

UNIVERSIDADE DO VALE DO RIO DOS SINOS

Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada

Gustavo Pessin

Evolução de Estratégias e Controle Inteligente em
Sistemas Multi-Robóticos Robustos



Universidade do Vale do Rio dos Sinos – UNISINOS
Ciências Exatas e Tecnológicas
Programa Interdisciplinar de Pós-Graduação em Computação Aplicada - PIPCA

GUSTAVO PESSIN

EVOLUÇÃO DE ESTRATÉGIAS E CONTROLE INTELIGENTE
EM SISTEMAS MULTI-ROBÓTICOS ROBUSTOS

Dissertação de Mestrado apresentada para a obtenção
do título de Mestre em Computação Aplicada na
Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador: Prof. Dr. Fernando Santos Osório

São Leopoldo, 2008

Pessin, Gustavo

Evolução de Estratégias e Controle Inteligente em Sistemas Multi-Robóticos Robustos / Gustavo Pessin. -- São Leopoldo: Unisinos / Ciências Exatas e Tecnológicas, 2008.

133 f. : il.

Orientador: Fernando Santos Osório.

Dissertação (mestrado) – Unisinos / Ciências Exatas e Tecnológicas, 2008.

Referências bibliográficas: f. 122-133

1. Inteligência Artificial 2. Sistema Multi-Agente. 3. Rede Neural Artificial. 4. Algoritmos Genéticos. 5. Simulação. 6. Ambiente Virtual 3D. 7. Robôs Móveis Autônomos. I. Osório, Fernando Santos. II. Universidade do Vale do Rio dos Sinos, Ciências Exatas e Tecnológicas. III. Título.



UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

Aluno: **Gustavo Pessin**

Título da Dissertação: **Evolução de Estratégias e Controle Inteligente em Sistemas Multi-Robóticos Robustos.**

Banca:

Dr. Fernando Santos Osório



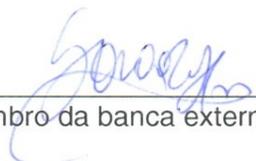
Presidente/Orientador

Dr. Jorge Luis Victória Barbosa



Membro da banca interno

Dra. Soraia Raupp Musse



Membro da banca externo

A banca examinadora da Dissertação, sob registro de Ata nº 05/2008 - PIPCA, em cumprimento ao Regimento do Programa Interdisciplinar de Pós-graduação em Computação Aplicada, julga esta Dissertação aprovada para o processo de obtenção de título de Mestre a Gustavo Pessin.

São Leopoldo, 22 de fevereiro de 2008.

Não devemos acreditar nos muitos que dizem que só as pessoas livres devem ser educadas, devemos antes acreditar nos filósofos que dizem que apenas as pessoas educadas são livres.

Epictetus, filósofo Grego e ex-escravo.

AGRADECIMENTOS

Eu havia lido, há cerca de 5 anos, uma frase do livro *O mundo assombrado pelos demônios* de Carl Sagan, esta frase dizia: “*Em muitos exemplos, o que está errado é parcialmente certo, ou estimula aos outros a descobrir o que está certo.*” Na época em que li esta frase eu não havia entendido seu significado com muita profundidade.

O mestrado, para mim, foi um período de dedicação máxima. Eu nunca havia realizado um trabalho assustadoramente tão grande, nem abdicado de tantas outras coisas a fim de realizar um trabalho. Quando os primeiros erros começaram a surgir, logo no começo, me passaram a impressão que eu não seria capaz de ir até o fim.

Fernando Santos Osório, meu orientador, me fez recordar aquela frase, mostrou-me que *os outros*, na verdade, era eu mesmo; me fez entender profundamente o significado de *parcialmente certo*. Meu mais sincero agradecimento vai a ele que me apresentou um novo mundo onde foi possível conviver e evoluir através de experiências aparentemente erradas e, acima de tudo, mostrou-me como nascem as idéias que transformam todos os nossos experimentos em trabalho válido.

Não posso deixar de lembrar, neste agradecimento, de todos os professores do PIPCA, assim como os colegas do mestrado. Muito obrigado por auxiliarem a preencher muitas das lacunas de conhecimento que eu possuía. Vocês foram, inclusive, muito bons exemplos de modelos de convivência e dedicação.

Agradeço também minha família e meus amigos: vocês foram fundamentais no desenvolvimento deste trabalho; especialmente minha namorada Renata Parisotto e meu “irmão de mãe trocada” Júlio S. Valandro. Desculpem-me pela ausência.

Sou, de certa maneira, cético, paradoxalmente, meu último agradecimento vai a Deus; sei que Ele esteve do meu lado. Para finalizar, é bom lembrar: “*Uma vez escoteiro, sempre escoteiro!*”

RESUMO

Este trabalho está relacionado com a aplicação de técnicas de Inteligência Artificial no desenvolvimento de um Sistema Multi-Agente robótico aplicado ao problema da monitoração e combate a incêndios em áreas florestais. O objetivo macro é evoluir estratégias de formação de equipes de combate a incêndio (unidade de controle) e criar métodos robustos de navegação em agentes robóticos (unidades de combate), considerando um ambiente virtual de simulação realística. No sistema proposto, uma equipe de agentes autônomos trabalha cooperativamente a fim de realizar com sucesso a identificação e o combate a incêndios em áreas florestais, sem intervenção humana. O ambiente virtual 3D suporta uma série de características fundamentais para a simulação realística da operação, como terrenos irregulares, processos naturais e restrições físicas na criação e uso de robôs móveis. Este ambiente foi implementado através do uso das bibliotecas OSG, ODE e *Demeter*. A operação multi-agente depende essencialmente de duas etapas: planejamento e ação. No planejamento, usamos Algoritmos Genéticos (AG) para evoluir estratégias de posicionamento de atuação dos robôs bombeiros. Para a ação foram criados robôs de combate a incêndios fisicamente simulados, sendo as informações sensoriais de cada robô (*e.g.* GPS, bússola, sonar) usadas na entrada de uma Rede Neural Artificial (RNA). Esta RNA que controla os atuadores do veículo permite a navegação robusta com desvio de obstáculos. Os resultados de diversos experimentos demonstram que a Rede Neural é capaz de controlar de modo plenamente satisfatório os robôs móveis e, que o uso de Algoritmos Genéticos permite definir uma estratégia de combate a incêndios, obtendo posições muito satisfatórias de ataque ao foco do incêndio. O sistema multi-agente proposto pode vir a ter um papel muito importante no planejamento e execução de operações reais de combate a incêndios florestais, como foi demonstrado pelo sucesso das simulações executadas pela equipe de robôs bombeiros.

Palavras-chave: Sistema Multi-Agente, Rede Neural Artificial, Algoritmos Genéticos, Simulação, Ambiente Virtual 3D, Robôs Móveis Autônomos.

ABSTRACT

This work is related to the application of Artificial Intelligence techniques to develop a Multi-Agent Robotic System applied to the problem of monitoring wild forest fires and to the execution of fire fighting actions. Our main goal was to evolve strategies (control unit) in order to define the positioning of the fire-fighting autonomous robotic team and to create robust navigation methods used to control robotic agents (combat units). This work was developed based on simulations accomplished using a realistic 3D virtual environment, specially implemented for this purpose, using the software libraries OSG, Demeter and ODE. In the proposed system, a team of autonomous agents work cooperatively in order to successfully perform the identification and fighting of forest fires, without any human intervention. The 3D virtual environment includes several features for realistic simulation of this task, as for example, adoption of irregular terrains, natural processes simulation (*e.g.* fire propagation), and simulation of physical restrictions (*e.g.* gravity, collision detection, object dynamics) when operating with mobile robots. The multi-agent fire-fighting global task execution depends of two main steps: planning and action. The planning task is achieved using Genetic Algorithms (GA), used to optimize the fire-fighting robots strategies, which consist of obtaining key action positions where they should start to work, clearing around the wood to prevent the spreading of fire. For the wood clearing action task, we simulated physically based fire-fighting robots, which make use of sensory information (*e.g.* GPS, compass, sonar) in order to navigate in the forest, from their base station up to the fire-fighting working area. The sensors information were used as the inputs of an Artificial Neural Network (ANN), and the ANN outputs generate control signals sent to the robot actuators. The ANN controls the vehicle actuators and allows a robust navigation in direction to a specified position, avoiding colliding with the obstacles in the path, like trees for example. The results obtained from several experiments show that the Neural Network controls very satisfactorily the mobile robots, allowing navigating from one point to another one without colliding with the obstacles. The use of GA allowed us to obtain well optimized fire-fighting positions and, using the GA optimized strategy, the proposed multi-agent system was able to extinguish the fire in almost all the simulated situations. The experiments demonstrate that this system could have a very important role in the near future applied to planning and execution of operations for real fire fighting in wild forests.

Keywords: Multi-Agent System, Artificial Neural Network, Genetic Algorithms, Simulation, 3D Virtual Environment, Autonomous Mobile Robots.

LISTA DE FIGURAS

1.1	Formações para uma equipe de quatro agentes (da esquerda para a direita: linha, coluna, circular ou ferradura)(BALCH; ARKIN, 1998).	19
2.1	Classificação segundo a anatomia (PIERI, 2002).	26
2.2	Aeronave automatizada brasileira, fruto do projeto <i>SiDeVAAN</i> (DEMEC/UFMG, 2007), para ser usada em pulverização de lavouras, inspeção de linhas de transmissão ou em missões de controle do espaço aéreo.	27
2.3	Robôs aéreos de combate: (a) Protótipo <i>X-45</i> americano (BOEING, 2007); (b) Protótipo <i>nEUROn</i> (DASSAULT, 2007), da comunidade europeia.	27
2.4	Robôs aquáticos de exploração: (a) <i>Depthx</i> (WETTERGREEN, 2007); (b) <i>Isis</i> (NOC, 2007).	27
2.5	Robôs com pernas: (a) <i>Asimo</i> (HONDA, 2007); (b) <i>Spiderbot</i> (JPL/NASA, 2007), robô para trabalho fino em ambientes confinados.	28
2.6	<i>R-Gator</i> (IROBOT, 2007), um veículo que permite tele-operação, navegação automática entre pontos tipo “ensinar-repetir”, desvio de obstáculos e operação manual.	29
2.7	<i>Urbie</i> (JPL/NASA, 2007), robô para fins militares de reconhecimento de terrenos urbanos.	29
2.8	Robôs de serviço: (a) Aspirador de pó <i>Roomba</i> (IROBOT, 2007); (b) Colheitadora autônoma <i>Demeter</i> (RI/CMU, 2007).	30
2.9	Robôs de campo: (a) <i>Mars Exploration Rover</i> (JPL/NASA, 2007) para exploração espacial; (b) <i>Dante II</i> (JPL/NASA, 2007) para exploração de cavernas e vulcões.	31
2.10	Robôs pessoais operados por controle remoto: (a) <i>RoboSapien</i> (WOWWEE, 2007), pequeno robô humanóide; (b) Cão <i>AIBO</i> (SONY, 2007).	31
2.11	Visão esquemática de um <i>Mars Exploration Rover</i> (JPL/NASA, 2007).	33
2.12	Sensor Laser <i>LDN 41 A</i> (ASTECH, 2007).	35

2.13	Controle reativo (OSÓRIO, 1998): (a) Desvio de paredes; (b) Seguindo paredes.	38
2.14	Comportamento deliberativo com o planejamento e execução de trajetória considerando o mapa do ambiente (HEINEN, 2000).	38
2.15	<i>Microsoft Robotics Studio</i> : (a) Exemplo de uma simulação; (b) Exemplo da <i>Visual Programming Language</i> .	40
2.16	Exemplo de telas do <i>Webots</i> : (a) Sumo robótico; (b) Robô holonômico.	41
2.17	Exemplo de telas: (a) <i>JUICE</i> ; (b) <i>Simulator BOB</i> .	41
2.18	Articulações disponíveis na ODE (SMITH, 2006).	43
2.19	Esquema típico de um agente (REIS, 2003).	45
2.20	Exemplo de um sistema federado (FONSECA, 2000).	50
2.21	Exemplo de um sistema baseado em quadro-negro (FONSECA, 2000).	50
2.22	Neurônio de McCulloch e Pitts (MCP), retirado de (REZENDE, 2003).	53
2.23	Exemplo de uma função <i>sigmoid</i> .	53
2.24	Descida do gradiente de uma superfície de erro.	54
2.25	Curvas de erro em aprendizado e validação.	54
2.26	Exemplo de tela do JavaNNS treinando uma RNA.	56
2.27	<i>Crossover</i> e mutação, adaptado de (LACERDA; CARVALHO, 1999).	58
2.28	Métodos de seleção: (a) Roleta; (b) Amostragem universal estocástica. Nestes exemplos, I1 representa 50%, I2 representa 25%, I3 representa 12,5%, e I4 e I5 representam 7,25% cada.	59
2.29	Desempenho do GA com elitismo (LACERDA; CARVALHO, 1999).	59
2.30	Avião-tanque <i>Canadair CL-415</i> (BOMBARDIER, 2007), transporta 6t de água, abastece sem pousar em uma superfície de água com 1,2km de comprimento e 2m de profundidade.	65
2.31	Foco de incêndio próximo ao Parque Nacional dos Aparados da Serra, imagem da Divisão de Processamento de Imagens do INPE, satélite <i>LandSat 2000</i> (DPI/INPE, 2007).	68
2.32	Tabela de informações disponibilizada junto a foto do foco de incêndio (DPI/INPE, 2007).	68

3.1	Raposa, projeto português de robô para auxílio em operações de resgate (ID-MIND, 2007).	73
3.2	Robô SACI (Sistema de Apoio ao Combate de Incidentes) (MACEDO; MACEDO; DUARTE, 2007).	73
3.3	Distribuição dos sensores do tipo sonar no veículo usado do SEVA3D (OSÓRIO et al., 2006b).	74
3.4	Distribuição dos sensores no veículo (ZHAO; COLLINS, 2005).	75
3.5	Resultados dos protótipos (PESSIN, 2007): (a) Modelo 2D com dezoito agentes cercando um alvo; (b) Modelo 3D com seis agentes.	78
4.1	Diagrama da operação multi-agente.	81
4.2	Fases do sistema proposto.	82
4.3	Motoniveladora.	84
4.4	(a) Fragmento de carta topográfica de UTM (0578000, 6764000) até (0584000, 6770000) representando um quadrante de 6km x 6km; (b) Mapa de modelos de combustíveis (MA, 2006).	84
4.5	Exemplo de quatro cenas dos veículos desenvolvidos.	86
4.6	Bússola e GPS: responsáveis pela orientação, localização e obtenção do azimute.	86
4.7	Trajetos de ação dos agentes de combate: (a) posicionamento teórico sem considerar erro no GPS; (b) posicionamento com redundância, considerando erro no GPS.	87
4.8	Posições para coleta de dados.	92
4.9	Cenas de simulações realizadas para obtenção de dados usando o sistema de regras. Todas as Figuras apresentam terreno irregular, com exceção de (a) e (b) que apresentam chão plano.	94
5.1	Evolução do <i>fitness</i> de acordo com o número de gerações para 4 agentes de combate e diferentes quantidades de indivíduos: (a) Menor <i>fitness</i> de toda população, em valor absoluto; (b) Média do <i>fitness</i> da população, em valor normalizado.	98

5.2	Resultado de duas evoluções satisfatórias, para 4 agentes de combate, utilizando os parâmetros citados acima. A zona escura é a área consumida pelo fogo, que foi contido pelo aceiro criado pelos 4 robôs.	99
5.3	Resultado de duas evoluções não satisfatórias, para 4 agentes de combate, utilizando 20 indivíduos e 700 gerações. O fogo não é contido pelo aceiro em (b).	100
5.4	Resultado de duas evoluções não satisfatórias, para 4 agentes de combate, utilizando 50 indivíduos e 700 gerações. O aceiro criado não está adequadamente bem distribuído entre os robôs.	100
5.5	Resultado de duas evoluções não satisfatórias, para 4 agentes de combate, utilizando 100 indivíduos e 700 gerações. Em (b) a área do aceiro é excessivamente grande em relação ao que seria necessário.	100
5.6	Quatro cenas de resultados satisfatórios de evoluções de estratégias usando o GA com 150 indivíduos e 700 gerações.	101
5.7	Evolução do <i>fitness</i> de acordo com o número de gerações para 2 agentes de combate e diferentes quantidades de indivíduos: (a) Menor <i>fitness</i> de toda a população, em valor absoluto; (b) Média do <i>fitness</i> da população, em valor normalizado.	102
5.8	Resultado de duas evoluções satisfatórias, para 2 agentes de combate, utilizando os parâmetros citados acima.	103
5.9	Quatro cenas de resultados satisfatórios de evoluções de estratégias usando o GA com 150 indivíduos e 700 gerações.	104
5.10	Evolução do <i>fitness</i> máximo, médio e mínimo, em valor absoluto, de acordo com o número de gerações para 4 agentes de combate; GA com 150 indivíduos e mutação Uniforme.	105
5.11	E_{MSE} da RNA de 24 neurônios: (a) Visão completa (de 0 até 100.000 ciclos); (b) Visão ampliada da inversão das curvas - Ponto Ótimo de Generalização (de 20.000 até 30.000 ciclos).	109
5.12	Trajetos realizados com simulações usando a RNA: Todas as Figuras apresentam taxa de ocupação por árvores de 10%, com exceção da (f) que apresenta um ambiente com 2,5% de ocupação.	112

5.13	Duas cenas de navegação considerando erro nos sensores. Em cada uma das imagens é apresentado um veículo com ruído simulado e outro com sensores sem ruído.	114
5.14	Seqüências de uma simulação com navegação e desvio satisfatórios, para observação das posições de parada dos RMAs. (a) até (d) especificam a ordem da seqüência.	115

LISTA DE TABELAS

4.1	Genoma desenvolvido para um grupo de 4 agentes.	89
4.2	Genoma desenvolvido para um grupo de 2 agentes.	89
5.1	Melhores indivíduos resultantes de 3 simulações para 4 agentes; utilizando no GA 150 indivíduos e 700 gerações.	99
5.2	Melhores indivíduos resultantes de 3 simulações para 2 agentes; utilizando no GA 150 indivíduos e 700 gerações.	103
5.3	Melhores indivíduos resultantes de 5 simulações para 4 agentes; utilizando no GA 150 indivíduos, 400 gerações e mutação do tipo Uniforme.	105
5.4	E_{MAE} medido na base de teste usando as RNAs com 4 e 9 neurônios na camada oculta.	108
5.5	E_{MAE} medido na base de teste usando as RNAs com 18, 24, 30 e 36 neurônios na camada oculta.	108
5.6	E_{MAE} da RNA com 24 neurônios na camada oculta: Erro no giro aumenta a partir de 32.500 ciclos	110
5.7	Resultado das simulações usando a RNA com diferentes ocupações no ambiente.	111
5.8	Resultado das simulações aplicando ruído na leitura dos 5 sonares.	113
5.9	Resultado das simulações aplicando ruído nos 2 atuadores.	113
5.10	Resultado das simulações aplicando ruído de até 100% em um sensor aleatório.	114

LISTA DE ABREVIATURAS

- AG - Algoritmos Genéticos (*Genetic Algorithms*)
- AGV - *Automated Guided Vehicles* (Veículos Autoguiados)
- AI - *Artificial Intelligence* (Inteligência Artificial)
- ANN - *Artificial Neural Network* (Redes Neurais Artificiais)
- CEPED - Centro de Estudos para a Prevenção de Emergências e Desastres
- FINEP - Financiadora de Estudos e Projetos
- GA - *Genetic Algorithms* (Algoritmos Genéticos)
- GDAL - *Geospatial Data Abstraction Library*
- GPS - *Global Positioning System* (Sistema de Posicionamento Global)
- GPVA - Grupo de Pesquisas em Veículos Autônomos
- IA - Inteligência Artificial (*Artificial Intelligence*)
- IAVI - Inteligência Artificial Aplicada a Veículos e Agentes Móveis Inteligentes
- MAS - *Multi-Agent System* (Sistema Multi-Agente)
- MCP - Neurônio de McCulloch e Pitts
- ML - *Machine Learning* (Aprendizado de Máquina)
- ODE - *Open Dynamics Engine*
- OSG - *Open Scene Graph*
- PIPICA - Programa de Pós-Graduação em Computação Aplicada
- RBV - Rede Brasileira de Visualização (FINEP)
- RMA - Robô Móvel Autônomo
- RNA - Rede Neural Artificial (*Artificial Neural Network*)
- SDL - *Simple DirectMedia Layer*
- SMA - Sistemas Multi-Agente (*Multi-Agent System*)

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Contextualização	17
1.2	Definição do problema	18
1.3	Objetivos e metodologia	21
1.4	Organização do texto	23
2	BASE CONCEITUAL	25
2.1	Robótica Móvel	25
2.1.1	Introdução	25
2.1.2	Classificações	26
2.1.2.1	Anatomia	26
2.1.2.2	Tipo de controle	28
2.1.2.3	Funcionalidade	29
2.1.3	Desafios da robótica móvel	30
2.1.4	Sensores e atuadores	32
2.1.5	Controle	35
2.1.5.1	Estratégias	35
2.1.5.2	Arquitetura	36
2.1.6	Simulação e modelagem	39
2.1.6.1	Ferramentas Pesquisadas	40
2.1.6.2	Bibliotecas	41
2.2	Agentes Inteligentes	45

2.2.1	Introdução	45
2.2.2	Agentes	45
2.2.3	Sistemas Multi-Agentes	48
2.2.4	Cooperação	49
2.2.5	Comunicação	49
2.2.6	Considerações finais	50
2.3	Aprendizado de Máquina	51
2.3.1	Introdução	51
2.3.2	Redes Neurais Artificiais	52
2.3.2.1	SNNS (SNNS, 2006)	55
2.3.3	Algoritmos Genéticos	55
2.3.3.1	O Algoritmo Genético Típico	56
2.3.3.2	Operadores Genéticos	57
2.3.3.3	Seleção	58
2.3.3.4	Elitismo	59
2.3.3.5	Parâmetros Genéticos	59
2.3.3.6	GAlib (WALL, 2007)	60
2.4	Incêndios em Ambientes Naturais	61
2.4.1	Introdução	61
2.4.2	Combustíveis Florestais	61
2.4.3	Técnicas Reais de Operação	63
2.4.3.1	Operações de combate	63
2.4.3.2	Equipamentos de combate	64
2.4.3.3	Métodos e estratégias de combate	66
2.4.3.4	Sistemas de detecção	66
2.4.4	Considerações sobre Incêndios em Ambientes Naturais	67

2.5	Considerações Finais	69
3	TRABALHOS RELACIONADOS	72
3.1	Operações com Robôs Reais	72
3.2	Formas de Controle e de Evolução de Estratégias	74
3.3	Formação de Grupos e Formas de Comunicação	76
3.4	Considerações Finais	78
4	MODELO PROPOSTO	80
4.1	Descrição Geral	80
4.2	Morfologia dos Robôs Móveis	85
4.3	Evolução de Estratégias	87
4.3.1	Escolha do <i>Fitness</i>	90
4.4	Controle de Navegação	91
4.5	Considerações Finais	95
5	EXPERIMENTOS E RESULTADOS	96
5.1	Evolução de Estratégias	96
5.2	Controle de Navegação	106
5.2.1	Teste e Seleção da RNA	107
5.2.2	Aplicação da RNA no Controle	110
6	CONSIDERAÇÕES FINAIS	116
6.1	Revisão do Tema	116
6.2	Resultados e Conclusões	117
6.3	Trabalhos Futuros	120
	Referências	122

1 INTRODUÇÃO

1.1 Contextualização

Com a evolução das pesquisas em robótica, cada vez mais os robôs estão se tornando complexos em termos físicos. A grande variedade de estudos em morfologia robótica tem desenvolvido variações de robôs dotados de diversos meios de locomoção, como pernas (*e.g.* bípedes, quadrúpedes, hexápodes), rodas (*e.g.* automóveis, motocicletas, triciclos) ou esteiras. Em paralelo a este desenvolvimento temos a evolução constante de uma gama extremamente grande de sensores (*e.g.* sistemas de visualização, posicionamento, detecção de obstáculos). O desenvolvimento de algoritmos e técnicas de sistemas baseados em regras para coordenar estes conjuntos físicos em um ambiente dinâmico é um desafio extremamente complexo (GO; BROWNING; VELOSO, 2004). Dotar robôs autônomos de capacidade de raciocínio inteligente e de interação com o meio em que estão inseridos é uma área de pesquisa que tem atraído a atenção de um grande número de pesquisadores (DUDEK; JENKIN, 2000).

Um robô móvel autônomo (RMA) pode ser definido como um agente inteligente (GARCIA; SICHMAN, 2003): móvel por ter capacidade de locomoção em um ambiente; autônomo para ter o mínimo necessário de interação com seres humanos; e, inteligente para, a partir de dados de sensores, poder atuar e se adaptar de maneira coerente no ambiente onde está inserido. A robótica móvel é uma área de pesquisa que lida com o controle de veículos autônomos ou semi-autônomos (HEINEN, 2000; BEKEY, 2005).

Existem diversas áreas onde a habilidade de um único agente autônomo não é suficiente ou eficiente para a realização de uma tarefa, assim, a utilização de um único agente definitivamente não é a melhor solução. Em alguns destes casos, como patrulhamento, vigilância, resgate ou exploração o mais indicado é a aplicação de sistemas multi-robóticos. Sistemas multi-robóticos são sistemas onde robôs autônomos trabalham cooperativamente a fim de cumprir uma missão, podendo existir interação entre os robôs ou não (OSAGIE, 2006). É um tipo de sistema extremamente dependente de técnicas de controle. No que diz respeito à aplicabilidade

de robôs autônomos, o trabalho com sistemas multi-robóticos pode adicionar mobilidade, flexibilidade e robustez de uma nova maneira e a uma nova gama de aplicações (MONDADA et al., 2005).

Um grande sonho de nossa sociedade é a aplicação de sistemas robóticos substituindo as pessoas em atividades que coloquem em risco a vida humana, onde a atuação humana é deficitária ou onde é, de certa forma, ineficiente. A tarefa de monitoração e combate a incêndios em áreas florestais é um caso onde a aplicação de um sistema multi-robótico poderia diminuir consideravelmente os prejuízos humanos, materiais e ambientais.

Com relação a incêndios florestais, anualmente registram-se cerca de 45.000 incêndios nas florestas da Europa. Entre 1989 e 1993, só na zona mediterrânea, 2,6 milhões de hectares florestais foram destruídos pelo fogo, o equivalente ao desaparecimento de um território com a dimensão da Bélgica em cinco anos (CE, 2006). A extensão territorial do Brasil e a diversidade de sua cobertura vegetal, bem como o número expressivo de ocorrências de incêndios florestais verificado no país, são fatores que enfatizam a necessidade de um sistema cada vez mais aprimorado e consistente de detecção e combate a incêndios florestais (BATISTA, 2004). Diversas iniciativas a fim de incrementar a capacidade de reação dos órgãos públicos e civis, no sentido de evitar desastres, tem sido uma das preocupações junto a órgãos como a Secretaria Nacional de Defesa Civil, levando a criação de novos Centros de Estudos para a Prevenção de Emergências e Desastres (CEPED). Importantes iniciativas, como a Rede Brasileira de Visualização (RBV), financiada pela FINEP, também têm sido incentivadas, onde a competência de Segurança e Defesa (Civil e Militar) da Rede vem sendo foco de desenvolvimentos junto a nossa Universidade, na qual este projeto se integra.

Os incêndios florestais causam diversos tipos de danos humanos, materiais e ambientais. Danos ambientais na fauna e flora afetam desde o solo até o aspecto de sobrevivência de áreas não incendiadas, prejudicando os ambientes naturais e o planejamento florestal. As perdas humanas são mais dramáticas, onde este custo dificilmente pode ser quantificado. Como exemplos de prejuízos humanos, na Austrália, um incêndio em 1983 atingiu cerca de 400.000ha e matou 75 pessoas; no Canadá e EUA, incêndios entre 1969 e 1994 mataram cerca de 52 pessoas; no Brasil, um incêndio no Paraná, em 1973 provocou 110 mortes (LIF, 2006).

1.2 Definição do problema

O desenvolvimento de um sistema multi-robótico para atuação em monitoração e combate a incêndios em áreas florestais é um problema multidisciplinar. Assim, o sucesso desta

tarefa envolve o uso de uma variedade de tecnologias de diferentes campos, como robótica móvel (*e.g.* controle de movimentação, fusão de sensores, modelagem robótica). Envolve a coleta de informações de especialistas sobre incêndios (*e.g.* comportamento do fogo em ambientes naturais, técnicas e métodos de monitoração, estratégias reais de combate). Além de, neste trabalho, envolver estudos sobre Sistemas Multi-Agentes e Aprendizado de Máquina.

Na tarefa de controle de incêndio, uma das questões mais importantes tem relação com a estratégia de posicionamento dos agentes de combate. De acordo com a capacidade de atuação de cada agente, as condições climáticas (vento, chuva), a topografia e a vegetação, diferentes formações podem ser sugeridas. A Figura 1.1 apresenta quatro formações estruturadas de posicionamento. No caso de incêndios florestais, formações não estruturadas podem ser mais eficientes. Estas estratégias, quando sugeridas por especialistas, não conseguem levar em conta um número muito grande de variáveis, assim, a definição do posicionamento poderia fazer uso de técnicas de otimização e Aprendizado de Máquina (*Machine Learning* - ML). Uma técnica de ML indicada para estes casos é a de Algoritmos Genéticos (*Genetic Algorithms* - GA) (HOLLAND, 1975; MITCHELL, 1996). Algoritmos Genéticos são algoritmos de otimização global que empregam estratégia de busca paralela e estruturada, embora aleatória, direcionada a busca de pontos de aptidão (CARVALHO; BRAGA; LUDERMIR, 2003). Permitem assim a realização de busca multi-critério em um espaço multi-dimensional e, por serem não supervisionados, não necessitam de nenhuma base de informação de antemão e, se corretamente utilizados, são capazes de escapar de mínimos locais (HEINEN; OSÓRIO, 2006).

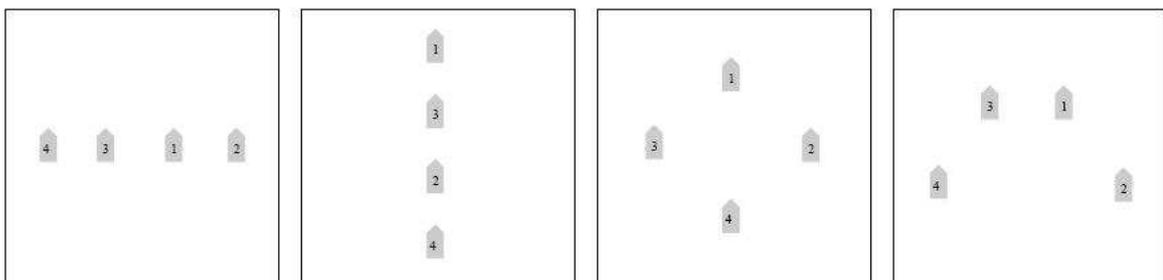


Figura 1.1: Formações para uma equipe de quatro agentes (da esquerda para a direita: linha, coluna, circular ou ferradura)(BALCH; ARKIN, 1998).

O que diferencia a robótica móvel de outras áreas de pesquisa em robótica, como a robótica de manipuladores, é sua ênfase nos problemas relacionados com a operação (locomoção) em ambientes complexos de larga escala (que se modificam dinamicamente, compostos de obstáculos estáticos e móveis). Para operar neste tipo de ambiente o robô deve ser capaz de adquirir e utilizar conhecimento sobre o ambiente, estimar uma posição, possuir a habilidade de reconhecer obstáculos e responder em tempo real as diferentes situações que pos-

sam ocorrer; todas estas funcionalidades devem operar em conjunto. As tarefas de perceber, se localizar, e se mover pelo ambiente evitando colisões são problemas fundamentais no estudo dos RMA (HEINEN, 2000; DUDEK; JENKIN, 2000). Um problema fundamental da robótica é a navegação: processo de mover-se em um ambiente de trabalho, geralmente povoado de obstáculos, de uma posição e orientação inicial para uma posição e orientação destino (HEINEN, 2000). Ir de um lugar para outro depende de três aspectos fundamentais: localização, orientação e controle motor. Para conhecer tanto sua localização como sua orientação, um RMA deve possuir sensores adequados (*e.g.* GPS, bússola). Para o controle motor, um RMA deve possuir um número adequado de motores (um veículo autônomo, por exemplo, usualmente possui um motor angular, para o controle da direção das rodas e um motor linear, para tração). Normalmente sensores e atuadores são sujeitos a erros e interferências, assim o controle das ações de um robô deve sempre levar em conta a imprecisão dos sensores e motores envolvidos. Um sistema robusto deve permitir que, mesmo com sensores e atuadores imprecisos, o agente cumpra o seu objetivo. Uma técnica de ML indicada para este caso é a de Redes Neurais Artificiais (RNA), dada sua capacidade de aprendizado a partir de exemplos e a respectiva generalização e adaptação das saídas. O uso de Redes Neurais Artificiais com o algoritmo *Backpropagation* é aconselhado quando possuímos de antemão uma base de dados que possa servir para correção de erro através do cálculo do gradiente (REEVE, 1999). É uma técnica muito utilizada no controle de navegação de sistemas reativos (OSÓRIO, 1998; MARCHI, 2001; RIBEIRO; COSTA; ROMERO, 2001; HEINEN, 2000; JUNG et al., 2005; OSÓRIO et al., 2006a).

Para que seja possível uma implementação física real, o sistema multi-agente que propomos deve ser projetado, desenvolvido e testado anteriormente em ambientes de simulação realísticos. A simulação de sistemas robóticos é especialmente necessária para robôs caros, grandes, ou frágeis (GO; BROWNING; VELOSO, 2004). É uma ferramenta extremamente poderosa para agilizar o ciclo de desenvolvimento de sistemas de controle robóticos eliminando desperdício de recursos, tanto financeiros como computacionais. Para que uma simulação seja útil, entretanto, ele deve capturar características importantes do mundo físico, onde o termo *importantes* tem relação ao problema em questão (GO; BROWNING; VELOSO, 2004). Ambientes com terrenos 3D gerados com a biblioteca de programação OSG (OSG, 2007) junto com a biblioteca de programação *Demeter* (DEMETER, 2007) permitem combinar o mapa de elevação juntamente com uma determinada distribuição de vegetação, criando assim um terreno bastante realista (OSÓRIO et al., 2006b). Além disso, o uso da biblioteca de programação ODE (ODE, 2007) permite implementar atributos físicos de cinemática e dinâmica (*e.g.* atrito, fricção, gravidade, colisão) tornando o sistema ainda mais realístico.

Em nossos estudos e pesquisas, pelo que podemos constatar, não existe um ambiente

de simulação aberto que combine todas as características requisitadas para este trabalho que são: (i) realismo físico de modelagem robótica; (ii) realismo físico de interação de agentes em terrenos 3D não planos¹; (iii) facilidades de comunicação entre os agentes; (iv) capacidade de aplicação de diversos métodos de controle (*e.g.* Regras, Redes Neurais Artificiais); (v) capacidade de aplicação de métodos de evolução de estratégias (*e.g.* Algoritmos Genéticos); e (vi) simulação de processos naturais (como a propagação do fogo). Um dos objetivos desta dissertação é o desenvolvimento deste ambiente.

1.3 Objetivos e metodologia

Este trabalho está relacionado com a aplicação de técnicas de Inteligência Artificial no desenvolvimento de um Sistema Multi-Agente robótico aplicado ao problema da monitoração e combate a incêndios em áreas florestais. O objetivo macro é evoluir estratégias de posicionamento de equipes de combate a incêndio em uma unidade monitora (agente deliberativo) e criar métodos robustos de navegação em unidades de combate fisicamente simuladas (agentes híbridos), considerando terreno e vegetação.

Os objetivos gerais desta dissertação são:

- Criar um modelo de evolução de estratégias de posicionamento em uma unidade monitora, usando técnicas de Algoritmos Genéticos. Este agente deliberativo (simulado como um satélite ou torre de vigilância automatizada) deve identificar focos de incêndio, ativar um módulo de evolução, e através do envio de mensagens deve ser capaz de solicitar a formação de uma equipe de robôs móveis que realize o combate do incêndio. A evolução de estratégias refere-se, especificamente, a evolução de um vetor de posições para um grupo de robôs de combate. O uso de Algoritmos Genéticos deve permitir a otimização dos parâmetros: (i) posição chave inicial no combate de cada agente do time; e (ii) posição chave final no combate de cada agente do time. Esta otimização tem relação com: (i) quantidade de agentes disponíveis; (ii) posição do foco de incêndio; (iii) orientação do vento; e (iv) velocidade de propagação do fogo (que é relacionada com tipo de vegetação e com a velocidade do vento). A função *fitness* a ser explorada tem relação com a área de vegetação salva e com a minimização da área de atuação das unidades de combate.
- Criar um modelo robusto de navegação em unidades de combate, usando Redes Neurais Artificiais. Estes agentes, de comportamento híbrido, realizam deslocamento de sua

¹É relativamente comum que aplicações 3D usem chão plano (completamente regular), porém, neste trabalho o chão é um terreno irregular, como um ambiente natural.

posição e orientação inicial (base de operação) até sua posição chave de combate, solicitada pela unidade monitora; usando apenas informações obtidas localmente. Para tanto, devem possuir uma quantidade de sensores e motores simulados a ser estudada. Os agentes de combate a incêndios serão fisicamente simulados com a biblioteca de simulação física *Open Dynamics Engine* (ODE) (ODE, 2007) e terão atuação em um ambiente 3D irregular desenvolvido com a biblioteca gráfica *Open Scene Graph* (OSG) (OSG, 2007) e a biblioteca de criação de terrenos *Demeter* (DEMETER, 2007). A integração das bibliotecas OSG (OSG, 2007), ODE (ODE, 2007) e *Demeter* (DEMETER, 2007) permite a criação de um ambiente bastante realístico.

Como objetivos específicos temos a realização dos seguintes trabalhos:

- Recolher informações sobre dados florestais, tipos de vegetação, topografia e comportamento de incêndios;
- Pesquisar ferramentas e técnicas de combate a incêndios florestais utilizadas por bombeiros;
- Implementar o ambiente de simulação com OSG, ODE e *Demeter*, reproduzindo de forma realística um ambiente natural e um modelo simplificado de propagação de incêndio;
- Desenvolver a arquitetura de controle e de gestão de estratégia do sistema multi-robótico, completamente autônomo;
- Implementar os robôs bombeiros, fisicamente simulados, junto ao ambiente, com o conjunto completo de sensores e atuadores; bem como sua capacidade de atuação no combate ao incêndio;
- Desenvolver e avaliar uma Rede Neural Artificial de controle de navegação e desvio de obstáculos para robôs móveis (unidades de combate); esta RNA deve ter como entrada apenas informações obtidas através de sensores; deve também permitir a operação sem nenhum autômato auxiliar;
- Desenvolver e avaliar um Algoritmo Genético para evolução de estratégias para diferentes características do ambiente (*e.g.* quantidade de robôs disponíveis, velocidade e orientação do vento).

1.4 Organização do texto

Esta dissertação está organizada em 6 Capítulos, que abordam a base conceitual, trabalhos relacionados, o modelo proposto, experimentos e resultados, bem como as considerações finais. A estrutura e os conteúdos abordados neste documento, em cada um de seus Capítulos, estão resumidos abaixo:

- O Capítulo 2 tem por objetivo revisar conceitos e tecnologias importantes relacionadas ao problema proposto. Está dividido em quatro Seções, a primeira trata de robótica móvel. São detalhados os princípios envolvidos em sistemas de controle, sensores e atuadores, além de descrever aspectos de morfologia e de simulação de sistemas multi-robóticos fisicamente realísticos. A segunda Seção trata de agentes e de sistemas multi-agentes, onde serão apresentados conceitos e características importantes que tenham relação com a criação de sistemas multi-robóticos. A terceira Seção trata de Aprendizado de Máquina, onde serão detalhadas as principais técnicas abordadas neste trabalho, que são Algoritmos Genéticos e Redes Neurais Artificiais. Na última Seção, sobre incêndios em ambientes naturais, serão descritas as técnicas reais de operação, as ferramentas e os métodos utilizados atualmente por grupos humanos na identificação e combate de incêndios florestais.
- O Capítulo 3 descreve trabalhos desenvolvidos em outros projetos ou outros centros de pesquisa que tenham relação com o foco desta dissertação.
- O Capítulo 4 apresenta o modelo do sistema multi-robótico proposto nesta dissertação. Detalha a arquitetura do sistema proposto e a morfologia dos robôs móveis. Além disso, descreve como serão utilizadas as técnicas de Algoritmos Genéticos na evolução de estratégias e de Redes Neurais Artificiais no controle de navegação.
- O Capítulo 5 descreve os diversos experimentos realizados com os protótipos desenvolvidos. Iniciamos descrevendo os experimentos e as análises sobre o uso do Algoritmo Genético proposto para realizar a evolução de estratégias de posicionamento de grupos robóticos. Em seguida detalhamos o treino e teste sobre as diversas topologias de RNA propostas. Após, apresentamos os experimentos e as análises com o uso da RNA escolhida para o controle inteligente de navegação dos agentes de combate a incêndio, finalizando com experimentos de navegação com aplicação de erro simulado nos sensores e atuadores.
- O Capítulo 6 apresenta as considerações finais da dissertação. Serão revisados os objetivos e conceitos que motivaram o desenvolvimento deste trabalho. Serão detalhadas as

contribuições resultantes, finalizando com a descrição dos trabalhos futuros vislumbrados durante o desenvolvimento desta pesquisa.

A dissertação proposta neste documento tem como base cinco áreas de conhecimento: (i) simulação realística com robótica móvel; (ii) sistemas multi-agentes; (iii) aprendizado de máquina aplicado a robótica; (iv) simulação de ambientes naturais; e (v) aperfeiçoamento de técnicas reais de monitoração e combate de incêndios em ambientes naturais.

2 BASE CONCEITUAL

Este Capítulo tem por objetivo revisar conceitos e tecnologias importantes relacionadas ao problema proposto. Está dividido em quatro Seções, a primeira trata de robótica móvel. São detalhados os princípios envolvidos em sistemas de controle, sensores e atuadores, além de descrever aspectos de morfologia e de simulação de sistemas multi-robóticos fisicamente realísticos. A segunda Seção trata de agentes inteligentes e de sistemas multi-agentes, onde serão apresentados conceitos e características importantes que tenham relação com a criação de sistemas multi-robóticos. A terceira Seção trata de Aprendizado de Máquina, onde serão detalhadas as principais técnicas abordadas neste trabalho, que são Algoritmos Genéticos e Redes Neurais Artificiais. A última Seção, sobre incêndios em ambientes naturais, descreve as técnicas reais de operação, as ferramentas e os métodos utilizados atualmente por grupos humanos na identificação e combate a incêndios florestais.

2.1 Robótica Móvel

2.1.1 Introdução

Um robô móvel é um dispositivo eletromecânico montado sobre uma base não fixa, que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente (MARCHI, 2001; BEKEY, 2005). A interação com o ambiente se dá através de ciclos de percepção-ação que consistem em três passos fundamentais (RIBEIRO; COSTA; ROMERO, 2001):

1. Obtenção de informação através de sensores;
2. Processamento das informações para seleção da ação;
3. Execução da ação através do acionamento dos atuadores.

Este conjunto de operações, em uma análise superficial, pode parecer simples, porém o controle robusto de sistemas robóticos tem complicações físicas e mecânicas (*e.g.* cinemática

e dinâmica), eletrônicas (*e.g.* falta de precisão de sensores) e computacionais que tornam a criação de um conjunto de regras de controle uma tarefa árdua e sujeita a erros.

Esta Seção tem por objetivo situar o leitor no mundo da robótica, seus desafios e conceitos. Além de apresentar trabalhos que demonstram o avanço dos sistemas robóticos atuais.

2.1.2 Classificações

Para que possamos desenvolver agentes robóticos apropriados ao tratamento de nosso problema, é interessante conhecer as formas atuais da classificação. Diversas taxonomias têm sido utilizadas para classificar robôs móveis, nenhuma que seja definitiva. De uma forma geral, podemos agrupar os robôs existentes de acordo com três aspectos: anatomia, tipo de controle e funcionalidade (PIERI, 2002).

2.1.2.1 Anatomia

Quanto a anatomia, os robôs móveis podem ser classificados em três grandes grupos (Figura 2.1): robôs aéreos, aquáticos e terrestres.

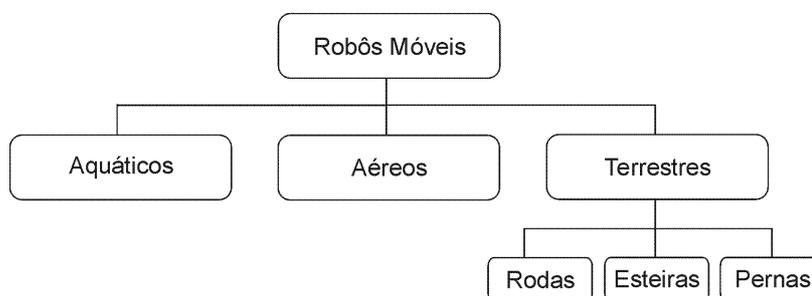


Figura 2.1: Classificação segundo a anatomia (PIERI, 2002).

Robôs aéreos são geralmente aeromodelos, balões ou dirigíveis. Podem ser equipados com câmeras de vídeo e usualmente apresentam utilidade na monitoração de grandes áreas, como o projeto *SiDeVAAN* (DEMEC/UFMG, 2007) (Figura 2.2), ou os protótipos militares *Boeing X-45* (BOEING, 2007) (Figura 2.3(a)) que, sob supervisão humana em terra (sem nenhum piloto embarcado), está sendo testado para missões de combate e o *nEUROn* (DASSAULT, 2007) (Figura 2.3(b)), protótipo europeu com a mesma finalidade.

Robôs aquáticos são geralmente plataformas equipadas com propulsores e balões de ar que permitem ao robô permanecer submerso. O *Depthx* (WETTERGREEN, 2007) (Figura 2.4(a)) é fruto de um projeto para criação de um robô autônomo de mapeamento de solo sub-aquático. O *Isis* (Figura 2.4(b)) (NOC, 2007) é um veículo remotamente operado, fruto de um projeto



Figura 2.2: Aeronave automatizada brasileira, fruto do projeto *SiDeVAAN* (DEMEC/UFMG, 2007), para ser usada em pulverização de lavouras, inspeção de linhas de transmissão ou em missões de controle do espaço aéreo.



Figura 2.3: Robôs aéreos de combate: (a) Protótipo *X-45* americano (BOEING, 2007); (b) Protótipo *nEUROn* (DASSAULT, 2007), da comunidade europeia.

inglês de exploração marítima. Outros modelos, mais simples, servem para monitoração de tanques de combustíveis (VIDEORAY, 2007). Um modelo híbrido, que se desloca em água e terra pode ser visto em (ENGADGET, 2007).

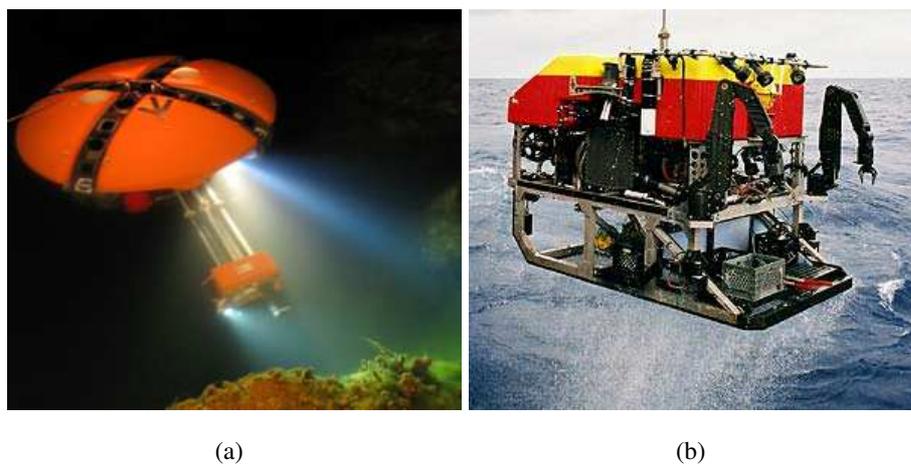


Figura 2.4: Robôs aquáticos de exploração: (a) *Depthx* (WETTERGREEN, 2007); (b) *Isis* (NOC, 2007).

Robôs terrestres são relativamente populares, geralmente utilizam pernas (Figura 2.5), rodas (Figura 2.6) ou esteiras (Figura 2.7). Robôs com rodas não necessitam de hardware tão complexo quanto os robôs com pernas (PIERI, 2002). Assim, robôs com rodas necessitam de uma quantidade de energia menor para trabalhar, porém, dependendo do terreno, podem ter um desempenho não satisfatório. Os robôs de esteiras, em geral, são utilizados em ambientes irregulares, com solo fofo e pedras. Apresentam a desvantagem de dissipar energia causada pelo movimento de giro da própria esteira e pelas rodas no seu interior (PIERI, 2002). Existem ainda robôs conceito que são desenvolvidos para explorar características particulares do ambiente, como o *Tumbleweed* (JPL/NASA, 2007) que é completamente esférico e depende de vento para se movimentar. Em um robô com roda, esta deve possuir raio igual ou maior aos obstáculos que ele irá transpor (caso não desvie destes). Podemos perceber aqui que o tipo de locomoção de um robô móvel é diretamente associado com o tipo de ambiente de operação.

Em terrenos abertos e matas não muito densas, é usual a utilização de veículos com rodas ou esteiras. Neste trabalho optamos por trabalhar com veículos terrestres com rodas.

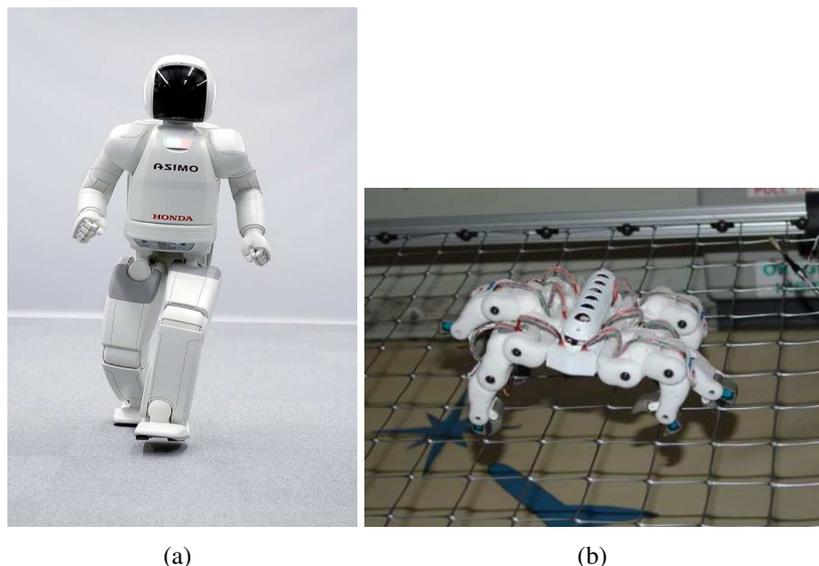


Figura 2.5: Robôs com pernas: (a) *Asimo* (HONDA, 2007); (b) *Spiderbot* (JPL/NASA, 2007), robô para trabalho fino em ambientes confinados.

2.1.2.2 Tipo de controle

Segundo o tipo de controle, robôs podem ser separados em três categorias:

- **Tele-operados:** O operador controla remotamente todos os movimentos que o robô deve fazer.



Figura 2.6: *R-Gator* (IROBOT, 2007), um veículo que permite tele-operação, navegação automática entre pontos tipo “ensinar-repetir”, desvio de obstáculos e operação manual.

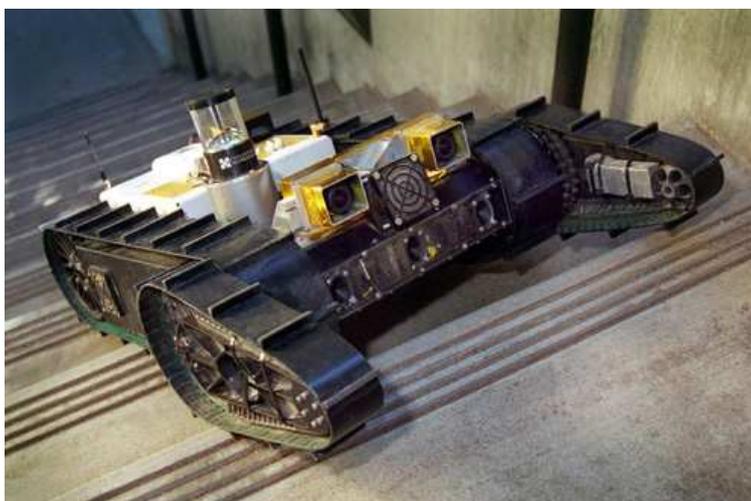


Figura 2.7: *Urbie* (JPL/NASA, 2007), robô para fins militares de reconhecimento de terrenos urbanos.

- **Semi-autônomos:** O operador indica o macro comando a ser executado e o robô o realiza sozinho.
- **Autônomos:** O robô realiza a tarefa sozinho, tomando suas próprias decisões baseando-se nos dados obtidos do ambiente.

2.1.2.3 Funcionalidade

Quanto a funcionalidade dos robôs móveis, podemos obter grupos de acordo com o tipo de atuação, o nível de autonomia e o ambiente de atuação.

- **Robôs industriais:** Em geral, robôs móveis industriais são plataformas móveis utiliza-

das para tarefas pesadas, como transporte de materiais e produtos finais em sistemas de manufatura. São usualmente programados para seguir linhas desenhadas no chão. O ambiente é completamente estruturado e ajustado para a execução da tarefa e o robô tem conhecimento exato da sua posição e da posição dos objetos.

- **Robôs de serviço:** Devem possuir um modelo do ambiente e possuir certo nível de autonomia para operar em ambientes dinâmicos, assim podem desviar de pessoas e objetos. Estes robôs recebem macro-comandos da tarefa que devem realizar, como o aspirador de pó autônomo *Roomba* (IROBOT, 2007) (Figura 2.8(a)). Os robôs de serviço de campo normalmente precisam de muito mais informação sensorial para ter comportamento adequado (Figura 2.8(b)).

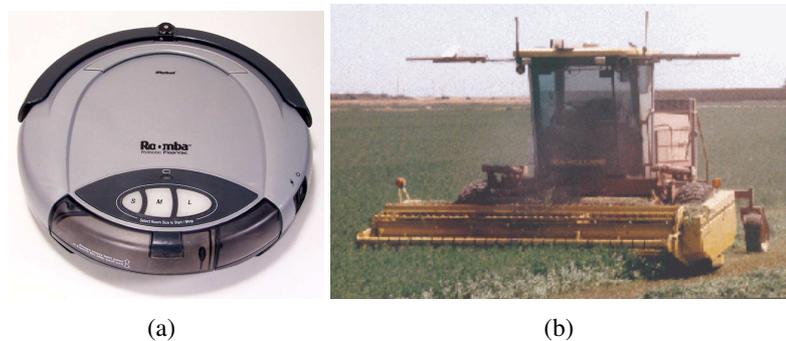


Figura 2.8: Robôs de serviço: (a) Aspirador de pó *Roomba* (IROBOT, 2007); (b) Colheitadora autônoma *Demeter* (RI/CMU, 2007).

- **Robôs de campo:** Trabalham em ambientes completamente não estruturados, desconhecidos e geralmente perigosos. As principais atividades destes robôs são a exploração de ambientes hostis (*e.g.* espacial, cavernas, vulcões) e tarefas como limpeza de resíduos nucleares. A Figura 2.9(a) apresenta o robô *Mars Exploration Rover* (JPL/NASA, 2007), de exploração espacial e a Figura 2.9(b) apresenta o robô *Dante II*, de exploração de cavernas e vulcões.
- **Robôs Pessoais:** Não desenvolvem tarefas específicas. Sua principal função é interagir com os humanos com finalidade de entretenimento (Figura 2.10).

2.1.3 Desafios da robótica móvel

Neste trabalho, os robôs móveis autônomos serão desenvolvidos em um simulador que respeita propriedades físicas e mecânicas e possuirão um conjunto adequado de sensores e atuadores. A construção de um robô fisicamente simulado é quase tão árdua quanto a construção

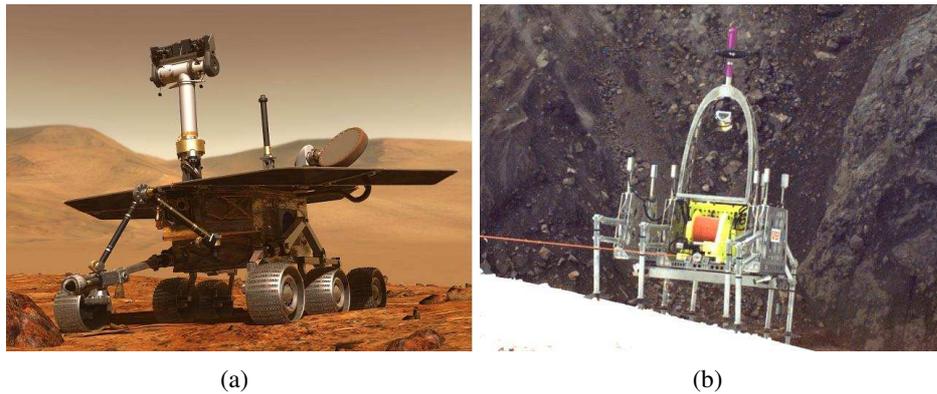


Figura 2.9: Robôs de campo: (a) *Mars Exploration Rover* (JPL/NASA, 2007) para exploração espacial; (b) *Dante II* (JPL/NASA, 2007) para exploração de cavernas e vulcões.

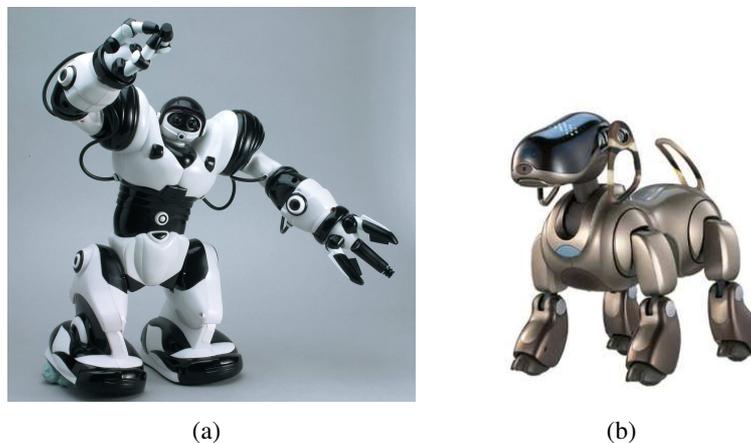


Figura 2.10: Robôs pessoais operados por controle remoto: (a) *RoboSapien* (WOWWEE, 2007), pequeno robô humanóide; (b) *Cão AIBO* (SONY, 2007).

de um robô real, salvo algumas restrições de projeto. Os desafios básicos da robótica móvel, seja na implementação de um sistema real ou simulado, são os seguintes (PIERI, 2002):

- **Reconhecimento:** Como fazer um dispositivo robótico capaz de reconhecer o ambiente no qual está inserido? Existem sérias restrições computacionais com relação ao processamento de imagens. Diferenças de luminosidade e ruído nos sensores adicionam ainda mais possibilidades de erros.
- **Navegação:** Como criar formas que permitam a representação de um ambiente? Até que ponto um ambiente precisa ser refinado e livre de erros? Até que ponto podemos utilizar a navegação reativa e ainda assim maximizar o desempenho do robô?
- **Aprendizado:** Adicionar sensores e atuadores a um robô adiciona complexidade ao trabalho de programação, dificultando a tarefa de calibração dos sensores e o controle de

ações. Técnicas de Aprendizado de Máquina devem permitir que um robô desempenhe a sua tarefa de forma autônoma e adaptável.

- **Cooperação:** Um grupo de robôs pequenos e baratos pode realizar qualquer tipo de tarefa complexa apenas usando cooperação?
- **Cognição:** Até que ponto podemos criar formas para armazenar conhecimento e utilizá-lo para a tomada de decisão dotando o robô de capacidades cognitivas de alto nível que o permitam ter total autonomia com relação ao meio?

Outra maneira de pensar em desafios da robótica foi sugerida por (RUSSEL; NORVIG, 1995) e se deve as seguintes constatações:

- **O mundo é inacessível:** Sensores não são perfeitos em perceber o mundo real.
- **O mundo é não-determinístico:** Devido às incertezas envolvidas, o resultado final não pode ser completamente conhecido.
- **O mundo é não-episódico:** Os efeitos de uma ação mudam com o passar do tempo, problemas relacionados a decisão e aprendizado devem ser tratados.
- **O mundo é dinâmico:** É preciso saber quando planejar e quando agir.
- **O mundo é contínuo:** A evolução dos estados internos da programação do robô evoluem de forma discreta enquanto que o mundo ao redor é contínuo, dificultando a programação e a estimativa de ações e comportamentos.

No decorrer deste trabalho são apresentadas propostas e experiências que abordam estes desafios básicos. Como, por exemplo, a quantidade adequada de sensores e atuadores que permitam reconhecimento do ambiente e navegação com desvio de obstáculos, técnicas de IA para o aprendizado, e regras de cooperação.

2.1.4 Sensores e atuadores

Sensores são os mecanismos de percepção de um robô. Realizam medições físicas (*e.g.* contato, distância, luz, som, rotação, tensão, magnetismo, cheiro, temperatura, inclinação, altitude) que provém sinais ou dados crus que precisam ser interpretados pelo “cérebro” do robô. A interpretação destes sinais deve ser a única maneira de um robô autônomo entender o ambiente que o cerca para poder realizar as mudanças de ação necessárias. A Figura 2.11 apresenta os sensores utilizados em um *Mars Exploration Rover* (JPL/NASA, 2007).

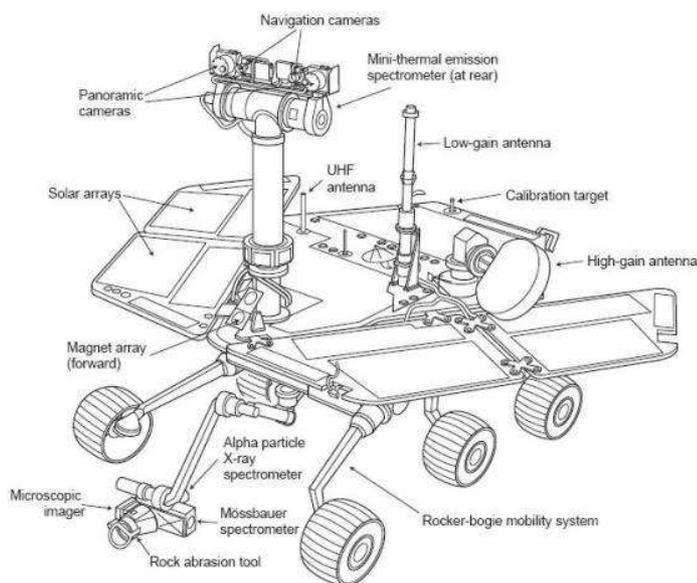


Figura 2.11: Visão esquemática de um *Mars Exploration Rover* (JPL/NASA, 2007).

Atuadores são os mecanismos de ação de um robô, controlados por circuitos eletrônicos que recebem valores de ação que devem ser calculados pelo “cérebro” do robô e devem estar de acordo com as especificações de cada fabricante. Os atuadores são as ferramentas de ação (*e.g.* motores, pistões, braços manipuladores), podem servir para mobilidade, bem como para o posicionamento de sensores e de ferramentas (como uma ferramenta de análise científica colocada no final de um braço robótico).

No projeto de um RMA, a escolha de sensores deve considerar, para o foco da missão do robô, os seguintes detalhes (EVERETT, 1995; BORENSTEIN et al., 1997): campo de visão, capacidade de alcance, acurácia, tempo de resposta, facilidade na interpretação dos dados, redundância, modularidade, simplicidade, consumo de energia e tamanho.

Quanto aos tipos de sensores, é interessante conhecer para este trabalho:

- **GPS:** Sistema de Posicionamento Global (*Global Positioning System* - GPS) desenvolvido pelo Departamento de Defesa dos Estados Unidos. Seu funcionamento é baseado em um conjunto de satélites artificiais emissores de um sinal de rádio-frequência. Estes sinais são captados por aparelhos receptores que calculam as coordenadas de uma posição na Terra baseado na triangularização das unidades emissoras. Aparelhos de GPS atuais permitem a aquisição de vários dados derivados da localização, além do posicionamento propriamente dito, como orientação (bússola digital), odômetro e altímetro.
- **Odômetro:** Deve realizar a contagem da distância percorrida em um trajeto. Mecânica-

mente funciona como um contador de giros de roda. Modelos utilizando GPS existem também, realizam a contagem da distância percorrida realizando cálculos de distância vetorial entre os pontos de deslocamento do GPS.

- **Bússola:** Pela bússola obtemos a orientação de um objeto em relação ao Norte. As bússolas do tipo mecânico possuem uma seta magnetizada que indica o Norte Magnético. Bússolas eletrônicas (derivadas de GPS) indicam o Norte Geográfico. Os graus de uma bússola vão de 0 a 360 em sentido horário, sendo 0 ou 360 graus o Norte, 180 graus o Sul, 90 graus Leste e 270 graus Oeste. A direção em que se encontra um objeto ao redor de um observador, em relação ao Norte, é chamada de Azimute. Ao considerar a Crosta Terrestre como um Plano Cartesiano, o deslocamento positivo de um objeto no Eixo X pode ser traduzido como deslocamento no Azimute de 90 graus.
- **Inclinômetro:** Permite obter o grau de inclinação de um objeto em relação ao Eixo de Gravidade (DUDEK; JENKIN, 2000). A informação do inclinômetro tem relação direta com o equilíbrio dos objetos e, além disso, permite obter informações sobre a inclinação do terreno onde se encontra o RMA.
- **Bumpers:** São sensores de contato, utilizados para simular o sentido do tato. Os modelos mais simples são interruptores que retornam valores binários (aberto/fechado) (DUDEK; JENKIN, 2000).
- **Anemoscópio:** Permite obter a direção do vento.
- **Anemômetro:** Permite obter a velocidade do vento.
- **Pluviômetro:** Permite obter a quantidade de chuva em um período de tempo.
- **Infravermelho:** Permite obter a distância na qual nos encontramos de um objeto. Existem muitos modelos no mercado e são relativamente baratos. Funcionam com um diodo emissor de luz, que, ao atingir um objeto tem uma porção de luz refletida captada através de um receptor ótico. Dependendo da posição do objeto, o ângulo de incidência da luz refletida é diferente, com isso pode-se calcular a distância deste objeto por triangulação (HEINEN, 2002). O ângulo de ação é, em geral, bastante estreito, sendo assim normalmente simulado como um único raio ou um pequeno cone. Por exemplo, o sensor *GP2Y3A003K0F* (SHARP, 2007) é formado por 5 diodos emissores, com 5 graus de amplitude cada e é capaz de medir distâncias entre 40cm e 300cm.
- **Laser:** Permite obter a distância na qual nos encontramos de um objeto. Utiliza o mesmo processo de emissão e recepção de luz dos sensores infravermelhos. São sensores mais

caros que os sensores de infravermelho, porém são muito mais precisos e tem alcance bem maior, como o modelo *LDM 41 A* (Figura 2.12) (ASTECH, 2007) que é capaz de medir distâncias entre 0,1m e 30m de superfícies naturais ou até 100m dependendo da refletividade da superfície, com erro de até 5mm.



Figura 2.12: Sensor Laser *LDM 41 A* (ASTECH, 2007).

- **Sonar:** Permite obter a distância na qual nos encontramos de um objeto. É um sensor bastante utilizado em robótica móvel devido a baixo custo e a pouca necessidade de recursos computacionais (HEINEN, 2002). Seu funcionamento consiste em um transdutor que emite uma onda sonora de alta frequência. Quando esta onda atinge um objeto, ela reflete e é captada novamente pelo transdutor. A distância entre o transdutor e o objeto pode ser calculada pelo tempo entre a emissão e o recebimento da onda (HEINEN, 2002). O sensor *Polaroid 6500* (POLAROID, 2007) permite medir distâncias entre 15cm e 11m.

Quanto aos tipos de atuadores, é interessante conhecer para este trabalho:

- **Locomoção:** Atuadores de locomoção podem ser: motor de passo (para velocidade, direção, rotação), perna (alto consumo de energia, problema do equilíbrio), motor de propulsão (para submarinos e veículos aéreos).
- **Manipulação:** Atuadores de manipulação podem ser: garras, pinças e braços robóticos.

É bastante comum na robótica móvel a utilização de sensores do tipo sonar para obtenção de distâncias, assim, nosso veículo contará com este tipo de sensor. Para este trabalho, simularemos também os sensores bússola e GPS. Para a navegação autônoma, cada veículo será implementado com dois atuadores. A morfologia dos veículos será detalhada na Seção 4.2.

2.1.5 Controle

2.1.5.1 Estratégias

Estratégias de controle são as maneiras de obter informações sobre o estado de um sistema robótico ou de um ambiente no qual um robô está inserido para poder tomar decisões e

agir. Podem fazer uso de sensores inseridos no sistema controlado, no ambiente, em ambos ou em nenhum. As técnicas de controle citadas por (SMITHERS, 1990; HALLAM, 1990) são:

- **Open Loop:** Não utiliza nenhum sensor. Cálculos baseados em modelos físicos (*e.g* aceleração, gravidade, inércia) são utilizados para prever e obter informações sobre o estado do sistema ou do ambiente. Este modelo é utilizado em ambientes estáticos e previsíveis.
- **Feedforward:** Utiliza sensores somente para perceber o ambiente. Medições do ambiente são utilizadas para inicializar variáveis no modelo do sistema. Este modelo é utilizado em ambientes previsíveis.
- **Feedback:** É o sistema de controle mais utilizado na robótica. Medições contínuas dos sensores devem permitir a atualização dos comandos de ação dos atuadores.

Os agentes de combate a incêndio implementados neste trabalho realizam navegação com constante leitura de informação sensorial, sendo assim essencialmente do tipo *feedback*; o agente deliberativo (planejador de estratégia) receberá informação do ambiente apenas na inicialização, assim, pode ser considerado como possuindo estratégia de controle do tipo *feed-forward*.

2.1.5.2 Arquitetura

Sistemas planejados para controlar a execução de tarefas complexas requerem uma arquitetura computacional de controle (JUNG et al., 2005; MEDEIROS, 1998). As tarefas da arquitetura de controle podem envolver (JUNG et al., 2005):

- Capacidade para ler e interpretar os sinais recebidos dos sensores do veículo.
- Capacidade de evitar obstáculos presentes no caminho do veículo.
- Capacidade para reagir a eventos, inclusive imprevistos.
- Capacidade de planejamento e execução de tarefas, tendo conhecimento ou não, de um mapa do ambiente.
- Capacidade de gerenciar os diversos componentes do sistema, de modo a gerar os comandos na ordem adequada e com os parâmetros corretos, a fim de que a tarefa que foi planejada possa ser executada.

O sistema computacional de controle deve, portanto, executar tarefas como: preservar a integridade do veículo; preservar a integridade das entidades presentes no ambiente onde o veículo atua; planejar soluções para a execução de tarefas; e, em alguns casos, interagir com outros sistemas (JUNG et al., 2005).

As arquiteturas de controle mais citadas na literatura (RUSSEL; NORVIG, 1995; WOOLDRIDGE; JENNINGS, 1995; MEDEIROS, 1998; DUDEK; JENKIN, 2000; HEINEN, 2002; JUNG et al., 2005) serão detalhadas a seguir.

- **Controle Reativo:** Consiste de um sistema simples de reação sensorial-motora. O modelo reativo puro não necessita de muitos recursos computacionais para sua implementação. Deve existir um laço de: (i) leitura dos sensores; (ii) imediato processamento destas informações; e, (iii) geração de comandos para os atuadores. Usualmente um esquema de controle reativo considera apenas as leituras sensoriais realizadas no instante atual para a geração de comandos de saída (WOOLDRIDGE; JENNINGS, 1995; MEDEIROS, 1998; SANTOS, 2004). Um sistema reativo é bastante útil para implementar comportamentos como desvio de obstáculos (reagindo a presença de um obstáculo) ou de seguir um objeto (acompanhar um elemento guia) (JUNG et al., 2005). A Figura 2.13 (OSÓRIO, 1998) apresenta um esquema que pode ser considerado como um sistema de controle reativo, caso a tomada de decisão do agente-robô não utilize mapas, memória, nem qualquer outra informação adicional, fazendo uso apenas das informações sensoriais para gerar uma nova ação.
- **Controle Deliberativo:** O controle deliberativo consiste da aplicação de um mecanismo de planejamento de ações, onde é feito um plano prévio de execução de uma seqüência de ações, baseado no conhecimento que o sistema possui sobre o problema a ser resolvido, como o mapa do ambiente com as rotas disponíveis (JUNG et al., 2005). No controle deliberativo é assumida a existência de um processo de alto nível de raciocínio e tomada de decisões, usualmente mais complexo de ser implementado do que o controle reativo. Este processo permite que sejam planejadas ações de modo a tratar e executar tarefas que exigem um nível de controle mais sofisticado, como por exemplo, definir (traçar uma rota) e executar a tarefa de se deslocar de um ponto a outro do ambiente considerando o conhecimento do mapa. O controle deliberativo puro possui limitações quando colocado frente a eventos imprevistos, como por exemplo, um obstáculo que se moveu obstruindo a sua rota (HEINEN, 2000), neste caso, o controle deliberativo puro terá dificuldades de reagir a uma nova configuração do ambiente. A Figura 2.14 (HEINEN, 2002) apresenta um ambiente com sistema de controle deliberativo, o agente, por conhecer o mapa, testa

previamente vários trajetos e segue pelo melhor.

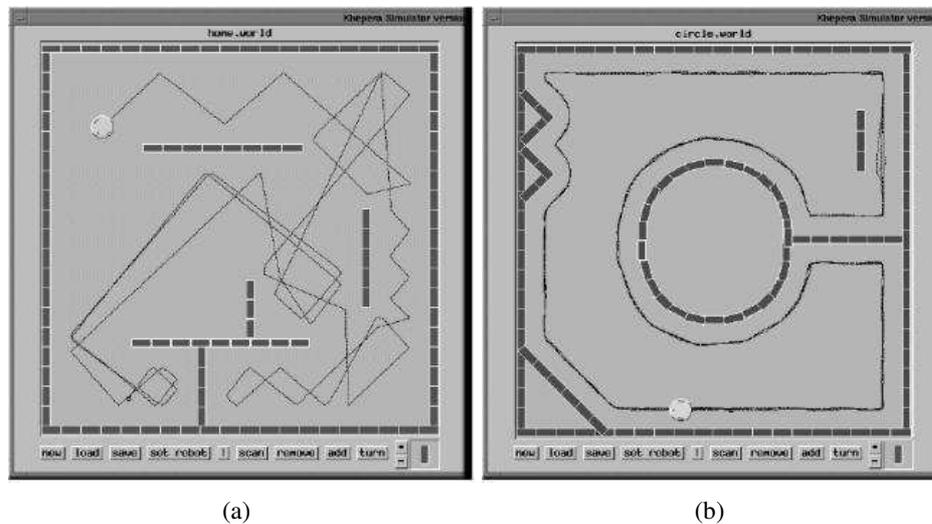


Figura 2.13: Controle reativo (OSÓRIO, 1998): (a) Desvio de paredes; (b) Seguindo paredes.

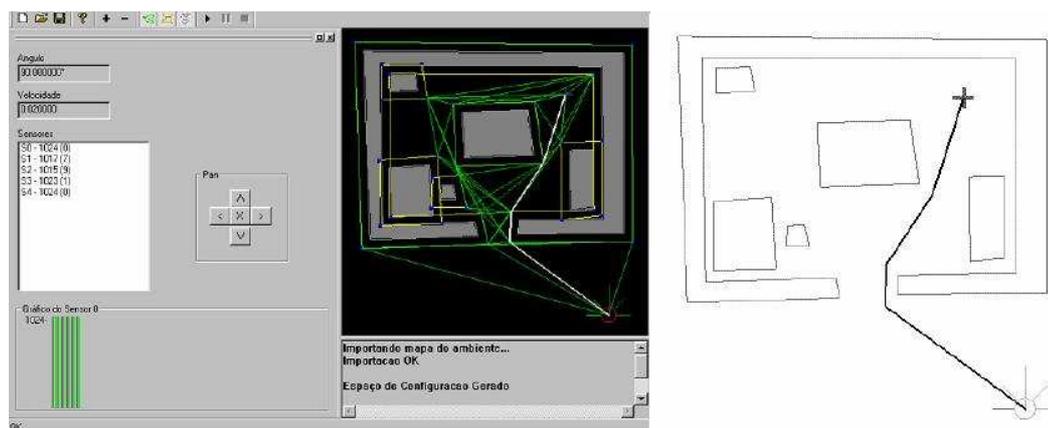


Figura 2.14: Comportamento deliberativo com o planejamento e execução de trajetória considerando o mapa do ambiente (HEINEN, 2000).

- **Controle Híbrido:** Consiste na combinação de múltiplos módulos de controle reativo e deliberativo em camadas dispostas de modo que estes possam operar de modo hierárquico ou paralelo. Os sistemas híbridos apresentam a vantagem de poder combinar os comportamentos obtidos de seus diferentes módulos a fim de obter um comportamento mais robusto e uma execução de tarefas mais complexas. Um exemplo de aplicação deste tipo de sistema de controle seria a composição do planejamento e execução de uma tarefa de navegação do ponto A ao ponto B, onde um módulo reativo estaria encarregado de preservar a integridade do robô, desviando dos obstáculos que forem percebidos através de seus sensores. Neste exemplo, pode-se imaginar que o robô passa para um controle reativo ao detectar um obstáculo que esteja dificultando seu deslocamento, e uma vez “resolvido o

problema”, ele retorna a execução de sua tarefa principal, onde pode inclusive solicitar ao módulo deliberativo um novo planejamento de trajetória (JUNG et al., 2005).

Neste trabalho, o agente responsável pela evolução de estratégias é essencialmente deliberativo; os agentes de combate a incêndio, por receberem um macro-comando de “*desloque-se autonomamente até (x,y)*” possuem um plano para execução desta navegação, além de possuir um sistema de reação sensorial-motora, assim, podem ser considerados como agentes híbridos.

2.1.6 Simulação e modelagem

Experimentos em robótica móvel podem ser realizados de duas formas: diretos em um robô real ou em um robô simulado em um ambiente virtual realista (PFEIFER; SCHEIER, 1999). Usualmente, experimentos em robótica móvel utilizando um robô real exigem enorme despendimento de tempo e de recursos financeiros, portanto, como mencionado anteriormente, a simulação de sistemas robóticos é especialmente necessária para robôs caros, grandes ou frágeis (GO; BROWNING; VELOSO, 2004). É uma ferramenta extremamente poderosa para agilizar o ciclo de desenvolvimento de sistemas de controle robóticos eliminando desperdício de recursos, tanto financeiros como computacionais. Porém, para que uma simulação seja útil, ela deve capturar o máximo possível de características relacionadas ao sistema desenvolvido, ou seja, as características fundamentais dos agentes e do ambiente que tenham impacto na implementação física real do robô, no seu ambiente de atuação. No caso deste trabalho, é fundamental que existam restrições físicas no modelo e que exista a possibilidade de trabalho em um terreno irregular.

Para este trabalho, foram pesquisadas algumas ferramentas de simulação. Nenhuma das ferramentas pesquisadas mostrou possuir o conjunto completo das características requisitadas, as quais são: (i) realismo físico de modelagem robótica; (ii) realismo físico de interação de agentes em terrenos 3D irregulares; (iii) facilidades de comunicação entre os agentes; (iv) capacidade de aplicação de diversos métodos de controle (*e.g.* Regras, Redes Neurais Artificiais); (v) capacidade de aplicação de métodos de evolução de estratégias (*e.g.* Algoritmos Genéticos); e (vi) simulação de processos naturais (como a propagação do fogo). Assim, decidimos por criar nosso próprio ambiente de simulação integrando estas características. As ferramentas pesquisadas serviram como inspiração para o desenvolvimento do nosso ambiente, assim, serão brevemente descritas a seguir.

2.1.6.1 Ferramentas Pesquisadas

A versão 1.0 do *Microsoft Robotics Studio* (MICROSOFT, 2007) (Figura 2.15) foi liberada em dezembro de 2006, sendo em maio de 2007 liberada a atualização para versão 1.5. O *Microsoft Robotics Studio* é um ambiente para desenvolvimento de simulação robótica e usa uma biblioteca de simulação física. Permite a simulação em terrenos irregulares. É livre apenas para uso não comercial e, por padrão, utiliza uma linguagem de programação visual (*Visual Programming Language*). Por ser um ambiente novo, poucas referências foram encontradas além das do fabricante. O desenvolvimento dos sensores, dos controles inteligentes e da comunicação entre os robôs ainda depende de programação (*e.g. C#, VB.NET*).

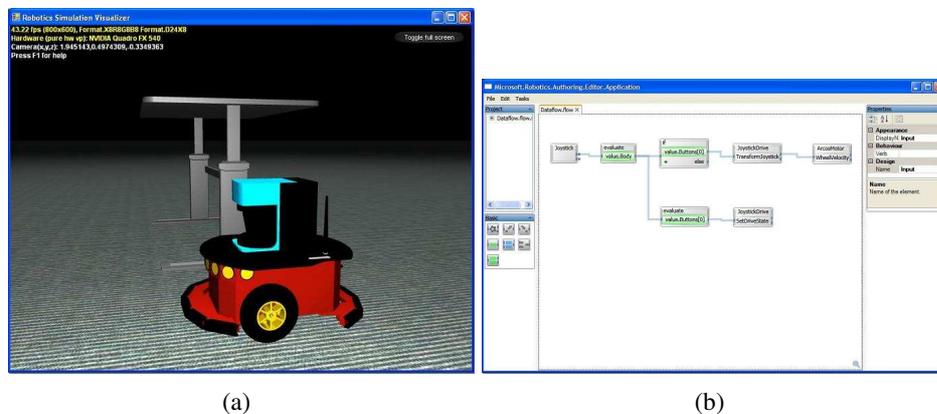


Figura 2.15: *Microsoft Robotics Studio*: (a) Exemplo de uma simulação; (b) Exemplo da *Visual Programming Language*.

O *Webots* (CYBERBOTICS, 2007) (Figura 2.16) é um simulador de robôs móveis comercializado pela *Cyberbotics* que permite, além de outros modelos, a simulação de robôs de duas rodas com cinemática diferencial, tem como base a biblioteca de simulação física ODE. Seu público alvo são pesquisadores e professores da área de agentes autônomos, visão computacional e Inteligência Artificial. O *Webots* inclui modelos prontos de alguns robôs comerciais e modela sensores com a capacidade de detecção de obstáculos, visão e manipuladores simples. O usuário pode programar cada robô utilizando *C/C++*. O ambiente virtual é tridimensional, porém o terreno é um chão plano. Neste tipo de ambiente usualmente são inseridos obstáculos no formato de figuras geométricas simples (*e.g. caixas*). É um dos simuladores mais avançados e utilizados na área de robótica móvel, porém é um produto comercial.

Outros ambientes de simulação bastante utilizados em pesquisas na área de robótica são o *Khepera Simulator* (MICHEL, 2007), o *Mission Simulation Facility* (CSD/NASA, 2007), o *JUICE* (WADDOUPS, 2007) (Figura 2.17(a)) e o *Simulator BOB* (STELLMANN, 2007) (Figura 2.17(b)). As ferramentas *JUICE* e *Simulator BOB* podem ser consideradas uma camada sobre

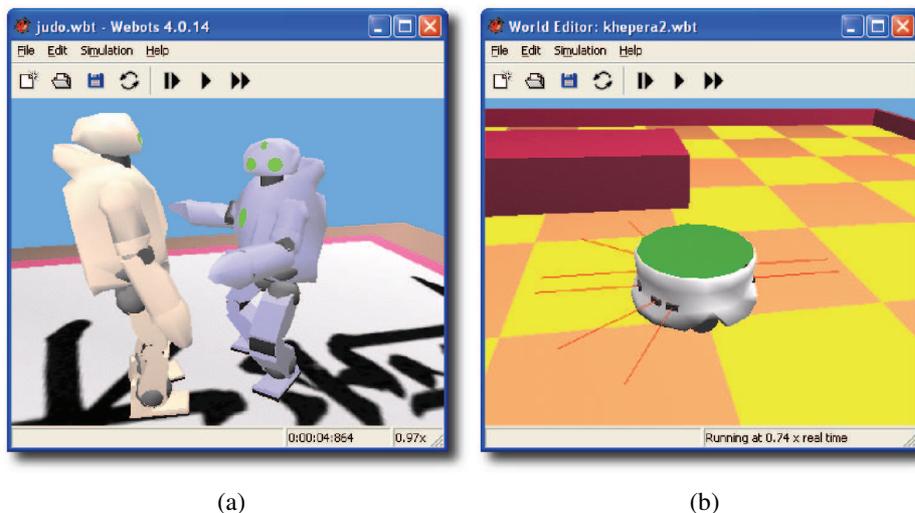


Figura 2.16: Exemplo de telas do *Webots*: (a) Sumo robótico; (b) Robô holonômico.

a biblioteca de simulação física ODE, nestas ferramentas, a leitura da morfologia de um robô é feita através de um arquivo do tipo *Extensible Markup Language* (XML). O *Mission Simulation Facility* utiliza uma biblioteca de simulação física própria, porém baseada na ODE.

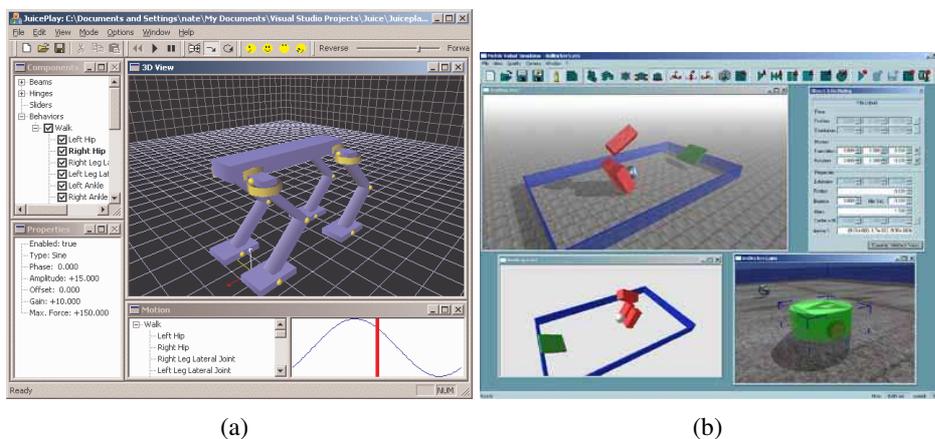


Figura 2.17: Exemplo de telas: (a) *JUICE*; (b) *Simulator BOB*.

2.1.6.2 Bibliotecas

Como optamos por desenvolver nosso próprio ambiente, todas as bibliotecas de programação utilizadas no desenvolvimento do ambiente e dos agentes robóticos são software livre e em linguagem *C/C++*. Uma pequena descrição de cada uma das bibliotecas utilizadas é fornecida a seguir.

- **Open Dynamics Engine:** A *Open Dynamics Engine* (ODE) (SMITH, 2006; ODE, 2007) é uma biblioteca desenvolvida para a simulação física de estruturas de corpos rígidos ar-

articulados. Uma estrutura articulada é criada quando corpos rígidos de vários tipos são conectados por algum tipo de articulação (junta), como por exemplo um veículo terrestre que tem a conexão de rodas em um chassi. A ODE foi projetada para ser utilizada de modo interativo em simulações de tempo real e é especialmente indicada para a simulação de objetos móveis em ambientes dinâmicos. A ODE não tem como objetivo realizar simulação de outras dinâmicas além da de corpos rígidos, como de partículas, roupas, ondas, fluídos, corpos flexíveis ou fraturas (ODE, 2007). O integrador de primeira ordem utilizado pela ODE é altamente estável, assim, erros da simulação nunca devem crescer a ponto de deixar o sistema fora de controle. Segundo o criador da biblioteca (SMITH, 2006) a ODE é bastante confiável em termos de simulação física, porém enfatiza estabilidade e velocidade sobre a acuracidade. A distribuição de massa é arbitrária aos corpos rígidos. Possui um modelo de fricção/contato baseado no *Dantzig LCP solver* (ANDREW; BARAFF, 1993). A ODE possui contatos rígidos, assim, quando dois corpos colidem não há penetração. O sistema de detecção de colisão é nativo e suporta as seguintes primitivas: esfera (*sphere*), caixa (*box*), cilindro (*cylinder*), plano (*plane*) e raio (*ray*). O código típico de uma simulação envolve:

1. Criação de um mundo dinâmico;
2. Criação dos corpos no mundo dinâmico;
3. Ajuste da posição e orientação dos corpos;
4. Criação de articulações no mundo dinâmico;
5. Conexão das articulações nos corpos;
6. Ajuste dos parâmetros das articulações;
7. Criação de um mundo colisivo e da geometria dos objetos colisivos;
8. Repetir, para cada passo de simulação:
 - (a) Aplicar forças nos corpos como necessário;
 - (b) Ajustar os parâmetros das articulações;
 - (c) Chamar rotina de teste de colisão;
9. Destruir os mundos dinâmico e colisivo.

Do ponto de vista físico, um robô pode ser considerado simplesmente como um conjunto de corpos rígidos conectados. Cada um destes corpos pode interagir com os demais, assim como um motor faz com que um veículo se movimente. Além disso a atuação da gravidade e da inércia devem estar presentes. Na ODE, estas propriedades são tratadas

em dois conceitos: corpos rígidos e articulações. Com relação aos **corpos rígidos**, suas propriedades, do ponto de vista da simulação, que mudam com o passar do tempo, são: (i) vetor de posição (x, y, z) do ponto de referência do corpo; (ii) velocidade linear do ponto de referência do corpo, representada como um vetor (v_x, v_y, v_z) ; (iii) orientação do corpo, representada como um *quaternion* (q_s, q_x, q_y, q_z) ou uma matriz de rotação; (iv) velocidade angular, a qual descreve como a orientação muda com o passar do tempo; (v) massa do corpo; (vi) posição do centro de massa; e (vii) matriz de inércia, a qual descreve como a massa do corpo está distribuída a redor do centro de massa. As **articulações** (Figura 2.18) são relacionamentos entre dois corpos de modo que possam existir posições e orientações que sejam relativas a ambos os corpos. Este relacionamento realiza certos tipos de restrições/obrigações de movimento. A cada passo de simulação, o integrador aplica num corpo uma força de atuação, mas move este corpo preservando as restrições das articulações.

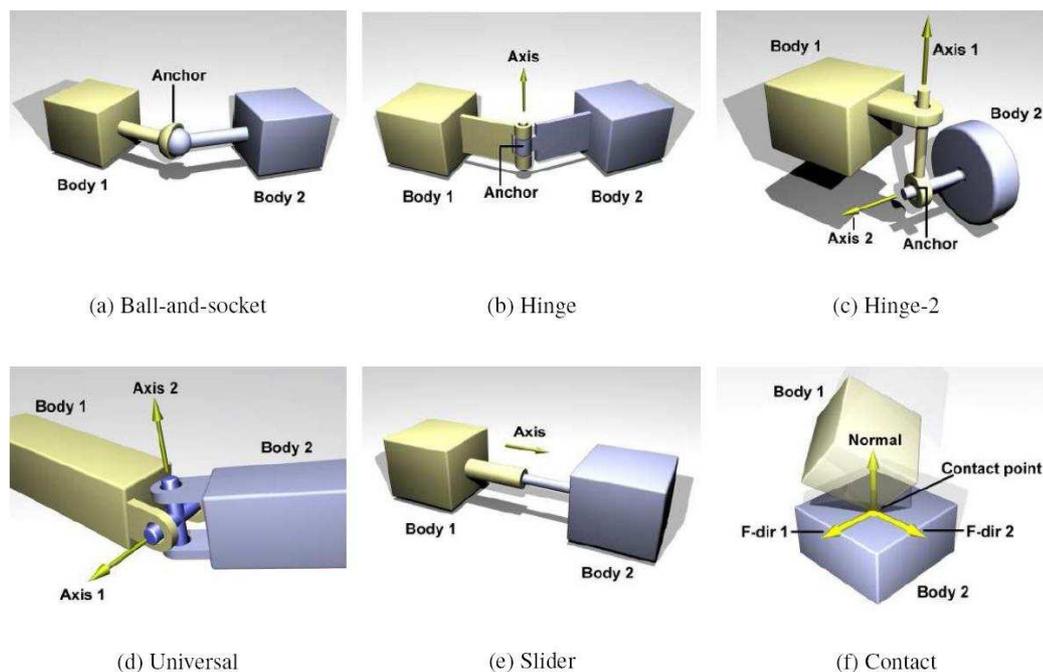


Figura 2.18: Articulações disponíveis na ODE (SMITH, 2006).

- **DrawStuff:** A ODE (ODE, 2007) é uma biblioteca de simulação física de corpos rígidos articulados completamente independente de visualizador. Iniciar a criação dos corpos e das simulações pode ser uma tarefa bastante árdua caso não tenhamos uma forma simples e fácil de visualizar os objetos. Por este motivo, a biblioteca *DrawStuff* (ODE, 2007) é disponibilizada em conjunto com a ODE. Basicamente, o *DrawStuff* é um ambiente de visualização de objetos 3D que tem o propósito de permitir a demonstração visual da

ODE sendo uma biblioteca bastante simples e rápida para utilização. Não encontramos referências que demonstrem que o *Drawstuff* permita integração com o *Demeter*, nem a leitura de modelos complexos (como árvores ou fogo).

- **Demeter:** O *Demeter Terrain Engine* (DEMETER, 2007) é uma biblioteca desenvolvida para renderizar terrenos 3D. Desenvolvida para ter rápida performance e boa qualidade visual, pode renderizar grandes terrenos em tempo-real sem necessidade de hardware especial. É escrita em componentes para ser facilmente inserida em qualquer tipo de aplicação, como jogos, simuladores, sistemas para engenharia civil, exploração mineral ou qualquer outro tipo de aplicação que necessite de terrenos 3D. A biblioteca *Demeter* depende de algumas outras bibliotecas, como a *Simple DirectMedia Layer* (SDL) (SDL, 2007) para realizar o tratamento de texturas e imagens e a *Geospatial Data Abstraction Library* (GDAL) (GDAL, 2007) para carregar arquivos de elevação.
- **Simple DirectMedia Layer:** A *Simple DirectMedia Layer* (SDL) (SDL, 2007) foi utilizada no desenvolvimento do protótipo 2D e, na versão 3D só é utilizada pelo *Demeter*. É uma biblioteca desenvolvida para permitir acesso de baixo nível a funções de trabalho multimídia, como áudio, teclado, *mouse*, *joystick*, e *framebuffer* de vídeo.
- **Open Scene Graph:** A *Open Scene Graph* (OSG) é uma biblioteca para o desenvolvimento de aplicações gráficas 3D de alta performance, como simuladores, jogos, aplicações de realidade virtual e visualização científica (OSG, 2007). Baseada no conceito de grafos de cena, provê ao desenvolvedor um ambiente orientado a objeto sobre a OpenGL (OPENGL, 2007), assim, libera este da necessidade de implementação e otimização de chamadas gráficas de baixo nível. Além disso, provê muitas funções adicionais para desenvolvimento rápido de aplicações gráficas. Um grafo de cena permite a representação de objetos em uma cena com a estrutura de um grafo, a qual permite a criação de grupos que podem compartilhar algumas propriedades, assim, podemos definir uma propriedade comum em um nível hierárquico mais alto e todos os objetos inferiores receberão esta propriedade (BARROS; GONZAGA; RAPOSO, 2007). A OSG faz o controle automático de níveis de detalhe de imagens, otimizando assim o uso do hardware ao desenhar uma cena e permite integração com a biblioteca de terrenos *Demeter* e com a biblioteca física ODE sem maiores complicações. É uma biblioteca bastante utilizada junto a Unisinos no CROMOS Lab e em trabalhos relacionados aos projetos da RBV.

2.2 Agentes Inteligentes

2.2.1 Introdução

Agentes inteligentes constituem um paradigma de desenvolvimento de sistemas que tem sido aplicado com êxito na solução de uma enorme diversidade de problemas (FONSECA, 2000; BORDINI; VIEIRA; MOREIRA, 2001; GARCIA; SICHMAN, 2003; REIS, 2003; SANTOS, 2004). Muitos dos conceitos do domínio da área de agentes inteligentes vão de encontro direto com as necessidades dos robôs móveis autônomos (HEINEN, 2000; DUDEK; JENKIN, 2000; BEKEY, 2005).

Esta Seção tem como objetivo situar o leitor nos conceitos e características do domínio de agentes que são importantes para a criação do sistema multi-robótico proposto neste trabalho.

2.2.2 Agentes

Uma das primeiras e mais simples definições é de que um agente é um sistema computacional que habita um dado ambiente, sente e age autonomamente nesse ambiente; ao fazê-lo realiza um conjunto de objetivos ou tarefas para o qual foi projetado (MAES, 1996). Segundo (PFEIFER; SCHEIER, 1994), um agente de software pode ser visto como um sistema dinâmico, onde a percepção e a ação constituem processos simultâneos e inseparáveis. Em termos gerais, ambiente é onde um agente ou um conjunto de agentes está inserido, pode ser físico (como ambientes onde estão inseridos robôs), de software ou de realidade virtual (onde se faz simulação do ambiente físico) (GARCIA; SICHMAN, 2003). A Figura 2.19 mostra o esquema típico de um agente (REIS, 2003).

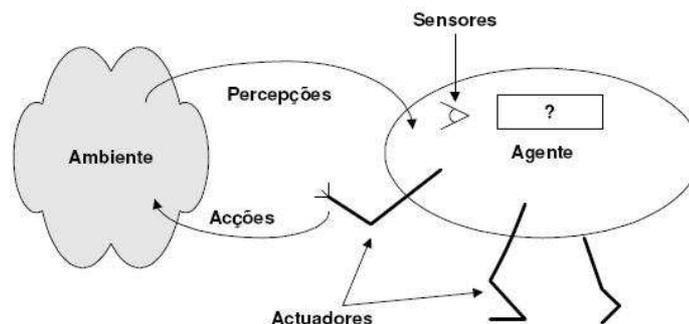


Figura 2.19: Esquema típico de um agente (REIS, 2003).

De acordo com (PFEIFER; SCHEIER, 1999) o projeto de um agente de software envolve a definição de três componentes que são rigorosamente interconectados e mutuamente interdependentes:

1. Definição do ambiente para atuação;
2. Estabelecimento de comportamentos e tarefas a serem realizadas;
3. Determinação da morfologia do agente.

A natureza dos ambientes em que um agente pode habitar varia significativamente, assim, nenhum agente pode adaptar-se fisicamente e cognitivamente para lidar com todas as variações possíveis. Por essa razão, o ambiente deve ser fixado anteriormente ao projeto do agente (HEINEN, 2007).

A concretização do projeto de um agente robótico deve priorizar quatro aspectos principais (PFEIFER; IIDA; BONGARD, 2005):

- **Autonomia:** Deve ser capaz de operar com o mínimo possível de intervenção, supervisão e instrução humana.
- **Auto-suficiência:** Deve ser capaz de operar por tempo prolongado, ou, pelo menos, suficiente para realizar a tarefa e buscar recarga de energia.
- **Localização:** Deve ser capaz de adquirir informação sobre o ambiente somente através de seus próprios sensores.
- **Corporificação:** Deve ser desenvolvido como um sistema físico ou computacional, porém a corporificação não se dá necessariamente pela materialização, como a apresentada em animais ou robôs físicos, mas por uma relação dinâmica com o ambiente, ou seja, a corporificação pode ser realizada em ambientes de simulação computacionais, desde que estes ambientes sejam realísticos do ponto de vista físico (PFEIFER; IIDA; BONGARD, 2005).

Ainda, de acordo com (PFEIFER; SCHEIER, 1999), quando trabalhamos com agentes robóticos, devemos respeitar os seguintes princípios:

- **Princípio da redundância e do equilíbrio ecológico:** Não basta que um agente de software possua variados mecanismos de obtenção de informação, é interessante incorporar redundância nos dispositivos sensoriais, assim, os sensores podem estar posicionados no agente de tal forma que exista sobreposição espacial nas informações adquiridas. A redundância deve promover correlações que ajudam o agente a reduzir drasticamente a incerteza do ambiente e a prever eventos (PFEIFER; SCHEIER, 1999). Com relação ao equilíbrio, um sistema de controle extremamente complexo é desnecessário se o agente

ou o ambiente são extremamente simples, assim como um agente com um sistema de controle muito simples pode ter dificuldades em atuar em um ambiente complexo (PFEIFER; SCHEIER, 1999; HEINEN, 2007).

- **Princípio dos processos paralelos e fracamente acoplados:** Agentes de software devem possuir diversos comportamentos a fim de cumprir determinadas tarefas. Alguns comportamentos são compatíveis, mas outros são mutuamente exclusivos. Por causa disto, uma decisão deve ser tomada para selecionar, a cada momento, aquelas ações que são coerentes com o contexto atual do agente e do ambiente (BROOKS, 1986). Um mecanismo que vem despontando como uma nova opção para esta problemática é assumir um número elevado de processos heterogêneos, paralelos e fracamente acoplados que são conectados ao aparato sensório-motor do agente (PFEIFER; SCHEIER, 1999). Estes processos não necessitam de um supervisor, o controle é descentralizado e distribuído. A arquitetura de controle pode ser construída de forma gradual, com a adição de novos processos, assim como acontece na evolução biológica. Além disso, a união de todos os processos deve possibilitar que emerjam novos comportamentos, possivelmente não previstos no projeto.
- **Princípio da coordenação sensório-motora:** O estudo da integração sensório-motora tem sido um tópico de pesquisa bastante ativo na área das ciências cognitivas (PFEIFER; SCHEIER, 1999). Existe consenso entre os pesquisadores da área que, em organismos biológicos, as informações oriundas de múltiplos sentidos (*e.g.* visão, audição, tato) são integradas e diretamente mapeadas sobre um conjunto apropriado de comandos motores (*e.g.* músculos, glândulas). Este processo sensório-motor, quando aplicado a agentes de software, conduz a inúmeros benefícios e simplificações de projeto (PFEIFER; SCHEIER, 1994, 1997; NOLFI, 2002). Dados sensoriais não são apenas adquiridos mas, sobretudo, gerados e correlacionados, onde a correlação reduz a alta dimensionalidade presente nos dados obtidos do ambiente que, por sua vez, capacita o agente a fazer associações entre diferentes modalidades sensitivas (BOEKHORST; LUNGARELLA; PFEIFER, 2003).
- **Princípio da aprendizagem:** Os seres vivos têm a capacidade de aprender. O aprendizado, do ponto de vista da neurociência, ocorre através de mudanças estruturais nas conexões sinápticas entre os neurônios. Estas alterações podem ser realizadas de variadas formas, o que acaba por gerar inúmeros tipos de aprendizagem. Não existe ainda uma teoria unificada ou uma abordagem única e comum aceita na comunidade científica para a aprendizagem robótica. Dessa forma, não há consenso de qual o melhor paradigma de aprendizagem (PFEIFER; IIDA; BONGARD, 2005). Apesar disto, pode-se enumerar uma série de características desejáveis e necessárias em um algoritmo de aprendizagem para

agentes de software autônomos: robustez e tolerância a ruídos; convergência rápida; tratabilidade computacional; adaptatividade a eventuais mudanças ambientais; dependência de informações que possam ser extraídas apenas dos sensores do agente, não aquelas fornecidas por um projetista ou observador externo (PFEIFER; SCHEIER, 1999).

Com base nestes princípios, várias pesquisas na área de robótica móvel autônoma vem sendo desenvolvidas pelo Grupo de Pesquisas em Veículos Autônomos (GPVA) da Unisinos (OSÓRIO, 2007).

2.2.3 Sistemas Multi-Agentes

Sistemas multi-agentes são sistemas constituídos de múltiplos agentes que interagem ou trabalham em conjunto de forma a realizar um determinado conjunto de tarefas ou objetivos. Um sistema multi-agente pode ser visto como uma rede, fracamente acoplada, de solucionadores de problemas que trabalham em conjunto para resolver problemas que vão além da sua capacidade individual. Estes solucionadores de problemas são essencialmente autônomos e, muitas vezes, heterogêneos em sua natureza (SILVEIRA, 2006).

Um sistema multi-agente deve possuir alguma forma de controle. A utilização de ambientes dinâmicos, de robôs com mau-funcionamento, de múltiplas regras e restrições em comportamentos individuais adicionam complexidade ao problema de controle (OSAGIE, 2006). Nos trabalhos de (STONE; VELOSO, 2000; MONDADA et al., 2005; OSAGIE, 2006) sistemas multi-agentes são definidos em duas arquiteturas de controle: sistemas distribuídos ou sistemas centralizados.

Em um sistema multi-agente centralizado, uma única unidade detém todo o poder de tomada de decisão para controle e definição de objetivos dos outros agentes do sistema. É comum que neste tipo de sistema exista sobrecarga da unidade de controle (OSAGIE, 2006). A abordagem centralizada ainda tem relação com o problema de falha em pontos únicos do sistema, que podem inviabilizar toda a operação, porém, traz a vantagem de poder produzir planos de ação globais ótimos que podem ser teoricamente comprovados como corretos (OSAGIE, 2006).

Em um sistema multi-agente distribuído, cada robô é responsável pela coordenação de suas próprias ações. Em um sistema completamente distribuído, a busca da solução do problema surge da interação entre os agentes. Estas soluções são tipicamente difíceis ou mesmo impossíveis de terem sua eficiência comprovada analiticamente (OSAGIE, 2006).

2.2.4 Cooperação

Cooperação pode ser entendida como a capacidade que os agentes têm de trabalhar em conjunto de forma a concluírem tarefas de interesse comum (REIS, 2003). Nwana (NWANA, 1996) acredita que a cooperação entre agentes é fundamental, sendo a razão principal para a existência de um ambiente multi-agente. Para que exista cooperação, deve existir alguma forma de interação entre os agentes, ou com humanos, através de alguma forma de comunicação (WOOLDRIDGE; JENNINGS, 1995).

A cooperação não depende obrigatoriamente de comunicação entre todos os agentes, visto que, como no mundo real, um coordenador (unidade de controle) pode delegar tarefas (enviar mensagens) para um grupo de pessoas (agentes) que não tenham comunicação direta entre si. Neste caso, as pessoas podem trabalhar em prol do mesmo objetivo sem ter conhecimento do que as outras pessoas estão fazendo.

2.2.5 Comunicação

A morfologia de um robô deve, ao incluir a capacidade de comunicação, possuir um módulo de controle de comunicações, assim, deve possuir componentes de percepção (recepção de mensagens) e de ação (envio de mensagens). O módulo de comunicações está diretamente ligado ao módulo de controle do agente (REIS, 2003).

No nível referente aos mecanismos de comunicação, é possível identificar quatro diferentes modelos de trocas de mensagens entre agentes (WOOLDRIDGE, 2000):

- **Troca direta de mensagens:** ou comunicação direta, envolve o estabelecimento de uma ligação direta entre os agentes, através de um conjunto de protocolos que permite o estabelecimento de comunicação ponto a ponto. Este tipo de comunicação implica que cada um dos agentes envolvidos deve ter conhecimento da existência dos seus parceiros e da forma de endereçar mensagens a eles (SILVEIRA, 2006).
- **Sistemas federados:** Neste modelo são introduzidos agentes facilitadores (GENESERETH; KETCHPEL, 1994) que são agentes especiais que provem serviços de comunicação. Uma estrutura hierárquica de agentes é estabelecida e toda troca de mensagens se dá através destes agentes facilitadores como mostra a Figura 2.20.
- **Comunicação por difusão de mensagens:** ou *broadcasting*, utilizado quando for necessária comunicação entre todos os agentes do sistema; ou quando o remetente da mensagem não sabe quem é o destinatário ou qual é seu endereço (SILVEIRA, 2006).

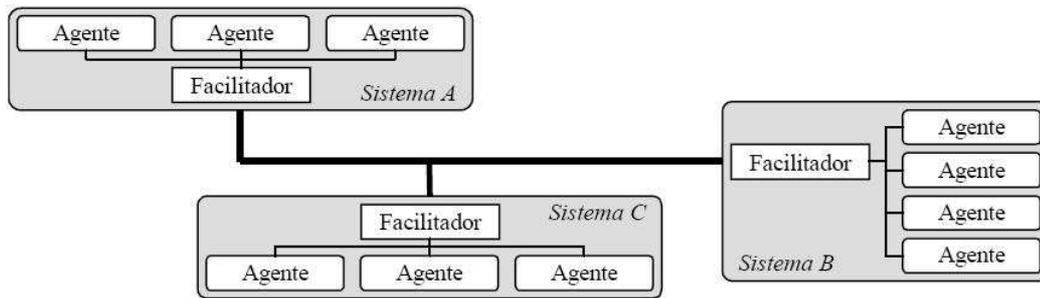


Figura 2.20: Exemplo de um sistema federado (FONSECA, 2000).

- **Quadro-negro:** No modelo de comunicação através de quadro-negro (*blackboard*) a comunicação se dá através de escrita e leitura em uma estrutura de dados comum (ERMAN et al., 1980; HAYES-ROTH, 1995). Neste tipo de modelo não existe qualquer tipo de comunicação direta entre agentes, sendo esta efetuada sempre através de uma estrutura de dados comum (REIS, 2003), como mostra a Figura 2.21.

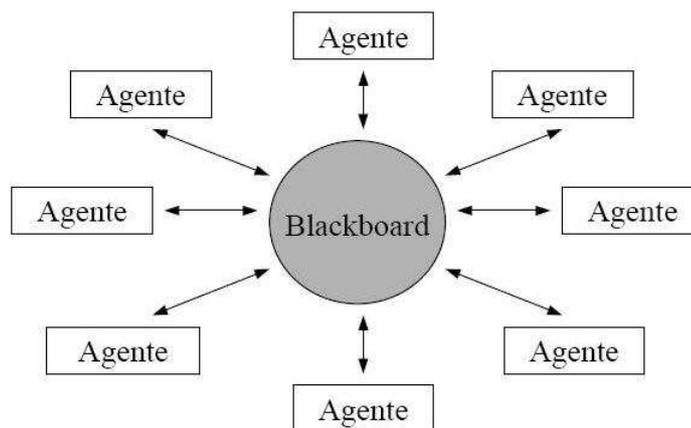


Figura 2.21: Exemplo de um sistema baseado em quadro-negro (FONSECA, 2000).

2.2.6 Considerações finais

A tarefa de monitoração e combate a incêndios apresenta um problema onde a atuação de um sistema multi-agente pode trazer grandes benefícios, tanto por poder mapear o funcionamento de uma operação real como também, no caso deste trabalho, por apresentar um modelo com veículos autônomos, o que reduz os riscos humanos na operação. Neste trabalho, a cooperação é resultado de uma evolução usando Algoritmos Genéticos, assim, não teremos comunicação entre agentes para este fim. A comunicação simulada neste trabalho é detalhada na Seção 4.1. Com relação aos quatro aspectos principais que concretizam o projeto de um agente robótico: desenvolvemos os agentes de combate a incêndio com capacidade de autono-

mia e localização; por serem fisicamente simulados apresentam também a corporificação; no entanto, não consideramos carga de bateria ou combustível, como se fossem auto-suficientes a ponto de cumprir a missão que lhes foi atribuída sem necessidade de realizar um reabastecimento. No desenvolvimento do sistema multi-agente implementamos a definição da estratégia de forma centralizada e a atuação no combate a incêndio de forma distribuída.

2.3 Aprendizado de Máquina

2.3.1 Introdução

Aprendizado de Máquina (*Machine Learning* - ML) é uma área da Inteligência Artificial que tem como objetivo desenvolver técnicas computacionais de aprendizado e de aquisição de conhecimentos (REZENDE, 2003). Essas técnicas devem exibir um comportamento inteligente e realizar tarefas complexas com um nível de competência equivalente ou superior ao de um especialista humano (NIKOLOPOULOS, 1997).

As técnicas de ML mais importantes citadas pelas bibliografias são: (i) aprendizado por analogia (sistemas baseados em casos - CBR) (REZENDE, 2003); (ii) aprendizado por indução: árvores de decisão (SHIBA et al., 2005; WILKING; ROFER, 2004) e *Inductive Logic Programming*; (iii) aprendizado por evolução: algoritmos genéticos (NOLFI; FLOREANO, 2000; HEINEN; OSÓRIO, 2007; REZENDE, 2003; SILVEIRA, 2006); (iv) aprendizado conexionista: redes neurais artificiais (BRAGA; CARVALHO; LUDERMIR, 2000; SOUZA et al., 2007; HEINEN; OSÓRIO, 2007), redes de função de base radial e *Support Vector Machines*; e, (v) aprendizado por reforço (REZENDE, 2003).

Esta Seção tem por objetivo situar o leitor nos fundamentos das técnicas de IA utilizadas para a construção do modelo proposto, que será focado no uso de RNAs e GAs. A área de Aprendizado de Máquina possui diversas técnicas. Cada uma delas possui diversas vantagens e desvantagens, bem como pontos fracos e limitações. Não existe uma técnica de Aprendizado de Máquina ideal que funcione de forma satisfatória para todas as classes de problemas existentes (MITCHELL, 1996). Algumas técnicas podem ser combinadas para formar um sistema híbrido. No caso deste sistema aplicamos técnicas de IA puras, específicas para o nível de especialização dos agentes.

A vantagem de se utilizar GA para a otimização das posições é que eles não necessitam de informações locais para a correção dos erros, ou seja, eles não necessitam de uma base com os dados de treinamento. No caso do combate a incêndios, estas informações seriam muito

difíceis ou mesmo impossíveis de serem obtidas. O aprendizado ocorre através da interação das possíveis soluções com o ambiente, o que está de acordo com princípios descritos na Seção 2.2.2.

O uso de RNA no controle de navegação é justificado por serem as técnicas mais robustas em relação a situações novas e inesperadas, por possuírem alto grau de generalização e por fornecerem uma arquitetura em conformidade com os princípios descritos na Seção 2.2.2. Além disso, como o desenvolvimento de regras de controle é uma operação bastante complexa, podemos criar uma base de treino através da obtenção de dados de um veículo operado manualmente, para posterior treino e teste da RNA.

2.3.2 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são sistemas paralelos distribuídos, compostos por unidades de processamento simples que calculam determinadas funções matemáticas, normalmente não-lineares (BRAGA; CARVALHO; LUDERMIR, 2000). Seus atributos básicos podem ser divididos em arquitetura e neurodinâmica. A arquitetura determina a estrutura da rede, ou seja, o número de neurônios e sua interconectividade e a neurodinâmica, por sua vez, define as propriedades funcionais da rede, ou seja, como ela aprende, recupera, associa e compara novas informações com o conhecimento já armazenado (KARTALOPOULOS, 1996). Matematicamente, RNAs são aproximadores universais, que realizam mapeamentos em espaços de funções multivariáveis (HORNIK; STINCHCOMBE; WHITE, 1989)

A capacidade de aprender e generalizar¹ das RNAs é um dos seus maiores atrativos, o que lhe proporciona um poder muito além do simples mapeamento de entradas e saídas.

O processamento da informação em uma RNA é feito por meio de estruturas neurais artificiais (REZENDE, 2003), sendo que esta estrutura, bem como o próprio neurônio artificial são uma analogia biológica ao funcionamento do cérebro. O neurônio artificial proposto por (MCCULLOCH; PITTS, 1943) (Figura 2.22) pode ser descrito como um modelo com n terminais de entrada $x_1, x_2, x_3, \dots, x_n$ que, para cada entrada, possui um peso w_i correspondente. A soma das entradas x_i ponderadas pelos pesos correspondentes w_i produzem a chamada *saída linear* u . A saída y do neurônio é obtida pela aplicação de uma função $f(\cdot)$ à saída linear u . A função $f(\cdot)$ é chamada *função de ativação* e pode assumir diferentes formas (e.g. linear, semi-linear ou *sigmoid*).

¹Generalizar pode ser considerado como a produção de saídas aceitáveis para entradas não apresentadas na fase de aprendizado.

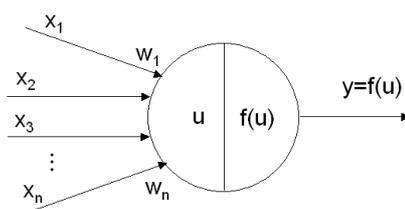


Figura 2.22: Neurônio de McCulloch e Pitts (MCP), retirado de (REZENDE, 2003).

A função de ativação *sigmoid* (Figura 2.23) faz com que a saída de um neurônio seja normalizada entre 0 e 1. Devido a sua curvatura ela é usada no processo de aprendizado para gerar uma certa estabilidade no sistema, evitando que os pesos variem muito, depois que o valor da saída já se encontra próximo a um dos extremos (RUMELHART; HINTON; WILLIAMS, 1986).

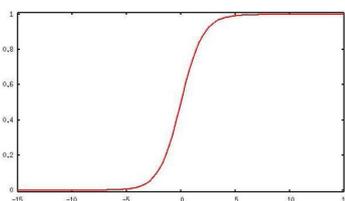


Figura 2.23: Exemplo de uma função *sigmoid*.

A partir do trabalho de (MCCULLOCH; PITTS, 1943) que enfocou a modelagem de um neurônio, (ROSEMBLATT, 1958) propôs o modelo *Perceptron*, introduzindo o conceito de aprendizado em RNA. O modelo *Perceptron* era composto por uma estrutura de rede tendo como unidades básicas nodos MCP e de uma regra de aprendizado. O aprendizado no modelo *Perceptron* sempre converge caso o problema em questão seja linearmente separável (ROSEMBLATT, 1962). Para a solução de problemas não linearmente separáveis seria necessária mais de uma camada de neurônios (BRAGA; CARVALHO; LUDERMIR, 2000). A solução de aprendizado para redes com *Perceptrons* combinados em diversas camadas, formando a chamada Rede Neural do Tipo *Multi Layer Perceptron* (MLP) foi proposta por (RUMELHART; MCCLELLAND, 1986) com o algoritmo de aprendizado *Backpropagation*.

O algoritmo *Backpropagation* é um algoritmo supervisionado. A base de treinamento é um conjunto de dados que deve apresentar, para cada entrada, a saída prevista do sistema. Este tipo de aprendizado ocorre em várias épocas; cada época representa a apresentação do conjunto inteiro de dados a rede neural para o ajuste dos pesos (OSÓRIO; BITTENCOURT, 2000). O treinamento no *Backpropagation* ocorre em duas fases (BRAGA; CARVALHO; LUDERMIR, 2000): a fase *forward* é utilizada para definir a saída da rede para um dado padrão de entrada; a fase *backward* utiliza a saída desejada e a saída fornecida pela rede para atualizar o peso das conexões. O *Backpropagation* é baseado na Regra Delta Generalizada (WIDROW; HOFF, 1960) e

os ajustes dos pesos são realizados pelo método da Descida do Gradiente (BRAGA; CARVALHO; LUDERMIR, 2000).

Por utilizar o método da Descida do Gradiente, não há garantia de que o aprendizado não tenha ficado preso em mínimos locais (Figura 2.24), assim, o treinamento de uma rede neural deve envolver várias rodadas de simulação, inicializando os pesos de forma aleatória. Outra questão importante é o grau de generalização. Quando uma Rede Neural é treinada, os pesos sinápticos vão se ajustando para que sejam dadas respostas satisfatórias à base de treinamento apresentada. Se o aprendizado for feito por um número muito grande de épocas pode ocorrer o chamado *overfitting*, onde a Rede Neural começa a decorar a saída, perdendo sua capacidade de generalização (OSÓRIO; VIEIRA, 1999). A solução deste problema usualmente envolve o uso de uma base de validação usada em paralelo a base de treinamento, como mostra a Figura 2.25.

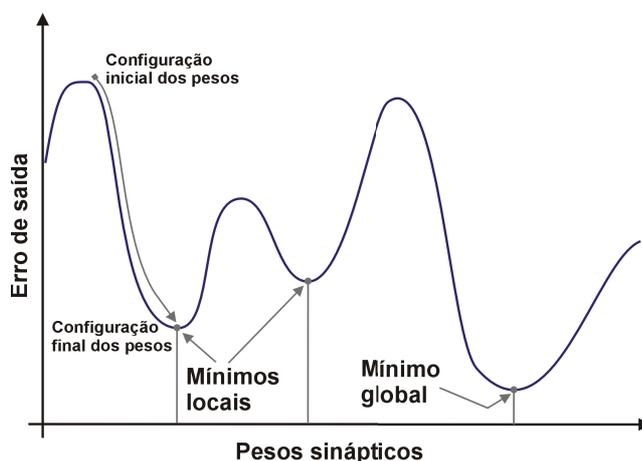


Figura 2.24: Descida do gradiente de uma superfície de erro.

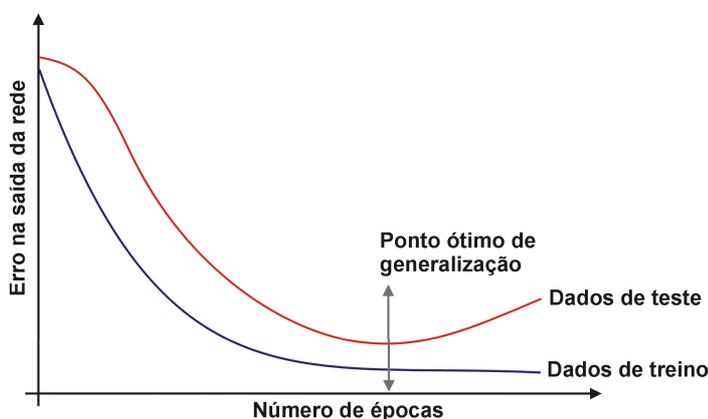


Figura 2.25: Curvas de erro em aprendizado e validação.

O algoritmo *Backpropagation* padrão é muito lento e seu desempenho piora sensivelmente para problemas maiores e mais complexos; mesmo para problemas relativamente sim-

ples, geralmente requer que todos os padrões de treinamento sejam apresentados centenas ou milhares de vezes, o que limita sua utilização prática (BRAGA; CARVALHO; LUDERMIR, 2000). Os algoritmos atualmente empregados no treinamento são variações do *Backpropagation*, que tem em geral o foco de aumentar a velocidade do treinamento ou melhorar a classificação de padrões. As variações mais conhecidas são: *Backpropagation* com *Momentum* (RUMELHART; MCCLELLAND, 1986); *Quickprop* (FAHLMAN, 1988); e *Resilient Backpropagation* - (RProp) (RIEDMILLER; BRAUN, 1994).

2.3.2.1 SNNS (SNNS, 2006)

O *Stuttgart Neural Network Simulator* (SNNS, 2006) é um simulador de Redes Neurais Artificiais criado no *Institute for Parallel and Distributed High Performance Systems* (IPVR) da Universidade de Stuttgart. Proporciona um ambiente eficiente e flexível para auxiliar a criação, o treinamento e a manutenção de Redes Neurais. O SNNS possui um grande número de algoritmos de aprendizado, como *Backpropagation*, *Quickprop*, *RProp*, *Backpercolation*, *Counterpropagation*, *Generalized radial basis functions*, entre outros. O *kernel* do sistema é desenvolvido em C e sua utilização pode ser completamente feita através de linha de comando, porém também possui uma interface desenvolvida em JAVA (JavaNNS) (Figura 2.26). Esta interface gráfica possibilita fácil criação de diversas topologias de Redes Neurais Artificiais, além de permitir acompanhar a evolução de taxas de erro e aprendizado da RNA através de gráficos. Um aplicativo do pacote SNNS, o SNNS2C, permite a conversão de uma RNA em código C, que pode então ser facilmente inserido em uma outra aplicação.

Usamos o SNNS nesta dissertação como a ferramenta de desenvolvimento de topologia, treino e teste das RNAs, bem como para a conversão da melhor RNA encontrada em uma rotina em C, para ser inserida na parte de controle inteligente de navegação com desvio de obstáculo do nosso simulador. Todas estas operações estão detalhadas nos Capítulos 4 e 5.

2.3.3 Algoritmos Genéticos

Algoritmos Genéticos (*Genetic Algorithms* - GA) (HOLLAND, 1975; MITCHELL, 1996) são técnicas de otimização² global que empregam estratégia de busca paralela e estruturada, embora aleatória, direcionada a busca de pontos de aptidão (CARVALHO; BRAGA; LUDERMIR, 2003). Permitem a realização de busca multi-critério em um espaço multi-dimensional. Devido a ser uma técnica estocástica³, é classificado como não supervisionado, visto que não necessita

²Otimização é a busca da melhor solução para um dado problema.

³Estocásticos são padrões que surgem através de eventos aleatórios.

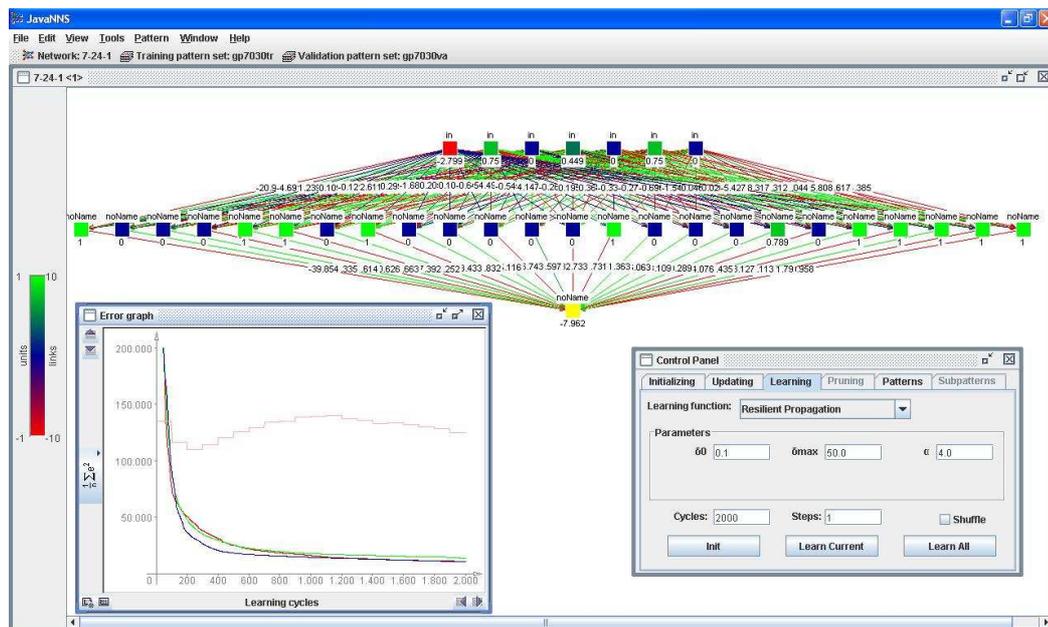


Figura 2.26: Exemplo de tela do JavaNNS treinando uma RNA.

de nenhuma base de informação de antemão.

Os Algoritmos Genéticos utilizam procedimentos iterativos que simulam o processo de evolução de uma população de possíveis soluções de um determinado problema. O processo de evolução é aleatório, porém guiado por um mecanismo de seleção baseado na aptidão de estruturas individuais. A cada iteração do algoritmo (uma geração), um novo conjunto de estruturas é criado através da troca de informações (bits ou blocos) entre estruturas bem adaptadas selecionadas da geração anterior (GOLDBERG, 1989). Novas estruturas são geradas aleatoriamente com uma dada probabilidade e incluídas na população. O resultado tende a ser um aumento da aptidão de indivíduos ao meio, podendo acarretar também em um aumento global da aptidão da população a cada nova geração. Neste caso, a população evolui a cada geração se aproximando de uma solução ótima (GOLDBERG, 1989).

Um GA é estruturado de forma que as informações referentes a um determinado sistema possam ser codificadas de maneira análoga a cromossomos biológicos, como uma seqüência de valores, onde usualmente cada fragmento da seqüência representa uma variável.

2.3.3.1 O Algoritmo Genético Típico

O primeiro passo de um Algoritmo Genético típico é a geração de uma população inicial, usualmente formada por um conjunto aleatório de cromossomos que representam possíveis soluções do problema. Durante o processo evolutivo, cada indivíduo da população é avaliado

através de uma função de aptidão (*fitness*) que, em geral, permite manter os indivíduos mais aptos e descartar os menos aptos. Nos indivíduos selecionados, é feito *crossover* e mutação, gerando descendentes para a próxima geração (LACERDA; CARVALHO, 1999; NOLFI; FLOREANO, 2000; CARVALHO; BRAGA; LUDERMIR, 2003). Este processo é repetido até que a condição de parada seja satisfeita, como mostra o Algoritmo 2.1.

Algoritmo 2.1 Algoritmo Genético Típico

Seja $S(t)$ a população de cromossomos na geração t

$t \leftarrow 0$

inicializar $S(t)$

avaliar $S(t)$

enquanto o critério de parada não for satisfeito **faça**

$t \leftarrow t+1$

 selecionar $S(t)$ a partir de $S(t-1)$

 aplicar *crossover* sobre $S(t)$

 aplicar mutação sobre $S(t)$

 avaliar $S(t)$

fim enquanto

2.3.3.2 Operadores Genéticos

Operadores genéticos são as rotinas que transformam uma população através de sucessivas gerações, possibilitam assim que o GA percorra os melhores pontos do espaço de busca até chegar a um resultado satisfatório (GOLDBERG, 1989). Os operadores genéticos são (CARVALHO; BRAGA; LUDERMIR, 2003): *crossover* e mutação. Estas operações ocorrem após a seleção dos indivíduos pela função de aptidão.

Assim, *crossover* e mutação são os mecanismos de busca dos GAs para explorar regiões desconhecidas do espaço de busca (LACERDA; CARVALHO, 1999). A aplicação do *crossover* faz com que, para cada dupla de cromossomos pais selecionados, sejam feitos um ou mais pontos de corte na seqüência, e, para cada ponto destes, seja feita a troca das seqüências, gerando cromossomos filhos. A mutação faz com que um número na seqüência genética do cromossomo seja modificado, isso melhora a diversidade da população, porém, se a taxa de mutação for muito elevada, a população nunca convergirá, ficando essencialmente aleatória (CARVALHO; BRAGA; LUDERMIR, 2003). Um exemplo de *crossover* e mutação pode ser visto na Figura 2.27.

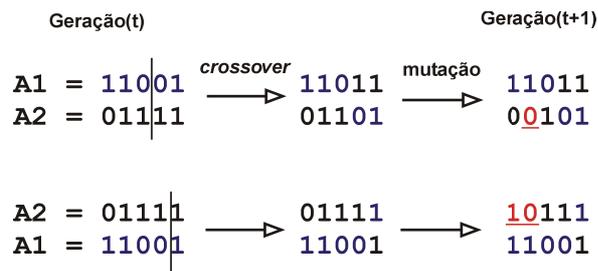


Figura 2.27: *Crossover* e mutação, adaptado de (LACERDA; CARVALHO, 1999).

2.3.3.3 Seleção

O princípio básico do funcionamento dos GAs é que um critério de seleção permita que, a partir de uma população inicial, obtenhamos uma população final mais apta a resolver o problema em questão. Para que isto ocorra, é necessário que consigamos medir, de alguma forma, o quanto cada indivíduo está apto a resolver o problema. Esta medida é feita através de uma função de aptidão (*fitness*) que, usualmente, é uma fórmula matemática. Esta fórmula matemática, em geral, pode estar relacionada ao modo como as percepções de um indivíduo geram reações afetando o próprio indivíduo e o ambiente. Um exemplo pode ser um caminhão de bombeiros autônomo que pode ou não liberar água, no caso de não liberar água, não salva o prédio, assim, caso o *fitness* considere a quantidade salva dos prédios presentes no ambiente, provavelmente, o caminhão que não libera água, não será selecionado para gerar descendentes.

Um método de seleção é aplicado considerando o *fitness* de cada indivíduo. Como os GAs são de natureza estocástica, os métodos de seleção mais utilizados também o são. Os métodos de seleção mais comuns são (NOLFI; FLOREANO, 2000; CARVALHO; BRAGA; LUDERMIR, 2003):

- **Método da roleta:** Cada indivíduo é representado em uma roleta por uma fatia proporcional a sua aptidão, como mostra a Figura 2.28(a). Uma seta é girada n vezes, selecionado em cada giro um indivíduo que será usado para gerar descendentes. Quanto maior a aptidão de um indivíduo, maior sua chance de ser selecionado.
- **Método do torneio:** Uma quantidade n de indivíduos é selecionada de uma população com m indivíduos, o indivíduo com maior *fitness* nos n selecionados é armazenado em uma população intermediária. O processo é repetido até que a população intermediária esteja completa. Normalmente n é igual a 3 (CARVALHO; BRAGA; LUDERMIR, 2003).
- **Método da amostragem universal estocástica:** É uma variação do método da roleta, onde, ao invés de 1 única seta ser girada n vezes, n setas equidistantes são utilizadas em

apenas um giro da roleta, como mostra a Figura 2.28(b).

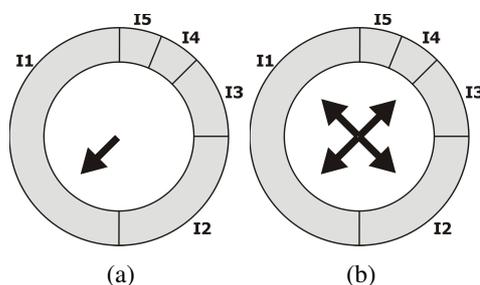


Figura 2.28: Métodos de seleção: (a) Roleta; (b) Amostragem universal estocástica. Nestes exemplos, I1 representa 50%, I2 representa 25%, I3 representa 12,5%, e I4 e I5 representam 7,25% cada.

2.3.3.4 Elitismo

O melhor indivíduo pode ser perdido de uma geração para outra devido ao corte do *crossover* e devido a mutação, assim, a melhor solução encontrada até o presente poderia ser perdida. A pesquisa realizada por (DEJONG, 1975) mostrou que a transferência do melhor indivíduo de uma geração para a outra, sem a ocorrência de *crossover* e mutação pode ser vantajosa em alguns casos. A Figura 2.29 mostra a aplicação de um GA no cálculo do seno de uma função com e sem elitismo (LACERDA; CARVALHO, 1999).

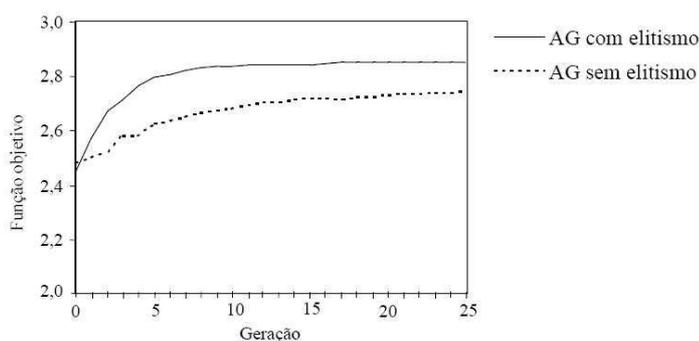


Figura 2.29: Desempenho do GA com elitismo (LACERDA; CARVALHO, 1999).

2.3.3.5 Parâmetros Genéticos

Parâmetros genéticos são fatores que influenciam fortemente o desempenho de um GA. Para obter uma boa performance é importante analisar como os parâmetros podem ser utilizados diante das necessidades do problema e dos recursos disponíveis (JONG, 1980). Estes parâmetros são (LACERDA; CARVALHO, 1999; CARVALHO; BRAGA; LUDERMIR, 2003):

- **Tamanho da população:** Uma população pequena representa uma pequena cobertura do espaço de busca, assim, o GA pode cair em mínimos locais mais facilmente que quando usando populações maiores. Populações maiores exigem mais recursos computacionais.
- **Taxa de mutação:** Uma taxa baixa previne que a busca fique estagnada em sub-regiões do espaço de busca. Assim, permite que qualquer ponto do espaço de busca seja atingido. Com uma taxa alta, a busca torna-se essencialmente aleatória (CARVALHO; BRAGA; LUDERMIR, 2003). Se a taxa de mutação for igual a zero, pode ocorrer a chamada convergência prematura, que é a estagnação da otimização em sub-regiões do espaço de busca.
- **Taxa de crossover:** A taxa de *crossover* (cruzamento) define a probabilidade de ocorrer um cruzamento, sendo que os valores mais utilizados ficam entre 0,5 e 0,9. Se não ocorrer o cruzamento entre um par de indivíduos, os dois filhos gerados serão cópias idênticas dos pais. As formas mais comuns de crossover são: cruzamento em um ponto; cruzamento em dois pontos; e cruzamento em múltiplos pontos ou uniforme (MITCHELL, 1996).
- **Intervalo de geração:** É a porcentagem da população que será substituída pela próxima geração. Valores altos podem ocasionar, dependendo do método de seleção, perda de estruturas de alta aptidão.
- **Método de seleção:** Alguns métodos de seleção possuem desempenho superior a outros, assim, devemos considerar para o nosso problema o tipo de seleção mais indicado. Goldberg (GOLDBERG, 1989) apresenta o método *Stochastic Remainder Sampling Selector* como superior ao método da roleta.
- **Critério de parada:** Tem maior relação com o uso de recursos. Os critérios de parada podem ser: por atingir um número estipulado de gerações; por não ter alteração no *fitness* médio após n gerações; por não ter alteração no *fitness* máximo após n gerações; ou por atingir um *fitness* pré-estabelecido.
- **Fitness:** O *fitness* usualmente é o resultado de uma expressão matemática. Ele pode ser maximizado ou minimizado. Deve-se ter muito cuidado com possíveis inversões dos valores parciais que formam o valor final do *fitness*.

2.3.3.6 GALib (WALL, 2007)

A GALib (WALL, 2007) é uma biblioteca de software livre desenvolvida por Matthew Wall do *Massachusetts Institute of Technology* (MIT) em C++. Esta biblioteca contém um conjunto bastante amplo de funções relacionadas a programação de Algoritmos Genéticos. A

GALib é uma das mais completas e eficientes bibliotecas de software para a simulação de Algoritmos Genéticos (HEINEN, 2007). A GALib permite facilmente a experiência com diferentes funções objetivo (*fitness*), diversas representações genéticas, diversos operadores genéticos, e diversos métodos de seleção.

Usamos a GALib nesta dissertação como a biblioteca responsável pela evolução das estratégias; a forma de uso desta biblioteca, bem como a definição dos diversos parâmetros genéticos escolhidos são detalhados nos Capítulos 4 e 5.

2.4 Incêndios em Ambientes Naturais

2.4.1 Introdução

Fogo, de um modo geral, é o termo aplicado ao fenômeno físico resultante da rápida combinação entre o oxigênio e uma substância combustível com produção de calor, luz e chamas. A reação da combustão completa da madeira, que poderia ser generalizada para todo o material combustível da floresta, envolve três elementos básicos: combustível, oxigênio e calor. Em qualquer incêndio florestal é necessário haver combustível para queimar, oxigênio para manter as chamas e calor para iniciar e continuar o processo de queima. Essa inter-relação entre os três elementos básicos da combustão é conhecida como triângulo do fogo. A ausência, ou redução abaixo de certos níveis, de qualquer um dos componentes do triângulo do fogo inviabiliza o processo da combustão (LIF, 2006).

Esta Seção tem como objetivo detalhar informações sobre a relação do comportamento do fogo de acordo com combustíveis florestais, as técnicas reais de operação, ferramentas e métodos utilizados atualmente por grupos humanos na identificação e combate de incêndios florestais.

2.4.2 Combustíveis Florestais

Os combustíveis florestais, produto da cobertura vegetal e sua dinâmica, são importantes parâmetros nos processos de ignição e propagação dos incêndios. O conhecimento das características básicas dos combustíveis, tais como: tipo, quantidade, continuidade e arranjo são muito úteis, tanto na estimativa do risco como na previsão do comportamento dos incêndios (CASTRO; TUDELA; SEBASTIA, 2001; CHUVIECO; CONGALTON, 1989; GOUMA; CHRONOPOULOU, 1998). Estudos dos modelos de florestas e resíduos florestais são de grande importância para o aprimoramento dos modelos de simulação a serem implementados em ambientes virtu-

ais. Além destes modelos, também é importante que seja feito um mapeamento do terreno e do tipo de vegetação da cobertura deste terreno, caso se deseje realizar uma reprodução mais fiel de um determinado ambiente.

Quanto aos tipos de combustíveis florestais, sugere-se os seguintes modelos (MA, 2006):

- **Modelo 1, herbáceo:** Pasto fino, seco e baixo, com altura abaixo do joelho, que cobre completamente o solo. O mato ou as árvores cobrem menos de $1/3$ da superfície. Os incêndios propagam-se com grande velocidade pelo pasto fino.
- **Modelo 2, herbáceo:** Pasto contínuo, fino, seco e baixo, com presença de mato ou árvores que cobrem entre $1/3$ e $2/3$ da superfície. Os combustíveis são formados pelo pasto seco, folhada e ramos caídos da vegetação lenhosa. Os incêndios propagam-se rapidamente pelo pasto fino.
- **Modelo 3, herbáceo:** Pasto contínuo, espesso, com mais de $1m$ sendo que $1/3$ ou mais do pasto deverá estar seco. Os incêndios são mais rápidos e de maior intensidade.
- **Modelo 4, arbustivo:** Matos ou árvores jovens muito densos, com cerca de $2m$ de altura. Continuidade horizontal e vertical do combustível. Abundância de combustível lenhoso morto (ramos) sobre as plantas vivas. O fogo propaga-se rapidamente sobre as copas dos matos com grande intensidade e com chamas grandes.
- **Modelo 5, arbustivo:** Mato denso baixo, com uma altura inferior a $0,6m$. Apresenta cargas ligeiras de folhada do mesmo mato, que contribui para a propagação do fogo em situação de ventos fracos. Fogos de intensidade moderada.
- **Modelo 6, arbustivo:** Mato mais velho do que no modelo 5, com alturas compreendidas entre os $0,6$ e $2m$ de altura. Os combustíveis vivos são mais escassos e dispersos. No conjunto é mais inflamável do que o modelo 5. O fogo propaga-se através do mato com ventos moderados a fortes.
- **Modelo 7, arbustivo:** Mato de espécies muito inflamáveis, de $0,6$ a $2m$ de altura, que propaga o fogo debaixo das árvores. O incêndio desenvolve-se com teores mais altos de umidade do combustível morto do que nos outros modelos, devido à natureza mais inflamável dos outros combustíveis vivos.
- **Modelo 8, manta morta:** Folhada em bosque denso. A folhada forma uma capa compacta ao estar formada de agulhas pequenas ($5cm$ ou menos) ou por folhas planas não muito grandes. Os fogos são de fraca intensidade, com chamas curtas e que avançam lentamente.

- **Modelo 9, manta morta:** Folhada em bosque denso que se diferencia do modelo 8 por formar uma camada pouco compacta e arejada. Fogo mais rápido e com chamas mais compridas do que as do modelo 8.
- **Modelo 10, manta morta:** Restos lenhosos originados naturalmente, incluindo lenha grossa caída como consequência de vendavais, pragas intensas ou excessiva maturação da massa, com presença de vegetação herbácea que cresce entre os restos lenhosos.
- **Modelo 11, resíduos lenhosos:** Resíduos ligeiros com diâmetro menor que *7,5cm*, recentes, de tratamentos silvícolas ou de aproveitamentos, formando uma capa pouco compacta de altura por volta de *30cm*. Incêndios com intensidades elevadas.
- **Modelo 12, resíduos lenhosos:** Resíduos de exploração mais pesados do que no modelo 11, formando uma capa contínua de altura até *60cm*. Incêndios com intensidades elevadas.
- **Modelo 13, resíduos lenhosos:** Grandes acúmulos de resíduos de exploração grossos com diâmetro maior que *7,5cm* e pesados, cobrindo todo o solo.

No trabalho (PESSIN et al., 2007e) apresentamos uma simulação de propagação de incêndios contendo áreas com os 13 modelos de combustíveis florestais sugeridos. Para esta dissertação, a simulação de propagação foi simplificada para respeitar o tipo de terreno escolhido para navegação dos agentes de combate a incêndio. Tentamos deixar o ambiente mais parecido possível com regiões como o Parque Nacional dos Aparados da Serra e o Parque Nacional da Serra Geral, onde predomina o Modelo 1 de combustível florestal.

2.4.3 Técnicas Reais de Operação

O estudo das técnicas reais de operação permite que possamos planejar melhor o formato da operação e as estratégias a serem implementadas no sistema. O conjunto das técnicas listadas abaixo é de consenso geral para aplicações em monitoramento e combate a incêndio.

2.4.3.1 Operações de combate

A operação de combate ou supressão de um incêndio envolve seis etapas distintas. Essas etapas são as seguintes (LIF, 2006):

1. **Detecção:** Tempo decorrido entre a ignição ou início do fogo e o momento em que ele é visto por alguém. Alguns métodos são: uso das torres de vigilância; patrulhamento terrestre; patrulhamento por avião; ou imagens de satélites.

2. **Comunicação:** Tempo compreendido entre a detecção do fogo e o recebimento da informação por um responsável. Os aparelhos usados para comunicação nas torres de observação são o rádio e o telefone.
3. **Mobilização:** Tempo gasto entre o recebimento da informação da existência do fogo e a saída do pessoal para combate. É necessário que haja uma pessoa responsável pela ação inicial, que deve definir claramente as atribuições e responsabilidade de todo o pessoal envolvido no combate ao incêndio.
4. **Deslocamento:** Tempo compreendido entre a saída do pessoal de combate e a chegada da primeira turma ao local do incêndio. O deslocamento das equipes depende de planejamento e definição da estratégia de ataque ao foco de incêndio. Usualmente temos um quartel-general de onde partem as ordens de deslocamento, que deve ser planejado de forma inteligente, assim como a manutenção das posições das brigadas durante o combate ao incêndio.
5. **Planejamento:** Tempo gasto pelo responsável pelo combate para avaliar o comportamento do fogo e planejar a estratégia de combate. O planejamento inicial do deslocamento das brigadas permite um ataque ao foco de incêndio de forma coordenada, visando um melhor resultado no combate ao incêndio. O plano inicial deve ser constantemente revisado de modo a considerar a evolução da situação do deslocamento das brigadas (*e.g.* atrasos, bloqueios imprevistos, dificuldades de avanço) e da propagação do incêndio (*e.g.* mudança na direção dos ventos, novos focos de incêndio).
6. **Combate:** Tempo consumido na operação de combate ou eliminação do incêndio. Devem ser consideradas para o tempo de combate as diferentes técnicas de combate a incêndios.

2.4.3.2 Equipamentos de combate

O tipo e a quantidade de equipamento para combate a incêndios florestais em uma instituição florestal dependem de fatores como tipo de vegetação, topografia, tamanho da área e pessoal disponível. De um modo geral, as ferramentas e equipamentos usados no combate aos incêndios florestais são os seguintes (LIF, 2006):

- **Ferramentas manuais:** enxadas, machados, foices, pás, abafadores, extintores e lança-chamas (usado na criação de contra-fogos).
- **Equipamentos de apoio:** lanternas (durante a noite), capacetes, cantis e equipamentos de primeiros socorros (com medicamentos para queimaduras, cortes e soro antiofídico).

- **Equipamentos motorizados:** moto-serra (usada para derrubar rapidamente árvores que estejam queimando ou para abrir pequenos aceiros) e atomizador costal (usado para lançar água ou retardante químico em incêndios superficiais).
- **Equipamentos pesados:** trator com lâmina (indispensável no combate a grandes incêndios, principalmente quando se precisa abrir aceiros⁴ em vegetação pesada) e moto-niveladora (usada para abrir ou ampliar aceiros).
- **Equipamentos de bombeamento de água:** moto-bomba portátil (útil quando existe uma boa rede de pontos de captação de água), carro-tanque (somente opera com eficiência em boas estradas), helicóptero e avião-tanque (Figura 2.30) (BOMBARDIER, 2007) (despejam água retardando a propagação do fogo até que as equipes de terra realizem a extinção total).



Figura 2.30: Avião-tanque *Canadair CL-415* (BOMBARDIER, 2007), transporta 6t de água, abastece sem pousar em uma superfície de água com 1,2km de comprimento e 2m de profundidade.

É interessante ressaltar que é necessário aplicar cerca de 10 vezes o peso do material combustível existente em água para se extinguir o fogo, assim, mais eficiente que a água é a aplicação aérea de retardantes químicos, como sulfato de amônia, diamônia fosfato, borato de cálcio e sódio sobre a vegetação ainda não atingida pelo fogo (RIBEIRO et al., 2006).

⁴Aceiro é uma área livre de vegetação, onde o fogo, por não encontrar material combustível, se extingue.

2.4.3.3 Métodos e estratégias de combate

Existem pelo menos quatro métodos de combate ao fogo nos incêndios florestais (LIF, 2006):

- **Método direto:** Usado quando a intensidade do fogo permite uma aproximação suficiente da brigada à linha de fogo. Pode-se utilizar água, terra (pás) ou abafadores.
- **Método paralelo ou intermediário:** Usado quando não é possível o método direto e a intensidade do fogo não é muito grande. Consiste em limpar, com ferramentas manuais, uma estreita faixa (aceiro), próxima ao fogo, para deter o seu avanço e possibilitar o ataque direto.
- **Método indireto:** Usado em incêndios de intensidade muito grande. Consiste em abrir aceiros com equipamento pesado (tratores). Pode-se utilizar um contra-fogo para ampliar a faixa limpa e deter o fogo antes que chegue ao aceiro.
- **Método aéreo:** Usado nos incêndios de copa, de grande intensidade e área e em locais de difícil acesso às brigadas de incêndio. São usados aviões e helicópteros especialmente construídos ou adaptados para o combate ao incêndio.

Neste trabalho, o agente monitor é responsável pelas etapas de detecção, planejamento e comunicação. A forma de combate a incêndio implementada utiliza equipamento pesado, assim os veículos de combate a incêndio são motoniveladoras; e o método de combate é o indireto; utilizando as motoniveladoras para criação de um aceiro.

2.4.3.4 Sistemas de detecção

Quanto mais rápida a detecção de um foco de incêndio, maior serão as chances de êxito no combate e na minimização dos danos. Os métodos de detecção podem ser (RAMOS, 1995):

- **Sistemas de monitoramento local:** Realizado por pessoal designado especificamente para missão de patrulhamento. Os meios de suporte para a realização do monitoramento podem variar, como: usando torres de observação, binóculos, veículos de patrulhamento ou sensores.
- **Sistema de monitoramento aéreo:** Identificação dos focos realizada por aeronaves, baseado em reconhecimento visual.

- **Sistemas de monitoramento por satélite:** Identificação dos focos realizada através de fotos de satélite.

Para países de grande extensão territorial, como o Brasil, o monitoramento dos incêndios florestais através de imagens de satélites é o meio mais eficiente e de baixo custo quando comparado com os demais meios de detecção (BATISTA, 2004).

O lançamento em 1972 do primeiro satélite *Landsat* possibilitou detectar alterações nas áreas florestais através do espaço. Desde então, imagens termais têm sido usadas na detecção de incêndios e estudos de mapeamento, permitindo que áreas queimadas e não queimadas sejam detectadas através do contraste entre os gradientes térmicos (PERERA; REMMEL, 2001). O reconhecimento por satélite é uma realidade, por exemplo, no projeto PREVFOGO (Sistema Nacional de Prevenção e Combate aos Incêndios Florestais), a identificação dos focos é realizada por meio de sensores térmicos AVHRR (*Advanced Very High Resolution Radiometer*) do satélite meteorológico da série NOAA-11, que realiza evoluções a cada seis horas, numa órbita situada a 800km de altitude. Os dados são recebidos pelo INPE, que os processa retransmitindo as informações ao Centro Nacional de Monitoramento e Controle de Incêndios Florestais (CNMC) em Brasília, onde são gerados mapas nos quais são identificadas a localização e a intensidade do foco (RAMOS, 1995).

O Instituto Nacional de Pesquisas Espaciais (INPE), através do Sistema de Monitoramento, Prevenção e Controle de Incêndios Florestais na Amazônia (PROARCO), monitora as ocorrências de incêndios para todos os estados brasileiros através de sensores espaciais (BATISTA, 2004). Sensores AVHRR captam e registram qualquer temperatura acima de 47°C, interpretadas como sendo um foco de calor. Apesar da resolução espacial ser baixa (1,1km), queimadas com áreas de no mínimo 900m² podem ser detectadas (ANTUNES, 2000). Dados obtidos através das imagens termais de satélites meteorológicos são integrados a dois sistemas de informação geográfica que podem ser acessados pela internet, como mostram as Figuras 2.31 e 2.32 obtidas no endereço <http://www.dpi.inpe.br/proarco/bdqueimadas>. Todo o Brasil e grande parte da América do Sul são cobertos pelas imagens (CPTEC/INPE, 2007).

2.4.4 Considerações sobre Incêndios em Ambientes Naturais

Os aceiros têm sido uma alternativa viável e de amplo uso no meio florestal, nas unidades de conservação e nas margens de rodovias para a redução na propagação do fogo (RIBEIRO et al., 2006). O principal objetivo de um aceiro é o de quebrar a continuidade do material combustível, dividindo uma área em compartimentos. O surgimento de novos produtos e de novos



Figura 2.31: Foco de incêndio próximo ao Parque Nacional dos Aparados da Serra, imagem da Divisão de Processamento de Imagens do INPE, satélite *LandSat* 2000 (DPI/INPE, 2007).

CAMPO	VALOR
ID	S290654050064820060926171400
Lat	-29.115000
Long	-50.146667
LatGMS	S 29 6 54.00
LongGMS	O 50 8 48.00
Data-Hora/Date-Time/Fecha-Hora	2006-09-26 17:14:00
Satélite/Satellite/Satélite	AQUA
Município/City/Localidad	Cambará do Sul
Estado/State/Provincia	RS
Unidade Conservacao/Conservation Unit/Area de Conservación	P.N. de Aparados da Serra
Terra Indigena/Reserved for Indian/Área Indígena	Amarelo
Floresta Nacional/National Forest/Floresta Nacional	0.00

Figura 2.32: Tabela de informações disponibilizada junto a foto do foco de incêndio (DPI/INPE, 2007).

equipamentos de aplicação tem sugerido que aceiros químicos podem ser muito eficientes em combate a incêndios florestais (RIBEIRO et al., 2006). A eficiência de um aceiro depende da composição da vegetação e da estrutura do material combustível existente em suas proximidades, principalmente pelas características das espécies dominantes, altura e proximidade da vegetação, assim, vários estratos da vegetação devem ser considerados (RIGOLOT, 2002).

Existem modelos matemáticos, bastante complexos, para se estimar a velocidade de propagação de um incêndio (LIF, 2006), como o modelo proposto por (ROTHERMEL, 1983) que é um modelo empírico, baseado em experiências laboratoriais. No nosso sistema utilizaremos velocidades de propagação não baseadas em modelos matemáticos, mas sim na coleta de dados de campo, com base no trabalho de (KOPROSKI, 2005) que cita, por exemplo: vento de 20km/h com velocidade de propagação do incêndio de $0,09\text{m/s}$ (324m/h) em vegetação como a do Modelo 4 e velocidade do vento de 60km/h com velocidade de propagação do incêndio de $1,94\text{m/s}$ ($6,984\text{km/h}$) em vegetação como a do Modelo 1.

2.5 Considerações Finais

O objetivo deste Capítulo foi revisar conceitos e tecnologias importantes relacionadas ao problema proposto.

A primeira Seção, sobre robótica móvel, detalhou os princípios envolvidos em sistemas de controle, sensores e atuadores, além de descrever aspectos de morfologia e de simulação de sistemas multi-robóticos fisicamente realísticos. Podemos ver que é bastante comum na robótica móvel a utilização de sensores do tipo sonar para obtenção de distâncias, assim, nosso veículo contará com este tipo de sensor, além dos sensores bússola e GPS e de dois atuadores. Os agentes de combate a incêndio implementados realizam navegação com constante leitura de informação sensorial, sendo assim essencialmente do tipo *feedback*; o agente deliberativo (planejador de estratégia) receberá informação do ambiente apenas na inicialização, assim, pode ser considerado como possuindo estratégia de controle do tipo *feedforward*. Respeitamos, na definição da arquitetura de controle, todas as tarefas descritas na Seção 2.1.5.2. Ainda, na questão de controle, o agente responsável pela evolução de estratégias é essencialmente deliberativo; os agentes de combate a incêndio, por receberem um macro-comando de “*desloque-se autonomamente até (x,y)*” possuem um plano para execução desta navegação, além de possuir um sistema de reação sensorial-motora, assim, podem ser considerados como agentes híbridos. As ferramentas de simulação pesquisadas serviram como inspiração para o desenvolvimento do nosso ambiente; descrevemos tanto as ferramentas estudadas como as bibliotecas de software

escolhidas para o desenvolvimento do nosso ambiente de simulação.

A segunda Seção, sobre agentes inteligentes e sistemas multi-agentes, apresentou conceitos e características importantes relacionadas com a criação de sistemas multi-robóticos. Podemos ver que a tarefa de monitoração e combate a incêndios apresenta um problema onde a atuação de um sistema multi-agente pode trazer grandes benefícios, tanto por poder mapear o funcionamento de uma operação real como por, no caso deste trabalho, apresentar um modelo com veículos autônomos, o que reduz os riscos humanos na operação. Neste trabalho, a cooperação é resultado de uma evolução usando Algoritmos Genéticos, assim, não teremos comunicação entre agentes para este fim. A comunicação simulada neste trabalho é detalhada na Seção 4.1. Com relação aos quatro aspectos principais que concretizam o projeto de um agente robótico: desenvolvemos os agentes de combate a incêndio com capacidade de autonomia e localização; por serem fisicamente simulados apresentam também a corporificação; no entanto, não consideramos carga de bateria ou combustível, como se fossem auto-suficientes a ponto de cumprir a missão que lhes foi atribuída sem necessidade de realizar um reabastecimento. No desenvolvimento do sistema multi-agente implementamos a definição da estratégia de forma centralizada e a atuação no combate a incêndio de forma distribuída.

A terceira Seção tratou de Aprendizado de Máquina, onde foram detalhadas as principais técnicas abordadas neste trabalho. Propomos o uso de Algoritmos Genéticos para a evolução de estratégias de posicionamento do grupo robótico de combate a incêndios, a vantagem de se utilizar GA para a otimização das posições é que eles não necessitam de uma base com os dados de treinamento; no caso do combate a incêndios, estas informações seriam muito difíceis ou mesmo impossíveis de serem obtidas. Propomos o uso de uma Rede Neural Artificial para realizar o controle inteligente em um RMA que deve ser capaz de navegar desviando de obstáculos; o uso de RNA no controle de navegação é justificado por serem as técnicas mais robustas em relação a situações novas e inesperadas, por possuírem alto grau de generalização e, como o desenvolvimento de regras de controle é uma operação bastante complexa, podemos criar uma base de treino através da obtenção de dados de um veículo operado manualmente, para posterior treino e teste da RNA.

A última Seção, sobre incêndios em ambientes naturais, descreveu as técnicas reais de operação, as ferramentas e os métodos utilizados atualmente por grupos humanos na identificação e combate a incêndios florestais. Podemos ver que os aceiros têm sido uma alternativa viável e de amplo uso no meio florestal, nas unidades de conservação e nas margens de rodovias para a redução na propagação do fogo (RIBEIRO et al., 2006). Propomos um agente monitor responsável pelas etapas de detecção, planejamento e comunicação. Sendo a forma de

combate a incêndio implementada utilizando equipamento pesado, assim os veículos de combate a incêndio são motoniveladoras; e o método de combate é o indireto; utilizando as motoniveladoras para criação de um aceiro. O terreno escolhido é semelhante aos parques nacionais da região nordeste do Rio Grande do Sul, que apresentam predomínio de combustível florestal do Modelo 1.

O próximo Capítulo descreve os trabalhos relacionados com o foco desta dissertação. Após este levantamento bibliográfico podemos propor o modelo sensorial-motor mais adequado para os agentes de combate a incêndios no terreno proposto; e nos dá a base para implementar, simular e avaliar a eficiência de: técnicas de RNA aplicadas no controle de navegação dos agentes robóticos; e técnicas de GA aplicadas na evolução de estratégias de posicionamento e cooperação de grupos robóticos.

3 TRABALHOS RELACIONADOS

Segundo (KITANO, 2000), a robótica e a Inteligência Artificial podem contribuir para melhorar as ferramentas usadas em casos de desastres de duas maneiras: com a construção de um simulador que permita compreender e otimizar as estratégias de resgate; e com o desenvolvimento de robôs ou times de robôs para exploração e atuação em ambientes hostis.

Este Capítulo descreve trabalhos desenvolvidos em outros projetos ou outros centros de pesquisa que tenham relação com o foco desta dissertação. Iniciamos apresentando projetos reais de robôs aplicados ao problema de operação em ambientes hostis, em seguida são apresentados trabalhos relacionados com as áreas de controle robótico e evolução de estratégias. Após isso são apresentados trabalhos sobre formação de grupos e sobre formas de comunicação. Finalizando com as considerações sobre o Capítulo.

3.1 Operações com Robôs Reais

A Seção 2.1 apresentou diversos trabalhos atuais em robótica móvel, bem como ferramentas de simulação. Além destes trabalhos, é interessante conhecer mais alguns projetos diretamente aplicados ao problema de operação em ambientes hostis.

O Raposa (IDMIND, 2007) (Robô semi-autônomo para operações de salvamento) (Figura 3.1) é projetado com a finalidade de auxílio em resgate. É um projeto português que envolve o *Regimento de Bombeiros* de Lisboa e o *Perceptual Robotics Lab* da Universidade da Flórida do Sul (EUA) na criação de um robô tele-operado para atuar em ambientes hostis à presença humana, tais como escombros resultantes de um terremoto ou atentado (IDMIND, 2007). O Raposa possui 17 sonares e um GPS.

O SACI (Sistema de Apoio ao Combate de Incidentes) (MACEDO; MACEDO; DUARTE, 2007) é um robô tele-operado projetado para atuar em combate de incêndios em áreas hostis à presença humana. Sua capacidade extintora envolve a possibilidade de gerar névoa ou de liberar um jato sólido de água ou espuma. O SACI deve carregar junto de si a fonte de água ou



Figura 3.1: Raposa, projeto português de robô para auxílio em operações de resgate (IDMIND, 2007).

da espuma, ou seja, mangueiras devem ser acopladas a sua estrutura, como mostra a Figura 3.2. A última versão do robô conta com controle remoto sem fio e câmeras de vídeo. É construído para suportar temperaturas de até 100°C e trabalhar sem recarga de bateria por até 6h.



Figura 3.2: Robô SACI (Sistema de Apoio ao Combate de Incidentes) (MACEDO; MACEDO; DUARTE, 2007).

O *Fire-Fighting Home Robot Contest* (TRINITY, 2007), o *Concurso Robô Bombeiro* (GUARDA, 2007) e o *Robot Challenge and Exhibit* (PENNSTATE, 2007) são competições robóticas que põem à prova pequenos robôs móveis autônomos que têm como missão encontrar e apagar uma vela, como um incêndio simulado, num modelo de aproximadamente $2m^2$ representando um casa formada por corredores e quartos. Esta tarefa simula a operação real de um robô autônomo realizando proteção contra incêndios em residências (TRINITY, 2007). Em ambas as competições, os robôs devem ser construídos pelos competidores; vence o robô que apaga a vela no menor espaço de tempo. Os trabalhos (MILLER et al., 2003; PACK et al., 2004) apresentam o desenvolvimento do hardware de robôs designados a participar de competições como as descritas acima.

Os trabalhos descritos nesta Seção demonstram que a aplicação de robótica móvel em controle de incidentes é um tópico ativo e importante de pesquisa e desenvolvimento. As diver-

sas competições nesta área também demonstram que ainda não existe uma solução definitiva ou mais adequada ao problema, e que este é um tema de pesquisa “em aberto”.

3.2 Formas de Controle e de Evolução de Estratégias

Um trabalho de referência na questão de controle inteligente baseado em técnicas de controle reativo, deliberativo e híbrido utilizando Redes Neurais Artificiais é o SEVA3D (OSÓRIO et al., 2006a; HEINEN; OSÓRIO, 2005), fruto de pesquisas desenvolvidas pelo Grupo de Inteligência Artificial do PIPCA (Unisinos). Inicialmente, o desenvolvimento do sistema SEVA (Simulador de Estacionamento de Veículos Autônomos) (OSÓRIO; HEINEN; FORTES, 2001), realizava a tarefa de estacionamento de um robô não-holonômico (tipo carro) em uma vaga paralela em um ambiente bidimensional (2D) utilizando 6 sensores do tipo infra-vermelho; controlado através do uso de um autômato finito (SEVA-A) ou através de uma Rede Neural (SEVA-N) do tipo *Jordan Cascade-Correlation* (OSÓRIO; HEINEN; FORTES, 2002). A evolução para uma versão com ambiente tridimensional foi feita com o simulador SimRob3D (HEINEN, 2002) e tornou o modelo mais realista. As principais alterações derivadas da evolução do modelo foram a troca dos sensores de infra-vermelho para sonar e a remoção de um dos sensores, ficando com um total de 5 sensores do tipo sonar como mostra a Figura 3.3.



Figura 3.3: Distribuição dos sensores do tipo sonar no veículo usado do SEVA3D (OSÓRIO et al., 2006b).

Outros trabalhos onde existe aplicação de controle reativo em agentes robóticos podem ser vistos em (BIANCHI; SIMÕES; COSTA, 2001; COSTA; GOMES; BIANCHI, 2003; HOFFMANN; CASTRO; SILVA, 2004; SELVATICI, 2005). Estes trabalhos realizam o aprendizado utilizando técnicas de Aprendizado por Reforço (REZENDE, 2003). Ainda, o trabalho de (ZHAO; COLLINS, 2005) usou um controlador baseado em Lógica Fuzzy (REZENDE, 2003) para realizar o controle de deslocamento em um veículo equipado com 17 sensores do tipo sonar, disposto como mostra a Figura 3.4 e um sensor do tipo odômetro.

Originalmente, controles baseados em regras foram apresentados ainda na década de 50.

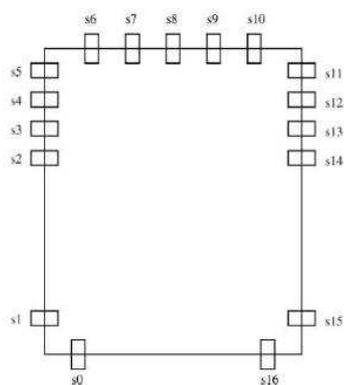


Figura 3.4: Distribuição dos sensores no veículo (ZHAO; COLLINS, 2005).

William Walter (WALTER, 1950) construiu um robô móvel capaz de executar tarefas tais como desviar de obstáculos e seguir fontes luminosas. O robô móvel *SHAKY* desenvolvido por Nils Nilsson (NILSSON, 1969) em Stanford, utilizava dois motores de passo, usando cinemática diferencial para se locomover; era equipado com sensores de distância, câmeras de vídeo e sensores táteis. Conectado a dois computadores utilizava programas para percepção, modelagem e atuação no ambiente. As tarefas desempenhadas pelo robô incluíam desviar de obstáculos e a movimentação de blocos coloridos. Este robô tinha grandes dificuldades em processar e interpretar as informações sensoriais obtidas do ambiente, sendo que nunca foi capaz de completar uma seqüência completa de ações em um ambiente real.

Na metade da década de 1990 foi proposto o *ALVINN* (*Autonomous land vehicle in a neural network*) (JOCHEM; POMERLEAU; THORPE, 1993; POMERLEAU, 1994; JOCHEM; POMERLEAU; THORPE, 1995; BALUJA, 1996), um programa baseado em Redes Neurais Artificiais que, a partir de um prévio treinamento supervisionado, era capaz de guiar um veículo em uma pista, usando o sistema de visão computacional do NavLab (CMU, 2007). Desde 2004, o *DARPA Grand Challenge* (DARPA, 2007), financiado pelo *Defense Advanced Research Projects Agency*, a principal organização de pesquisa do departamento de defesa dos Estados Unidos, propõe uma competição que tem como objetivo construir veículos autônomos para navegar grandes distâncias em estradas. O objetivo principal é a pesquisa para o desenvolvimento de tecnologias para a criação do primeiro veículo completamente autônomo capaz de completar um longo trecho em estradas de terra em um tempo limitado. Em (GRAYMATTER, 2007; OSHKOSH, 2007) é possível conhecer mais detalhes sobre alguns grupos que participam desta competição. Em (WHITTAKER, 2004) podemos ver a descrição completa da construção de um dos veículos participante do evento.

Constata-se, a partir dos trabalhos citados acima, que o controle de navegação em robôs móveis autônomos não é uma tarefa simples, onde muitas pesquisas foram direcionadas no

sentido de adotar técnicas de Aprendizado de Máquina, notadamente através do uso de RNAs para o controle de RMAs.

Quanto a evolução de estratégias, pesquisas utilizando Algoritmos Genéticos tem apresentado grande sucesso nesta área. Em (GONZALEZ; MARENGONI, 2005) utiliza-se um GA para evoluir a direção, o sentido e o tempo de acionamento de motores de um braço robótico; o GA proposto realiza satisfatoriamente a evolução dos valores a serem aplicados no manipulador robótico. Em (HEINEN, 2007), um modelo de GA evolui satisfatoriamente valores de força e tempo de aplicação das forças para realizar o caminhar robótico em um robô com pernas fisicamente simulado. Neste trabalho, o uso dos sensores odômetro e giroscópio foram fundamentais para o cálculo do *fitness*, baseado na distância total percorrida e na estabilidade do agente.

O trabalho de (ZHOU, 2005) apresenta um modelo de GA para evoluir a forma como um único robô móvel explora um ambiente desconhecido; para este trabalho, o terreno de atuação considerado era plano. Não foi apresentada a simulação em protótipo, sendo o resultado satisfatório demonstrado através de gráficos de caminhadas (sequências de coordenadas). Um outro trabalho, de (LUCIDARME, 2004) propõe evoluir formas de controle para navegação de robôs móveis em ambientes usando comandos do tipo: gire para a direita, gire para a esquerda, ande para frente e ande para trás.

Estes dois últimos trabalhos descritos apresentam resultados satisfatórios apenas para ambientes estáticos; qualquer mudança no ambiente necessita de uma nova evolução, além disso, o resultado destes modelos de GA não são funcionais em ambientes com obstáculos dinâmicos. O trabalho de (BURCHARDT; SALOMON, 2006) apresenta uma possível solução para o problema de operação em ambientes dinâmicos, descrevendo uma operação usando um GA que evolui trajetórias entre coordenadas específicas, sendo o robô munido de sensores de obstáculos. Quando o robô identifica uma possível colisão, ele pára e reativa o módulo de planejamento usando o GA, desta maneira o sistema se torna de possível uso em ambientes com obstáculos móveis.

Todos os trabalhos descritos sobre evolução usando GA apresentaram resultados satisfatórios, sendo que cada um usa um conjunto de parâmetros genéticos bastante específico para cada caso.

3.3 Formação de Grupos e Formas de Comunicação

Os trabalhos de (BALCH; ARKIN, 1995, 1998) abordam a formação de grupos de 3 maneiras: referência ao líder, onde todos os agentes seguem um, assim, o agente líder não se preocupa

com a formação, apenas se move pelo ambiente e os outros agentes devem acompanhá-lo; referência ao vizinho, onde a formação é mantida através de posições relativas pré-determinadas dos vizinhos; ou, referência ao ponto central, onde cada robô calcula sua posição (x, y) envolvida na formação. Uma forma parecida a esta última foi usada em (PESSIN et al., 2007e). Outro trabalho interessante na área de formação de grupos, usando regras, pode ser visto em (OSAGIE, 2006).

Trabalhos com formação de grupos são o cerne da cooperação, Parker (PARKER, 2000) categoriza sistemas cooperativos multi-agentes em dois tipos: cooperação intencional e cooperação coletiva emergente. Nos dois modelos, cada agente deve conhecer a meta global, porém, na cooperação emergente os agentes recebem um tempo para negociar o comportamento cooperativo, ao passo que na cooperação intencional, a regra de cooperação está pré-programada nos agentes. Os trabalhos de (TAKAHASHI; TAMURA; ASADA, 2001; DORIGO et al., 2004; MONDADA et al., 2005) apresentam métodos para emergir cooperação em grupos robóticos.

A Seção 2.2.5 apresentou algumas formas de comunicação em sistemas multi-agentes. O uso da comunicação em sistemas robóticos reais é bastante suscetível a erros e interferências. O trabalho de (REMLEY et al., 2007) mostra uma análise sobre comunicação *wireless* em robôs de resgate e exploração urbanos, apresentando que entre os 14 robôs testados, 10 apresentaram problemas de comunicação advindos de interferência. Mesmo utilizando frequência reservada de uso industrial, científico e médico, designada a minimizar a interferência, os pesquisadores não conseguiram garantir comunicação 100% confiável. Algumas sugestões para melhorar a comunicação urbana foram apresentadas, como novas mudanças de frequências, protocolos de transmissão, força de transmissão e uso de repetidores. Os trabalhos de (DATCU et al., 2004; MONDADA et al., 2005) apresentam grupos de agentes com comunicação *wireless*; os robôs são utilizados fisicamente próximos uns dos outros.

Em comunicação simulada, (HEINEN, 2002) apresenta um trabalho com arquitetura de controle híbrida, onde a comunicação entre as camadas faz uso de um *blackboard*. Um trabalho relacionado à área de coordenação em sistemas multi-agentes, descrito em (PESSIN, 2007), apresenta um sistema multi-agente distribuído onde, através de comunicação, agentes negociam suas rotas a fim de cercar um alvo (objetivo comum) de forma a maximizar a velocidade da operação. A velocidade da operação é maximizada fazendo com que os agentes mais distantes do alvo se desloquem ao ponto final mais próximo possível (Figura 3.5). A comunicação foi implementada através de uso de quadro-negro (*blackboard*) e o controle de movimentação e a heurística da escolha dos trajetos foram definidos através de regras. A abordagem distribuída se

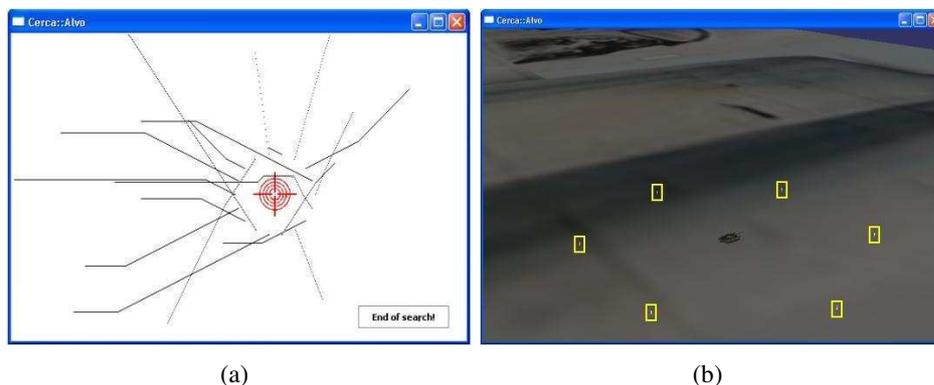


Figura 3.5: Resultados dos protótipos (PESSIN, 2007): (a) Modelo 2D com dezoito agentes cercando um alvo; (b) Modelo 3D com seis agentes.

dá devido ao fato de que a negociação das rotas é realizada através de comunicação entre todos os agentes, por este motivo, esta abordagem gerou um número excessivamente grande de trocas de mensagens. O trabalho apresentado em (PESSIN et al., 2007e) simula Troca Direta de Mensagens; que especificam remetente, destinatário e tipo da mensagem (*e.g.* aviso de incêndio, aviso de fim de incêndio, *timestamp*, solicitação de formação).

3.4 Considerações Finais

Neste Capítulo foram descritos trabalhos desenvolvidos em outros projetos ou outros centros de pesquisa que tem relação com o foco desta dissertação. Iniciamos apresentando projetos reais de robôs aplicados ao problema de operação em ambientes hostis, em seguida apresentamos trabalhos relacionados com as áreas de controle robótico e evolução de estratégias. Após isso apresentamos trabalhos sobre formação de grupos e sobre formas de comunicação.

Os trabalhos descritos nesta Seção demonstram que esta dissertação tem como base tópicos ativos e importantes de pesquisa e desenvolvimento. Como apresentado, a tarefa de controle de navegação ainda é uma área de pesquisa que conta com grandes financiamentos e com interesse militar. Vimos também que os trabalhos descritos sobre evolução de estratégia usando GA apresentaram resultados satisfatórios, porém, cada trabalho usou um conjunto de parâmetros genéticos bastante específico para cada caso.

Alguns dos trabalhos descritos em (PESSIN, 2007) mostraram aspectos como o quão árdua é a criação de regras de controle em sistemas robóticos, o que nos leva a buscar a evolução do modelo proposto em (PESSIN et al., 2007e) substituindo as regras por técnicas de Inteligência Artificial; tanto no controle como na evolução de estratégias. Em um trabalho anteriormente desenvolvido, o uso de uma RNA permitiu realizar satisfatoriamente o aprendizado em um agente

híbrido com dois objetivos: desviar de obstáculos e atingir um alvo, porém implementado em um ambiente 2D e sem simulação física (PESSIN et al., 2007f).

Trabalhos com comunicação simulada através de *blackboard* ou de Troca Direta de Mensagens mostraram-se eficientes e simples de desenvolver, porém, uma implementação física real deve levar em conta um estudo mais aprofundado sobre este tópico.

O uso de uma biblioteca de simulação física também se faz de suma importância, devido a possibilidade de nos permitir observar erros usualmente não observados nas simulações que não usam bibliotecas físicas, como bloqueio de veículos em obstáculos ou em áreas intransponíveis do terrenos (*e.g.* buracos, subidas íngremes). Precisaremos desenvolver um conjunto de sensores adequados para o correto comportamento relacionado com o ambiente, para que, pelo menos, respeitemos as propriedades de velocidade dos veículos e do incêndio. O teste com sensores reais também se mostra importante; os erros médios dos sensores devem ser observados e tratados.

Tomamos por base exemplos de outros trabalhos principalmente na questão referente ao posicionamento e ao tipo dos sensores; com relação ao desenvolvimento do ambiente, seguimos o exemplo de ferramentas como o *Webots* (CYBERBOTICS, 2007) e o *Microsoft Robotics Studio* (MICROSOFT, 2007), porém adaptadas a nossa necessidade, muitas delas específicas para o tipo de aplicação abordada: o combate a incêndios em ambientes florestais.

Em conjunto com o Capítulo anterior, o levantamento bibliográfico apresentado neste Capítulo nos permite propor o modelo sensorial-motor mais adequado para os agentes de combate a incêndios no terreno proposto; e nos dá a base para implementar, simular e avaliar a eficiência das técnicas de RNA aplicadas no controle de navegação dos agentes robóticos e das técnicas de GA aplicadas na evolução de estratégias de posicionamento e cooperação de grupos robóticos.

4 MODELO PROPOSTO

O modelo de operação multi-agente para identificação e combate a incêndios foi inicialmente descrito em (PESSIN et al., 2007e). Este modelo teve um protótipo *2D* desenvolvido com a biblioteca SDL, sendo a operação multi-agente completamente baseada em regras. O modelo de controle baseado em RNA foi inicialmente descrito em (PESSIN et al., 2007f). Este modelo de controle foi aplicado satisfatoriamente no mesmo protótipo *2D* desenvolvido para o trabalho de (PESSIN et al., 2007e). A proposta inicial da RNA aplicada no controle inteligente de veículos autônomos fisicamente simulados (para ação), usando um protótipo *3D* com chão plano desenvolvido com ODE e *Drawstuff* pode ser vista em (PESSIN et al., 2007b, 2007g); posteriormente esta RNA foi aplicada em um protótipo *3D* com chão irregular, desenvolvido com as bibliotecas ODE, OSG e *Demeter*. A proposta do modelo de operação (arquitetura) usando GA e RNA foi inicialmente descrita em (PESSIN et al., 2007c, 2007a). O modelo de AG para evolução de estratégia (planejamento), desenvolvido com a GALib, foi inicialmente descrito em (PESSIN et al., 2007d).

Este Capítulo apresenta o modelo do sistema multi-robótico proposto nesta dissertação. Dado que uma equipe de agentes autônomos deve trabalhar cooperativamente a fim de realizar com sucesso a identificação e o combate a incêndios em áreas florestais, necessitamos de um ambiente que suporte características fundamentais para a simulação realística da operação, com terrenos irregulares, processos naturais e restrições físicas na criação e uso de robôs móveis. Neste ambiente de simulação é realizada a operação multi-agente propriamente dita, que depende essencialmente de duas etapas: **planejamento e ação**.

4.1 Descrição Geral

Imagine uma área de floresta, como um parque ou uma reserva nacional de proteção ao meio ambiente. Incêndios relativamente aleatórios podem iniciar nestas áreas, sendo assim imprescindível a existência de uma ou mais unidades de monitoração de focos de incêndio. A unidade monitora deve, ao identificar um foco de incêndio, ter capacidade de *planejar* a

formação de atuação (estratégias de combate mais eficientes) para a quantidade de unidades de combate a incêndios disponíveis. Usamos técnicas de Algoritmos Genéticos em um agente deliberativo (unidade monitora) a fim de planejar e otimizar as posições de atuação de um grupo de unidades de combate. As unidades de combate (agentes híbridos), fisicamente simuladas, devem ser capazes de realizar navegação com desvio de obstáculos de forma autônoma, assim fazem uso de uma Rede Neural Artificial que, baseada em dados de sensores, coordena os atuadores e a **ação** destas unidades. O controle de navegação é local e sensível a mudanças no ambiente, assim, permite a operação dos robôs móveis em um ambiente dinâmico.

Como sugerido por (BATISTA, 2004) temos um agente monitor de focos de incêndio que faz uso de informações como as captadas por satélites (como as informações das Figuras 2.31 e 2.32). Ao identificar um foco, o agente monitor dispara o módulo de evolução de estratégias. Ao finalizar a evolução, envia uma mensagem passando, para cada agente de combate: “ $agir(x_i, y_i)(x_f, y_f)$ ”, então cada agente de combate deve se deslocar de sua posição atual até a posição (x_i, y_i) e começar a criar um aceiro até a posição (x_f, y_f) . Os agentes de combate inicialmente ficam em uma base de operação. O método de atuação de combate é, como sugerido por (LIF, 2006) o de criação de aceiros. A Figura 4.1 apresenta o diagrama da operação.

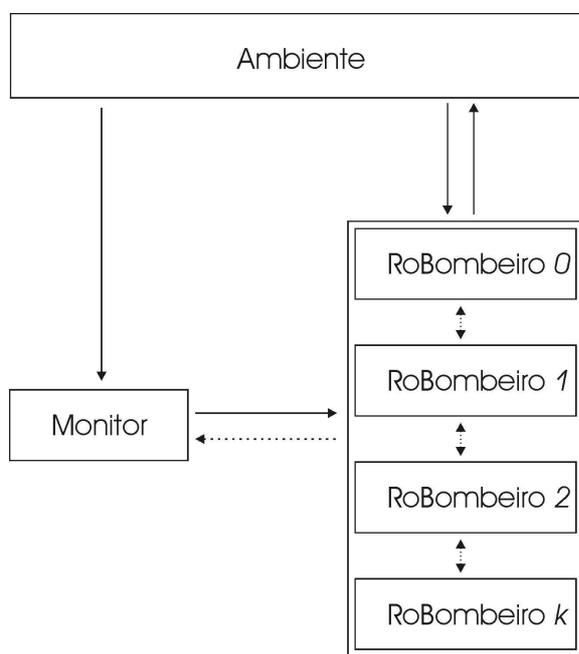


Figura 4.1: Diagrama da operação multi-agente.

Na Figura 4.1, a seta entre o ambiente e o agente monitor apresenta a capacidade do agente monitor perceber o ambiente. As setas entre o ambiente e os agentes de combate apresentam a capacidade destes agentes sentirem e atuarem no ambiente. A seta entre o agente monitor

e os agentes de combate apresenta a capacidade de comunicação da ordem de execução. A seta entre os agentes de combate e o agente monitor apresenta a capacidade de comunicação de *feedback* de operação. As seta entre os agentes de combate apresentam a capacidade comunicação entre estes agentes. Cada uma destas operações será detalhada no decorrer do texto. As setas tracejadas apresentam capacidades implementadas parcialmente ou não implementadas nos protótipos.

Sendo assim, o sistema proposto possui duas fases: a primeira é o planejamento da estratégia de ação usando um AG; a segunda é a execução da ordem do planejamento por robôs de combate fisicamente simulados usando uma RNA para controle da navegação com desvio de obstáculos, como apresenta a Figura 4.2.

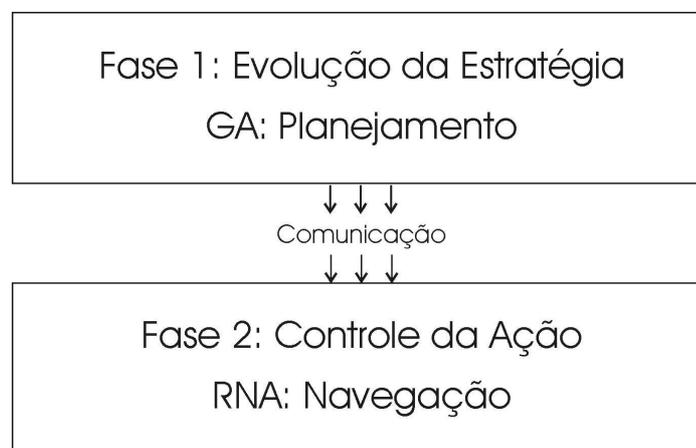


Figura 4.2: Fases do sistema proposto.

Cada fase do sistema proposto, cujos modelos são detalhados nas próximas seções, foram implementadas em protótipos: o primeiro protótipo é referente a fase 1, realiza a evolução das estratégias usando um Algoritmo Genético, não necessita obrigatoriamente de visualização, porém está implementado com a possibilidade de uma saída gráfica 2D desenvolvida com SDL; o segundo protótipo é referente a fase 2, suporta a ação dos robôs bombeiros, é simulado em um ambiente tridimensional e usa a biblioteca OSG, responsável pela saída gráfica, a biblioteca *Demeter*, responsável pelo terreno irregular e a biblioteca ODE, responsável pelo realismo físico, tanto da morfologia robótica como da colisão entre os objetos presentes no ambiente (*e.g.* robôs, árvores, inclinação de terreno). O uso da biblioteca ODE permite que os robôs simulados fisicamente respeitem questões como gravidade, inércia e atrito. Por exemplo, mantendo uma força f constante nos motores lineares (torque) um veículo terá velocidade v em regiões planas, em regiões de declive terá v maior e em acíves terá v menor.

Ambos os protótipos possuem o mesmo modelo de simulação de propagação do fogo,

simplificado em relação ao modelo proposto em (PESSIN et al., 2007e). A integração entre os protótipos se dá através de um arquivo texto; após realizar a evolução, o protótipo responsável pela estratégia cria um arquivo com as posições de atuação que é lido na inicialização do protótipo de ação no combate. Este arquivo texto simula a comunicação entre o agente monitor e os agentes de combate. Assim, para os protótipos desenvolvidos para a dissertação, a única forma de comunicação implementada é a comunicação entre o agente monitor e os agentes de combate a incêndio. A comunicação entre as unidades de combate a incêndio, visando minimizar o trajeto percorrido, não foi implementada para o protótipo da Fase 2 desta dissertação, esta pode ser vista em (PESSIN, 2007); assim como a comunicação do *feedback* dos agentes de combate para o agente monitor e comunicação entre agentes com a finalidade de avisar atrasos; que são propostos como trabalhos futuros (Capítulo 6).

A descrição detalhada dos modelos de cada uma das fases de operação, bem como seus respectivos protótipos, serão apresentados nas próximas Seções.

Para o desenvolvimento do RoBombeiros (PESSIN et al., 2007e, 2007f, 2007b, 2007g, 2007c, 2007a, 2007d; OSÓRIO et al., 2007) foi realizado um estudo extensivo sobre incêndios em ambientes naturais. Foi visto que o vento e o tipo de vegetação são os parâmetros de maior influência na direção e intensidade do fogo (CE, 2006; BATISTA, 2004; LIF, 2006; KOPROSKI, 2005; MA, 2006). Dados sobre técnicas reais de operação, combate, equipamentos e monitoração de incêndios florestais retirados de (BATISTA, 2004; KOPROSKI, 2005; CE, 2006; LIF, 2006; PERERA; REMMEL, 2001; ANTUNES, 2000; CPTEC/INPE, 2007) mostraram que equipes de combate usualmente são formadas por 2, 4 ou 8 pessoas e que, para incêndios de grande porte, um dos métodos de combate indicado é a criação de aceiros mecânicos. O uso de aceiros químicos é, em se considerando incêndios de grande porte, ainda um tópico de pesquisa na engenharia florestal.

Como dito anteriormente, para incêndios em áreas com predomínio de vegetação herbácea, a criação de aceiros, em geral, faz uso de motoniveladoras (Figura 4.3), onde optamos por este tipo de método de combate no nosso modelo.

Para a simulação da vegetação e propagação do fogo, existe uma matriz oculta sob o terreno. Esta matriz possui, para cada área do terreno, o tipo de vegetação presente, assim, considerando orientação do vento, intensidade do vento e tipo de vegetação de uma área podemos construir a simulação de propagação do fogo. Como dito anteriormente, a velocidade de propagação respeita dados do modelo retirados de (KOPROSKI, 2005). Quanto ao vento, tanto a sua intensidade como a sua orientação podem ser geradas aleatoriamente ou configuradas a partir de dados parametrizados pelo usuário. O tempo de permanência do fogo em uma área é



Figura 4.3: Motoniveladora.

relacionado diretamente ao tipo da vegetação presente. Ambos os protótipos possuem um mapa que simula a integração das informações de vegetação, topografia e comportamento de fogo detalhados na Seção 2.4. A criação dos mapas teve como base cartas topográficas e o mapa de modelos de combustíveis florestais, que podem ser vistos nas Figuras 4.4(a) e 4.4(b).

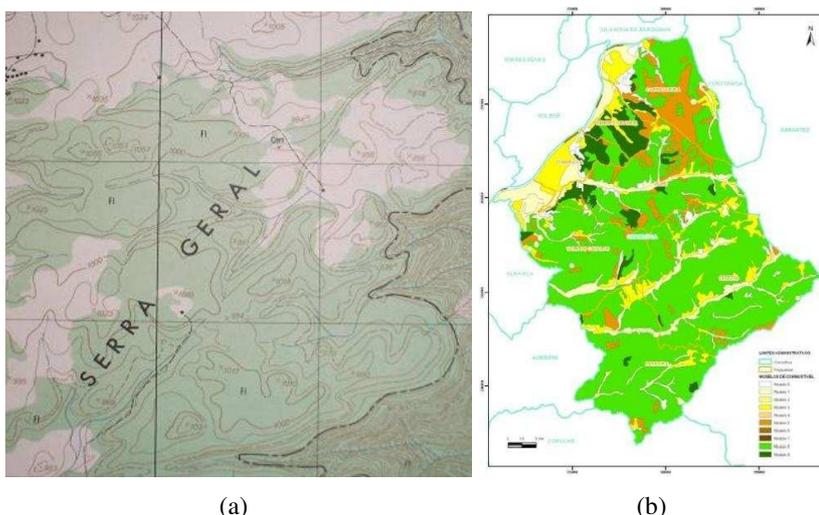


Figura 4.4: (a) Fragmento de carta topográfica de UTM (0578000, 6764000) até (0584000, 6770000) representando um quadrante de 6km x 6km; (b) Mapa de modelos de combustíveis (MA, 2006).

O controle da propagação do incêndio é feito através de uma matriz. Esta matriz, contém, para cada ponto do terreno:

- Quantidade de combustível: de -1 a 13 , onde os valores de 1 até 13 representam tipos de combustível e suas respectivas propriedades como velocidade de propagação e tempo de permanência. O valor 0 significa que a área já queimou e o valor -1 significa área onde foi criado o aceiro; o protótipo desenvolvido para a dissertação está implementado apenas com o Modelo 1 de combustível florestal, simplificado em relação ao descrito em

(PESSIN et al., 2007e).

- Variáveis lógicas de controle de visualização do fogo.

Esta matriz não possui nenhuma informação relacionada ao posicionamento dos agentes. O terreno possui coordenadas com dimensões de [15.000,15.000] e a matriz possui coordenadas com dimensões [600,600], assim, o tamanho de cada quadra é de [25,25] unidades.

O modelo de propagação de incêndio pode ser facilmente adaptado ou substituído por outra rotina de simulação de incêndio mais realista (considerando a umidade do ar e do solo, aclives e declives, etc). O importante é perceber que o modelo de AG proposto é capaz de otimizar a estratégia de combate independentemente do modelo de propagação de incêndio; o AG é capaz de otimizar os pontos chave de combate, que podem ser obtidos mesmo que a rotina de simulação de incêndio seja trocada. De um modo geral, podemos dizer que o *fitness* independe de parâmetros específicos da simulação do fogo, mas sim do resultado desta simulação. Mudar o cálculo da propagação, pode mudar a forma como o fogo se propaga, mas o AG vai otimizar a área salva, e por consequência, é capaz de achar soluções para diferentes situações onde o fogo se propaga de modo diferente.

4.2 Morfologia dos Robôs Móveis

Os robôs móveis foram desenvolvidos com a biblioteca de simulação de corpos rígidos articulados ODE, sendo que a representação gráfica dos robôs é idêntica ao objeto físico simulado existente no sistema. A Figura 4.5 apresenta o veículo desenvolvido. Para que seja possível realizar a navegação autônoma, o veículo possui um número adequado de sensores e atuadores. Com relação aos atuadores, cada veículo possui dois motores. Um motor angular (para o giro do volante) e um motor linear (para o torque) - Modelo de cinemática tipo Ackerman (DUDK; JENKIN, 2000). Com relação aos sensores, cada veículo possui um GPS, responsável pela obtenção da localização, uma bússola, necessária para a obtenção da orientação do veículo, e 5 sensores de distância (sonares simulados), que apresentam as características de capacidade de medir distâncias entre 15cm e 11m, como o *Polaroid 6500* (www.senscomp.com). Sensores e atuadores, como são dispositivos de hardware, apresentam erros e interferências que devem ser tratadas em nossa simulação; detalharemos as experiências com simulação de erros no decorrer do texto.

O cálculo do azimute (ângulo para o alvo) pelo veículo é realizado a partir da obtenção da posição atual (GPS) e da posição do alvo (recebida por mensagem), como mostra a Figura

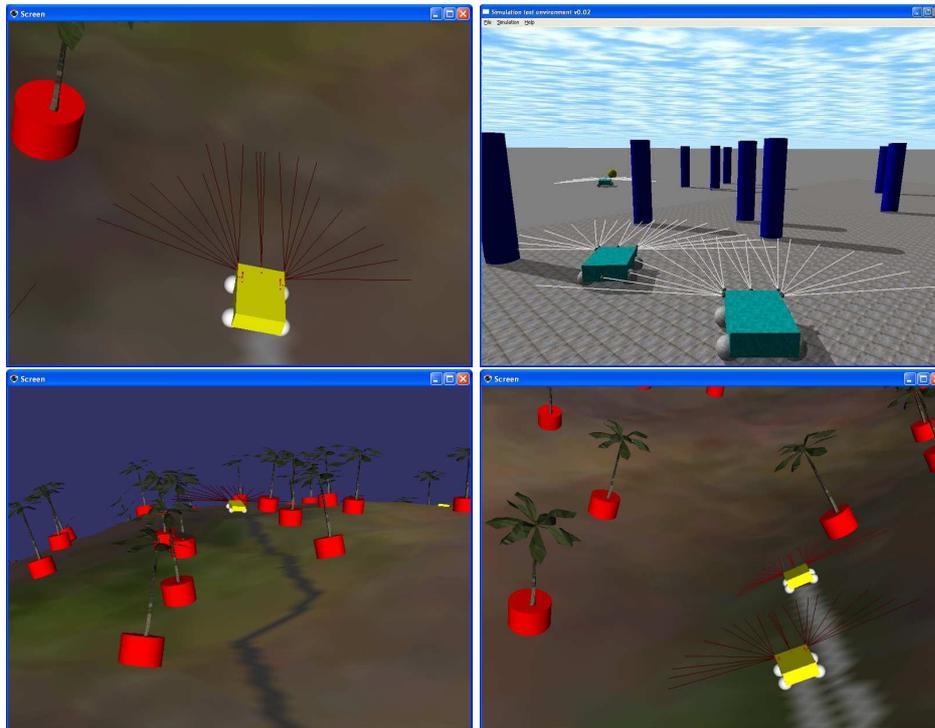


Figura 4.5: Exemplo de quatro cenas dos veículos desenvolvidos.

4.6.

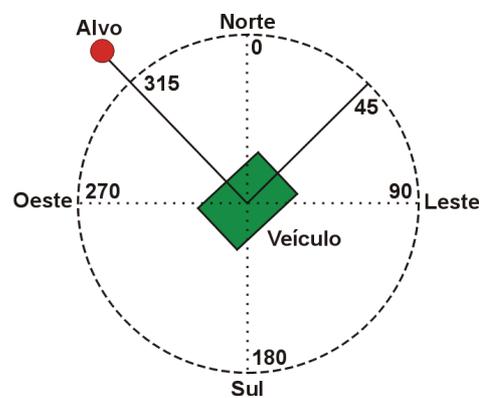


Figura 4.6: Bússola e GPS: responsáveis pela orientação, localização e obtenção do azimute.

Dada a existência das restrições físicas fornecidas pela biblioteca ODE, a única maneira de realizar o controle dos veículos é com a aplicação de forças em seus dois motores simulados. No caso da aplicação de forças erradas nos atuadores o veículo pode tombar.

Quanto ao controle de posicionamento, um sensor de localização (GPS) é implementado em cada agente. Para que este sensor respeitasse o funcionamento de um GPS real, realizamos um estudo de campo medindo os dados de um GPS *Garmin Etrex* (GARMIN, 2007) durante cerca de duas horas. A análise estatística sobre os dados coletados indicou erro médio de 18,6

metros para este modelo de GPS, assim, levamos em consideração este erro para o correto posicionamento dos agentes de combate. Considerando que cada agente possui seu próprio GPS, o tratamento deste erro é crucial na criação dos aceiros. Tratamos esta informação da seguinte maneira: o erro médio deste sensor durante o deslocamento é usado somado a distância de criação do aceiro e também somado ao final da área de criação. A importância de se considerar o erro do GPS se dá devido a equipe robótica adotar a técnica de criação de aceiros - áreas livre de vegetação que impedem a passagem do fogo - caso o erro do GPS não seja considerado, regiões sem tratamento (com vegetação) podem continuar a existir, assim o fogo pode passar por esta pequena região e continuar a devastar a área florestal. A operação confiável de combate a incêndios depende então da consideração do erro do GPS na criação de aceiros preferivelmente redundantes. Assim, para evitar falhas nos aceiros, o erro do GPS é somado ou diminuído de acordo do a posição de atuação no combate de cada agente, como mostra a Figura 4.7, gerando uma pequena área de sobreposição (uma “margem de segurança” devido ao erro do GPS).

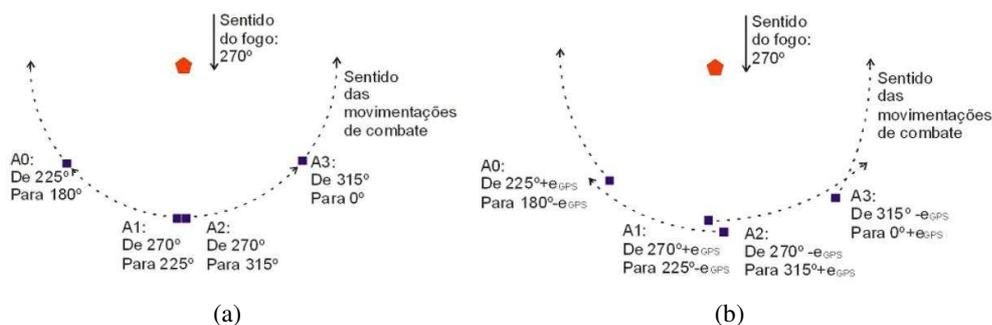


Figura 4.7: Trajetos de ação dos agentes de combate: (a) posicionamento teórico sem considerar erro no GPS; (b) posicionamento com redundância, considerando erro no GPS.

A morfologia do Robô Móvel Autônomo respeita os quatro aspectos principais que concretizam o projeto de um agente robótico, detalhados na Seção 2.2.2. A quantidade de sensores e atuadores, bem como o formato do veículo, foi desenvolvido tendo como base os trabalhos descritos nos Capítulos 2 e 3. Além disso, respeitamos, na definição da arquitetura de controle, todas as tarefas descritas na Seção 2.1.5.2. A simulação de erro dos outros sensores e atuadores será explorada na Seção 5.2.

4.3 Evolução de Estratégias

A evolução de estratégias está implementada com a biblioteca de programação de Algoritmos Genéticos GALib. Este protótipo busca otimizar as posições de atuação de um grupo de agentes de combate a incêndios. Para realizar a otimização do vetor de posições, a unidade

monitora deve:

- Saber a quantidade de agentes de combate disponíveis;
- Conhecer a velocidade de operação dos agentes de combate;
- Saber a posição inicial dos agentes de combate;
- Possuir capacidade de simular a propagação de incêndio.

Para ser capaz de simular a propagação de incêndio, a unidade monitora deve ser capaz de obter ainda:

- Coordenada inicial do incêndio;
- Sentido do vento;
- Direção do vento;
- Cópia simplificada do mapa (terreno e vegetação).

Após realizar a evolução, a unidade monitora tem capacidade de enviar mensagens de comando para ativar as unidades de combate. Ao final da evolução, a unidade monitora deve permitir a otimização dos parâmetros:

- Posição chave inicial no combate, para cada agente do time (ponto de início da criação de aceiro);
- Posição chave final no combate de cada agente do time (ponto final da criação de aceiro).

Propomos inicialmente um genoma onde as posições de cada agente eram independentes. A execução de algumas simulações não apresentou nenhum resultado satisfatório, assim, propomos um novo genoma onde a posição final de um agente é a posição inicial do próximo, como apresentam as tabelas 4.1 e 4.2. O tipo de atuação proposto (criação de aceiro) necessita que cada agente inicie sua operação exatamente na posição onde o agente anterior terminou a sua atuação. Assim, o uso de um genoma que considera a posição final de um agente com o a inicial do próximo é, de certa forma, uma simplificação para a simulação com o GA.

As posições de atuação são calculadas aplicando no melhor indivíduo resultante da evolução as Equações 4.1 e 4.2.

$$x_f = x_a + r_i \times \cos(a_i) \quad (4.1)$$

Tabela 4.1: Genoma desenvolvido para um grupo de 4 agentes.

Gene	Função	Valor Mínimo	Valor Máximo
0	Posição inicial do agente 0	0,000	360,000
1	Posição final do agente 0 e inicial do agente 1	0,000	360,000
2	Posição final do agente 1 e inicial do agente 2	0,000	360,000
3	Posição final do agente 2 e inicial do agente 3	0,000	360,000
4	Posição final do agente 3	0,000	360,000
5	Raio inicial do agente 0	10,000	100,000
6	Raio final do agente 0 e inicial do agente 1	10,000	100,000
7	Raio final do agente 1 e inicial do agente 2	10,000	100,000
8	Raio final do agente 2 e inicial do agente 3	10,000	100,000
9	Raio final do agente 3	10,000	100,000

Tabela 4.2: Genoma desenvolvido para um grupo de 2 agentes.

Gene	Função	Valor Mínimo	Valor Máximo
0	Posição inicial do agente 0	0,000	360,000
1	Posição final do agente 0 e inicial do agente 1	0,000	360,000
2	Posição final do agente 1	0,000	360,000
3	Raio inicial do agente 0	10,000	100,000
4	Raio final do agente 0 e inicial do agente 1	10,000	100,000
5	Raio final do agente 1	10,000	100,000

$$y_f = y_a + r_i \times \sin(a_i) \quad (4.2)$$

Onde (x_f, y_f) é a coordenada da posição final dos robôs, (x_a, y_a) é a coordenada da posição inicial do incêndio, r_i é o raio (genes 5 a 9 na Tabela 4.1) e a_i é o ângulo (genes 0 a 4 na Tabela 4.1). O raio, bem como o ângulo, são específicos para cada operação de cada robô (coordenada inicial e final de criação de aceiro).

Com relação ao Algoritmo Genético, utilizamos, na GALib, o modelo com populações sobrepostas (*GASteadyStateGA*) proposto por (DEJONG, 1975). Para reduzir o espaço de busca utilizamos alelos (*GARealAlleleSetArray*) que limitam o conjunto de valores gerados para cada atributo, assim, o raio de atuação é de no mínimo 10 unidades de medida e de no máximo 100 unidades de medida e o raio fica entre 0 e 360. Foram adotados genomas do tipo real, otimizados para operações com números de ponto flutuante (*GARealGenome*). Análises comparativas envolvendo representação binária e representação com ponto flutuante (*GARealGenome*) apresentaram que a representação em ponto flutuante apresenta vantagens significativas principalmente com relação a precisão e velocidade de convergência (GOLDBERG, 1991; MICHALEWICZ, 1996). O esquema de seleção adotado (*Selector*) foi o *Stochastic Remainder Sampling Selector* (*GASRSSSelector*), que segundo (GOLDBERG, 1989) possui um desempenho superior ao esquema da roleta (*GARouletteWheelSelector*). Outros parâmetros genéticos utilizados nas

nossas experiências serão detalhados na Seção 5.1.

Como dito anteriormente, a vantagem de se utilizar GA para o ajuste das posições é que eles não necessitam de informações locais para a correção dos erros, ou seja, eles não necessitam de uma base com os dados de treinamento. No caso do combate de incêndios, estas informações seriam muito difíceis de serem obtidas. O aprendizado ocorre através da interação das possíveis soluções com o ambiente e o desenrolar da simulação, o que está de acordo com princípios descritos na Seção 2.2.2.

4.3.1 Escolha do *Fitness*

O Algoritmo Genético tem sua otimização guiada pela função *fitness*. No nosso modelo, esta função tem relação com a área de vegetação salva do fogo e com a taxa de utilização das unidades de combate (robôs bombeiros).

Assim, o *fitness* desenvolvido acumula os seguintes valores finais de cada simulação:

- Total de área queimada: busca minimizar a área queimada;
- Total de área com aceiro: busca minimizar a área de trabalho dos robôs, evitando que sejam criados aceiros em áreas onde o fogo não atingiria;
- Erro médio absoluto: busca minimizar a diferença entre a média geral de aceiros úteis em relação ao aceiro útil de cada indivíduo, assim, o tamanho das áreas de trabalho tende a se equalizar.

Otimizar o *fitness*, no nosso caso, é minimizar o seu valor, ou seja, quanto menor o *fitness* final, menos vegetação foi queimada, menos aceiro foi criado, e menor é a diferença entre o tamanho dos aceiros. Buscamos, nas simulações, minimizar o valor do *fitness*. Deste modo, realizamos a otimização dos indivíduos da população (conjunto de genes) que representam os valores a serem adotados como sendo a estratégia de atuação no combate. Estes valores (genes) apresentam os pontos chave para onde os robôs móveis devem se deslocar.

Na Seção seguinte será apresentado o sistema de controle de navegação dos RMAs, que permite que os robôs realizem o deslocamento definido pela unidade monitora.

4.4 Controle de Navegação

O controle inteligente de navegação foi desenvolvido com uma Rede Neural Artificial do tipo *Multi Layer Perceptron* (MLP) treinada com o algoritmo de aprendizagem *Resilent Backpropagation* (RProp). Esta RNA foi desenvolvida e treinada no SNNS (SNNS, 2006). O controle inteligente realizado pela RNA permite navegação e desvio de obstáculos em um ambiente dinâmico, usando apenas informações disponíveis localmente, obtidas pelos sensores dos veículos.

Esta RNA terá como entradas as seguintes informações:

- Orientação (ângulo) do veículo, em relação ao plano (x, y) , obtido através de uma bússola simulada;
- Azimute (ângulo para o alvo) do veículo, obtido através da bússola, do GPS e da mensagem contendo a coordenada do alvo;
- Valores dos sensores de distância (sonares).

As saídas desta RNA são:

- Força a aplicar no motor angular (giro da barra de direção, de -1.5 a 1.5);
- Força a aplicar no motor linear (torque, de 0.0 a 6.0);

Experiências iniciais mostraram que é imprescindível que o veículo diminua a velocidade em caso de curvas, principalmente quando bruscas. Usamos uma única RNA que controla tanto o giro como o torque, sendo capaz de orientar o veículo e realizar a navegação sem a necessidade de um controle humano ou de uma codificação prévia de um autômato que informe quando desviar de um obstáculo ou como navegar.

A base de dados de treinamento foi obtida a partir de um sistema simples baseado em regras. Foi realizado um total de 32 simulações para obter dados. Um vídeo¹ exemplo de uma simulação de obtenção de dados pode ser obtido na página web² do projeto, onde estão disponibilizados outros materiais complementares referentes a este projeto. A Figura 4.8 apresenta as posições de início e fim das simulações usadas para obtenção da base de dados de treino da RNA. As 16 setas apresentam as posições de partida e o centro do gráfico apresenta a posição de

¹Vídeo exemplo de simulação para obtenção de dados para treino e validação da RNA disponível em <http://pessin.googlepages.com/GravaLogParaRNA1.wmv>

²Página de internet do projeto: <http://pessin.googlepages.com/robombeiros>

chegada. Foram realizadas 16 simulações a fim de se obter apenas dados de orientação; outras 8 simulações foram realizadas para obter dados de desvio, usando chão plano e um ambiente com 10% de ocupação por árvores; ainda, outras 8 simulações foram realizadas para obter dados de desvio usando chão irregular e 10% de ocupação por árvores. A base de dados final obtida apresentou 4.985 registros, sendo dividida em 70% para treino e 30% para validação.

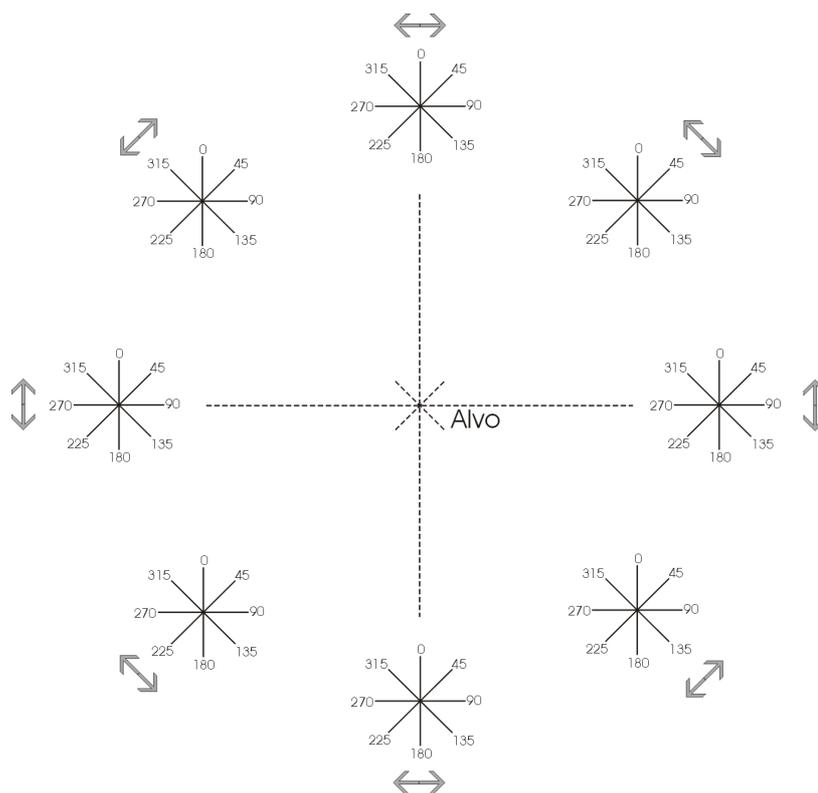


Figura 4.8: Posições para coleta de dados.

As regras criadas para as simulações e que foram utilizadas na extração de dados para o treino da RNA foram as seguintes:

- Para a velocidade: velocidade padrão igual a 6 unidades de velocidade; quando qualquer sensor de distância percebe um obstáculo a menos de 9 unidades de distância, a velocidade cai a 3 unidades de velocidade; e, quando qualquer sensor de distância percebe um obstáculo a menos de 3 unidades de distância, a velocidade cai a 1,5 unidades de velocidade.
- Para a orientação: caso a diferença entre orientação do veículo e o azimute for maior que 25°, então giro da barra de direção recebe 0.5; caso a diferença entre orientação do veículo e o azimute for maior que 5° e menor que 25°, então giro da barra de direção recebe 0.2; caso a diferença entre orientação do veículo e o azimute for maior que 0,1° e menor que

5°, então giro da barra de direção recebe 0.1; caso a diferença entre orientação do veículo e o azimute for menor que 0,1°, então giro da barra de direção recebe 0.0.

- Para o desvio: foram implementadas 24 regras que tem relação com proximidade de obstáculos detectada pelos sensores e respectivos comandos para giro da barra de direção; o sistema prioriza o desvio antes da regra de orientação e prioriza os sensores frontais antes dos laterais, aumentando a importância de acordo com a proximidade percebida. As regras completas podem ser observadas ³ no código fonte desenvolvido.

Uma vez obtidos os dados para treino e validação, partimos para definição da topologia da RNA. Testamos 6 diferentes topologias de RNA, com 4, 9, 18, 24, 30 e 36 neurônios na camada oculta. A análise e escolha da melhor RNA foi feita através do Erro Médio Absoluto (*Mean Absolute Error* - MAE) - Equação 4.3 e do Erro Médio Quadrático (*Mean Squared Error* - MSE) - Equação 4.4. O MSE é fornecido pela própria ferramenta de simulação de redes neurais e apresenta o erro geral da rede, o MAE é calculado por uma rotina desenvolvida a parte e fornecerá valores próprios para cada neurônio de saída da RNA.

$$E_{MAE} = \frac{1}{n} \sum_{i=1}^n |E_i - O_i| \quad (4.3)$$

$$E_{MSE} = \frac{1}{n} \sum_{i=1}^n |E_i - O_i|^2 \quad (4.4)$$

Como dito anteriormente, o uso de RNA no controle de navegação é justificado por serem as técnicas mais robustas em relação a situações novas e inesperadas, por possuírem alto grau de generalização e por fornecem uma arquitetura em conformidade com os princípios descritos na Seção 2.2.2. Além disso, como o desenvolvimento de regras de controle é uma operação bastante complexa, podemos criar uma base de treino através da obtenção de dados de um veículo operado manualmente, para posterior treino e teste da RNA. A Figura 4.9 apresenta cenas de simulações realizadas para obtenção de dados usando o sistema de regras.

O treino e a validação da RNA serão detalhados na Seção 5.2.1; uma vez treinada e selecionada a melhor RNA, esta é convertida em um programa em C para ser incorporada ao simulador e passa a controlar o Robô Móvel Autônomo em sua tarefa de navegação. Em um robô real, não simulado, a RNA seria utilizada no sistema de controle embarcado do veículo.

³As regras completas podem ser observadas no código fonte desenvolvido, disponível em <http://pessin.googlepages.com/RoBombeiros0.66.8GeraLOG.rar>

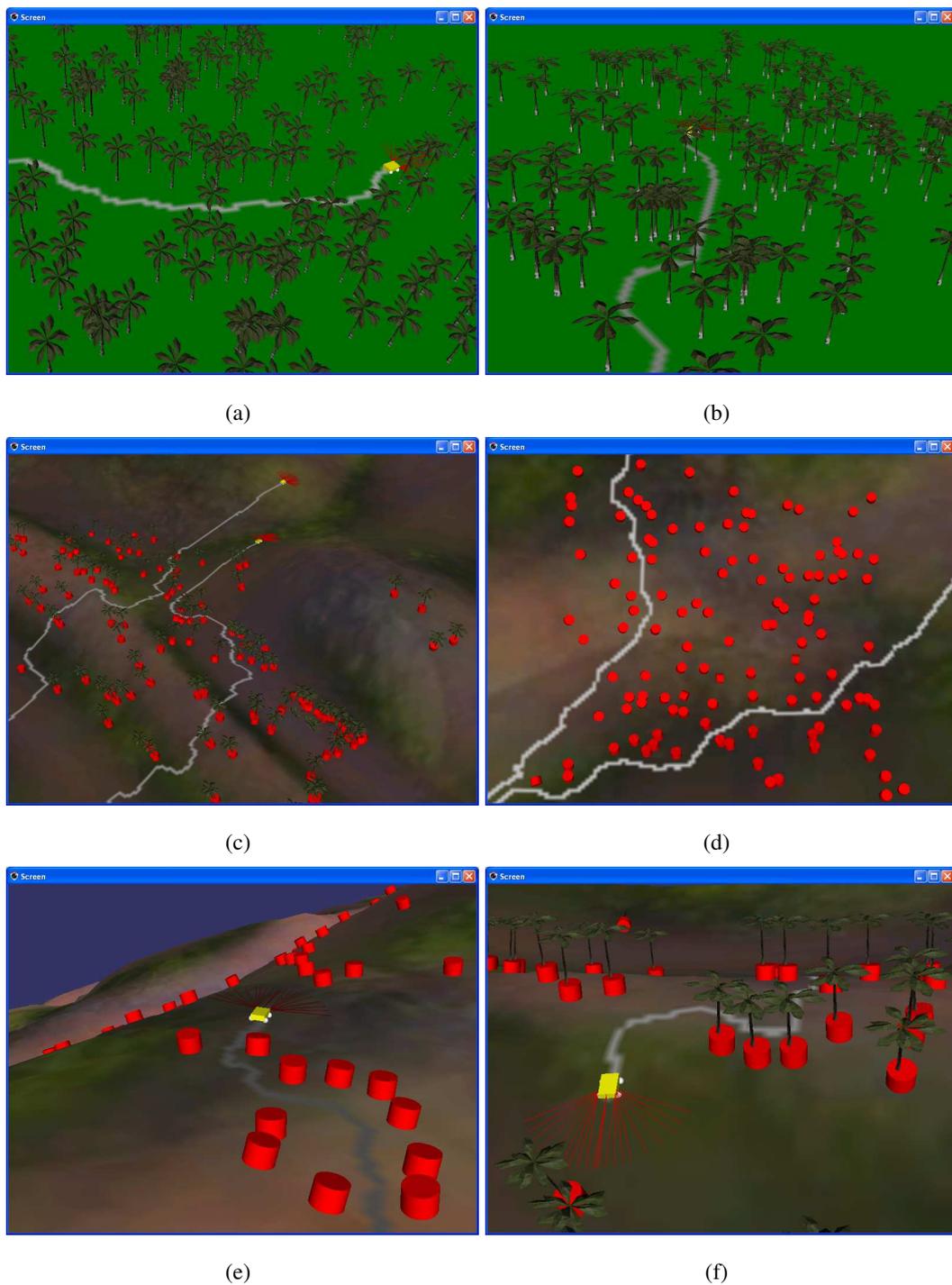


Figura 4.9: Cenas de simulações realizadas para obtenção de dados usando o sistema de regras. Todas as Figuras apresentam terreno irregular, com exceção de (a) e (b) que apresentam chão plano.

4.5 Considerações Finais

Este Capítulo apresentou o modelo do sistema multi-robótico proposto nesta dissertação, apresentando a descrição detalhada do sistema proposto, onde foi visto que uma equipe de agentes autônomos trabalha cooperativamente a fim de realizar com sucesso a identificação e o combate a incêndios em áreas florestais, em um ambiente que suporta características fundamentais para a simulação realística da operação, com terrenos irregulares, processos naturais e restrições físicas na criação e uso de robôs móveis. Neste ambiente de simulação é realizada a operação multi-agente propriamente dita, que depende essencialmente de duas etapas: planejamento e ação; cada uma desenvolvida com um protótipo específico.

A morfologia dos robôs móveis respeita os quatro aspectos principais que concretizam o projeto de um agente robótico, detalhados na Seção 2.2.2. A quantidade de sensores e atuadores, bem como o formato do veículo, foi desenvolvido tendo como base os trabalhos descritos nos Capítulos 2 e 3. Além disso, respeitamos, na definição da arquitetura de controle, todas as tarefas descritas na Seção 2.1.5.2. Mostramos como foi realizado o tratamento do erro do GPS, sendo a simulação de erro dos outros sensores e atuadores explorada na Seção 5.2.

Descrevemos o modelo proposto de GA e de RNA, desenvolvidos tendo como base os trabalhos descritos nos Capítulos 2 e 3. Assim, no próximo Capítulo serão apresentados experimentos que demonstram: a eficiência das técnicas de RNA, aplicadas no controle de navegação dos agentes robóticos; e a eficiência das técnicas de GA, aplicadas na evolução de estratégias de posicionamento de grupos robóticos.

5 EXPERIMENTOS E RESULTADOS

O Capítulo anterior apresentou o modelo do sistema multi-robótico proposto nesta dissertação, apresentando a descrição detalhada do sistema proposto, onde foi visto que uma equipe de agentes autônomos trabalha cooperativamente a fim de realizar com sucesso a identificação e o combate a incêndios em áreas florestais, em um ambiente que suporta características fundamentais para a simulação realística da operação, com terrenos irregulares, processos naturais e restrições físicas na criação e uso de robôs móveis. Neste ambiente de simulação é realizada a operação multi-agente propriamente dita, que depende essencialmente de duas etapas: planejamento e ação; cada uma desenvolvida em um protótipo específico.

Neste Capítulo são descritos os diversos experimentos realizados com os protótipos desenvolvidos. Iniciamos descrevendo os experimentos e as análises sobre o uso do Algoritmo Genético proposto para realizar a evolução de estratégias de posicionamento de grupos robóticos. Em seguida detalhamos o treino e teste sobre as diversas topologias de RNA propostas. Após, apresentamos os experimentos e as análises com o uso da RNA escolhida para o controle inteligente de navegação dos agentes de combate a incêndio, finalizando com experimentos de navegação onde é aplicado um ruído simulado aos sensores e atuadores, visando demonstrar a robustez do sistema desenvolvido.

5.1 Evolução de Estratégias

Como descrito na Seção 4.3, a evolução de estratégias está implementada com a biblioteca de programação de Algoritmos Genéticos GALib. Este protótipo busca otimizar as posições de atuação de um grupo de agentes de combate a incêndios. Para realizar a otimização do vetor de posições, a unidade monitora deve: saber a quantidade de agentes de combate disponíveis; conhecer a velocidade de operação dos agentes de combate; saber a posição inicial dos agentes de combate; e possuir capacidade de simular a propagação de incêndio. Para ser capaz de simular a propagação de incêndio, a unidade monitora deve ser capaz de obter: coordenada inicial do incêndio; sentido do vento; direção do vento; e uma cópia simplificada do mapa (terreno e

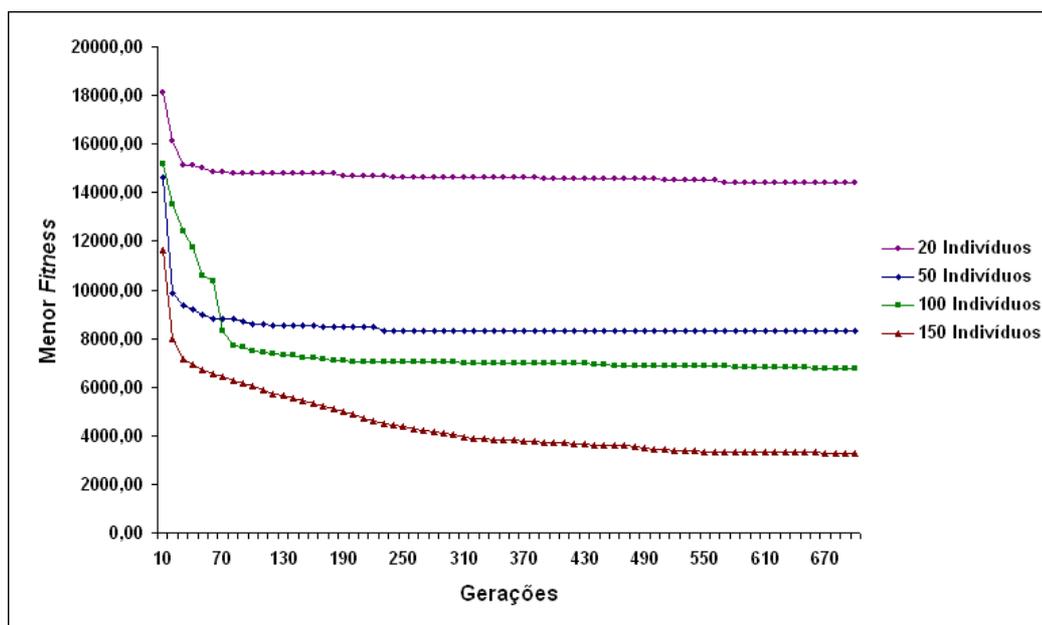
vegetação). Após realizar a evolução, a unidade monitora tem capacidade de enviar mensagens de comando para ativar as unidades de combate. Ao final da evolução, a unidade monitora deve permitir a otimização dos parâmetros: coordenada de início da criação de aceiro de cada agente de combate; e coordenada final da criação de aceiro de cada agente de combate.

Dois tipos de mutação foram utilizados em diferentes simulações com o Algoritmo Genético desenvolvido; inicialmente foi utilizada mutação do tipo Gaussiana (*GARealGaussianMutator*) e em seguida foi utilizada mutação do tipo Uniforme *GARealUniformMutator*. Na mutação do tipo Gaussiana, a partir da identificação de um genoma razoável, são realizados preferencialmente pequenos ajustes até o final das gerações (genes são substituídos por números aleatórios de uma distribuição gaussiana). A mutação Uniforme substitui um gene por um número aleatório qualquer, levando a mudanças mais bruscas nos valores a serem otimizados, entretanto, ambas apresentaram resultados satisfatórios.

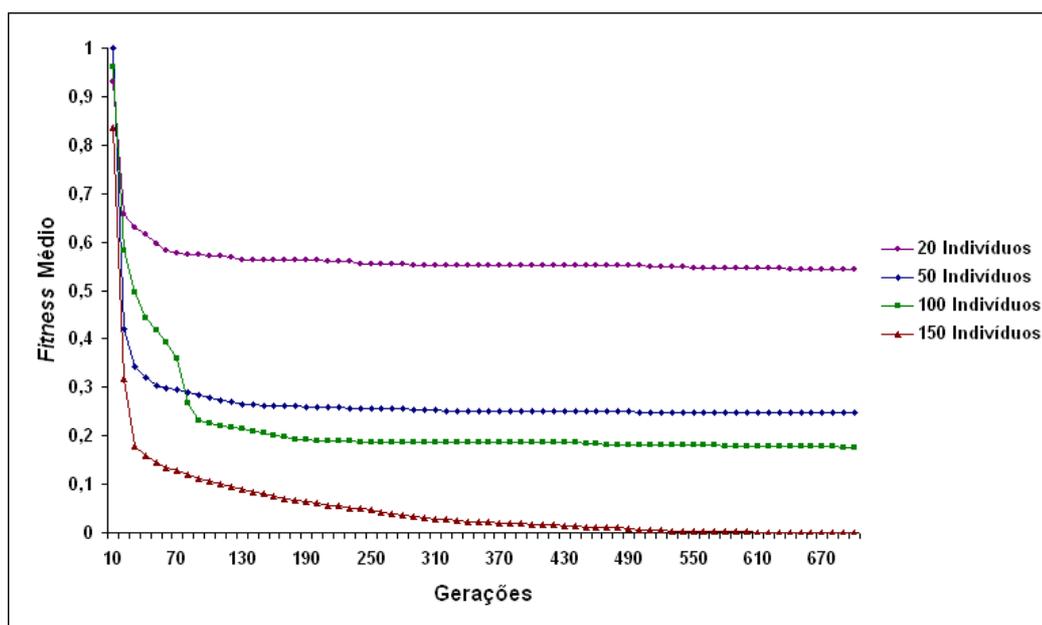
O *fitness*, apresentado na Seção 4.3.1, é o resultado da soma de três valores: (i) Total de área queimada: busca minimizar a área queimada; (ii) Total de área com aceiro: busca minimizar a área de trabalho dos robôs; e (iii) Erro médio absoluto: busca minimizar a diferença entre a média geral de aceiros úteis em relação ao aceiro útil de cada indivíduo, assim, o tamanho das áreas de trabalho tende a se equalizar entre os diferentes robôs. Destes valores, o que mais impacta no valor final é o parâmetro *Total de Área Queimada*, assim, os gráficos apresentam uma queda brusca do *Fitness* quando a simulação já é capaz de apagar o incêndio, a partir disso, o GA realiza a otimização dos outros valores impactantes no *fitness* final.

Foram realizadas simulações com 2 e 4 agentes. Estas simulações demonstram que é possível otimizar adequadamente o posicionamento dos grupos robóticos, obtendo soluções satisfatórias para o problema proposto. Buscamos com estas simulações verificar também qual a quantidade necessária de indivíduos na população e qual a quantidade de gerações para obtermos resultados satisfatórios. Realizamos diversas simulações a fim de observar o gráfico de evolução da simulação através do valor do *fitness*.

O gráfico de evolução da simulação através do valor do *fitness*, para quatro agentes de combate, pode ser visto na Figura 5.1, este gráfico apresenta a média de 3 simulações, usando diferentes quantidades de indivíduos. Estas simulações foram realizadas com mutação Gaussiana, com taxa de *crossover* igual a 90% e com taxa de mutação igual a 10%. A propagação do fogo considerou vento no sentido Leste-Oeste (270°) e velocidade relativas: como sendo vento de 7km/h; velocidade de navegação dos agentes de 35km/h; e sendo os agentes de combate posicionados em uma base de operação com distância de cerca de 2km do ponto inicial do incêndio.



(a)



(b)

Figura 5.1: Evolução do *fitness* de acordo com o número de gerações para 4 agentes de combate e diferentes quantidades de indivíduos: (a) Menor *fitness* de toda população, em valor absoluto; (b) Média do *fitness* da população, em valor normalizado.

O gráfico apresentado na Figura 5.1 mostra que o melhor *fitness* é obtido com uma quantidade de indivíduos igual a 150; e que, entre 400 e 700 gerações, a otimização do *fitness* praticamente estabiliza. Os melhores indivíduos resultantes das simulações de evolução de

estratégia, cujos parâmetros foram descritos acima podem ser vistos na Tabela 5.1.

Tabela 5.1: Melhores indivíduos resultantes de 3 simulações para 4 agentes; utilizando no GA 150 indivíduos e 700 gerações.

Simulação	Graus					Distância				
1	225,72	199,26	174,12	155,20	136,38	27,35	30,45	33,94	38,08	35,96
2	224,89	200,77	177,86	160,68	136,64	28,05	29,75	30,58	31,12	33,24
3	134,82	142,65	172,33	196,14	244,84	54,94	42,95	35,84	30,37	26,33

O resultado satisfatório da evolução, para o experimento descrito acima, com 150 indivíduos e 700 gerações (o melhor indivíduo, como visto na Tabela 5.1) é selecionado como estratégia vencedora e deve ser passado ao módulo de ação de combate. O indivíduo não é passado ao módulo de ação no formato resultante da evolução; são passadas as coordenadas de atuação de cada agente, calculadas com as Equações 4.1 e 4.2 de acordo com o especificado na Tabela 4.1. A Figura 5.2 apresenta a visualização 2D do posicionamento de atuação; esta opção de visualização está desenvolvida junto ao protótipo responsável pela evolução (GAlib+SDL).

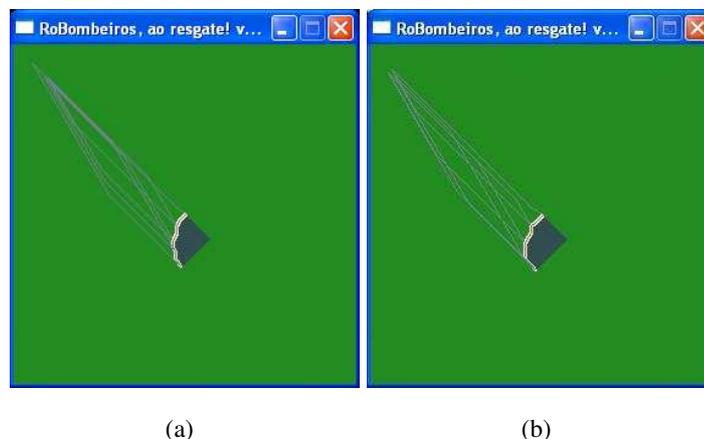


Figura 5.2: Resultado de duas evoluções satisfatórias, para 4 agentes de combate, utilizando os parâmetros citados acima. A zona escura é a área consumida pelo fogo, que foi contido pelo aço criado pelos 4 robôs.

Resultados não satisfatórios da evolução, para o experimento descrito acima, foram os que utilizaram 20, 50 e 100 indivíduos e 700 gerações. As Figuras 5.3, 5.4 e 5.5 apresentam a visualização da aplicação do melhor indivíduo de cada uma destas simulações (que não alcançaram um resultado plenamente satisfatório).



(a)

(b)

Figura 5.3: Resultado de duas evoluções não satisfatórias, para 4 agentes de combate, utilizando 20 indivíduos e 700 gerações. O fogo não é contido pelo aceiro em (b).



(a)

(b)

Figura 5.4: Resultado de duas evoluções não satisfatórias, para 4 agentes de combate, utilizando 50 indivíduos e 700 gerações. O aceiro criado não está adequadamente bem distribuído entre os robôs.



(a)

(b)

Figura 5.5: Resultado de duas evoluções não satisfatórias, para 4 agentes de combate, utilizando 100 indivíduos e 700 gerações. Em (b) a área do aceiro é excessivamente grande em relação ao que seria necessário.

A Figura 5.6 apresenta a aplicação do resultado satisfatório da evolução, para o melhor

experimento descrito acima, com 150 indivíduos e 700 gerações no protótipo de combate 3D. Podemos ver que os agentes de combate cercam perfeitamente o fogo, bem como criam os aceiros de forma satisfatória. A velocidade de propagação do fogo, para esta visualização, foi diminuída arbitrariamente, já que não desenvolvemos neste protótipo a capacidade de envio de avisos de atrasos para o agente monitor; que ocorrem dado que a navegação é em terreno irregular e necessita que sejam feitos desvios de obstáculos, o que acaba atrasando o deslocamento dos robôs.

É importante salientar que nas simulações apresentadas nas Figuras 5.3, 5.4 e 5.5, apenas na Figura 5.3(b) os robôs não foram capazes de conter o incêndio.

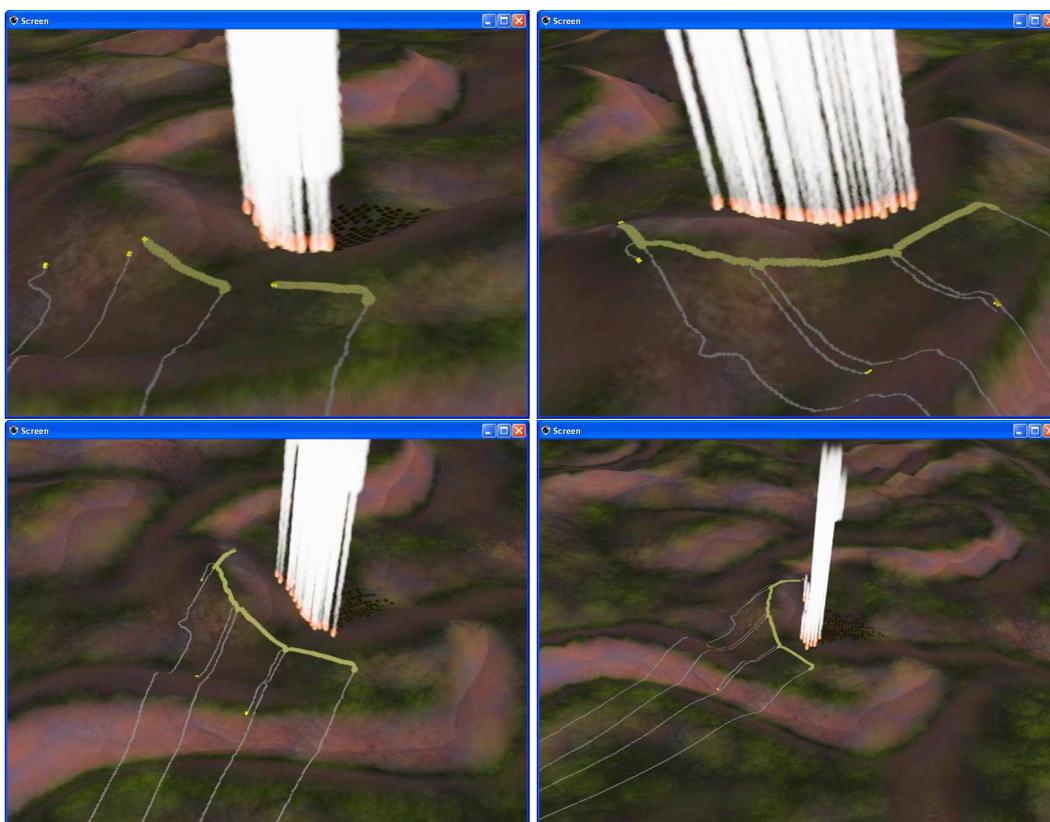
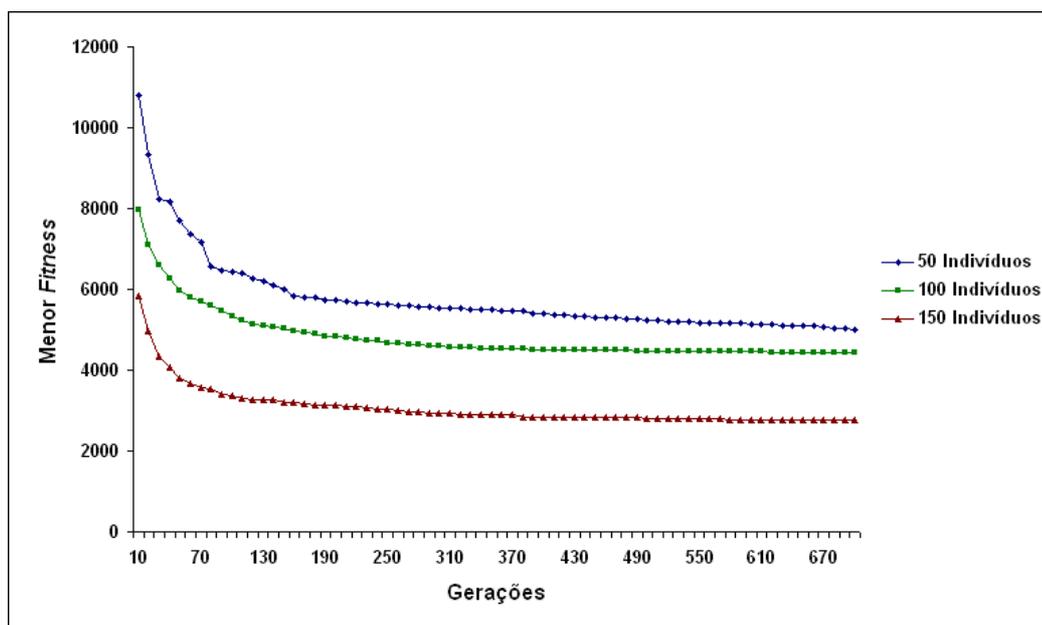
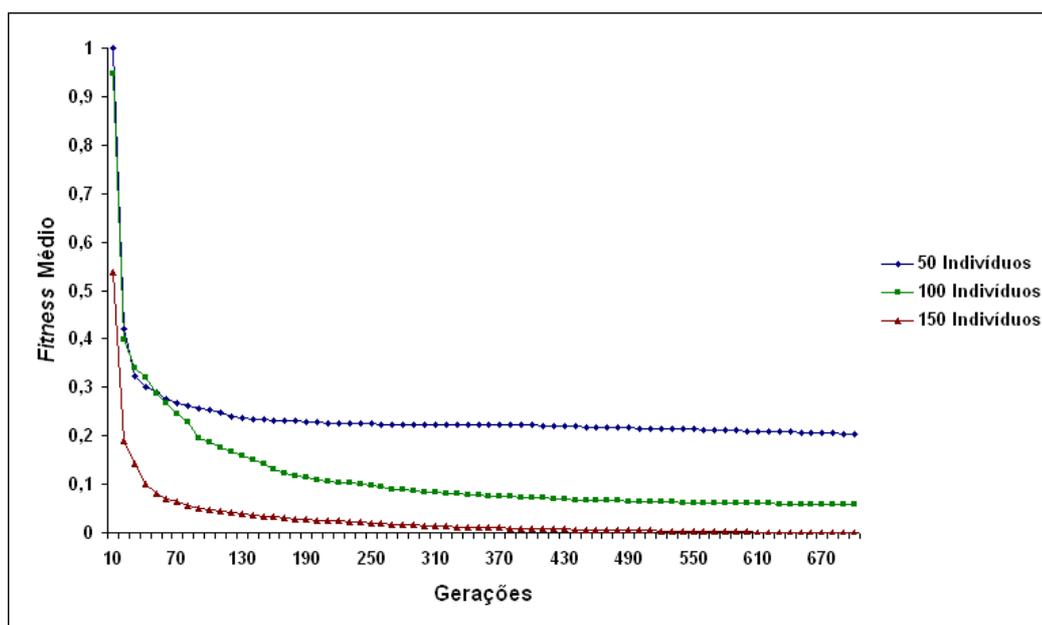


Figura 5.6: Quatro cenas de resultados satisfatórios de evoluções de estratégias usando o GA com 150 indivíduos e 700 gerações.



(a)



(b)

Figura 5.7: Evolução do *fitness* de acordo com o número de gerações para 2 agentes de combate e diferentes quantidades de indivíduos: (a) Menor *fitness* de toda a população, em valor absoluto; (b) Média do *fitness* da população, em valor normalizado.

O gráfico de evolução da simulação através do valor do *fitness*, para dois agentes de combate, pode ser visto na Figura 5.7, este gráfico apresenta a média de 3 simulações, usando diferentes quantidades de indivíduos. Estas simulações foram realizadas com mutação Gaus-

siana, com taxa de *crossover* igual a 90% e com taxa de mutação igual a 10%. A propagação do fogo considerou vento no sentido Leste-Oeste (270°) e velocidade relativas: como sendo vento de 7km/h; velocidade de navegação dos agentes de 35km/h; e sendo os agentes de combate posicionados em uma base de operação com distância de cerca de 2km do ponto inicial do incêndio.

O gráfico apresentado na Figura 5.7 mostra que o melhor *fitness* é obtido com uma quantidade de indivíduos igual a 150; e que, entre 400 e 700 gerações, a otimização do *fitness* praticamente estabiliza. Os melhores indivíduos resultantes das simulações de evolução de estratégia, cujos parâmetros foram descritos acima podem ser vistos na Tabela 5.2.

Tabela 5.2: Melhores indivíduos resultantes de 3 simulações para 2 agentes; utilizando no GA 150 indivíduos e 700 gerações.

Simulação	Graus			Distância		
1	226,82	179,97	124,98	24,92	28,34	41,92
2	228,47	176,93	136,38	25,11	28,22	34,98
3	224,09	175,62	136,02	28,86	29,60	32,66

O resultado satisfatório da evolução, para o experimento descrito acima, com 150 indivíduos e 700 gerações (o melhor indivíduo, como visto na Tabela 5.2) é selecionado como estratégia vencedora e deve ser passado ao módulo de ação de combate. O indivíduo não é passado ao módulo de ação no formato resultante da evolução; são passadas as coordenadas de atuação de cada agente, calculadas com as Equações 4.1 e 4.2 de acordo com o especificado na Tabela 4.2. A Figura 5.8 apresenta a visualização 2D do posicionamento de atuação; esta opção de visualização está desenvolvida junto ao protótipo responsável pela evolução (GALib+SDL).

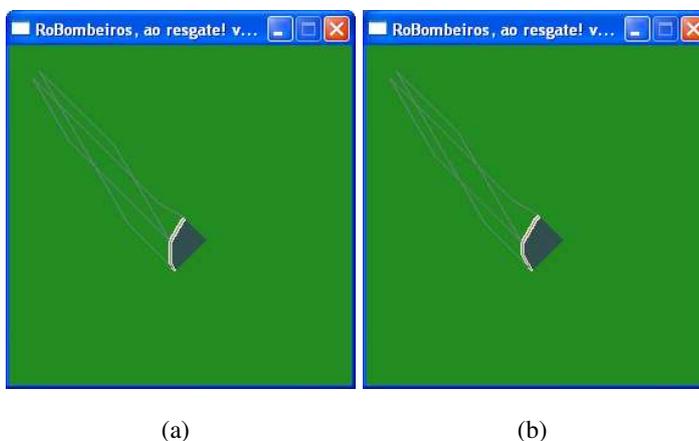


Figura 5.8: Resultado de duas evoluções satisfatórias, para 2 agentes de combate, utilizando os parâmetros citados acima.

A Figura 5.9 apresenta a aplicação do resultado satisfatório da evolução, para o experimento descrito acima, com 150 indivíduos e 700 gerações no protótipo de combate 3D. Podemos ver que os agentes de combate cercam o fogo, bem como criam os aceiros de forma satisfatória (que evitam a propagação do incêndio). A velocidade de propagação do fogo, para esta visualização, foi diminuída arbitrariamente, já que não desenvolvemos neste protótipo a capacidade de avisos de atrasos.

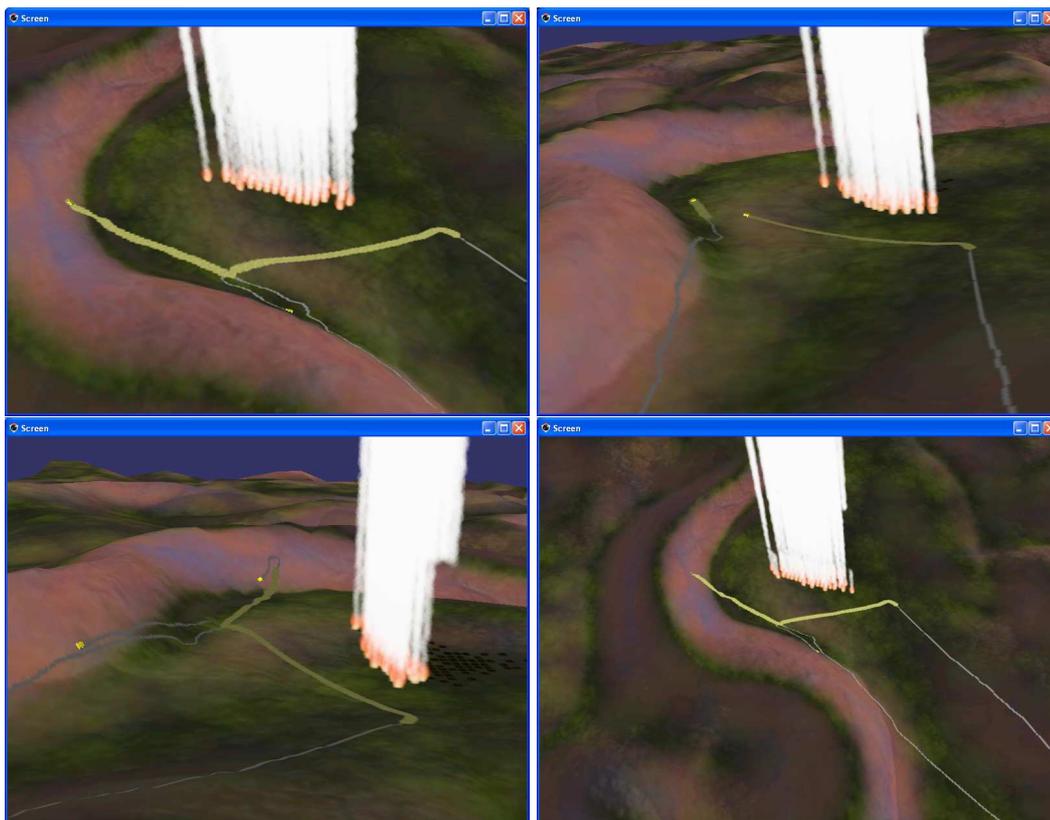


Figura 5.9: Quatro cenas de resultados satisfatórios de evoluções de estratégias usando o GA com 150 indivíduos e 700 gerações.

Usando mutação Uniforme, o gráfico de evolução da simulação através do valor do *fitness*, para quatro agentes de combate, pode ser visto na Figura 5.10, este gráfico apresenta a média de 3 simulações, usando 150 indivíduos. Estas simulações foram realizadas com taxa de *crossover* igual a 90% e com taxa de mutação igual a 10%. A propagação do fogo considerou vento no sentido Leste-Oeste (270°) e velocidades: como sendo vento de 7km/h ; velocidade de navegação dos agentes de 35km/h ; e sendo os agentes de combate posicionados em uma base de operação com distância de cerca de 2km do ponto inicial do incêndio.

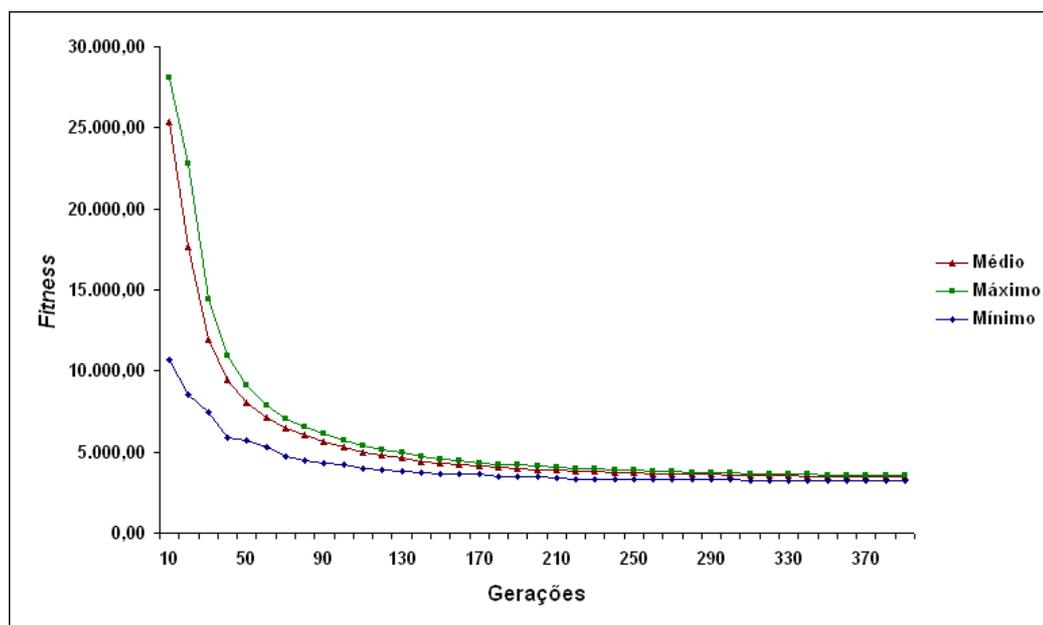


Figura 5.10: Evolução do *fitness* máximo, médio e mínimo, em valor absoluto, de acordo com o número de gerações para 4 agentes de combate; GA com 150 indivíduos e mutação Uniforme.

O gráfico apresentado na Figura 5.10 mostra que existe convergência entre as curvas de *fitness* máximo, médio e mínimo; utilizamos, neste experimento, 400 e não mais 700 gerações, como nos dois experimentos descritos anteriormente. Os melhores indivíduos resultantes das simulações de evolução de estratégia, cujos parâmetros foram descritos acima podem ser vistos na Tabela 5.3.

Tabela 5.3: Melhores indivíduos resultantes de 5 simulações para 4 agentes; utilizando no GA 150 indivíduos, 400 gerações e mutação do tipo Uniforme.

Simulação	Graus					Distância				
	1	133,41	151,68	171,67	196,88	225,49	41,10	42,63	36,22	31,92
2	227,16	204,77	176,16	150,40	134,81	29,67	31,84	31,89	34,89	36,18
3	229,15	196,31	174,89	154,50	135,59	26,00	26,63	32,24	36,93	36,40
4	133,99	154,72	179,66	202,56	225,19	30,86	35,59	38,97	35,31	28,04
5	227,26	198,38	179,25	154,67	135,86	32,70	29,17	31,70	32,45	32,95

O modelo proposto de GA mostrou ser capaz de evoluir a formação para grupos de 2 e 4 agentes de forma eficiente usando os parâmetros propostos. Usamos tanto a mutação Gaussiana como a mutação Uniforme e obtivemos resultados satisfatórios usando uma quantidade de 150 indivíduos e, pelo menos, 400 gerações. Ao todo realizamos 20 simulações com cada conjunto

de parâmetros; para os parâmetros de GA sugeridos, 100% das simulações foram capazes de cercar o incêndio de forma satisfatória.

É importante salientar que algumas simulações usando diferentes velocidades foram realizadas, porém em número menor de experimentos, estas não estão detalhadas no texto, mas também apresentaram resultados satisfatórios.

Exemplos de simulações de evolução, com visualização 2D, podem ser obtidos na página¹ web do projeto. Exemplos da aplicação do melhor indivíduo resultante das evoluções, como os apresentados nas Tabelas 5.1, 5.2 e 5.3, com visualização 2D e 3D podem ser obtidos no mesmo endereço.

Como descrito anteriormente, o melhor indivíduo resultante de uma evolução é passado ao módulo de ação de combate através de um arquivo texto. A partir deste indivíduo são calculadas as posições de atuação de cada agente de combate com as Equações 4.1 e 4.2 de acordo com o especificado nas Tabelas 4.1 e 4.2, para, respectivamente, grupos de 4 e 2 robôs de combate. De posse das coordenadas das posições de atuação indicadas para cada robô, estes iniciam o processo de navegação até estas posições a fim de iniciar o combate ao incêndio.

5.2 Controle de Navegação

Como descrito na Seção 4.4 o controle inteligente de navegação foi desenvolvido com uma Rede Neural Artificial do tipo *Multi Layer Perceptron* (MLP) treinada com o algoritmo de aprendizagem *Resilient Backpropagation* (RProp). Esta RNA foi desenvolvida e treinada no SNNS (SNNS, 2006). O controle inteligente realizado pela RNA permite navegação e desvio de obstáculos em um ambiente dinâmico, usando apenas informações disponíveis localmente, obtidas pelos sensores dos veículos. O ambiente foi desenvolvido com terreno irregular e com a simulação de propriedades físicas disponíveis através da ODE.

A única forma do veículo desenvolvido, descrito na Seção 4.2, obter dados do ambiente é através dos seus sensores. O controle robusto é feito com a simulação de erro na leitura dos dados sensoriais e com a aplicação de simulação de erro nos comandos passados aos atuadores. Para que seja possível realizar a navegação autônoma, o veículo possui um número adequado de sensores e atuadores. Com relação aos atuadores, cada veículo possui dois motores. Um motor angular (para o giro do volante) e um motor linear (para o torque). Com relação aos sensores, cada veículo possui um GPS, responsável pela obtenção da localização, uma bússola, necessária para a obtenção da orientação do veículo, e 5 sensores de distância (sonares simulados), que

¹Vídeos de exemplos de simulação disponíveis em <http://pessin.googlepages.com/robombeiros>

apresentam as características de capacidade de medir distâncias entre 15cm e 11m, como o *Polaroid 6500* (www.senscomp.com). Sensores e atuadores, como são dispositivos de hardware, apresentam erros e interferências que devem ser tratadas em nossa simulação; detalharemos as experiências com simulação de erros no final desta Seção.

Cada agente robótico recebe uma mensagem passada pelo agente monitor com duas coordenadas, a primeira é a coordenada de início de aceiro e a segunda é a coordenada de fim de aceiro. O robô usa o GPS e a coordenada recebida para calcular qual deve ser a sua orientação para o alvo; usa a bússola para verificar sua orientação; usa o GPS para verificar sua localização; e usa os sensores de distância para verificar a proximidade de obstáculos.

Nesta Seção descreveremos inicialmente como treinamos, testamos e escolhemos a RNA a ser aplicada no controle; em seguida descrevemos as experiências e os resultados da aplicação da RNA realizando o controle robusto de navegação com desvio de obstáculos.

5.2.1 Teste e Seleção da RNA

Como apresentado na Seção 4.4, usamos a ferramenta SNNS (SNNS, 2006) para desenvolver 6 diferentes topologias de RNA. As redes possuem 4, 9, 18, 24, 30 e 36 neurônios na camada oculta. Inicialmente realizamos 3 treinos de cada topologia de RNA para análise do Erro Médio Absoluto (E_{MAE} - Equação 4.3). Sendo cada treino realizado com uma semente (*random seed*) aleatória diferente. O E_{MAE} usando nesta validação foi obtido a partir da análise do resultado dos testes das RNA com 5.000, 10.000, 20.000, 40.000, 60.000, 80.000 e 100.000 ciclos. Na Seção 4.4 foram apresentadas também: quais as entradas e as saídas da RNA; o tamanho da base obtida e a forma de validação. A base de dados² selecionada para este trabalho pode ser obtida na página do projeto.

Os primeiros treinos, com as RNAs com 4 e 9 neurônios na camada oculta, mostrou que estas RNAs não apresentam capacidade de aprender. A tabela 5.4 apresenta o E_{MAE} destas Redes; podemos ver que o erro permanece constante para todos os ciclos apresentados, bem como para as diferentes sementes aleatórias.

²Este arquivo pode ser obtido, especificamente, em <http://pessin.googlepages.com/logrna.rar>.

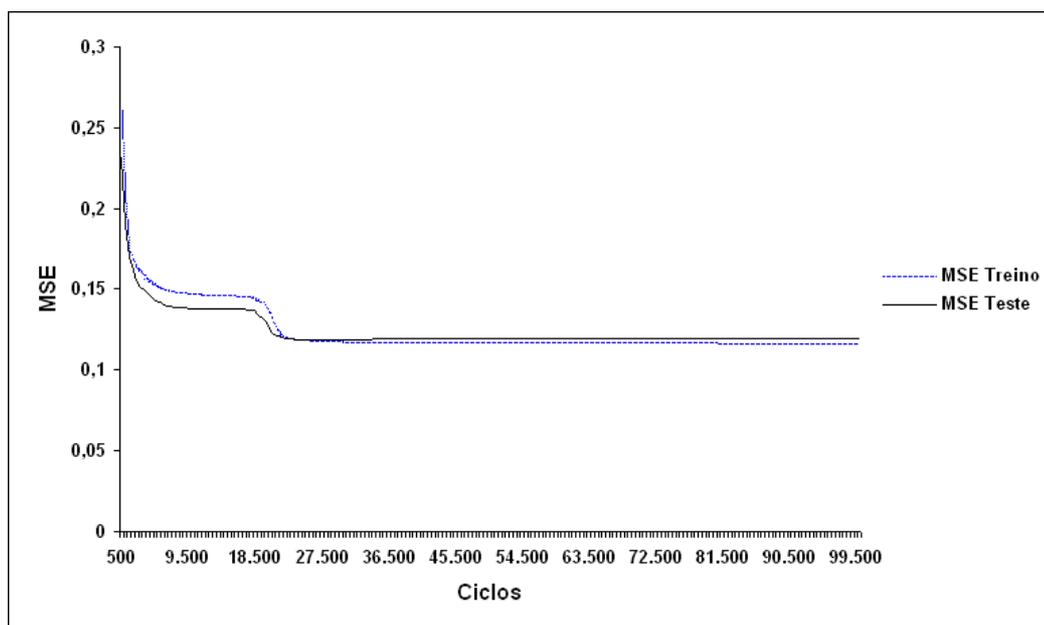
Tabela 5.4: E_{MAE} medido na base de teste usando as RNAs com 4 e 9 neurônios na camada oculta.

RNA	Seed	5.000 Ciclos		20.000 Ciclos		80.000 Ciclos		100.000 Ciclos	
		Giro	Veloc.	Giro	Veloc.	Giro	Veloc.	Giro	Veloc.
7x4x2	A	0,306	2,000	0,306	2,000	0,306	2,000	0,306	2,000
7x4x2	B	0,306	2,000	0,306	2,000	0,306	2,000	0,306	2,000
7x4x2	C	0,306	2,000	0,306	2,000	0,306	2,000	0,306	2,000
7x9x2	A	0,306	2,000	0,306	2,000	0,306	2,000	0,306	2,000
7x9x2	B	0,306	2,000	0,306	2,000	0,306	2,000	0,306	2,000
7x9x2	C	0,306	2,000	0,306	2,000	0,306	2,000	0,306	2,000

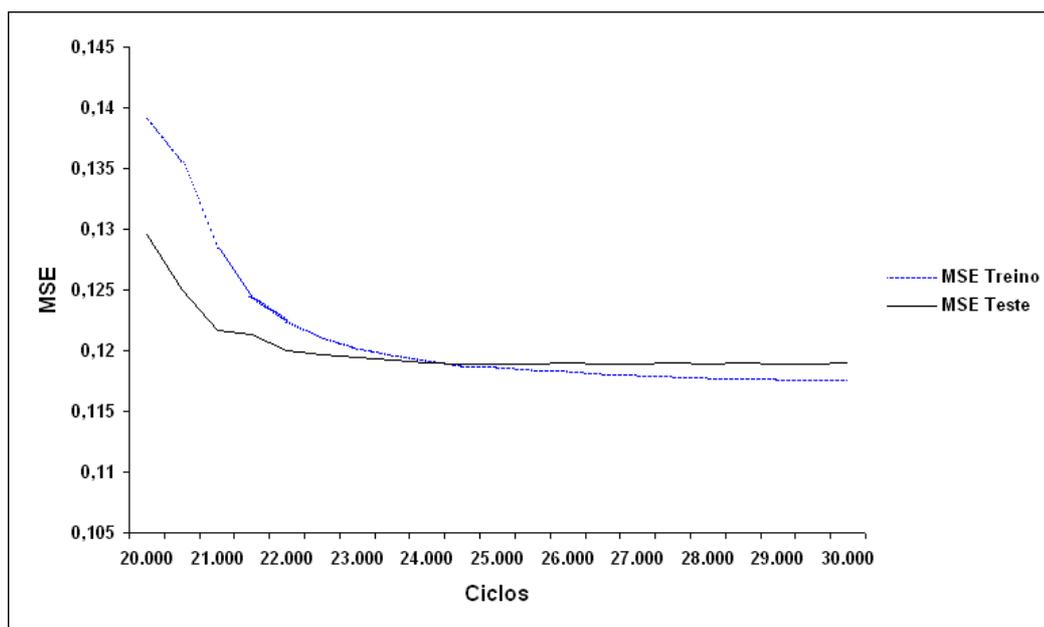
O treino nas RNAs com 18, 24, 30 e 36 neurônios na camada oculta mostrou que todas estas RNAs apresentam capacidade de aprender. A tabela 5.5 apresenta o E_{MAE} destas Redes; podemos ver que a RNA com 24 neurônios na camada oculta apresenta menor erro no valor de giro de volante que as outras. Consideramos o valor do giro mais importante que o valor da velocidade. Assim, esta RNA foi escolhida para análise do Erro Médio Quadrático (E_{MSE} - Equação 4.4), apresentado na Figura 5.11.

Tabela 5.5: E_{MAE} medido na base de teste usando as RNAs com 18, 24, 30 e 36 neurônios na camada oculta.

RNA	Seed	5.000 Ciclos		20.000 Ciclos		80.000 Ciclos		100.000 Ciclos	
		Giro	Veloc.	Giro	Veloc.	Giro	Veloc.	Giro	Veloc.
7x18x2	A	0,241	0,185	0,239	0,176	0,233	0,175	0,231	0,159
7x18x2	B	0,228	0,113	0,228	0,110	0,228	0,110	0,228	0,111
7x18x2	C	0,236	0,153	0,218	0,108	0,218	0,109	0,218	0,109
7x24x2	A	0,132	0,111	0,129	0,111	0,129	0,111	0,129	0,111
7x24x2	B	0,149	0,132	0,147	0,132	0,147	0,132	0,147	0,132
7x24x2	C	0,138	0,121	0,129	0,108	0,118	0,106	0,118	0,107
7x30x2	A	0,190	0,128	0,180	0,123	0,180	0,123	0,180	0,124
7x30x2	B	0,183	0,107	0,182	0,103	0,173	0,103	0,173	0,103
7x30x2	C	0,168	0,112	0,161	0,108	0,163	0,109	0,164	0,109
7x36x2	A	0,152	0,097	0,148	0,097	0,148	0,097	0,148	0,097
7x36x2	B	0,159	0,141	0,159	0,119	0,154	0,110	0,154	0,110
7x36x2	C	0,146	0,104	0,139	0,106	0,140	0,107	0,140	0,107



(a)



(b)

Figura 5.11: E_{MSE} da RNA de 24 neurônios: (a) Visão completa (de 0 até 100.000 ciclos); (b) Visão ampliada da inversão das curvas - Ponto Ótimo de Generalização (de 20.000 até 30.000 ciclos).

Como a Figura 5.11 mostra que, aproximadamente a partir de 24.000 ciclos existe a inversão das curvas de treino e teste (Ponto Ótimo de Generalização, descrito na Seção 2.3.2), foi feita uma nova análise do E_{MAE} , com ciclos de teste próximos ao Ponto Ótimo (de 500 em

500 ciclos), a fim de observar exatamente quando o erro no giro começa a subir. A Tabela 5.6 apresenta os resultados da análise do E_{MAE} próxima ao Ponto Ótimo de Generalização.

Tabela 5.6: E_{MAE} da RNA com 24 neurônios na camada oculta: Erro no giro aumenta a partir de 32.500 ciclos

Ciclos	Giro	Velocidade
23.000	0,12051	0,10561
24.000	0,11945	0,10545
25.000	0,11898	0,10590
26.000	0,11870	0,10506
27.000	0,11842	0,10611
28.000	0,11832	0,10613
29.000	0,11814	0,10620
30.000	0,11805	0,10624
30.500	0,11797	0,10631
31.000	0,11789	0,10631
31.500	0,11780	0,10623
32.000	0,11769	0,10627
32.500	0,11767	0,10628
33.000	0,11771	0,10625
34.000	0,11771	0,10626

Podemos ver na Tabela 5.6 que o ponto de inversão do erro E_{MAE} no giro ocorre entre as RNAs com 32.500 e 33.000 ciclos. Assim, escolhemos a RNA treinada com 32.500 ciclos para ser convertida em um programa em C e aplicada no controle inteligente dos agentes de combate a incêndio.

O treino destas RNA, feito no SNNS, teve como base um arquivo de *script*³ disponível no site do projeto.

5.2.2 Aplicação da RNA no Controle

Após implementar a RNA selecionada no controle dos robôs móveis, inserindo no sistema de controle o código gerado como descrito na Seção anterior, buscamos verificar se o controle proporcionado por ela é eficiente para realizar a navegação entre os pontos iniciais e fi-

³Este arquivo pode ser obtido, especificamente, em <http://pessin.googlepages.com/rbbscript.txt>

nais solicitados pelo agente monitor. Assim, realizamos inúmeros experimentos com diferentes quantidades de árvores no ambiente e com diferentes topografias do terreno. Também buscamos avaliar a quantidade de erro que a RNA suporta, mantendo a navegação correta.

Dados os pontos iniciais e finais para um grupo de agentes, a navegação no ambiente desenvolvido pode ser vista na Figura 5.12. O ambiente foi parametrizado com 4 níveis de ocupação por árvores, calculamos a área de ocupação do total de árvores em relação à área do terreno. As ocupações são aproximadamente 10%; 5%; 2,5% e 0,625%.

Resultados serão considerados satisfatórios da seguinte forma: cada veículo deve ser capaz de atravessar uma região simulada representando uma área de cerca de 1km utilizando as diferentes taxas de ocupação por árvores. Os veículos não podem colidir com árvores nem com outros veículos.

O resultado das simulações com diferentes tipos de ocupação do ambiente por árvores, podem ser vistos na Tabela 5.7. Podemos ver nesta Tabela que, para ambientes com ocupação de 5%, ou menos, a RNA foi capaz de realizar a navegação com desvios satisfatórios para 98% e 100% das simulações. Os erros que ocorreram advém da entrada dos veículos em regiões de afunilamento; uma forma inicial para tratar este erro seria a aplicação de capacidade de dar ré, que não foi implementada em nosso modelo.

Tabela 5.7: Resultado das simulações usando a RNA com diferentes ocupações no ambiente.

Quantidade de Simulações de Navegação	Área Ocupada com Árvores	Resultados Satisfatórios com a RNA
50	10,00%	42 (84%)
50	5,000%	49 (98%)
50	2,500%	49 (98%)
50	0,625%	50 (100%)

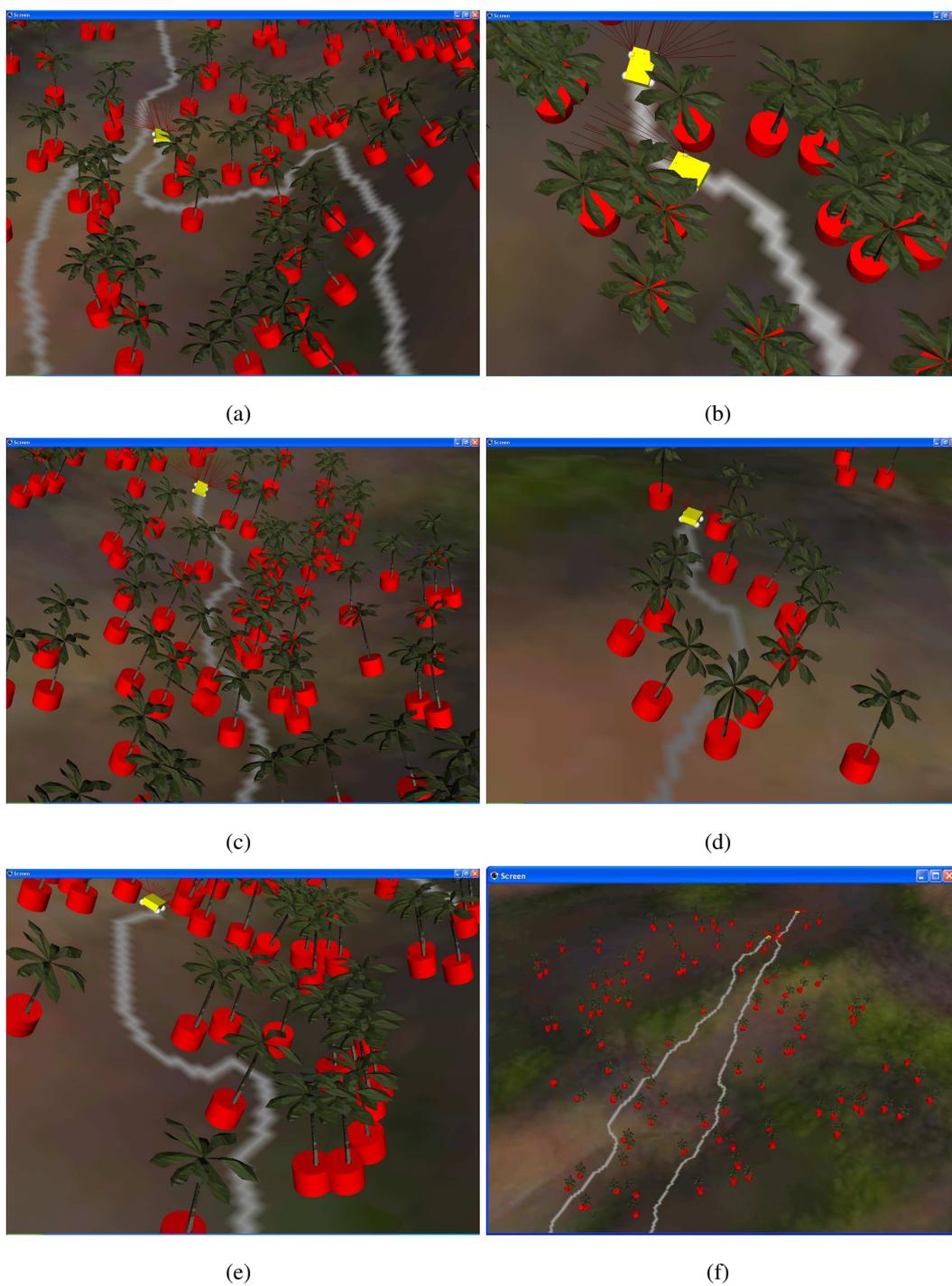


Figura 5.12: Trajetos realizados com simulações usando a RNA: Todas as Figuras apresentam taxa de ocupação por árvores de 10%, com exceção da (f) que apresenta um ambiente com 2,5% de ocupação.

A partir das simulações apresentadas na Tabela 5.7, escolhemos o ambiente com 5% de ocupação para realizar novas simulações aplicando ruído simulado nos sensores e nos atuadores.

Essa simulação de ruído é feita com a aplicação de um valor percentual aleatório entre *-Ruído* e *+Ruído* sobre a faixa de valor original do sensor/atuador. A Tabela 5.7 apresenta resultados de 50 simulações. Para as simulações que seguem com aplicação de ruído, foram realizadas não mais 50 mas sim 20 simulações.

A Tabela 5.8 apresenta os resultados de simulações aplicando ruído simulado na leitura de todos os 5 sonares. Podemos ver que os resultados são 100% satisfatórios com aplicação de até 10% de ruído. A aplicação de 20% de ruído apresentou falhas em desvios simples.

Tabela 5.8: Resultado das simulações aplicando ruído na leitura dos 5 sonares.

Quantidade de Simulações de Navegação	Ruído Aplicado nos 5 Sonares	Resultados Satisfatórios com a RNA
20	Até 5,00%	20 (100,00%)
20	Até 10,00%	20 (100,00%)
20	Até 20,00%	18 (90,00%)

A Tabela 5.9 apresenta os resultados de simulações aplicando ruído simulado nos valores de comando dos dois atuadores. Essa simulação de ruído foi feita da mesma forma descrita acima. Podemos ver que os resultados são 100% satisfatórios com aplicação de até 10% de ruído. A aplicação de 20% de ruído também apresentou falhas em desvios simples.

Tabela 5.9: Resultado das simulações aplicando ruído nos 2 atuadores.

Quantidade de Simulações de Navegação	Ruído Aplicado nos 2 Atuadores	Resultados Satisfatórios com a RNA
20	Até 5,00%	20 (100,00%)
20	Até 10,00%	20 (100,00%)
20	Até 20,00%	18 (90,00%)

A Tabela 5.10 apresenta os resultados de simulações aplicando ruído simulado de até 100% na leitura de um sensor de distância (escolhido aleatoriamente a cada ciclo). Podemos ver que os resultados são satisfatórios com aplicação de até 100% de ruído em apenas um sensor; quando aplicado em dois sensores, o controle apresenta falha ao percorrer o trajeto.

Tabela 5.10: Resultado das simulações aplicando ruído de até 100% em um sensor aleatório.

Quantidade de Simulações de Navegação	Quantidade de Sonares com até 100% de Ruído	Resultados Satisfatórios com a RNA
20	1	20 (100,00%)
20	2	16 (80,00%)

A Figura 5.13 apresenta imagens de simulações com aplicação de ruído de até 100% na leitura de um sensor aleatório, como as apresentadas na Tabela 5.10. Esta Figura apresenta um veículo com sensores sem ruído e um veículo com aplicação de ruído simulado; podemos ver a trajetória perfeitamente reta do veículo sem erro nos sensores e a trajetória irregular do veículo com erro simulado. A trajetória torna-se irregular devido ao fato de o veículo interpretar o ruído como um possível aviso de colisão; o valor do comando aplicado no giro das rodas é maior no desvio do que para a reorientação para o alvo.

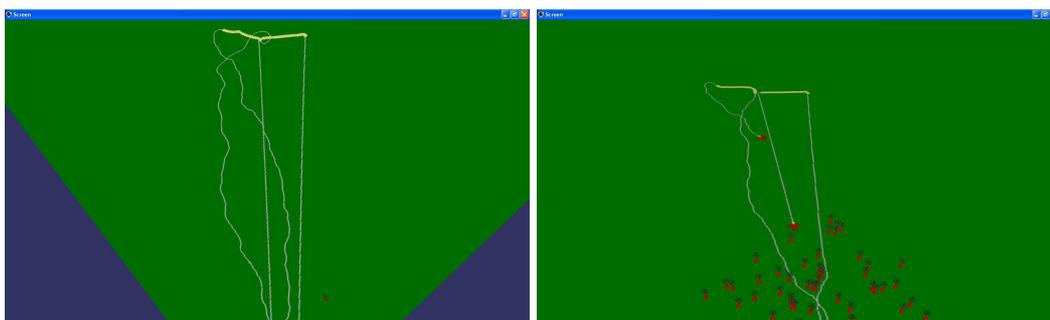


Figura 5.13: Duas cenas de navegação considerando erro nos sensores. Em cada uma das imagens é apresentado um veículo com ruído simulado e outro com sensores sem ruído.

Um último experimento foi realizado tendo como objetivo verificar o posicionamento final dos robôs móveis quando solicitada navegação para o mesmo ponto, considerando uma margem de erro no GPS. A Figura 5.14 apresenta cenas da simulação onde podemos ver que os agentes realizaram desvio entre si de forma satisfatória.

O modelo proposto de RNA mostrou ser capaz de controlar os atuadores do robô móvel desenvolvido usando apenas informações disponíveis localmente, obtidas pelos sensores dos veículos em um ambiente dinâmico, com terreno irregular e com as propriedades físicas disponibilizadas pela ODE. O controle robusto foi validado com a demonstração de resultados aplicando ruídos nos sensores e nos atuadores dos RMAs.

Exemplos de simulações de navegação no ambiente de simulação 3D desenvolvido po-

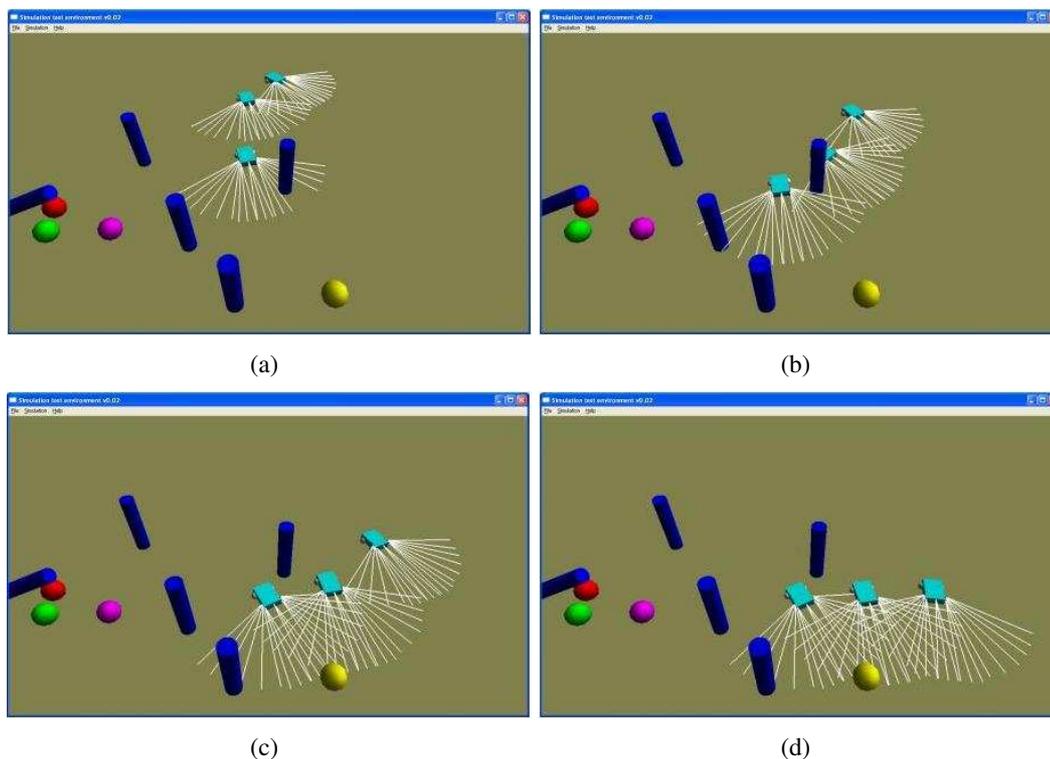


Figura 5.14: Seqüências de uma simulação com navegação e desvio satisfatórios, para observação das posições de parada dos RMAs. (a) até (d) especificam a ordem da seqüência.

dem ser vistos nos vídeos⁴ disponíveis na página do projeto.

O próximo Capítulo apresenta as considerações finais da dissertação, onde serão revisados os objetivos e conceitos que motivaram o desenvolvimento deste trabalho, detalhadas as contribuições resultantes e apresentados os possíveis trabalhos futuros.

⁴Vídeos: NavegaRNA2agEx1.rar, NavegaRNA4agEx2.rar e NavegaRNA2agEx3.rar disponíveis em <http://pessin.googlepages.com>

6 CONSIDERAÇÕES FINAIS

Este Capítulo apresenta as considerações finais da dissertação. Serão revisados os objetivos e conceitos que motivaram o desenvolvimento deste trabalho. Em seguida serão detalhadas as contribuições resultantes. Finalizando com a descrição dos trabalhos futuros vislumbrados durante o desenvolvimento desta pesquisa.

6.1 Revisão do Tema

Dado o contínuo aumento da complexidade no controle de sistemas robóticos, bem como na aplicação de sistemas multi-robóticos substituindo as pessoas em atividades que coloquem em risco a vida humana, onde a atuação humana é deficitária ou onde é, de certa forma, ineficiente; propomos uma pesquisa relacionada com a aplicação de técnicas de Inteligência Artificial no desenvolvimento de um Sistema Multi-Agente robótico aplicado ao problema da monitoração e combate a incêndios em áreas florestais. O objetivo macro é evoluir estratégias de posicionamento de equipes de combate a incêndio em uma unidade monitora (agente deliberativo) e criar métodos robustos de navegação em unidades de combate fisicamente simuladas (agentes híbridos), considerando terreno e vegetação.

Os incêndios florestais causam diversos tipos de danos humanos, materiais e ambientais. O desenvolvimento de um sistema multi-robótico para atuação em monitoração e combate a incêndios em áreas florestais é um problema multidisciplinar. Assim, o sucesso desta tarefa envolve o uso de uma variedade de tecnologias de diferentes campos, como robótica móvel (*e.g.* controle de movimentação, fusão de sensores, modelagem robótica). Envolve a coleta de informações de especialistas sobre incêndios (*e.g.* comportamento do fogo em ambientes naturais, técnicas e métodos de monitoração, estratégias reais de combate). Além de, neste trabalho, envolver estudos sobre Sistemas Multi-Agentes e Aprendizado de Máquina.

Na tarefa de controle de incêndio, uma das questões mais importantes tem relação com a estratégia de posicionamento dos agentes de combate. De acordo com a capacidade de atuação

de cada agente, as condições climáticas (vento, chuva), a topografia e a vegetação, diferentes formações podem ser sugeridas. Estas estratégias, quando sugeridas por especialistas, não conseguem levar em conta um número muito grande de variáveis.

A tarefa de controle de navegação é um problema fundamental nos estudos em robótica móvel. Para operar em ambientes complexos de larga escala (que se modificam dinamicamente, compostos de obstáculos estáticos e móveis) um robô móvel deve ser capaz de adquirir e utilizar conhecimento sobre o ambiente, estimar uma posição, possuir a habilidade de reconhecer obstáculos e responder em tempo real as diferentes situações que possam ocorrer; todas estas funcionalidades devem operar em conjunto. Normalmente sensores e atuadores são sujeitos a erros e interferências, assim o controle das ações de um robô deve sempre levar em conta a imprecisão dos sensores e motores envolvidos. Um sistema robusto deve permitir que, mesmo com sensores e atuadores imprecisos, o agente cumpra o seu objetivo.

6.2 Resultados e Conclusões

Este trabalho descreveu a proposta de implementação de um modelo multi-agente para monitoração e combate a incêndios florestais por robôs móveis. Apresentamos, no decorrer do texto, que é bastante comum na robótica móvel a utilização de sensores do tipo sonar para obtenção de distâncias, sendo assim, nosso veículo foi desenvolvido com este tipo de sensor, além dos sensores bússola e GPS e de dois atuadores. Os agentes de combate a incêndio implementados realizam navegação com constante leitura de informação sensorial, sendo assim essencialmente do tipo *feedback*; o agente deliberativo (planejador de estratégia) recebe informação do ambiente apenas na inicialização, assim, pode ser considerado como possuindo estratégia de controle do tipo *feedforward*.

Respeitamos, na definição da arquitetura de controle, todas as tarefas descritas na Seção 2.1.5.2. Ainda, na questão de controle, o agente responsável pela evolução de estratégias é essencialmente deliberativo; os agentes de combate a incêndio, por receberem um macro-comando de “*desloque-se autonomamente até (x,y)* ” possuem um plano para execução desta navegação, além de possuir um sistema de reação sensorial-motora, assim, podem ser considerados como agentes híbridos. As ferramentas de simulação pesquisadas serviram como inspiração para o desenvolvimento do nosso ambiente; descrevemos tanto as ferramentas estudadas como as bibliotecas de software escolhidas para o desenvolvimento do nosso ambiente de simulação. Podemos ver que a tarefa de monitoração e combate a incêndios apresenta um problema onde a atuação de um sistema multi-agente pode trazer grandes benefícios, tanto por

poder mapear o funcionamento de uma operação real como por, no caso deste trabalho, apresentar um modelo com veículos autônomos, o que reduz os riscos humanos na operação. A cooperação apresentada é resultado de uma evolução de uma estratégia de grupo usando Algoritmos Genéticos, assim, não desenvolvemos a comunicação entre agentes para este fim. A comunicação simulada neste trabalho é detalhada na Seção 4.1.

Com relação aos quatro aspectos principais que concretizam o projeto de um agente robótico: desenvolvemos os agentes de combate a incêndio com capacidade de autonomia e localização, por serem fisicamente simulados apresentam também a corporificação. No entanto, não consideramos carga de bateria ou combustível, como se fossem auto-suficientes a ponto de cumprir a missão que lhes foi atribuída sem necessidade de realizar um reabastecimento. No desenvolvimento do sistema multi-agente implementamos a definição da estratégia de forma centralizada e a atuação no combate a incêndio de forma distribuída.

Usamos Algoritmos Genéticos para a evolução de estratégias de posicionamento do grupo robótico de combate a incêndios. A vantagem de se utilizar GA para a otimização das posições é que eles não necessitam de uma base com os dados de treinamento. No caso do combate a incêndios, estas informações seriam muito difíceis ou mesmo impossíveis de serem obtidas. Propomos o uso de uma Rede Neural Artificial para realizar o controle inteligente em um RMA que deve ser capaz de navegar desviando de obstáculos. O uso de RNA no controle de navegação é justificado por serem as técnicas mais robustas em relação a situações novas e inesperadas, por possuírem alto grau de generalização e, como o desenvolvimento de regras de controle é uma operação bastante complexa, podemos criar uma base de treino através da obtenção de dados de um veículo operado manualmente, para posterior treino e teste da RNA.

Ainda, descrevemos as técnicas reais de operação, as ferramentas e os métodos utilizados atualmente por grupos humanos na identificação e combate a incêndios florestais. Podemos ver que os aceiros têm sido uma alternativa viável e de amplo uso no meio florestal, nas unidades de conservação e nas margens de rodovias para a redução na propagação do fogo (RIBEIRO et al., 2006). Desenvolvemos um agente monitor responsável pelas etapas de detecção, planejamento e comunicação. A forma de combate a incêndio implementada utiliza equipamento pesado, assim os veículos de combate a incêndio são motoniveladoras; e o método de combate é o indireto; utilizando as motoniveladoras para criação de um aceiro. O terreno escolhido é semelhante aos parques nacionais da região nordeste do Rio Grande do Sul, que apresentam predomínio de combustível florestal do Modelo 1.

Por fim, implementamos ferramentas que permitiram validar, através de simulações, tanto o modelo de evolução de estratégias baseado em GAs, quanto a navegação e controle

robusto dos robôs móveis autônomos, realizada através do uso de uma RNA. Demonstramos também que o modelo proposto é plenamente capaz de atuar e combater de modo bastante satisfatório os focos de incêndio.

Em forma de itens, as principais contribuições deste trabalho são:

- O desenvolvimento de um ambiente 3D realístico de simulação física usando as bibliotecas *Open Dynamics Engine* (ODE) (ODE, 2007), *Open Scene Graph* (OSG) (OSG, 2007) e *Demeter* (DEMETER, 2007). Este ambiente¹ foi utilizado para validar a capacidade de navegação dos robôs móveis desenvolvidos e pode servir para diversas outras aplicações, visto que foi construído de forma bastante estruturada; pode ser utilizado desde em jogos até a utilização em aplicações de Realidade Virtual e simulação.
- O desenvolvimento e a validação de um modelo de RNA capaz de realizar a orientação e navegação robusta dos robôs móveis de modo totalmente autônomo. Essa RNA se mostrou capaz de realizar navegação e desvio de obstáculos satisfatórios com a aplicação de grande quantidade de ruído nos dados dos sensores e dos atuadores em um ambiente virtual 3D com realismo físico. Além disso, demonstramos de modo prático o uso da RNA no controle robusto de navegação robótica.
- O desenvolvimento de um modelo de Algoritmo Genético capaz de formar satisfatoriamente grupos de 2 e 4 agentes, que salvam a maior área possível de vegetação e maximizam a utilidade de atuação de cada robô, minimizando operação sem necessidade. Além disso, demonstramos o uso com sucesso do AG na estratégia de combate a incêndios e detalhamos a estrutura do cromossomo e do *fitness* desenvolvidos.

Ainda, outras contribuições podem ser consideradas, como:

- Construção de um modelo simplificado de propagação de incêndio e a implementação deste modelo em dois simuladores, um 2D usando a biblioteca SDL e um 3D usando as bibliotecas OSG, ODE e *Demeter*.
- Apresentação de modelos de comunicação multi-agente, detalhados em (PESSIN, 2007).
- Levantamento bibliográfico e documentação das informações sobre dados florestais, tipos de vegetação, topografia, e comportamento de incêndios.
- Levantamento bibliográfico e documentação das informações sobre ferramentas e técnicas de combate à incêndio florestais utilizadas por bombeiros.

¹Código fonte disponível em <http://pessin.googlepages.com>

- Proposta de um modelo de arquitetura de operação multi-agente para identificação e combate a incêndios em áreas florestais.
- Diversas implementações de protótipos, com bibliotecas de código livre em C/C++.
- Diversos artigos publicados (PESSIN et al., 2007e, 2007f, 2007b, 2007g, 2007c, 2007a, 2007d; OSÓRIO et al., 2007).

6.3 Trabalhos Futuros

A fim de permitir um ambiente de simulação mais realístico, dois trabalhos futuros poderiam ser desenvolvidos neste sentido: a criação de uma simulação de fogo mais realística; e a possibilidade de importação de modelos reais de terrenos e de vegetação. O ambiente desenvolvido poderia ainda sofrer modificações com o objetivo de permitir o estudo de diferentes estratégias de combate a incêndios, para comparação com a atual estratégia implementada, bem como para comparação com técnicas convencionais.

Com relação ao controle, seria interessante implementar alguma forma de planejamento macro de trajetória, antes de iniciar a navegação dos RMAs; uma possível opção é a implementação de um algoritmo A* para desvio de rios, lagos e encostas íngremes; a RNA continuaria importante para permitir o desvio de obstáculos não considerados pelo A* (objetos não detalhados no mapa do ambiente), bem como para obstáculos móveis (como os outros agentes de combate). Além disso, também seria interessante implementar a capacidade de navegar para trás (dar ré) em caso de identificação de colisão iminente (afunilamento).

Com relação à evolução de estratégias, esta é realizada apenas uma vez, no início da identificação do incêndio, sem considerar um possível replanejamento devido a eventuais mudanças imprevistas (como, por exemplo, características do vento). Algo neste sentido poderia ser feito de modo que o planejamento fosse refeito a cada identificação de mudanças no comportamento do vento, atualizando assim a formação do grupo robótico. Além disso, poderiam ser consideradas as formas de comunicação, com avisos de falhas de operação por parte dos agentes do combate, para que a unidade monitora solicitasse a ativação de outra unidade de combate ou ativasse o replanejamento com o novo número de agentes disponíveis.

Outro trabalho futuro é a aplicação do AG sobre o protótipo desenvolvido com simulação física em 3D, a fim de otimizar os trajetos e até a otimização da morfologia do robô; este trabalho não foi realizado devido a necessidades extremamente grandes de capacidade de processamento, que inviabilizariam a conclusão do mesmo em tempo hábil.

Ainda - é bom lembrar - que os modelos propostos, tanto de evolução de estratégias como de controle inteligente, podem ser adaptados para outras missões, vistos que são módulos distintos. Alguns outros exemplos de uso poderiam ser: aplicações de recuperação e controle de outros tipos de desastres além de incêndios, como acidentes nucleares, enchentes, acidentes químicos, ou mesmo exploração espacial.

REFERÊNCIAS

- ANDREW, W.; BARAFF, D. Differential equations basics. In: *Proceedings of 20th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'93)*. Anaheim, California, USA: Assn for Computing Machinery, 1993. p. B1–B8.
- ANTUNES, M. A. H. Uso de satélites para detecção de queimadas e para avaliação do risco de fogo. *Revista Ação Ambiental*, Viçosa, MG, v. 12, p. 24–27, 2000.
- ASTECH. *ASTECH Angewandte Sensortechnik GmbH Website*. Acesso em maio de 2007. Disponível em: <<http://www.astech.de>>.
- BALCH, T.; ARKIN, R. C. Motor schema-based formation control for multiagent robot teams. In: *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*. San Francisco, CA, USA: AAAI Press, 1995. p. 10–16.
- BALCH, T.; ARKIN, R. C. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, v. 14, n. 6, p. 926–939, 1998.
- BALUJA, S. Evolution of an artificial neural network based autonomous land vehicle controller. *IEEE Transactions on Systems, Man and Cybernetics*, v. 26, n. 3, p. 450–463, 1996.
- BARROS, L. M.; GONZAGA, L.; RAPOSO, A. B. Open scene graph: conceitos básicos e aplicações em realidade virtual. In: *Anais do IX Symposium on Virtual and Augmented Reality (SVR'07)*. Petrópolis, RJ, Brasil: LNCC, 2007.
- BATISTA, A. Detecção de incêndios florestais por satélite. *Revista Floresta (ISSN: 00153826)*, Curitiba, Pr, v. 34, n. 2, p. 237–241, 2004.
- BEKEY, G. A. *Autonomous Robots: From Biological Inspiration to Implementation and Control*. Cambridge, Massachusetts, USA: The MIT Press, 2005. 577 p.
- BIANCHI, R. A. C.; SIMÕES, A. S.; COSTA, A. H. R. Comportamentos reativos para seguir pistas em um robô móvel guiado por visão. In: *Anais do V Simpósio Brasileiro de Automação Inteligente (SBAI'01)*. Canela, RS: Sociedade Brasileira de Automática, 2001.
- BOEING. *The Boeing Company Website*. Acesso em maio de 2007. Disponível em: <<http://www.boeing.com/defense-space/military/x-45/index.html>>.
- BOEKHORST, R. J. A.; LUNGARELLA, M.; PFEIFER, R. Dimensionality reduction through sensory-motor coordination. In: *Proceedings of the 10th Joint International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP'03)*. Istanbul, Turkey: Springer-Verlag, 2003. v. 2714, p. 496–503.
- BOMBARDIER. *Bombardier Inc. Website*. Acesso em maio de 2007. Disponível em: <<http://www.canadair.com/>>.

BORDINI, R. H.; VIEIRA, R.; MOREIRA, A. F. Fundamentos de sistemas multiagentes. In: *Anais do XXI Congresso da Sociedade Brasileira de Computação (CSBC'01), XX Jornada de Atualização em Informática (JAI)*. Fortaleza, CE, Brasil: Sociedade Brasileira de Computação, 2001. v. 2, p. 3–41.

BORENSTEIN, J. et al. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, v. 14, n. 4, p. 231–249, 1997.

BRAGA, A. P.; CARVALHO, A. P. L. F. de; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e aplicações*. Rio de Janeiro, RJ, Brasil: LTC, 2000. 262 p.

BROOKS, R. A. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, v. 2, n. 1, p. 14–23, 1986.

BURCHARDT, H.; SALOMON, R. Implementation of path planning using genetic algorithms on mobile robots. In: IEEE PRESS. *Proceedings of IEEE Congress on Evolutionary Computation*. Vancouver, BC, Canada, 2006.

CARVALHO, A. C. F.; BRAGA, A. P.; LUDERMIR, T. B. Sistemas inteligentes: Fundamentos e aplicações. In: _____. Barueri, SP, Brasil: Manole, 2003. cap. Computação Evolutiva, p. 225–248.

CASTRO, F. X.; TUDELA, A.; SEBASTIA, M. T. Modeling moisture content in shrubs to predict fire risk in catalonia. *Agricultural and Forest Meteorology*, v. 116, p. 49–59, 2001.

CE. *Comissão Européia: O que faz a Europa? Incêndios Florestais*. Acesso em setembro de 2006. Disponível em: <<http://ec.europa.eu/research/leaflets/disasters/pt/forest.html>>.

CHUVIECO, E.; CONGALTON, R. G. Application of remote sensing and geographic information systems to forest fire hazard mapping. *Remote Sensing of the Environment*, v. 29, p. 147–159, 1989.

CMU. *Carnegie Mellon University: Navigation Laboratory*. Acesso em dezembro de 2007. Disponível em: <<http://www.cs.cmu.edu/tyata/Project/NavLab11.html>>.

COSTA, E. R.; GOMES, M. L.; BIANCHI, R. A. C. Um mini robô móvel seguidor de pistas guiado por visão local. In: *Anais do VI Simpósio Brasileiro de Automação Inteligente (SBAI'03)*. Bauru, SP: Sociedade Brasileira de Automática, 2003. p. 710–715.

CPTEC/INPE. *Centro de Previsão do Tempo e Estudos Climáticos/Instituto Nacional de Pesquisas Espaciais*. Acesso em maio de 2007. Disponível em: <<http://www.cptec.inpe.br/queimadas>>.

CSD/NASA. *Computational Sciences Division/NASA Website*. Acesso em maio de 2007. Disponível em: <<http://ase.arc.nasa.gov/msf/>>.

CYBERBOTICS. *Cyberbotics Ltd. Website*. Acesso em maio 2007. Disponível em: <<http://www.cyberbotics.com/>>.

DARPA. *DARPA Grand Challenge Webpage: Defense Advanced Research Projects Agency*. 2007. Disponível em: <<http://www.darpa.mil/grandchallenge/>>.

DASSAULT. *Dassault Aviation Website*. Acesso em maio de 2007. Disponível em: <<http://www.dassault-aviation.com>>.

DATCU, D. et al. Aibo robot as a soccer and rescue game player. In: *Proceedings of 5th annual European GAME-ON Conference (GAME-ON'2004)*. Het Pand, Ghent, Belgium: Ghent University, 2004.

DEJONG, K. *The analysis and behaviour of a class of genetic adaptive systems*. Tese (Doutorado) — University of Michigan, USA, 1975.

DEMEC/UFGM. *Centro de Estudos Aeronáuticos, Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais (UFMG)*. Acesso em maio de 2007. Disponível em: <<http://www.demec.ufmg.br/Cea/principal.html>>.

DEMETER. *Demeter Terrain Engine*. Acesso em maio de 2007. Disponível em: <<http://www.tbgssoftware.com/>>.

DORIGO, M. et al. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, v. 17, p. 223–245, 2004.

DPI/INPE. *Divisão de Processamento de Imagens/Instituto Nacional de Pesquisas Espaciais*. Acesso em maio de 2007. Disponível em: <<http://www.dpi.inpe.br/proarco/bdqueimadas>>.

DUDEK, G.; JENKIN, M. *Computational Principles of Mobile Robotics*. Cambridge, London, UK: The MIT Press, 2000. 280 p.

ENGADGET. *Weblogs Inc. Network Website*. Acesso em maio de 2007. Disponível em: <<http://www.engadget.com/2006/06/16/robot-strider-walks-on-water-and-eventually-land>>.

ERMAN, L. D. et al. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (ISSN: 0360-0300)*, v. 12, n. 2, p. 213–253, 1980.

EVERETT, H. *Sensors for Mobile Robots: Theory and Application*. Wellesley, MA, USA: A.K. Peters, 1995. 544 p.

FAHLMAN, S. E. *Computer Science Technical Report: An empirical study of learning speed in back-propagation networks*. Carnegie-Mellon University, USA, 1988.

FONSECA, J. M. *Protocolos de negociação com coligações em sistemas multiagente*. Tese (Doutorado) — Universidade Nova de Lisboa, Portugal, 2000.

GARCIA, A. C. B.; SICHMAN, J. S. Sistemas inteligentes: Fundamentos e aplicações. In: _____. Barueri, SP, Brasil: Manole, 2003. cap. Agentes e Sistemas Multi-Agentes, p. 269–306.

GARMIN. *Garmin International Inc. Website*. Acesso em maio de 2007. Disponível em: <<http://www.garmin.com>>.

GDAL. *Geospatial Data Abstraction Library*. Acesso em maio de 2007. Disponível em: <<http://www.gdal.org/>>.

GENESERETH, M. R.; KETCHPEL, S. P. Software agents. *Communications of the ACM*, v. 37, n. 7, p. 49–53, 1994.

- GO, J.; BROWNING, B.; VELOSO, M. Accurate and flexible simulation for dynamic, vision-centric robots. In: *Proceedings of 3th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*. New York, NY: IEEE Computer Society, 2004.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Professional, 1989. 432 p.
- GOLDBERG, D. E. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, v. 5, p. 139–167, 1991.
- GONZALEZ, J. A.; MARENGONI, M. Controle baseado em algoritmo genético para um braço robótico móvel. In: *Anais do VII Simpósio Brasileiro de Automação Inteligente (SBAI'05)*. São Luís, MA: Sociedade Brasileira de Automática, 2005.
- GOUMA, V.; CHRONOPOULOU, A. Wildland fire danger zoning: A methodology. *International Journal of Wildland Fire*, v. 8, n. 1, p. 37–43, 1998.
- GRAYMATTER. *GrayMatter Inc. Website*. Acesso em dezembro de 2007. Disponível em: <<http://www.graymatterinc.com/teamgray/theteam.shtml>>.
- GUARDA. *Concurso Robô Bombeiro, Escola Superior de Tecnologia e Gestão da Guarda, Portugal*. Acesso em dezembro de 2007. Disponível em: <<http://robobombeiro.ipg.pt/>>.
- HALLAM, J. *Lecture Notes on Intelligent Sensing and Control: Notes on Control Theory*. University of Edinburgh, UK, 1990.
- HAYES-ROTH, B. An architecture for adaptive intelligent systems. *Artificial Intelligence*, v. 72, p. 329–365, 1995.
- HEINEN, F. J. *Robótica Autônoma: Integração entre Planificação e Comportamento Reativo*. São Leopoldo, RS: Unisinos, 2000. 84 p.
- HEINEN, F. J. *Sistema de Controle Híbrido para Robôs Móveis Autônomos*. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, 2002.
- HEINEN, M. R. *Controle Inteligente do Caminhar de Robôs Móveis Simulados*. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, 2007.
- HEINEN, M. R.; OSÓRIO, F. S. Estacionamento de um veículo de forma autônoma simulado em um ambiente tridimensional realístico. In: *Anais do XIV Seminário de Computação (SEMINCO'05)*. Blumenau, SC, Brasil: Universidade Regional de Blumenau (FURB), 2005. v. 1, p. 56–65.
- HEINEN, M. R.; OSÓRIO, F. S. Algoritmos genéticos aplicados ao problema de roteamento de veículos. *Revista Hifen*, Uruguaiana, RS, v. 30, n. 58, p. 89–96, 2006.
- HEINEN, M. R.; OSÓRIO, F. S. Co-evolução da morfologia e controle de robôs móveis simulados utilizando realidade virtual. In: *Anais do IX Symposium on Virtual and Augmented Reality (SVR'07)*. Petrópolis, RJ, Brasil: LNCC, 2007. v. 1, p. 187–196.
- HOFFMANN, L. T.; CASTRO, A. P. A.; SILVA, J. D. S. Controle inteligente de um robô móvel utilizando imagens. In: *Anais do XXIV Congresso da Sociedade Brasileira de Computação (CSBC'04), Jornada de Robótica Inteligente (JRI)*. Salvador, BA: Sociedade Brasileira de Computação, 2004.

HOLLAND, J. *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan, USA: The University of Michigan Press, 1975.

HONDA. *American Honda Motor Co. Inc. Website*. Acesso em maio de 2007. Disponível em: <<http://asimo.honda.com>>.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks (ISSN:0893-6080)*, v. 2, p. 359–366, 1989.

IDMIND. *Projecto RAPOSA: Robot semi-Autônomo Para Operações de SALvamento*. Acesso em maio de 2007. Disponível em: <<http://raposa.idmind.pt/>>.

IROBOT. *iRobot Corporation Website*. Acesso em maio de 2007. Disponível em: <<http://www.irobot.com>>.

JOCHEM, T.; POMERLEAU, D.; THORPE, C. Next generation neurally based autonomous road follower. In: *Proceedings of the International Conference on Intelligent Autonomous Systems*. Pittsburgh, Pennsylvania: IOS Press, 1993.

JOCHEM, T.; POMERLEAU, D.; THORPE, C. Vision guided lane-transition. In: *Proceedings of the Intelligent Vehicles Symposium*. Detroit, MI, USA: IEEE Service Center, 1995. p. 30–35.

JONG, K. A. D. Adaptive system design: A genetic approach. *IEEE Transactions on Systems, Man and Cybernetics*, v. 10, n. 9, p. 566–574, 1980.

JPL/NASA. *Jet Propulsion Laboratory/NASA*. Acesso em maio de 2007. Disponível em: <<http://www-robotics.jpl.nasa.gov>>.

JUNG, C. R. et al. Computação embarcada: Projeto e implementação de veículos autônomos inteligentes. In: *Anais do XXV Congresso da Sociedade Brasileira de Computação (CSBC'05), XXIV Jornada de Atualização em Informática (JAI)*. São Leopoldo, RS: Sociedade Brasileira de Computação, 2005. v. 1, p. 1358–1406.

KARTALOPOULOS, S. V. *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*. New York, USA: IEEE Press Understanding Science and Technology Series, 1996. 232 p.

KITANO, H. Robocup rescue: a grand challenge for multi-agent systems. In: *Proceedings of Fourth International Conference on MultiAgent Systems*. Boston, MA, USA: IEEE Press, 2000. p. 5–12.

KOPROSKI, L. P. *O Fogo e Seus Efeitos Sobre a Heperto e a Mastofauna Terrestre no Parque Nacional de Ilha Grande*. Dissertação (Mestrado) — Universidade Federal do Paraná (UFPR), 2005.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. Introdução aos algoritmos genéticos. In: *Anais do XIX Congresso da Sociedade Brasileira de Computação (CSBC'99), XVIII Jornada de Atualização em Informática (JAI)*. Rio de Janeiro, RJ: Sociedade Brasileira de Computação, 1999. v. 2, p. 51–126.

LIF. *Laboratório de Incêndios Florestais: Pesquisas e projetos em prevenção e combate de incêndios florestais, Universidade Federal do Paraná (UFPR)*. Acesso em outubro de 2006. Disponível em: <<http://www.floresta.ufpr.br/firelab>>.

LUCIDARME, P. An evolutionary algorithm for multi-robot unsupervised learning. In: *Proceedings of Evolutionary Computation Congress (CEC'04)*. Portland, Oregon: IEEE Press, 2004. v. 2, p. 19–23.

MA. *Guia metodológico para elaboração de plano municipal/intermunicipal de defesa da floresta contra incêndios, Apêndice 1: Metodologia para a elaboração da carta dos combustíveis florestais*. Ministério da Agricultura do Brasil, 2006.

MACEDO, A. R. M.; MACEDO, A. R. L.; DUARTE, J. B. F. Robótica aplicada ao combate a incidentes. *Revista TN Petróleo*, Rio de Janeiro, RJ, n. 53, p. 108–113, 2007.

MAES, P. Intelligent software: easing the burdens that computers put on people. *IEEE Expert (ISSN: 0885-9000)*, v. 11, p. 62–63, 1996.

MARCHI, J. *Navegação de robôs móveis autônomos: Estudo e implementação de abordagens*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2001.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943.

MEDEIROS, A. A. D. A survey of control architectures for autonomous mobile robots. *Journal of the Brazilian Computer Society: Special Issue on Robotics*, v. 4, n. 3, 1998.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. New York, USA: Springer-Verlag, 1996. 387 p.

MICHEL, O. *Khepera Simulator Website*. Acesso em maio de 2007. Disponível em: <<http://diwww.epfl.ch/lami/team/michel/khep-sim/>>.

MICROSOFT. *Microsoft Robotics Studio Website*. Acesso em maio de 2007. Disponível em: <msdn.microsoft.com/robotics/>.

MILLER, L. et al. Firebot: Design of an autonomous fire fighting robot. In: *Proceedings of Florida Conference on Recent Advances in Robotics (FCRAR'03)*. Florida, USA: Florida Atlantic University, 2003.

MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: The MIT Press, 1996.

MONDADA, F. et al. The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robotics and Automation Magazine (ISSN: 1070-9932)*, v. 12, p. 21–28, 2005.

NIKOLOPOULOS, C. *Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. New York, USA: Marcel Dekker Inc. Press, 1997. 327 p.

NILSSON, N. J. A mobile automaton: An application of artificial intelligence techniques. *Autonomous Mobile Robots: Control, Planning and Architecture*, v. 2, n. 1, p. 233–239, 1969.

NOC. *UK National Oceanography Centre Website*. Acesso em maio de 2007. Disponível em: <<http://www.noc.soton.ac.uk>>.

NOLFI, S. Power and the limits of reactive agents. *Neurocomputing*, v. 42, n. 1, p. 119–145, 2002.

NOLFI, S.; FLOREANO, D. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, Massachusetts, USA: The MIT Press, 2000. 320 p.

NWANA, H. S. Software agents: An overview. *Knowledge Engineering Review*, Cambridge, UK, v. 11, n. 3, p. 205–244, 1996.

ODE. *Open Dynamics Engine Website*. Acesso em maio de 2007. Disponível em: <<http://www.ode.org>>.

OPENGL. *Open Graphics Library Website*. Acesso em maio de 2007. Disponível em: <www.opengl.org>.

OSAGIE, P. *Distributed Control for Networked Autonomous Vehicles*. Dissertação (Mestrado) — KTH Computer Science and Communication, Royal Institute of Technology, Sweden, 2006.

OSG. *Open Scene Graph, OSG Community Website*. Acesso em maio de 2007. Disponível em: <<http://www.openscenegraph.com>>.

OSHKOSH. *Oshkosh Truck Corporation Website*. Acesso em dezembro de 2007. Disponível em: <<http://www.terramax.com/>>.

OSÓRIO, F. et al. Inteligência artificial para jogos: Agentes especiais com permissão para matar... e raciocinar! In: *Anais do VI Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES'07)*. São Leopoldo, RS: Universidade do Vale do Rio dos Sinos (UNISINOS), 2007.

OSÓRIO, F. S. *INSS: Un Système Hybride Neuro-Symbolique pour l'Apprentissage Automatique Constructif*. Tese (Doutorado) — Institut National Polytechnique de Grenoble (INPG), Grenoble, França, 1998.

OSÓRIO, F. S. *Página de internet do Grupo de Pesquisas em Veículos Autônomos*. Acesso em maio de 2007. Disponível em: <<http://www.exatec.unisinos.br/autonom/html/br/index.htm>>.

OSÓRIO, F. S.; BITTENCOURT, J. R. Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens. In: *Anais do I Workshop de Inteligência Artificial (WIA'00)*. Santa Cruz do Sul, RS: Universidade de Santa Cruz do Sul (Unisc), 2000.

OSÓRIO, F. S.; HEINEN, F. J.; FORTES, L. Controle inteligente de veículos autônomos: Automatização do processo de estacionamento de carros. In: *Anais do X Seminário de Computação (SEMINCO'01)*. Blumenau, SC: Universidade Regional de Blumenau (FURB), 2001.

OSÓRIO, F. S.; HEINEN, F. J.; FORTES, L. Controle da tarefa de estacionamento de um veículo autônomo através do aprendizado de um autônomo finito usando uma rede neural j-cc. In: *Anais do VII Simpósio Brasileiro de Redes Neurais (SBRN'02)*. Recife, CE: Sociedade Brasileira de Computação, 2002.

OSÓRIO, F. S. et al. Seva3d: Using artificial neural networks to autonomous vehicle parking control. In: *Proceedings of IEEE World Congress On Computational Intelligence, International Joint Conference On Neural Networks (IJCNN'06)*. Vancouver, Canadá: IEEE Press, 2006. p. 4704–4711.

- OSÓRIO, F. S. et al. Increasing reality in virtual reality applications through physical and behavioural simulation. In: *Proceedings of the Virtual Concept Conference: Research in Interactive Design*. Porto Alegre, RS: Springer Verlag, 2006. v. 12.
- OSÓRIO, F. S.; VIEIRA, R. Sistemas híbridos inteligentes. In: *Anais do XIX Congresso da Sociedade Brasileira de Computação (CSBC'99), Encontro Nacional de Inteligência Artificial (ENIA)*. Rio de Janeiro, RJ: Sociedade Brasileira de Computação, 1999.
- PACK, D. J. et al. Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives. *IEEE Transactions on Education*, v. 4t, n. 3, p. 369–376, 2004.
- PARKER, L. E. Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual robot performance. *Autonomous Robots*, Springer Netherlands, v. 8, n. 3, p. 239–267, June 2000.
- PENNSSTATE. *Penn State Mobile Robotics Program Website*. Acesso em dezembro de 2007. Disponível em: <<http://www.ecsel.psu.edu/avanzato/robots/contests/firefighting/index.htm>>.
- PERERA, A. H.; REMMEL, T. K. Fire mapping in a northern boreal forest: assessing avhrr/ndvi methods of change detection. *Forest Ecology and Management*, v. 152, p. 119–129, 2001.
- PESSIN, G. *Evolução de Estratégias e Controle Inteligente em Sistemas Multi-Robóticos Robustos*. Proposta de Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS, Junho 2007.
- PESSIN, G. et al. Ambiente virtual para simulação de robôs móveis inteligentes aplicados no combate de incêndios florestais. *Revista Disciplinarum Scientia, Série: Ciências Naturais e Tecnológicas (ISSN 1981-2841)*, Santa Maria, RS, Brasil, k, p. x–y, Dezembro 2007.
- PESSIN, G. et al. Controle inteligente em um sistema multi-robótico aplicado na identificação e combate de incêndios em Áreas florestais. In: *Anais do IV Observatório - Sessões Técnicas de Informática*. Chapecó, SC, Brasil: Universidade Comunitária Regional de Chapecó (UNOCHAPECÓ), 2007.
- PESSIN, G. et al. Desenvolvimento de um ambiente para simulação de robótica móvel em Áreas florestais. In: *Anais do VI Simpósio de Informática da Região Centro do RS (SIRC-RS'07)*. Santa Maria, RS, Brasil: Centro Universtário Franciscano (UNIFRA), 2007.
- PESSIN, G. et al. Evoluindo estratégias de posicionamento em um sistema multi-robótico aplicado ao combate de incêndios florestais. *Revista Hifen (ISSN 0103-1155)*, Uruguaiana, RS, Brasil, v. 31, n. 59, p. x–y, Dezembro 2007.
- PESSIN, G. et al. Simulação virtual de agentes autônomos para a identificação e controle de incêndios em reservas naturais. In: *Anais do IX Symposium on Virtual and Augmented Reality (SVR'07 - ISBN 857669111-6)*. Petrópolis, RJ, Brasil: LNCC, 2007. v. 1, p. 236–245.
- PESSIN, G. et al. Utilizando agentes autônomos com aprendizado para a identificação e combate de incêndios em Áreas florestais. In: *Anais do VII Simpósio de Informática do Planalto Médio (SIPM'07)*. Passo Fundo, RS, Brasil: Universidade de Passo Fundo (UPF), 2007.

PESSIN, G. et al. Utilizando redes neurais artificiais no controle de robôs móveis aplicados ao combate de incêndios florestais. In: *Anais do XVI Seminário de Computação (SEMINCO'07)*. Blumenau, SC, Brasil: Universidade Regional de Blumenau (FURB), 2007. v. 1, p. 19–30.

PFEIFER, R.; IIDA, F.; BONGARD, J. New robotics: Design principles for intelligent systems. *Artificial Life*, v. 11, p. 99–120, 2005.

PFEIFER, R.; SCHEIER, C. From perception to action: The right direction? In: *Proceedings of IEEE International Conference: From Perception to Action*. Lausanne, Switzerland: IEEE Computer Society Press, 1994. p. 1–11.

PFEIFER, R.; SCHEIER, C. Sensory-motor coordination: The metaphor and beyond. *Robotics and Autonomous Systems*, v. 20, n. 2, p. 157–178, 1997.

PFEIFER, R.; SCHEIER, C. *Understanding Intelligence*. Cambridge, Massachusetts, USA: The MIT Press, 1999. 697 p.

PIERI, E. R. D. *Curso de Robótica Móvel*. Universidade Federal de Santa Catarina (UFSC), 2002.

POLAROID. *Polaroid Corporation Website*. Acesso em maio de 2007. Disponível em: <<http://www.polaroid.com>>.

POMERLEAU, D. Defense and civilian applications of the alvinn robot driving system. In: *Proceedings of Government Microcircuit Applications Conference*. San Diego, CA: ARPA, 1994. p. 358–362.

RAMOS, P. C. M. Sistema nacional de prevenção e combate a incêndios florestais. In: *Anais do I Fórum Nacional sobre Incêndios Florestais (IPEF'95)*. Piracicaba, SP: Instituto de Pesquisas e Estudos Florestais, 1995. p. 29–38.

REEVE, R. E. *Generating Walking Behaviours in Legged Robots*. Tese (Doutorado) — University of Edinburgh, Escócia, 1999.

REIS, L. P. *Coordenação em sistemas multiagente: aplicações na gestão universitária e futebol robótico*. Tese (Doutorado) — Universidade do Porto, Portugal, 2003.

REMLEY, K. A. et al. Standards development for wireless communications for urban search and rescue robots. In: *9th Annual International Symposium on Advanced Radio Technologies (ISART'2007)*. Boulder, Colorado, USA: National Telecommunications and Information Administration, 2007.

REZENDE, S. O. *Sistemas Inteligentes: Fundamentos e Aplicações*. Barueri, SP: Manole, 2003. 525 p.

RIBEIRO, C.; COSTA, A. H. R.; ROMERO, R. A. F. Robôs móveis inteligentes: Princípios e técnicas. In: *Anais do XXI Congresso da Sociedade Brasileira de Computação (CSBC'01), XX Jornada de Atualização em Informática (JAI)*. Fortaleza, CE, Brasil: Sociedade Brasileira de Computação, 2001. v. 3, p. 258–306.

RIBEIRO, G. A. et al. Eficiência de um retardante de longa duração na redução da propagação do fogo. *Revista Árvore*, v. 30, n. 6, p. 1025–1031, 2006.

RI/CMU. *Robotics Institute Website, School of Computer Science, Carnegie Mellon University*. Acesso em maio de 2007. Disponível em: <<http://www.ri.cmu.edu/>>.

RIEDMILLER, M.; BRAUN, H. *Technical Report: RPROP, Description and Implementation Details*. Universität Karlsruhe, Deutschland, 1994.

RIGOLOT, E. Fuel-break assessment with an expert appraisal approach. In: *Proceedings of IV International Conference on Forest Fire Research*. Coimbra, Portugal: Millpress Science Publishers, 2002.

ROSEMBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol*, v. 65, n. 6, p. 386–408, 1958.

ROSEMBLATT, F. *Principles of Neurodynamics*. New York, USA: Spartan, 1962.

ROTHERMEL, R. C. *General Technical Report INT-143: How to predict the spread and intensity of forest and range fires*. United States Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, 161p, 1983.

RUMELHART, D.; MCCLELLAND, J. *Parallel Distributed Processing*. Cambridge: The MIT Press, 1986.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning Internal Representations by Error Propagation*. Cambridge, MA, USA: The MIT Press, 1986.

RUSSEL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. New Jersey, USA: Prentice Hall, 1995. 932 p.

SANTOS, C. T. *Um Ambiente Virtual Inteligente e Adaptativo baseado em Modelos de Usuário e Conteúdo*. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, 2004.

SDL. *Simple DirectMedia Layer Website*. Acesso em maio de 2007. Disponível em: <<http://www.libsdl.org/>>.

SELVATICI, A. H. P. *AAREACT: Uma Arquitetura Comportamental Adaptativa para Robôs Móveis que Integra Visão, Sonares e Odometria*. Dissertação (Mestrado) — Universidade de São Paulo (USP), São Paulo, SP, 2005.

SHARP. *Sharp Electronics Corporation*. Acesso em maio de 2007. Disponível em: <<http://www.sharpsma.com>>.

SHIBA, M. H. et al. Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. In: *Anais do XII Simpósio Brasileiro de Sensoriamento Remoto (SBSR'05)*. Goiânia, GO: INPE, 2005. v. 1, p. 4319–4326.

SILVEIRA, R. A. *Introdução a Sistemas Multiagente*. Universidade Federal de Santa Catarina (UFSC), 2006.

SMITH, R. *Guia do usuário: Open Dynamics Engine v0.5*. Disponível em: <http://www.ode.org/ode-latest-userguide.pdf>, Acesso em maio de 2006.

SMITHERS, T. *Lecture Notes on Intelligent Sensing and Control: Control and Control Strategies*. University of Edinburgh, UK, 1990.

SNNS. *Stuttgart Neural Network Simulator Website*. Acesso em maio de 2006. Disponível em: <<http://www-ra.informatik.uni-tuebingen.de/SNNS/>>.

SONY. *Sony Corporation Website*. Acesso em maio de 2007. Disponível em: <<http://www.sony.net/Products/aibo/index.html>>.

SOUZA, N. F. et al. Animando humanos virtuais em tempo-real usando visão computacional e redes neurais. In: *Anais do IX Symposium on Virtual and Augmented Reality (SVR'07)*. Petrópolis, RJ, Brasil: LNCC, 2007. v. 1, p. 180–186.

STELLMANN, P. *Simulator Bob Website*. Acesso em maio de 2007. Disponível em: <<http://www.tu-harburg.de/ti6/mitarbeiter/pst/Sim/Simulator.html>>.

STONE, P.; VELOSO, M. M. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, v. 8, n. 3, p. 345–383, 2000.

TAKAHASHI, Y.; TAMURA, T.; ASADA, M. Cooperation via environmental dynamics caused by multi robots in a hostile environment. In: *Proceedings of 4th IFAC Symposium on Intelligent Autonomous Vehicles*. Sapporo, Japan: Elsevier, 2001.

TRINITY. *Firefighting Robot Contest Website, Trinity College*. Acesso em dezembro de 2007. Disponível em: <<http://www.trincoll.edu/events/robot/>>.

VIDEORAY. *VideoRay LLC Website*. Acesso em maio de 2007. Disponível em: <<http://www.videoray.com>>.

WADDOUPS, N. *Juice Website*. Acesso em maio de 2007. Disponível em: <<http://www.natew.com/juice/>>.

WALL, M. *Genetic Algorithms Library, Massachusetts Institute of Technology*. Acesso em maio de 2007. Disponível em: <<http://lancet.mit.edu/ga/>>.

WALTER, W. An imitation of life. *Scientific American*, v. 182, p. 42–45, 1950.

WETTERGREEN, D. *DEPTHX: Deep Phreatic Thermal Explorer Website*. Acesso em maio de 2007. Disponível em: <<http://www.ri.cmu.edu/projects/project-544.html>>.

WHITTAKER, W. R. L. *Red Team: DARPA Grand Challenge Technical Paper*. Carnegie Mellon University, Robotics Institute, 2004.

WIDROW, B.; HOFF, M. E. Adaptative switching circuits. In: *Institute of Radio Engineers (IRE) Western Electronic Show and Convention Record (WESCON)*. New York: Institute of Radio Engineers, 1960. p. 96–104.

WILKING, D.; ROFER, T. Realtime object recognition using decision tree learning. In: *Proceedings of RoboCup 2004: Robot World Cup VIII, Lecture Notes in Artificial Intelligence (LNAI 3276)*. Lisboa, Portugal: Springer, 2004. v. 1, p. 556–563.

WOOLDRIDGE, M. *Reasoning about Rational Agents*. Cambridge, Massachusetts, USA: The MIT Press, 2000. 240 p.

WOOLDRIDGE, M.; JENNINGS, N. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, v. 10, n. 2, p. 115–152, 1995.

WOWWEE. *Wow Wee Robosapien Website*. Acesso em maio de 2007. Disponível em:
<www.wowwee.com/robosapien/robo1/robomain.html>.

ZHAO, Y.; COLLINS, E. G. Robust automatic parallel parking in tight spaces via fuzzy logic. *Robotics and Autonomous Systems*, v. 51, p. 111–127, 2005.

ZHOU, Y. *An Area Exploration Strategy Evolved by Genetic Algorithm*. Dissertação (Mestrado) — University of Georgia, 2005.