmore et al. (2004) e Bak e Jakobsen (2004) e à arquitetura proposta para robôs submersível em Williams et al. (2001) e em Rosenblatt, Williams e Durrant-Whyte (2002), os comportamentos e os árbitros atuam de forma descentralizada, permitindo a construção de um sistema com processamento distribuído e com flexibilidade para inserção ou remoção de novos comportamentos e árbitros. Além disso, é adequada à operação sobre sistemas de controle via rede e permite que dispositivos sejam inseridos ou removidos de acordo com as necessidades de cada operação.

6. ROBÔ AGRÍCOLA MÓVEL - APONTAMENTOS SOBRE A IMPLEMENTAÇÃO DA ARQUITETURA PROPOSTA

O Laboratório de Simulação e Controle do Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos da USP (SEM/EESC-USP) e a Embrapa Instrumentação Agropecuária de São Carlos (Embrapa Cnpdia) desenvolvem através de projetos colaborativos um veículo autônomo agrícola para ser uma plataforma robótica experimental de campo para dar suporte à pesquisa e desenvolvimento de sistemas de automação para máquinas agrícolas. O objetivo maior desse projeto é iniciar trabalhos interdisciplinares, bem como iniciar ações que integrem grupos multidisciplinares e multi-institucional promovendo e fomentando o desenvolvimento das referidas tecnologias no país.

A plataforma robótica agrícola experimental proposta e que está sendo desenvolvida é composta por dois sub-sistemas principais: um sub-sistema constituído por uma estrutura física com sistemas mecânicos e eletrônicos que será o VAA ou o RAM propriamente dito para atuar em campo – Sub-sistema de Campo; e um sub-sistema de desenvolvimento, controle e monitoramento montado em uma estação base, como por exemplo um laboratório ou escritório de uma fazenda – Sub-sistema Laboratório/Estação Base. A Figura 38 ilustra a plataforma proposta.

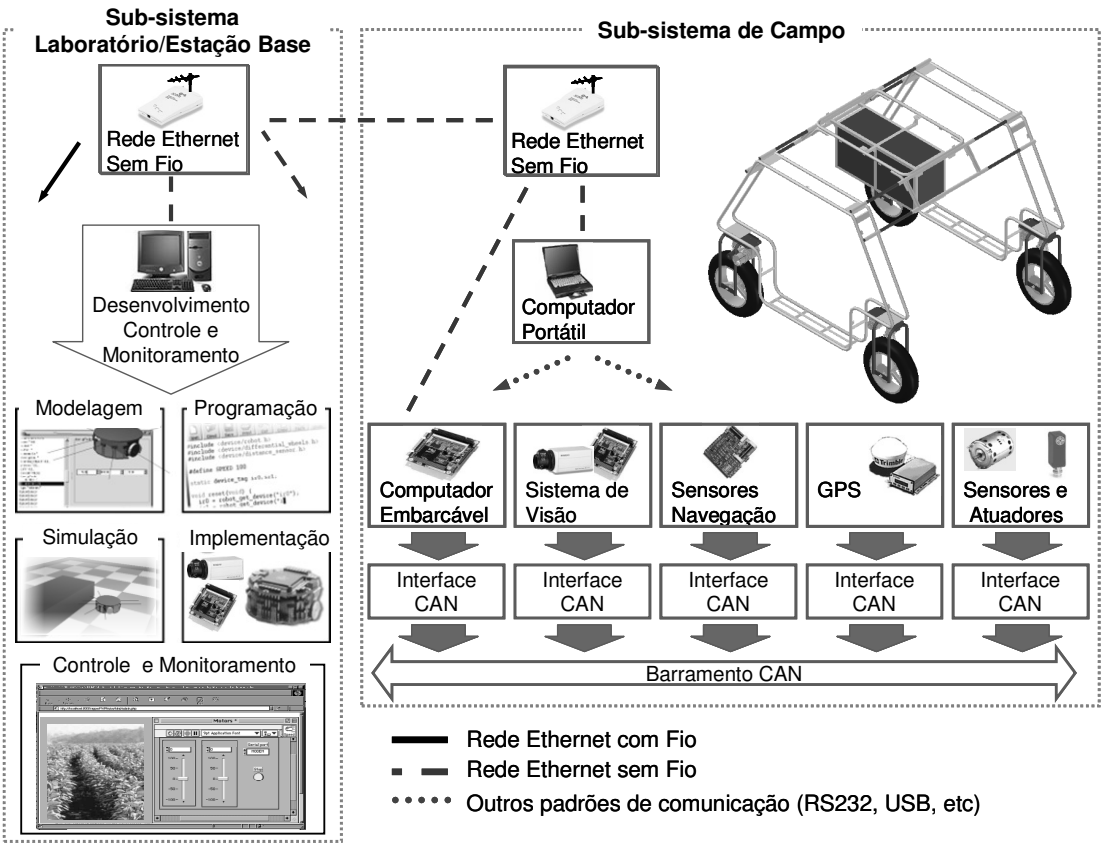


Figura 38 - Sub-sistema Laboratório/Estação Base e Sub-sistema de Campo componentes da plataforma robótica agrícola experimental.

Como pode ser visto na Figura 38, o Sub-sistema Laboratório/Estação Base é composto de laboratório de desenvolvimento de sistemas mecatrônicos, laboratório de modelagem e simulação e ambientes de desenvolvimento de programas computacionais. O Sub-sistema de Campo constitui-se de um veículo com sistemas mecatrônicos embarcados. Uma rede CAN é responsável pela integração dos dispositivos eletrônicos de percepção, computação e atuação. A estrutura do veículo foi desenvolvida para ser re-dimensionável e para permitir diferentes montagens de acordo com os testes e tarefas a serem realizados em campo.

A partir da viabilização da operação da plataforma para realização de tarefas no campo, o Sub-sistema de Laboratório/Estação Base tem a função de apoio ao

Sub-sistema de Campo, permitindo planejamento, controle e monitoramento de tarefas em tempo real através de enlace digital de comunicação de dados. A rede CAN desenvolvida e embarcada no veículo de campo possibilita não somente a integração dos sensores, atuadores e sistemas computacionais relativos a tarefas de guiagem e navegação, assim como a integração de dispositivos relacionando medidas de variáveis agronômicas para aquisição de dados agronômicos.

Para implementar a arquitetura híbrida na plataforma robótica agrícola em desenvolvimento demandam-se algumas adaptações em relação à implementação no mini-robô, principalmente em relação aos dispositivos dos módulos de percepção e atuação. Entretanto, a sua forma básica apresentada no Item 5.1 e ilustrada na Figura 14 pode ser mantida.

Para atender as necessidades de percepção da arquitetura híbrida para guiagem e para navegação autônomas são definidos os seguintes componentes para o módulo de perceptivo da plataforma robótica agrícola:

- *Distância de obstáculos*: sensores ultrassônicos ou sonares para identificar a proximidade de obstáculos no caminho do robô. Associado ao comportamento *evitar obstáculo*;

- *Visão de linhas de plantio*: receptor DGPS e *scanner laser* para identificar e representar virtualmente em um sistema de coordenadas as linhas de plantio para que se possam definir trajetórias para o robô nas ruas entre as linhas de plantio. Associados ao comportamento *seguir linha*;

- *Detecção de linhas de plantio*: sensores ultrassônicos ou sonares para identificar a distância lateral das linhas de plantio ao longo das ruas da plantação. Associado aos sub-comportamentos *beirar linha esquerda* e *beirar linha direita*;

- *Posicionamento relativo e posicionamento absoluto*: receptor DGPS, odômetros e sistema inercial composto por GDS e giroscópio. Associados ao módulo de mapeamento;

Entre os dispositivos propostos para atender as necessidades de percepção da arquitetura híbrida para guiagem e para navegação autônomas, o *scanner laser* e os sensores ultrassônicos são elementos que possuem aplicações crescentes, mas que demandam esforços de pesquisa e desenvolvimento para sua aplicação em sistemas autônomos agrícolas como identificado na descrição no Capítulo 4.

O *scanner laser*, assim como a utilização de visão computacional baseada em câmeras CCD, utiliza técnicas para aquisição e processamento de dados que necessita de um estudo mais extenso e que foge ao escopo deste trabalho de doutorado. Entretanto, a utilização de sensores ultrassônicos para navegação em ruas entre linhas de plantio utilizando o comportamento *identificar objeto* e seus sub-comportamentos, *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*, é notada como essencial para definição do módulo de perceptivo da plataforma robótica agrícola. Dessa forma, foi realizado um estudo para caracterização de sensores ultrassônicos para identificação de distância em relação a linhas de plantas.

Nos itens seguintes (Itens 6.1, 6.2 e 6.3) é apresentada a caracterização de sensores ultrassônicos. Também, baseando-se no trabalho Godoy (2007), utiliza-se

um modelo para simulação e análise de um sistema de controle via rede CAN. Para simulação da rede são considerados os dispositivos definidos para compor o módulo perceptivo da plataforma robótica agrícola, com o objetivo de avaliar alguns parâmetros de desempenho da rede CAN. Nesses itens a seguir, a plataforma robótica agrícola experimental é referida simplesmente como RAM.

6.1. Estrutura mecatrônica do RAM

A estrutura mecânica do RAM é constituída por uma estrutura física em pórtico (travamento superior). O pórtico é redimensionável e permite o ajuste da largura da estrutura para operar em diferentes culturas. A estrutura é modular e composta por quatro partes principais apresentadas na Figura 39 e descritas a seguir:

- *Chassis inferiores (CHI1 e CHI2):* são duas estruturas laterais que conectam e suportam os MGP's, além de transportar baterias e dispositivos diversos;

- *Chassi superior (CHS):* estrutura responsável pela conexão dos dois chassis inferiores e pelo travamento superior da estrutura;

- *Módulos de guiagem e propulsão: (MGPn: MGP1, MGP2, MGP3, MGP4):* são quatro sistemas eletro-mecânicos responsáveis pelo controle direcional e pela propulsão do RAM;

- *Gabinete:* estrutura fixada ao CHS que acomoda os computadores principais do RAM, além de dispositivos relacionados com o controle de energia, alarmes e bloqueadores para segurança.

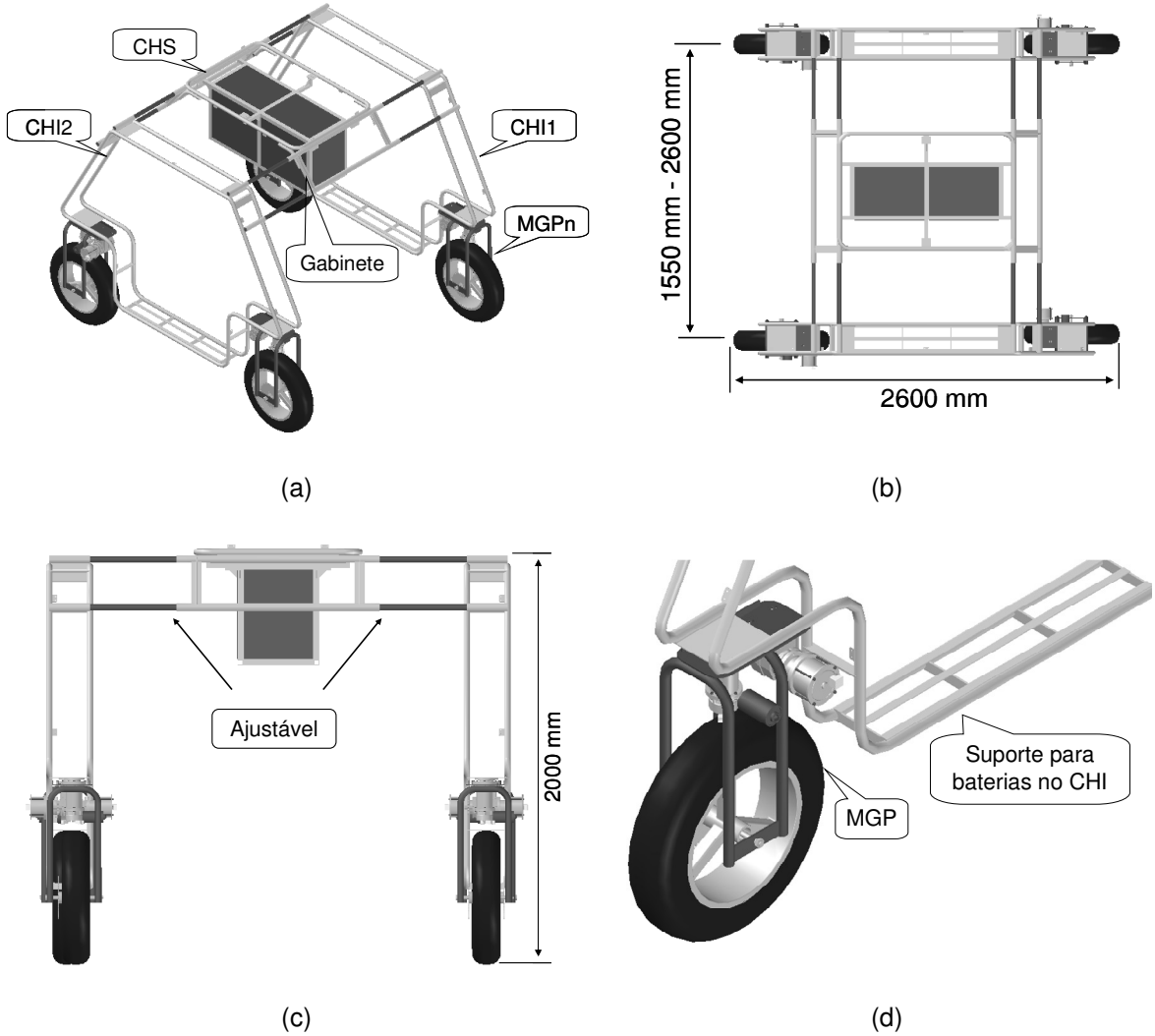


Figura 39 - Vista isométrica da estrutura mecânica do RAM: (a) vista tridimensional com ilustração dos módulos principais; (b) vista superior com detalhamento das medidas principais; (c) vista frontal com detalhamento da altura da estrutura e do mecanismo de redimensionamento; e (d) vista tridimensional com ilustração de um MGP e do suporte para baterias em um CHI.

Cada MGP contém dois motores elétricos, um para guiagem e outro para propulsão, sendo que cada motor é acoplado a um redutor. Dessa forma, a tração é

realizada nas 4 rodas, e a guiagem pode ser implementada através de duas ou quatro rodas. A Figura 40 mostra vistas do MGP projetado destacando as suas principais partes eletromecânicas componentes.

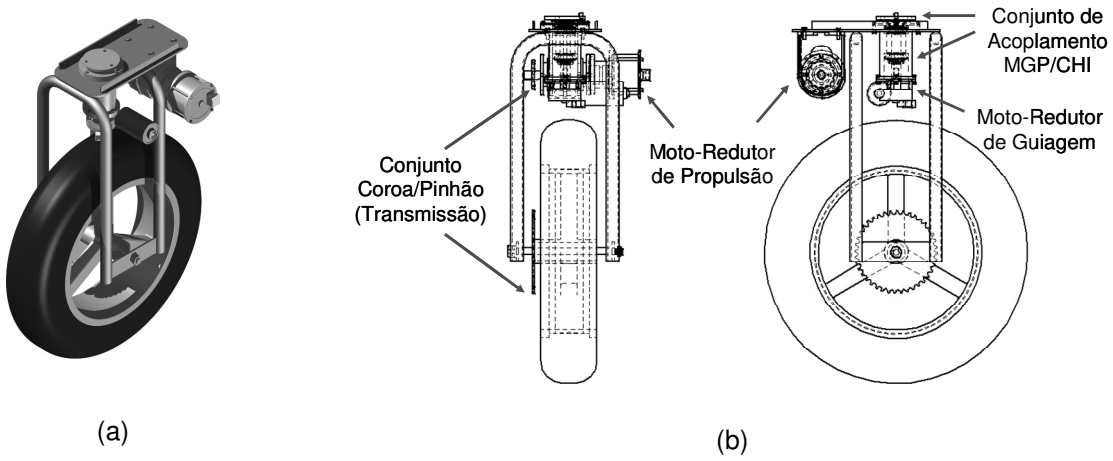
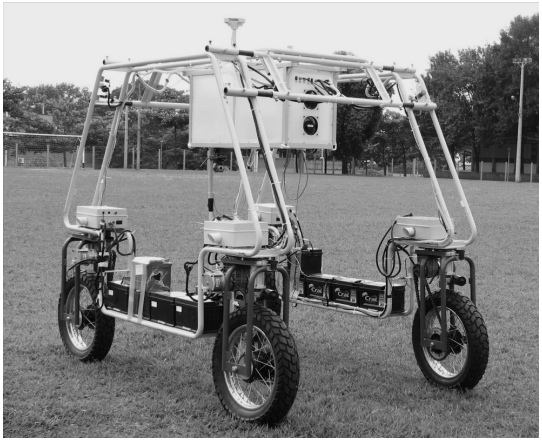


Figura 40 - Vistas do MGP: (a) vista isométrica; e (b) vistas lateral e frontal com detalhamento das principais partes componentes.

Como pode ser visto na Figura 40b, o MGP possui um conjunto de acoplamento que permite a rotação do MGP em relação ao CHI. Além disso, a Figura 40b mostra que o sistema de propulsão possui ainda um sistema de transmissão baseado em corrente com coroa e pinhão para transmissão do torque entre o eixo da roda e a saída do conjunto moto-redutor de propulsão. Tanto o sistema de transmissão (coroa, pinhão, corrente, eixo e rolamentos) como as rodas e pneus são componentes de motocicletas comerciais. Optou-se por adequar o projeto para utilização de componentes mecânicos comercias para reduzir custos de projeto e fabricação de peças especiais. A Figura 41 mostra fotos da estrutura completa e do MGP montado.



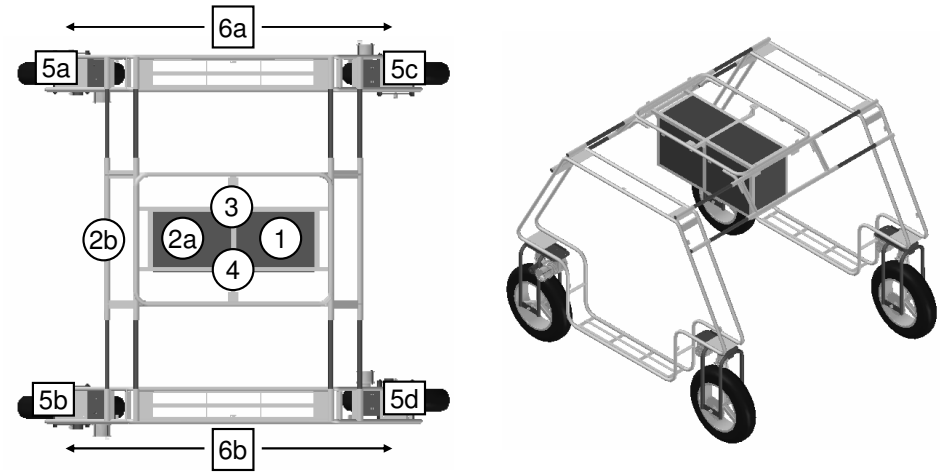
(a)

(b)

Figura 41- Fotos do RAM: (a) estrutura mecânica completa; e (b) MGP acoplado a CHI.

Os principais dispositivos de percepção, computação e atuação possuem posicionamento definido e ilustrado pela Figura 42. Todos os dispositivos serão integrados pela rede CAN embarcada no RAM, sendo que os conjuntos controladores e os computadores fazem parte da rede CAN atualmente.

Cada dispositivo do módulo de perceptivo listado na Figura 42 está relacionado com uma ou mais funções na arquitetura híbrida para guiagem e navegação como descrito anteriormente neste mesmo capítulo (Capítulo 6).



1	Computador principal
2a	Computado auxiliar
2b	Scanner laser
3	DGPS
4	Sistema inercial (giroscópio, acelerômetro e bússola)
5a,b,c,d	Conjunto composto por 2 controladores de motor (guiagem e propulsão), sensor potenciométrico e encoder
6a,b	Conjuntos de sensores ultrassônicos

Figura 42 - Vistas da estrutura do RAM ilustrando os dispositivos principais interligados pela rede CAN.

Os conjuntos 5a, 5b, 5c e 5d da Figura 42 contêm os motos-redutores especificados para compor o MGP: motor Bosch modelo GPC (210 W) associado a um redutor Macopema modelo MC50 (1:25) para propulsão; e um motor-redutor Bosch modelo CPD (50 W) para guiagem. Esses conjuntos contêm também os controladores dos motores e os sensores de velocidade (*encoder*) e direção (potenciômetros). Em cada conjunto, os controladores e sensores foram integrados a interface CAN padrão como ilustrado no diagrama na Figura 43.

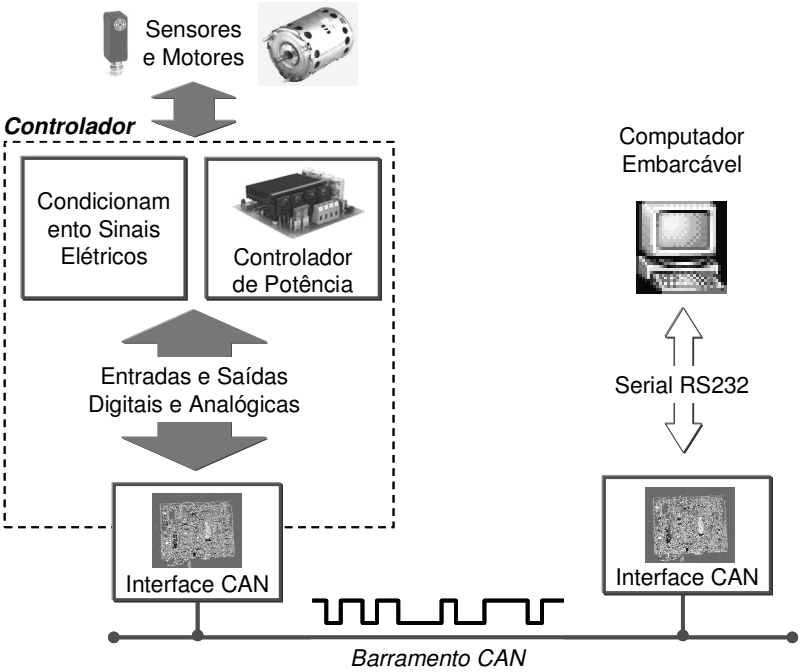


Figura 43 - Rede CAN composta por nó controlador de motor e computador embarcável.

O computador principal, ilustrado na Figura 42 (dispositivo 1) e na Figura 43, será utilizado para processamento dos comportamentos da arquitetura híbrida proposta para controle autônomo de navegação e guiagem. Esse computador central utiliza as informações de todos os sensores do módulo de perceptivo e gera comandos para os motores de guiagem e propulsão. A rede CAN permite que além dos dispositivos relacionados com a navegação, os dispositivos para aquisição de dados compartilhem o mesmo barramento. Desta forma, é possível inserção e remoção de forma simples de sensores e de sistemas computacionais que permitem a coleta e armazenamento de dados, assim como o gerenciamento destas atividades.

A interface CAN padrão utilizada para integrar os dispositivos à rede CAN do RAM foi projetada e construída baseando-se na proposta de SOUSA (2002). Na

implementação de um padrão baseado no CAN, os nós são tipicamente constituídos por três módulos elementares, que são:

- *Transceptor CAN:* módulo responsável pela adaptação dos níveis de tensão entre circuito do nó e do barramento CAN. Também promove a adaptação de impedâncias e a proteção do nó à faltas;

- *Controlador CAN:* módulo central da implementação e do controle do protocolo;

- *Sistema Computacional Central:* módulo constituído por CPU, memória, programas computacionais e interface com o controlador CAN e interfaces para outros dispositivos como sensores, atuadores e terminais de controle. Através desse módulo são implementados os protocolos de alto nível.

O protocolo CAN, como desenvolvido por Robert Bosch GmbH, foi padronizado e documentado internacionalmente pela ISO, gerando a norma ISO 11898, para aplicações de alta velocidade, e a norma ISO 11519, para aplicações de baixa velocidade. Estes padrões apresentam camada física e camada de enlace de dados em acordo com o modelo OSI (*Open Systems Interconnection*). Muitos dos padrões presentes no mercado adotaram as especificações das normas ISO 11898 e ISO 11519, devido a isso, existem no mercado muitos circuito integrados (CI's) que obedecem a essas especificações.

Os transceptores e controladores são os elementos básicos para elaboração de redes CAN, pois determinam as características de operação de rede, como velocidade de transferência de dados (taxa de bits), imunidade a interferências, robustez, capacidade de carga (número de nós), detecção, sinalização e distinção de erros, detecção de faltas, custo, entre outras. As normas da ISO recomendam

que os transceptores de interface sejam desenvolvidos de maneira que a comunicação possa continuar mesmo que um dos dois fios do barramento esteja rompido ou curto-circuitado à fonte de alimentação, ou ainda curto-circuitado à referência (ao terra). Os transceptores CAN podem ser adquiridos no mercado na forma de CI's ou podem ser construídos com componentes discretos.

A montagem dos diferentes quadros, o processo de arbitragem, o gerenciamento de envio de mensagens e recebimento de confirmação, a identificação de erros de comunicação e outras definições referentes ao CAN são implementadas pelos controladores CAN. Tais controladores podem ser encontrados no mercado sob diversas formas, desde placas tipo PCI (*Peripheral Component Interconnect*) para conexão em computadores pessoais, assim como cartões tipo PCMCIA (*Personal Computer Memory Card International Association*) e circuitos integrados simples. Atualmente no mercado podem ser encontrados microcontroladores com controladores CAN integrados ou microcontroladores com interfaces que permitem a integração fácil de controladores externos ao microcontrolador.

Para desenvolvimento do circuito de uma ECU descrita no Item 4.3, sendo o controlador responsável pela implementação do protocolo CAN, as demais regras definidas pela norma devem ser implementadas em um sistema computacional que se comunique com o controlador. Tal sistema computacional deverá ter capacidade de processamento e de memória relacionado com as características da ECU, podendo ser, por exemplo, um simples microcontrolador se a ECU for associada com sensores, ou um computador embarcável se a ECU for um terminal IHM (interface homem-máquina). A Figura 44 ilustra através de diagrama o circuito eletrônico de uma interface CAN desenvolvida e testada em laboratório para permitir

a conexão de dispositivos com interface RS232 ou sensores e atuadores através de interfaces analógicas ou digitais.

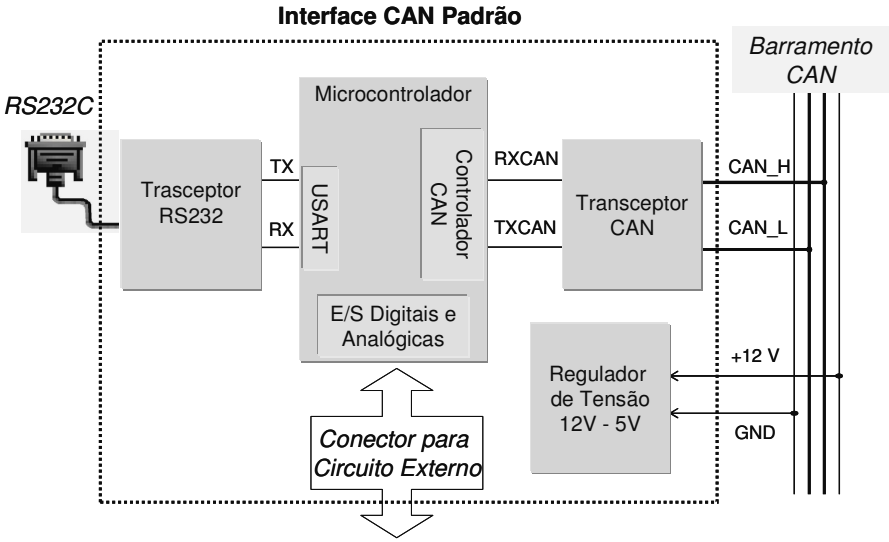


Figura 44 - Diagrama da interface CAN.

Como mostrado na Figura 44, o circuito da interface é composto basicamente por três circuitos integrados que são: microcontrolador, transceptor CAN e transceptor RS232. Os transceptores têm a função de adaptar os níveis de tensão dos sinais de comunicação para os níveis definidos para os respectivos padrões. A porta de comunicação RS232C é baseada em um controlador tipo USART (Universal Synchronous/Asynchronous Receiver Transmitter) e a porta CAN baseada em um controlador CAN. A interface possui também um módulo para regulação de tensão que abaixa a tensão provida pelo barramento, normalmente 12 V ou 24 V, para 5 V, que é a tensão de alimentação necessária aos circuitos integrados da interface. O módulo de conexão para circuito externo permite que entradas e saídas, analógicas ou digitais, estejam disponíveis diretamente para conexão de sensores e atuadores

ou através de circuito de condicionamento de sinais. Esse circuito constitui uma interface padrão que pode ser utilizada para compor diferentes nós CAN e formar uma rede CAN.

6.2. Caracterização e avaliação do comportamento de sensor ultrassônico para sistemas agrícolas autônomos

Sensores ultrassônicos são muitos utilizados no subsistema de veículos autônomos e robôs móveis em outras áreas de conhecimento. Características como baixo custo, robustez e simplicidade de implementação tornam a sua utilização atrativa para a detecção de objetos e determinação de distâncias em relação a esses objetos (SIEGWART e NOURBAKHSH, 2004).

Trabalhos recentes buscam determinar condições para aplicação de sensores ultrassônicos para o sistema de guiagem e navegação de máquinas agrícolas autônomas. Iida e Burks (2002) utilizaram arranjos laterais de sensores ultrassônicos em um trator para medir distância em relação às árvores de um pomar para promover a navegação segura entre as árvores sem danificá-las. Em Guo, Zhang e Han (2001) é descrito o desenvolvimento de um sistema baseado em dois sensores ultrassônicos para detectar objetos em movimento e permitir a navegação segura de máquinas agrícolas no campo. Os experimentos descritos em Hagras et al (2002) e detalhados anteriormente no Item 3.3 e no Item 4.2 utilizam sensores ultrassônicos para orientar a navegação de um robô. Porém, os experimentos apresentados em Hagras et al (2002) foram realizados em ambiente aberto controlado e não são apresentados detalhes sobre a implementação dos sensores.

Diante da carência de informações técnicas e científicas sobre o comportamento de sensores ultrassônicos para aplicação em máquinas agrícolas, montou-se um experimento para caracterização e para avaliação do comportamento de um sensor ultrassônico para medidas de distância em relação a obstáculos típicos do ambiente agrícola através da comparação com medidas de referência realizadas em laboratório.

6.2.1 Materiais e métodos aplicados para caracterização do sensor ultrassônico

Foi utilizado na caracterização o sensor ultrassônico modelo 6500 da Polaroid (POLAROID, 2004) que permite medir distâncias entre 0,15 m e 10,00 m com um erro absoluto de ± 1%. Esse sensor é composto de uma pastilha cerâmica (elemento sensor) que é conectada a um circuito transdutor que disponibiliza seus diversos sinais de controle e de transmissão de dados ao circuito controlador, nesse caso a interface CAN, que os repassa ao pela rede CAN a um computador tipo PC conectado a outra interface CAN. O funcionamento do sensor é baseado na reflexão de ondas sonoras sendo que no modo de operação *Single-Echo*, na qual o emissor envia 16 pulsos sonoros a uma freqüência de 49,4 kHz (ultrassom) que viajam pelo ar a uma velocidade de 30,48 cm a cada 0,9 ms. O sinal de retorno (eco) é recebido pelo receptor e condicionado eletricamente pelo circuito transdutor. O tempo (Δt) entre a emissão do primeiro pulso e o recebimento do sinal de eco é proporcional à distância do objeto.

Para o controle e transmissão das medidas foi desenvolvida uma rotina de programa para o microcontrolador da interface CAN (microcontrolador PIC18F258 da

empresa Microchip) e um programa para PC em Labview para solicitação, recebimento e armazenamento de dados. A Figura 45 mostra um fluxograma geral do programa do microcontrolador e a interface gráfica do programa para PC.

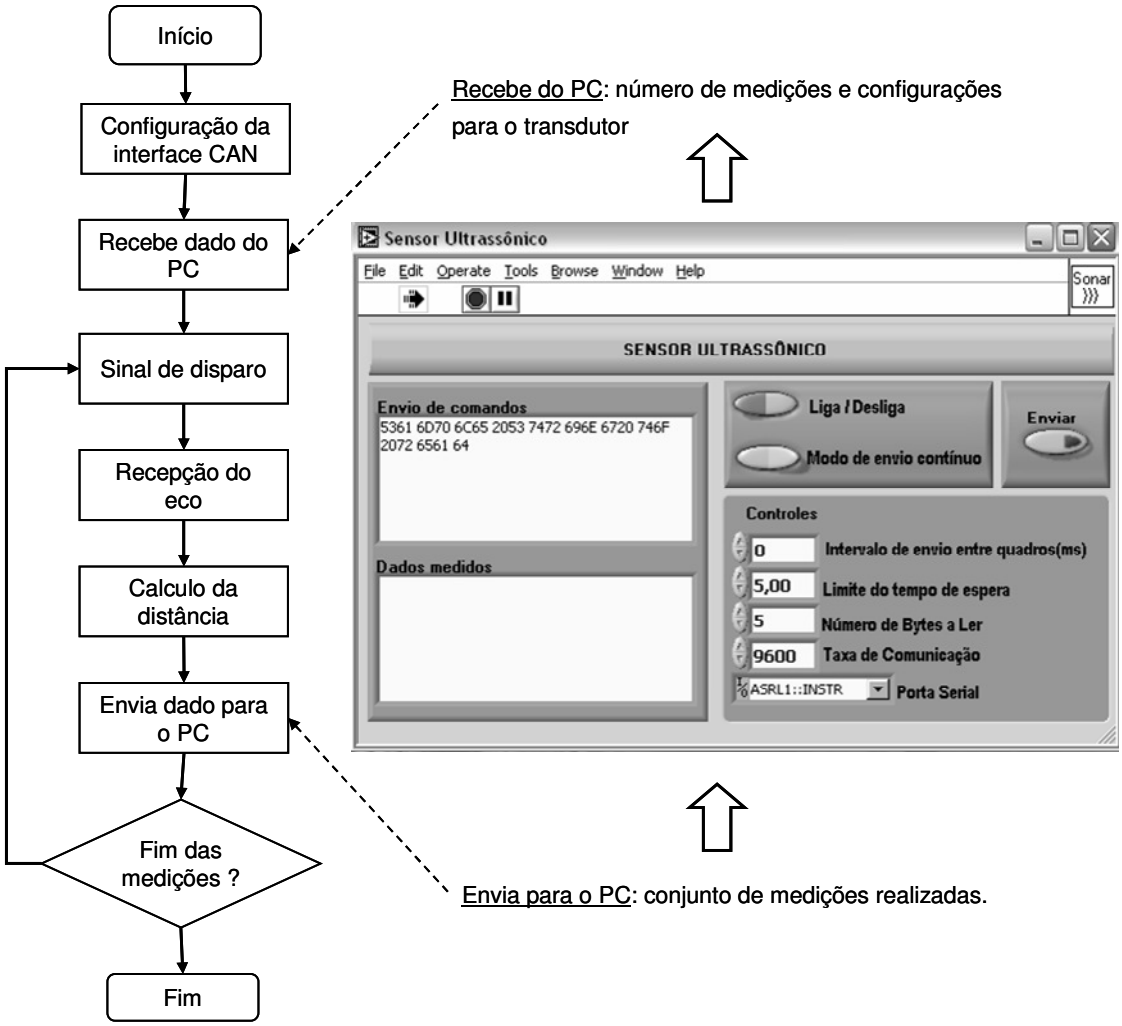


Figura 45 - Fluxograma do programa de controle do transdutor do sensor Polaroid 6500 e interface gráfica do programa para PC para controle das medições.

Como pode ser visto na Figura 45, o microcontrolador da interface CAN ao receber os dados de configuração e início da medição, entra em uma rotina com repetição para realização de medições. Essa rotina iterativa dispara uma medição enviando um sinal ao transdutor do sensor ultrassônico. Após o disparo, um

contador interno inicia a contagem de tempo com resolução de 400 ns até que um sinal enviado pelo tradutor informa o retorno do pulso de eco.

Conforme ilustrado no fluxograma da Figura 45, as medições foram processadas antes de serem transmitidas ao PC. Esse processamento consistia no cálculo da distância da pastilha do sensor ao anteparo a partir da equação.

$$d = \mathrm{T} \cdot v \cdot k^{-1} \tag{30}$$

Onde: *d* é a distância do sensor ao anteparo expressa em centímetros, *T* é o total de unidades de tempo contadas pelo microcontrolador, *v* é a velocidade do som no ar (33866,67 cm/s) e *k* é uma constante dependente do temporizador do microcontrolador para conversão do valor de T em segundos e para considerar apenas o tempo de ida do pulso sonoro ($k = 2 \times 3{,}2 \times 10^{-6}$).

Através de uma sub-rotina do programa do microcontrolador é possível controlar o sensor ultrassônico, efetuando as medições e transmitindo-as ao PC depois de processadas. O transdutor da Polaroid tem um tempo mínimo de segurança entre medições para garantir o amortecimento das vibrações da pastilha entre o disparo para envio do pulso ultrassônico e o recebimento do pulso ultrassônico do eco. Para possibilitar medições de distâncias inferiores a 40 cm é necessário utilizar a função BINH (*Blanking Inhibit*), disponibilizada pelo circuito transdutor através sinal BINH para o controle do tempo mínimo (atraso) denominado T_BINH.

A metodologia adotada para avaliação da viabilidade do sensor baseia-se na comparação de alguns parâmetros estatísticos calculados sobre dois conjuntos de medidas. O primeiro conjunto de medições foi realizado em laboratório em relação a

um anteparo plano de madeira. O segundo conjunto de medições foi realizado em relação a um anteparo vegetal (planta), simulando condições físicas de superfícies encontradas em uma plantação. A Figura 46 mostra a configuração da instrumentação para o segundo conjunto de medidas realizado em relação ao anteparo vegetal.

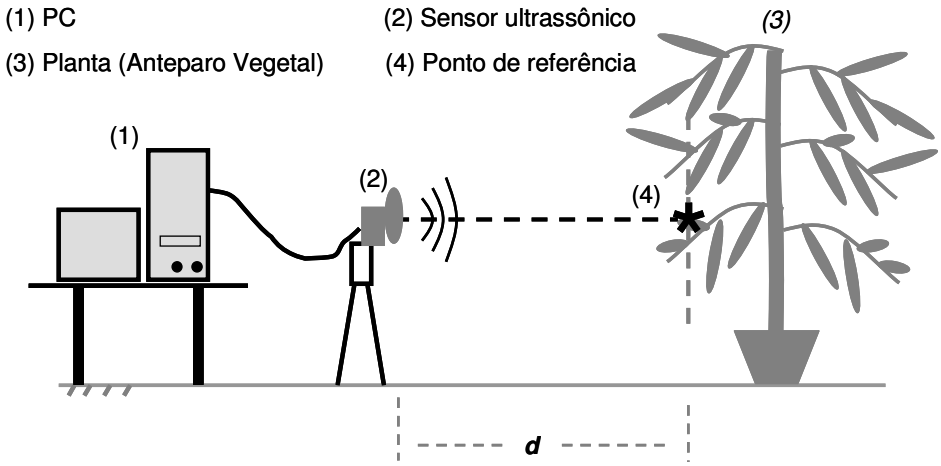


Figura 46 - Medições de distâncias (*d*) com o sensor ultrassônico em relação a um anteparo vegetal.

Inicialmente, as medições com a instrumentação ilustrada na Figura 46 foram realizadas com a planta estática, e, posteriormente, com o anteparo vegetal em movimento sendo rotacionada em pequenos ângulos de um lado para o outro. Em ambos os casos, foram efetuados medições entre o sensor e o anteparo (madeira e planta) em diferentes distâncias. Em cada uma dessas situações foi variado o tempo de atraso do sinal BINH (T_BINH) entre os valores de 0,6 ms e 1,2 ms.

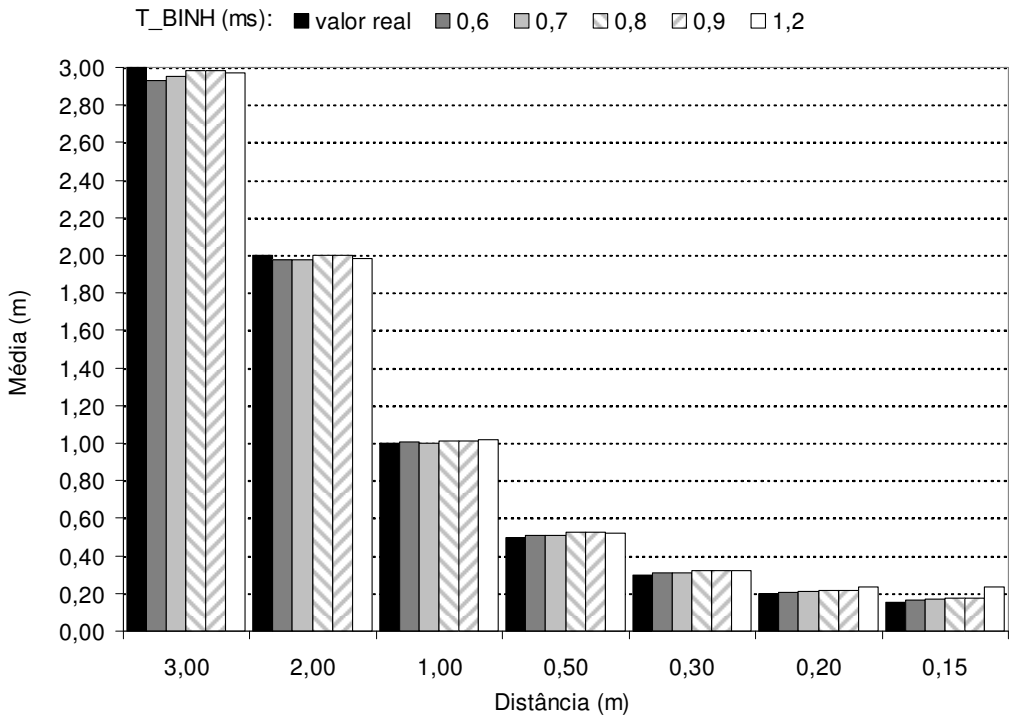
As distâncias *d* entre o anteparo e a pastilha do transdutor foram determinadas previamente com o uso de uma trena. Para a estimativa prévia das

distâncias no segundo conjunto de medidas foi adotado como referência um ponto na parte frontal da planta.

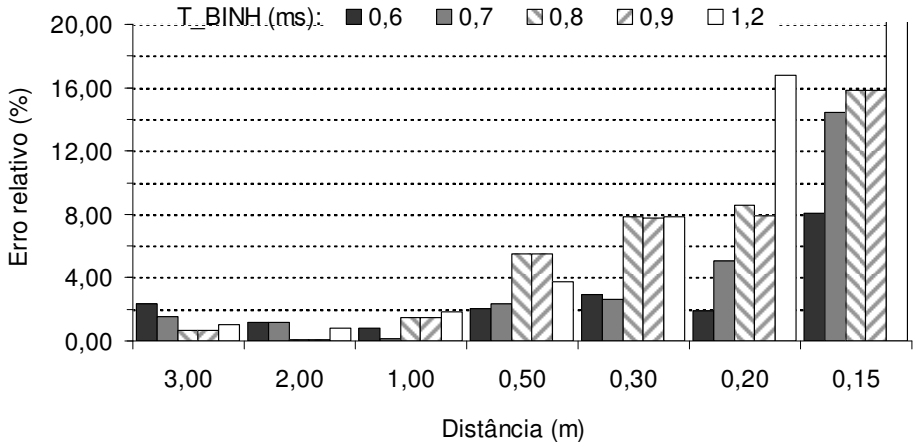
6.2.2 Medições de distância com o sensor ultrassônico

As distâncias escolhidas para avaliar o sensor foram definidas pelas distâncias mais comuns em que ele será empregado no RAM, variando entre 0,15 m e 3,00 m. Para cada distância medida com trena foram realizadas em torno de 40 (quarenta) medições com o sensor ultrassônico para diferentes valores de T_BINH.

As medições de referência efetuadas entre o sensor e o anteparo plano de madeira foram realizadas para os valores de T_BINH iguais a 0,6 ms , 0,7ms , 0,8 ms , 0,9 ms e 1,2 ms. Para cada distância e para cada T_BINH foram selecionadas 30 (trinta) medições após a filtragem dos valores discrepantes. A Figura 47 apresenta os gráficos com a média e com o erro relativo (VUOLO, 1991) das 30 medições restantes, efetuadas para cada distância e para cada T_BINH adotado.



(a)



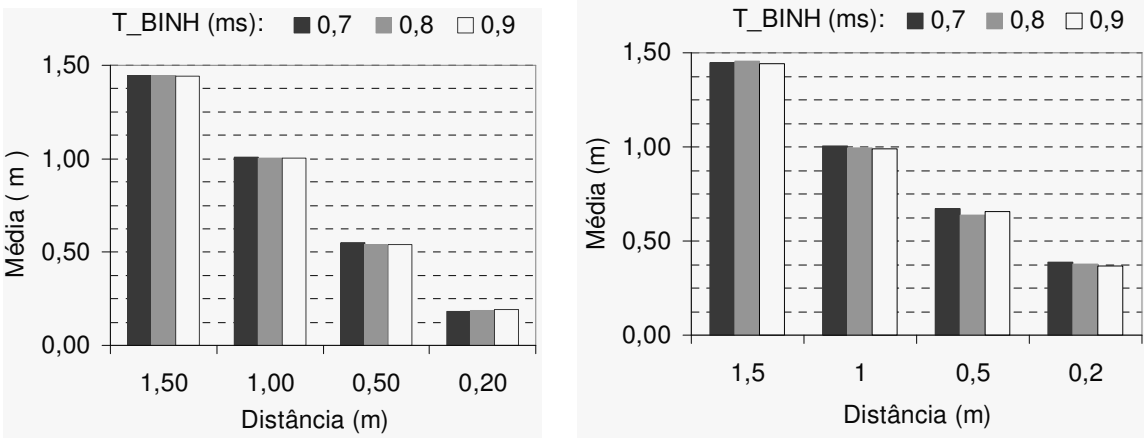
(b)

Figura 47 - Medições de referência realizadas em relação ao anteparo plano de madeira: médias (a) e erros relativos (b)

Como pode ser observada na Figura 47, as medições efetuadas pelo sensor apresentam pequeno erro relativo para medidas superiores a 0,25 m. O valor de T_BINH influencia o comportamento do erro, sendo mais notada essa influência para medidas inferiores a 0,50 m. Para essas distâncias o erro aumenta com o aumento

de T_BINH. Observando-se os erros relativos, as melhores medições (erro relativo menor) acima de 1,00 m foram efetuadas com T_BINH igual a 0,8 ms e igual a 0,9 ms. Entretanto, as melhores medições abaixo de 1,00 m foram efetuadas com T_BINH igual a 0,6 ms e igual a 0,7 ms. As medições realizadas na distância de 0,15 m possuem maior erro relativo com valor superior a 8,00%, sendo que para T_BINH igual a 1,2 ms o erro supera 20,00%.

As medições efetuadas em relação ao anteparo vegetal passaram pelos mesmos tratamentos para a determinação dos parâmetros estatísticos utilizados na caracterização em relação ao anteparo plano de madeira. Entretanto, foram consideradas as medições para as distâncias 1,50 m , 1,00 m , 0,50 m e 0,20 m com T_BINH com valores iguais a 0,7 ms , 0,8 ms e 0,9 ms. A Figura 48 apresenta os gráficos com as médias das medições realizadas com a planta estática e em movimento, em função da distância e para cada T_BINH.



(a) (b)

Figura 48 - Média das medições em relação ao anteparo vegetal: (a) anteparo parado e (b) anteparo em movimento.

Como pode ser observado na Figura 48, a média das medições efetuadas pelo sensor com o anteparo vegetal estático apresentam diferenças significativas em relação às medições realizadas com o anteparo vegetal em movimento para as distâncias de 0,50 m e 0,20 m. Para essas distâncias e com a planta em movimento, as médias apresentam valores superiores a 0,10 m em relação aos valores reais. Para as mesmas distâncias e com a planta estática, os valores aproximam-se mais do valor real. Em ambos os casos, quando esses resultados são comparados com os resultados apresentados na Figura 47 para as distâncias de 0,50 m e 0,20 m, observa-se a mesma tendência dos valores médios das medições serem maiores que o valor real. Nesse último caso, as médias das medições efetuadas em relação ao anteparo vegetal estático são ligeiramente superiores às médias das medições efetuadas em relação ao anteparo plano de madeira. As médias das medições efetuadas em relação ao anteparo vegetal em movimento são significativamente superiores às médias das medições efetuadas em relação ao anteparo plano de madeira.

Observando-se a Figura 47 e a Figura 48, verifica-se para as distâncias de 1,00 m e 1,50 m que as médias de ambos os experimentos apresentam valores próximos ao valor real, sendo ligeiramente inferiores aos valores reais para a distância de 1,50 m.

Nota-se pela Figura 48 que o período T_BINH com valores iguais a 0,7 ms , 0,8 ms e 0,9 ms influencia pouco nas diferenças das médias das medições em relação ao valor real. Essa tendência também é observada para os mesmos valores do período T_BINH nas medições de referência em relação ao anteparo plano de madeira apresentadas na Figura 47.

A Figura 49 apresenta os gráficos com os valores do erro relativo em função da distância e para cada período T_BINH das medições com o anteparo vegetal estático e em movimento.

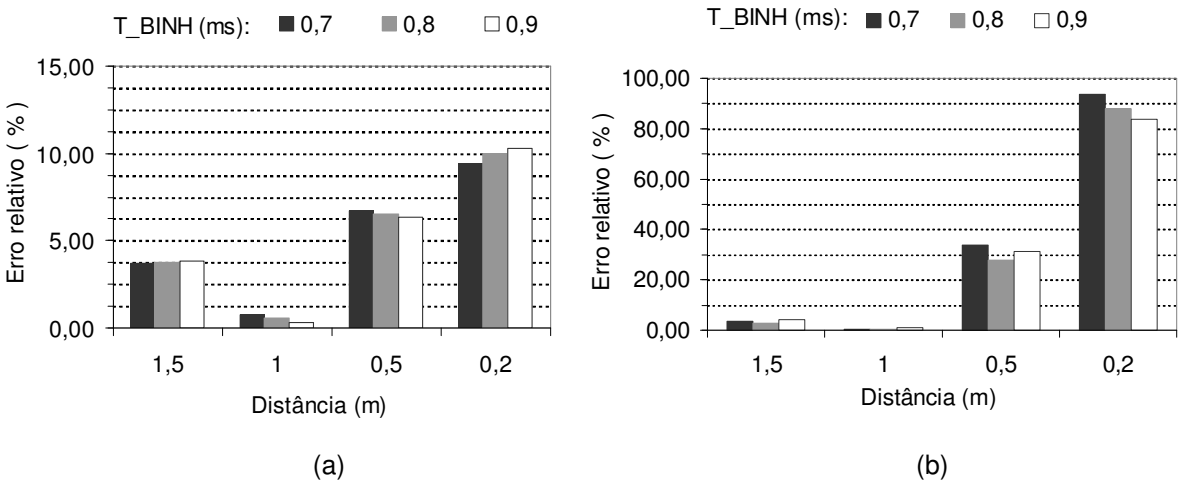


Figura 49 - Erro relativo das medições em relação ao anteparo vegetal: (a) anteparo parado e (b) anteparo em movimento.

Considerando-se a Figura 47 e a Figura 49, verifica-se que os valores do erro relativo das medições com o anteparo vegetal tiveram um aumento significativo em relação aos encontrados na caracterização de referência em relação ao anteparo plano de madeira, principalmente para distâncias pequenas (abaixo de 0,50 m). Essa diferença é mais acentuada para as medições realizadas em relação ao anteparo vegetal em movimento, sendo que, nesse caso, esses erros superaram 20,00% para a distância de 0,50 m e 80,00% para a distância de 0,20 m. Para as distâncias de 1,00 m e 1,50 m os erros relativos das medições realizadas em relação ao anteparo vegetal ficaram abaixo de 10%.

De forma geral, os resultados anteriores mostram a influência da superfície do anteparo. As medidas de grandes distâncias (1,00 m a 1,5 m) possuem pequeno erro considerando-se as condições às quais o sensor foi submetido. Para essas distâncias, o pulso ultrassônico é refletido com maior eficiência, mesmo considerando a concentração de folhas, que é baixa, e o tipo das folhas da planta. O aumento da distância do sensor à planta causa um aumento da área de seção de choque, isto é, da área de incidência, do pulso ultrassônico. Esse fato se deve ao formato do pulso do sensor, que segundo o manual do fabricante, é emitido com uma angulação de ± 15º formando um cone e não um cilindro de base circular. Isso torna maior a probabilidade de um pulso ser refletido por alguma folha da parte frontal para distanciais maiores. Por outro lado, a elevação do erro apresentado pelas medições de pequenas distâncias é causada, provavelmente, pela reflexão do pulso ultrassônico ter acontecido em folhas mais internas ou externas em relação ao ponto de referência de medida na planta.

6.3. Simulação e análise de rede CAN com dispositivos para a arquitetura robótica proposta

Atualmente, as redes de campo têm sido aplicadas em sistemas distribuídos propiciando maior eficiência, flexibilidade e confiabilidade em relação às topologias centralizadas, principalmente pela redução de fios e pela distribuição do controle. Apesar das vantagens e dos potenciais, a utilização de redes de comunicação torna a análise e o projeto de um sistema de controle via rede (*Networked Control System - NCS*) mais complicado (YANG, 2006).

Pesquisas e desenvolvimentos recentes têm buscado metodologias e ferramentas para auxiliar na análise e no projeto de tais sistemas (OTHMAN et al., 2006; YANG, 2006). Um dos desafios fundamentais nesta área é o desenvolvimento de modelos e ferramentas de análise para prever o comportamento dos tempos envolvidos na comunicação e para simular a operação do NCS (LIAN, MOYNE e TILBURY, 2002). Os parâmetros característicos de redes de comunicação associados aos parâmetros do sistema de percepção, computação e atuação influenciam diretamente no desempenho do sistema de NCS (LIAN, MOYNE e TILBURY, 2002).

Com relação às redes baseadas no CAN, os trabalhos Tindell e Burns (1994) e Tindell, Burns e Wellings (1995) tornaram-se referência como o estado da arte em análise desse tipo de rede. A partir destes trabalhos, diversas adaptações e variações desses métodos e equações de análise de rede CAN têm sido propostas (FRUTOS et al. 2006; GODOY, 2007).

Godoy (2007) apresenta o desenvolvimento e a aplicação de uma metodologia de análise de rede CAN para máquinas agrícolas. A metodologia é implementada em um programa computacional de simulação para auxiliar o projeto e o dimensionamento de parâmetros de rede CAN como, por exemplo, velocidade de transmissão do barramento, tamanho das mensagens de dados, esquema de prioridade das mensagens e período de amostragem dos dispositivos. O programa computacional desenvolvido por Godoy (2007) é utilizado no projeto e na análise da rede embarcada baseada no protocolo CAN para integrar dispositivos do RAM.

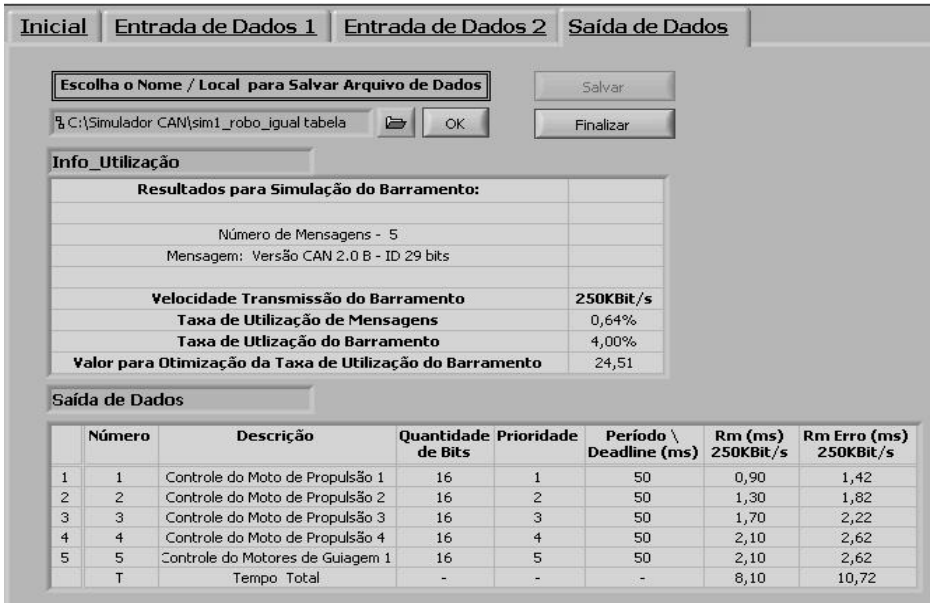
6.3.1 Programa de simulação para rede CAN

O programa computacional é composto basicamente por três telas de dados, sendo duas telas relacionadas à entrada de dados a ser realizada pelo usuário e uma tela referente à saída de dados que mostra os resultados da simulação. Outras características importantes do programa desenvolvido são funções como, por exemplo, validação dos dados de entrada (o que não deixa o usuário continuar a execução do programa se existir dados de entrada incorretos) e geração de relatórios dos dados de saída (geração das tabelas de saída de dados em planilha eletrônica). As telas do programa desenvolvido são mostradas nas Figura 50 e Figura 51.

(a) (b)

Figura 50 - Telas de entrada de dados do programa de simulação de rede CAN: (a) Entrada de Dados 1; (b) Entrada de Dados 2.

Nas telas de entrada de dados da Figura 50 são definidos os parâmetros de entrada que são utilizados pelo modelo de equações para gerar os dados de saída relacionados à rede CAN. Na tela Entrada de Dados 1, Figura 50a, são definidas as configurações principais da rede CAN como, por exemplo: versão do protocolo CAN (CAN 2A, CAN 2B), número de mensagens de dados e velocidade de transmissão da rede. No caso do campo velocidade de transmissão, pode ser selecionado mais de um valor para análise, sendo ele fixo ou definido pelo usuário. Na tela Entrada de Dados 2, Figura 50b, de acordo com o número de mensagens selecionado anteriormente, são definidos os campos que descrevem o conjunto de mensagens da rede. Para cada mensagem são requeridos os seguintes dados: descrição da função, quantidade de bits de dados, prioridade e o período de amostragem em milisegundos (ms).



	Número	Descrição	Quantidade de Bits	Prioridade	Período \ Deadline (ms)	Rm (ms) 250KBit/s	Rm Erro (ms) 250KBit/s
1	1	Controle do Moto de Propulsão 1	16	1	50	0,90	1,42
2	2	Controle do Moto de Propulsão 2	16	2	50	1,30	1,82
3	3	Controle do Moto de Propulsão 3	16	3	50	1,70	2,22
4	4	Controle do Moto de Propulsão 4	16	4	50	2,10	2,62
5	5	Controle do Motores de Guiagem 1	16	5	50	2,10	2,62
	T	Tempo Total	-	-	-	8,10	10,72

Figura 51- Tela de saída de dados do programa de simulação de rede CAN

Na tela de Saída de Dados, Figura 51, são mostrados os resultados obtidos para a simulação da rede CAN. São mostrados resultados como, por exemplo, o tempo de transmissão das mensagens para condições normais e condições de erro de transmissão. Também são mostrados três dados relacionados ocupação da rede, que são:

- *Taxas de utilização de mensagens:* indicativo do tempo de espera para que as mensagens sejam transmitidas em relação aos períodos de amostragem. Esse valor é calculado pela somatória dos valores porcentuais de todas as relações entre tempo de espera para transmissão das mensagens e de seus períodos de amostragem. A partir desse valor pode-se identificar a possibilidade de formação de filas de transmissão nos nós da rede;

- *Taxa de utilização do barramento:* indicativo da carga de dados que trafegam pelo barramento diante de sua capacidade. Esse valor é calculado pela somatória dos valores porcentuais de todas as relações entre tempo de transmissão das mensagens e de seus períodos de amostragem. Permite identificar se a rede está sobrecarregada ou ociosa;

- *Otimização da taxa de utilização do barramento:* corresponde ao valor pelo qual se podem dividir todos os períodos de amostragem das mensagens para otimizar e atingir a taxa de utilização máxima do barramento (o maior valor possível para a taxa de utilização da rede) com o cumprimento dos requisitos temporais das mensagens.

6.3.2 Metodologia de análise para rede do RAM

Para a utilização do modelo matemático apresentado em Godoy (2007), implementado no programa de simulação para rede CAN, são evidenciados na Tabela 2 os tipos de parâmetros que podem ser relacionados com aplicações agrícolas de forma a direcionar a utilização desse modelo ao estudo do comportamento da rede CAN e da influência de parâmetros em diversas situações.

Tabela 2- Dados para a utilização do modelo matemático da rede CAN

Dados de Entrada	Velocidade de comunicação (ex. 125 kbits/s 250 kbits/s)
	Versão do protocolo CAN (CAN 2.0 A / CAN 2.0 B)
	Período de amostragem das mensagens dos equipamentos
	Tamanho (bytes) das mensagens dos equipamentos
Dados de Saída	Tempo de transmissão das mensagens
	Taxa utilização do barramento
	Otimização / simulação de comportamento
	Definição do esquema de prioridades

A partir da definição dos parâmetros de entrada, as equações que definem o modelo são utilizadas para obter os dados de saída apresentados na Tabela 2. Através da análise desses dados de saída pode-se determinar se os parâmetros de entrada conduzem a um bom funcionamento da rede, ou senão, orientar uma nova escolha de entrada de dados. Essa escolha pode ser realizada de acordo com as especificações de uma norma da aplicação desejada (por exemplo: ISO11783). Essa

estrutura iterativa origina uma metodologia de análise de rede CAN, com um fluxograma de utilização mostrada Figura 52.

Figura 52- Fluxograma baseado a metodologia de análise CAN. Fonte: adaptado de Godoy (2007).

A metodologia de análise pode ser aplicada para estudo de viabilidade de implementação e do comportamento de redes CAN configuradas sob os parâmetros fixados por normas de protocolo. Para o caso da rede CAN sob a norma ISO11783, os dados levados em consideração são a velocidade de transmissão de 250Kbits/s, e versão CAN 2.0 B para tipo de mensagem, além de outras características definidas pelas Partes 4 a 10 da norma citada (SOUSA, 2002). Para o caso da aplicação da rede embarcada do robô agrícola móvel, outros parâmetros são testados e analisados para auxiliar o dimensionamento de parâmetros e o estudo do comportamento da rede CAN projetada com os dispositivos do RAM.

Os dispositivos de percepção, computação e atuação propostos para implementação da arquitetura híbrida são listados na Tabela 3 com os respectivos tamanhos de dados o período de amostragem (período de geração da mensagem).

Tabela 3– Dispositivos para a rede CAN referentes à arquitetura híbrida proposta.

Nº	Descrição das Mensagens	Dados (bits)	Período (ms)
1	Controle do motor de propulsão 1	16	50
2	Controle do motor de propulsão 2	16	50
3	Controle do motor de propulsão 3	16	50
4	Controle do motor de propulsão 4	16	50
5	Controle do motor de guiagem 1	16	50
6	Controle do motor de guiagem 2	16	50
7	Gerenciamento/PC embarcado para dispositivo	16	50
8	Gerenciamento/dispositivos para PC embarcado	16	50
9	Velocidade/odômetro do motor de propulsão 1	16	100
10	Velocidade/odômetro do motor de propulsão 2	16	100
11	Velocidade/odômetro do motor de propulsão 3	16	100
12	Velocidade/odômetro do motor de propulsão 4	16	100
13	Deslocamento angular/potenciômetro do motor de guiagem 1	16	100
14	Deslocamento angular/potenciômetro do motor de guiagem 2	16	100
15	Posicionamento relativo/bússola digital	16	100
16	Posicionamento relativo/DGPS	16	100
17	Distância de linhas de plantio/conjunto de sensores ultrassônicos do lado direito	64	300
18	Distância de linhas de plantio/conjunto de sensores ultrassônicos do lado esquerdo	64	300
19	Posicionamento absoluto/giroscópio e acelerômetro	32	500
20	Gerenciamento/PC embarcado para dispositivos	32	250
21	Gerenciamento/ dispositivos para PC embarcado	32	1000
22	Visão de linhas de plantio/scanner laser	64	1000

A utilização dos dispositivos listados na Tabela 3 na arquitetura robótica proposta é discutida no Item 6.1 e ilustrada na Figura 42. Os tamanhos em bits dos dados e os períodos de amostragem para tais dispositivos foram definidos a partir das saídas disponíveis em dispositivos comerciais. Blackmore e Griepentrong (2006) apresentam uma sistematização sobre informações dos sensores utilizados em sistemas autônomos agrícolas que inclui alguns dos dispositivos listados na Tabela 3.

Os valores postos na Tabela 3 foram definidos considerando os casos extremos, com os tamanhos de dados e os períodos de amostragem menores possíveis para os dispositivos listados. Com relação aos sensores de distância de linhas de plantio foi considerada a aplicação de conjuntos de sensores ultrassônicos. Devido ao tempo para realização da medição mais o tempo de espera para de amortecimento das vibrações da pastilha há um atraso em torno de 50 ms entre uma leitura e outra de cada sensor ultrassônico. Assim, foram considerados 300 ms para período de amostragem de cada conjunto. Nesse caso, em cada lateral poderiam ser utilizados até seis sensores ultrassônicos junto aos comportamentos *beirar linha direita* e *beirar linha esquerda* da arquitetura robótica proposta.

6.3.3 Resultados da simulação para rede do RAM

O conjunto de mensagens da Tabela 3 é utilizado em três tipos de simulação diferentes. Para essas simulações foram considerados os seguintes parâmetros de entrada: velocidades de transmissão do barramento de 125 e 250 kbits/s, versão CAN 2B (Campo Identificado com 29 bits e Campo de Dados com até 64 bits ou 8

bytes), configuração de prioridades das mensagens de acordo com a primeira coluna (Nº) da Tabela 3.

No primeiro tipo de simulação, além dos parâmetros de entrada citados no parágrafo anterior, foram utilizados como entradas o número de bits dos dados de cada mensagem e o período de amostragem (geração de mensagens) de todos os dispositivos, colocados respectivamente na terceira e na quarta colunas da Tabela 3. Os primeiros resultados obtidos com os dados de entrada da Tabela 3 são mostrados nas Tabela 4 e Tabela 5. Na Tabela 4 são mostrados dados referentes aos tempos mais extremos de espera (se todos os outros dispositivos desejam transmitir ao mesmo momento) para transmissão das mensagens para cada dispositivo.

Como pode ser visto na Tabela 4, todos os tempos de transmissão são inferiores aos períodos de amostragem correspondente a cada dispositivo. No pior caso, o oitavo dispositivo, na taxa de 125 kbit/s e sob condição de erro deve aguardar 8,83 ms para transmitir, se todos os outros dispositivos fossem transmitir no mesmo momento. Nesse caso, esse dispositivo teria que aguardar todos os outros dispositivos com maior prioridade transmitirem. Esse tempo corresponde a 17,66 % (dezessete por cento e sessenta e seis décimos) de seu período de amostragem.

Na situação mais crítica para o dispositivo com maior prioridade, o vigésimo segundo dispositivo, este teria que aguardar 21,15 ms para transmitir um dado na taxa de 125 kbit/s e sob condição de erro, se todos os outros dispositivos fossem transmitir no mesmo momento. Esse tempo corresponde a aproximadamente 2,16% de seu período de amostragem.

Tabela 4– Tempo de transmissão para o conjunto de mensagens da rede CAN.

	Caracterização do Conjunto de Mensagens			**Tempo de Transmissão sob Condições Normais (ms)**		**Tempo de Transmissão sob Condições de Erro (ms)**	
Nº	**Quantidade de Bits**	**Prioridades**	**Período (ms)**	**Velocidade 125 kbit/s**	**Velocidade 250 kbit/s**	**Velocidade 125 kbit/s**	**Velocidade 250 kbit/s**
1	16	1	50	2,18	1,14	3,23	1,66
2	16	2	50	2,98	1,54	4,03	2,06
3	16	3	50	3,78	1,94	4,83	2,46
4	16	4	50	4,58	2,34	5,63	2,86
5	16	5	50	5,38	2,74	6,43	3,26
6	16	6	50	6,18	3,14	7,23	3,66
7	16	7	50	6,98	3,54	8,03	4,06
8	16	8	50	7,78	3,94	8,83	4,46
9	16	9	100	8,58	4,34	9,63	4,86
10	16	10	100	9,38	4,74	10,43	5,26
11	16	11	100	10,18	5,14	11,23	5,66
12	16	12	100	10,98	5,54	12,03	6,06
13	16	13	100	11,78	5,94	12,83	6,46
14	16	14	100	12,58	6,34	13,63	6,86
15	16	15	100	13,38	6,74	14,43	7,26
16	16	16	100	14,18	7,14	15,23	7,66
17	64	17	300	15,46	7,78	16,99	8,54
18	64	18	300	16,74	8,42	18,27	9,18
19	32	19	500	17,7	8,9	19,23	9,66
20	32	20	250	18,66	9,38	20,19	10,14
21	32	21	1000	19,62	9,86	21,15	10,62
22	64	22	1000	19,62	9,86	21,15	10,62
Total				238,68	120,44	264,62	133,41

Na Tabela 5 são mostrados dados referentes às taxas de utilização e os valores para otimização da taxa de utilização. Esses parâmetros e seus significados são descritos no Item 6.3.1.

Tabela 5– Resultados obtidos para a rede CAN na primeira simulação.

Velocidade de transmissão do barramento	**125 kbit/s**	**250 kbit/s**
Taxa de utilização de mensagens	3,64%	1,82%
Taxa de utilização do barramento	20,85%	10,43%
Valor para otimização da taxa de utilização do barramento	4,7	9,4

Pela Tabela 5 verifica-se que, as taxas de utilização de mensagens e taxas de utilização do barramento são pequenas, e confirmam as tendências observadas pela Tabela 4. O tempo de utilização das mensagens representa apenas 3,64% para 125 kbit/s e 1,82% para 250 kbit/s, indicando que o tempo de espera das mensagens para serem transmitidas é pequeno em relação aos períodos de amostragem. As taxas de utilização do barramento de 20,85% para 125 kbit/s e 10,43% para 250 kbit/s indicam que existe possibilidade para aumentar a carga ou tráfego de mensagens na rede.

Para o segundo tipo de simulação foram alterados os valores dos períodos de amostragem, de 50ms para 100ms, para os dispositivos 1 a 8 da Tabela 3. Com a definição desses parâmetros foram obtidos os resultados mostrados na Tabela 6.

Tabela 6– Resultados obtidos para a rede CAN na segunda simulação.

Velocidade de transmissão do barramento	**125KBit/s**	**250KBit/s**
Taxa de utilização de mensagens	2,62%	1,31%
Taxa de utilização do barramento	14,45%	7,23%

Na Tabela 6, os resultados da segunda simulação demonstram que o parâmetro período de amostragem influi sobre os valores das taxas de utilização calculadas. Isso pode ser visto, por exemplo, para a velocidade de 250Kbits/s, onde

o aumento do período de amostragem dos dispositivos 1 a 8 da Tabela 3 conduziu a uma diminuição da taxa de utilização do barramento de 10,43% para 7,23%.

Para o terceiro tipo de simulação foram alterados os valores das quantidades de bits de dados da mensagem, de 16 para 64 bits, para os dispositivos 1 a 16 da Tabela 3. Com a definição desses parâmetros, obtiveram-se os resultados mostrados na Tabela 7.

Tabela 7– Resultados obtidos para a rede CAN na terceira simulação.

Velocidade de transmissão do barramento	**125 kbit/s**	**250 kbit/s**
Taxa de utilização de mensagens	12,86%	6,43%
Taxa de utilização do barramento	32,37%	16,19%
Tempo total de transmissão do conjunto de mensagens sob condições normais	350,04 ms	383,66ms
Tempo total de transmissão do conjunto de mensagens sob condições de erro	176,12ms	192,93ms

Os resultados do terceiro tipo de simulação, mostrados na Tabela 7, permitem identificar que o parâmetro quantidade de bits de dados da mensagem, além de influenciar os valores das taxas de utilização calculadas, também influi sobre os valores dos tempos de transmissão das mensagens na rede. Como mostrado pelos dados referentes à velocidade de 250 kbit/s, o aumento da quantidade de bits de dados conduziu a um aumento das taxas de utilização, de 10,43% para 16,19% para a taxa de utilização do barramento e de 1,82% para 6,43% para a taxa de utilização das mensagens. Em relação ao tempo de transmissão das mensagens na rede, o aumento da quantidade de bits provocou um aumento do tempo de resposta da rede. Como pode ser visto, com a velocidade de 125kbit/s, o tempo de transmissão

total do conjunto de mensagens aumentou de 120,44 ms para 176,12 ms para condições normais e de 238,68 ms para 350,04 ms para operação da rede sob condições de erro.

6.4. Considerações gerais sobre a implementação da arquitetura robótica no RAM

Os resultados dos experimentos para caracterização da aplicação do sensor ultrassônico mostram que as medições de distância superiores a 0,50 m possuem pequeno erro relativo considerando-se as condições às quais o sensor foi submetido. Na condição de movimento entre relativo entre a planta e o sensor, os erros relativos das medições são inferiores a 10% indicando que esse tipo de sensor pode ser aplicado na arquitetura robótica junto aos comportamentos *evitar obstáculo*, *beirar linha direita* e *beirar linha esquerda*, para distâncias superiores a 0,50 m.

Sob o ponto de vista do controle via rede, os períodos de amostragem e o tamanho dos dados em bits definidos para cada conjunto de sensores ultrassônicos não afetam o desempenho da rede e do sistema de controle. Essa informação é confirmada considerando-se que para o caso mais crítico (terceiro tipo de simulação), no qual utilizou-se taxa de comunicação de 250 kbit/s, a quantidade de bits de dados máxima para todas as mensagens (64 bits) e condição de erro, a taxa de utilização do barramento é de 16,19% e a taxa de utilização das mensagens é 6,43%. Assim, nota-se que existe folga para diminuir os períodos de amostragem e

para inserção de outros grupos de sensores e mensagens para ampliação do sistema de percepção da arquitetura robótica proposta.

Com a possibilidade de ampliação do sistema de percepção sob o ponto de vista da capacidade da rede CAN, o comportamento *identificar objeto* pode ser enriquecido através da implementação de outros sub-comportamentos além dos implementados no mini-robô, *evitar obstáculo*, *beirar linha esquerda* e *beirar linha direita*. Sob o ponto de vista da estrutura da arquitetura proposta, isso é possível com a criação de novos controladores difusos para utilizar esses sensores e com a adaptação do árbitro-α para selecionar em cada instante de amostragem entre os sub-comportamentos disponíveis. Também, é possível inserir novos comportamentos além dos comportamentos *seguir linha, identificar objeto* e *manter rota* utilizados no mini-robô de acordo com o contexto de aplicação apresentado. A adaptação do árbitro-β é realizada pela criação de novas variáveis difusas e da ampliação das regras difusas que compõem a base de conhecimento.

Os resultados da caracterização do sensor ultrassônico mostram que para distâncias inferiores a 0,50 m as medições apresentam erro grande, principalmente para condição de movimento relativo entre a planta e o sensor. Nessa última condição, o erro relativo supera 40% e pode inviabilizar sua aplicação na arquitetura robótica junto aos comportamentos *evitar obstáculo*, *beirar linha direita* e *beirar linha esquerda*. Uma solução para esse fato é a utilização de outro tipo de sensor que permita medir distâncias inferiores a 0,50 m. Como citado neste item (Item 6.4) e discutido no Item 5.7, existe não só a possibilidade de ampliação da rede CAN como também, a arquitetura permite inserção de novos comportamentos e sub-comportamentos. Dessa forma, novas variáveis difusas podem ser elaboradas para

adequar as leituras desses novos sensores para o contexto da lógica difusa, e assim, viabilizar a sua integração à arquitetura robótica proposta.

A metodologia utilizada para caracterização do sensor ultrassônico pode ser utilizada para avaliação de outros sensores para medição de distâncias em relação a linhas de plantio, sendo que alguns candidatos são: sensor capacitivo, sensor IR e sensor laser. Uma característica importante identificada nos experimentos para caracterização é que a angulação de ± 15º em forma de cone do pulso emitido torna maior a probabilidade de um pulso ser refletido por folhas da linha de plantio. Assim, o efeito das lacunas existentes entre as folhas da planta, que poderiam permitir a passagem do sinal de medição, é minimizado pela reflexão do pulso ultrassônico acontecer em folhas mais internas ou externas dentro do volume do cone. Outra possibilidade é a aplicação dos sensores em um arranjo com várias unidades. Isso pode permitir a aquisição de medidas em vários pontos da planta, as quais seriam processadas para gerar um valor mais apurado da distância em cada instante de amostragem. Esse fato pode ser avaliado pela metodologia aplicada para caracterização do sensor ultrassônico.

A análise por simulação da rede CAN permite observar que existe viabilidade para a aplicação da rede CAN junto à arquitetura robótica proposta. Isso é confirmado pelos baixos valores referentes às taxas de utilização da rede CAN mostrados na Tabela 5, na Tabela 6 e na Tabela 7. Tanto para condição normal como para condição de erro, o sistema de controle para a rede CAN do RAM apresenta um tempo de resposta satisfatório para cumprir o requisito temporal de cada mensagem, ou seja, o tempo de transmissão de cada uma das mensagens, sob condições normais e de erro, é menor que o seu período de amostragem. Esses dados demonstram que o sistema apresenta capacidade para inserção de novos

dispositivos, aumento da carga de dados (enviados por mensagens) e aumento do tráfego (número) de mensagens na rede CAN.

7. CONCLUSÕES

No presente trabalho uma arquitetura robótica baseada em comportamentos é desenvolvida para permitir guiagem e navegação de RAM´s e VAA´s. Regras difusas são utilizadas para compor e coordenar comportamentos primitivos e comportamentos complexos. Foi apresentada a implementação e a simulação da arquitetura em um mini-robô, a avaliação e a caracterização de sensores para o módulo perceptivo da arquitetura e a aplicação de um método de análise da rede de comunicação digital baseada no protocolo CAN para sistemas de controle robóticos. Experimentos foram realizados para avaliar a capacidade de operação dos diversos módulos componentes da arquitetura híbrida proposta na realização da guiagem e da navegação autônomas. Também foram executados experimentos para avaliar especificamente os comportamentos difusos e o processo de arbitragem baseado em árbitros difusos e não difusos. Os resultados mostraram a viabilidade da abordagem proposta diante de tarefas e de situações comuns ao ambiente agrícola e simuladas nos experimentos.

7.1. Principais contribuições

Constatou-se que a utilização de controladores difusos, hierárquicos e descentralizados proporcionou modularidade à arquitetura e flexibilidade para inserção de novos módulos, dispositivos e comportamentos. A estrutura hierárquica

desses controladores simplificou o processo de desenvolvimento e implementação da arquitetura robótica. O processo de arbitragem empregado mostrou-se como um método de simples implementação, mas que viabiliza a coordenação e a composição de comportamentos mais complexos com flexibilidade para inserção de novos comportamentos. A possibilidade de um ou mais árbitros, constituídos ou não por controladores difusos, serem empregados para coordenação dos comportamentos e a possibilidade do emprego de comportamentos difusos com diversos níveis de abstração são pontos fortes arquitetura proposta. Além disso, a arquitetura possibilita que diferentes comportamentos construídos a partir da aplicação de diferentes métodos possam compartilhar o controle robótico.

A proposta se beneficia do emprego da lógica difusa, principalmente pela sua eficácia em lidar com diferentes graus de incerteza e de traduzir a relação robô-ambiente para estruturas com expressões linguísticas mais intuitivas para o ser humano, facilitando assim a construção do sistema de controle robótico. A sua utilização proporcionou uma suavização na transição entre os comportamentos robóticos, pois entre regiões onde apenas um comportamento prevalece, há uma região de transição em que os comportamentos alternam a ativação até que o novo contexto fique claramente identificado. Essa característica confere robustez à arquitetura, pois a alternância entre comportamentos na região de transição não prejudica a navegação e promove a imunidade a ruídos aleatórios que poderiam causar ações indesejáveis.

A identificação do estado-da-arte das metodologias e tecnologias aplicadas à robôs móveis em outras áreas que não a área agrícola, assim como a identificação estado-da-arte das metodologias e tecnologias máquinas agrícolas autônomas permitiu conjugar conhecimentos e propor um arquitetura que aproveita-se dos

conhecimentos mais avançados de outras áreas e das peculiaridades dos sistemas agrícolas. Assim, contribui-se com o estado-da-arte da pesquisa em VAA's e RAM's através de uma proposta formal para projeto e construção de sistemas robóticos para guiagem e pa navegação baseados em comportamentos difusos. A arquitetura proposta destaca-se na aplicação em tarefas agrícolas na medida em que potencializa a utilização de comportamentos robóticos e de dispositivos, principalmente de percepção, voltados para aplicações agrícolas e que exploram as características estruturais do ambiente agrícola.

A avaliação e caracterização de sensores permitiram a determinação de condições operacionais para a utilização desses sensores em VAA´s e em RAM´s. Em especial, os resultados dos experimentos para caracterização do sensor ultrassônico mostraram a viabilidade de sua aplicação em ambiente agrícola e permitiram definir as condições para sua utilização. Como contribuição aos trabalhos de pesquisa na área agrícola, evidenciou-se as distâncias e os erros de medição para condição de movimento entre relativo entre a planta e o sensor. Dessa forma, pode-se concluir que esse tipo de sensor pode ser aplicado na arquitetura robótica junto aos comportamentos relacionados com identificação de distâncias em relação linhas de plantio. Além disso, uma característica importante identificada nos experimentos para caracterização é que a angulação na difusão do sinal emitido pelo sensor torna maior a probabilidade do sinal ser refletido por folhas da linha de plantio. Assim, o efeito das lacunas existentes entre as folhas da planta, que poderiam permitir a passagem do sinal de medição, é minimizado pela reflexão do sinal disperso, nas folhas mais internas ou externas, dentro da área relacionada com a angulação.

A análise por simulação da rede CAN permitiu observar que existe viabilidade para a aplicação da rede CAN junto à arquitetura robótica proposta. Além disso, a aplicação do método de análise baseado em um modelo matemático para auxiliar a composição da rede CAN permitiu a análise de desempenho da rede CAN do RAM em desenvolvimento. Foram realizadas simulações para diferentes equipamentos e sob diferentes parâmetros de configuração que permitiram definir as condições de operação da rede junto a arquitetura proposta. Tanto para condição normal como para condição de erro, o sistema de controle para a rede CAN do RAM apresenta um tempo de resposta satisfatório para cumprir o requisito temporal de cada mensagem.

A análise dos dados das simulações rede CAN demonstram que há capacidade para inserção de novos dispositivos. Também, verificou-se que, sob o ponto de vista do controle via rede, os períodos de amostragem e o tamanho dos dados em bits definidos permitem a utilização de conjuntos de sensores ultrassônicos, além de permitir a inserção de outros grupos de sensores e mensagens para ampliação do sistema de percepção da arquitetura robótica proposta.

7.2. Trabalhos futuros

A arquitetura proposta e a rede CAN permitem a inserção de novos comportamentos e sub-comportamentos na arquitetura. Dessa forma, novas variáveis difusas podem ser elaboradas para adequar as leituras de novos sensores para o contexto da lógica difusa, e assim, viabilizar estudos de novas estruturas para

a arquitetura robótica proposta. O módulo deliberativo da arquitetura pode ser expandindo no sentido de se criarem interfaces com sistemas externos para permitir a configuração de missões, assim como o telecomando e a teleoperação do robô em situações específicas. Além disso, da forma como foi implementada, a arquitetura ainda não tem possibilidade de realizar aprendizado, assim a pesquisa de formas e algoritmos para desenvolvimento de um sub-módulo para viabilizar o aprendizado mostra-se como uma atividade interessante. Entre as possibilidades promissoras, a utilização de algoritmos genéticos junto aos controladores difusos são focos de pesquisas atuais.

A metodologia utilizada para caracterização do sensor ultrassônico, diante dos resultados positivos apresentados, constitui uma referência que pode ser utilizada para avaliação de outros sensores para medição de distâncias em relação a linhas de plantio e para compor a arquitetura proposta. Outra possibilidade é a aplicação dos sensores em um arranjo com várias unidades. Isso pode permitir a aquisição de medidas em vários pontos da planta, as quais seriam processadas para gerar um valor mais apurado da distância em relação a linhas de plantio.

O acréscimo de comportamentos na arquitetura e de mensagens na rede CAN pode elevar os atrasos e o tempo de processamento de informações, que, consequentemente, podem prejudicar o desempenho dos sistemas de controle robóticos. Assim, a busca por metodologias de análise que envolva parâmetros de rede e da arquitetura é conveniente para definir os limites ou as condições para ampliação da rede e da arquitetura. Também, a utilização da arquitetura junto a uma rede baseada na norma ISO 11783 demanda estudos no mesmo sentido, pois a implementação dessa norma eleva a carga da rede devido às mensagens próprias de gerenciamento e manutenção dessa norma.

8. REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, A.; CORREIA, L. A Fuzzy Behavior-based Architecture for Decision Control in Autonomous Vehicles. In: IEEE: INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL, 2001, Cidade do México, México. **Resumo...** México: IEEE, 2001, p. 370-375.

ADRIAN, A.M.; NORWOOD, S.H.; MASKC, P.L. Producers' perceptions and attitudes toward precision agriculture technologies. **Computers and Electronics in Agriculture**. v.48, p. 256–271, 2005.

ARKIN, R. **Behavior-based Robotics**. Cambridge: The MIT Press, 1998.

ARKIN, R.C. Motor schema based navigation for a mobile robot: An Approach to Programming by Behavior. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1987, Raleigh, NC, EUA. **Resumos...** Raleigh: IEEE, 1987, p. 264-271.

ARKIN, R.C. Path Planning for a Vision-based Autonomous Robot. In: SPIE CONFERENCE ON MOBILE ROBOTS, 1986, Cambridge, MA, EUA. **Resumos...** Cambridge: SPIE, 1986, p. 240-249.

ARKIN, R.C. **Towards Cosmopolitan Robots: Intelligent Navigation in Extended Man-made Environments.** 1987, Tese (Doutorado). University of Massachusetts, Department of Computer and Information Science. Massachusetts, EUA, 1987.

ARKIN, R.C. Towards the unification of Navigational planning and reactive Control. In: SPRING SYMPOSIUM ON ROBOT NAVIGATION, 1989, Stanford University, California, EUA. **Resumos...** Stanford University: Working Notes of the American Association for Artificial Intelligence, 1989**.** Disponível: <http://www.cc.gatech.edu/ai/robot-lab/online-publications/stanford.pdf>

ÅSTRAND, B.; BAERVELDT, A. J. Mechatronics in Agriculture - robust recognition of plant rows. In: THE 2ND INTERNATIONAL CONFERENCE ON RECENT ADVANCES IN MECHATRONICS, 2, 1999, Istanbul, Turquia. **Resumos...** Istanbul: ICRAM'99, 1999, p.135-141.

ÅSTRAND, B.; BAERVELDT, A.J. An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control. **Autonomous Robots**. v.13, p. 21-35, 2002.

AUERNHAMMER, H. Off-Road Automation Technology in European Agriculture - State of the Art and expected Trends. In: AUTOMATION TECHNOLOGY FOR OFF-ROAD EQUIPMENT, ASAE INTERNATIONAL CONFERENCE, Kyoto, Japão, 2004. **Resumos...** Kyoto, ASAE, 2004, p.10-23.

AUERNHAMMER, H. Precision farming - the environmental challenge. **Computers and Electronics in Agriculture.** v.30, n.1-3, p.31 - 43, 2001.

BAK, T.; JAKOBSON, H. Agricultural Robotic Platform with Four Wheel Steering for Weed Detection. **Biosystems Engineering**. v.87, n.2, p.126-136, 2004.

BENNEWEIS, R.K. Status of the ISO11783 Serial control and communications data network standard. In: 2005 ASAE ANNUAL INTERNATIONAL MEETING, 2005, Florida, EUA. **Resumos...** Florida: American Society of Agricultural and Biological Engineers, 2005. Número Artigo: 051167.

BENSON, E.; STOMBAUGH, T.; NOGUCHI, N.; WILL, J.; REID, J.F. An evaluation of a geomagnetic direction sensor for vehicle guidance in precision agriculture applications. In: ASABE ANNUAL INTERNATIONAL MEETING**,** 1998, St.Joseph, Michigan, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 1998. Número Artigo: 983203.

BENSON, E.R.; REID, J.F.; ZHANG, Q. Machine vision–based guidance system for an agricultural small–grain harvester. **Transactions of the American Society of Agricultural Engineers (ASAE)**. v.46, n.4, p.1255–1264, 2003.

BLACKMORE, B.; SIMON, E.; GRIEPENTRONG, H.W. Mechatronics and Applications. In: CIGR – The International Commission of Agricultural Engineering, **Handbooks of Agricultural Engineering.** Michigan, USA: Axel Munack, 2006, p.204-215.

BLACKMORE, B.S.; FOUNTAS, S.; VOUGIOUKAS, S.; TANG, L.; SØRENSEN, C.G.; JØRGENSEN, R. A method to define agricultural robot behaviours. In: AUTOMATION TECHNOLOGY FOR OFF-ROAD EQUIPMENT, 2004, Kyoto, Japão. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 2004. Número Artigo: 701P1004.

BLACKMORE, S.; HAVE, H.; FOUNTAS, S. Specification of Behavioral Requirements for an Autonomous Tractor. In: AUTOMATION TECHNOLOGY FOR OFF-ROAD EQUIPMENT, 2002, Chicago, Illinois, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 2002. Número Artigo: 701P0502.

BONARINI, A. Anytime learning and adaptation of hierarchical fuzzy logic behaviors. **Adaptive Behavior Journal**, v.5, p. 281–315, 1997. (Special Issue on Complete Agent Learning in Complex Environments).

BORENSTEIN, J.; FENG, L. Measurement and Correction of Systematic Odometry Errors in Mobile Robot. **IEEE Transactions on Robotic and Automation**, v.12, n.6, p.869-880, 1996.

CHUNG, H.; OJEDA, L.; BORENSTEIN, J. Sensor fusion for Mobile Robot Dead-reckoning with a precision-calibrated fiber optic Gyroscope. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2001, Seoul, Corea do Sul. **Resumos...** Corea do Sul: ICRA , 2001, v. 4, p. 3588-3593.

CONNELL, J.H. **Minimalist Mobile Robotics: A Colony Architecture for an Artificial Creature.** San Diego, California, EUA: Academic Press, 1990.

DARR, M.J.; STOMBAUGH, T.S.; SHEARER, S. Controller area network based distributed control for autonomous vehicles. **Transactions of the ASAE.** v.48, n.2, p.479-490, 2005.

EARL, R., THOMAS, G.; BLACKMORE, B.S. The potential role of GIS in autonomous field operations. **Computers and Electronics in Agriculture.** v.25, p.107–120, 2000.

FAO, FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. Agriculture after September 11. Disponível: <http://www.fao.org/ag/magazine/0112sp2.htm>. Acesso: Maio, 2007.

FTI, FORÇA TAREFA ISOBUS BRASIL. Disponível: <http://www.isobus.org.br>. Acesso: Maio 2007.

GARCIA-ALEGRE, M.C.; BUSTOS, P.; GUINEA, D. Complex behavior generation on autonomous robots: A case study. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEM MAN AND CYBERNETICS: INTELLIGENT SYSTEMS FOR THE 21ST CENTURY, 1995, Vancouver, Canada. **Resumos...** Vancouver: IEEE, 1995, v.2, p.1729-1734.

GARCIA-ALEGRE, M.C.; GUINEA, D. Building an Architecture for a Farming Robot. In: INTERNATIONAL WORKSHOP ON ROBOTICS & AUTOMATED MACHINERY FOR BIO-ROBOTIC 1997, Gandía, Espanha. **Resumos...** Espanha: (F.Yuste *et al*. Eds) BIO-ROBOTICS97, 1997, p. 255-260.

GARCIA-ALEGRE, M.C.; RECIO, F. Basic Visual and Motor Agents for Increasingly Complex Behavior Generation on a Mobile Robot. **Autonomous Robots.** v.5, p.19-28, 1998.

GARCIA-ALEGRE, M.C.; RIBEIRO, A.; GARCIA-PEREZ, L.; MARTINEZ, R.,; GUINEA, D.; POZO-RUZ, A. Autonomous robot in agriculture tasks. In: THIRD EUROPEAN CONFERENCE ON PRECISION AGRICULTURE, 3, 2001, Montpellier, França. **Resumos...** Montpellier:ECPA, 2001, p.25-30. (ISBN: 2-900792-13-4)

GASÓS, J.; GARCÍA-ALEGRE, M.C.; GARCÍA, R. Fuzzy strategies for the navigation of autonomous mobile robots. In: FUZZY ENGINEERING TOWARD HUMAN FRIENDLY SYSTEMS - INTERNATIONAL FUZZY ENGINEERING SYMPOSIUM, 1991, Yokohoma, Japão. **Resumos...** Japão: IFES'91, 1991, p. 1024-1034.

GAT, E. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling realworld mobile robots. In: AAAI TENTH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1992, San Jose, California, EUA. **Resumos...** California: AAAI Press, 1992, p. 809-815.

GAT, E. **Reliable Goal-directed Reactive Control for Real-world Autonomous Mobile Robots**. 1991. Tese (Doutorado) - Virginia Polytechnic Institute and State University, Blacksburg. Virginia, EUA, 1991.

GIRALT, G.; CHATILA, R.; VAISSET, M. An Integrated Navigation and Motion Control System for Multisensory Robots. In: GIRALT, G.; CHATILA, R.; VAISSET, M. **Robotics Research**. USA: MIT Press, 1984, p.191-214.

GODOY, E.P., **Desenvolvimento de uma Ferramenta de Análise de Desempenho de Redes CAN (Controller Area Network) para Aplicação em Sistemas Agrícolas**. 2007, 93p. Dissertação (Mestrado em Engenharia Mecânica) – Escola de Engenharia de São Carlos, USP. São Carlos - 2007.

GOODRIDGE, S.G.; KAY, M.G.; LUO, R.C. Multi-layered fuzzy behavior fusion for real-time reactive control of systems with multiple sensors. **IEEE Transactions on Industrial Electronics**. v.43, n.3, p.387-394, 1996.

GOODRIDGE, S.G.; KAY, M.G.; LUO, R.C. Multi-layered fuzzy behavior fusion for reactive control of an autonomous mobile robot. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 1997, Barcelona, Espanha. **Resumos...** Barcelona: IEEE, 1997, p. 579-584.

GOODRIDGE, S.G.; LUO, R.C. Fuzzy behavior fusion for reactive control of an autonomous mobile robot: MARGE. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1994, San Diego, California, EUA. **Resumos...** California: IEEE, 1994, p. 1622-1627.

GRASSI-JUNIOR, V. **Arquitetura híbrida para robôs móveis baseada em funções de navegação com interação humana.** 2006, 118p. Tese (Doutorado em Engenharia Mecânica). Escola Politécnica da Universidade de São Paulo – Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, São Paulo, São Paulo., 2006.

GUO, L. S.; ZHANG, Q.; FENG, L. A low-cost integrated positioning system of GPS and inertial sensors for autonomous agricultural vehicles. In: ASABE ANNUAL INTERNATIONAL MEETING, 2003, St.Joseph, Michigan, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 2003. Número Artigo: 033112.

GUO, L.; Q. ZHANG; S. HAN. Agricultural machinery safety alert system using ultrasonic sensors. **Journal of Agricultural Safety and Health.** v. 8, n. 4, p. 385-396, 2001.

HAGRAS, H.; CALLAGHAN, V.; COLLEY, M. Learning and Adaptation of an Intelligent Mobile Robot Navigator Operating in Unstructured Environment Based on a Novel Online Fuzzy-Genetic System. **Journal of Fuzzy Sets and Systems**. v.141, n.1, p.107-160, 2004.

HAGRAS, H.; CALLAGHAN, V.; COLLEY, M. Prototyping design and learning in outdoor mobile robots operating in unstructured outdoor environments. **IEEE International Magazine on Robotic and Automation**. v.8, n.3, p.53–69, 2001.

HAGRAS, H.; COLLEY, M.; CALLAGHAN, V.; CARR-WEST, M. Online Learning and Adaptation of Autonomous Mobile Robots for Sustainable Agriculture. **Autonomous Robots.** v.13, p.37–52, 2002.

HAGUE, T.; MARCHANT, J.A.; TILLETT, N.D. Ground based sensing systems for autonomous agricultural vehicles. **Computers and Electronics in Agriculture.** v.25, n.1-2 , p.11-28, 2000.

HAGUE, T.; SOUTHALL, B.; TILLETT, N.D. An Autonomous Crop Treatment Robot: Part II. Real Time Implementation. **International Journal of Robotics Research**, v.21, n.1, p. 75-85, 2002.

HAYES-ROTH, B.; PFLEGER, K.; LALANDA, P.; MORIGNOT, P.; BALABANOVIC, M. A domain-specific software architecture for adaptive intelligent systems. **IEEE Transactions on Software Engineering**. v.21, n.4, p.288-301, 1995.

HOWARD, A.; SERAJI, H. Vision-based terrain characterization and traversability assessment. **Journal of Robotic Systems.** v.18, n.10, p.577–587, 2001.

IIDA, M.; BURKS, T.F. Ultrasonic sensor development for automatic steering control of orchard tractor. In: CONFERENCE IN AUTOMATION TECHNOLOGY FOR OFF-ROAD EQUIPMENTE (ATOE), AMERICAN SOCIETY OF AGRICULTURAL ENGINEERS (ASAE), 2002, Chicago, Illinois, EUA. **Resumos...** [SI: s.n.], 2002. Número do artigo: 701P0502.

IP, Y.L.; RAD, A.B.; WONG, Y.K. Autonomous exploration and mapping on unknown environment. In: THIRD INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 2004, Shanghai, China. **Resumos...** Shanghai: ICML, 2004, p. 4194-4199.

JAHNS, G. Automatic Guidance of Agricultural Field Machinery. In: JOINT INTERNATIONAL CONFERENCE ON AGRICULTURAL ENGINEERING & TECHNOLOGY EXHIBITION, 1997, Dhaka, Bangladesh. **Resumos...** [SI: s.n.], 1997, p. 15-18.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transaction of the ASME - Journal of Basic Engineering**. v.82, p. 35-45, 1960.

KASSLER, M. Agricultural Automation in the new Millennium. **Computers and Electronics in Agriculture**. v.30, n.1, p.237–240, 2001.

KEICHER, R.; SEUFERT, H. Automatic guidance for agricultural vehicles in Europe. **Computers and Electronics in Agriculture.** v.25, n.1-2, p.169-194, 2000.

KHATIP, O. Real-time Obstacle Avoidance for Manipulators and Mobile robots. **The International Journal of Robotics Research**. v.5, n.1, p. 90-98,1986.

KONOLIGE, K.; MYERS, K.L.; RUSPINI, E.H.; SAFFIOTTI; A. The Saphira architecture: A design for autonomy. **Journal of experimental & theoretical artificial intelligence: JETAI**. v.9, n.1, p.215-235, 1997.

K-TEAM CORP. Disponível: <http://www.k-team.com/>. Acesso: Abril, 2007.

KUTZBACH, H.D. Trends in Power and Machinery. **Journal of Agriculture Engineering Research.** v.76, p.237-247, 2000.

LE BARS, J.M.; BOFFETY, D.; TRASSOUDAINE, L.; ALIZON, J.; IMBERBIS, C. Location improvement by combining a DGPS system with on-field vehicle sensors. In: FIRST EUROPEAN CONFERENCE ON PRECISION AGRICULTURE, 1997, Warwick, Reino Unido. **Resumos...** Warwick: BIOS Scientific Publishers Ltd, 1997, p. 585-591.

LI, S.T.; LI, Y.C. An autonomous mobile robot control method based on hierarchical fuzzy behaviors. In: FOURTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 4, 2005, Guangzhou, China. **Resumo...** Guangzhou: ICMLC, 2005, p 1327-1332.

LIN, C. H.; WANG, L.L. Intelligent collision avoidance by fuzzy logic control. **Robotics and Autonomous Systems.** v. 20, n.1, p. 61–83, 1997.

MAALOUFA, E.; SAADA, M.; SALIAHB, H. A higher level path tracking controller for a four-wheel differentially steered mobile robot. **Robotics and Autonomous Systems.** v.54, p.23-33, 2006.

MAES, P. The Dynamics of Action Selection. In: ELEVENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 11, 1989, Detroit, MI, USA. **Resumos...** Detroit: IJCAI, 1989. v.2, p. 991-997.

MALHOTRA, R.; SARKAR, A. Development of a fuzzy logic based mobile robot for dynamic obstacle avoidance and goal acquisition in an unstructured environment In: IEEE/ASME INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENT MECHATRONICS, 2005, Monterey, California, EUA. **Resumos...** California: AIM, 2005, p.1198-1203.

MATARIC, M.J. Behavior-Based Systems: Main Properties and Implications. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION – WORKSHOP IN ARCHITECTURES FOR INTELLIGENT CONTROL SYSTEMS, 1992, Nice, França. **Resumos...** Nice: IEEE, 1992, p. 46-54.

MATTHIES, L.; GATT, E.; HARRISON, R.; WILCOX, B.; VOLPE, R.; LITWIN, T. Mars microrover navigation: Performance evaluation and enhancement. **Journal of Robotics and Automation**. v. 2, n. 4, p. 291–311, 1995.

MEYSTEL, A., Nested Hierarchical Control. In: **An Introduction to Intelligent and Autonomous Control**, Boston- USA, Kluwer Academic Publishers, p.129- 161, 1993.

MIZUSHIMA, A.; NOGUCHI, N.; ISHII, K. Development of Robot Tractor Using the Low-Cost GPS/INS System. In: 2005 ASAE ANNUAL INTERNATIONAL MEETING, 2005, Florida, EUA. **Resumos...** Florida: American Society of Agricultural and Biological Engineers, 2005. Número Artigo: 051138.

MIZUSHIMA, A.; NOGUCHI, N.; ISHII, K.; TERAO, H. Development of Navigation Sensor Unit for the Agricultural Vehicle. In: IEEE/ASME INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENT MECHATRONICS, 2003, Kobe, Japão. **Resumos...** Kobe: AIM, 2003.

MOBILEROBOTS INC. Disponível: http://www.mobilerobots.com/corporate.html, Acesso: Abril, 2007.

MORAVEC, H.; CHO, D.W. A Bayesian Method for Certainty Grids. In:, SYMPOSIUM ON MOBILE ROBOTS, 1989, Stanford, California EUA. **Resumos...** California: Spring Symposium Series, 1989, p.57-60.

MURATA, S.; HIROSE, T. Onboard Locating System Using Real-Time Image Processing for Self-Navigation Vehicle. **IEEE Transaction on Industrial Electronics.** v. 40, n.1, p.145 - 154, 1993.

NILSSON, N. Shakey the Robot, Technical Note 323, SRI International, Menlo Park, CA, 1984. Disponível: <http://www.ai.sri.com/movies/Shakey.ram>. Acesso: Abril de 2006.

NOGUCHI, N.; ISHII, K.; TERAO, H. Development of an agricultural mobile robot using a geomagnetic direction sensor and image sensors. **Journal of Agricultural Engineering Research.** v.67, p.1-15, 1997.

NOGUCHI, N.; REID, J.F.; BENSON, E.; WILL, J.; STOMBAUGH, T. Vehicle automation system based on multi-sensor integration. In: ASABE ANNUAL INTERNATIONAL MEETING**,** 1998, St.Joseph, Michigan, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 1998. Número Artigo: 983111.

OKSANEN, T.; LINJA, M.; VISALA, A. Low-cost positioning system for agricultural vehicles. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, 2005, Espoo, Finland. **Resumos...** Espoo: IEEE (CIRA), 2005, p. 297-302.

OTHMAN, H.F.; AJI, Y.R.; FAKHREDDIN, F.T.; AL-ALI, A.R. Controller Area Networks: Evolution and Applications. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES, 2, 2006. **Resumos...** ICTTA, 2006, v.2, n.6, p. 3088-3093.

PARHI, D.R. Navigation of Mobile Robots Using a Fuzzy Logic Controller. **Journal of Intelligent and Robotic Systems**. v.42, p.253–273, 2005.

PAYTON, D.; ROSENBLATT, J.; KEIRSEY, D. Plan Guided Reaction. **IEEE Transactions on Systems, Man and Cybernetics**. v.20, n.6, p.1370-1382, 1990.

PAYTON, D.W. An architecture for reflexive autonomous vehicle control. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1986, San Francisco, California, EUA. **Resumos...** Nova York: IEEE, 1986, p.1838-1845.

PEDERSEN, S.M.; FOUNTAS, S.; HAVE, H.; BLACKMORE, B.S. Agricultural robots: an economic feasibility study. **Precision Agriculture**. v. 5, p,589-595, 2005.

PIN, F. G.; WATANABE, Y. Navigation of mobile robots using a fuzzy behaviorist approach and custom-designed fuzzy inference board. **Robotica**. v.12, p.491-503, 1994.

PIRJANIAN, P.; MATARIC, M.J. Fuzzy Logic Techniques for Autonomous Vehicle Navigation. In: DRAINKOV, D.; SAFFIOTTI, A. **Studies on Fuzziness and Soft Computing.** Springer: Verlag, 2000, p.235-253.

PLATAFORMA LATTES. Disponível: <http://lattes.cnpq.br>, Acesso em: Novembro, 2002.

PNI CORP. Disponível: <http://www.precisionnav.com>. Acesso: Abril, 2007.

PRITI, K.; GAONKAR, A.D.; RATTAN, K.S. Fuzzy Navigation for an Autonomous Mobile Robot. ANNUAL MEETING OF THE NORTH AMERICAN FUZZY INFORMATION PROCESSING SOCIETY (NAFIPS), 2005, Ann Arbor, Michigan, EUA. **Resumos...** Michigan: NAFIPS, 2005, p.412-417.

REID, J.F.; ZHANG, Q; NOGUCHI, N.; DICKSON, M. Agricultural automatic guidance research in North America. **Computers and Electronics in Agriculture.** v.25, n.1-2, p.155-167, 2000.

REMONDINI, D., SAFFIOTTI, A. A modular, hierarchical, configurable controller for autonomous robots. In: 12th INTERNATIONAL CONFERENCE ON METHODS AND MODELS IN AUTOMATION AND ROBOTICS, 12, 2006, Miedzyzdroje, Poland. **Resumos...** 2006, p. 585-590.

RIBEIRO, C.H.C.; REALI COSTA, A.H.; ROMERO, R.A.F. Robôs Móveis Inteligentes: Princípios e Técnicas. In: XXI CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2001, Fortaleza, Brasil. **Anais...** Fortaleza: Jornada de Atualização em Inteligência Artificial, A.T. Martins e D.L. Borges, 2001, v. 3, p. 257-306. ISBN 85-88442-03-5.

ROSENBLATT, J. D. **A Distributed Architecture for Mobile Navigation**. 1997. Tese (Doutorado). Carnegie Mellon University Robotics Institute Technical Report. Pittsburgh, PA, USA, 1997.

ROSENBLATT, J. The Distributed Architecture for Mobile Navigation. **Journal of Experimental and Theoretical Artificial Intelligence,** v.9, n.2/3, p.339-360, 1997.

ROSENBLATT, J.; WILLAMS, S.; DURRANT-WHYTE, H. Behavior-based control for autonomous underwater exploration. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2000, San Francisco, California, EUA. **Resumo**: San Francisco: IEEE, 2000.

ROSENBLATT, J.; WILLIAMS, S.; DURRANT-WHYTE, H. A Behavior-Based Architecture for Autonomous Underwater Exploration. **International Journal of Information Sciences**. v.145, n.1-2, p.69-87, 2002.

ROSENBLATT, J.K.; PAYTON, D.W. A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control. In: IEEE/INNS INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 1989, Washington DC, EUA. **Resumos...** [SI: s.n.], 1989, v.2, p.317-324.

ROSENSCHEIN, S.; KAELBLING, L., The Synthesis of Digital Machines with Provable Epistemic Properties. In: CONFERENCE ON THEORETICAL ASPECTS OF REASONING ABOUT KNOWLEDGE, 1986, Monterey, California, EUA. **Resumos...** San Francisco: Morgan Kaufmann Publishers Inc, 1986, p. 83- 98.

ROSS, T.J. **Fuzzy logic with engineering applications.** New York: McGraw-Hill International Edition, 1995.

ROUMELIOTIS, S.I.; SUKHATME, G.S.; BEKEY, G.A. Circumventing Dynamic Modeling: Evaluation of the Error-State Kalman Filter applied to Mobile Robot Localization. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1999, Detroit, EUA. **Resumos...** Detroit: IEEE, 1999, p.1656-1663.

ROVIRA-MÁS, F.; HAN, S. Kalman Filter for Sensor Fusion of GPS and Machine Vision. In: ASABE ANNUAL INTERNATIONAL MEETING, 2006, Portland, Oregon, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 2006. Número Artigo: 063034.

ROVIRA-MÁS, F.; ZHANG, Q.; REID, J.F. Automatic Navigation with Stereo Disparity. In: ASAE ANNUAL INTERNATIONAL MEETING, 2003, Las Vegas, Nevada, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 2003. Número Artigo: 033113

RUSPINI, E.H. Fuzzy logic in the Flakey robot. In: INTERNATIONAL CONFERENCE ON FUZZY LOGIC AND NEURAL NETWORKS, 1990, Iizuka, Japão. **Resumos...** [SI: s.n.], 1990, p.767-770.

SAFFIOTTI, A. The uses of fuzzy logic for autonomous robot navigation. **Soft Computing**. v. 1, p. 180–197, 1997.

SAFFIOTTI, A.; KONOLIGE, K.; RUSPINI; E. H. A multivalued logic approach to integrating planning and control. **Artificial Intelligence**. v.76, n.1-2, p.481-526, 1995.

SAFFIOTTI, A.; RUSPINI, E. H.; KONOLIGE, K. Blending Reactivity and Goal-Directedness in a Fuzzy Controller. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 1993, Orlando, Florida, EUA. **Resumos**: Florida: IEEE Computer Press, 1993, p. 134-139.

SAFFIOTTI, A.; RUSPINI, E.H.; KONOLIGE, K.. Using Fuzzy Logic for Mobile Robot Control. In: ZIMMERMANN, H.J. **Practical Applications of Fuzzy Technologies, Handbooks of Fuzzy Sets series.** USA: Kluwer Academic, 1999, v.6, p. 185-205.

SANCHIZ, J.M.; MARCHANT, J.A.; PLA, F.; HAGUE, T. Real-time visual sensing for task planning in a field navigation vehicles. **Real-Time Imaging**. v.4, n.1, p.55-65, 1998.

SANDI, F.A; HEMERLY, E.M.; LAGES, W.F. Estimação em Tempo Real de Posição e Orientação de Robôs Móveis Utilizando Sensores com Diferentes Taxas de Amostragem. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 3, 1997, Vitória, Espírito Santo, Brasil. **Anais...** [SI: s.n.], 1997, p.453-458.

SERAJI, H. Fuzzy traversability index: A new concept for terrain-based navigation. **Journal Robotic Systems.** v.17, n.2, p.75–91, 2000.

SERAJI, H.; HOWARD, A. Behavior-Based Navigation on Challenging Terrain: A Fuzzy Logic Approach. **IEEE Transactions on Robotics and Automation**. v.18, n.3, p.308-321, 2002.

SIEGWART, R.; NOURBAKHSH, I.R. **Introduction to Autonomous Mobile Robots. A Bradford Book**. Massachusetts London, England: The Massachusetts Institute of Technology Press Cambridge, 2004.

SOUSA, R.V. **CAN (Controller Area Network): uma abordagem para automação e controle na área agrícola.** 2002, 84p. Dissertação (Mestrado em Engenharia Mecânica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, S.P. - 2002.

SOUTHALL, B.; HAGUE, T.; MARCHANT, J. A.; BUXTON, B. F. An Autonomous Crop Treatment Robot: Part I. A Kalman Filter Model for Localization and Crop/Weed Classification. **International Journal of Robotics Research**. v.21, n.1, p.61-74, 2002.

STOMBAUGH, T.; BENSON, E.; HUMMEL, J.W. Automatic guidance of agricultural vehicles at high field speeds. In: ASABE ANNUAL INTERNATIONAL MEETING**,** 1998, St.Joseph, Michigan, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 1998. Número Artigo: 983110.

STOYTCHEV, A.; ARKIN, R.C. Combining Deliberation, Reactivity, and Motivation in the Context of a Behavior-based Robot Architecture. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, 2001, Banff, Alberta, Canada. **Resumos...** Alberta: IEEE, 2001.

SUGENO, M.; NISHIDA, M. Fuzzy control of a model car. **Fuzzy Sets and Systems**. v. 16, p.103–113, 1985.

TAKAHASHI, T.; ZHANG, S.; FUKUCHI, H. Acquisition of 3-D Information by Binocular Stereo Vision for Vehicle Navigation through an Orchard. In: AUTOMATION TECHNOLOGY FOR OFF-ROAD EQUIPMENT, 2002, Chicago, Illinois, EUA. **Resumos...** Michigan: American Society of Agricultural and Biological Engineers, 2002. p.337-346. Número Artigo: 701P0502.

TAKEUCHI, T.; NAGAI, Y.; ENOMOTO. N. Fuzzy control of a mobile robot for obstacle avoidance. **Information Sciences.** v. 45, p.231–248, 1988.

TECA,TECHNOLOGY FOR AGRICULTURE. Disponível: <http://www.fao.org/sd/teca/>. Acesso: Maio 2007.

TORII, T. Research in autonomous agriculture vehicles in Japan. **Computers and Electronics in Agriculture.** v.25, n.1-2, p.133-153, 2000.

TUNSTEL, E. T.; JAMSHIDI, L.M. Behavior hierarchy for autonomous mobile robots: fuzzy behavior modulation and evolution. **International Journal of Intelligent Automation and Soft Computing**. v.3, n.1, p.37-49, 1997.

TUNSTEL, E., DE OLIVEIRA, M.A.A., BERMAN, S. Fuzzy Behavior Hierarchies for Multi-Robot Control. **International Journal of Intelligent Systems.** v.17, n.5, p. 449-470, 2002. Special Issue: "Hierarchical Fuzzy Systems".

VELOSO, M.M.; CARBONELL, J.G. Derivational Analogy in Prodigy: Automating Case Acquisition, Storage, and Utilization. **Machine Learning.** v.10, p.249-278, 1993.

VUOLO, J. H. **Introdução à Teoria de Erros**. São Paulo: IF/USP, 1991, v.2, p. 16-23.

WATANABE, K.; IZUMI, K.; MAKI, J.; FUJIMOTO, K. A Fuzzy Behavior-Based Control for Mobile Robots Using Adaptive Fusion Units. **Journal of Intelligent and Robotic Systems.** v.42, p27–49, 2005.

WELCH, G.; BISHOP, G. **An Introduction to the Kalman Filter**. Department of Computer Science, University of North Carolina at Chappell Hill, 2002.

WILLIAMS, S. B.; NEWMAN, P.; ROSENBLATT, J.; DISSANAYAKE, G.; DURRANT-WHYTE, H. Autonomous Underwater Navigation and Control. **Robotica**. v.19, n.5, p.481-496, 2001.

YAGER, R.R.; FILEV, D. **Essentials of Fuzzy Modeling and Control**. New York: John Wiley & Sons, Inc.1994.

YANG, T.C. Networked control system: a brief survey. **IEE Proceedings: Control Theory and Applications.** v.153, n.4, p.403-412, 2006.

YANG, X.; MOALLEM, M.; PATEL, R.V. Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation. **IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics.** v.35, n.6, p.1203-1213, 2005.

YAVUZ, H.; BRADSHAW, A. A New Conceptual Approach to the Design of Hybrid Control Architecture for Autonomous Mobile Robots. **Journal of Intelligent and Robotic Systems.** v.34, p.1–26, 2002.

YEN, J.; PFLUGER, N. A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation. **IEEE Transactions System, Man Cybernetics.** v.25, n.6, p.971-978, 1995.

YEN, J.; PFLUGER, N. Path planning and execution using fuzzy logic. In: AIAA GUIDANCE, NAVIGATION AND CONTROL CONFERENCE, 1991, New Orleans, Los Angeles, EUA. **Resumo...** New Orleans: AIAA, 1991, p. 1691–1698.

ZADEH, L. A. Fuzzy Sets. **Information and Control.** v. 8, p. 338-353, 1965.

ZADEH, L. A. Outline of a new approach to the analysis of complex systems and decision processes. **IEEE Transaction on Systems, Man and Cybernetics**. v.3, n.1, p. 28-44, 1973.

ZIMMERMANN, H. J. **Fuzzy Set Theory and Its Applications**. Boston: Kluwer, 2 ed revisada, 1991.