

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Síntese de Controladores Autônomos em Robótica Móvel por meio de Computação Bio-inspirada

Renato Reder Cazangi

Prof. Dr. Fernando José Von Zuben (Orientador)

Tese apresentada à Pós-graduação da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como requisito parcial à obtenção do grau de **Doutor em Engenharia Elétrica** na área de Engenharia de Computação.

Banca examinadora:

Prof. Dr. Fernando José Von Zuben – FEEC/UNICAMP

Prof. Dr. Christiano Lyra Filho – FEEC/UNICAMP

Prof. Dr. Romis Ribeiro de Faissol Attux – FEEC/UNICAMP

Prof. Dr. Maurício Fernandes Figueiredo – DC/UFSCAR

Prof^ª. Dr^ª. Roseli Aparecida Francelin Romero – ICMC/USP

Campinas, 18 de Dezembro de 2008

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Renato Reder Cazangi

Data da Defesa: 18 de dezembro de 2008

Título da Tese: "Síntese de Controladores Autônomos em Robótica Móvel por Meio de Computação Bio-inspirada"

Prof. Dr. Fernando José Von Zuben (Presidente): Fernando José Von Zuben
Profa. Dra. Roseli Aparecida Francelin Romero: Roseli Ap. Francelin Romero
Prof. Dr. Maurício Fernandes Figueiredo: Maurício F. Figueiredo
Prof. Dr. Christiano Lyra Filho: _____
Prof. Dr. Romis Ribeiro de Faissol Attux: Romis Ribeiro de Faissol Attux

Resumo

Novas técnicas de navegação autônoma de robôs móveis visam suprir a crescente demanda pelo emprego de robôs em diversos setores da sociedade e junto a uma ampla gama de tarefas. Os desafios envolvidos no desenvolvimento do sistema que controla o robô permitem afirmar que a inteligência embarcada em robôs atuais ainda encontra-se em um nível incipiente e limitado. Neste trabalho, cinco frentes de pesquisa complementares são propostas visando estudar, teórica e praticamente, aspectos fundamentais de projeto e implementação de controladores autônomos inteligentes para robótica móvel. Metodologias de computação bio-inspirada e de robótica evolutiva provêm os conceitos e ferramentas que fundamentam as cinco propostas, as quais são validadas com base em sistemas de navegação concebidos e aplicados a problemas relevantes da área. Uma série de simulações computacionais em ambientes virtuais e experimentos com robôs reais é realizada, permitindo medir o alcance das contribuições e apontar as principais frentes de atuação que se abrem como perspectivas futuras da pesquisa.

Abstract

Novel techniques for autonomous robot navigation aim at fulfilling the growing demand for mobile robots in multiple segments of society and in a plethora of tasks. The challenges involved in developing the system which controls the robot allow to say that the intelligence embedded in the current robots is found to be still incipient and limited. In this work, five complementary research fronts are proposed intending to study, theoretical and practically, aspects which are fundamental to the design and implementation of intelligent autonomous controllers for mobile robotics. Bio-inspired computing and evolutionary robotics methodologies provide the concepts and tools underlying the five proposals, which are validated through navigation systems devised and applied to important problems. Numerous real robot experiments as well as computational simulations taking place in virtual environments are carried out, allowing for the evaluation of contributions and also the discussion of future possibilities.

Trabalhos Afins Publicados pelo Autor

Cazangi, R. R., Von Zuben, F. J., & Figueiredo, M. F. (2005). Autonomous navigation system applied to collective robotics with ant-inspired communication. Proceedings of the 2005 Conference on Genetic and Evolutionary Computation. vol. 1, pp. 121-128. Washington DC: ACM Press.

Cazangi, R. R., & Von Zuben, F. J. (2006). Evolutionary Stigmergy in Multipurpose Navigation Systems. Proceedings of the 2006 IEEE Congress on Evolutionary Computation, (pp. 1094-1101). Vancouver.

Cazangi, R. R., & Von Zuben, F. J. (2006). Immune Learning Classifier Networks: Evolving Nodes and Connections. Proceedings of the 2006 IEEE Congress on Evolutionary Computation, (pp. 7994-8001). Vancouver.

Cazangi, R. R., Von Zuben, F. J., & Figueiredo, M. F. (2006). Stigmergic autonomous navigation in collective robotics. In A. Abraham, C. Grosan, & V. Ramos, Stigmergic Optimization (pp. 25-64). Springer-Verlag.

Cazangi, R. R., Von Zuben, F. J., Feied, C., Gillam, M., Handler, J., & Smith, M. (2007). An Evolutionary Approach for Autonomous Robotic Tracking of Dynamic Targets in Healthcare Environments. Proceedings of the 2007 IEEE Congress on Evolutionary Computation, (pp. 3654-3661). Cingapura.

Von Zuben, F. J., & Cazangi, R. R. (2007). Computação Bio-Inspirada em Controle de Processos. In L. A. Aguirre (Ed.), Enciclopédia de Automática - Controle & Automação (Vol. 3, pp. 283-324). São Paulo, SP: Editora Edgar Blucher.

É com muito orgulho que dedico este trabalho aos meus pais, Vilma e João; aos meus irmãos, Gustavo e Humberto, aos meus avôs, Zenaide e Paulo (*in memoriam*), e à minha esposa Fernanda

Agradecimentos

Ao professor Fernando Von Zuben por tamanha confiança, constante motivação e fundamental orientação, e por ser um exemplo de dedicação, competência e simplicidade.

Aos professores Christiano Lyra, Romis Attux, Maurício Figueiredo e Roseli Romero pela participação na banca julgadora e todas as contribuições a este trabalho.

À CAPES pelo suporte financeiro, à Faculdade de Engenharia Elétrica e de Computação e ao Departamento de Engenharia de Computação e Automação Industrial por fornecer valiosos subsídios à minha pesquisa.

À Microsoft e ao Washington Hospital Center pelo suporte em diversos aspectos e pelas oportunidades que me permitiram realizar conquistas antes inimagináveis. Special thanks to Michael Gillam, Uri Kartoun, Juan Wachs, John Gillotte, Francisco Mendez, Alexei Kontsevoy, Liz Tondreau and all other folks in Washington D.C. who, somehow, helped me with this work.

Ao LBiC e a todos os seus membros que me auxiliaram na construção desta pesquisa e, principalmente, tornaram o laboratório no melhor ambiente de trabalho e amizade que eu já tive o prazer de fazer parte. Um imenso obrigado aos meus amigos Tiago Barra, Marcelo Caetano, Helder Knidel, Renan Moiola, Pablo Dalbem, George Barreto, Eurípedes Pinheiro, Guilherme Coelho, Wilfredo Villanueva e Hamilton Ferreira.

Aos meus pais João e Vilma, irmãos Humberto e Gustavo e avôs, cada qual a seu jeito, por serem o alicerce, a força e a inspiração para que eu seja o que sou. Quanto mais longe eu fui, mais perto eles estiveram. Absolutamente nada disto seria possível sem eles.

À minha esposa Fernanda, pelo amor, carinho e enorme compreensão ao longo de tantos anos. Obrigado ainda a sua família por ser também a minha.

Aos amigos, familiares e todas as pessoas que de alguma maneira participaram desta caminhada.

Sumário

CAPÍTULO 1: INTRODUÇÃO	1
1.1. MOTIVAÇÃO	1
1.2. ROBÓTICA MÓVEL E CONTROLE AUTÔNOMO	6
1.3. FUNDAMENTOS DESTE TRABALHO	8
1.3.1. <i>Bio-inspiração</i>	8
1.3.2. <i>Robótica Evolutiva</i>	9
1.3.3. <i>Técnicas de Computação Bio-inspirada Utilizadas</i>	10
1.3.3.1. Algoritmos Evolutivos	10
1.3.3.2. Sistemas Classificadores com Aprendizado	10
1.3.3.3. Sistemas Imunológicos Artificiais	11
1.3.3.4. Inteligência de Enxame	11
1.4. CONTRIBUIÇÕES	11
1.5. CRONOLOGIA E ORGANIZAÇÃO DO TRABALHO	12
CAPÍTULO 2: ASPECTOS E ESPECIFICAÇÕES DE SISTEMAS DE NAVEGAÇÃO AUTÔNOMOS.....	15
2.1. ASPECTOS DE SISTEMAS DE NAVEGAÇÃO AUTÔNOMOS.....	15
2.2. COMPORTAMENTO E ARQUITETURA	15
2.2.1. <i>Puramente Reativo</i>	16
2.2.2. <i>Puramente Deliberativo</i>	18
2.2.3. <i>Híbrido</i>	19
2.3. AMBIENTE E ROBÔ.....	22
2.3.1. <i>Virtual</i>	22
2.3.2. <i>Real</i>	23
2.3.3. <i>Especificações Adotadas</i>	25
2.3.3.1. Khepera II	25
2.3.3.2. PatrolBot	27
2.4. TAREFAS E APLICAÇÕES.....	28
2.5. APRENDIZAGEM	29
2.5.1. <i>Evolução com Aprendizagem</i>	30
2.5.2. <i>Aprendizagem por Evolução</i>	31
2.6. INDIVIDUALIDADE E SOCIEDADE	32
2.7. OTIMIZAÇÃO	34
2.8. ESPECIFICAÇÕES DAS FRENTE DE PESQUISA	35
2.9. DESCRIÇÃO DOS VÍDEOS	36
CAPÍTULO 3: SISTEMA DE NAVEGAÇÃO AUTÔNOMO BASEADO EM SISTEMA CLASSIFICADOR COM APRENDIZAGEM	37
3.1. INTRODUÇÃO	37
3.2. CONTEXTUALIZAÇÃO DO CAPÍTULO	37
3.3. DESCRIÇÃO DO SISTEMA DE NAVEGAÇÃO AUTÔNOMO EVOLUTIVO	39
3.3.1. <i>População de Regras</i>	42
3.3.2. <i>Módulo de Competição</i>	43
3.3.3. <i>Evolução</i>	44
3.3.3.1. Colisão	46

3.3.3.1.1. Módulo de Avaliação da Evolução Disparada por um Evento de Colisão	46
3.3.3.1.2. Módulo de Reprodução da Evolução Disparada por um Evento de Colisão	47
3.3.3.2. Captura	48
3.3.3.2.1. Módulo de Avaliação da Evolução Disparada por um Evento de Captura	48
3.3.3.2.2. Módulo de Reprodução da Evolução Disparada por um Evento de Captura	49
3.3.3.3. Monotonia	49
3.3.3.3.1. Módulo de Avaliação da Evolução Disparada por um Evento de Monotonia	50
3.3.3.3.2. Módulo de Reprodução da Evolução Disparada por um Evento de Monotonia	50
3.3.3.4. Taxa de Procriação	50
3.4. EXPERIMENTOS E RESULTADOS	52
3.5. CONSIDERAÇÕES FINAIS	56
3.6. DESCRIÇÃO DOS VÍDEOS	58
CAPÍTULO 4: ROBÓTICA COLETIVA INSPIRADA EM FORMIGAS.....	59
4.1. INTRODUÇÃO	59
4.2. CONTEXTUALIZAÇÃO DO CAPÍTULO	60
4.3. FUNDAMENTAÇÃO BIOLÓGICA	62
4.4. REVISÃO DE TRABALHOS RELACIONADOS.....	64
4.5. ABORDAGEM EVOLUTIVA PARA ROBÓTICA COLETIVA.....	68
4.5.1. <i>Estratégia de comunicação baseada em estigmergia</i>	68
4.5.2. <i>Mecanismo de Feromônio Artificial</i>	69
4.5.3. <i>Feromônios e Tomada de Decisão pelo Robô</i>	71
4.5.4. <i>Estratégias para Liberação de Feromônios</i>	71
4.5.4.1. Determinística	71
4.5.4.2. Evolutiva	72
4.6. SISTEMA DE NAVEGAÇÃO AUTÔNOMO	72
4.6.1. <i>Colisão</i>	74
4.6.2. <i>Captura</i>	74
4.6.3. <i>Monotonia</i>	75
4.7. EXPERIMENTOS E RESULTADOS.....	75
4.7.1. <i>Análise paramétrica</i>	77
4.7.1.1. Taxa de Evaporação.....	77
4.7.1.2. Complexidade do Sistema	79
4.7.1.3. Posição Inicial	81
4.7.1.4. Número de Robôs.....	83
4.7.2. <i>Experimentos Adicionais</i>	85
4.7.2.1. Experimento I	85
4.7.2.2. Experimento II	86
4.7.2.3. Experimento III	87
4.7.3. <i>Adaptabilidade</i>	88
4.7.4. <i>Otimização de Itinerários</i>	91
4.8. CONCLUSÕES	94
4.9. DESCRIÇÃO DOS VÍDEOS	97
CAPÍTULO 5: REDE IMUNO-CLASSIFICADORA COM APRENDIZAGEM PARA NAVEGAÇÃO AUTÔNOMA DE ROBÔS	99
5.1. INTRODUÇÃO	99
5.2. CONTEXTUALIZAÇÃO DO CAPÍTULO	101

5.3. TRABALHOS CORRELATOS.....	103
5.4. ASPECTOS BÁSICOS DA TEORIA DA REDE IMUNOLÓGICA.....	105
5.5. REDE IMUNO-CLASSIFICADORA COM APRENDIZAGEM COMO UM SISTEMA DE NAVEGAÇÃO AUTÔNOMO.....	106
5.5.1. <i>Classificadores e Conexões</i>	108
5.5.2. <i>Dinâmica da Rede para Tomada de Decisão</i>	109
5.5.3. <i>Processos Evolutivos</i>	111
5.5.3.1. Evolução dos Classificadores.....	112
5.5.3.2. Evolução da Rede.....	112
5.6. EXPERIMENTOS E RESULTADOS.....	114
5.6.1. <i>Experimentos Reativos</i>	115
5.6.2. <i>Experimentos com Mínimos Locais</i>	117
5.6.3. <i>Análise da Rede</i>	123
5.7. CONCLUSÃO.....	125
5.8. DESCRIÇÃO DOS VÍDEOS.....	127
CAPÍTULO 6: ABORDAGEM EVOLUTIVA PARA RASTREAMENTO ROBÓTICO DE ALVOS MÓVEIS.....	129
6.1. INTRODUÇÃO.....	129
6.2. CONTEXTUALIZAÇÃO DO CAPÍTULO.....	131
6.3. LOCALIZAÇÃO E POSICIONAMENTO POR RÁDIO FREQUÊNCIA.....	133
6.4. ABORDAGEM EVOLUTIVA PARA PLANEJAMENTO DE TRAJETÓRIAS E RASTREAMENTO DE ALVOS MÓVEIS.....	133
6.4.1. <i>Visão Geral</i>	133
6.4.2. <i>Codificação</i>	135
6.4.3. <i>Inicialização</i>	136
6.4.4. <i>Avaliação</i>	137
6.4.5. <i>Seleção</i>	137
6.4.6. <i>Geração de Descendentes</i>	138
6.4.7. <i>Reinício da Evolução</i>	139
6.4.8. <i>Critérios de Parada</i>	139
6.4.9. <i>Movimentos do Alvo</i>	140
6.4.10. <i>Controle de Navegação</i>	140
6.4.11. <i>Parâmetros</i>	142
6.5. EXPERIMENTOS E RESULTADOS.....	142
6.5.1. <i>Cenários Estáticos</i>	143
6.5.2. <i>Cenários Dinâmicos</i>	144
6.5.2.1. Desviando de Obstáculos Imprevistos.....	144
6.5.2.2. Rastreando Alvos Móveis.....	145
6.5.2.3. Estudo de Caso: Seguindo Paciente em um Corredor Ocupado.....	146
6.5.2.4. Estudo de Caso: Médico Visitando Quartos.....	148
6.5.3. <i>Testes Reais</i>	150
6.5.3.1. Primeiro Experimento.....	151
6.5.3.2. Segundo Experimento.....	151
6.5.3.3. Terceiro Experimento.....	152
6.5.3.4. Quarto Experimento.....	153
6.6. CONCLUSÃO.....	155
6.7. DESCRIÇÃO DOS VÍDEOS.....	156

CAPÍTULO 7: PLATAFORMA PARA INSPEÇÃO DE AMBIENTES E COLETA DE DADOS.....	157
7.1. INTRODUÇÃO.....	157
7.2. CONTEXTUALIZAÇÃO DO CAPÍTULO	159
7.3. SISTEMA DE INSPEÇÃO E COLETA DE DADOS.....	160
7.3.1. <i>Visão Geral</i>	160
7.3.2. <i>Configuração</i>	162
7.3.3. <i>Distribuição dos Pontos de Coleta</i>	162
7.3.3.1. Codificação e Inicialização	162
7.3.3.2. Avaliação	163
7.3.3.3. Operadores Evolutivos	164
7.3.3.4. Critérios de Parada	164
7.3.4. <i>Evolução do Itinerário</i>	165
7.3.4.1. Codificação e Inicialização	165
7.3.4.2. Avaliação	166
7.3.4.3. Operadores Evolutivos	167
7.3.4.4. Critérios de Parada	168
7.3.5. <i>Execução</i>	168
7.3.6. <i>Interpretação dos Dados</i>	169
7.3.7. <i>Parâmetros</i>	169
7.4. EXPERIMENTOS E RESULTADOS.....	170
7.4.1. <i>Análise Paramétrica</i>	171
7.4.2. <i>Distribuição de Pontos de Coleta</i>	172
7.4.3. <i>Simulações</i>	173
7.4.4. <i>Experimentos Reais</i>	177
7.5. CONCLUSÃO	182
7.6. DESCRIÇÃO DOS VÍDEOS	185
CAPÍTULO 8: CONCLUSÃO	187
8.1. VISÃO GERAL.....	187
8.2. REVISÃO E DISCUSSÃO	187
8.3. CONSIDERAÇÕES FINAIS.....	191
REFERÊNCIAS BIBLIOGRÁFICAS	193
ÍNDICE DE TRABALHOS CITADOS.....	201

Índice de Figuras

FIGURA 1.1: (A) VEÍCULO VENCEDOR DO DARPA GRAND CHALLENGE 2008, (B) ROOMBA (ROBÔ ASPIRADOR DE PÓ), (C) ROBÔ SPIRIT DA NASA, E (D) O SUCESSO DE CINEMA WALL-E.	3
FIGURA 2.1: CARACTERÍSTICAS E DISTINÇÕES ENTRE SISTEMAS COM COMPORTAMENTO REATIVO E DELIBERATIVO.	16
FIGURA 2.2: ORGANIZAÇÃO TÍPICA DE UM SISTEMA HÍBRIDO REATIVO-DELIBERATIVO.....	21
FIGURA 2.3 : MINI-ROBÔ KHEPERA II, CUJO TAMANHO É DE 80 MM X 30 MM.	26
FIGURA 2.4: DISTRIBUIÇÃO DOS SENSORES NA ESTRUTURA DO ROBÔ, CUJA FRENTE É APONTADA PELA SETA.	26
FIGURA 2.5: FOTO DO ROBÔ PATROLBOT, INDICANDO SEUS PRINCIPAIS SENSORES E DISPOSITIVOS. VEJA TAMBÉM O VÍDEO 2.1.....	27
FIGURA 3.1: ALGORITMO SIMPLIFICADO DO SISTEMA CLASSIFICADOR IMPLEMENTADO NESTE TRABALHO.	39
FIGURA 3.2: DIAGRAMA QUE REPRESENTA A ESTRUTURA E FUNCIONAMENTO DO SISTEMA DE CONTROLE.....	40
FIGURA 3.3: ORGANIZAÇÃO E FLUXO DO PROCESSO EVOLUTIVO RESPONSÁVEL PELA APRENDIZAGEM DO SNA.	41
FIGURA 3.4: EXEMPLO DO CROMOSSOMO DE UMA REGRA CUJOS CONSEQÜENTES DETERMINAM GIRO DE -3° E AUMENTO DE VELOCIDADE.	42
FIGURA 3.5: EXEMPLO QUE ILUSTRA COMO AS REGRAS COMPETEM PARA QUE SEJA DETERMINADA A AÇÃO USADA PELO SISTEMA.	44
FIGURA 3.6: EXPERIMENTO SIMULADO EM AMBIENTE COM DUAS REGIÕES DE ALVOS E UM OBSTÁCULO ENTRE ELAS.....	53
FIGURA 3.7: AMBIENTE REAL COM UM OBSTÁCULO ENTRE O ROBÔ E A FONTE DE LUZ.	53
FIGURA 3.8: DESEMPENHO DO SISTEMA EM AMBIENTE REAL COM UM OBSTÁCULO CENTRAL.	54
FIGURA 3.9: PERFORMANCE DO ROBÔ COM APRENDIZAGEM TOTALMENTE EM AMBIENTE REAL.	54
FIGURA 3.10: AMBIENTE REAL COM TRÊS OBSTÁCULOS.....	55
FIGURA 3.11: ATUAÇÃO DO ROBÔ EM UM AMBIENTE COMPLEXO (FIGURA 3.10).	56
FIGURA 4.1: ESTRUTURA (CROMOSSOMO) DAS REGRAS UTILIZADAS PELO SISTEMA.	73
FIGURA 4.2: AMBIENTE COM TRILHAS DE FEROMÔNIO (ESQUERDA) E GRÁFICO DA VARIAÇÃO DA DISTÂNCIA MÉDIA ENTRE CAPTURAS (DIREITA), PARA A SIMULAÇÃO COM MELHORES RESULTADOS USANDO A ABORDAGEM EVOLUTIVA (VEJA TAMBÉM O VÍDEO 4.1). ..	78
FIGURA 4.3: AMBIENTE COM TRILHAS DE FEROMÔNIO (ESQUERDA) E GRÁFICO DA VARIAÇÃO DA DISTÂNCIA MÉDIA ENTRE CAPTURAS (DIREITA), PARA A SIMULAÇÃO COM MELHORES RESULTADOS USANDO A ABORDAGEM DETERMINÍSTICA.	79
FIGURA 4.4: CONFORMAÇÃO FINAL DAS TRILHAS DE FEROMÔNIO PARA A SIMULAÇÃO COM MELHORES RESULTADOS USANDO A VERSÃO EVOLUTIVA.	81
FIGURA 4.5: SIMULAÇÕES EM QUE OS ROBÔS FORAM INICIALMENTE POSICIONADOS NO 1º QUADRANTE (ALTA ESQUERDA), 2º QUADRANTE (ALTA DIREITA), 3º QUADRANTE (BAIXA ESQUERDA) E 4º QUADRANTE (BAIXA DIREITA).	82
FIGURA 4.6: EXPERIMENTO EVOLUTIVO (ESQUERDA) E DETERMINÍSTICO (DIREITA) REFERENTES ÀS MELHORES SIMULAÇÕES COM 12 ROBÔS. NOTE QUE OS ROBÔS NÃO SÃO MOSTRADOS À DIREITA APENAS POR UMA FALHA NA CAPTURA DA IMAGEM, SEM OUTRAS IMPLICAÇÕES.....	84
FIGURA 4.7: EXPERIMENTO EVOLUTIVO (ESQUERDA) E DETERMINÍSTICO (DIREITA) REFERENTES ÀS MELHORES SIMULAÇÕES COM 20 ROBÔS.	85
FIGURA 4.8: MELHORES SIMULAÇÕES PARA OS EXPERIMENTOS COM ABORDAGEM EVOLUTIVA (ESQUERDA) E DETERMINÍSTICA (DIREITA). ..	86
FIGURA 4.9: MELHORES SIMULAÇÕES PARA OS EXPERIMENTOS COM ABORDAGEM EVOLUTIVA (ESQUERDA) E DETERMINÍSTICA (DIREITA). ..	87
FIGURA 4.10: MELHORES SIMULAÇÕES PARA OS EXPERIMENTOS COM ABORDAGEM EVOLUTIVA (ESQUERDA) (VEJA TAMBÉM O VÍDEO 4.2) E DETERMINÍSTICA (DIREITA).....	88
FIGURA 4.11: SIMULAÇÃO COM A ABORDAGEM EVOLUTIVA ANTES (ESQUERDA) E DEPOIS (DIREITA) DO NOVO OBSTÁCULO CENTRAL SER INSERIDO ENTRE OS DOIS ALVOS.	89
FIGURA 4.12: GRÁFICOS DA VARIAÇÃO DA DISTÂNCIA MÉDIA PARA A SIMULAÇÃO COM AUSÊNCIA DE COMUNICAÇÃO ANTES (ESQUERDA) E DEPOIS (DIREITA) DO NOVO OBSTÁCULO SER INSERIDO ENTRE OS ALVOS.	90
FIGURA 4.13: INSTÂNCIA ARTIFICIAL DO PROBLEMA DO CAIXEIRO VIAJANTE COM 12 CIDADES.....	92

FIGURA 4.14: CIRCUITO ÓTIMO OBTIDO PARA A INSTÂNCIA COM 12 CIDADES DO PROBLEMA DO CAIXEIRO VIAJANTE EM UM CENÁRIO DE ROBÓTICA COLETIVA, USANDO O SNA COM A ABORDAGEM EVOLUTIVA. OCORRERAM 64 COLISÕES ENTRE ROBÔS, 32 COLISÕES EM OBSTÁCULOS, 182 EVENTOS DE MONOTONIA, E 5434 CAPTURAS.....	93
FIGURA 4.15: SIMULAÇÃO COM 8 ROBÔS CONTROLADOS PELO SNA COM COMUNICAÇÃO EVOLUTIVA EM UMA INSTÂNCIA REAL DO PROBLEMA DO CAIXEIRO VIAJANTE COM 29 CIDADES. NO RESULTADO À ESQUERDA, OS ROBÔS PUDEAM ESCOLHER LIVREMENTE A SEQÜÊNCIA DE ALVOS A CAPTURAR. JÁ À DIREITA, A SEQÜÊNCIA ÓTIMA DE CIDADES CONHECIDA FOI IMPOSTA AOS ROBÔS.....	94
FIGURA 5.1: EXEMPLO DE UMA REDE IMUNOLÓGICA SIMPLES EM QUE O ANTICORPO 1 ESTIMULA O ANTICORPO 2 E SUPRIME O ANTICORPO 3; O ANTICORPO 2 SE AUTO ESTIMULA; O ANTICORPO 3 É SUPRIMIDO PELO ANTICORPO 1 E SUPRIME O ANTICORPO 4; E O ANTICORPO 4 ESTIMULA OS ANTICORPOS 1 E 3.	106
FIGURA 5.2: COMPONENTES DO SISTEMA DE CONTROLE DA REDE IMUNO-CLASSIFICADORA COM APRENDIZAGEM.....	107
FIGURA 5.3: O CROMOSSOMO DE UMA REGRA EM QUE AS PARTES CONSEQÜENTES RD E RV DETERMINAM UM AJUSTE DE DIREÇÃO DE 3° E AUMENTO DA VELOCIDADE, RESPECTIVAMENTE. RC INDICA QUE ESTE INDIVÍDUO ESTÁ CONECTADO AOS CLASSIFICADORES 3, 7, 12 E 75.	109
FIGURA 5.4: PROCESSO DE EVOLUÇÃO DA ARQUITETURA DA REDE NO SISTEMA ILCN.....	113
FIGURA 5.5: COMPORTAMENTO FINAL DE NAVEGAÇÃO OBSERVADO PARA O ILCN (À ESQUERDA) E O SCA DO CAPÍTULO 3 (À DIREITA) NO AMBIENTE REATIVO.	116
FIGURA 5.6: O MELHOR SISTEMA ILCN EVOLUÍDO EM SIMULAÇÃO FOI TRANSFERIDO PARA O ROBÔ REAL PARA SER TESTADO. PARTE DESTA EXPERIMENTO PODE SER VISTO NO VÍDEO 5.1.....	117
FIGURA 5.7: COMPORTAMENTO DE NAVEGAÇÃO DO ILCN EVOLUÍDO, COM FITNESS 16 (À ESQUERDA) E 40 (À DIREITA).....	119
FIGURA 5.8: À ESQUERDA, O ILCN EVOLUÍDO CONSEGUE ESCAPAR DO BECO. À DIREITA, O MESMO SISTEMA, MAS SEM CONEXÕES ENTRE CLASSIFICADORES, ENTRA EM UM CICLO INFINITO POR MOTIVO DO MÍNIMO LOCAL.	120
FIGURA 5.9: AMBIENTES MONTADOS PARA EXPERIMENTAÇÃO REAL DA REDE IMUNO-CLASSIFICADORA. NO AMBIENTE DA ESQUERDA SE TEM UM BECO CURTO E, À DIREITA, UM BECO MAIS LONGO. A LÂMPADA É O ALVO A SER CAPTURADO PELO ROBÔ.	121
FIGURA 5.10: SEQÜÊNCIA DE QUADROS RETIRADA DO VÍDEO 5.3 QUE MOSTRA O ROBÔ ESCAPANDO POR DUAS VEZES DO BECO.	122
FIGURA 5.11: SEQÜÊNCIA DE QUADROS PROVENIENTE DO VÍDEO 5.4 EM QUE O ROBÔ ESCAPA DE UM BECO MAIS LONGO E ATINGE A FONTE LUMINOSA.	122
FIGURA 5.12: ARQUITETURA DA REDE EVOLUÍDA DO ILCN CAPAZ DE CAPTURAR 4 ALVOS A CADA MIL ITERAÇÕES. HÁ 80 CLASSIFICADORES E 337 CONEXÕES. OS NÓS ACIMA E ABAIXO SÃO OS MESMOS, SENDO ASSIM DISPOSTOS PARA MELHOR VISUALIZAÇÃO.....	123
FIGURA 5.13: MESMA REDE DA FIGURA ANTERIOR MOSTRADA DE UMA FORMA DIFERENTE. OS NÓS COM MAIOR GRAU DE CONECTIVIDADE SÃO REPRESENTADOS POR CÍRCULOS MAIORES. OS NÓS 37, 29 E 54 SÃO OS MAIS ATIVADOS DURANTE A NAVEGAÇÃO DO ROBÔ, AO MESMO TEMPO EM QUE OS NÓS 24 E 61 SÃO OS MAIS IMPORTANTES GLOBALMENTE. A POSIÇÃO DOS NÓS NA FIGURA FOI DEFINIDA ARBITRARIAMENTE PELO SOFTWARE DE VISUALIZAÇÃO UTILIZADO (NETDRAW®).	124
FIGURA 6.1: DIAGRAMA QUE REPRESENTA A ORGANIZAÇÃO E FUNCIONAMENTO DA ABORDAGEM PARA RASTREAMENTO.....	134
FIGURA 6.2: EXEMPLO DE TRAJETÓRIA COM TRÊS NÓS INTERMEDIÁRIOS (JUNTAS) E MARGENS PARALELAS.	135
FIGURA 6.3: ILUSTRAÇÃO QUE MOSTRA COMO OS NÓS SÃO INICIALMENTE INSERIDOS PARA CRIAR UMA TRAJETÓRIA GLOBAL.	137
FIGURA 6.4: ILUSTRAÇÃO QUE EXEMPLIFICA COMO SE DÁ O PLANEJAMENTO E NAVEGAÇÃO DE TRAJETÓRIAS PELO ROBÔ.....	141
FIGURA 6.5: EVOLUÇÃO DE TRAJETÓRIAS GLOBAIS EM DOIS CENÁRIOS ESTÁTICOS EM UM HOSPITAL.	143
FIGURA 6.6: EXPERIMENTO EM AMBIENTE DINÂMICO EM QUE O ROBÔ DESVIOU DE OBSTÁCULOS IMPREVISTOS (LINHAS QUE ATRAVESSAM A TRAJETÓRIA PLANEJADA).....	144
FIGURA 6.7: EXPERIMENTO EM QUE O ROBÔ SEGUE UM ALVO MÓVEL.....	146
FIGURA 6.8: SIMULAÇÃO DO ROBÔ (CÍRCULO) SEGUINDO UM PACIENTE (RETÂNGULO) POR UM CORREDOR LOTADO (OBSTÁCULOS EM VERMELHO) DE 100 M DE COMPRIMENTO. AS CENAS NUMERADAS FORAM CAPTURADAS AO LONGO DO EXPERIMENTO, SENDO QUE A IMAGEM MAIS ABAIXO MOSTRA A TRAJETÓRIA COMPLETA PERCORRIDA PELO ROBÔ.	147
FIGURA 6.9: SIMULAÇÃO COM OBSTÁCULOS (LINHAS VERMELHAS) QUE SE MOVIAM EM DIREÇÃO AO ROBÔ.	148
FIGURA 6.10: SIMULAÇÃO DO ROBÔ RASTREANDO UM MÉDICO (NÃO MOSTRADO) POR 16 QUARTOS.....	150
FIGURA 6.11: ROBÔ SEGUINDO UMA PESSOA EM UMA SALA. A PESSOA APENAS SE MOVE QUANDO O ROBÔ A ALCANÇA (VÍDEO 6.1). ..	151

FIGURA 6.12: EXPERIMENTO EM QUE O ROBÔ RASTREIA O CÂMERA POR UM CORREDOR ESTREITO, DESVIANDO DE OBSTÁCULOS (VÍDEO 6.2).	152
FIGURA 6.13: A PESSOA CARREGANDO A TAG É SEGUIDA CONTINUAMENTE PELO ROBÔ POR UM CORREDOR DE 20 METROS (VÍDEO 6.3).	153
FIGURA 6.14: EXPERIMENTO EM QUE O ROBÔ FAZ RASTREAMENTO EM MEIO A DIVERSAS PESSOAS E OBSTÁCULOS NO HOSPITAL (VÍDEO 6.4).	154
FIGURA 7.1: ARQUITETURA DA PLATAFORMA DE INSPEÇÃO E COLETA DE DADOS, ORGANIZADA EM TRÊS CAMADAS.	161
FIGURA 7.2: UM EXEMPLO DE INDIVÍDUO, SEU GENÓTIPO E FENÓTIPO.	166
FIGURA 7.3: ESCALA DE CORES USADA PARA INDICAR A QUALIDADE DO SINAL MEDIDO, SENDO ZERO (AZUL), O MELHOR RESULTADO POSSÍVEL.	170
FIGURA 7.4: AMBIENTE UTILIZADO PARA OS EXPERIMENTOS DE ANÁLISE PARAMÉTRICA. O ITINERÁRIO MOSTRADO FOI O MELHOR OBTIDO, TOTALIZANDO 118,5 METROS DE EXTENSÃO.	171
FIGURA 7.5: SIMULAÇÕES QUE MOSTRAM O RESULTADO DA DISTRIBUIÇÃO EVOLUTIVA DE LOCAIS DE COLETA COM DENSIDADE DE 1 PTO/M ² , À ESQUERDA, E 3,9 PTO/M ² , À DIREITA.	173
FIGURA 7.6: AMBIENTE DA PRIMEIRA SIMULAÇÃO COM DUAS ANTENAS VIRTUAIS, CUJO RAIOS DE ALCANCE É IGUAL A 5 METROS.	174
FIGURA 7.7: MAPA COM OS RESULTADOS DA COLETA DE DADOS FEITA NA PRIMEIRA SIMULAÇÃO.	175
FIGURA 7.8: À ESQUERDA, AMBIENTE COM QUATRO ANTENAS DE 5 M DE ALCANCE E O ITINERÁRIO PLANEJADO PARA A SEGUNDA SIMULAÇÃO. À DIREITA, O MAPA DA QUALIDADE DO SINAL OBTIDO APÓS A INSPEÇÃO COM O ROBÔ.	175
FIGURA 7.9: MAPA DO SINAL COM EXCELENTE QUALIDADE (DISTÂNCIA DAS ANTENAS SEMPRE MENOR QUE 2 M) (DIREITA), PRODUZIDO A PARTIR DA SIMULAÇÃO DE COLETA DE DADOS REALIZADA EM UM AMBIENTE COM ALTA DENSIDADE DE TRANSMISSORES (ESQUERDA).	176
FIGURA 7.10: CENAS NÃO SEQÜENCIAIS CAPTURADAS DURANTE O PROCESSO DE INSPEÇÃO E COLETA DE DADOS REALIZADO NA SALA DE EMERGÊNCIA, AS QUAIS FAZEM PARTE DO VÍDEO 7.1.	178
FIGURA 7.11: RESULTADOS RELATIVOS AO ERRO ABSOLUTO NOS DOIS EXPERIMENTOS REAIS, SENDO CONSIDERADOS AQUI APENAS OS PONTOS ONDE O ROBÔ FICOU PARADO POR 5 SEGUNDOS OU MAIS. EM (A) E (B) MOSTRAM-SE MAPAS DOS DADOS COLETADOS NO PRIMEIRO E SEGUNDO EXPERIMENTOS, RESPECTIVAMENTE. O GRÁFICO (D) MOSTRA A MELHORA ACENTUADA DA QUALIDADE DOS RESULTADOS EM COMPARAÇÃO AO GRÁFICO DO PRIMEIRO TESTE (C).	179
FIGURA 7.12: RESULTADOS REFERENTES AO ERRO ABSOLUTO NAS DUAS INSPEÇÕES NA SALA DE EMERGÊNCIA, SENDO EXIBIDAS TODAS AS AMOSTRAS COLETADAS, INCLUSIVE COM O ROBÔ EM MOVIMENTO. (A) E (B) REPRESENTAM AS AMOSTRAS EM FORMA DE MAPA PARA O PRIMEIRO E SEGUNDO EXPERIMENTOS, RESPECTIVAMENTE. NOVAMENTE NOTA-SE UM MAIOR NÚMERO DE REGIÕES COM MENOR NÍVEL DE ERRO MÉDIO NO GRÁFICO (D), CORRESPONDENTE AO SEGUNDO TESTE, DO QUE NO GRÁFICO DO PRIMEIRO TESTE (C).	180
FIGURA 7.13: RESULTADOS QUE INDICAM O DESVIO PADRÃO DO ERRO RELATIVO (ENTRE TAGS) NAS DUAS INSPEÇÕES, TENDO SIDO USADOS APENAS DADOS DOS LOCAIS EM QUE O ROBÔ ESTEVE POR 30 S OU MAIS. OS MAPAS DA PRIMEIRA E SEGUNDA COLETAS SÃO APRESENTADOS EM (A) E (B), RESPECTIVAMENTE. PERCEBE-SE NOS GRÁFICOS (C), DO PRIMEIRO EXPERIMENTO, E (D), DO SEGUNDO, QUE HÁ UM AUMENTO DE 80,4% PARA 91,8% NO NÚMERO DE REGIÕES COM DESVIO PADRÃO MENOR QUE 2 METROS.	181

Índice de Vídeos

VÍDEO 2.1: UMA RÁPIDA VISUALIZAÇÃO DO ROBÔ PATROLBOT.	36
VÍDEO 3.1: SIMULAÇÃO QUE MOSTRA O SISTEMA APRENDENDO A DESVIAR DE OBSTÁCULOS E CAPTURAR ALVOS. AS COLISÕES SÃO INDICADAS POR QUADRADOS VERMELHOS, AS CAPTURAS DE ALVOS POR QUADRADOS VERDES E OS EVENTOS DE MONOTONIA SÃO SINALIZADOS POR QUADRADOS DOURADOS.	58
VÍDEO 4.1: SIMULAÇÃO EM QUE OS ROBÔS DEVEM CAPTURAR DOIS ALVOS ALTERNADAMENTE, BUSCANDO ENCONTRAR O CAMINHO MAIS CURTO ENTRE ELAS POR MEIO DO USO DE FEROMÔNIOS ARTIFICIAIS. O SISTEMA COM MODO DE COMUNICAÇÃO EVOLUTIVO É ADOTADO. O GRÁFICO NA PARTE ALTA DA IMAGEM INDICA COMO A DISTÂNCIA MÉDIA REALIZADA PELOS ROBÔS ENTRE OS DOIS ALVOS É REDUZIDA AO LONGO DA SIMULAÇÃO.	97
VÍDEO 4.2: DIVERSOS ROBÔS DEVEM CAPTURAR SEIS ALVOS SEQUENCIAL E CONTINUAMENTE NESTA SIMULAÇÃO. PODE-SE VER NO VÍDEO QUE, POR MEIO DA UTILIZAÇÃO DOS FEROMÔNIOS ARTIFICIAIS, OS ROBÔS SÃO CAPAZES DE ENCONTRAR OS CAMINHOS MAIS CURTOS ENTRE OS ALVOS, O QUE FAZ COM QUE A DISTÂNCIA MÉDIA ENTRE AS CAPTURAS DECAIA CONFORME A CONCENTRAÇÃO DAS TRILHAS AUMENTA (MOSTRADO NO GRÁFICO).	97
VÍDEO 5.1: PARTE DO EXPERIMENTO EM QUE O SISTEMA CONEXIONISTA (ILCN) É TESTADO EM UM AMBIENTE SEM MÍNIMOS LOCAIS...	127
VÍDEO 5.2: SIMULAÇÃO QUE MOSTRA O COMPORTAMENTO DO ROBÔ CONTROLADO MELHOR REDE DE CLASSIFICADORES EVOLUÍDA (FITNESS IGUAL A 40) EM UM AMBIENTE COM DOIS BECOS. AS BARRAS MOSTRADAS NA PARTE BAIXA DO VÍDEO INDICAM A VARIAÇÃO DA CONCENTRAÇÃO DOS CLASSIFICADORES DE ACORDO COM A NAVEGAÇÃO DO ROBÔ.	127
VÍDEO 5.3: EXPERIMENTO EM QUE O SISTEMA OBTIDO VIA SIMULAÇÃO É TESTADO EM UM AMBIENTE REAL COM BECOS. TÃO LOGO O ROBÔ ALCANÇA A FONTE LUMINOSA, A ARMADILHA É INVERTIDA E A LÂMPADA MOVIDA PARA O LADO CONTRÁRIO DA ARENA.	127
VÍDEO 5.4: EXPERIMENTO EM QUE O AMBIENTE POSSUI UM BECO LONGO, TORNANDO MAIS DIFÍCIL PARA QUE O ROBÔ CONTROLADO PELA REDE IMUNO-CLASSIFICADORA COM APRENDIZAGEM ESCAPE DELE.	127
VÍDEO 6.1: TESTE REALIZADO EM UMA SALA PEQUENA, SENDO QUE A PESSOA SE MOVE PARA UMA POSIÇÃO E ESPERA ATÉ QUE O ROBÔ CHEGUE LÁ PARA ENTÃO SE ENCAMINHAR ATÉ O PRÓXIMO LOCAL. O ROBÔ ALCANÇA A PESSOA QUATRO VEZES.	156
VÍDEO 6.2: O EXPERIMENTO EXIGE QUE O ROBÔ SIGA A PESSOA (NO CASO O PRÓPRIO CÂMERA MAN) POR UM CORREDOR ESTREITO. TESTA-SE A CAPACIDADE DE DESVIO DE OBSTÁCULOS DO SISTEMA, JÁ QUE VÁRIOS OBSTÁCULOS SÃO COLOCADOS DE FORMA SURPRESA NO CAMINHO DO ROBÔ.	156
VÍDEO 6.3: O EXPERIMENTO TEVE COMO OBJETIVO DE FAZER O ROBÔ SEGUIR A PESSOA COM A TAG POR UM CORREDOR DE APROXIMADAMENTE 20 METROS DE COMPRIMENTO. UM DETALHE QUE ILUSTRA BEM COMO A DINÂMICA DO AMBIENTE TRAZ COMPLICAÇÕES AO PROBLEMA DE NAVEGAÇÃO AUTÔNOMA PODE SER VISTO: AO SAIR DE UMA SALA, UM MÉDICO NÃO PERCEBEU A PRESENÇA DO ROBÔ, O QUE QUASE LEVOU À COLISÃO.	156
VÍDEO 6.4: EXPERIMENTO EXECUTADO EM MEIO A UMA GRANDE MOVIMENTAÇÃO DE PESSOAS, MACAS E EQUIPAMENTOS NAS CERCANIAS DA SALA DE EMERGÊNCIA. O ROBÔ TEVE QUE RASTREAR A PESSOA COM A TAG, A QUAL, EM ALGUNS ESTÁ ANDANDO E, EM OUTROS, PARADA.	156
VÍDEO 7.1: EDIÇÃO COM VÁRIAS CENAS FILMADAS DURANTE O PROCESSO DE INSPEÇÃO E COLETA DE DADOS REALIZADO NA SALA DE EMERGÊNCIA DO WASHINGTON HOSPITAL CENTER. O OBJETIVO ERA AVALIAR A QUALIDADE E ACURÁCIA DO SISTEMA DE LOCALIZAÇÃO POR RADIOFREQUÊNCIA LÁ INSTALADO.	185

Índice de Tabelas

TABELA 2.1: RESUMO DAS CARACTERÍSTICAS DE CADA FRENTE DE PESQUISA CONTIDA NESTE TRABALHO.	35
TABELA 3.1: CARACTERÍSTICAS DOS VETORES QUE COMPÕEM UM INDIVÍDUO (REGRA) DA POPULAÇÃO.	42
TABELA 3.2: INFORMAÇÕES DO MÓDULO DE AVALIAÇÃO EM CASO DE DISPARO POR COLISÃO.	47
TABELA 3.3: INFORMAÇÕES DO MÓDULO DE AVALIAÇÃO EM CASO DE DISPARO POR CAPTURA.	49
TABELA 3.4: RELAÇÃO ENTRE CC, TXP E NÚMERO DE FILHOS PRODUZIDOS POR GERAÇÃO COM POPULAÇÃO DE 200 INDIVÍDUOS.	52
TABELA 4.1: ANALOGIAS ENTRE FORMIGAS E ROBÔS NESTE TRABALHO.	69
TABELA 4.2: RESULTADOS DOS EXPERIMENTOS COM VARIAÇÃO DA TAXA DE EVAPORAÇÃO DE FEROMÔNIO.	78
TABELA 4.3: RESULTADOS OBTIDOS POR SISTEMAS DE NAVEGAÇÃO CONTENDO DE 40 A 100 CLASSIFICADORES.	80
TABELA 4.4: RESULTADOS DOS EXPERIMENTOS COM 4, 6, 8, 12 E 20 ROBÔS. UM AMBIENTE MAIOR FOI USADO PARA 20 ROBÔS.	84
TABELA 4.5: RESULTADOS DO EXPERIMENTO REALIZADO NO AMBIENTE APRESENTADO NA FIGURA 4.8.	86
TABELA 4.6: RESULTADOS DO EXPERIMENTO REALIZADO NO AMBIENTE APRESENTADO NA FIGURA 4.9.	86
TABELA 4.7: RESULTADOS DO EXPERIMENTO REALIZADO NO AMBIENTE APRESENTADO NA FIGURA 4.10.	88
TABELA 4.8: RESULTADOS OBTIDOS PARA OS EXPERIMENTOS DE ADAPTABILIDADE A MUDANÇAS NO AMBIENTE.	90
TABELA 5.1: ANALOGIAS ENTRE O SISTEMA IMUNOLÓGICO BIOLÓGICO E O SISTEMA HÍBRIDO PROPOSTO, ILCN.	107
TABELA 5.2: PROPRIEDADES DOS PROCESSOS EVOLUTIVOS DO SISTEMA PROPOSTO.	111
TABELA 5.3: RESULTADOS OBTIDOS NO EXPERIMENTO REATIVO COMPARANDO O SISTEMA CLASSIFICADOR DO CAPÍTULO 3 COM A REDE IMUNO-CLASSIFICADORA APÓS 10 SIMULAÇÕES.	116
TABELA 5.4: RESULTADOS OBTIDOS EM 10 SIMULAÇÕES NO AMBIENTE COM BECOS DO ILCN.	118
TABELA 6.1: PARÂMETROS DO SISTEMA EVOLUTIVO.	142
TABELA 6.2: DADOS DO EXPERIMENTO DA SEÇÃO 6.5.2.4, EM QUE O ROBÔ SEGUE UM MÉDICO POR 15 QUARTOS DE PACIENTES.	149
TABELA 7.1: PARÂMETROS DO SISTEMA EVOLUTIVO.	169
TABELA 7.2: RESULTADOS DA ANÁLISE PARAMÉTRICA REALIZADA PARA O SISTEMA PROPOSTO.	172

Capítulo 1: Introdução

1.1. Motivação

Especulações tecnológicas não devem, em geral, ser tomadas como garantia de acontecimentos futuros. Contudo, não é por simples acaso que diversos especialistas renomados ousam profetizar um futuro grandioso para a robótica. Fatos concretos e atuais têm mostrado que os robôs estão sim a caminho das ruas, locais de trabalho e residências das pessoas, deixando de ser populares apenas nas peças de ficção científica. Se há discordância acerca de quantos anos isto levará, 10 para uns, 15 ou 20 para outros, existe consenso em relação à importância e ao impacto da presença disseminada dos robôs no cotidiano humano. Confira algumas opiniões sintetizadas nas citações a seguir.

“Eu estou convencido que os robôs estão hoje onde os computadores estavam em 1978. Foi por volta daquele ano que os computadores começaram a aparecer em nosso meio do modo como os robôs estão surgindo agora. É claro, foram necessários ainda outros 15 anos até que os computadores realmente se difundissem nas nossas vidas. Eu penso que em 15 anos, os robôs estarão em todos os lugares, assim como se dá atualmente com o e-mail e a Internet.”

Rodney Brooks, pesquisador e co-fundador da iRobot
Technology Review, 2/2004

“O estágio atual da área de robótica se assemelha muito àquele em que se encontravam os computadores 30 anos atrás. Pode-se dizer que os braços robóticos industriais são hoje os robôs mais comuns, assim como os mainframes eram os computadores em destaque naquela época. Embora tenham sua importância, é evidente o quão limitados são (e eram) estes dispositivos. A partir do momento que os robôs se tornarem acessíveis aos consumidores, eles poderão causar um impacto tão grande quanto aquele dos PCs há três décadas.”

Bill Gates, chairman da Microsoft
Scientific American, 16/12/2006

“A próxima grande indústria, com força comparável ao boom dos computadores pessoais nos anos 80 e à revolução da internet nos 90, será a robótica. Assim como nos acostumamos a ver Bill Gates (da Microsoft) nas capas de revista dos anos 80, ou Jeff Bezos (da Amazon) nas manchetes dos anos 90 e agora a dupla dos fundadores do Google, eu aposto que dentro de 10 anos nossos heróis serão os construtores de robôs.”

Paul Saffo, especialista em tecnologias do futuro
O Estado de São Paulo, 12/02/2006

“Eu acredito que durante os próximos 20 anos nós veremos robôs autônomos em muitos aspectos da nossa vida pessoal e profissional. Pode ser que até não os reconheçamos, dado que eles podem estar embutidos em carros, eletrodomésticos e outros objetos. Robôs móveis autônomos também se tornarão cada vez mais evidentes, não apenas na exploração de planetas ou ambientes subaquáticos, mas também na execução de variados serviços na área de saúde, industrial, ambiental e doméstica. Estes serão, sem dúvida, anos estimulantes para os roboticistas.”

George Bekey, pesquisador
Autonomous Robots, 2005

Sinais expressivos desta “revolução”, que deve levar à profusão dos robôs, já são observados nos dias presentes em todas as etapas envolvidas em sua cadeia de desenvolvimento. No campo acadêmico-científico, são inúmeros os pesquisadores e projetos dedicados à robótica, sendo cada vez mais freqüente a divulgação de avanços e inovações. Um marco importante são as competições que vêm sendo organizadas pela DARPA (*Defense Advanced Research Projects Agency*) ano após ano nos EUA, desafiando cientista e engenheiros a criar veículos totalmente autônomos. Na competição de 2007, os veículos participantes deveriam navegar de forma autônoma por uma área urbana que reproduzia o tráfego de uma cidade, respeitando todas as leis de trânsito e os demais carros. O vencedor (mostrado na Figura 1.1.a), da universidade de Carnegie Mellon, foi capaz de cumprir o percurso de 96 km em pouco mais de 4 horas, sem acidentes e, o mais importante, sem qualquer intervenção humana.

Na indústria, os robôs já são bastante comuns e, segundo a Federação Internacional da Robótica, o número de unidades tem crescido a uma média anual de 5,5%, com uma estimativa total de mais de um milhão de robôs em funcionamento em 2010. Só para se ter uma idéia, a fábrica mais moderna da BMW, em Leipzig, Alemanha, possui, somente na seção responsável pela produção do chassi do carro, cerca de 700 braços robóticos atuando de forma automática, o que deixa apenas 3% das atividades para operários humanos. Graças à presença maciça de robôs, neste e em outros setores da fábrica, ela pode alcançar uma taxa de montagem de impressionantes 700 carros por dia.

Outra área fundamental para a popularização dos robôs é a comercial, em que soluções para uso doméstico estão no foco das atenções. Uma popularização jamais vista em qualquer produto robótico à venda para pessoas comuns foi obtida por um robô-aspirador de pó (veja a Figura 1.1.b). O Roomba (produzido pela empresa iRobot) tornou-se o robô mais vendido da história, atingindo uma marca de mais de 2,5 milhões de unidades desde seu lançamento, em 2002, até 2008.

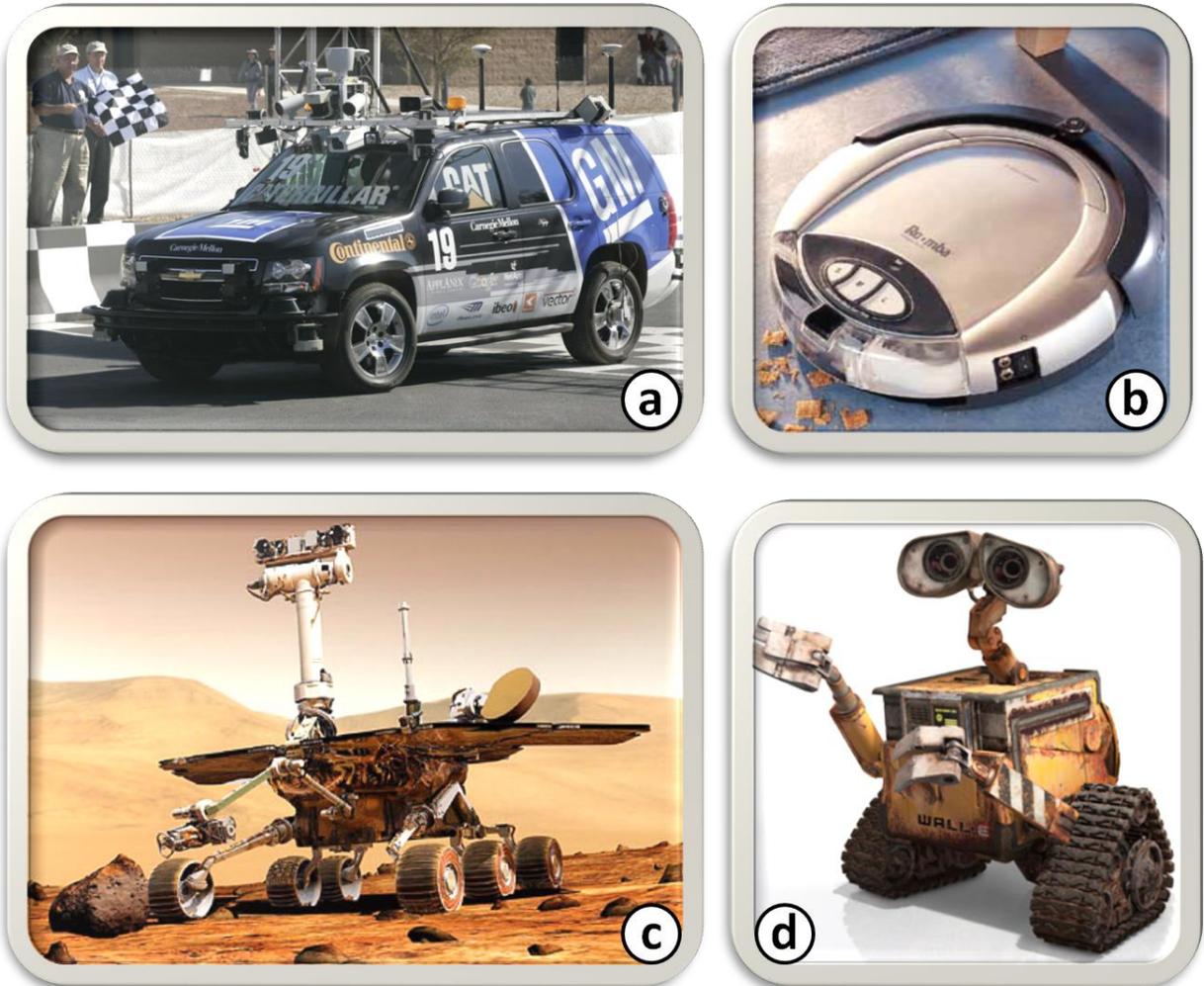


Figura 1.1: (a) Veículo vencedor do DARPA Grand Challenge 2008, (b) Roomba (robô aspirador de pó), (c) robô Spirit da NASA, e (d) o sucesso de cinema Wall-E.

Até o espaço sideral está sendo “dominado” pelos robôs, como o Spirit (ilustrado na Figura 1.1.c) e Opportunity. Enviados pela NASA ao planeta Marte em 2003, e ainda ativos, eles têm possibilitado a obtenção de informações até então inalcançáveis em lugares nunca antes explorados. Por fim, vale mencionar que, nos filmes de cinema, meio sempre repleto de ficções com robôs, o interesse por tais personagens continua alto. Em 2008, Wall-E (veja o robô na Figura 1.1.d), um filme que mostra um mundo que vive em função dos robôs, faturou quase 500 milhões de dólares.

O presente e o futuro da robótica, seu potencial e impacto esperado são temas populares, inclusive fora do meio especializado. Mas, por que a robótica é importante e quais as razões que têm colocado os robôs como um dos grandes avanços tecnológicos vindouros? É necessário dar um passo atrás em busca destas respostas.

A necessidade e conveniência dos robôs estão longe de se basear em razões apenas abstratas ou ideológicas. Os robôs não são uma criação arbitrária ou oportunista, mas sim o próximo passo natural no processo de automatização por que praticamente todas as atividades humanas vêm passando. Em particular, a informática e os sistemas de informação têm abrangido e reinventado inúmeros tipos de processos antes manuais ou semi-automatizados. Esta revolução tem se dado graças à invenção e disseminação dos computadores pessoais. No entanto, certos processos ou atividades, principalmente aqueles que requerem mobilidade e interação física com o ambiente externo, como deslocamento de objetos, por exemplo, estão além do alcance dos computadores e seus sofisticados sistemas. Faz-se necessário algo que possa, então, estender o potencial da computação para tais atividades, para que, desta forma, a evolução da automatização siga seu caminho natural e continue facilitando a vida das pessoas.

Para ilustrar a argumentação acima, o seguinte exemplo pode ser utilizado. O autor desta tese trabalha em um grupo (*Microsoft Health Solutions Group*) que desenvolve um sistema de informações, chamado Microsoft Amalga®, voltado para o uso corporativo em instituições de saúde. O sistema integra inúmeros tipos de dados provenientes de diferentes fontes, como equipamento de eletrocardiograma, raio-X, exames de laboratórios, etc., os organiza e disponibiliza para os usuários. Embora o Microsoft Amalga® seja bastante sofisticado e automatizado, ele depende de intervenção humana para muitas tarefas. Por exemplo, algo simples e rotineiro como medir a temperatura de pacientes deve ser feito por uma enfermeira, que então registrará tais informações no sistema de forma manual. Seja como coletores ou muniadores de informação (sensores), ou como executores de ações requisitadas (atuadores), são evidentes a relevância e adequação dos robôs quando integrados a sistemas de informação. Neste contexto, o robô pode ser visto como uma extensão de sistemas informatizados, automatizando tarefas infactíveis por meio de software ou que requeiram intervenção humana.

Mesmo que as perspectivas em relação à robótica sejam animadoras e otimistas, é fundamental esclarecer que elas partem do princípio de que as limitações existentes nos robôs atuais deverão ser superadas nos próximos anos. Mesmo os robôs hoje considerados bem-sucedidos ainda apresentam comportamentos relativamente primitivos e falhas relativamente ingênuas. O Roomba (robô aspirador-de-pó), por exemplo, apesar de ser um eficiente auxiliar na limpeza doméstica, se perde facilmente (paralisando sua operação) caso seja deslocado enquanto navega. Manipuladores robóticos industriais usados em montadoras de carros, por sua vez, deixam de funcionar adequadamente caso as peças

necessárias sejam fornecidas fora da ordem pré-programada ou sem sincronia perfeita com os movimentos repetitivos do robô (Wang, Tan, & Chew, 2006).

As tecnologias atuais impõem restrições tanto em relação ao corpo do robô (hardware) quanto à inteligência que o controla (software). Não há dúvidas de que, em casos como os mencionados acima, um hardware mais avançado, e.g. sensores adaptáveis, poderia resultar em melhoras de desempenho. Todavia, a constituição de um robô vai muito além de sua estrutura física, o que permite caracterizá-lo como uma máquina que sente, pensa e atua. Para sentir, é preciso sensores, para atuar, atuadores e, para pensar, há necessidade de uma capacidade de processamento que demonstre certas propriedades cognitivas, as quais fomentam a inteligência do robô. Independentemente de quais sejam os mecanismos físicos usados para operar o robô e artefatos responsáveis por manipular o mundo externo, o componente de maior importância para um robô autônomo são os princípios computacionais que o governam (Dudek & Jenkin, 2000).

Embora sensores e atuadores sejam essenciais no sentido de permitir a interação do robô com o ambiente externo, sem uma “inteligência” suficientemente sofisticada, o hardware acaba por tornar-se pouco útil. De forma análoga à inteligência natural dos seres vivos, a inteligência artificial de um robô é a grande responsável por capacitá-lo a atuar de modo autônomo, robusto e eficaz. Com base na impressionante evolução da capacidade de processamento e sua maior disponibilidade, Bekey (2005) constata que as limitações nas habilidades cognitivas dos robôs, e conseqüentemente na capacidade deles de executar tarefas cada vez mais complexas, se devem muito mais ao software do que ao hardware.

Tamanho potencial e relevância do componente inteligência dentro da engenharia de robôs motivam e justificam a intensa dedicação da comunidade científica a este tema. A transposição das dificuldades atuais e a conseqüente ascensão da coadjuvante robótica a um papel principal estão intimamente vinculadas à evolução das capacidades dos sistemas que controlam os robôs. Somente uma inteligência com nível suficientemente sofisticado permitirá que os prognósticos citados no início do capítulo se concretizem. Almejando que tal nível de avanço seja atingido o quanto antes, o que colaboraria com as previsões mais otimistas, este trabalho vem se unir a este esforço científico com o objetivo de produzir contribuições na pesquisa e desenvolvimento de sistemas inteligentes para controle autônomo de robôs móveis.

1.2. Robótica Móvel e Controle Autônomo

O estudo de robôs móveis é, por natureza, multidisciplinar. O design do veículo e seus mecanismos envolve a engenharia mecânica. A engenharia elétrica atua na integração dos componentes eletrônicos, nos sensores e nas comunicações. O sistema de controle e a “inteligência” por trás dele são objetos de pesquisa da ciência da computação. Outras ciências, como a psicologia cognitiva e a neurociência, têm papel relevante no entendimento de como organismos biológicos resolvem problemas similares (Dudek & Jenkin, 2000).

Embora este trabalho abranja diversos campos de estudo, sua raiz se encontra na área de controle de sistemas. Busca-se, nesta seção, situá-lo dentro desta grande e complexa disciplina, tentando evidenciar o posicionamento do trabalho em uma divisão da área de controle em que a pesquisa é extremamente ativa, havendo muita divergência e pouca padronização: o controle inteligente.

A área conhecida como controle se divide em diversas subáreas, sendo boa parte delas, principalmente aquelas com forte base em modelos matemáticos, amplamente estudada e bem consolidada (para uma revisão completa e detalhada da área de controle, veja Aguirre (2007)). As teorias de controle mais bem estabelecidas, como o controle linear, passando pelo não-linear até o adaptativo, têm sido aplicadas a diferentes problemas de robótica há décadas (Dorf & Bishop, 2008).

Na natureza, assim como em problemas reais complexos, o caráter não-linear é preponderante (Bekey, 2005). Construir modelos matemáticos em tais cenários é, em geral, muito complicado ou mesmo inviável, o que desfavorece métodos de controle convencionais. Em robótica móvel, a história não é diferente. Por serem capazes de navegar, robôs móveis operam em um espaço muito maior (potencialmente infinito) do que o espaço acessível a partir de um único ponto fixo do ambiente. Atuar em tais espaços requer lidar não só com aquisição incremental de conhecimento, estimação de erros, habilidade de reconhecer objetos ou locais importantes e resposta em tempo-real, mas também que todas estas funcionalidades operem em consonância (Dudek & Jenkin, 2000).

Uma crítica comum às abordagens de controle tradicionais é que elas são projetadas para atuar em cenários específicos e limitados, sendo que, como mencionado no parágrafo anterior, na realidade espera-se que robôs móveis autônomos operem em variados tipos de ambientes e situações. Assim, uma única estratégia fixa de controle pode não ser satisfatória na perspectiva global. Uma solução óbvia é utilizar múltiplos controladores, chaveando entre eles de acordo com a necessidade. Esta é uma forma

de controle adaptativo. Outra possibilidade contempla a inclusão de capacidades de aprendizagem no controlador, de modo que as experiências sofridas pelo robô permitam que o sistema se ajuste automaticamente.

Devido à grande variedade de situações com que o robô pode se defrontar ao longo da navegação, sendo algumas não previsíveis, o desempenho de um sistema de navegação depende de sua capacidade de aprendizagem e adaptação. Ou seja, somente aprimorando sua estratégia de navegação, por meio da incorporação de conhecimento (adquirido por experimentação) e ajuste de parâmetros, um sistema de navegação torna-se apto a guiar eficientemente o robô visando maximizar o atendimento dos objetivos de navegação. Justamente devido à importância da aprendizagem, intensos esforços de pesquisa têm sido dedicados ao aperfeiçoamento desta potencialidade em sistemas de navegação (Figueiredo, 1999). Já no tocante aos mecanismos de adaptação (ajuste de parâmetros), técnicas sofisticadas e eficientes derivadas da teoria de controle automático (Kuo & Golnaraghi, 2003) podem ser prontamente empregadas.

Além disso, sistemas de controle incapazes de aprender e se adaptar dependem fortemente da existência de modelos matemáticos que descrevam com certa precisão o problema tratado. É evidente que nem sempre tais representações existem, em particular para os problemas mais complexos. Nestes casos, é fundamental que o sistema consiga se ajustar de maneira a aproximar o comportamento de entrada-saída esperado, fazendo mínimo uso possível de conhecimento inato, posto que este raramente está disponível. O processo de obtenção de tal aproximação é, basicamente, o de aprendizado (Bekey, 2005).

As técnicas de controle inteligente são assim classificadas pois apresentam capacidades de aprendizado e adaptação especialmente adequadas a problemas complexos e dinâmicos, cuja descrição matemática é muito difícil, senão impossível. Estas capacidades são oriundas, principalmente, da reprodução artificial de propriedades e mecanismos observados na natureza, caracterizando esta categoria de técnicas de controle como bio-inspirada (Passino, 2005).

As redes neurais artificiais, inspiradas em modelos do sistema nervoso, e a computação evolutiva, motivada pela teoria da evolução das espécies, são dois dos exemplos mais conhecidos de técnicas bio-inspiradas em controle inteligente (veja Von Zuben & Cazangi (2007) para uma discussão mais específica sobre controle via computação bio-inspirada e aplicações em robótica).

1.3. Fundamentos deste Trabalho

Como mencionado, este trabalho se situa na área de controle inteligente, compartilhando os mesmos conceitos que a fundamentam. Dois princípios, no entanto, têm papéis destacados neste estudo, e, portanto, receberão comentários adicionais: bio-inspiração e robótica evolutiva. Também serão abordadas a seguir, de forma breve, as principais técnicas para navegação autônoma utilizadas nesta pesquisa.

1.3.1. Bio-inspiração

Este trabalho possui uma forte inspiração em sistemas biológicos e mecanismos naturais. Praticamente todas as propostas desenvolvidas têm conexões com fenômenos observados na natureza, os quais apresentam propriedades interessantes do ponto de vista de problemas de engenharia e, em particular, de aplicações de sistemas de navegação autônomos.

As pessoas em geral nem sempre notam o grau de influência de eventos da natureza, particularmente dos fenômenos biológicos, como metáforas inspiradoras para a síntese de artefatos tecnológicos. Existem diversos exemplos que foram concebidos e aperfeiçoados em grande parte a partir da bio-inspiração: roupas em uso, ferramentas mecânicas, formas de exploração de fontes de energia, e até habitações construídas (de Castro & Von Zuben, 2005).

Na busca por formas artificiais de inteligência, Arkin (1998) considera razoável o emprego da bio-inspiração, julgando serem os seres biológicos a maior manifestação de comportamentos inteligentes existente e, portanto, o principal ponto de partida para estudos com tais objetivos. Tanto a abordagem que se concentra na criação de modelos fiéis aos sistemas biológicos originais quanto aquela que usa apenas abstrações têm produzido contribuições significativas à pesquisa em robôs móveis inteligentes.

A inspiração na biologia como metáfora para a concepção de algoritmos visando a solução de problemas, denominada computação bio-inspirada (de Castro & Von Zuben, 2005), se justifica pela possibilidade de reprodução artificial de características proveitosas intrínsecas a sistemas biológicos, como tendência à descentralização e existência de mecanismos de adaptação e auto-organização. O processo de síntese de sistemas de controle inteligente e de sistemas autônomos recorre a soluções computacionais via bio-inspiração. O que se busca com estratégias avançadas de controle fundamentadas na computação bio-inspirada é o projeto de controladores que apresentem habilidades

diferenciadas, tais como robustez, autonomia, processamento não-linear, capacidade de aprendizado e condicionamento.

1.3.2. Robótica Evolutiva

A robótica evolutiva é outro paradigma cujos princípios são empregados de modo determinante nas abordagens pesquisadas ao longo deste trabalho. É certo que, por ser baseada na teoria da evolução das espécies, a robótica evolutiva se caracteriza como bio-inspirada e, com isso, vem compor a linha de pensamento que fundamenta este estudo.

Modelos computacionais que simulam os processos evolutivos observados na natureza têm sido utilizados com sucesso nos mais diferentes tipos de problemas (Michalewicz, 1996). A grande eficiência destes métodos de evolução artificial viabiliza sua adoção na proposição estrutural de modelos, no desenvolvimento integral de sistemas de controle e na concepção e operação de robôs autônomos. Esta união de algoritmos evolutivos com robótica, conhecida como Robótica Evolutiva (Nolfi & Floreano, 2000), é bastante ambiciosa, pois propõe que tanto os componentes físicos (hardware) do robô, quando seu sistema de controle (software), sejam obtidos por meios evolutivos.

A robótica evolutiva sugere que se determine o design de um robô ao longo das gerações, combinando candidatos e favorecendo os indivíduos com melhores características, algo análogo à evidentemente bem-sucedida evolução biológica. Embora a evolução física do hardware ainda não seja algo totalmente consolidado, se restringindo à existência de uns poucos protótipos, a possibilidade de simular (e acelerar) etapas do processo torna este tipo de evolução bastante atrativo. Já em relação ao sistema de controle, todo o potencial dos algoritmos evolutivos, aliado à crescente capacidade de processamento hoje disponível, pode ser (e tem sido) utilizado de forma integral para síntese automática de controladores.

Um quesito muito forte em prol dos sistemas evolutivos se refere à autonomia. Ao se construir um sistema autônomo, seja robótico ou computacional, normalmente delega-se esta atividade a engenheiros. Porém, a própria evolução pode assumir o papel de projetista do sistema (Dawkins, 1997) de forma eficiente. Além disso, agentes que possuam características evolutivas podem, em princípio, ter seu nível de autonomia bastante incrementado. A evolução torna o agente mais independente de seu projetista e, portanto, propicia a ele potencial para ser mais autônomo que agentes cujos sistemas de controle não contemplam o evolucionismo (Pfeifer & Scheier, 1999).

1.3.3. Técnicas de Computação Bio-inspirada Utilizadas

As frentes de pesquisa apresentadas ao longo deste trabalho implementam algumas técnicas de inteligência computacional, as quais não apenas se alinham com os conceitos discutidos anteriormente como também sintetizam seus fundamentos teóricos em métodos computacionais aplicados. Conforme será expresso a seguir, tais técnicas, além de biologicamente inspiradas, são compatíveis com os princípios da robótica evolutiva, motivos pelos quais elas foram adotadas.

1.3.3.1. Algoritmos Evolutivos

Os algoritmos evolutivos se inspiram nos processos presentes na teoria da evolução natural das espécies, proposta por Darwin (1859), e na genética. Uma das vertentes, os algoritmos genéticos, foram introduzidos por Holland (1975) com o objetivo de formalizar matematicamente e explicar rigorosamente processos de adaptação em sistemas naturais e desenvolver sistemas artificiais (simulados em computador) que retivessem os mecanismos originais encontrados em sistemas naturais. Tais algoritmos são implementados como uma simulação de computador em que uma população de representações abstratas de solução é manipulada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criadas aleatoriamente e é realizada através de gerações. A cada geração, a adaptação de cada solução na população é avaliada e alguns indivíduos são selecionados para a próxima geração, sendo re combinados ou mutados para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo. Quando uma solução suficientemente adequada é encontrada, encerra-se a evolução.

1.3.3.2. Sistemas Classificadores com Aprendizado

Os sistemas classificadores com aprendizado podem ser descritos como métodos e princípios para criação e atualização de regras de inferência, chamadas classificadores, que codificam eventuais ações a serem tomadas por um agente sob condições específicas do ambiente (Booker, Goldberg, & Holland, 1989). Um dos componentes fundamentais do sistema, responsável pela atualização do conteúdo das regras e conseqüente aperfeiçoamento das mesmas, são os algoritmos genéticos. A presença deles confere o caráter evolutivo aos sistemas classificadores. Por serem capazes de combinar, em um único sistema, evolução e aprendizagem de forma integrada, os sistemas classificadores podem ser considerados modelos apropriados para a síntese de sistemas complexos adaptativos (Holland J. , 1995)

1.3.3.3. Sistemas Imunológicos Artificiais

O sistema imunológico biológico dos animais vertebrados, especialmente os mamíferos, consiste de um sistema distribuído que opera por meio de mecanismos intrincados envolvendo um grande número de células e moléculas no organismo. O sistema imunológico tem o objetivo principal de proteger o corpo de qualquer invasor prejudicial e manter a integridade do organismo. Para tanto, ele apresenta habilidades cognitivas importantes, tais como aprendizagem, memória e adaptação (de Castro L. N., 2003). Inspirados em tais propriedades, modelos computacionais têm sido propostos visando reproduzi-las de modo que possam ser úteis na resolução de problemas computacionais desafiadores. Estes modelos são conhecidos como sistemas imunológicos artificiais (SIA) (de Castro & Timmis, 2002).

1.3.3.4. Inteligência de Enxame

Sistemas baseados em inteligência de enxame são geralmente compostos por uma população de agentes simples que interagem localmente uns com os outros e também com seu ambiente. Exemplos naturais que motivam esta teoria incluem as colônias de formigas, cardumes de peixes, bandos de pássaros e outros. Os agentes seguem regras simples e, embora não haja uma estrutura de controle centralizada que dite o comportamento de cada elemento, as interações locais levam à emergência de comportamentos globais complexos (Bonabeau, Dorigo, & Theraulaz, 1999). A aplicação de princípios da inteligência de enxame à robótica é conhecida como robótica de enxame, a qual é uma das vertentes da robótica coletiva.

1.4. Contribuições

De uma forma geral, pode-se dizer que esta pesquisa tem como maior contribuição o estudo compreensivo dos aspectos pertinentes ao processo de produção de um sistema de navegação autônomo, uma vez que contempla desde a discussão teórica dos conceitos envolvidos, passando pelas etapas de projeto e implementação prática, até a fase de testes, primeiramente em simulação computacional e, finalmente, com a utilização dos controladores em ambientes e robôs reais. Além de discutir os fatores envolvidos em cada etapa, atenção especial é dedicada à busca por correlacioná-los ao longo do processo como um todo, de modo a apontar as principais implicações de decisões iniciais nos resultados finais obtidos. A seguir, são apresentadas as principais contribuições deste trabalho:

- Revisão de trabalhos correlatos e levantamento do estado da arte para cada aplicação;

- Identificação e apreciação dos principais aspectos técnicos envolvidos no projeto e desenvolvimento de sistemas de controle para robôs móveis autônomos, o que inclui itens como arquitetura do controlador, características do robô e propriedades do ambiente;
- Análise e reprodução artificial de atributos cognitivos essenciais em sistemas inteligentes, a saber: aprendizagem, adaptabilidade, memória, planejamento e comunicação;
- Descrição dos problemas de robótica móvel tratados, ressaltando sua importância prática, os desafios pertinentes, e qual seu papel nas decisões de projeto;
- Discussão da adequação, vantagens e desvantagens de cada técnica de controle inteligente (enumeradas na Seção 1.3.3) em relação aos problemas tratados e qual seu posicionamento em face aos diferentes aspectos de projeto;
- Proposição e desenvolvimento de cinco sistemas de navegação autônomos baseados em técnicas de computação bio-inspirada, as quais são detalhadas ao longo do texto;
- Aplicação dos controladores em problemas significativos da área de robótica móvel, apresentando o potencial e as limitações em cada caso. Os problemas, em particular, são: captura de alvos, desvio de obstáculos, otimização de rotas, fuga de mínimos locais, planejamento de trajetórias e missões;
- Realização de mais de 40 diferentes experimentos com o objetivo de testar extensivamente os cinco controladores implementados, tanto em simulação computacional quanto em experimentos reais, explorando as diversas características dos sistemas, principalmente em termos de robustez, flexibilidade e eficiência;
- Clarificação das fases e cenários mais apropriados do desenvolvimento para se usar simulação, experimentação real, e como combiná-los de forma produtiva;
- Mostrar como as decisões de projeto são colocadas em prática em cada um dos casos pesquisados e qual as implicações delas nos resultados.

1.5. Cronologia e Organização do Trabalho

A cronologia do doutorado apresentada aqui tem como objetivo prover uma visão geral da trajetória da pesquisa, contextualizando cada frente de atuação em relação às diferentes etapas transpostas ao longo do período do doutoramento. Desta forma, busca-se delinear os porquês de cada caminho escolhido, cada projeto desenvolvido e, principalmente, as sinergias entre todos eles. Espera-se

que, assim, as sucessivas transições entre as frentes de trabalho que ocorrerão ao longo do documento aconteçam de maneira gradual e compreensível.

Este doutorado se iniciou em 2004, como uma continuação do mestrado recém-concluído, motivado pela mesma temática de controle autônomo de robôs móveis por meio de técnicas de inteligência computacional. O objetivo, a partir de então, foi o de se aprofundar e contribuir nas principais questões pertinentes à área, explorando-as em aplicações a problemas variados, relevantes para robótica autônoma. Tais aspectos e especificações são discutidos detalhadamente no Capítulo 2.

Como resultado das investigações deste doutorado, cinco frentes de pesquisa com certas características distintas, mas inerentemente correlacionadas, foram desenvolvidas neste trabalho. Sob uma perspectiva geral, embora todas as linhas tenham o mesmo objetivo comum citado, as frentes de pesquisa podem ser informalmente divididas em duas fases complementares com viés diferente: científico e aplicado.

A primeira fase do doutorado, compreendida de 2004 a meados de 2006, apresenta uma tendência mais científica no sentido de se concentrar em estudos e contribuições de cunho mais teórico. Embora existam aplicações práticas e implementações reais, estas representam casos de estudo definidos com o propósito específico de validar as propostas de controladores autônomos.

Durante esta etapa mais científica, em que o autor esteve trabalhando em tempo integral no Laboratório de Bioinformática e Computação Bio-inspirada (LBiC) – Unicamp, três frentes de pesquisa foram investigadas:

- Sistema classificador com aprendizado, descrito no Capítulo 3;
- Robótica coletiva inspirada em estigmergia feita por formigas, discutido no Capítulo 4;
- Sistema imuno-classificador com aprendizado, apresentado no Capítulo 5.

A partir de meados de 2006, como parte do doutorado, o autor começou um estágio de pesquisa (doutorado sanduíche) em Washington, D.C., EUA, no Departamento de Informática Médica do maior hospital da cidade, o Washington Hospital Center. Neste estágio, posteriormente transferido para o Medical Media Lab (M3L) pertencente à Microsoft Corp., houve uma inclinação natural a problemas mais práticos e de aplicação imediata, dado o grande potencial da navegação autônoma em diversas atividades do hospital.

Em vista das circunstâncias, o trabalho de doutorado tomou um enfoque mais aplicado, em que uma visão de design rápido e eficiente de soluções, geralmente associado às engenharias, conduziu a pesquisa. Os aspectos teóricos continuam sendo explorados nesta etapa, havendo, porém, uma maior ênfase em aplicações específicas, as quais somente foram possíveis graças à base sólida de conhecimento construída na etapa anterior, com a qual esta se mantém alinhada.

Durante esta fase mais aplicada, duas frentes de pesquisa foram desenvolvidas:

- Sistema evolutivo para rastreamento de alvos móveis, apreciado no Capítulo 6;
- Plataforma de inspeção e coleta de dados baseada em abordagens evolutivas, proposta no Capítulo 7.

O autor considera extremamente importante a complementação que as duas fases proporcionaram, sendo cruciais para o amadurecimento do trabalho. Pretende-se apresentar todas as frentes de pesquisa de forma inter-relacionada e contínua, já que elas seguem os mesmos princípios que motivam este trabalho. Ao final do documento, no Capítulo 8, esta e as demais conclusões são delineadas.

Uma nota complementar deve ser feita para informar aos leitores de que, nesta tese, são citados diversos vídeos associados a experimentos e resultados apresentados ao longo dos capítulos. Tais vídeos estão disponíveis em formato eletrônico em um CD anexado ao documento impresso e, caso necessário, podem ser obtidos junto ao autor da tese.

Capítulo 2: Aspectos e Especificações de Sistemas de Navegação Autônomos

2.1. Aspectos de Sistemas de Navegação Autônomos

Tendo motivado e posicionado o trabalho num âmbito mais geral da ciência, é chegada a hora de explorar os aspectos específicos envolvidos no projeto e implementação de sistemas para navegação autônoma de robôs. A maior parte dos conceitos discutidos a seguir está relacionada ao projeto do sistema de controle autônomo e seus componentes, uma vez que este é o foco do trabalho. Além disso, outras características importantes são abordadas, as quais, embora não sejam intrínsecas ao controlador, afetam diretamente sua construção e operação.

As considerações feitas nas próximas seções buscam manter um tom mais teórico, uma vez que seu lado aplicado será amplamente investigado em cada uma das frentes de trabalho (Capítulos 3 a 7). Vale ressaltar também que os conceitos envolvidos na engenharia de robôs são, no fundo, multidisciplinares, abrangendo ciências biológicas, da computação e engenharia elétrica, dentre outras disciplinas.

2.2. Comportamento e Arquitetura

Ao se projetar e desenvolver um sistema de navegação autônomo é necessário determinar suas especificações em diversos âmbitos. Parte destas decisões derivam diretamente de fatos conhecidos, como o problema a ser tratado e o(s) robô(s) utilizado(s). Por outro lado, há decisões que não são tão óbvias e requerem maior atenção. A arquitetura do sistema é uma delas.

De acordo com o problema, ou seja, o ambiente e as tarefas enfrentados, é possível definir que tipos de comportamentos serão exigidos do robô. A partir do comportamento, determina-se a arquitetura do sistema, isto é, a organização e composição interna do sistema manifestam suas ações externas como comportamentos do robô. Pelo fato de existir uma forte ligação entre comportamento e arquitetura do sistema, eles serão discutidos em conjunto.

Há três tipos de comportamentos, ou arquiteturas, investigados ao longo deste trabalho: reativo, deliberativo e híbrido. Tanto o reativo quanto o deliberativo, em suas versões puras, possuem características bastante particulares e distintas. O tipo híbrido é, logicamente, a combinação dos dois

tipos primários, a qual pode se dar em variados graus. O quadro da Figura 2.1 sintetiza as propriedades gerais de cada abordagem, sendo que os detalhes serão discutidos a seguir.

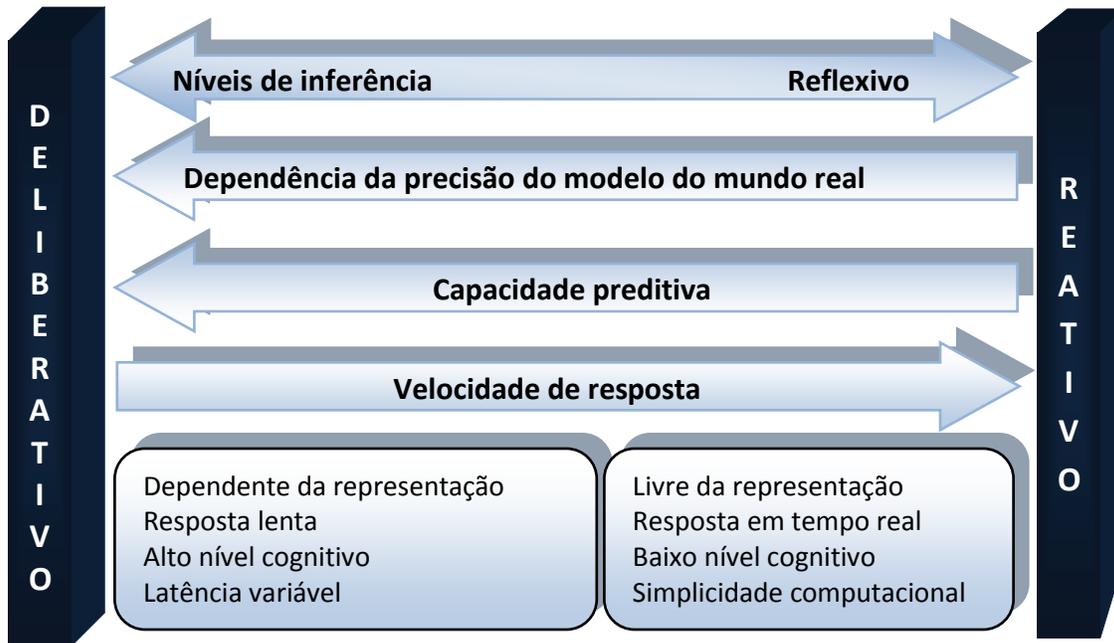


Figura 2.1: Características e distinções entre sistemas com comportamento reativo e deliberativo.

É importante ressaltar que a preferência por um ou outro tipo é algo altamente polêmico e debatido na literatura, sendo que cada qual possui advogados bastante fervorosos. Não é a intenção deste trabalho ser parcial, mas sim contribuir positivamente para este debate, explorando o que cada tipo tem de melhor e como eles se adequam, ou não, aos problemas abordados.

2.2.1. Puramente Reativo

Em uma simples definição, proposta por Arkin (1998), controle reativo é uma técnica que prima por uma ligação estrita entre percepção e ação, visando produzir respostas rápidas em ambientes dinâmicos e não estruturados. Sua principal característica é a operação direta sobre o ambiente, evitando-se o uso de abstrações, representações de mundo internas e conhecimento *a priori*.

Sistemas puramente reativos reagem diretamente ao mundo exterior assim como ele é visto (sensoriado). Esta reatividade é desejável em ambientes dinâmicos, desconhecidos e com alto grau de incerteza. Nestes cenários, decisões tomadas com atraso são ineficientes, dada a constante modificação

do ambiente. Além disso, não é possível garantir que suposições ou abstrações provenientes de conhecimentos inatos sejam sempre válidas.

A inteligência exibida por sistemas reativos não deve ser, em geral, incorporada *a priori* pelo projetista, mas sim uma propriedade que emerge a partir das interações do robô com o ambiente. Costuma-se afirmar que a estratégia de controle não é determinada pelo robô e nem pelo ambiente, mas sim pelo resultado da interação deles.

Tecnicamente, comportamentos reativos consistem de pares sensório-motores, em que um certo estímulo sensorial dispara de modo direto uma certa ação motora. Ou seja, não há interferência externa na tomada de decisões; a leitura sensorial imediata é toda a informação de que o sistema de controle precisa para tomar ações localizadas.

Quando se fala em ações de controle determinadas por um sistema reativo, geralmente se refere ao controle chamado de baixo nível. O controle de baixo nível se dá diretamente nos atuadores do robô, por exemplo, ajustando direção e velocidade das rodas de um robô móvel a cada movimento. Por ser um procedimento repetitivo, cujo intervalo entre a percepção e a ação tem que ser curto (muitas vezes imediato), a utilização de um controlador reativo parece ser a opção mais apropriada (Bekey, 2005).

Este caráter altamente responsivo de sistemas reativos, em termos de percepção e ação, tem um custo importante: baixo nível de cognição. Isto é, a reatividade não leva em conta aspectos cognitivos como planejamento e raciocínio, por exemplo. Na verdade, é justamente da ausência de cognição de alto nível (i.e. deliberação) que vem muito do poder de sistemas reativos (Brooks & Robust, 1986). No entanto, Arkin (1998) constata que, ao ignorar a cognição, robôs reativos ficam muitas vezes restritos a reproduzir o comportamento apenas de seres vivos simples.

Um exemplo ilustrativo de comportamento reativo em sistemas biológicos pode ser observado quando moscas se deparam com uma barreira de vidro, colidindo contra ela repetidas vezes, sem conseguir transpô-la. Dado que os estímulos são sempre os mesmos (o vidro é transparente) a mosca reage sempre da mesma forma, tentando atravessar o obstáculo. Devido a seu baixo nível cognitivo, a mosca não consegue se “lembrar” das tentativas anteriores mal fadadas e assim repete a mesma ação errônea continuamente.

2.2.2. Puramente Deliberativo

Um robô cujo sistema seja puramente deliberativo requer praticamente o conhecimento total sobre o mundo externo em que atua para que possa ser capaz de prever o resultado de suas ações. Em geral, tal conhecimento envolve presunções rigorosas de que o modelo de mundo é consistente, confiável e preciso. Caso estas hipóteses não se confirmem, ou caso o ambiente de navegação (mundo) seja modificado, o sistema pode falhar seriamente (Arkin, 1998).

Em ambientes controlados, onde eventos ocorrem de forma previsível e organizada, é possível utilizar controladores puramente deliberativos. Entretanto, é evidente que em ambientes dinâmicos, em que a estrutura do mesmo está em constante mudança, confiar cegamente em um sistema de controle puramente deliberativo é extremamente arriscado. Uma vez que a maioria dos problemas envolvendo ambientes e robôs reais são parcial ou totalmente dinâmicos, o uso exclusivo da deliberação torna-se inviável nestes casos. Além da questão da dinâmica do mundo real, há também a forte dependência de conhecimento inato (inserido no sistema pelo próprio projetista), que, com o passar do tempo, pode não ser mais válido.

Pelas razões mencionadas, a Inteligência Artificial (IA) tradicional, que se baseava em sistemas puramente deliberativos e simbólicos, não foi capaz de responder a demandas cada vez mais exigentes dos sistemas de processamento de informação a partir de meados dos anos 80.

Embora o termo deliberativo carregue uma certa controvérsia proveniente das técnicas da IA tradicional, não se pode ignorar o seu significado cognitivo e sua importância em sistemas biológicos. A deliberação está diretamente vinculada a propriedades cognitivas de alto nível, representando a habilidade de se pensar ou refletir antes de agir.

Em robótica, o aspecto deliberativo pode ser visto como o uso de algo a mais, além dos estímulos sensoriais para tomada de decisão. Este algo a mais é informação ou conhecimento representando por um modelo de mundo contendo mapas, planos ou estratégias. O robô utiliza tal conhecimento suplementar, geralmente armazenado em algum tipo de memória, para planejar ações antes de executá-las, prever resultados, estabelecer estratégias de longo prazo e memorizar outros conhecimentos relativos a eventos passados. A percepção do ambiente não é usada para escolha de ações diretamente, mas sim para ajuste do modelo de mundo, o qual serve de base para tomada de decisões, em lugar da própria realidade.

A maior complexidade cognitiva associada a sistemas deliberativos é responsável por fazê-los serem menos responsivos e mais lentos para tomada de decisão quando comparados a sistemas puramente reativos. Bekey (2005) cita um exemplo de sistema de controle de mísseis balísticos cuja versão deliberativa levava alguns minutos para determinar a ação, enquanto a versão reativa o fazia mais rápido em várias ordens de magnitude, em torno de alguma centenas de vezes por segundo. Obviamente, neste caso, a complexidade das ações tomadas também era distinta (a primeira de longo termo e a segunda de curto termo). Mesmo assim, esta observação serve para mostrar a que situações cada caso se adequa.

Continuando a linha de pensamento de Bekey (2005), é certo que controladores deliberativos trariam dificuldades caso fossem usados em controle de baixo nível. Por outro lado, o uso de deliberação é muito bem-vindo em controle de alto nível, o qual é responsável, por exemplo, por produzir ações de longo prazo, como determinação da estratégia de navegação e planejamento de trajetórias. Estas tarefas não exigem respostas necessariamente rápidas, ao passo que requerem o uso de capacidades cognitivas mais sofisticadas.

É importante enfatizar que, neste trabalho, o uso dos termos deliberação ou deliberativo não quer invocar o paradigma simbólico da IA clássica, mas quer sim se referir à cognição e suas capacidades, como planejamento e memória.

2.2.3. Híbrido

Como discutido, sistemas puramente deliberativos funcionariam de forma eficaz em ambientes previsíveis e imutáveis. No entanto, no mundo real, onde há outros agentes biológicos interagindo, as condições favorecendo o uso de controladores puramente deliberativos virtualmente não existem (Arkin, 1998). Sistemas puramente reativos, embora possam ser robustos e bem-sucedidos em certos ambientes complexos e dinâmicos, também têm suas desvantagens. Entre elas, a incapacidade de situar o robô em relação ao modelo de mundo e a impossibilidade de execução de tarefas relativamente complexas, dado o seu baixo grau cognitivo.

Tendo em vista as limitações de cada paradigma e, principalmente, suas virtudes, parece bastante razoável propor abordagens híbridas, em que características reativas complementam as deliberativas. A propósito, muitos pesquisadores [Lyons & Hendriks (1995), Mataric (1992), Dudek & Jenkin (2000) e

Ranganathan & Koenig (2003)] defendem esta proposta como essencial para um melhor proveito do potencial de cada uma das estratégias.

Na natureza, também existem inspirações para sistemas reativo-deliberativos. Schneider & Shiffrin (1977) sugerem que o comportamento biológico, de forma geral, se dá de dois modos: automático e controlado. Modelos de tais modos foram propostos por Norman & Shallice (1986) no contexto de comportamento humano, sendo que o modo automático pode ser associado à reatividade já que envolve a execução de ações involuntárias (e.g., instintos, reflexos), enquanto o modo controlado tem a ver com a deliberação por agir com intencionalidade (e.g., planejamento). Eysenck (2000) corrobora as analogias ao afirmar que os processos automáticos funcionam de maneira rápida e em paralelo, mas sofrem de uma ausência de flexibilidade; os processos controlados são flexíveis e versáteis, mas funcionam de forma serial e relativamente lenta.

Bekey (2005) também apóia a idéia híbrida pois, segundo ele, métodos deliberativos tradicionais são proibitivamente dependentes de conhecimento apurado da dinâmica e cinemática do robô, assim como de mapas extremamente detalhados do ambiente em que operam. Bekey (2005) complementa que a opção mais apropriada, portanto, é o emprego de estratégias reativas que incorporam capacidades deliberativas, como planejamento e auto-localização.

A organização típica de um sistema híbrido reativo-deliberativo consiste de três camadas, conforme mostrado na Figura 2.2 (Bekey, 2005). A mais alta é responsável pelos componentes cognitivos (deliberação), ao passo que a mais baixa contém o módulo reativo que, geralmente, interage diretamente com sensores e atuadores. A camada intermediária representa a interface entre deliberação e reação, gerenciando tal relação. Adicionalmente, quando há algum tipo de contato com um usuário externo, a interação se dá na camada deliberativa.

Segundo Arkin (1998), a camada deliberativa pode assumir alguns papéis em relação à reativa:

- Seleção: determinam-se os parâmetros que são utilizados pelo módulo reativo, reconfigurando-os sempre que necessário;
- Orientação: é feita a sugestão de quais planos ou ações devem ser executados, podendo o componente reativo acatá-los ou não;
- Adaptação: a estratégia reativa é diretamente modificada por meio de deliberação;
- Adiamento: ações deliberativas somente são executadas em último caso.

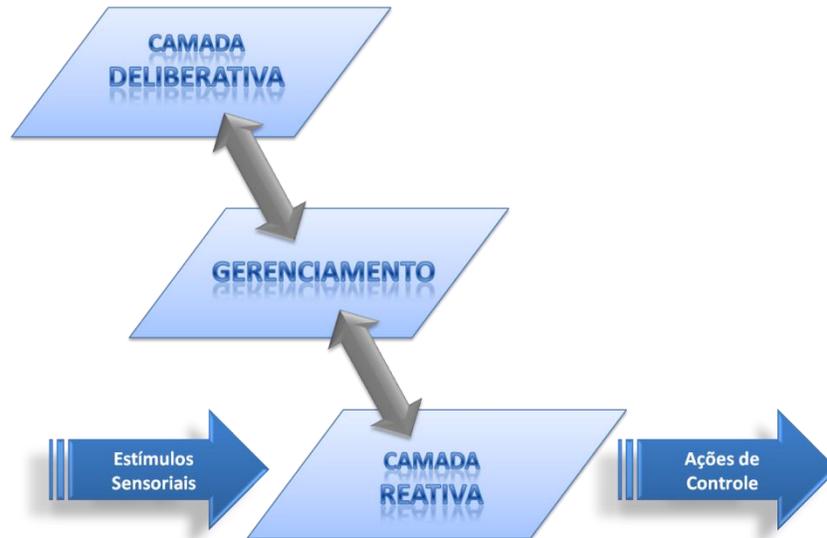


Figura 2.2: Organização típica de um sistema híbrido reativo-deliberativo.

Como apontado anteriormente, funções deliberativas requerem a existência explícita de representações de conhecimento, as quais podem afetar o grau de reatividade do controlador, todavia capacitando-o a realizar tarefas antes infactíveis. É possível incorporar conhecimento ao sistema híbrido de variadas formas:

- Memória de curto prazo: informações efêmeras de contexto local adquiridas, armazenadas e utilizadas por comportamentos específicos;
- Memória de longo prazo: conhecimento constantemente coletado do ambiente, empregado para a construção de um modelo de mundo genérico que, embora seja persistente, é plástico no sentido de ser continuamente atualizado.
- Conhecimento inato: informações como mapas ou estratégias de operação inseridas *a priori* (antes da execução) no sistema, as quais não são, em geral, obtidas pelo próprio robô, mas sim por agentes externos. Aqui é possível a simbiose com a computação simbólica.

Cuidado precisa ser tomado ao integrar propriedades deliberativas a um sistema reativo (e vice-versa) para evitar conflitos que causem sua deterioração. Se, por exemplo, um processo de planejamento tentar interferir em todos os detalhes do módulo reativo em um ambiente dinâmico ou buscar projetar demasiados passos futuros, o processo pode se complicar tanto que inviabiliza a obtenção de uma solução. Logo, é importante enfatizar que não basta apenas combinar de qualquer forma reatividade e deliberação, mas se faz essencial realizar a hibridização com zelo e adequação. Arkin

(1998) inclusive menciona que um sistema híbrido, se bem arquitetado, pode congrega o melhor dos dois mundos; contudo, caso feito incorretamente, pode render o pior de ambos os mundos.

Ainda que os argumentos em favor da abordagem híbrida se mostrem convincentes, não significa que ela é sempre a melhor opção. Note que há diversos cenários em que um controlador reativo simples é suficiente, como também existem situações (mais raras, é verdade) em que sistemas puramente deliberativos podem atuar. Além disso, a hibridização vem a um custo (aumento de complexidade, carga de processamento, etc.) que precisa ser ponderado ao se projetar a arquitetura do sistema robótico.

Um dos principais objetivos deste trabalho está justamente relacionado à investigação prática dos aspectos que vêm sendo tratados até aqui. Os múltiplos sistemas de navegação autônomos descritos ao longo dos próximos capítulos implementam variações de sistemas reativos, deliberativos e híbridos, explorando seus detalhes desde seu desenvolvimento até serem testados. Com esta análise, espera-se contribuir com a compreensão de quando e como utilizar cada vertente apropriadamente.

2.3. Ambiente e Robô

Os protagonistas deste trabalho são robôs, os quais são constituídos por duas entidades fundamentais: corpo e mente. A mente é o objeto de estudo principal deste trabalho, sendo representada pelas diversas abordagens exploradas ao longo do texto. A mente do robô é, de fato, um sistema de tomada de decisão que executa a função de mapeamento das possíveis entradas (estímulos) a saídas (ações de controle), guiando o robô durante sua operação.

O corpo do robô, por sua vez, é a entidade física ou aparente, a qual interage diretamente com o ambiente externo. O corpo pode ser de variados tipos e tamanhos, ou, alternativamente, pode ser apenas virtual, ou seja, um agente de software. As propriedades de cada tipo de robô são discutidas a seguir.

2.3.1. Virtual

Um robô virtual não possui uma estrutura física palpável, mas sim software que emula suas características corporais baseado em algum modelo pré-estabelecido. Estes robôs são utilizados em simulações computacionais em que o ambiente também é virtual. Esta vertente da engenharia robótica é importante por dois motivos básicos: proficiência e tempo. Nem todos os engenheiros possuem aptidão técnica e disponibilidade de recursos para construir e manipular o hardware de um robô real

(e.g., circuitos eletrônicos, dispositivos mecânicos, sensores e atuadores). Além disso, a investigação de comportamentos complexos em robôs reais pode demandar testes exaustivos, dificultando ou até inviabilizando a implementação. Para se ter idéia do tempo despendido com experimentos com robôs reais em relação às simulações, Nolfi & Floreano (2000) citam um experimento específico cuja realização completa com robôs reais levou cerca de 66 horas. Caso fosse feito em simulação, o mesmo experimento seria reduzido a apenas 1 hora. Note que essa relação de tempo gasto em cada caso não necessariamente se aplica a outros cenários em geral.

Graças aos recursos computacionais disponíveis atualmente, é possível realizar simulações bastante completas, por meio de modelagem matemática, capazes de reproduzir virtualmente condições detalhadas de operação em ambientes naturais. Ziemke (2003) afirma que as simulações são ferramentas poderosas para a execução e análise de testes devotados ao estudo da interação de um agente com o ambiente, pois, em muitos casos, elas tornam possível experimentos mais extensos, com uma sistemática mais complexa. Por meio de simulação, se gasta menos tempo e podem-se testar casos que raramente seriam viáveis devido à ausência de recursos ou tecnologia. Logo, ao invés de focar o trabalho em poucos experimentos com robôs reais, muitas questões podem ser melhor investigadas por meio de um número bem maior de testes, com um elenco amplo de morfologias, arquiteturas de controle e/ou ambientes.

O dilema virtual versus real tem gerado muita discussão entre pesquisadores, expondo as vantagens e desvantagens da simulação computacional quando comparada à implementação em robôs reais. Apesar dos benefícios e facilidades do uso da simulação, o consenso é que ela não pode substituir plenamente a experimentação em robôs reais, já que as características de operação no mundo real são extremamente difíceis de serem modeladas computacionalmente com exatidão e completude.

Segundo Pfeifer & Scheier (1999), à primeira vista parece melhor usar simulações devido à rapidez, custo reduzido e flexibilidade. Entretanto, sendo o objetivo final a operação em robôs reais, é premente que o sistema desenvolvido seja submetido a uma fase de teste embarcado. Portanto, dizem os autores, a conclusão é que ambas as etapas devem ser realizadas de forma complementar, sempre que possível.

2.3.2. Real

Robôs reais existem de todas as formas imagináveis, com uma diversidade enorme de características, como modo de locomoção, material de composição, arquitetura de hardware e outras,

as quais fogem do escopo deste trabalho. Um aspecto pertinente que poderá ser observado ao longo deste estudo está relacionado às dimensões do robô. Nas pesquisas atuais, as seguintes categorias são as mais comuns: nano, mini, normal e humanóide.

Nano-robôs são máquinas minúsculas, invisíveis a olho nu, cujo tamanho varia de 0,1 a 10 μm . Tais robôs ainda são mais um conceito do que realidade, já que apenas versões primitivas construídas com material molecular foram testadas. A principal aplicação em potencial da nano-robótica é na área de medicina para tarefas realizadas dentro do corpo humano, como detecção e destruição de células cancerosas (Drexler, 1987). Estes robôs ainda requerem um grande avanço tecnológico para que possam se tornar realmente viáveis.

Mini-robôs são de tamanho pequeno, não existindo uma definição padrão para suas dimensões. Basicamente, pode-se dizer que mini-robôs são robôs em miniatura, pois operam em ambientes restritos, bem menores que as construções usadas pelas pessoas. Tais robôs são muito versáteis e úteis para experimentos científicos, dado que não há necessidade de se reservar grandes espaços, como salas ou corredores, para que eles possam operar. Uma simples mesa é suficiente, o que facilita bastante a confecção de cenários de teste. É claro que não é possível explorar adequadamente comportamentos complexos de alto nível (e.g., interação com humanos) com este tipo de robô. Entretanto, boa parte das tarefas básicas de navegação autônoma pode ser devidamente investigada em ambientes miniaturizados.

Robôs em escala real, por sua vez, são os mais apropriados para operar em ambientes comumente utilizados por seres humanos. Devido a seu tamanho relativamente proporcional ao de um ser humano, este tipo de robô reproduz as condições enfrentadas pelas pessoas. Alguns pesquisadores preferem mini-robôs a robôs de tamanho normal, devido ao custo mais barato e à limitação de espaço nos laboratórios.

A categoria de robôs conhecidos como humanóides é, talvez, a que possui maior associação à imagem popular de um robô. Estes robôs são construídos de forma a reproduzir as características físicas dos seres humanos, em geral possuindo braços, pernas e cabeça. Algumas das principais aplicações de robôs humanóides estão relacionadas ao estudo das emoções, linguagem corporal e locomoção.

2.3.3. Especificações Adotadas

Neste trabalho, faz-se uso extensivo tanto de simulações em ambientes virtuais quanto de experimentação real, sendo que há uma grande ênfase na integração dos dois modos. Ou seja, mesclam-se etapas de simulação computacional com testes embarcados em robôs reais, sendo a primeira dedicada à evolução e aprimoramento do sistema proposto, e a segunda empregada visando sua validação efetiva.

Como citado, agentes robóticos virtuais são um dos tipos empregados aqui. Em relação aos robôs reais, um mini-robô, Khepera II, e um robô normal, PatrolBot, são utilizados ao longo da pesquisa. Ambos os robôs têm sido expressivamente aproveitados por grupos de pesquisa e empresas em todo o mundo. Os detalhes de cada um serão apresentados a seguir.

2.3.3.1. Khepera II

O mini-robô Khepera II, que é fabricado pela empresa K-Team S.A., foi originalmente desenvolvido por um grupo de pesquisadores do *Microprocessor and Interface Laboratory* (LAMI) da Suíça, especialmente voltado à utilização em pesquisas científicas. Sua concepção obedeceu a alguns critérios como miniaturização, modularidade e interface simples. Por ser pequeno, o ambiente de navegação pode ser construído ocupando espaços reduzidos. O robô é modular e expansível, admitindo a adição de novos componentes, como outros sensores (câmeras) e outros atuadores (garras). Os modos de comunicação com o computador são de dois tipos: por sinais de rádio ou por conexão serial via cabo.

O Khepera II, mostrado na Figura 2.3, possui 80 mm de diâmetro, 30 mm de altura e tem peso de aproximadamente 80 gramas. Ele se movimenta por meio de duas rodas laterais independentes (plataforma diferencial), cada qual controlada por um motor de passo, podendo assim girar com precisão nos sentidos horário e anti-horário. Este mecanismo confere ao robô a capacidade de executar manobras curtas tanto para sua direita como para sua esquerda. A velocidade máxima atingida pelo robô é de 1 m/s.



Figura 2.3 : Mini-robô Khepera II, cujo tamanho é de 80 mm x 30 mm.

A aquisição de informações do ambiente é feita por 8 sensores infravermelhos que capturam estímulos. Uma leitura completa de todos os sensores é feita a cada 20 ms, dado que cada sensor leva 2,5 ms para fazer uma medição. Esses estímulos, capturados pelos sensores, correspondem a uma leitura instantânea da situação do robô no ambiente, a partir do ponto de vista do robô (proximal). Os sensores possuem duas funcionalidades: detecção da intensidade luminosa e medição da distância a obstáculos. No primeiro caso, os sensores (às vezes denominados sensores de alvo) medem a quantidade de luz infravermelha no ambiente, que é equivalente à quantidade de luz visível. No segundo caso, os sensores (chamados de sensores de obstáculo) aferem a distância entre o robô e os obstáculos situados em frente aos respectivos sensores, por meio da emissão e reflexão de raios infravermelhos. As duas funcionalidades estão conjugadas em cada um dos 8 sensores.

Os 8 sensores são dispostos ao redor do robô, sendo 6 na frente e 2 atrás, conforme é mostrado na Figura 2.4. O alcance máximo dos sensores de distância a obstáculos é de 100 mm. Já os sensores de luminosidade são bem mais sensíveis e podem detectar gradientes luminosos a longas distâncias. A autonomia do robô é de mais ou menos 1 h, sendo necessário o mesmo tempo para recarga total de suas baterias.

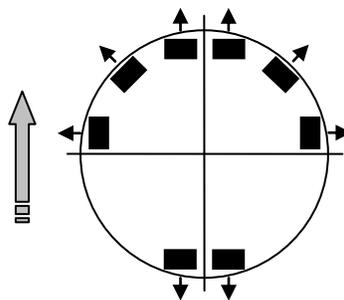


Figura 2.4: Distribuição dos sensores na estrutura do robô, cuja frente é apontada pela seta.

2.3.3.2. PatrolBot

O robô PatrolBot, visto na Figura 2.5 e no Vídeo 2.1, é fabricado pela empresa americana MobileRobots Inc. Trata-se de um dos mais sofisticados robôs disponíveis no mercado civil atualmente. Seu propósito é de uso geral e, embora seu nome sugira patrulhamento, ele tem sido utilizado para as mais diversas aplicações. As dimensões do robô são 59 cm de comprimento por 48 de largura e 1 metro de altura, pesando 46 Kg e podendo transportar um carga adicional de até 40 Kg.

O PatrolBot também é uma plataforma holonômica diferencial, com duas rodas controladas por motores independentes. A velocidade máxima que ele pode alcançar é de 1,7 m/s. Em termos de energia, sua autonomia é por volta de 4 h, requerendo o mesmo tempo para recarga. O robô é equipado com um conjunto grande de sensores de diversos tipos. Para detecção de distância a obstáculos, o robô possui 16 sensores de ultra-som (8 atrás e 8 na frente) e um sensor laser (LRF – *laser range finder*) capaz de realizar leituras de 180° em frente ao robô a uma frequência de até 75 Hz (alcance máximo de 80 m). Além disso, o robô contém 12 sensores de contato (bumpers), uma câmera móvel, alto-falantes e microfone.

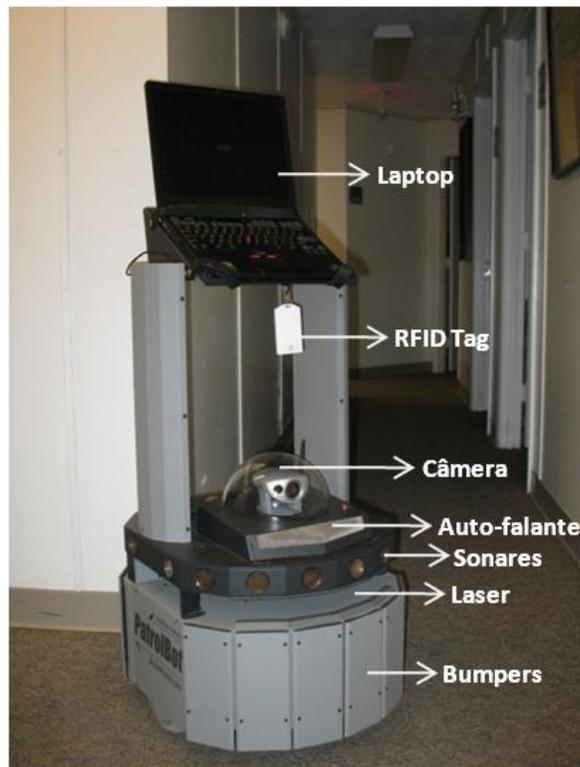


Figura 2.5: Foto do Robô PatrolBot, indicando seus principais sensores e dispositivos. Veja também o Vídeo 2.1.

O controle do robô é feito por meio de um computador *on-board* completo (P4 1600 GHz). Uma placa de rede sem-fio 802.11b/g permite que o robô se conecte à Internet e, assim, possa se comunicar e ser controlado remotamente em qualquer parte do planeta. Adicionalmente, o PatrolBot carrega um laptop independente (veja Figura 2.5) que serve para interação com pessoas próximas.

2.4. Tarefas e Aplicações

Ao se imaginar quais tarefas um robô móvel autônomo pode realizar, surgem inúmeras opções. O robô pode explorar planetas desconhecidos, patrulhar áreas de segurança, limpar ambientes, transportar materiais, fazer inspeção em tubulações e fiações, coletar objetos ou dados, guiar pessoas, etc.

Apesar da grande diversidade de aplicações citadas, todas elas possuem características em comum, independentes inclusive do modo de locomoção do robô (e.g., aquático, terrestre ou aéreo). Isto é, todas podem ser descritas com base em combinações ou repetições do seguinte procedimento: o robô navega de um local de origem até outro de destino, executando tarefas específicas, durante o trajeto e/ou ao seu final. As tarefas específicas podem ser, por exemplo, controlar uma câmera (no caso de patrulhamento), uma enceradeira (no caso de limpeza) ou um braço mecânico (no caso de coleta). Note que todas essas tarefas requerem, em geral, sistemas de controle independentes do sistema de navegação autônomo.

O ponto que se quer chegar aqui é que, da perspectiva do engenheiro ou projetista do robô móvel, há um conjunto finito de tarefas base, relacionadas ao processo de navegação do robô, que, ao serem implementadas, habilitam a utilização do mesmo para qualquer outra aplicação. Sendo assim, este trabalho focará no desenvolvimento de controladores para navegação autônoma, associados ou não a outras tarefas complementares.

Serão considerados dois objetivos básicos junto ao processo de navegação autônoma. Para que o robô possa navegar no ambiente, ele precisa desviar de eventuais obstáculos. Portanto, uma habilidade essencial é o desvio de obstáculos, sejam eles dinâmicos ou estáticos. Outra habilidade, geralmente denominada de captura de alvo ao longo do texto (em particular nos capítulos cujas aplicações são genéricas), consiste em executar uma trajetória que ligue um ponto de partida (origem) a um local de destino (alvo).

2.5. Aprendizagem

Aprendizagem é um termo cuja definição é um tanto quanto polêmica. Um psicólogo certamente tem um conceito diferente do de um engenheiro, por exemplo. Se a definição é polêmica, ao menos algo é consenso: aprendizagem é um dos principais componentes da inteligência [(Arkin, 1998), (Bekey, 2005), (Pfeifer & Scheier, 1999)]. Dado que este é um trabalho de engenharia, assume-se que aprendizagem é a capacidade de um agente ou sistema de aprimorar seu desempenho, ou obter soluções melhoradas, ao longo do seu processo de operação.

Em robótica autônoma, aprendizagem é um princípio fundamental (Bekey, 2005). Um robô com capacidade de aprendizado é potencialmente mais autônomo que outro incapaz de aprender, visto que aquele pode se adaptar a situações inéditas ou imprevisíveis, o que é extremamente comum em ambientes reais. Logo, pode-se dizer que existem três principais tipos de conhecimento que um robô deve ser capaz de aprender: informações desconhecidas, ambientes dinâmicos e conhecimentos que não sejam passíveis de serem pré-programados.

Segundo Arkin (1998), o processo de aprendizagem em sistemas inteligentes pode se dar das seguintes maneiras:

- Introdução ou aquisição de novos conhecimentos;
- Generalização a partir de conhecimentos específicos ou especificação de conceitos genéricos;
- Reuso de experiências passadas;
- Reorganização interna do sistema;
- Criação ou descoberta de soluções para tarefas ou problemas sendo tratados.

Em inteligência artificial, diversas abordagens para aprendizagem artificial têm sido propostas, entre elas: treinamento supervisionado e não-supervisionado em redes neurais artificiais (Haykin, 1998), aprendizado por reforço (Sutton & Barto, 1998) e computação evolutiva (Holland J. , 1975). A última delas, computação evolutiva, é de particular interesse neste trabalho, pois seus métodos são predominantes nos sistemas desenvolvidos.

É importante esclarecer as sinergias entre aprendizagem e evolução. Basicamente, ambas são formas de adaptação biológica que diferem em espaço e tempo. Evolução é um processo de reprodução

seletiva que é capaz de capturar modificações relativamente lentas que ocorrem no ambiente ao longo de muitas gerações de uma população, levando a ajustes no código genético (genótipo) dos indivíduos. Aprendizagem, ao contrário, permite ao organismo se adaptar ao ambiente durante seu período de existência, por meio apenas da alteração de características fenotípicas do indivíduo (Nolfi & Floreano, 2000).

Diferentemente da evolução natural das espécies, a qual deve respeitar os intervalos de tempo que caracterizam o ciclo de vida dos indivíduos biológicos (Darwin, 1859), a evolução realizada computacionalmente pode simular a sucessão de gerações em escalas de tempo arbitrárias, muitas vezes da ordem de segundos ou minutos (Bekey, 2005). A esta “facilidade” se atribui boa parte da eficiência e sucesso dos algoritmos evolutivos em problemas computacionais.

Novamente: evolução e aprendizagem ocorrem biologicamente em intervalos de tempo diferentes; enquanto a primeira pode levar milhares de anos, a segunda pode se dar em dias ou semanas. Não obstante, a partir do momento em que se pode realizar, de maneira relativamente rápida, evolução artificial em computadores, algo surpreendente torna-se possível: aprender por meio de evolução.

No contexto de sistemas inteligentes, surgem, portanto, duas abordagens que combinam evolução e aprendizagem de formas diferentes. Ambas as abordagens estão diretamente relacionadas à Robótica Evolutiva (Nolfi & Floreano, 2000), uma das filosofias que norteiam este trabalho. Veja, nas seções a seguir, uma breve discussão de cada uma delas. Para uma revisão mais completa das sinergias entre evolução e aprendizagem, o livro de Passino (2005) é uma boa referência.

2.5.1. Evolução com Aprendizagem

Evolução com aprendizagem é uma abordagem análoga ao que ocorre biologicamente. Ou seja, os indivíduos da população evoluem ao longo das gerações e, ao mesmo tempo, cada um deles pode também aprender e se adaptar durante sua existência, em sua respectiva geração. Assim, tem-se o processo evolutivo modificando os indivíduos no longo prazo, enquanto o processo de aprendizagem os altera no curto prazo, o que pode ser visto como uma combinação de evolução com busca local.

Passino (2005) defende a aprendizagem como uma forma de acelerar a evolução pela incorporação de conhecimentos pelo sistema na forma de instinto. A aprendizagem resulta em uma pressão seletiva adicional no sentido de incluir informações consistentemente úteis ao sistema. Deve-se ter em mente,

porém, que existem custos adicionais associados à aprendizagem e armazenamento de conhecimento, e, por isso, é importante usar tais mecanismos com parcimônia.

Ambas, evolução e aprendizagem, têm como objetivo final a adaptação ao ambiente, sendo que a segunda complementa a primeira no sentido de permitir que o indivíduo se adeque a mudanças no ambiente que aconteçam rápido demais para serem incorporadas pelo processo evolutivo. Além disso, os efeitos do aprendizado podem influir no curso evolutivo, possibilitando que indivíduos originalmente não tão bons (conseqüentemente com chances menores de reprodução) se tornem mais aptos ao longo de sua existência e, assim, aumentem sua probabilidade de reprodução (Nolfi & Floreano, 2000).

Embora estas duas frentes de adaptação confirmem propriedades vantajosas para organismos artificiais e naturais, a capacidade de aprendizagem possui aspectos negativos a se considerar. Da mesma forma que indivíduos podem se aperfeiçoar, dependendo de vários fatores, eles podem também aprender erroneamente, trazendo efeitos negativos que comprometem a competitividade individual ou do grupo. Outro aspecto negativo bastante intuitivo é a maior necessidade de recursos computacionais (no caso artificial) ou de energia (no caso natural) para que os mecanismos de aprendizado atuem (Mayley, 1997).

Note que, quando se fala em indivíduos no caso de computação evolutiva, estes podem representar virtualmente qualquer tipo de solução para um problema. Em particular, neste trabalho, a estratégia de evolução com aprendizagem é usada no design de sistemas de controle para robôs móveis, significando, portanto, que cada indivíduo da população codifica um controlador autônomo. Evolui-se, assim, o controlador que equipa o robô, e não o robô propriamente.

2.5.2. Aprendizagem por Evolução

Ao contrário da estratégia anterior, esta não é biologicamente plausível. Realizar aprendizagem por meio de evolução implica que o genótipo do indivíduo seja modificado em vida, algo geneticamente infactível. Afortunadamente, no mundo computacional, as limitações supracitadas não existem, o que torna possível usar o poder da evolução como mecanismo de aprendizagem artificial.

Embora biologicamente a evolução dificilmente seja considerada uma forma de aprendizado, Bekey (2005) afirma que processos evolutivos em computador são sim capazes de emular processos adaptativos que podem ser caracterizados como aprendizagem artificial. Isto fica evidente quando se

observa um algoritmo evolutivo em ação: a partir de um estado (população) inicial de baixa qualidade, operações evolutivas efetuadas ao longo de sucessivas gerações levam à melhora da qualidade e, conseqüentemente, a resultados bem-sucedidos. Logo, pode-se assumir que houve aprendizado.

A aprendizagem por evolução pode ser empregada de duas formas distintas: contínua ou discreta. A primeira está associada a um processo cumulativo de aperfeiçoamento do comportamento ou performance global de um robô ou sistema. Isto é, os ganhos evolutivos ocorrem incrementalmente, estando cada etapa do processo vinculada às anteriores. Em geral, esta estratégia é lenta e pode, ou não, continuar indefinidamente. Exemplo: o robô deve aprender a desviar de obstáculos enquanto se move em um ambiente desconhecido.

No segundo caso, diz-se que a aprendizagem por evolução é discreta pelo fato de resolver instâncias específicas de um problema que compõe a tarefa geral sendo executada. Ela pode ser dita pontual porque, cada vez que uma solução local é obtida, esta é utilizada e descartada, não influenciando nas próximas atuações do sistema. Ou seja, ela não é cumulativa e, portanto, é uma aprendizagem considerada de curto prazo e contida em um intervalo limitado de tempo. Exemplo: em planejamento de trajetórias, cada rota evoluída e executada é independente das demais e é aprendida pelo sistema apenas para aquela situação específica.

2.6. Individualidade e Sociedade

Outro aspecto importante em engenharia robótica envolve a quantidade de robôs sendo controlados em operações conjuntas (no mesmo ambiente). Embora a robótica individual, isto é, a pesquisa focada no sistema de controle de um único robô em isolamento, seja a mais comum na literatura, a robótica coletiva tem atraído um crescente interesse nas últimas décadas (Arai, Pagello, & Parker, 2002).

Existem diversas motivações para o desenvolvimento de sistemas multi-robôs, sendo as que seguem as principais (Edelen, 2003):

- Muitas tarefas são complexas demais para um único robô executar (e.g. exploração de território);
- Múltiplos robôs de estrutura simples podem ser uma solução mais barata e fácil de se obter do que um único robô sofisticado;

- Sistemas multi-robôs são, em geral, mais flexíveis e robustos que um robô operando isoladamente, já que há maior tolerância a falhas e distribuição de responsabilidades;
- Certas tarefas que eventualmente possam ser decompostas em processos paralelos podem ser realizadas com maior eficiência por múltiplos robôs;

É evidente que o controle de um grande número de robôs inflige dificuldades antes não existentes em se tratando de robôs individuais. Veja a seguir algumas delas (Arkin, 1998):

- O espaço disponível para operação é reduzido pela ocupação dos vários robôs;
- Os robôs tornam-se obstáculos uns para os outros;
- A dinâmica e a complexidade geral do processo de navegação coletiva tendem a aumentar bastante, dado que cada robô age de forma autônoma.

fenômenos coletivos e seres que vivem em sociedades auto-organizadas observadas na natureza são as mais freqüentes inspirações para trabalhos de robótica coletiva. Vários exemplos de processos auto-organizados naturais podem ser citados. Em termos de objetos inanimados, um exemplo são os grãos de areia que formam dunas. Em se tratando de seres vivos, mesmo em organismos microscópios se podem observar tais fenômenos: por meio de incontáveis interações de células, o sistema imunológico humano é capaz de se auto-organizar em resposta a estímulos internos e externos (de Castro & Timmis, 2002). Em uma escala maior, também é possível identificar auto-organização em muitas espécies de insetos e animais superiores como peixes e pássaros, os quais apresentam habilidades para estabelecer e manter estruturas complexas (Deneubourg J. L., Aron, Goss, & Pasteels, 1990).

Quando se trata de sistemas com múltiplos robôs, muitos outros aspectos de projeto entram em cena. Assim como nas sociedades de organismos biológicos, uma sociedade de robôs possui características importantes como organização, distribuição espacial e de responsabilidades, hierarquia, comunicação e outras.

Apesar de a maioria das frentes pesquisadas neste trabalho ser dedicada exclusivamente à robótica individual, uma em particular investiga sistemas multi-robôs inspirados por colônias de insetos que se comunicam indiretamente por substâncias químicas. O Capítulo 4, dedicado a este tema, apresenta os detalhes da frente de pesquisa, buscando explorar na prática todos os atributos e implicações da coletividade em contraposição à individualidade.

2.7. Otimização

Muitos sistemas biológicos realizam otimização de algum tipo de critério de desempenho. Nem sempre o critério é precisamente determinado ou conhecido, mas, em geral, sistemas biológicos operam com base na minimização do gasto de energia. Por exemplo, Otis et al. (1950) supõem que o sistema que controla a respiração ajusta a frequência de inspiração/expiração procurando minimizar o esforço dos músculos envolvidos no processo. De forma análoga, os sistemas de controle artificiais também são arquitetados visando a otimização de fatores específicos. Em geral, os critérios mais comuns são:

- Minimização de algum tipo de erro;
- Minimização do consumo de energia;
- Minimização do tempo de resposta a uma dada entrada;
- Maximização da velocidade para atingir um objetivo.

Avaliando robôs autônomos capazes de aprender, a otimização pode ser vista como um complemento à aprendizagem. Suponha uma aplicação em que o robô deve coletar objetos no ambiente. Primeiramente, o robô precisa aprender como encontrar e coletar os objetos. Em seguida, espera-se que o robô consiga maximizar a quantidade de objetos coletados e minimizar o tempo gasto nesta tarefa. Portanto, enquanto a aprendizagem é fundamental, a otimização é determinante para que robôs possam não apenas cumprir suas tarefas, mas também executá-las com eficiência.

Abordagens evolutivas são métodos inerentemente otimizadores, já que buscam sempre descobrir as soluções mais aptas (maior *fitness*) para o problema em questão (Bekey, 2005). Logo, os algoritmos evolutivos podem realizar aprendizagem e otimização ao mesmo tempo, o que é extremamente útil e bastante explorado neste trabalho.

São vários os critérios que se procura otimizar no escopo de navegação autônoma de robôs ao longo dos próximos capítulos. Alguns deles se preocupam com a otimização de comportamentos, como desvio de obstáculos e captura de alvos (Capítulos 3,4 e 5); outros tentam minimizar a complexidade do sistema, tornando-o mais parcimonioso (Capítulo 5); e há ainda aqueles que se concentram na obtenção de soluções otimizadas, em particular: trajetórias mais curtas e menores circuitos entre dois locais arbitrários do ambiente de navegação (Capítulos 6 e 7).

2.8. Especificações das Frentes de Pesquisa

Sob a perspectiva dos aspectos considerados neste capítulo, a Tabela 2.1 sintetiza as principais características e especificações das frentes de pesquisa desenvolvidas neste trabalho. Além disso, cada capítulo apresentado ao longo do documento conterá uma seção exclusivamente dedicada à discussão detalhada das características de cada frente dentro do contexto do trabalho em geral e em relação aos conceitos comentados nas seções deste capítulo.

Tabela 2.1: Resumo das características de cada frente de pesquisa contida neste trabalho.

	SNA Original (Capítulo 3)	Robótica Coletiva Inspirada nas Formigas (Capítulo 4)	Abordagem Imuno- classificadora (Capítulo 5)	Sistema Evolutivo para Rastreamento (Capítulo 6)	Plataforma de Inspeção e Coleta de Dados (Capítulo 7)
Arquitetura e Comportamento	Reativo	Reativo	Híbrido	Híbrido	Híbrido
Ambiente	Virtual e real em miniatura	Virtual	Virtual e real em miniatura	Virtual e real (hospital)	Virtual e real (hospital)
Robô	Khepera II (mini)	Khepera II (virtual)	Khepera II (mini)	PatrolBot (normal)	PatrolBot (normal)
Modo dos Experimentos	Simulação e real	Simulação	Simulação e real	Simulação e real	Simulação e real
Aplicação	Navegação	Navegação e vida artificial	Navegação	Rastreamento	Inspeção e coleta de dados
Aprendizagem	Por meios evolutivos (contínua)	Por meios evolutivos (contínua)	Com evolução e por meios evolutivos (contínua)	Por meios evolutivos (discreta)	Por meios evolutivos (discreta)
Sociedade	Individual	Coletivo	Individual	Individual	Individual
Otimização	Comportamento	Comportamento e trajetórias	Comportamento e complexidade	Trajетórias	Missões e trajетórias
Técnicas Bio- Inspiradas	Computação Evolutiva + Sistemas Classificadores	Computação Evolutiva + Sist. Classificadores + <i>Swarm Robotics</i>	Computação Evolutiva + Sist. Classificadores + Sist. Imunológicos Artificiais	Computação Evolutiva	Computação Evolutiva

2.9. Descrição dos Vídeos

Vídeo 2.1: Uma rápida visualização do robô PatrolBot.

Capítulo 3: Sistema de Navegação Autônomo baseado em Sistema Classificador com Aprendizagem

3.1. Introdução

Neste capítulo, é apresentado um sistema de navegação autônomo (SNA) desenvolvido, em sua maioria, no trabalho de mestrado do autor, (Cazangi R. R., 2004). A inclusão deste capítulo se deve ao fato de que o sistema proposto é usado como base para dois outros sistemas que fazem parte deste trabalho, descritos nos Capítulos 4 e 5. Embora as principais informações deste sistema de navegação sejam descritas aqui, isto será feito de forma abreviada, já que a dissertação em (Cazangi R. R., 2004) é totalmente dedicada a ele, contendo o detalhamento completo.

A concepção e projeto do SNA se baseiam em sistemas classificadores e algoritmos genéticos, ambas as técnicas de natureza evolutiva. A idéia fundamental é promover a autonomia do sistema, dotando-o de capacidade de aprendizagem e adaptação mediante interação com o ambiente.

3.2. Contextualização do Capítulo

No contexto das outras frentes a serem apresentadas neste trabalho, os aspectos mais importantes deste sistema são os seguintes. O SNA se caracteriza por ser reativo, pois utiliza apenas estímulos instantâneos, capturados do ambiente pelos sensores do robô, para navegar. Não há uso de mapas (o ambiente é totalmente desconhecido) ou qualquer tipo de memória adicional. Além disso, o trabalho também é orientado pelo preceito de evitar a incorporação, em fase de projeto, de conhecimento abstrato. O conhecimento abstrato é definido aqui, sem formalismo, como o conhecimento resultante das capacidades sofisticadas da inteligência humana.

Por não fazer uso de um modelo explícito do mundo e por não contar com uma dinâmica interna sofisticada, o SNA pode ser classificado como de nível cognitivo simples. Ainda assim, o sistema incorpora um mecanismo de aprendizagem por meios evolutivos que é, de fato, a sua principal característica. Esta capacidade de aprendizagem atua incremental e continuamente nos comportamentos do robô, permitindo que estratégias reativas de navegação sejam desenvolvidas sem a participação de um projetista externo. O processo de aprendizagem está associado a experiências cumulativas sofridas pelo robô durante sua operação no ambiente, as quais disparam algoritmos evolutivos responsáveis pela descoberta e refinamento de conhecimentos úteis ao sistema.

Mesmo sendo reativo e de baixo nível cognitivo, o mecanismo de aprendizagem presente no SNA o habilita a executar tarefas relativamente complexas: capturar alvos e desviar de obstáculos. Dada a ausência de comportamentos inatos, o objetivo do sistema de navegação é, portanto, aprender a executar ambas as tarefas e, ao mesmo tempo, aprender também a coordenar adequadamente os comportamentos envolvidos, já que eles são potencialmente conflitantes.

Supondo que eventuais mudanças de cenário se apresentem durante a navegação e impliquem em um comprometimento de desempenho no atendimento dos objetivos, o sistema de navegação deve disparar automaticamente uma seqüência de eventos responsáveis pela adaptação à nova condição do ambiente, visando recompor o nível de atendimento dos objetivos da navegação. Em outras palavras, deve-se promover a síntese automática de novas estratégias de navegação que sejam compatíveis com o novo panorama.

Diferentemente de sistemas classificadores tradicionais, esta versão possui duas partes antecedentes distintas e duas partes conseqüentes também distintas, sendo que há uma mistura de codificações, desde binária até real. Outra inovação é que o algoritmo genético usado para descoberta e evolução de regras não é disparado segundo uma freqüência fixa de iterações, mas sim variável, dependendo única e exclusivamente da ocorrência de certos eventos associados à navegação do robô. São eles: colisão em obstáculo, captura de alvo e monotonia (falta de objetividade).

De forma geral, analisando a questão da aprendizagem do sistema (realizada por meios evolutivos), pode-se dizer que no SC de Holland (1986) ela ocorre somente na fase de treinamento. Isto é, o sistema é treinado inicialmente em uma etapa específica, na qual toda sua aprendizagem ocorre, para depois atuar efetivamente. Diferentemente, no sistema de navegação deste trabalho, não há uma fase determinada somente para treinamento. Na verdade, a aprendizagem está habilitada o tempo todo, enquanto o robô navega pelo ambiente, e depende somente do disparo dos eventos já mencionados para acontecer. Note que, embora o aprendizado possa ser feito com maior ou menor intensidade (taxa de procriação), seus mecanismos nunca são desativados completamente. O sistema sofre aprendizagem e adaptação efetivamente enquanto atua sem que haja interrupção da navegação.

É importante ressaltar que a emergência e coordenação de comportamentos é resultado de processos de aprendizagem e adaptação, que, desta forma, garantem a autonomia e flexibilidade do sistema. A aprendizagem se dá de forma contínua, ao longo da atuação on-line do sistema, e nunca pára, buscando sempre otimizar o comportamento do robô.

As tarefas que este sistema se propõe a realizar são de caráter genérico, já que não estão associadas a nenhuma aplicação prática específica (embora possam ser facilmente instanciadas). Logo, o intuito desta frente de pesquisa é explorar as propriedades do sistema autônomo sob uma visão geral, não se preocupando em propor soluções para qualquer tipo de aplicação em particular.

Por ser voltado aos princípios envolvidos no desenvolvimento do próprio SNA, apenas experimentos individuais (robôs isolados) são considerados aqui. Além de testes com robôs reais, usa-se também o expediente de simulações computacionais de forma extensiva. O mini-robô Khepera II é o modelo adotado neste capítulo.

3.3. Descrição do Sistema de Navegação Autônomo Evolutivo

O sistema de navegação é responsável pelo controle total do robô, tomando todo tipo de decisão, tanto relativa a estratégias de navegação como à aprendizagem, parte essencial do sistema. Ele é reativo, isto é, ao contrário de sistemas deliberativos, ele não utiliza estados internos ou histórico de decisões já tomadas para decidir a próxima ação de controle. Assim, o sistema se baseia apenas em informações (estímulos) capturados no instante da tomada de decisão. O pseudocódigo da Figura 3.1 descreve algoritmicamente o processo de funcionamento do sistema.

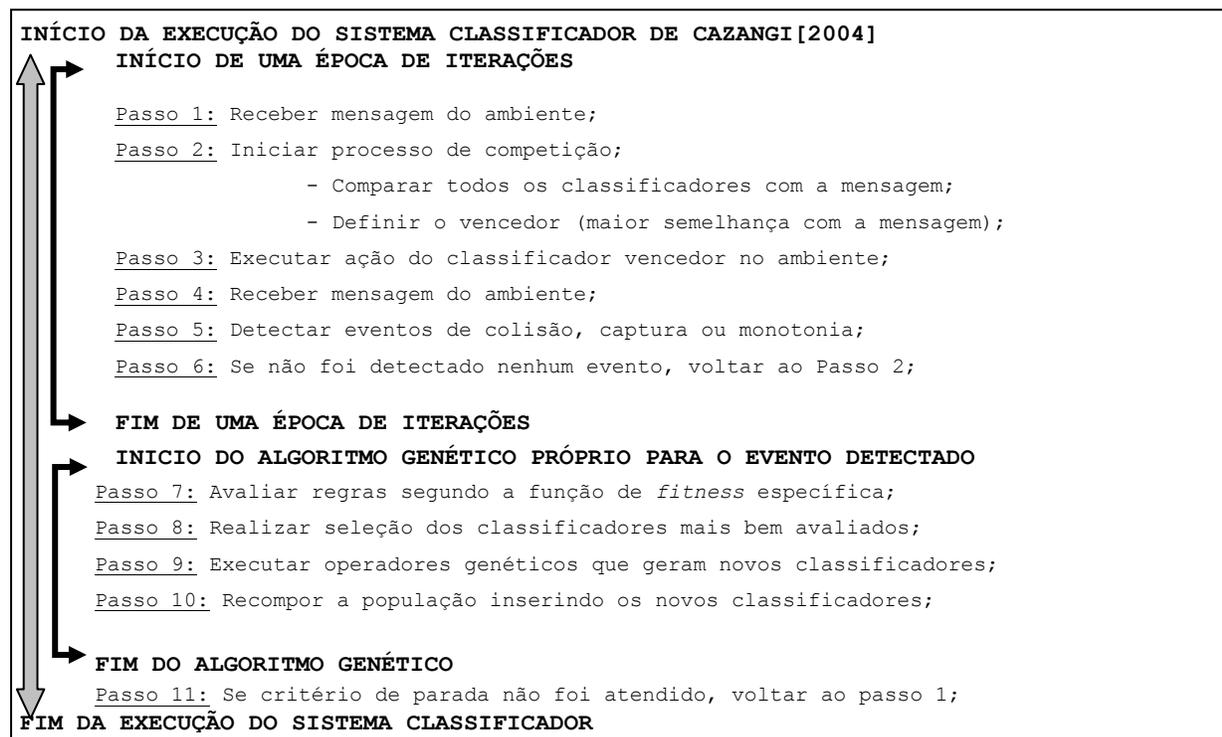


Figura 3.1: Algoritmo simplificado do sistema classificador implementado neste trabalho.

Inicialmente, não há conhecimento significativo presente no sistema, ou seja, o robô não possui estratégias de navegação definidas e desconhece a topologia do ambiente. A assimilação de conhecimento necessário para a execução da tarefa de navegação, a emergência de comportamentos e o conseqüente desenvolvimento de estratégias de navegação se dão pela ocorrência de processos de aprendizagem por meios evolutivos.

O SNA proposto, que se baseia nos sistemas classificadores, pode ser dividido em duas partes: sistema de controle e processo de evolução. O primeiro, responsável pela tomada de decisões, interage com o ambiente por meio de sensores e atuadores (mecanismo de ajuste de direção e velocidade), e possui dois componentes principais: população de regras e módulo de competição. Já a evolução, por sua vez, está associada à aprendizagem do sistema e contém dois componentes que manipulam a população de regras, a saber: módulo de avaliação e módulo de reprodução. A Figura 3.2 mostra o diagrama do sistema de controle, enquanto o processo de evolução é representado pela Figura 3.3.

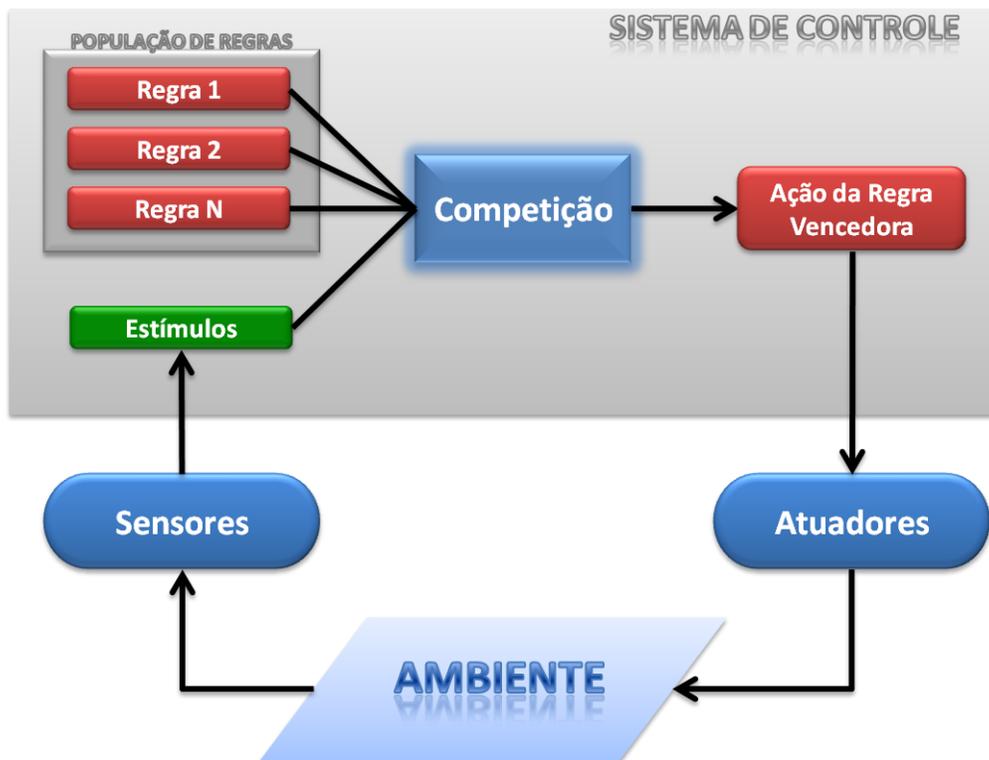


Figura 3.2: Diagrama que representa a estrutura e funcionamento do sistema de controle.

A população de regras representa o conhecimento do sistema e evolui durante a navegação do robô, mais precisamente entre dois ciclos sensório-motores, a serem definidos em seguida. Como mostrado na Figura 3.2, o módulo de competição recebe os estímulos, capturados pelos sensores de alvo e obstáculo, e os compara com as regras, escolhendo a mais apropriada. Essa regra vencedora, por sua vez, tem sua ação enviada para o atuador, que finalmente ajusta a velocidade e direção do robô. Este processo se repete a cada movimento do robô, compondo um ciclo sensório-motor. O ciclo somente é interrompido quando se faz necessária uma evolução da população (aprendizagem), caracterizando o fim de uma época de iterações. Assim, como indicado na Figura 3.3, três eventos particulares disparam a aprendizagem: colisão, captura e monotonia (evento virtual que ocorre quando o robô apresenta comportamento monótono). Nestes casos, as regras são avaliadas e alteradas, por meio de operadores evolutivos específicos para cada evento, produzindo uma nova população. Concluída a atualização da população, uma nova época é iniciada e o sistema volta a atuar no ambiente.

Os detalhes e pormenores de cada módulo que compõe o sistema de navegação autônomo, assim como a forma como cada um deles interage com os demais, serão apresentados nas seções a seguir. Na ordem, as regras e sua estrutura são explicadas primeiro; em seguida aparecem os componentes do sistema de controle e, finalmente, é discutido o processo evolutivo.

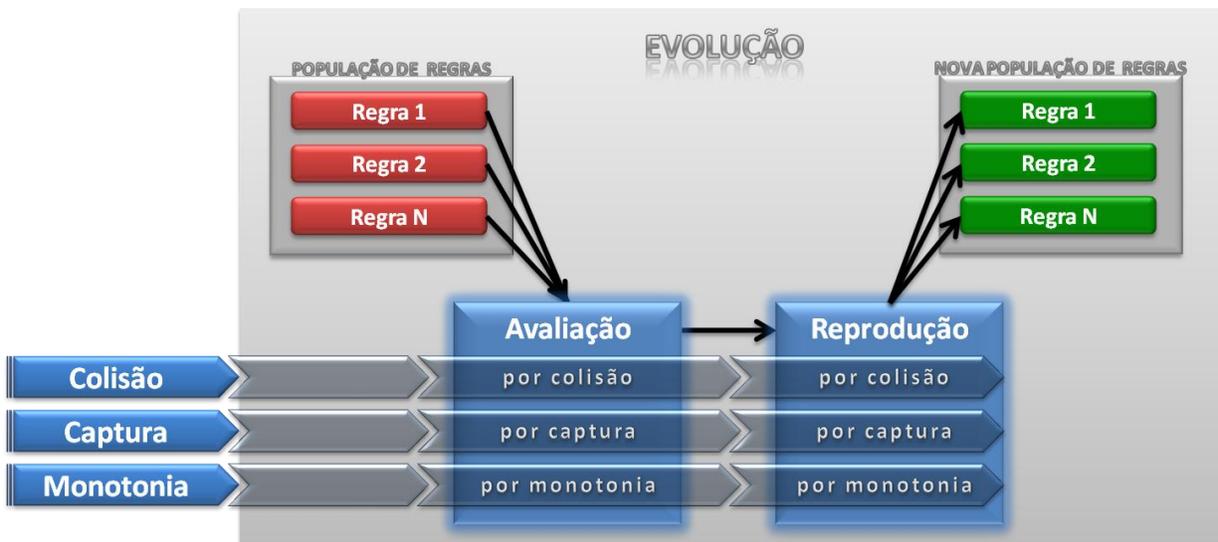


Figura 3.3: Organização e fluxo do processo evolutivo responsável pela aprendizagem do SNA.

3.3.1. População de Regras

Cada indivíduo da população é representado por uma regra de inferência do tipo se <condição> – então <ação>. A Figura 3.4 apresenta um exemplo de uma regra completa. O tamanho da população é fixo neste caso, contando com 200 regras. Este número foi definido empiricamente por intermédio de experimentos envolvendo vários tamanhos de população. A estrutura de cada regra, ou seja, seu cromossomo, é composta por quatro vetores: RO, RA, RD e RV. As características e significado de cada vetor estão na Tabela 3.1.

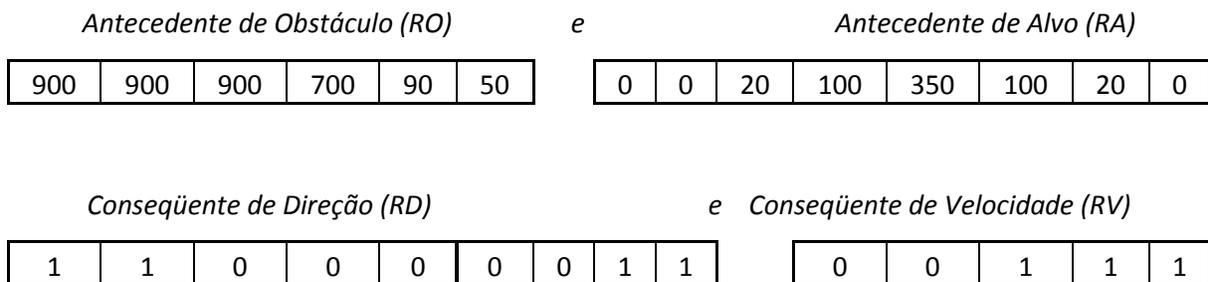


Figura 3.4: Exemplo do cromossomo de uma regra cujos conseqüentes determinam giro de -3° e aumento de velocidade.

Os vetores RO e RA compreendem a parte antecedente da regra, que representa uma possível situação do ambiente, ou seja, uma determinada leitura dos sensores do robô. Por isso, os vetores consistem, respectivamente, de 6 e 8 elementos, que são os números de sensores de proximidade e sensores de luminosidade, respectivamente.

Tabela 3.1: Características dos vetores que compõem um indivíduo (regra) da população.

Nome	Parte da Regra	Representação	Elementos	
			Quantidade	Tipo
RO	Antecedente	Distância a Obstáculos	6	Inteiro
RA	Antecedente	Luminosidade do Alvo	8	Inteiro
RD	Conseqüente	Ajuste de Direção	9	Binário
RV	Conseqüente	Ajuste de Velocidade	5	Binário

A parte conseqüente da regra é composta pelos vetores RD e RV. São 9 os elementos de RD, sendo que seus 5 bits mais significativos representam o sinal, e os 4 bits restantes indicam o valor absoluto do ajuste de direção proposto pela regra. No caso dos bits de sinal, se a maioria deles for 0, então o sinal é

negativo. Caso contrário, o sinal é positivo. O sinal, na prática, indica o lado do ajuste: negativo é direita (sentido horário) e positivo é esquerda (sentido anti-horário). Note que os 4 bits de valor absoluto cobrem 15° , que é a margem de manobra máxima do robô. Cabe uma explicação quanto à existência de 5 bits para determinação de um única variável binária (o sinal). Com mais bits evita-se que alterações esporádicas e pontuais causem mudanças drásticas na ação indicada pela regra. Isto é, a determinação do sinal não fica concentrada em um único bit, mas sim distribuída por vários bits.

Em relação ao vetor RV, a mesma idéia é empregada. Se a maioria de seus 5 bits for 0, a velocidade do robô deve ser reduzida, caso contrário, deve ser aumentada. Aqui não há variações diferentes de velocidade, simplesmente a regra sinaliza para que a velocidade seja incrementada ou decrementada, sendo sempre de um valor constante.

A construção da população inicial, no princípio da navegação, é realizada com a seleção de valores aleatórios para todos os campos das regras, em suas partes antecedente e conseqüente. É importante notar que, embora aleatórios, os valores de cada campo são escolhidos de acordo com seu tipo, em uma distribuição de probabilidade uniforme.

3.3.2. Módulo de Competição

A cada movimento do robô, os sensores de obstáculos e de alvo capturam estímulos do ambiente na forma de vetores, EO e EA, respectivamente, os quais são transferidos para o módulo de competição. Nesta etapa, todas as regras competem para definir a vencedora, ou seja, aquela cuja parte antecedente é mais semelhante aos estímulos. A Figura 3.5 mostra um exemplo que ilustra o processo de competição das regras. São mostradas as partes antecedentes de três regras candidatas e seus respectivos cromossomos, além dos estímulos sensoriais hipotéticos. A competição é feita usando a Equação 3.1, que calcula a similaridade das partes antecedentes das regras com os estímulos, determinando que a primeira regra é a vencedora. A seguir, sua parte conseqüente, que codifica a ação vencedora, é convertida para os valores a serem aplicados aos atuadores do robô que, neste exemplo, resultariam em uma rotação de 3° à direita e aumento da velocidade.

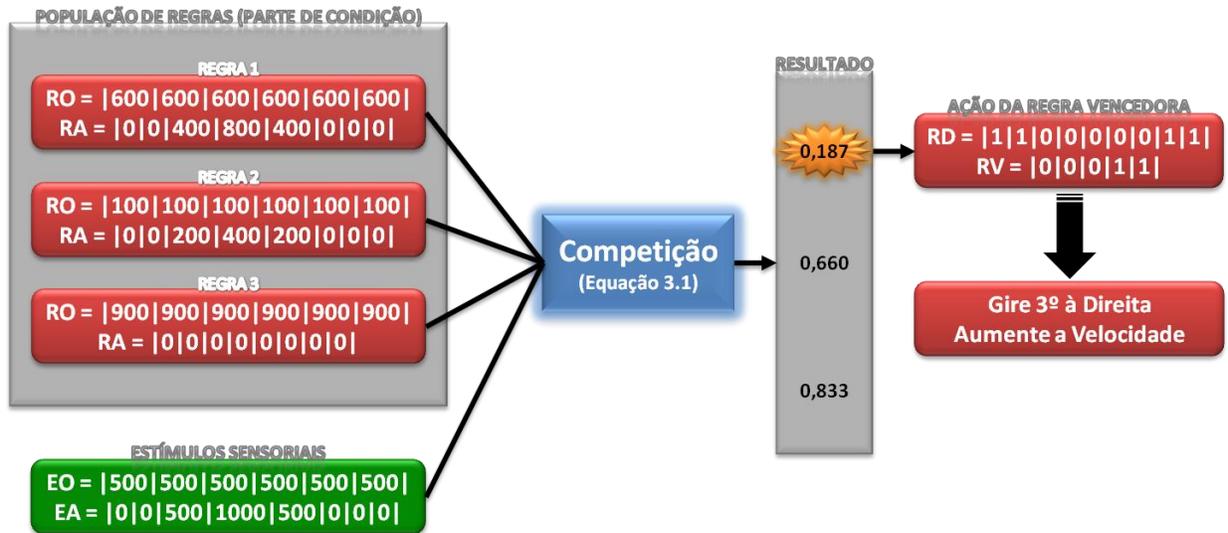


Figura 3.5: Exemplo que ilustra como as regras competem para que seja determinada a ação usada pelo sistema.

Como dito, a regra vencedora é determinada pelo cálculo da semelhança de cada regra r em relação aos estímulos, sendo a semelhança $S(r)$ obtida através da Equação 3.1. A regra de maior semelhança será aquela que obtiver o menor valor de $S(r)$.

$$S(r) = \frac{\|RO(r) - EO\|}{MaxO} + \frac{\|RA(r) - EA\|}{MaxA} \quad \text{Equação 3.1}$$

onde $MaxO$ e $MaxA$ representam valores adequados para normalização do primeiro e segundo termos de $S(r)$, respectivamente. $\|x - y\|$ é a norma euclidiana da diferença entre os vetores x e y .

A Equação 3.1, responsável por medir a similaridade entre regras e estímulos, é composta por dois termos normalizados. O primeiro termo indica a semelhança relativa padrão de distância a obstáculos, e o segundo termo é relativo à semelhança das partes ao padrão de distância ao alvo. A combinação dos dois termos (por meio de adição) em um único valor foi estabelecida como uma forma de impor que o sistema decidisse a regra vencedora baseado em um único critério. Portanto, a regra com menor $S(r)$ é escolhida para atuar.

3.3.3. Evolução

O processo de evolução da população de regras é executado pelos módulos de avaliação e reprodução. A evolução é disparada ao final de uma época de iterações, ou seja, sempre que um dos seguintes eventos é detectado: colisão em obstáculo (os sensores de proximidade medem um valor mínimo pré-estabelecido), captura de alvo (os sensores de luminosidade detectam valores acima de um

limiar pré-determinado) e monotonia. Um evento de monotonia é gerado se o robô não capturar um alvo por um período determinado, ou se o somatório dos ajustes de direção nas iterações anteriores superar um certo limiar.

Cabe aqui uma ressalva importante a respeito do evento de colisão em obstáculos. Quando uma colisão é detectada pelo sistema, na realidade não significa que o robô efetivamente chocou-se com o obstáculo. É feito um tratamento, durante a leitura dos sensores, que faz com que os eventos de colisão sejam detectados tão logo o robô se encontre a uma distância mínima predefinida do obstáculo (mas não encostado). Desta forma, não há danos físicos à estrutura do robô. Embora este evento passe uma imagem negativa, a ocorrência de colisões é essencial à evolução do sistema de navegação. Parece controverso, já que o objetivo da navegação é não colidir. Entretanto, se o robô não sofresse colisões, não aprenderia nunca a desviar de obstáculos, visto que este comportamento não é inato ao sistema e se desenvolve justamente ao longo da navegação. Em resumo, as colisões em obstáculos são negativas unicamente em relação aos objetivos do processo de navegação, mas são fundamentais à evolução do sistema de navegação autônomo. Logo, com a evolução do sistema a taxa de colisões deve cair rapidamente e, em certas aplicações, até zerar.

A ocorrência de colisões e monotônias é reflexo de deficiências no sistema de navegação, provenientes de um estado primitivo do sistema, ou então por regras malformadas, inconsistentes ou incompletas presentes na população. Portanto, estes dois eventos funcionam como um alerta para a necessidade de busca de novas regras, demandando a execução de processos evolutivos que venham a corrigir e aprimorar as regras que definem os comportamentos. Uma captura de alvo é um evento oposto aos outros dois, no sentido de que as regras que atuaram levando à captura foram competentes. Por isso, a captura de alvo é um evento que gera um alerta para a necessidade de reforçar e difundir junto à população características dessas regras competentes, incentivando a realização de processos evolutivos que implementam este reforço e difusão de propriedades desejadas.

Cada um dos três eventos dispara um processo evolutivo específico. Além disso, devido às partes antecedentes das regras serem de diferente natureza, elas são avaliadas e reproduzidas separada e independentemente. Ao final, as partes são reunidas gerando assim a nova população.

3.3.3.1. Colisão

Se uma colisão ocorre, um processo evolutivo é realizado com o objetivo de aprimorar o comportamento de desvio de obstáculos apresentado pelo robô. Neste sentido, os módulos de avaliação e reprodução são configurados visando este intuito.

3.3.3.1.1. Módulo de Avaliação da Evolução Disparada por um Evento de Colisão

A avaliação da parte antecedente das regras se baseia em sua similaridade com a situação instantânea na colisão. Ou seja, o valor da avaliação é igual a $S(r)$, obtido pela Equação 3.1, em que EO e EA são exatamente os mesmos vetores de estímulos capturados no momento da colisão. Quanto menor for $S(r)$, mais semelhante aos estímulos é o antecedente da regra r . Logo, os antecedentes são ordenados segundo $S(r)$, e aqueles que possuem menor valor terão, proporcionalmente, maiores probabilidades de serem selecionados para reprodução.

A parte conseqüente de ajuste de direção das regras é avaliada em termos de sua semelhança com o reflexo instintivo do robô, cujo valor depende do tipo de colisão que ocorreu. Por exemplo, se o robô colidiu com sua lateral esquerda, o reflexo instintivo assume o valor do ajuste de direção máximo para o lado direito (-15°) e vice-versa. Assim, os conseqüentes que possuem valores mais próximos ao reflexo instintivo terão maiores chances de se reproduzirem. A avaliação é dada por $T(r)$, calculado pela Equação 3.2.

$$T(r) = \begin{cases} |[RD(r)]_d - 15|, & \text{se colisão à esquerda;} \\ |[RD(r)]_d + 15|, & \text{caso contrário.} \end{cases} \quad \text{Equação 3.2}$$

onde $[RD(r)]_d$ é o ajuste de direção definido pelo conseqüente da regra r (número decimal).

A última avaliação é da parte conseqüente de ajuste de velocidade das regras. Ela é inversamente proporcional à distância *Hamming*, dada por $dH(r)$, entre o vetor $RV(r)$ e um vetor padrão que representa redução de velocidade (todos os elementos são 1). Considerando que em situações de iminente colisão a velocidade deve ser reduzida, os conseqüentes que sugerem redução de velocidade serão bem avaliados. A Equação 3.3 mostra como é feito o cálculo:

$$dH(r) = \sum_{i=1}^5 |RV(r)_i - PRV_i| \quad \text{Equação 3.3}$$

onde $RV(r)$ é o ajuste de velocidade indicado pelo conseqüente da regra r , $PRV=[1,1,1,1,1]$ e i representa o i -ésimo elemento do vetor.

Tabela 3.2: Informações do módulo de avaliação em caso de disparo por colisão.

Parte da Regra	Critério de Avaliação (normalizado de 0 a 1)	Equação
Antecedente	Semelhança com Situação de Colisão	Equação 3.1
Conseqüente de Direção	Semelhança com Reflexo Instintivo	Equação 3.2
Conseqüente de Velocidade	Semelhança com Vetor de Redução	Equação 3.3

3.3.3.1.2. Módulo de Reprodução da Evolução Disparada por um Evento de Colisão

Em seguida ao módulo de avaliação, executa-se o módulo de reprodução, onde entram em ação os operadores genéticos que produzirão os novos indivíduos. Deve-se enfatizar que o módulo de avaliação estabelece classificações diferentes para cada parte das regras e, portanto, os operadores irão agir em fases independentes: no grupo dos antecedentes e, depois, no grupo dos conseqüentes.

Um fator importante a ser mencionado antes do início da descrição do processo é o conceito de taxa de procriação (TxP). A taxa de procriação define quantos indivíduos filhos serão produzidos pelo processo evolutivo. Neste trabalho, o número de filhos gerados não é fixo, variando segundo TxP , a qual é descrita detalhadamente mais adiante, ainda neste capítulo, mas já se adianta que ela pode variar de 10% a 1%. Portanto, o número de filhos produzidos em cada geração (Nf) é calculado assim:

$$Nf = TamPop * TxP \quad \text{Equação 3.4}$$

onde $TamPop$ indica o tamanho da população e TxP é a taxa de procriação, que pode variar de 10% a 1%.

A primeira operação a ser executada é a seleção, feita pelo método da roleta (Michalewicz, 1996). São selecionados indivíduos em quantidade igual a Nf , levando-se em conta a pontuação feita pelo módulo de avaliação. Estes indivíduos selecionados serão os genitores (pais) que, tomados 2 a 2, formarão os filhos. A cada dois pais selecionados, dois filhos são gerados pelos operadores evolutivos.

Para se gerar as novas partes antecedentes de obstáculo (vetores RO), são aplicados o cruzamento de um ponto e a mutação tradicional. O cruzamento toma um par de pais selecionados e os combina gerando dois filhos. A mutação é ativada de acordo com uma certa taxa de probabilidade e altera

aleatoriamente o conteúdo de um gene do cromossomo. Os novos conseqüentes de direção e velocidade (vetores RD e RV) são gerados pelos mesmos operadores.

O antecedente de alvo (vetor RA) é gerado por um operador específico que usa o ângulo associado ao elemento de maior intensidade luminosa de cada pai e calcula o ângulo médio perturbado aleatoriamente (segundo uma certa variância) para que os novos vetores RA dos filhos sejam gerados.

Consumados os procedimentos do módulo de reprodução, tem-se um conjunto de novas partes antecedentes e um conjunto de novas partes conseqüentes isoladas. Para que os novos indivíduos estejam completos, escolhem-se aleatoriamente os vetores de cada parte que são então re-acoplados. Este processo se repete até que todas as N_f novas regras estejam formadas. Finalmente, os procedimentos evolutivos disparados pela colisão são concluídos, substituindo-se a fração de população antiga, cujos representantes são os pais, pelos novos indivíduos (os filhos). Então o robô volta a navegar normalmente pelo ambiente.

3.3.3.2. Captura

Ao contrário da colisão, o processo evolutivo disparado por uma captura de alvo visa reforçar ainda mais o comportamento de busca de alvos. Portanto, os módulos de avaliação e reprodução possuem configurações voltadas a este objetivo.

3.3.3.2.1. Módulo de Avaliação da Evolução Disparada por um Evento de Captura

A avaliação dos vetores antecedentes também é análoga àquela feita na colisão. Isto é, avaliam-se os antecedentes por similaridade com a situação instantânea de captura. Uma regra r é avaliada de acordo com seu $S(r)$ (Equação 3.1), cujos vetores EO e EA são definidos com os mesmos valores detectados pelos sensores no instante da captura do alvo. Conseqüentemente, os antecedentes mais semelhantes à condição do ambiente no momento da captura, cujo $S(r)$ é menor, terão maior probabilidade de serem selecionados para reprodução.

A avaliação do conseqüente de direção (vetor RD) depende do ângulo do sensor que detectou a captura (o mais próximo do alvo) e do próprio ajuste de direção da regra avaliada. A idéia é que os conseqüentes de direção cujos ajustes levariam a um maior alinhamento entre o sensor da captura e o alvo sejam mais valorizados. A avaliação de uma regra r , dada por $E(r)$, é calculada como segue:

$$E(r) = |[RD]_d - \alpha|$$

Equação 3.5

onde $[RD]_d$ é o ajuste de direção definido pelo conseqüente (número decimal) e α é o ângulo do sensor de captura que detectou o evento.

A terceira avaliação é da parte conseqüente de ajuste de velocidade das regras. O processo é idêntico ao caso de colisão, calculada inversamente proporcional à distância *Hamming* (Equação 3.3), dada por $dH(r)$, entre o vetor $RV(r)$ e um vetor padrão que representa redução de velocidade (todos os elementos são 1). Aqui também os conseqüentes que representam redução de velocidade são desejados, pois, geralmente, o ato de capturar um alvo pode estar associado à execução de tarefas adicionais envolvendo o alvo capturado. As configurações do módulo de avaliação são exibidas na Tabela 3.3.

Tabela 3.3: Informações do módulo de avaliação em caso de disparo por captura.

Parte da Regra	Critério de Avaliação (normalizado de 0 a 1)	Equação
Antecedente	Semelhança com Situação de Captura	Equação 3.2
Conseqüente de Direção	Semelhança com Ângulo do Sensor	Equação 3.5
Conseqüente de Velocidade	Semelhança com Vetor de Redução	Equação 3.4

3.3.3.2.2. Módulo de Reprodução da Evolução Disparada por um Evento de Captura

Os procedimentos do módulo de reprodução para captura são os mesmos que se realizam no caso de colisão, descritos na Seção 3.3.3.1. Após a execução deste módulo, o robô segue navegando pelo ambiente.

3.3.3.3. Monotonia

É importante ressaltar que inicialmente o sistema de navegação não possui qualquer estratégia, nem de prevenção de colisão, nem de captura de alvo. Portanto, no começo da navegação, e também em situações em que os comportamentos estão malformados, o robô vagueia pelo ambiente sem objetivo aparente, acontecendo capturas de alvo casualmente. Caso não ocorram eventos de colisão nem de captura por um certo período de tempo, um evento de monotonia é detectado, e o respectivo processo evolutivo é desencadeado. O objetivo deste processo é eliminar as regras que estão

contribuindo para que o robô apresente comportamento monótono e, indiretamente, estimular a busca de alvos por parte do robô. Além disso, o evento de monotonia tem certo respaldo nos sistemas biológicos. As formigas são um exemplo. Se uma formiga se isola do seu grupo, ela pode acabar perdida, vagueando a procura do formigueiro até sua morte por exaustão (Bonabeau, Dorigo, & Theraulaz, 1999).

3.3.3.3.1. Módulo de Avaliação da Evolução Disparada por um Evento de Monotonia

O processo de avaliação, no caso de monotonia, consiste em analisar um histórico recente das regras atuantes, até um certo número de iterações anteriores à corrente, verificando quais delas provocaram, por culpa de sua ação, o aumento da distância angular do robô ao alvo. Isto é, todas as regras que agiram em seqüência dentro do período de análise e levaram à piora do alinhamento do robô com o alvo são sinalizadas. Para isto, verifica-se, na seqüência, a variação da distância angular entre robô e alvo causada pela ação de uma regra em relação à iteração imediatamente anterior. Todas as regras da seqüência são punidas ou recompensadas conforme esta medida.

3.3.3.3.2. Módulo de Reprodução da Evolução Disparada por um Evento de Monotonia

Todas as regras sinalizadas pelo módulo de avaliação são consideradas improdutivoas e prejudiciais ao bom desempenho do robô. Por isso, após terem sido avaliadas e identificadas como causadoras de comportamento monótono, essas regras são simplesmente substituídas por novas regras aleatórias. Ou seja, o processo reprodutivo elimina os indivíduos responsáveis pela ocorrência de um evento de monotonia e insere novos indivíduos aleatórios. As demais regras da população são conservadas intactas durante o processo.

3.3.3.4. Taxa de Procriação

Conforme dito anteriormente, a taxa de procriação determina o número de filhos produzidos por geração. Este mecanismo é responsável por manter o equilíbrio populacional entre indivíduos representantes dos comportamentos de desvio de obstáculo e captura de alvo. Um mecanismo de equilíbrio é necessário, pois há uma tendência natural de disseminação das regras de captura devido à ocorrência consecutiva de capturas de alvos quando o sistema de navegação esta operando com desempenho satisfatório (atendendo aos objetivos eficientemente).

Da maneira como foi desenvolvido o SNA, é de se esperar que o robô, ao navegar por períodos longos, deixe de colidir em obstáculos e passe a capturar alvos com maior frequência. Considerando que a cada captura de alvo a população de regras sofre um processo evolutivo que gera novos indivíduos, a tendência é que a população seja dominada somente por regras de comportamento de captura, excluindo e deteriorando regras e conhecimentos adquiridos em processos evolutivos anteriores. Com isso, os comportamentos, principalmente de desvio de obstáculos, cujas regras não contribuem para a captura de alvo e, assim, se tornam candidatas à eliminação ou reestruturação, acabam se deteriorando ao longo do tempo. Esta deterioração, também confirmada empiricamente nos experimentos, causa uma instabilidade de desempenho periódica ao robô. Quando é conseguido o desempenho desejado (capturas consecutivas e sem colisões), este não é mantido, pois o comportamento de desvio de obstáculos passa a se deteriorar até voltar a um nível primitivo. Conseqüentemente, os eventos de colisão voltam a ocorrer de forma cíclica.

Portanto, para amenizar este problema de interferências prejudiciais entre processos evolutivos seqüenciais, o mecanismo de taxa de procriação é implementado. Seu funcionamento consiste em reduzir o número de novas regras produzidas por geração quando da ocorrência de capturas consecutivas. Ou seja, de acordo com a quantidade de alvos que vão sendo capturados em seqüência, o número de filhos produzidos por geração é reduzido gradualmente. O cálculo da taxa de procriação (TxP) é feito segundo a Equação 3.6. Esta equação foi definida desta forma com base na idéia de variação do número de filhos apresentada na Tabela 3.4.

$$TxP = \begin{cases} 1/100 & \text{se } CC > 9; \\ (10 - CC)/100 & \text{se } 2 \leq CC \leq 9; \\ 10/100 & \text{se } CC < 2. \end{cases} \quad \text{Equação 3.6}$$

onde CC é o número de capturas consecutivas.

As capturas consecutivas são computadas quando o robô captura alvos em seqüência, sem sofrer colisão nem monotonia. Caso um evento de colisão ou de monotonia volte a ocorrer, CC recebe zero imediatamente e TxP passa a valer 10%, pois é um sinal de que o sistema ainda não está bem formado. A Tabela 3.4 apresenta a relação entre TxP , CC e o número de filhos por geração. Observa-se que, nos casos em que o robô capturou apenas um alvo ou sofreu colisão/monotonia ($CC=0$), a quantidade de filhos gerados é de 20, que é o valor máximo. O número mínimo possível de filhos produzidos por geração é 2, no caso de ocorrerem mais que 8 capturas consecutivas. Portanto, o sistema nunca pára de

evoluir, pois pelo menos 2 filhos são sempre produzidos pelo processo evolutivo associado à captura de alvos (caso extremo).

Tabela 3.4: Relação entre CC, TxP e número de filhos produzidos por geração com população de 200 indivíduos.

CC	1	2	3	4	5	6	7	8	>8
TxP (%)	10	8	7	6	5	4	3	2	1
Nº de Filhos	20	16	14	12	10	8	6	4	2

3.4. Experimentos e Resultados

Esta seção apresenta alguns resultados obtidos nos testes feitos com o sistema de navegação autônomo (SNA) implementado. Na realidade, este sistema foi investigado em um extenso conjunto de experimentos que podem ser encontrados em (Cazangi R. R., 2004). A intenção aqui é apenas ilustrar brevemente o desempenho do sistema. Foram conduzidas duas modalidades de testes por meio de simulações e experimentos reais, além da associação delas.

Dois experimentos reais interligados foram conduzidos, sendo que no primeiro houve o treinamento do SNA até sua maturação por meio de simulação e sua validação no ambiente real, a bordo do robô Khepera II. É importante destacar que, mesmo quando encerrada a navegação no ambiente de treinamento (simulação), o SNA continua a evoluir normalmente, mesmo no ambiente de validação (robô móvel).

A simulação para treinamento do sistema foi feita no ambiente mostrado na Figura 3.6. O Vídeo 3.1 apresenta o início de uma simulação no mesmo ambiente, mostrando como ocorre a aprendizagem do sistema por meio dos eventos já mencionados. As regras do sistema foram inicializadas aleatoriamente. O experimento durou cerca de 10 minutos, contabilizando 50 mil iterações. Durante este período, o robô colidiu 15 vezes até estabilizar este comportamento, sofreu 12 eventos de monotonia e capturou 43 alvos, sendo as 26 últimas vezes consecutivas. O desempenho do SNA, apresentado no gráfico da Figura 3.6, mostra que, depois de 45 gerações do sistema evolutivo, o robô passou a executar ambas as tarefas adequadamente e, ao final do período, já mostrava uma conduta satisfatória, capturando alvos consecutivamente e não incorrendo em colisões.

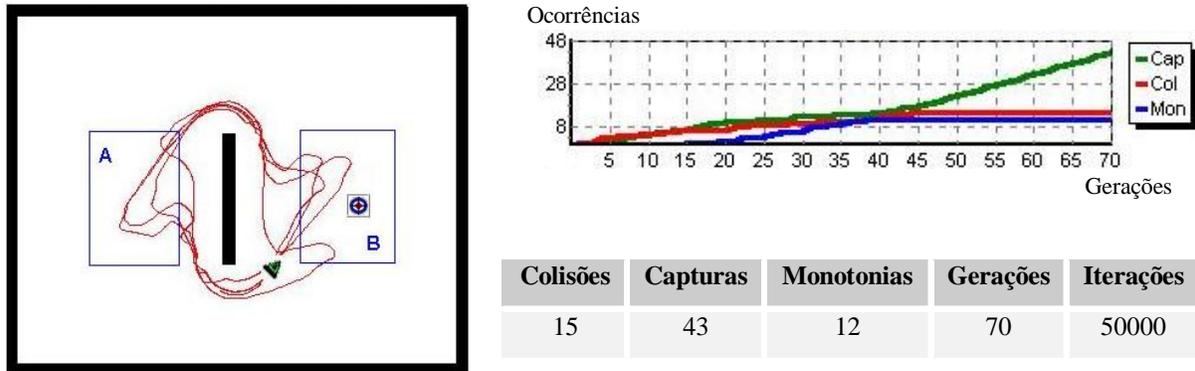


Figura 3.6: Experimento simulado em ambiente com duas regiões de alvos e um obstáculo entre elas.

Tendo havido, durante a simulação, o treinamento do SNA até sua maturação, este foi transferido para o robô Khepera II, e um novo experimento foi executado em um ambiente real, mostrado na Figura 3.7, de topologia análoga ao ambiente virtual. Quando o robô alcança a lâmpada, esta é colocada no lado oposto do ambiente, e assim por diante. Como já mencionado, apesar do treinamento do sistema ter sido feito em simulação, ao navegar pelo ambiente real o SNA continua evoluindo e se adaptando normalmente, de acordo com a ocorrência dos eventos evolutivos.

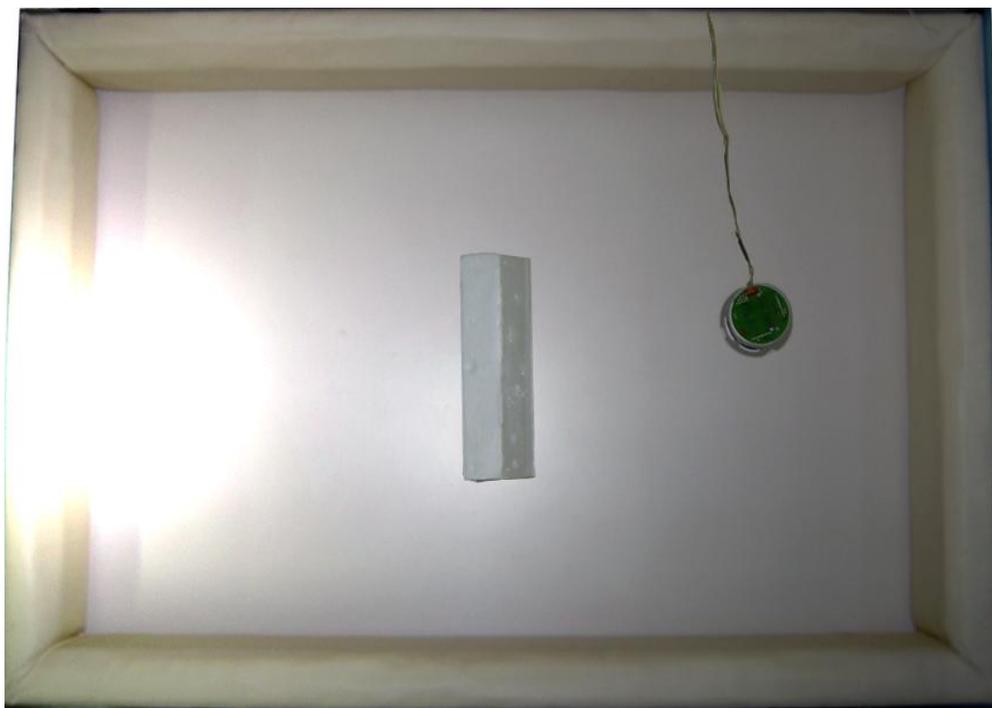
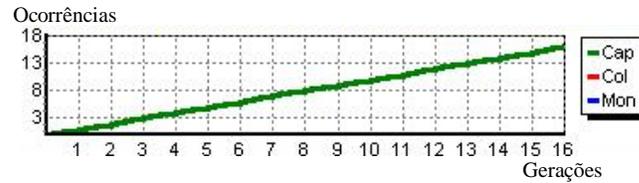


Figura 3.7: Ambiente real com um obstáculo entre o robô e a fonte de luz.

O desempenho obtido neste experimento está sintetizado na Figura 3.8. O teste durou 58 minutos e foi encerrado quando 5000 iterações foram executadas. Neste intervalo, o robô atingiu a fonte luminosa 16 vezes sem que colidisse e nem sequer sofresse eventos de monotonia.

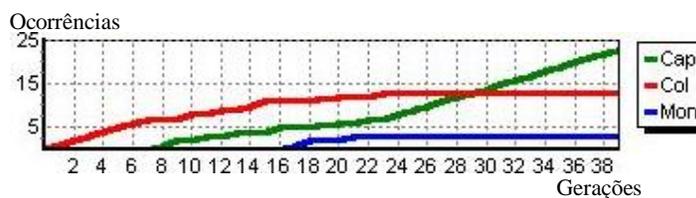


Colisões	Capturas	Monotonias	Gerações	Tempo	Iterações
-	16	-	16	58 min	5000

Figura 3.8: Desempenho do sistema em ambiente real com um obstáculo central.

Diante destes resultados, pode-se afirmar que os comportamentos que emergiram por meio da simulação foram consistentes e eficazes também na situação real. O SNA mostrou-se robusto, pois foi capaz de cumprir seus objetivos, tratando adequadamente certos fatores do ambiente real ausentes em seu treinamento virtual, como imprecisão na leitura dos sensores. Além de validar o sistema em situações reais, os dois experimentos anteriores indicam também que o simulador é suficientemente adequado ao treinamento, ou maturação, do sistema de navegação em ambientes que obedeçam a uma dinâmica não muito complexa. Ficou igualmente claro o dispendioso tempo gasto em experimentos reais, quando comparado à simulação.

O experimento seguinte não envolve simulação computacional. A evolução do sistema se dá diretamente no ambiente real, partindo da ausência de conhecimento inicial. O ambiente utilizado foi o mesmo apresentado na Figura 3.7. A performance do Khepera, exibida na Figura 3.9, ratifica a eficiência do SNA desenvolvido, desta feita sempre em situações reais. Observa-se que ocorreram 13 colisões, 3 eventos de monotonia e 23 capturas durante todo o experimento, cuja duração foi de 74 minutos (5000 passos do robô). As colisões e os eventos de monotonia cessaram a partir da geração 23.



Colisões	Capturas	Monotonias	Gerações	Tempo	Iterações
13	23	3	39	74 min	5000

Figura 3.9: Performance do robô com aprendizagem totalmente em ambiente real.

Tanto este experimento quanto o primeiro desta seção foram feitos em ambientes de mesma topologia, sendo um em modo de simulação e outro em modo real. O número de colisões necessárias para que o comportamento de desvio de obstáculos se estabilizasse foi próximo em ambos os experimentos, 15 e 13, respectivamente. A quantidade de capturas foi bem maior na simulação pelo fato do experimento ter sido mais longo em termos de iterações. Houve grande diferença em relação aos comportamentos monótonos: no experimento real, ocorreram muito poucos eventos de monotonia. Isto se explica por existir mais variação nos estímulos sensoriais no ambiente real, também motivado pelo fato dos sensores serem a todo tempo acometidos de ruídos, implicando em uma maior diversidade na ativação das regras.

O último experimento é uma continuação do terceiro e tem dois objetivos: complementar a validação do SNA em ambientes reais e examinar a capacidade de generalização do sistema em situações reais. Para isto, foi preparado um ambiente um pouco mais complexo, contendo 3 obstáculos, conforme mostra a Figura 3.10, isto é, um ambiente diferente daquele onde o robô havia sido treinado. Inicialmente, o sistema de navegação evoluído no experimento da Figura 3.7 foi transferido e colocado para controlar o robô.

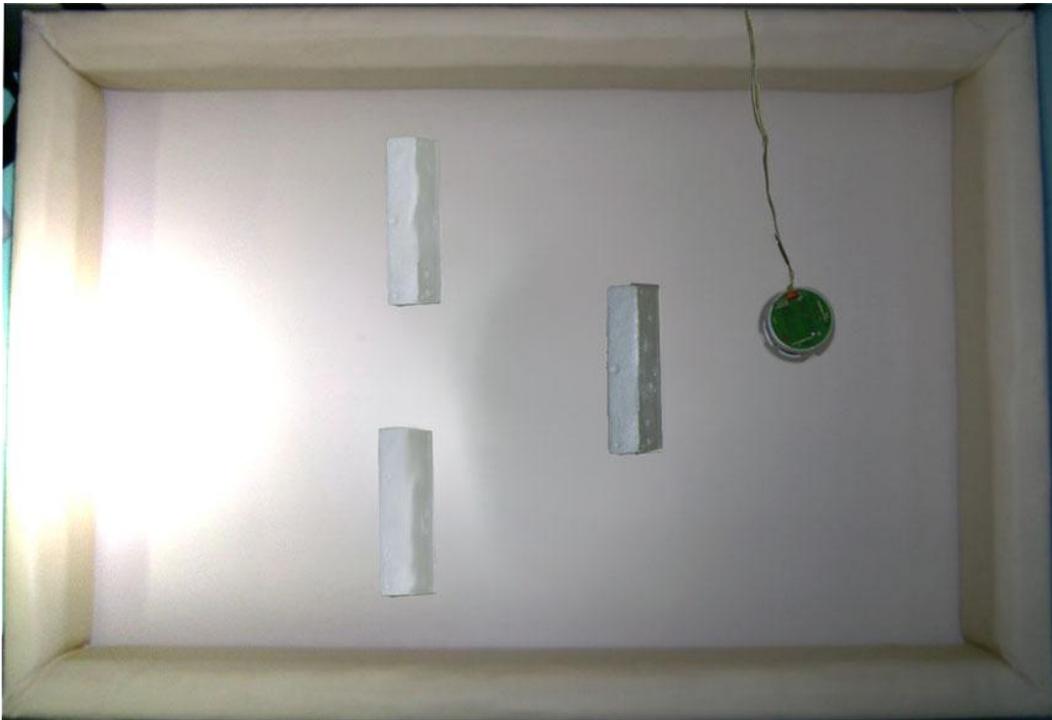
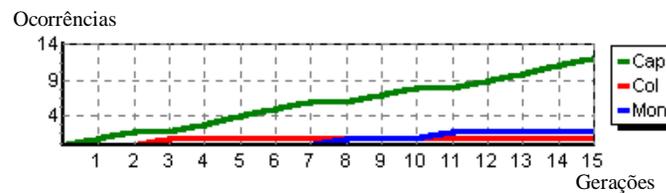


Figura 3.10: Ambiente real com três obstáculos.

A experiência durou 87 minutos e foi concluída ao completar 5000 iterações. Segundo a Figura 3.11, o robô capturou 12 alvos, sofreu 2 eventos de monotonia e colidiu uma única vez. A única colisão ocorreu na quina de um obstáculo próximo à lâmpada, possivelmente influenciada pela forte luminosidade, que provoca níveis elevados de distúrbio na leitura dos sensores de obstáculo. Quando incide sobre o robô uma luminosidade muito forte, ele se torna incapaz de detectar obstáculos e acaba colidindo. De qualquer forma, o importante é notar que o SNA foi capaz de generalizar o conhecimento produzido em um ambiente mais simples e atuar efetivamente em um ambiente diferente daquele onde ele havia se habituado anteriormente.

A duração deste experimento foi relativamente longa, prolongando-se por 1 hora e 27 minutos. Durante o período do experimento, ocorreram apenas 15 gerações. Isto se deve à complexidade do ambiente, cuja topologia continha passagens estreitas e, assim, exigia que o robô navegasse em velocidade mínima (0,2 cm/iteração). Navegando em baixa velocidade, o tempo gasto para chegar até o alvo foi bem maior que nos outros casos. Entretanto, isto garantiu uma navegação mais segura e, mais uma vez, justificou a importância do mecanismo de controle de velocidade, especialmente no caso de experimentos em ambientes reais.



Colisões	Capturas	Monotonias	Gerações	Tempo	Iterações
1	12	2	15	87 min	5000

Figura 3.11: Atuação do robô em um ambiente complexo (Figura 3.10).

3.5. Considerações Finais

Este sistema de navegação autônomo (SNA) tem como objetivo guiar, sem qualquer auxílio externo, um robô móvel por ambientes arbitrários e inicialmente desconhecidos, cumprindo simultaneamente duas tarefas: evitar colisões em obstáculos e capturar alvos dispostos arbitrariamente no ambiente de navegação.

O SNA possui capacidade de aprendizagem e adaptação que possibilitam que o robô, a partir da ausência de conhecimento inicial, adquira novos conhecimentos, proponha e coordene

comportamentos possivelmente conflitantes. O atendimento dos objetivos de navegação deve ser buscado mesmo perante situações inesperadas. Estas habilidades do SNA advêm da teoria que o fundamenta e inspira: os Sistemas Classificadores (SC) de Holland (1986), cujas propriedades de aprendizagem e adaptação são inerentes e sustentadas por mecanismos evolutivos. Diversas adequações junto aos SC originais foram incorporadas ao SNA e contemplam algumas de suas principais contribuições.

Além de evolutivo, o SNA é reativo, isto é, o sistema atua com base em informações instantâneas capturadas pelos sensores e não com base em dados estados internos ou histórico de decisões já tomadas. Já os processos evolutivos são disparados por eventos de colisão, captura e monotonia que ocorrem durante a navegação do robô. Cada um destes três eventos dispara processos evolutivos específicos responsáveis pela emergência dos comportamentos não-inatos de desvio de obstáculos e captura de alvos.

O SNA e seus diversos aspectos foram testados e validados por um extenso conjunto de experimentos de navegação. Os experimentos visaram apontar as virtudes do sistema e detectar deficiências. Embora uma grande diversidade de experimentos tenha sido realizada, eles podem ser agrupados em três classes:

- Puramente em simulação computacional;
- Mesclando simulação, até se obter um controlador hábil, com navegação empregando um robô real; e
- Puramente em um robô real.

3.6. Descrição dos Vídeos

Vídeo 3.1: Simulação que mostra o sistema aprendendo a desviar de obstáculos e capturar alvos. As colisões são indicadas por quadrados vermelhos, as capturas de alvos por quadrados verdes e os eventos de monotonia são sinalizados por quadrados dourados.

Capítulo 4: Robótica Coletiva Inspirada em Formigas

4.1. Introdução

Problemas de navegação autônoma de robôs são, sem dúvida, muito desafiadores devido a dificuldades técnicas e conceituais. Sem qualquer intervenção humana, robôs autônomos devem executar tarefas em ambientes muitas vezes desconhecidos e cuja dinâmica é imprevisível, tendo, para tanto, apenas as informações provenientes de sensores e os graus de liberdade associados aos atuadores, que nem sempre são suficientemente precisos.

A complexidade do problema se torna ainda maior quando múltiplos robôs passam a operar conjuntamente. A partir do momento em que vários robôs navegam em um mesmo ambiente, cada qual se mostra como um obstáculo móvel para os demais. A quantidade de situações e as possibilidades de interação são inúmeras. Tal complexidade justifica parcialmente o porquê de a maioria dos trabalhos e pesquisas se concentrarem em robótica individual. Apenas recentemente sistemas multi-robôs e robótica coletiva passaram a receber uma maior atenção (Arai, Pagello, & Parker, 2002). Este aumento de interesse também se deve às limitações naturais de operação de robôs isolados e às vantagens trazidas pelo trabalho coletivo. Muitas tarefas, como transporte de materiais, por exemplo, podem necessitar de mais de um robô para serem executadas. Já em outros casos, embora um robô possa ser suficiente, um maior número de elementos faz com que o desempenho seja muito melhor, especialmente se existe cooperação.

Algumas das abordagens usadas em robótica coletiva, possivelmente as mais comuns, são aquelas inspiradas na natureza. Tais abordagens se baseiam nas características sociais de insetos e animais vertebrados, principalmente as formigas, abelhas e pássaros. Esta forma de inspiração, observando os organismos que vivem em sociedade como metáforas no desenvolvimento de sistemas de robótica coletiva, é um dos principais fundamentos da área do conhecimento conhecida como Inteligência de Enxame (*Swarm Intelligence*). De uma forma sucinta, pode-se dizer que o sucesso das técnicas de Inteligência de Enxame pode ser atribuído às complexas propriedades emergentes resultantes de interações simples de múltiplos indivíduos (Bonabeau, Dorigo, & Theraulaz, 1999).

Tendo em vista a importância da interatividade para os seres que vivem em sociedade, a troca de informações entre indivíduos, ou simplesmente comunicação, tem um papel fundamental.

Analogamente, as questões relativas à comunicação devem receber especial atenção ao se desenvolver sistemas artificiais coletivos. Na literatura, há uma extensa discussão a respeito das formas de comunicação entre indivíduos simples que vivem em sociedade. O consenso é que, no âmbito computacional, a comunicação de forma indireta (implícita) traz mais benefícios que sua forma oposta, a direta (explícita) (Balch & Arkin, 1994). Tal conclusão tem forte embasamento nas extraordinárias propriedades da estigmergia, conceito cunhado por Grasse (1959) para descrever a forma de comunicação indireta observada em insetos sociais.

Muito da sofisticação vista em populações de formigas é explicada com base no uso recorrente de estigmergia, por meio de feromônios por parte desses insetos. Embora os efeitos deste fenômeno natural sejam poderosos, seu modo de funcionamento é tão simples quanto sua definição: comportamentos individuais modificam um ambiente comum, que por sua vez modifica o comportamento de outros indivíduos (Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001). Em robótica, a comunicação por estigmergia é atrativa porque não requer um processo adicional de codificação/decodificação, não opera com conhecimento estrutural e memória interna, e segue regras bastante simples. Tudo que é preciso para que um robô participe da comunicação é que ele passe perto o suficiente dos locais onde os “sinais” foram dispostos por outros robôs (Holland & Melhuish, 1999).

Robótica coletiva e estigmergia artificial estão no foco deste capítulo. Em síntese, propõe-se nesta frente de pesquisa um sistema de navegação autônomo envolvendo múltiplos robôs e que estende o sistema original, discutido no capítulo anterior, dotando-o de capacidade de comunicação indireta (estigmergia) que permite aos robôs cooperarem na execução de suas tarefas. Todos os detalhes da abordagem serão descritos a seguir.

4.2. Contextualização do Capítulo

Diferentemente do capítulo anterior, que tratou da navegação de um único robô, este capítulo contempla aspectos envolvidos em sistemas multi-robôs. Embora haja esta distinção, a abordagem proposta aqui pode ser vista como uma extensão do sistema de navegação autônomo do capítulo anterior, já que ele é usado como base, ganhando mecanismos adicionais e adaptações voltadas para aplicações de robótica coletiva.

Basicamente, todas as características de comportamento reativo e capacidade de aprendizagem do sistema desenvolvido para um único robô são herdadas pelo novo sistema utilizado aqui. A grande diferença, nestes aspectos, é que um componente fundamental é incorporado: comunicação entre robôs. Em termos cognitivos, a comunicação representa uma maior sofisticação, uma vez que um robô tem a seu dispor, no caso cooperativo, conhecimentos extras obtidos pelos outros robôs.

Especificamente neste sistema, cuja comunicação é inspirada no fenômeno de estigmergia associado às formigas, os robôs constroem trilhas de feromônio no ambiente, que são utilizadas pelos demais robôs como sinalização. Ou seja, de maneira cooperativa eles acabam por demarcar o ambiente, gerando um mapa de feromônios que pode ser visto como uma memória coletiva armazenada no próprio ambiente. Este tipo primitivo de memória pode ser classificado como de curto prazo, devido à sua natureza efêmera, resultante do processo natural de evaporação.

É certo que este sistema multi-robôs, avaliado no âmbito coletivo, apresenta um maior grau cognitivo do que o sistema original. Mesmo assim, trata-se de uma abordagem reativa, posto que, apesar de existir uma “memória” primitiva, ela é acessível apenas localmente e consumida imediatamente. Isto é, o robô detecta somente feromônios que estão próximos, informação esta que é fundida com os demais estímulos sensoriais imediatos (e.g., distância a obstáculos) e usada para tomada de decisão instantânea.

Em relação à aprendizagem, a comunicação indireta traz um desafio adicional: o sistema precisa aprender a utilizá-la proveitosamente. Ou seja, além de não saber desviar de obstáculos e capturar alvos, o robô inicialmente também não sabe como se comunicar. É necessário, por conseguinte, que o sistema aprenda quando deve liberar feromônio e o que fazer ao detectá-lo.

Na verdade, visando explorar e confrontar o impacto da incorporação de conhecimento *a priori* ou aprendizagem autônoma, duas versões do sistema são testadas. Na primeira, daqui em diante rotulada como determinística, os feromônios são depositados obedecendo a regras inatas (pré-programadas). Ao contrário, na segunda versão, chamada evolutiva, os robôs deverão aprender por conta própria a utilizar os feromônios. Obviamente, a abordagem evolutiva confia em algoritmos evolutivos para aprender.

Analogamente às formigas, espera-se que a presença de trilhas de feromônios possibilite que os robôs realizem suas tarefas com melhor desempenho. Além de uma maior eficiência no desvio de

obstáculos e captura de alvos, também deve acontecer a otimização (minimização) dos caminhos navegados pelos robôs como consequência do acúmulo de feromônios em locais apropriados.

Apenas experimentos simulados computacionalmente são realizados neste capítulo. O modelo do mini-robô Khepera II é adotado nas simulações por ser o mesmo utilizado no SNA original. Um novo sensor virtual é introduzido para detectar feromônios artificiais presentes no ambiente, assim como um atuador virtual que deposita a substância.

Embora o robô Khepera II real não seja capaz de liberar substâncias no ambiente e detectá-las, existem outros robôs com tais capacidades e que poderiam ser usados em eventuais experimentos reais. Alguns deles se utilizam de mecanismos físicos que fazem marcas no chão, por exemplo usando canetas. Numa forma mais sofisticada, substâncias solventes podem ser liberadas sobre uma superfície de modo a descolori-la (e.g. cartolina preta) (Svennebring & Koenig, 2002). Estas possibilidades de implementação em robôs reais ficam como perspectivas futuras do trabalho.

4.3. Fundamentação Biológica

Fenômenos coletivos e sistemas auto-organizáveis na natureza são as mais freqüentes inspirações para trabalhos de robótica coletiva. Estas abordagens consistem de um conjunto de técnicas motivadas pelas características dos insetos e animais vertebrados, sendo estudadas pela área de Inteligência de Enxame. Na natureza, existem inúmeros e diversificados fenômenos coletivos, de forma que a reprodução de sistemas auto-organizados em ambientes computacionais, além de contribuir para a síntese de ferramentas de engenharia de computação, pode auxiliar no estudo de fenômenos naturais (Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001)

De acordo com Bonabeau et al. (1997), a auto-organização não se baseia na complexidade individual para explicar as complexas propriedades espaço-temporais que emergem em nível de colônia. Essencialmente, a principal hipótese considera que as interações locais de indivíduos simples possam produzir comportamentos coletivos altamente estruturados.

Vários exemplos de processos auto-organizados naturais podem ser citados. Em termos de objetos inanimados, um exemplo são os grãos de areia que formam dunas. Em se tratando de seres vivos, mesmo em organismos microscópios se pode observar tais fenômenos: por meio de incontáveis interações de células, o sistema imunológico humano é capaz de se auto-organizar em resposta a

estímulos internos e externos (de Castro & Timmis, 2002). Em uma escala maior, também é possível identificar auto-organização em muitas espécies de insetos e animais superiores como peixes e pássaros, os quais apresentam habilidades para estabelecer e manter formações complexas enquanto se movem como uma única unidade social (Edelen, 2003).

Os processos de auto-organização mais pesquisados são, no entanto, aqueles relacionados a sociedades de insetos, principalmente colônias de formigas. Devido à extraordinária constatação de que a partir de indivíduos tão simples emergem capacidades coletivas complexas, as formigas têm atraído um grande interesse por parte da comunidade de Inteligência de Enxame. Outros fatores que motivam o estudo do comportamento das formigas são os seguintes (Resnick, 1994):

- As formigas são bastante comuns na natureza e, portanto, sua observação e experimentação se tornam mais factíveis;
- Elas são visíveis a olho nu e respondem bem a testes feitos em ambientes artificiais;
- Seu comportamento é regular, ou seja, não há mudanças significativas mesmo quando o ambiente sofre alterações de pequena monta.

Apesar das muitas características interessantes relacionadas às formigas, uma em especial é a que motiva esta pesquisa: a comunicação entre formigas. A maior parte dos tipos de comunicação que as formigas usam é dos seguintes tipos: química, tátil, visual e acústica. Dentre estas, a comunicação por meio de substâncias químicas é a mais comum. Holldobler & Wilson (1990) afirmam que os feromônios têm papel essencial na organização das sociedades de formigas. Supõem que, em média, uma colônia de formigas utiliza de 10 a 20 diferentes tipos de sinais, sendo que algumas espécies mais avançadas podem produzir até 20 qualidades de feromônios, cada qual com um significado distinto (Caetano, Klaus, & Zara, 2002).

A comunicação química realizada por meio da liberação e detecção de feromônios é classificada como indireta ou implícita. Embora as trilhas de feromônio sejam empregadas com o objetivo de expressar alguma informação para outras formigas, elas são indiretas por não serem direcionadas a nenhum indivíduo em particular e por não possuir um tempo específico para ocorrer. As trilhas podem ser detectadas a qualquer momento dentro de um intervalo de tempo (há o desvanecimento por evaporação) e por qualquer formiga. Também considerada implícita, pois a informação é transferida pelo ambiente, esta forma de comunicação é conhecida como estigmergia (Grasse, 1959).

Para explicar melhor o conceito de estigmergia e, conseqüentemente, o funcionamento das trilhas de feromônio na auto-organização vista em sociedades de formigas, maiores detalhes serão dados a seguir. Em geral, todo processo auto-organizável tem as interações de seus componentes como mecanismo mais importante. Ignorando por um momento a aleatoriedade intrínseca, pode-se dizer que há duas forças capazes de influenciar tais interações: realimentação positiva e negativa (Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001).

A realimentação positiva pode ser observada quando formigas isoladas encontram uma fonte de comida e liberam feromônio, formando uma trilha até seu ninho. Esta trilha atrai mais formigas que, mais uma vez, coletam comida e voltam reforçando a trilha, e assim em diante. Como resultado, a concentração de feromônio na trilha aumenta rapidamente, levando cada vez mais formigas a sair do ninho em direção à comida (Holldobler & Wilson, 1990).

A realimentação negativa, por sua vez, é responsável por equilibrar e estabilizar os efeitos da realimentação positiva, além de outras funções reguladoras. Nas sociedades de formigas, a realimentação negativa acontece em resposta a situações como: limitação no número de indivíduos, exaustão de uma fonte de alimento, competição entre fontes de alimento e evaporação de feromônios (Bonabeau, Dorigo, & Theraulaz, 1999).

Muito embora o papel destas forças no comportamento das formigas seja fundamental, sozinhas elas não seriam capazes de produzir as admiráveis características observadas em formigas. Um fenômeno natural onipresente chamado flutuações aleatórias se alia àquelas forças de modo a promover condições para a emergência de comportamentos ordenados. Dinâmicas envolvendo aleatoriedade são comuns na natureza e cruciais, pois permitem a descoberta de novos estados auto-organizados. Sabe-se que as formigas seguem as trilhas de forma imperfeita, especialmente trilhas com baixa concentração de feromônios (Deneubourg, Pasteels, & Verhaeghe, 1983). Quando um indivíduo se perde da trilha, existe a possibilidade de ele encontrar uma fonte de alimento melhor (e.g. mais próxima, mais rica ou maior) do que a que estava sendo explorada pela colônia (Pasteels, Deneubourg, & Goss, 1987).

4.4. Revisão de Trabalhos Relacionados

Embora a robótica coletiva esteja no foco de muitos trabalhos atualmente, esta linha de pesquisa era um tópico menor nas pesquisas de robótica móvel até poucas décadas atrás (Arai, Pagello, & Parker,

2002). A maior parte dos esforços era dedicada quase que exclusivamente a problemas com um único robô. Recentemente, esta tendência tem mudado para a coletividade. O aumento de interesse em sistemas multi-robô tem se dado devido a algumas razões principais (Cao, Fukunaga, & Kahng, 1997):

- Muitas tarefas são inerentemente complexas demais para um único robô executar;
- Múltiplos robôs de estrutura simples podem conduzir a uma solução mais barata e fácil do que aquela produzida por um único robô sofisticado;
- Sistemas multi-robô são, em geral, mais flexíveis e robustos que um robô operando isoladamente.

Em termos de comunicação entre robôs – em foco neste trabalho e aspecto fundamental em arquiteturas multi-robô – existem duas possibilidades: direta e indireta. De forma análoga à biologia, a comunicação direta consiste no envio e recebimento de mensagens com o objetivo de transmitir informações. Na versão indireta, também conhecida como estigmergia, o processo ocorre usando o ambiente como meio de comunicação e de forma não específica ou direcionada.

Balch & Arkin (1994) discutem ambos os métodos no uso em sistemas artificiais, concluindo que a comunicação intencional (direta) pode ser evitada quando é possível transmitir informações similares por meio de modificações impostas ao ambiente. Tal conclusão se confirma tendo em vista que, na maioria dos domínios, há um custo associado à comunicação explícita. Este tipo de comunicação requer, no mínimo, uma forma de representação padronizada e um protocolo específico para se transmitir/receber mensagens (Tambe, Adibi, Al-Onaizan, Kaminka, & Muslea, 1999). A estigmergia é atrativa por que faz uso do ambiente para armazenar informações ao invés de um modelo interno de mundo. Desta forma, as necessidades de memória do agente (robô) são reduzidas. Além disso, não é preciso se preocupar com a consistência entre mundo real e modelo de mundo dado que o mundo real serve como seu próprio e melhor modelo (Brooks R. A., 1991).

Notando as vantagens da comunicação indireta, ou estigmergia, muitos trabalhos em robótica coletiva tem sido desenvolvidos com base nesta inspiração biológica. A seguir, uma lista não-exaustiva de trabalhos relevantes é apresentada.

Wagner & Bruckstein (1997) desenvolveram um algoritmo para limpeza em que vários robôs cooperam deixando trilhas no chão. Ding, He, & Jiang (2003) utilizaram feromônios artificiais como quantificadores do nível de dificuldade de tarefas, ou seja, os robôs tentam executar suas tarefas e, de

acordo com o resultado, depositam feromônio em quantidade proporcional à complexidade. Tarefas mais difíceis, representadas por regiões com maior concentração de feromônios, tendem a atrair mais robôs para ajudar.

Em se tratando de estigmergia, boa parte dos trabalhos revisados procura reproduzir artificialmente tarefas executadas na natureza, sendo a busca por alimento feita pelas formigas a mais comum. Esta tarefa pode ser definida como a localização e coleta de objetos (ou alvos) (Drogoul & Ferber, 1992). Nos sistemas biológicos, os alvos são geralmente comida, como no caso de presas. Na implementação com robôs reais ou simulados, os objetos são dependentes do tipo de aplicação.

Edelen (2003) realizou uma investigação a respeito desta tarefa usando comunicação indireta entre indivíduos simples. Com base numa análise extensiva de modelos teóricos, técnicas para liberar e seguir trilhas são usadas na tentativa de reproduzir o processo de cooperação por estigmergia. Experimentos reais e simulação são executados, levando a resultados em que a estigmergia faz com que ocorra um aumento na eficiência coletiva e na habilidade de se adaptar a ambientes dinâmicos.

Outros tratam da busca por objetos de modos diferentes. Baseado em equações diferenciais, Sugawara & Watanabe (2002) desenvolveram um modelo para busca de uma única fonte de recursos, enquanto Nicolis & Deneubourg (1999) lidaram com o problema de competição entre múltiplas fontes. Em (Bonabeau, Dorigo, & Theraulaz, 1999) e (Deneubourg, Aron, Goss, & Pasteels, 1990), modelos probabilísticos são propostos e testados por meio de simulação. Embora autômatos celulares sejam considerados incapazes de modelar um sistema completo de busca de objetos, eles também foram aplicados para simular redes de trilhas em diversos trabalhos [(Camazine, Deneubourg, Franks, Sneyd, Theraulaz, & Bonabeau, 2001), (Ermentrout & Edelstein-Keshet, 1993)].

Um método diferente de liberação de feromônios é empregado em (Vaughan, Stoy, Sukhatme, & Mataric, 2002) para navegação entre lugares de interesse, em ambientes desconhecidos, por um time de robôs (localização e planejamento de trajetórias). Nesta proposta, há trilhas de marcos virtuais (pontos de referência) que não são dispostos no ambiente real, mas sim gravados em uma memória compartilhada e de acordo com as coordenadas locais de cada robô. Os resultados indicam que esta abordagem é robusta quanto a falhas e é capaz de convergir para as melhores rotas encontradas pelos robôs.

Planejamento de trajetórias também é o tema do trabalho em (Sauter, Matthews, Parunak, & Brueckner, 2002). Os autores desenvolveram um sistema para controlar os robôs usando feromônios sintéticos que podem ser detectados, depositados e seguidos de acordo com modelos matemáticos contendo diversos parâmetros. O ajuste desses parâmetros é feito por algoritmos evolutivos.

Agrupamento de objetos também é uma tarefa comum em colônias de formigas, o que inspira sua reprodução artificial. Por exemplo, Deneubourg et al. (1991) estudaram o desempenho de um algoritmo de ordenação distribuído, inspirado na forma como as formigas organizam sua prole. Os autores usam times de robôs simulados que se movem aleatoriamente, sem comunicação direta. A mesma proposta foi estendida para robôs reais em (Beckers, Holland, & Deneubourg, 1994). Holland & Melhuish (1999) fizeram um trabalho semelhante, envolvendo variados tipos de tarefas de agrupamento.

Comunicação estigmérgica por meio de trilhas de feromônios não é a única abordagem utilizada em sistemas coletivos. Há outros trabalhos que usam outros tipos de sinais provenientes do ambiente, ao invés de trilhas, para comunicação indireta entre robôs. Estas dicas podem ser, por exemplo, sinalizações, marcos e outros. Além disso, em alguns trabalhos os sinais não são emitidos pelos robôs, mas são elementos comuns que já existem no ambiente.

O trabalho inovador de Kube & Bonabeau (2000) lida com o problema do transporte cooperativo de caixas. Os autores implementaram um sistema robótico que imita o comportamento das formigas através de sinais sensorizados localmente. Eles adotam três tipos de sinais: para detecção de obstáculos (distâncias), de caixas (uma luz em cima da caixa) e para direcionamento do alvo (uma região iluminada perto do alvo). Cada sinal, quando detectado, dispara comportamentos inatos específicos. Os autores também investigam o papel dos diversos parâmetros do sistema (e.g., número de robôs), concluindo que um esforço coletivo coordenado é possível sem qualquer tipo de comunicação direta ou distinção entre robôs

Rybski et al. (2004) foram outros a usar pistas contidas no ambiente para busca de recursos. O sistema proposto se baseia em navegação aleatória e comportamentos reativos para desvio de obstáculos. O interesse dos autores é verificar as melhoras no desempenho do sistema dada a adição de capacidades simples de comunicação. Isto é, os robôs são equipados com faróis que, ao serem ativados, são usados como guia para outros robôs. O resultado obtido mostra que este tipo de comunicação implícita, dependendo da distância dos outros robôs e do momento de ativação do sinal, ajuda a reduzir o tempo de procura por alvos.

Agora, considere duas situações distintas: um robô consegue ver seu alvo, mas precisa primeiro se distanciar do mesmo para então ser capaz de alcançá-lo; e, o robô não pode ver o alvo e, portanto, deve navegar a esmo. Nestes problemas, sistemas puramente reativos não funcionam direito. Em (Wurr, 2003), um sistema reativo é proposto (baseado em esquemas) com um número variado de técnicas de estigmergia que envolvem o depósito e exploração de marcadores físicos como assistência para a navegação. As técnicas incluem marcas para locais de gargalo, de mínimos e máximos locais. A demarcação de gargalos consiste em sinalizar com trilhas áreas estreitas (i.e., passagens e corredores). Sinalizadores de máximo local ajudam os agentes em regiões de pobre informação sensorial, atraindo-os para que se agrupem e cooperem. Marcas de mínimo local são importantes no sentido de que antecipam ao robô a necessidade de se tomar certas ações mais apropriadas a determinadas situações do ambiente, identificadas anteriormente. As simulações indicam que estas técnicas melhoram a capacidade dos robôs de descobrirem e buscarem alvos escondidos.

Note que a maioria dos trabalhos citados é baseada em comportamentos inatos. Ou seja, os comportamentos básicos são desenvolvidos *a priori* pelo programador. O desafio para o sistema é em como gerenciar ou coordenar um número finito de comportamentos básicos. Diferentemente, na proposta desta tese, as principais contribuições estão associadas com a aquisição, no ambiente, do conhecimento necessário para promover a emergência dos comportamentos de navegação e comunicação.

4.5. Abordagem Evolutiva para Robótica Coletiva

A abordagem proposta aqui é uma extensão do sistema de navegação autônomo (SNA) apresentado no Capítulo 3, o qual focava apenas em robôs individuais atuando em isolamento. Em busca de adaptar o sistema para navegação multi-robôs, um mecanismo de comunicação indireta é proposto e algumas alterações pontuais são feitas no SNA, especificamente relacionadas ao módulo de tomada de decisão. Os principais aspectos de ambos os sistemas, original e coletivo, serão descritos a seguir.

4.5.1. Estratégia de comunicação baseada em estigmergia

Em contraste com robôs em isolamento, em que a única forma de interação é com o ambiente, em robótica coletiva há diversos agentes em potencial para interação, a qual pode se dar também de variadas formas.

As sociedades de formigas apresentam uma enorme complexidade em termos de interações, hierarquia, distribuição de tarefas, etc. Inclusive, até hoje ainda não se compreende muito bem todos os elementos envolvidos. Uma exceção é a forma de comunicação por meio de feromônios que as formigas usam, denominada estigmergia, e já dominada em seus aspectos fundamentais.

Inspirado nos processos naturais mencionados e levando em conta analogias entre múltiplos robôs e as formigas, mostradas na Tabela 4.1, um mecanismo para comunicação indireta entre robôs autônomos é desenvolvido neste trabalho. A proposta é delineada na próxima seção.

Tabela 4.1: Analogias entre formigas e robôs neste trabalho.

Características	Formigas	Robôs
Tomada de decisão	Autônoma	Autônoma
Capacidade sensorial	Reduzida ¹	Reduzida (8 sensores) ²
Comunicação	Direta e indireta ³	Indireta
Hierarquia	Descentralizada	Descentralizada
Tarefas	Busca por recursos, exploração do território, etc.	Busca por alvos e desvio de obstáculos

¹ Algumas espécies são bem limitadas e outras mais sofisticadas.

² Sensores infra-vermelhos com curto alcance.

³ Direta por contato e indireta por meio de feromônios.

4.5.2. Mecanismo de Feromônio Artificial

Em termos gerais, a idéia do mecanismo de trilhas de feromônio tem o intuito de aproveitar situações anteriores experimentadas pelos robôs de forma a ajudá-los na tomada de decisões enquanto navegam. Neste sentido, os robôs são capazes de liberar feromônios artificiais em qualquer local do ambiente no qual eles estão situados. Tais feromônios são usados para comunicação indireta entre robôs, viabilizando a implementação de processos de cooperação e aumento de desempenho no cumprimento de tarefas.

Embora algumas espécies de formigas tenham um número grande de tipos de feromônios que podem representar informações distintas, promovendo reações diferentes de acordo com outros fatores, os robôs deste trabalho lidam com os feromônios artificiais de uma maneira muito mais simplificada. Existe apenas um tipo de feromônio e sua presença ou ausência em certas regiões do ambiente tem significados bem definidos, causando efeitos específicos nas decisões dos robôs.

Por um lado, a presença de feromônios indica regiões atrativas do ambiente, as quais são interessantes para a exploração pelos robôs. Assim como no caso das formigas, espera-se que as regiões mais relevantes acabem por ser mais demarcadas e, conseqüentemente, mais visitadas pelos robôs (e vice-versa), resultando em melhoras de desempenho. Na prática, regiões relevantes são aquelas localizadas perto dos alvos e também as que estão no caminho entre eles. Sendo os robôs atraídos e visitando tais áreas, provavelmente ocorrerá um maior número de capturas de alvos por intervalo de tempo.

Por outro lado, regiões não demarcadas com feromônios são, provavelmente, menos promissoras e devem representar áreas perigosas. Por exemplo, regiões próximas de obstáculos, caracterizadas por um maior risco de colisão, terão, ao longo do tempo, menos visitas dos robôs e, portanto, não serão significativamente demarcadas (e vice-versa). Áreas do ambiente pobres em recursos (alvos) também tendem a ter menor volume de sinalização.

As trilhas de feromônio artificial podem ter diferentes níveis de concentração como resultado da repetida acumulação de feromônio em locais coincidentes, por um lado, e do efeito de evaporação, por outro. A concentração aumenta a uma taxa constante toda vez que mais feromônio é liberado no mesmo local. Um detalhe é que existe um limite máximo ou limiar de saturação para a concentração de feromônio, denotado por σ_{\max} .

Na natureza, a evaporação de feromônio é o processo pelo qual minúsculas partículas de uma substância em estado líquido ganha energia suficiente para entrar em estado gasoso. Em simulação, tal fenômeno é modelado como a redução da concentração do feromônio artificial a uma taxa constante. O limite inferior da concentração é zero.

Um aspecto adicional relacionado à implementação computacional desta proposta merece destaque. Embora o ambiente e os movimentos dos robôs sejam contínuos, a representação lógica do feromônio é discretizada. Cada posição do ambiente, para efeito de modelagem dos feromônios, é definida como uma célula, compondo um *grid*. Do ponto de vista do robô, tanto para detecção como liberação de feromônio, cada célula é considerada como uma unidade, ou seja, a concentração da célula responde por todo o espaço contínuo mapeado por ela.

4.5.3. Feromônios e Tomada de Decisão pelo Robô

Após a descrição dos aspectos principais a respeito do mecanismo de feromônio, o próximo assunto trata do papel desempenhado pelos feromônios no sistema de controle dos robôs. Apesar de o processamento químico de feromônios envolver uma dinâmica complexa em agentes biológicos, no caso artificial, as trilhas de feromônio afetam apenas uma função do sistema de navegação do robô: a forma com que ações de controle (direção e velocidade) a serem executadas a cada ciclo sensorio-motor são escolhidas. Este ponto específico será detalhado na Seção 4.6, que é dedicada à apresentação do sistema de navegação autônomo. Já se adianta que o estímulo feromonal é utilizado de uma maneira simples, influenciando os procedimentos de tomada de decisão em favor de ações de controle mais adequadas e eficientes.

4.5.4. Estratégias para Liberação de Feromônios

Duas abordagens distintas são propostas visando à liberação de feromônio no ambiente: determinística e evolutiva. Ambos os esquemas, descritos a seguir, são testados e comparados na Seção 4.7, devotada aos resultados. Observe que estas abordagens apenas afetam como os feromônios são depositados; todos os demais componentes do sistema permanecem como no sistema original.

4.5.4.1. Determinística

Nesta abordagem, a liberação de feromônio é controlada por duas regras estabelecidas *a priori* (seguindo um padrão de projeto baseado em comportamento) e está ligado à ocorrência de eventos específicos. O primeiro evento é a captura de alvo: imediatamente após qualquer captura, o robô retrocede navegando exatamente pelas últimas 10 posições, demarcando cada uma delas com feromônio artificial.

O segundo caso é mais sofisticado e envolve um conceito importante que será usado muitas vezes ao longo deste capítulo: trajetórias intactas. Uma trajetória intacta é uma trajetória livre de colisões e eventos de monotonia por parte do robô durante sua navegação de um alvo capturado até um próximo a capturar. Tal trajetória é chamada intacta justamente porque nenhum evento de aprendizado ocorre entre duas capturas consecutivas de alvo. Este tipo de rota é considerado ideal, uma vez que ele cumpre exatamente os objetivos principais do problema: capturar alvos e não colidir em obstáculos. O número de trajetórias intactas é um critério relevante para se medir o desempenho dos robôs. Além disso, o cálculo do comprimento de cada trajetória intacta fornece automaticamente a distância média

navegada pelos robôs entre dois alvos capturados, o que é uma medida de otimalidade. A distância média tem uma unidade genérica denotada por u.d. (unidade de distância). Na prática, há uma relação importante entre as trajetórias intactas e o depósito de feromônio. Na abordagem determinística, após a ocorrência de uma trajetória intacta, ela é automaticamente transformada em uma trilha de feromônios. Isto é, cada ponto da trajetória é demarcado com feromônio, como se o robô refizesse a trajetória, do seu fim para o seu começo.

4.5.4.2. Evolutiva

Ao contrário da abordagem determinística, a evolutiva não faz uso de nenhuma regra inata. Os robôs são responsáveis por decidirem como e quando depositar feromônio. Em cada iteração, além de determinar os ajustes de direção e velocidade, o sistema de navegação também controlará se deve ou não depositar feromônio na posição em que o robô se encontra.

Assim como enfatizado anteriormente, o sistema aprende como capturar alvos e desviar de obstáculos por meio de uma abordagem evolutiva. Nesta abordagem, o sistema também aprende evolutivamente como depositar feromônios de forma útil. Portanto, inicialmente, o robô acaba por liberar feromônio de modo aleatório e, então, espera-se que o sistema aprenda e aperfeiçoe sua habilidade no uso dos feromônios de acordo com suas experiências. As trajetórias intactas são utilizadas nesta abordagem apenas como medida de desempenho e para comparação com a outra abordagem. Ou seja, elas não influenciam de maneira alguma o processo de liberação de feromônios no ambiente. O processo evolutivo completo será apresentado na seção a seguir.

4.6. Sistema de Navegação Autônomo

Como já mencionado, o SNA utilizado aqui se baseia no sistema original apresentado no Capítulo 3. A estrutura e modo de funcionamento do sistema estão descritas em detalhes na Seção 3.3 e não serão repetidas aqui. Contudo, para que as extensões e novos componentes do sistema que serão delineados a seguir sejam mais facilmente compreendidos, eventualmente algumas explicações relevantes serão recapituladas.

É importante destacar que, embora se trate de robótica coletiva, cada um dos múltiplos robôs usados nos experimentos possui seu próprio e independente SNA. Além disso, quando se fala na versão

determinística do sistema, refere-se apenas à parte responsável pela comunicação, já que a parte de navegação original é exatamente a mesma em ambas as versões.

Enquanto a comunicação entre robôs é inspirada em estigmergia, o sistema em si tem raízes na teoria de sistemas classificadores com aprendizado, os quais são métodos e princípios para criação e atualização de regras de inferência, chamadas classificadores, que codificam eventuais ações a serem tomadas por um agente sob condições específicas do ambiente (Booker, Goldberg, & Holland, 1989).

A primeira modificação em relação ao sistema original, válida para a abordagem de comunicação evolutiva, se dá no cromossomo dos indivíduos (regras) da população. Além das duas partes antecedentes e duas conseqüentes originais, um novo conseqüente é adicionado, o qual codifica a ação de controle relativa à liberação ou não de feromônio. A nova parte conseqüente, chamada RP, é um vetor binário com cinco elementos: caso a maioria seja de 1, deposita-se feromônio, caso contrário, nada acontece. Um exemplo da nova estrutura das regras é mostrado na Figura 4.1.

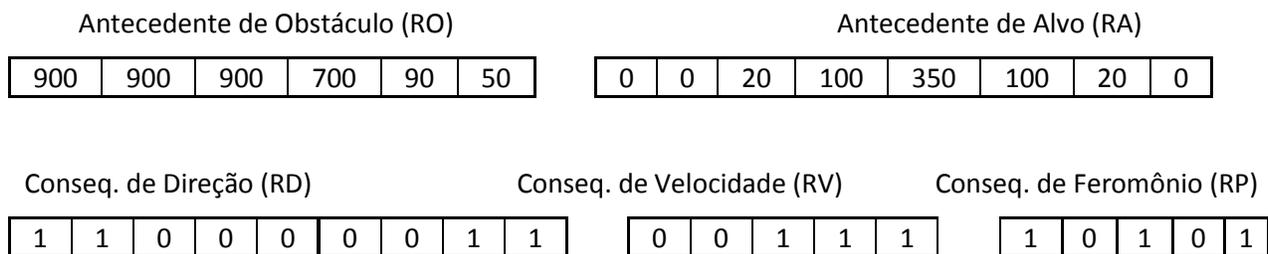


Figura 4.1: Estrutura (cromossomo) das regras utilizadas pelo sistema.

A tomada de decisão do SNA é feita pelo processo de escolha de qual regra da população atuará em cada situação. A Equação 3.1 original, usada como critério de seleção da regra vencedora, foi modificada para levar em conta a influência dos feromônios nas ações do sistema. A nova fórmula utilizada aqui está escrita na Equação 4.1:

$$S(r) = \frac{\|RO(r) - EO\|}{MaxO} + \frac{\|RA(r) - EA\|}{MaxA} CA \tag{Equação 4.1}$$

Observe que a única diferença é a inserção do termo CA, que representa exatamente o valor da concentração (intensidade) de feromônio detectada pelo robô. A idéia aqui é que CA favoreça regras candidatas que tenham maior afinidade com o comportamento de captura de alvo. Como o valor de CA é proporcional à concentração dos feromônios, quando o robô passa pelas regiões mais visitadas do

ambiente (e potencialmente com maior quantidade de feromônio) ele tende a ser estimulado a buscar alvos de maneira mais ativa.

Tendo entendido como os feromônios detectados são usados pelo sistema e como é determinada a sua ação de liberação ou não, resta explicar a forma pela qual o sistema aprenderá a usar tais artifícios. Como já descrito, o robô aprende a navegar por meio de procedimentos evolutivos iniciados por eventos de colisão em obstáculo, captura de alvo e monotonia. Estes mesmos eventos são também responsáveis por fazer o robô aprender a se comunicar. Ou seja, são incluídos nos procedimentos evolutivos passos adicionais específicos, descritos a seguir, que permitem ao sistema aprender integradamente a usar os feromônios, desviar de obstáculos e capturar alvos.

4.6.1. Colisão

Antes de qualquer coisa, deve-se informar que colisões entre robôs não causam nenhuma evolução do sistema. Isto porque, em robótica coletiva, a ocorrência de colisões entre robôs é muito freqüente, especialmente quando muitos robôs operam em conjunto. Logo, para evitar excessiva polarização da população de regras em relação a desvio de obstáculos, apenas colisões em obstáculos que não sejam robôs disparam os algoritmos evolutivos.

Além dos procedimentos originais contidos na Seção 3.3, em caso de colisão é feita a avaliação e reprodução do vetor RP (parte conseqüente de liberação de feromônios). O critério para avaliação e seleção é o seguinte: os melhores indivíduos são aqueles cuja ação não indica liberação de feromônios (maioria dos bits é 0). Assim, espera-se que, em situações de colisão (próximo a obstáculos), o robô evite o uso de feromônios, os quais estimulariam o comportamento de captura em detrimento do desvio.

Após selecionar os vetores RP mais aptos, todos são recombinados pelo operador de cruzamento de um ponto e sofrem mutação com uma baixa taxa de probabilidade. Os novos vetores gerados são finalmente reunidos aos demais vetores antecedentes e conseqüentes, produzindo a nova geração de regras.

4.6.2. Captura

Dado que o objetivo da aprendizagem por captura é diferente do por colisão, diferentes também são os procedimentos evolutivos. À exceção da etapa relacionada à evolução do conseqüente de

feromônio, todas as demais etapas estão descritas na Seção 3.3. Para evoluir os vetores RP, em caso de captura, o critério de avaliação é exatamente o oposto do de colisão. Os indivíduos com maior chance de serem selecionados são aqueles que codificam a ação de liberação de feromônio. Com isto, espera-se estimular o depósito de feromônio em regiões próximas a alvos ou em caminhos frequentemente usados para se chegar aos alvos. Feita a avaliação e seleção, os mesmos operadores genéticos já usados para a evolução por evento de colisão são empregados aqui, gerando os novos indivíduos.

4.6.3. Monotonia

Os eventos de monotonia são caracterizados quando o robô apresenta comportamentos de navegação pouco objetivos, que não levam à captura de alvos e nem sequer a colisões. Este tipo de comportamento é comum de acontecer, particularmente no início da operação. Para cessar o comportamento monótono, basicamente o mesmo procedimento usado no sistema original é adotado aqui. Ou seja, as regras que mais atuam em tais situações são deturpadas aleatoriamente, levando à mudança de atitude do robô. O mesmo ocorre com o vetor RP, que também é modificado de forma aleatória.

4.7. Experimentos e Resultados

Os experimentos apresentados a seguir foram realizados com o sistema de navegação autônomo (SNA) recém descrito, que faz uso das estratégias de depósito de feromônios como forma de comunicação indireta entre robôs. Os resultados obtidos derivam de comparações entre três estratégias de comunicação: evolutiva, determinística e ausência de comunicação.

O objetivo dos experimentos realizados é explorar, comparar e validar as propostas deste trabalho. Para tanto, diversos ambientes e variadas situações são empregadas de forma a verificar as implicações do uso, ou não, das estratégias de comunicação por estigmergia em um sistema multi-robôs, assim como analisar a capacidade do sistema autônomo em desviar de obstáculos, capturar alvos e otimizar trajetórias.

Um simulador desenvolvido pelo autor do trabalho foi utilizado em todos os experimentos contidos nesta seção. Como já dito, o modelo de robô simulado baseia-se no Khepera II. Muito embora a simulação propicie condições para a realização dos testes, é óbvio que ela não elimina a necessidade de testes com robôs reais, os quais são deixados como perspectiva futura. Cada simulação foi repetida três

vezes e a média dos valores resultantes ou número de eventos ocorridos foi considerada para análise. Seis critérios numéricos são adotados para medir o desempenho do sistema:

- Número de colisões em obstáculos;
- Número de colisões entre robôs;
- Número de capturas de alvos;
- Número de eventos de monotonia;
- Número de trajetórias intactas;
- Distância média percorrida pelos robôs entre alvos capturados.

Espera-se que o sistema apresente a maior quantidade de capturas de alvo possível, assim como o menor número de colisões em obstáculos e eventos de monotonia. Também se deseja que a frequência de ocorrência de trajetórias intactas seja alta e, é claro, que a distância média entre alvos capturados seja a mais curta possível.

Embora seja comum que muitos robôs naveguem simultaneamente em um mesmo ambiente, é importante enfatizar que cada robô é controlado pelo seu próprio SNA. Além disso, os robôs não se comunicam diretamente. Em relação aos ambientes, eles são sempre fechados, contendo obstáculos (retângulos em preto), alvos (círculos) e robôs (triângulos) dispostos de forma arbitrária.

Os alvos devem ser capturados pelos robôs sempre conforme uma seqüência pré-determinada antes do experimento. Quando um robô completa a seqüência (capturando o último alvo), esta é reiniciada com o primeiro alvo. Para facilitar a apresentação das figuras, os alvos serão mostrados com números que indicam a seqüência obrigatória de captura. Assim, é importante que fique claro que os robôs não competem por alvos, uma vez que cada robô cumpre a seqüência da sua forma. Nenhum alvo é removido ou inserido no ambiente durante a simulação. σ_{max} , que corresponde ao valor máximo da concentração de feromônio vale 30.

Finalmente, note que os robôs não possuem conhecimento inicial, ou seja, o ambiente é desconhecido e os comportamentos ou estratégias de navegação serão eventualmente desenvolvidos ao longo da simulação. Conseqüentemente, muitos eventos como colisões e monotonia devem acontecer em princípio, disparando com maior frequência a etapa evolutiva do sistema.

4.7.1. Análise paramétrica

Muitos parâmetros do sistema possuem papel importante no desempenho do mesmo, merecendo uma análise específica. A sensibilidade quanto à variação dos seguintes parâmetros é considerada: taxa de evaporação de feromônio, quantidade de regras do sistema (classificadores), condições iniciais do ambiente e número de robôs.

Observe que as seguintes abreviações serão usadas nas tabelas de resultados. A estratégia de liberação de feromônio por evolução será representada por “Ev.” e a abordagem determinística por “Det.”. Se ambos estão ausentes e a comunicação entre robôs está desabilitada, o símbolo “No” é usado.

4.7.1.1. Taxa de Evaporação

Na natureza, a evaporação de feromônios depende de fatores como características do clima, o solo e da própria colônia de formigas. É sabido que as trilhas de feromônio podem persistir de muitas horas até muitos meses (Bonabeau, Dorigo, & Theraulaz, 1999). Portanto, não há uma taxa específica de evaporação, corroborando a plausibilidade de se variar tal taxa também no sistema artificial.

Por um lado, altas taxas de evaporação podem levar a trajetórias mais aleatórias; por outro lado, taxas baixas demais podem causar convergência a rotas sub-ótimas. Estas circunstâncias motivam a investigação de qual a taxa mais adequada para o sistema proposto.

A evaporação de feromônio é modelada como uma subtração simples, reduzindo o valor da taxa por iteração da concentração atual da trilha. Três opções são consideradas: 1×10^{-5} , 5×10^{-5} e 1×10^{-4} . No ambiente mostrado na Figura 4.2, as simulações foram repetidas três vezes para cada caso e para cada uma das estratégias de comunicação (uma delas pode ser assistida no Vídeo 4.1). Havia quatro robôs na simulação, a qual durou 150 mil iterações. Os resultados finais estão na Tabela 4.2.

Analisando a Tabela 4.2, fica evidente que o aumento da taxa de evaporação causa diminuição de desempenho em termos de frequência de capturas e distância média entre os alvos. O fato é que taxas mais altas impedem a ocorrência de trilhas com concentração feromonal adequada, não permitindo que haja convergência para trajetórias mais curtas, o que leva à degradação do comportamento de captura de alvos. Em relação a eventos de colisão e monotonia, a influência da taxa de evaporação constatada foi insignificante.

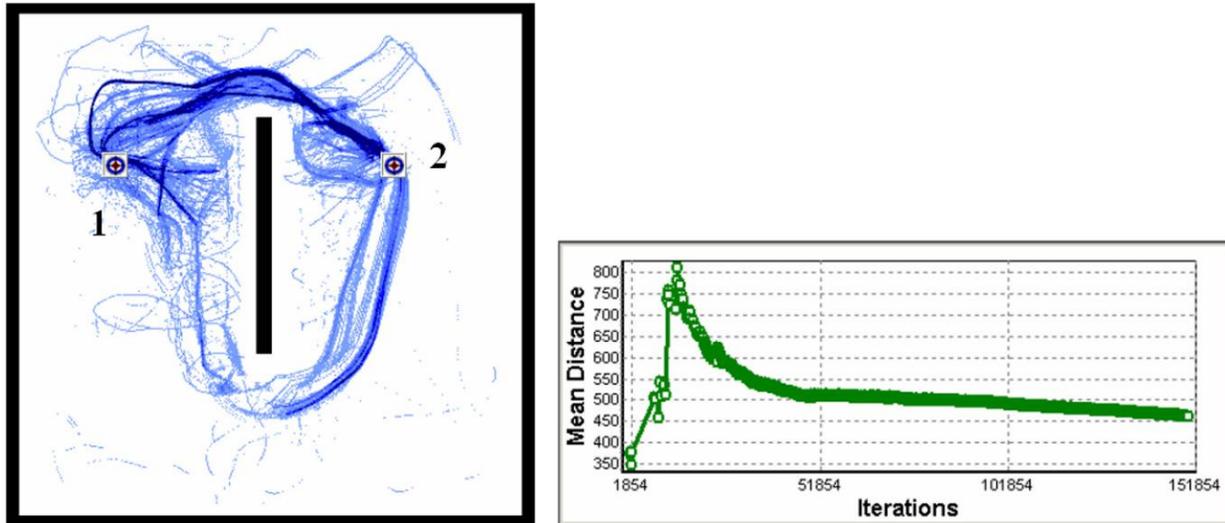


Figura 4.2: Ambiente com trilhas de feromônio (esquerda) e gráfico da variação da distância média entre capturas (direita), para a simulação com melhores resultados usando a abordagem evolutiva (veja também o Vídeo 4.1).

Além de identificar a taxa de evaporação de 1×10^{-5} por iteração como a opção mais indicada (sendo adotada em todas as simulações adiante), este primeiro grupo de experimentos também aponta claras vantagens da comunicação por estigmergia sobre a ausência total de comunicação. Muito embora a abordagem determinística pareça ser menos sensível à evaporação (em termos de distância média), a abordagem evolutiva mostrou-se, em geral, a mais eficiente.

Tabela 4.2: Resultados dos experimentos com variação da taxa de evaporação de feromônio.

Taxa de Evaporação (por iteração)	1×10^{-5}		5×10^{-5}		1×10^{-4}		No
	Ev.	Det.	Ev.	Det.	Ev.	Det.	
Monotonia	110,33	152,67	146,33	128,67	166	213,67	176,67
Colisão em Robôs	77,33	173,67	65,33	77,33	128,67	121	54,67
Colisão em Obstáculos	167,67	161,33	154	135,33	167	180,33	167
Capturas	861	695,33	713	704,67	567,33	493,67	454,67
Trilhas Intactas	740,67	562,33	614,33	601,67	435,33	359,33	333
Distância Média	491,23	563,72	501,80	560,62	604,67	574,06	650,36

A distância média mais curta obtida com evolução foi 461.78, sendo exatamente o caso apresentado na Figura 4.2. No ambiente, há dois caminhos de um alvo para o outro; os robôs claramente visitaram o mais curto mais freqüentemente, fazendo com que ele tivesse maior concentração de feromônio (tons mais fortes na figura). Na mesma figura, o gráfico à direita mostra a curva com a variação da distância

média ao longo do tempo (cada pequeno círculo indica a ocorrência de uma trajetória intacta). Inicialmente, as capturas aconteceram raramente, mas, com seu aumento (e conseqüente aumento da concentração das trilhas), a distância média tende a valores menores.

A Figura 4.3 mostra os resultados da melhor simulação realizada com a abordagem determinística (distância média = 530.16). Nesta simulação, o caminho mais curto não prevaleceu significativamente. Comparando a Figura 4.2 com a Figura 4.3, é possível notar os efeitos diferentes promovidos por cada estratégia de comunicação. Enquanto a evolutiva tende a convergir para uma única, mais concentrada trilha, a determinística produz várias trilhas que se destacam. Isto é, a interação indireta entre os robôs é mais fraca no segundo caso, mas ainda se mostra melhor do que a ausência de interação (comunicação).

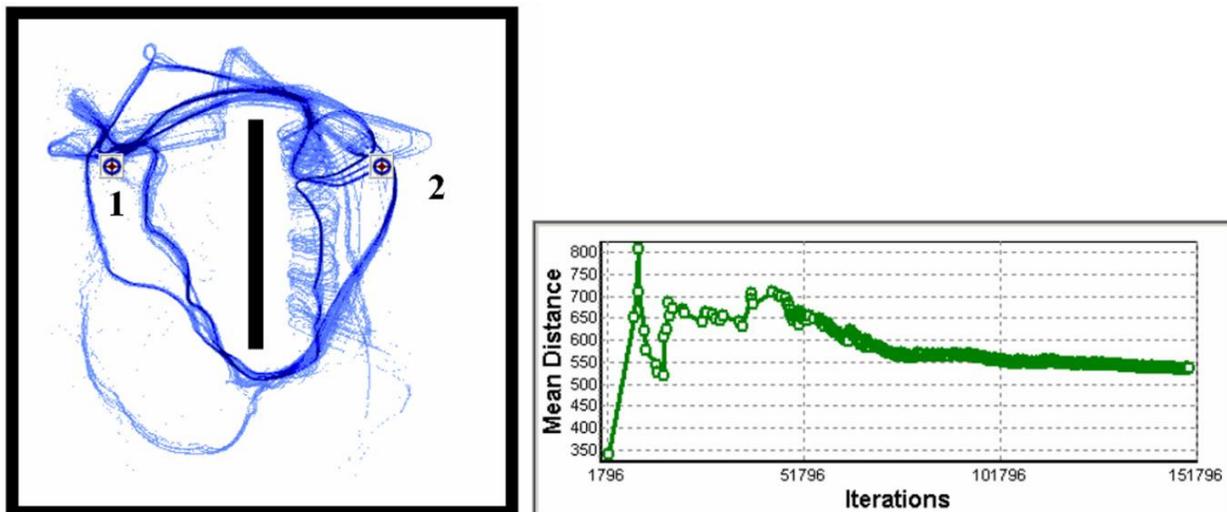


Figura 4.3: Ambiente com trilhas de feromônio (esquerda) e gráfico da variação da distância média entre capturas (direita), para a simulação com melhores resultados usando a abordagem determinística.

4.7.1.2. Complexidade do Sistema

Grande parte da admiração que os insetos sociais atraem está associada a sua capacidade de fazer coisas complexas, sendo eles organismos tão simples. Neste sentido, uma questão pode ser levantada: qual é a influência da complexidade individual no comportamento coletivo? Para tratar desta questão, experimentos específicos são propostos nesta seção.

A complexidade do SNA dos robôs está fortemente vinculada à quantidade de regras, ou classificadores, presentes no sistema. Buscando examinar o desempenho coletivo de navegação quando

sistemas mais simples estão controlando os robôs, o tamanho da população de regras é variada de 100 (valor original) até 40, ao longo de quatro experimentos. Um ambiente com quatro alvos e quatro robôs (veja Figura 4.4) foi configurado para as simulações, as quais tiveram duração de 150 mil iterações. Os resultados estão agrupados na Tabela 4.3.

Tabela 4.3: Resultados obtidos por sistemas de navegação contendo de 40 a 100 classificadores.

Nº de Classificadores	100			80		
	Ev.	Det.	No	Ev.	Det.	No.
Versão						
Monotonia	201,67	199	201	153,80	190,33	191,33
Colisão em Robôs	77,33	63,33	83,67	39,60	45,67	164,33
Colisão em Obstáculos	208,67	178,67	175,33	161,40	157	171,33
Capturas	733	679,33	589	869,40	669,67	667,33
Trilhas Intactas	585,67	546	456	745,40	544,67	531,33
Distância Média	381,84	414,66	426,72	388,10	415,36	440,97
Nº de Classificadores	60			40		
	Ev.	Det.	No	Ev.	Det.	No.
Versão						
Monotonia	185,67	172,60	176,50	172	216,67	198,60
Colisão em Robôs	60	69,33	48,75	27	85,33	55,60
Colisão em Obstáculos	178	160,67	153,25	159	180	167,80
Capturas	658,33	616,67	582,50	674,67	470	553,80
Trilhas Intactas	523,33	481,33	468,75	536,33	343,33	428,80
Distância Média	420,15	449,32	441,79	437,15	459,33	431,28

Quando os robôs não se comunicam, a complexidade do sistema parece não interferir muito no desempenho coletivo (em termos de distância média e trilhas intactas). Já em ambas as abordagens com comunicação, o desempenho obtido com 100 regras se mantém similar ao caso de 80 regras. Com 60 e 40 regras, entretanto, há uma degradação séria na distância média obtida das duas abordagens, sendo que a determinística se mostra a mais sensível (principalmente em relação às trajetórias intactas).

Nesta seção, uma vez mais a abordagem evolutiva atingiu resultados superiores à determinística. Tal superioridade é ainda maior quando se leva em conta a estratégia de ausência de comunicação, mas apenas para 100 e 80 regras. A melhor distância média obtida para o ambiente testado foi 346,68 com a abordagem evolutiva. A imagem final desta melhor simulação é apresentada na Figura 4.4.

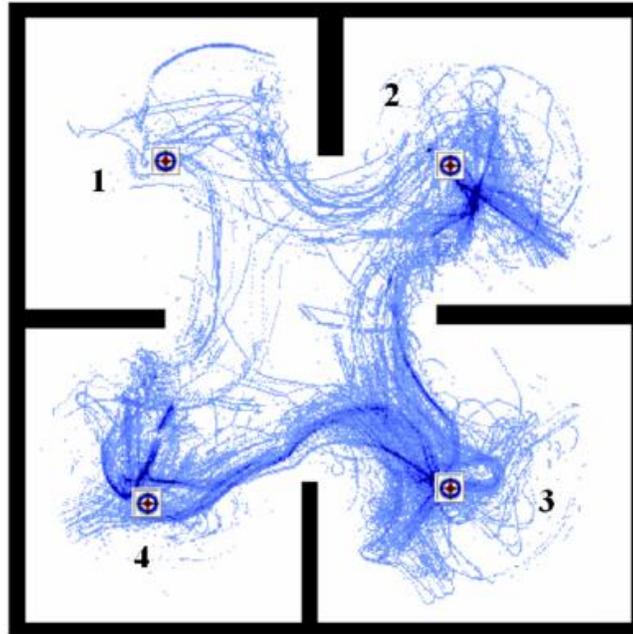


Figura 4.4: Conformação final das trilhas de feromônio para a simulação com melhores resultados usando a versão evolutiva.

4.7.1.3. Posição Inicial

Nesta seção, as condições iniciais da simulação são discutidas. Adotando o mesmo ambiente da seção anterior (com quatro robôs, quatro alvos e 150 mil iterações), o SNA apenas com a abordagem evolutiva é testado em quatro situações ligeiramente distintas. Em cada uma delas, os robôs são posicionados inicialmente em um quadrante diferente do ambiente, onde também se inicia a seqüência de alvos a serem capturados. As trilhas de feromônio, ao final das simulações, são mostradas na Figura 4.5.

Observando a Figura 4.5, nota-se que as regiões correspondentes aos quadrantes em que os robôs foram inicialmente localizados recebem um menor depósito de feromônio quando comparadas às demais. Este fato se justifica por uma razão simples. A região onde os robôs começam a simulação é, em geral, o local onde eles sofrem mais colisões devido aos seus sistemas se encontrarem em estado bastante primitivo. Com a ocorrência de colisões os sistemas evoluem e, como mencionado na descrição do mesmo, o aprendizado por meio de colisões produz regras que tendem a não liberar feromônios naquela área (considerando que tal região não é atrativa por causa das colisões). Logo, a ativação freqüente de regras anti-colisão em tais áreas faz com que os robôs raramente depositem feromônios ali. Por outro lado, as áreas localizadas nos outros quadrantes são consideradas menos perigosas pelos robôs e, logo, são mais exploradas obtendo maior concentração de trilhas de feromônio.

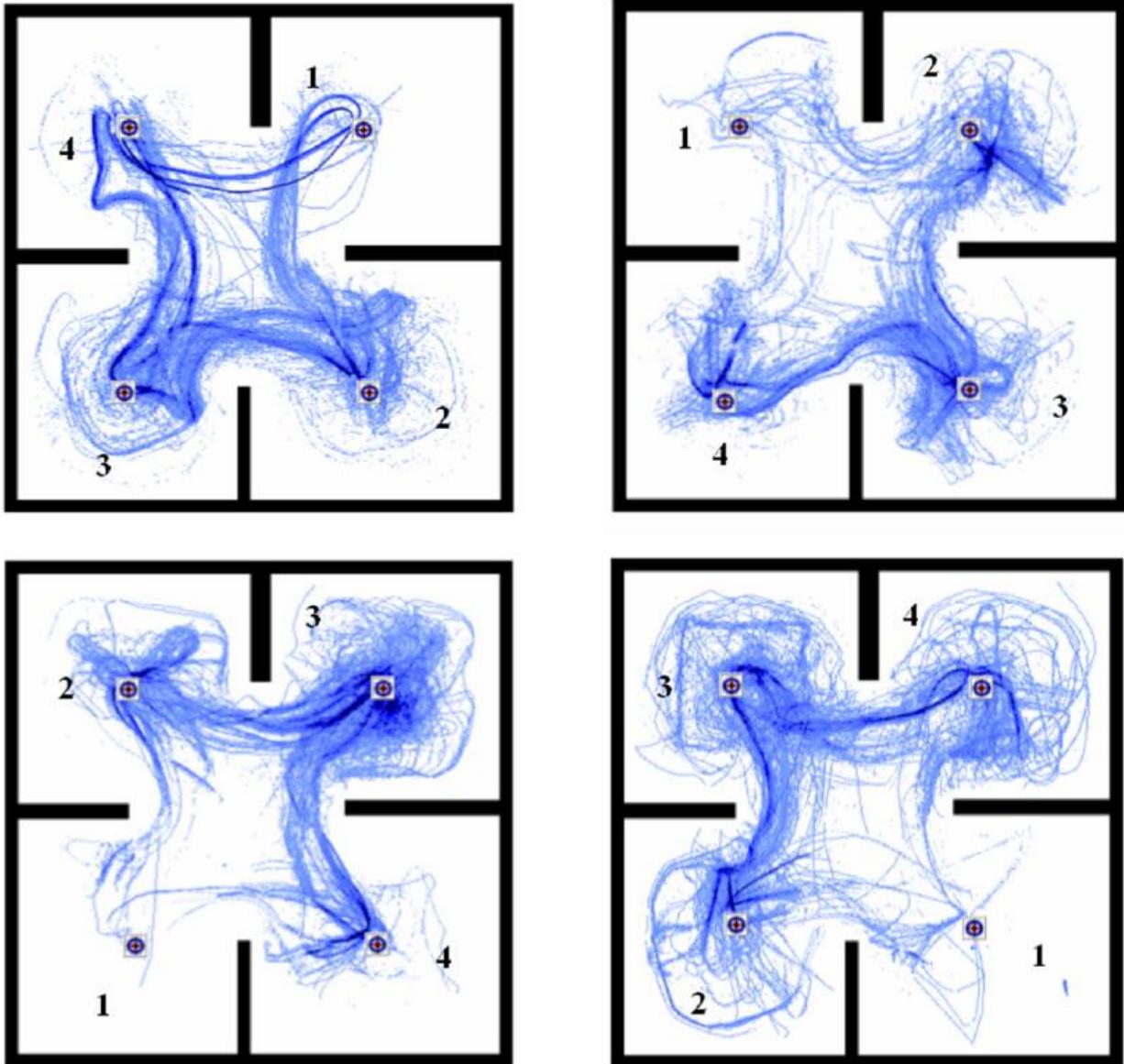


Figura 4.5: Simulações em que os robôs foram inicialmente posicionados no 1º quadrante (alta esquerda), 2º quadrante (alta direita), 3º quadrante (baixa esquerda) e 4º quadrante (baixa direita).

Apesar de a condição inicial analisada ter provado influenciar o comportamento aparente dos robôs, os efeitos no desempenho são pouco relevantes. Os resultados numéricos foram semelhantes nas quatro situações. O número de capturas de alvo variou de 1011 até 1254, e a distância média foi de 332,25 (obtida na simulação do 1º quadrante) até 346, com menos de 5% de desvio padrão. Isto é, pode-se dizer que o sistema é robusto quanto a diferentes condições iniciais deste tipo.

4.7.1.4. Número de Robôs

A coletividade é essencial para os insetos sociais. Existem espécies de formigas cujo número de indivíduos vivendo na mesma colônia pode chegar a 300 milhões (Caetano, Klaus, & Zara, 2002). É claro que, em robótica, é atualmente inconcebível se trabalhar um número tão grande de robôs. Nesta seção, será examinado o comportamento e desempenho do sistema proposto quando quantidades variadas de robôs atuam juntos.

A configuração do ambiente para os experimentos desta seção é a mesma usada na seção sobre evaporação. Há dois alvos a serem capturados alternadamente e cada simulação tem 150 mil iterações. Experimentos com 4, 6, 8, 12 e 20 robôs são realizados e seus resultados são mostrados na Tabela 4.4. É pertinente informar que um ambiente análogo aos demais, mas de tamanho maior, foi usado na simulação com 20 robôs, a qual durou 300 mil iterações.

Primeiramente, quanto maior o número de robôs, mais colisões entre robôs ocorrem, como pode ser visto na Tabela 4.4. Obviamente, tendo mais robôs em um ambiente de tamanho fixo implica em existir menos espaço livre e, logo, mais colisões. Excetuando-se o caso com 20 robôs, cujo ambiente era diferente, em média, o número de colisões entre robôs para a abordagem evolutiva cresceu 1,41 vezes quando robôs adicionais foram inseridos no ambiente; 1,31 vezes para a abordagem determinística, e 1,8 vezes para ausência de comunicação. Portanto, os robôs, ao não se comunicarem de forma alguma, acabam por sofrer um elevado número de colisões entre eles.

Menos espaço livre também causa mais colisões em obstáculos e eventos de monotonia, sendo que os resultados destes eventos seguem um padrão parecido para todas as três abordagens. Em termos de captura de alvo, de novo a abordagem evolutiva prevaleceu sobre as demais, sempre obtendo as distâncias médias entre alvos mais curtas. É interessante perceber, que quando uma maior quantidade de robôs navega pelo ambiente, há uma tendência geral de se obter trajetórias médias mais curtas do que quando se considera grupos menores. Esta tendência é mais forte nas abordagens com comunicação pelo fato de que mais robôs depositando feromônio no chão aumenta a probabilidade de se encontrar e estabelecer trilhas promissoras. As simulações com melhores resultados para o caso de 12 robôs são dispostas à esquerda (abordagem evolutiva) e à direita (abordagem determinística) da Figura 4.6.

Tabela 4.4: Resultados dos experimentos com 4, 6, 8, 12 e 20 robôs. Um ambiente maior foi usado para 20 robôs.

Robôs	4			6			8		
	Ev.	Det.	No	Ev.	Det.	No	Ev.	Det.	No
Monotonia	201,67	199	201	217,33	201	207	266,67	290,33	338
Colisão em Robôs	77,33	63,33	83,67	144,33	90,67	145	175	174	322,67
Colisão em Obst.	208,67	178,67	175,33	290,33	268	258	362,33	352,67	350,33
Capturas	733	679,33	589	946	826,67	849,67	1367,33	1218,33	955,67
Trilhas Intactas	585,67	546	456	737,67	629,67	644,33	1089,33	959	698,67
Distância Média	381,84	414,66	426,72	541,23	620,61	650,36	508,31	555,94	615,30

Robôs	12			20		
	Ev.	Det.	No	Ev.	Det.	No
Monotonia	496,33	485,33	416,33	1964	2517	2355
Colisão em Robôs	331	313,67	483	213,33	232,67	303
Colisão em Obst.	772,67	614	498,33	503,67	588,67	540
Capturas	1640,33	1585	1506,67	3548,33	2144,33	2541,33
Trilhas Intactas	1154,33	1101	1154,67	3042,33	1624,67	2029,33
Distância Média	568,36	596,25	628,72	705,94	732,83	756,18

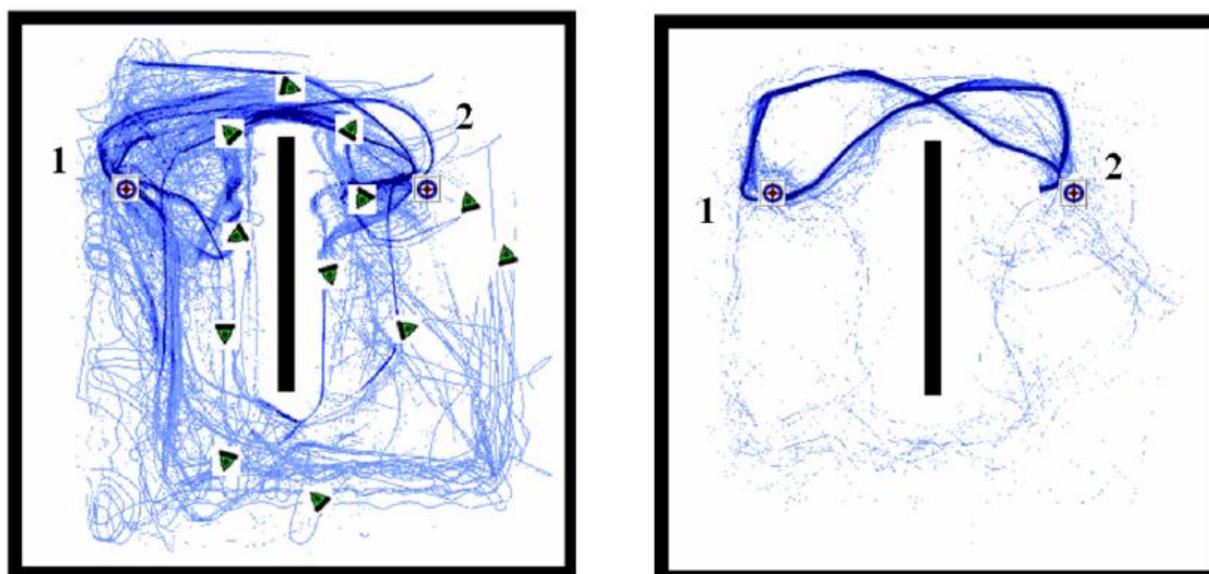


Figura 4.6: Experimento evolutivo (esquerda) e determinístico (direita) referentes às melhores simulações com 12 robôs. Note que os robôs não são mostrados à direita apenas por uma falha na captura da imagem, sem outras implicações.

Os experimentos com 20 robôs são considerados um caso especial, não levado em conta na discussão anterior. Mesmo assim, obedecendo às devidas proporções, as conclusões anteriores podem ser estendidas para este caso, isto é, a abordagem evolutiva superou as demais por uma ampla margem. A Figura 4.7 apresenta as imagens finais das simulações com 20 robôs para o caso evolutivo, à esquerda, e determinístico, à direita (distâncias médias iguais a 667,34 e 691,02, respectivamente). Pelas trilhas com mais forte concentração de feromônio, é possível notar que ambos convergiram para os lados do ambiente onde se tem os caminhos mais curtos. Embora se note que a convergência das trilhas do

sistema determinístico seja mais homogênea, os arcos que representam as trajetórias são mais longos do que aqueles obtidos pela versão evolutiva, o que justifica a diferença nas distâncias médias.

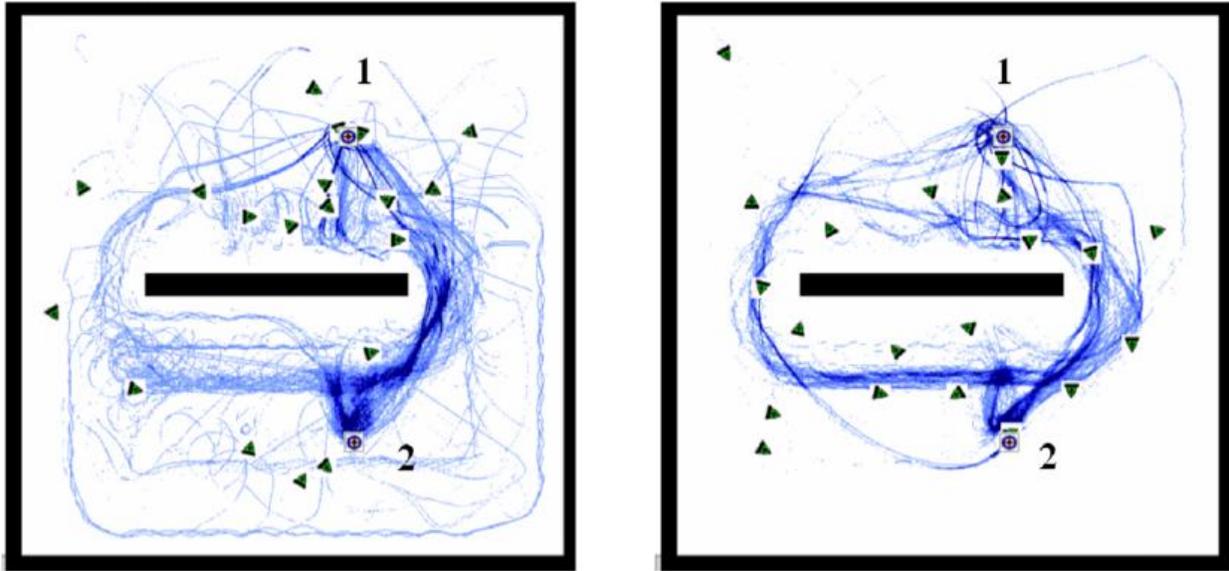


Figura 4.7: Experimento evolutivo (esquerda) e determinístico (direita) referentes às melhores simulações com 20 robôs.

4.7.2. Experimentos Adicionais

Nesta seção, são considerados três outros experimentos. Em todos eles, a taxa de evaporação é 10^{-5} e o sistema de navegação dos robôs possui 100 regras. Os outros parâmetros são escolhidos caso a caso.

4.7.2.1. Experimento I

O ambiente da Figura 4.8 foi configurado com 16 alvos, seis robôs, e as simulações tiveram duração de 300 mil iterações. Os robôs devem capturar um alvo externo, a seguir um alvo interno, e assim por diante. Analisando os resultados obtidos, apresentados na Tabela 4.5, pode-se ver que os eventos de colisão em obstáculo e monotonia ocorreram de uma maneira equivalente para as três abordagens. No entanto, a estratégia de ausência de comunicação teve quase o dobro do número de colisões entre robôs. Uma explicação possível é que, nas demais abordagens os robôs formam e seguem trilhas diferentes em face de situações conflitantes (e.g., dois robôs navegando em direções contrárias), o que ajuda a evitar colisões entre robôs. Em relação à distância média, novamente a abordagem evolutiva se saiu melhor que as demais.

Tabela 4.5: Resultados do experimento realizado no ambiente apresentado na Figura 4.8.

Versão	Ev.	Det.	No
Monotonia	403,00	415,33	467,00
Colisão em Robôs	70,00	77,67	123,00
Colisão em Obstáculos	866,33	835,33	803,67
Capturas	2588,33	2626,33	2072,00
Trilhas Intactas	1936,00	2008,33	1498,00
Distância Média	369,04	380,91	418,90

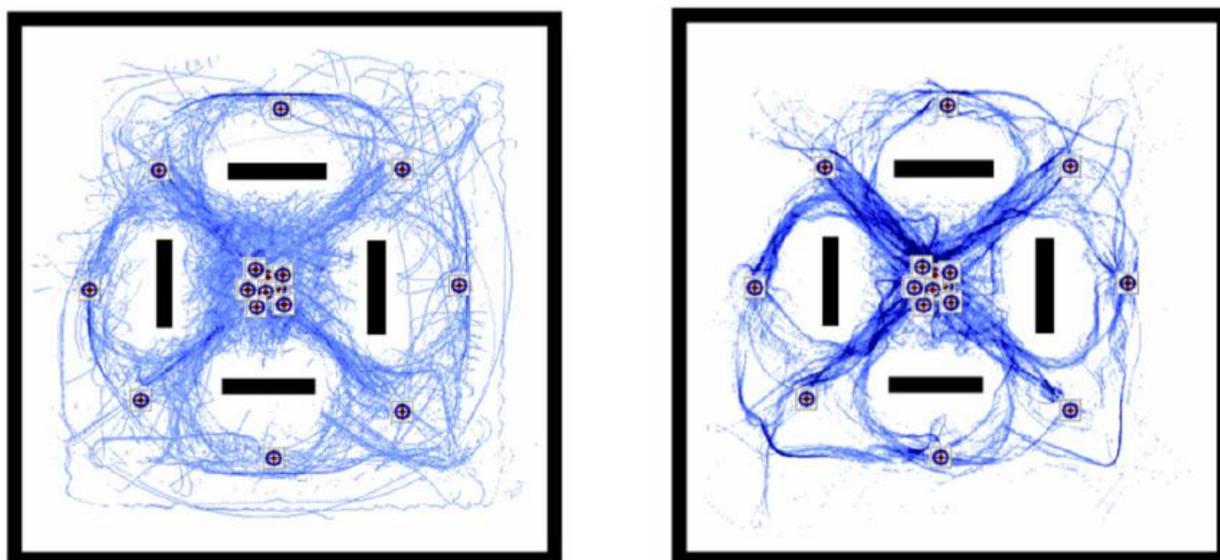


Figura 4.8: Melhores simulações para os experimentos com abordagem evolutiva (esquerda) e determinística (direita).

4.7.2.2. Experimento II

Neste experimento, o ambiente da Figura 4.9 é adotado, contendo quatro alvos e dez robôs. Claramente, o melhor caminho para capturar alvos está na diagonal, e o objetivo é verificar se os robôs realmente convergem para esta rota. As simulações duraram 500 mil iterações cada e os resultados estão resumidos na Tabela 4.6, com números favoráveis à abordagem evolutiva.

Tabela 4.6: Resultados do experimento realizado no ambiente apresentado na Figura 4.9.

Versão	Ev.	Det.	No
Monotonia	768,00	1018,33	888,00
Colisão em Robôs	83,33	186,67	101,67
Colisão em Obstáculos	723,67	884,00	773,67
Capturas	6272,00	4807,00	4651,33
Trilhas Intactas	5663,67	4086,33	3991,33
Distância Média	512,82	549,38	606,82

Na Figura 4.9, comprova-se que os robôs visitaram mais freqüentemente a diagonal do ambiente, depositando feromônio em larga escala. As trilhas espúrias ao redor dos obstáculos são resultado de flutuações aleatórias, as quais serão discutidas em mais detalhe no próximo experimento, e também devido ao fato de que não há espaço suficiente para todos os dez robôs navegarem simultaneamente no corredor diagonal, a não ser que eles naveguem em conformações bem específicas. Portanto, alguns dos robôs são forçados a capturar alvos pelos corredores externos.

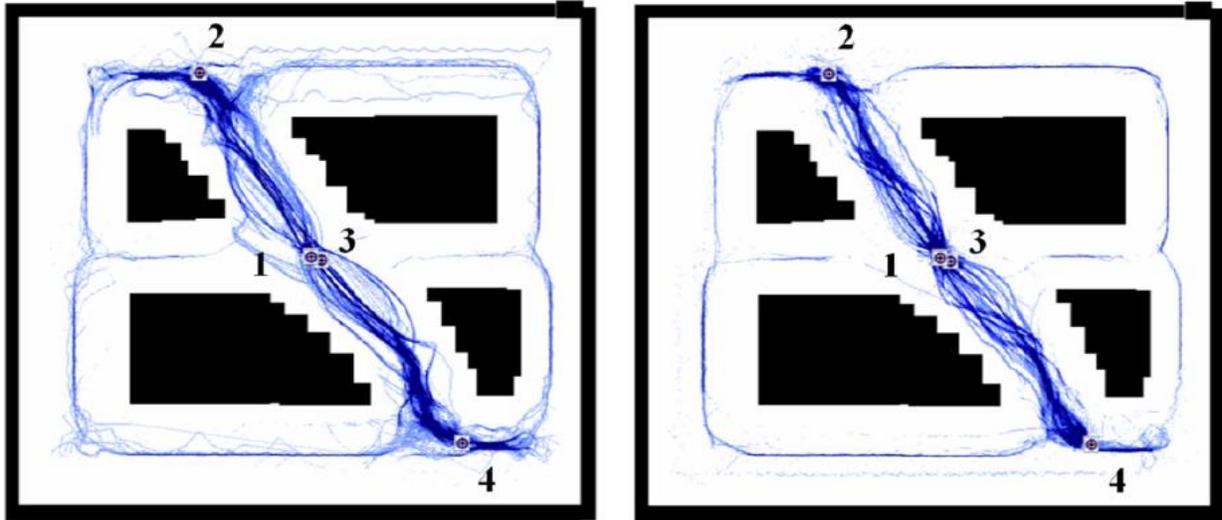


Figura 4.9: Melhores simulações para os experimentos com abordagem evolutiva (esquerda) e determinística (direita).

4.7.2.3. Experimento III

Este é um experimento de longa duração (1,2 milhões de iterações) executado no ambiente exibido na Figura 4.10. Há seis alvos e quatro robôs. Os resultados de simulação estão na Tabela 4.7, os quais indicam maior eficiência da abordagem evolutiva em comparação com as outras. Diferentemente do que aconteceu nos experimentos anteriores, neste as versões estigmérgicas obtiveram mais colisões em obstáculos que a versão sem comunicação. Devido à longa duração do experimento, muitas trilhas de feromônio acabaram por se acumular muito próximo dos obstáculos, levando a uma maior iminência de colisões. Pode se sugerir, portanto, que em simulações muito longas o uso de uma taxa maior de evaporação seja adequado para evitar o aumento das colisões. Parte deste experimento foi capturado em vídeo e pode ser visto no Vídeo 4.2.

Tabela 4.7: Resultados do experimento realizado no ambiente apresentado na Figura 4.10.

Versão	Ev.	Det.	No
Monotonia	1447,00	1431,00	1523,67
Colisão em Robôs	267,67	211,67	282,33
Colisão em Obstáculos	1641,67	1462,67	1377,33
Capturas	5209,33	3876,33	4259,00
Trilhas Intactas	3924,00	2757,00	3110,67
Distância Média	551,53	604,02	617,70

Examinando mais atentamente a Figura 4.10, nota-se que as trilhas com maior concentração ficaram mais largas do que o que vinha acontecendo. Além disso, várias trilhas com baixa concentração de feromônio, aparentemente desordenadas, podem ser vistas, principalmente na simulação evolutiva. Este fenômeno pode ser justificado pela aleatoriedade intrínseca do sistema, o que pode ser visto como uma analogia às flutuações aleatórias geralmente observadas junto aos insetos sociais (Deneubourg, Pasteels, & Verhaeghe, 1983). Esta hipótese se sustenta também com base nos resultados dos experimentos anteriores, uma vez que a abordagem evolutiva sempre faz com que ocorra um maior número de trilhas espúrias. É até provável que esta característica seja uma das razões pelas quais esta abordagem alcança trajetórias mais curtas.

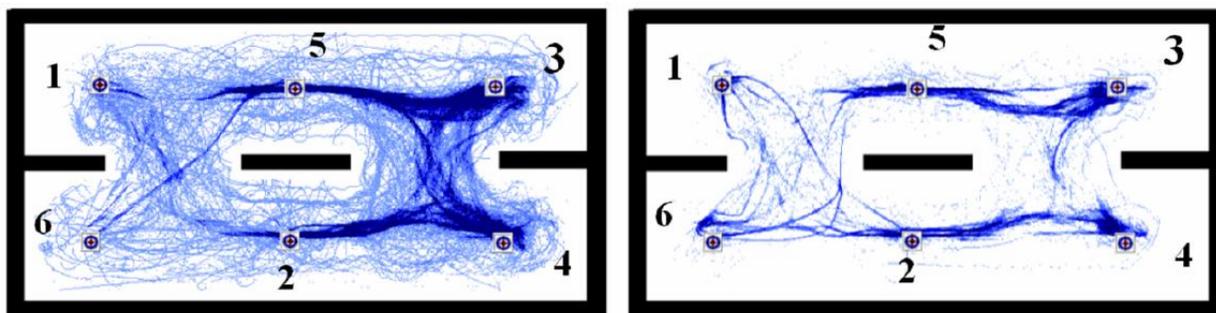


Figura 4.10: Melhores simulações para os experimentos com abordagem evolutiva (esquerda) (veja também o Vídeo 4.2) e determinística (direita).

4.7.3. Adaptabilidade

A proposta desta seção é estudar a capacidade de adaptação do sistema de navegação. O experimento é mostrado na Figura 4.11, consistindo de 100 mil iterações. O ambiente original (à esquerda da figura) foi modificado exatamente no meio da simulação, sendo um obstáculo central inserido entre os dois alvos. Tal evento forçou os quatro robôs a se adaptarem para continuarem a capturar alvos. Neste experimento, a abordagem evolutiva é usada.

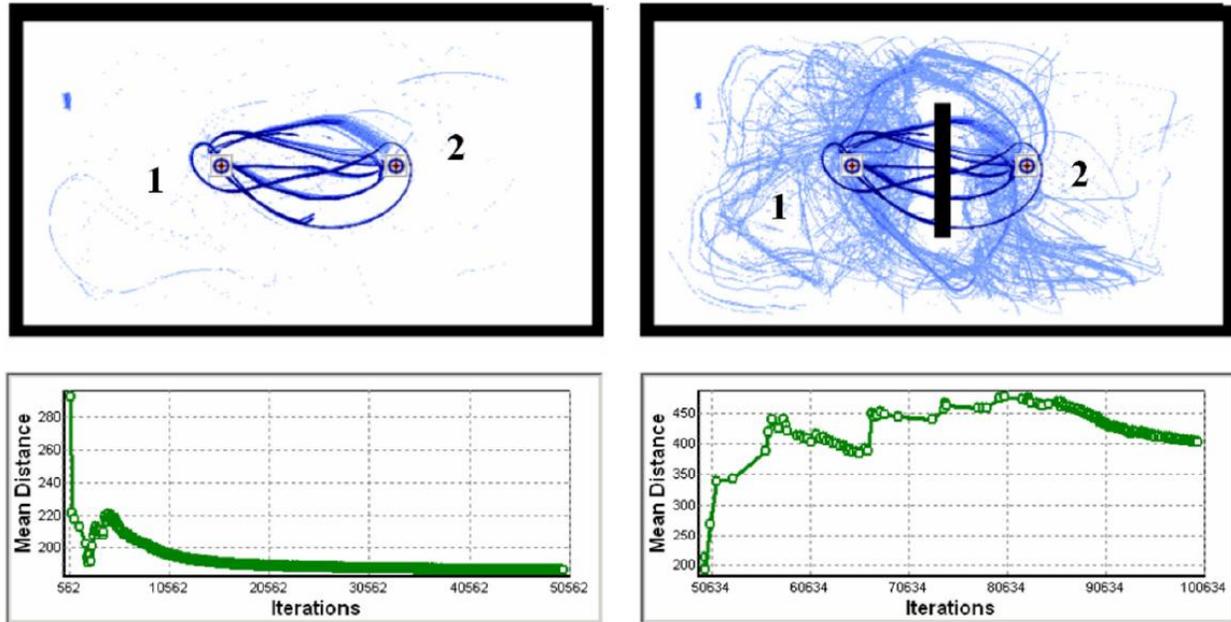


Figura 4.11: Simulação com a abordagem evolutiva antes (esquerda) e depois (direita) do novo obstáculo central ser inserido entre os dois alvos.

Olhando à esquerda da Figura 4.11, é evidente que houve uma rápida convergência para os caminhos mais curtos, terminando as primeiras 50 mil iterações com 990 trilhas intactas e distância média de 186,30. Quando o obstáculo central foi introduzido, é possível ver no gráfico à direita da figura que houve um período de readaptação (por volta de 30 mil iterações) até que os robôs recomeçassem a reduzir a distância média até 401,04. As trilhas ao redor do novo obstáculo confirmam que os robôs foram capazes de encontrar trajetórias alternativas para capturar os alvos.

Um experimento idêntico foi realizado, mas sem que os quatro robôs tivessem comunicação. A Figura 4.12 apresenta os gráficos de distância média ao longo das iterações. O gráfico à esquerda corresponde à primeira metade da simulação e, de novo, uma rápida convergência foi conseguida. Entretanto, uma diferença bem clara pode ser percebida no gráfico associado à segunda metade da simulação (quando o novo obstáculo foi introduzido). Após 5 mil iterações, os robôs já se readaptaram à mudança no ambiente e voltaram a apresentar bom desempenho. Isto é, a adaptação foi mais rápida quando não havia comunicação entre os robôs.

A Tabela 4.8 mostra os resultados de ambos os experimentos, divididos entre antes e depois da modificação no ambiente. Os resultados corroboram e detalham a forma como os robôs se adaptaram. Para a abordagem evolutiva, ocorreram 61 eventos de monotonia, duas colisões entre robôs e 75

colisões em obstáculos, até os robôs se familiarizarem com o novo ambiente. Por outro lado, na ausência de comunicação, os robôs se adaptaram mais facilmente: 52 eventos de monotonia e apenas 33 colisões em obstáculos após a entrada do obstáculo. Como resultado, esta abordagem foi capaz de completar o experimento com performance final, em termos de capturas e distância média, melhor do que a versão evolutiva.

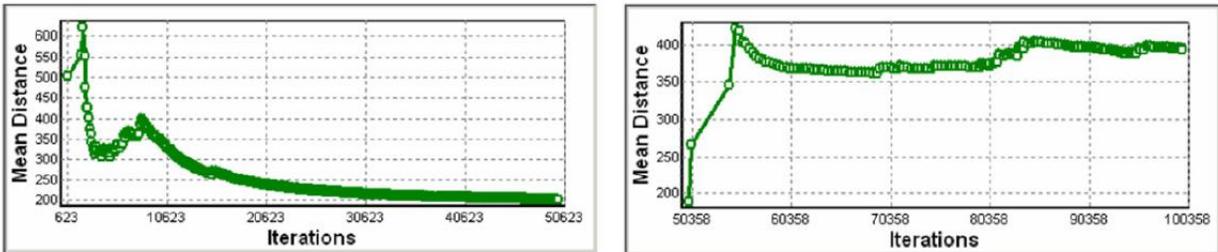


Figura 4.12: Gráficos da variação da distância média para a simulação com ausência de comunicação antes (esquerda) e depois (direita) do novo obstáculo ser inserido entre os alvos.

A melhor adaptabilidade dos robôs sem comunicação pode ser atribuída ao fato de que seus sistemas são mais reativos, dado que eles não usam nenhuma informação adicional como o mapa de trilhas de feromônio. Como o sistema com estratégia estigmérgica evolui baseado nas trilhas de feromônio, mudanças no ambiente, aliadas a uma baixa taxa de evaporação, implicam em uma adaptação mais lenta.

A conclusão mais relevante desta seção é que o SNA proposto com abordagem evolutiva é capaz de aprender e se adaptar a ambientes dinâmicos e/ou desconhecidos. Quando a dinâmica é dada por uma transição do tipo degrau (abrupta), a existência de uma memória associada com trilhas anteriormente de alta qualidade parece atrasar a descoberta de trilhas alternativas. Logo, quando a transição no ambiente acontece em uma taxa mais alta do que a de evaporação feromonal, a ausência de comunicação estigmérgica leva a uma reação mais efetiva e pronta.

Tabela 4.8: Resultados obtidos para os experimentos de adaptabilidade a mudanças no ambiente.

Versão	Evolutiva		Sem Comunicação	
	Antes	Depois	Antes	Depois
Inserção do Obstáculo				
Monotonia	4	61	4	52
Colisão em Robôs	73	2	15	0
Colisão em Obstáculos	18	75	22	33
Capturas	999	220	932	222
Trilhas Intactas	990	170	917	195
Distância Média	186,30	401,06	200,60	393,30

4.7.4. Otimização de Itinerários

O conjunto de experimentos apresentados anteriormente mostraram resultados convincentes em relação à habilidade do SNA com comunicação por estigmergia de minimizar trajetórias entre alvos. Para explorar mais profundamente esta capacidade de otimização, propõe-se empregar esta abordagem de robótica coletiva para resolver o problema do caixeiro viajante, um dos mais estudados no campo da otimização combinatória.

Dada uma coleção de cidades, a versão padrão do problema do caixeiro viajante consiste em encontrar o caminho mais curto para visitar todas as cidades uma única vez e retornar à cidade inicial. O sistema de navegação se aplica ao problema levando-se em conta que os alvos representam as cidades, e que os robôs devem capturá-los de forma a completar um circuito fechado, sendo que o comprimento total precisa ser minimizado.

Embora haja analogias entre os dois problemas, algumas modificações tiveram que ser feitas no SNA para que ele pudesse ser usado no problema do caixeiro viajante. Primeiramente, os robôs são capazes de detectar todos os alvos ao mesmo tempo, de modo que eles possam capturá-los na seqüência que quiserem. Quando um alvo é capturado, ele é desativado temporariamente (apenas para o robô que o capturou) até que o robô complete o circuito completo. Deste jeito, cada robô possui sua própria lista de alvos ativos, podendo capturar em qualquer ordem (não há competição por alvos). Uma vez que um robô tenha capturado todos os alvos, apenas o primeiro capturado é ativado, forçando o robô a completar o circuito. A seguir, todos os alvos são habilitados novamente, exceto o primeiro. Além disso, ao invés de medir apenas a distância média entre dois alvos capturados, o comprimento total do circuito é computado. A seqüência de alvos capturados também é armazenada (embora não seja usada pelo robô em nenhuma circunstância), dado que é um importante indicador do circuito ótimo.

Uma instância artificial do problema do caixeiro viajante foi usada neste experimento. Ela contém 12 cidades (alvos) dispostas arbitrariamente, conforme mostra a Figura 4.13. O circuito ótimo está em destaque, conectando todas as cidades com a menor distância percorrida. Esta instância foi adotada para sintetizar um ambiente para navegação de robôs por meio da conversão das cidades em alvos, assim como se vê na Figura 4.14. Quatro robôs controlados pelo SNA com abordagem evolutiva foram utilizados. A abordagem evolutiva foi escolhida por ter apresentado desempenho geral superior nos experimentos anteriores.

O melhor resultado obtido pelos robôs entre três simulações de 500 mil iterações cada é exibido na Figura 4.14. O circuito de cidades (alvos) que os robôs formaram por meio do depósito de feromônio coincide com a solução ótima, cuja seqüência é: 1-2-4-3-5-12-8-10-7-11-9-6-1. Em termos de comprimento ótimo, o resultado sugere uma boa solução, mas não a melhor, dado que o menor comprimento possível é 2340 e o comprimento obtido foi de 2673. Entretanto, analisando a natureza de cada solução, uma obtida apenas ligando diretamente as cidades na ordem correta, e a outra emergida pela navegação dos robôs em um ambiente dinâmico e desconhecido, a diferença de 14,23% no comprimento do circuito não é tão relevante. Mais do que isso, além de evitar colisões em obstáculos e robôs, as coordenadas dos alvos não são conhecidas; os robôs detectam todos os alvos simultaneamente e têm que decidir qual o próximo a capturar.

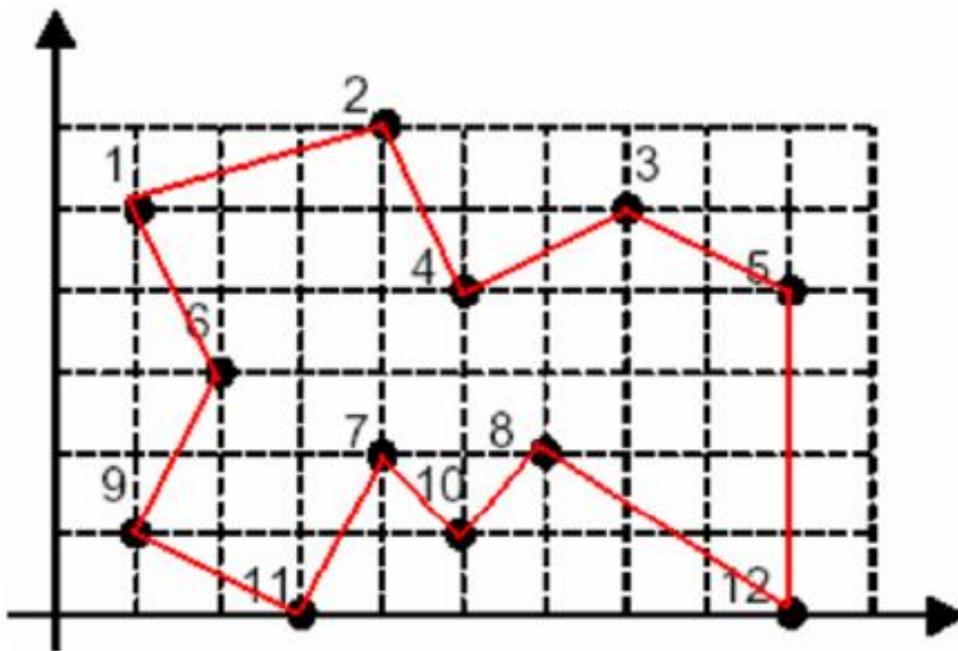


Figura 4.13: Instância artificial do problema do caixeiro viajante com 12 cidades.

É óbvio que existem muitos outros métodos para se lidar com problemas de caixeiro viajante. Inclusive, algumas técnicas bem-sucedidas se baseiam em estigmergia e formigas (Dorigo & Stützle, 2004). Também é claro que uma instância de 12 cidades não representa um desafio significativo para as abordagens consideradas estado da arte. É interessante notar, entretanto, que, no problema de navegação de robôs, não se sabe qual deve ser a melhor seqüência de alvos. Finalmente, o SNA com estigmergia foi capaz de cumprir não apenas o objetivo do problema de navegação, mas também foi capaz de resolver um problema de otimização combinatória, no caso, o caixeiro viajante.

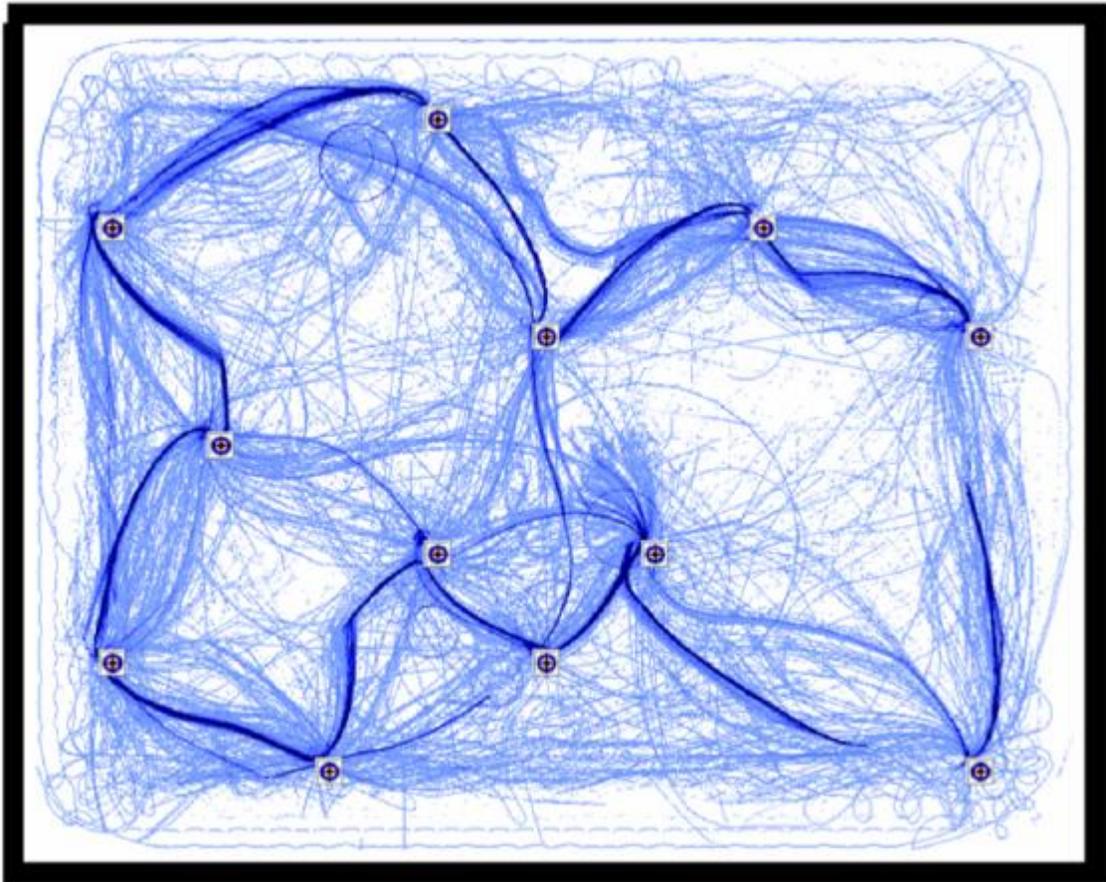


Figura 4.14: Circuito ótimo obtido para a instância com 12 cidades do problema do caixeiro viajante em um cenário de robótica coletiva, usando o SNA com a abordagem evolutiva. Ocorreram 64 colisões entre robôs, 32 colisões em obstáculos, 182 eventos de monotonia, e 5434 capturas.

Embora o itinerário ótimo tenha sido encontrado no experimento anterior, o SNA tem problemas quando lidando com instâncias que possuem quantidades maiores de cidades, nem sempre sendo capaz de descobrir a seqüência ótima. Um exemplo é a instância real conhecida como Bays29 (Reinelt, 1990), a qual é composta por 29 cidades. Ela foi utilizada na simulação cujos resultados são mostrados na Figura 4.15, sendo que os 8 robôs empregados podia escolher livremente quais alvos capturar e em que ordem. As soluções obtidas foram, em média, 40% acima do comprimento ótimo. O melhor circuito, mostrado à esquerda da figura, possui comprimento 12450 e, como se nota, houve uma grande diversidade de trajetórias realizadas pelos robôs sem que ocorresse uma convergência. Isto se deve ao grande número de alvos concentrados em regiões próximas, o que dificulta a definição de qual alvo capturar. Além disso, muitos robôs sendo atraídos para tais regiões acabam tendo que fazer desvios bruscos para evitar colisões com outros robôs, resultando em trajetórias irregulares. Em situações como esta, torna-se bastante difícil para esta abordagem alcançar resultados competitivos.

Em busca de verificar se os robôs são capazes de obter a extensão mínima desta instância, uma nova simulação foi realizada impondo aos robôs a seqüência ótima conhecida *a priori* para que eles capturassem os alvos. Novamente, 8 robôs foram usados e 10 simulações feitas. Em todas elas os robôs atingiram resultados não piores do que 1% do ótimo (9073). No melhor caso, apresentado à direita da Figura 4.15, a extensão total é igual a 9090.

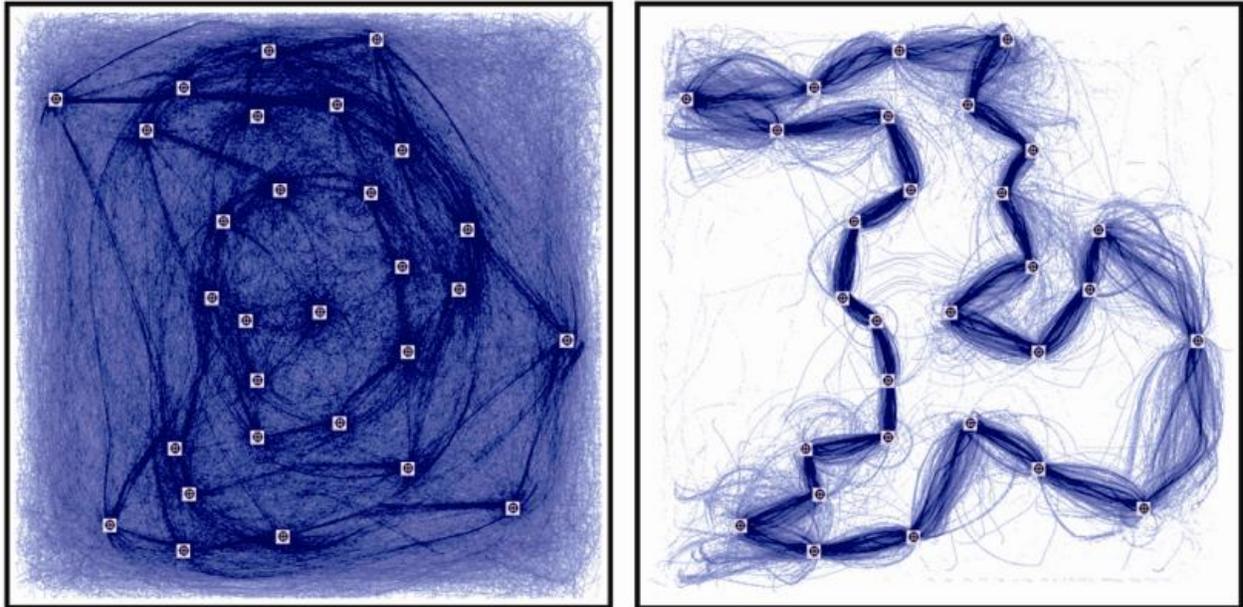


Figura 4.15: Simulação com 8 robôs controlados pelo SNA com comunicação evolutiva em uma instância real do problema do caixeiro viajante com 29 cidades. No resultado à esquerda, os robôs puderam escolher livremente a seqüência de alvos a capturar. Já à direita, a seqüência ótima de cidades conhecida foi imposta aos robôs.

4.8. Conclusões

Neste capítulo, foi proposta a extensão do sistema de navegação autônomo (SNA) já desenvolvido pelo autor para robótica coletiva, aplicado a problemas que envolvem ambientes complexos, dinâmicos e desconhecidos. Baseado nos sistemas classificadores com aprendizagem, o SNA compreende um controlador em contínua evolução, que não faz uso de conhecimento inicial, o qual é adquirido ao longo da operação do robô.

A expansão do SNA para situações com múltiplos robôs consiste, basicamente, no desenvolvimento de duas abordagens de comunicação entre os robôs, visando ganhos em termo de desempenho coletivo. Tais abordagens são inspiradas na estigmergia, uma forma indireta de comunicação usualmente executada por insetos sociais, em especial as formigas. Os robôs são capazes de detectar e

depositar feromônio artificial no ambiente, de modo a identificar regiões atrativas ou perigosas. Por meio de operadores evolutivos, os robôs aprendem durante a simulação como e quando liberar feromônios no ambiente: esta é a abordagem evolutiva. A outra abordagem, chamada determinística, faz uso de regras pré-programadas, introduzidas no sistema antes do início de sua operação, para detectar e liberar feromônios.

Com o objetivo de investigar e comparar o SNA com e sem as abordagens de comunicação em cenários coletivos, diversos experimentos foram realizados, envolvendo cerca de 130 simulações, em que o número de robôs navegando juntos variou de 4 a 20. Os resultados obtidos forneceram detalhes suficientes para se observar as vantagens e desvantagens de cada versão, as quais são discutidas a seguir.

Como já explorado na literatura, o uso de trilhas de feromônio por agentes, sejam eles vivos ou artificiais, provou levar a um desempenho geral consistentemente melhor do que aquele obtido quando não há comunicação. No caso específico deste trabalho, devido ao caráter contínuo do processo de aprendizagem, a abordagem evolutiva demonstrou ser mais flexível e, assim, mais eficiente do que a abordagem determinística, assistindo aos robôs na descoberta dos caminhos mais curtos entre alvos.

Detectou-se, no entanto, que altas taxas de evaporação causam degradação de performance. Portanto, é importante ajustar adequadamente tais taxas, dado que uma escolha inadequada pode tornar inútil ou prejudicial a comunicação por feromônios. Outro fator que influencia a abordagem evolutiva, afetando a forma como os feromônios são usados, é a posição inicial dos robôs no ambiente. Este fator é, entretanto, menos crítico do que a taxa de evaporação, tendo efeitos pouco relevantes no desempenho do sistema.

O número de regras do SNA, que representa sua complexidade, também foi analisado nos experimentos. Concluiu-se que a redução do número de regras causa uma degradação generalizada nos resultados quando há comunicação entre os robôs. A ausência de comunicação, neste caso, fez com que resultados melhores fossem obtidos quando os robôs eram controlados por sistemas relativamente mais simples. Em outro aspecto, a adaptabilidade, os robôs sem comunicação também se mostraram mais competentes. Eles foram capazes de se adaptar mais rapidamente a mudanças abruptas no ambiente do que os sistemas estigmérgicos. Já quando um número maior de robôs foi colocado para navegar simultaneamente, a falta de comunicação foi extremamente prejudicial, promovendo um aumento da frequência de colisões entre robôs.

Considerando a presença de comunicação por feromônios, também houve aumento no número de colisões quando uma grande quantidade de robôs operava em conjunto, o que era de se esperar. Entretanto, a probabilidade de encontrar trajetórias mais curtas entre alvos também foi maior, uma vez que, com mais robôs, mais espaço é explorado e maior a chance de bons caminhos serem localizados. A descoberta dos trechos mais curtos para se capturar alvos não foi o único tipo de otimização expressado pelo sistema de navegação. O SNA mostrou que é capaz de encontrar circuitos ou seqüências de alvos de alta qualidade em certas condições; os alvos, neste caso, representavam cidades no problema clássico do caixeiro viajante. Este resultado mostrou que o sistema proposto, além de conseguir resolver problemas de otimização, também é capaz de atender aos objetivos de navegação de forma eficiente.

Em resumo, pode-se dizer que o SNA, desenvolvido com um poderoso mecanismo de aprendizagem e uma abordagem evolutiva para comunicação por feromônios artificiais, mostrou ser eficiente no desvio de obstáculos, captura de alvos e otimização de trajetórias. Em especial, a emergência de comportamentos e soluções complexas sem que estas tenham sido pré-programadas pode ser avaliada como a maior virtude desta proposta.

Embora possa ser considerada apenas uma consequência indireta, a habilidade emergente do sistema de minimizar distâncias é promissora, especialmente porque o SNA não foi originalmente pensado como uma ferramenta de otimização. Por isso, as perspectivas futuras são voltadas a preparar o SNA para aplicações em problemas mais complexos de otimização e clusterização. Além disso, tarefas coletivas, como transporte de objetos e cenários co-evolutivos, também são vislumbrados.

4.9. Descrição dos Vídeos

Vídeo 4.1: Simulação em que os robôs devem capturar dois alvos alternadamente, buscando encontrar o caminho mais curto entre eles por meio do uso de feromônios artificiais. O sistema com modo de comunicação evolutivo é adotado. O gráfico na parte alta da imagem indica como a distância média realizada pelos robôs entre os dois alvos é reduzida ao longo da simulação.

Vídeo 4.2: Diversos robôs devem capturar seis alvos seqüencial e continuamente nesta simulação. Pode-se ver no vídeo que, por meio da utilização dos feromônios artificiais, os robôs são capazes de encontrar os caminhos mais curtos entre os alvos, o que faz com que a distância média entre as capturas decaia conforme a concentração das trilhas aumenta (mostrado no gráfico).

Capítulo 5: Rede Imuno-classificadora com Aprendizagem para Navegação Autônoma de Robôs

5.1. Introdução

Em geral, robôs autônomos devem executar uma ou mais tarefas, as quais podem estar associadas a múltiplos objetivos com variados graus de dificuldade. Por definição, um problema multi-objetivo possui ao menos duas metas, possivelmente conflitantes, para serem atingidas simultaneamente (Deb, 2001). Da mesma forma que nas frentes anteriores, esta também lida com tarefas de captura de alvos e desvio de obstáculos. Claramente estas tarefas possuem objetivos conflitantes, os quais foram coordenados com sucesso nas abordagens anteriores.

O grau de complexidade da navegação multi-objetivo de robôs pode elevar-se bastante se situações de mínimos locais tomarem parte do problema. Tais cenários exigem do sistema de controle algo contraditório: ignorar momentaneamente um objetivo que parece óbvio em favor de outro aparentemente desnecessário. Isto é, em mínimos locais (e.g., becos, labirintos), há muitas vezes a necessidade de o robô navegar de forma a se afastar do alvo temporariamente para contornar a armadilha e só então conseguir efetuar a captura.

Sistemas puramente reativos, que tomam decisões baseadas apenas nos estímulos sensoriais imediatos, têm extrema dificuldade e, geralmente, não são capazes de lidar adequadamente com estes tipos de situações (Ram, Arkin, K, & Clark, 1997). Para superar esta limitação, torna-se premente a adição de um aparato cognitivo mais avançado.

Um mínimo local pode ser detectado ao se observar o comportamento do robô controlado por um sistema reativo: ele entra num processo de repetição infinita da mesma trajetória, em geral, em formato circular. Isto ocorre freqüentemente se o ambiente contém obstáculos com formato côncavo (como becos) e labirintos entre o robô e o alvo. Para escapar dessas situações, o robô precisa ser capaz de perceber a repetição seqüencial das mesmas cenas do ambiente ou ações de controle, o que requer algum tipo de memória de experiências passadas (Krishna & Kalra, 2000).

Sistemas cognitivamente mais avançados podem ser implementados com base no paradigma conexionista (Fodor & Pylyshyn, 1988). É sabido que as conexões entre os elementos de uma rede são capazes de capturar e preservar representações complexas e valiosas de conhecimento [(Smolensky,

1987) e (Farmer, 1990)]. Isto é, a existência de interações internas (conexões) entre os componentes de um sistema abre-o para a possibilidade de agir, não apenas baseado em informações instantâneas, mas também em eventos passados (Bates & Elman, 2002).

Vislumbrando navegação autônoma em um cenário conexcionista, pode-se imaginar que o conhecimento do contexto em que o robô está inserido reforça a habilidade do robô em escapar de mínimos locais e estabelecer correlação entre situações similares. O seguinte exemplo ilustra os argumentos anteriores. Suponha que o robô está dentro de um beco (mínimo local) e precisa sair dele para capturar um alvo. Nesta situação, e considerando a completa ausência de conhecimento prévio, o robô tentará, casualmente, executar diversas estratégias locais para escapar. Estas estratégias locais são normalmente compostas por uma seqüência de ações. Tem-se que tentar diversas seqüências, e aquela que for bem-sucedida e conduzir o robô para fora do beco é uma candidata a ser memorizada. Em um sistema conexcionista, uma seqüência ordenada de ações de controle – por exemplo, gire para trás, vá para frente, vire à esquerda e então vá em frente – podem se implicitamente gravadas como padrões de conexões, de forma que a execução da primeira ação daquela seqüência estimula a segunda ação, que por sua vez leva à terceira, e assim por diante, em um processo de encadeamento. Como resultado, o robô tenderia a tomar a mesma seqüência de ações bem-sucedida em situações futuras semelhantes. Esta breve descrição, embora bem específica, revela a essência da abordagem proposta aqui.

Partindo do controlador reativo apresentado no Capítulo 3, investigado em robótica individual e coletiva (Capítulo 4), deseja-se sofisticá-lo com a introdução de habilidades cognitivas que o tornem capaz de lidar com mínimos locais, visando suplantá-los. É certo que o sistema do Capítulo 3 obteve resultados positivos em um extensivo conjunto de experimentos reativos. Quer-se agora levá-lo para este outro âmbito de problemas.

Já foi mencionado repetidas vezes que o controlador do Capítulo 3 é um sistema de navegação autônomo fundamentado em uma versão melhorada dos sistemas classificadores com aprendizagem (SCA), teoria proposta por Holland (1986). O SCA é uma abordagem evolutiva que sintetiza mecanismos adaptativos de inferência capazes de operar em condições variantes no tempo. O conhecimento é representado por uma população de regras de inferência, chamadas de classificadores. Muito embora o SCA seja eficiente na produção e atualização de conhecimento, seu mecanismo de tomada de decisão não enfatiza interações cooperativas e diretas das regras. Mais precisamente, até existem interações,

mas elas são esparsas e acontecem como consequência indireta de processos de competição (Farmer, 1990).

O sistema imunológico biológico, por sua vez, é composto de numerosas células e moléculas, funcionando com base em uma intensa dinâmica de interações que acabam por manter o organismo. Como resultado desta dinâmica, sabe-se que o sistema imunológico apresenta significativas habilidades cognitivas, como reconhecimento de padrões, adaptação a cenários desconhecidos e memória associativa [(Hershberg & Efroni, 2001)(de Castro L. N., 2003)]. A teoria da rede imunológica [(Jerne, 1974)(Farmer, Packard, & Perelson, 1986)], que é um dos ramos dos sistemas imunológicos artificiais (SIA) (de Castro & Timmis, 2002), incorpora tal dinâmica por meio de um modelo computacional. Farmer (1990) afirma que a rede imunológica pode ser capaz de realizar tarefas que seriam impossíveis para células individuais, dado que as interações da rede permitem, potencialmente, células diferentes e de tipos distintos se comunicarem e, assim, coletivamente, reforçarem ou suprimirem as respostas imunológicas umas das outras.

Visando converter o sistema de navegação do Capítulo 3 em uma versão mais elaborada cognitivamente, que seja apta a lidar com problemas de mínimo local, conceitos da teoria da rede imunológica serão aplicados para criar uma rede de regras (classificadores), resultando em um sistema de navegação autônomo conexionista, chamado Rede Imuno-Classificadora com Aprendizado, abreviado para ILCN (do inglês *Immune Learning Classifier Network*). Neste sistema híbrido, os classificadores representarão os nós da rede e as conexões serão as ligações entre eles. A natureza evolutiva desta proposta merece destaque, uma vez que os comportamentos elementares (classificadores) e também a dinâmica que os coordena (rede imunológica) serão evoluídos ao longo da navegação do robô.

5.2. Contextualização do Capítulo

Até aqui, apenas sistemas reativos foram estudados. Neste capítulo, um sistema mais complexo é proposto, o qual combina reatividade com capacidades cognitivas mais avançadas provenientes dos sistemas imunológicos artificiais. Conforme a classificação discutida na Seção 2.2, pode-se caracterizar o sistema ILCN como híbrido. As principais propriedades do SNA reativo, que são a rápida capacidade de atuação e seu mecanismo de aprendizagem, são herdadas pelo ILCN. O aspecto deliberativo vem da

memória imuno-inspirada, que consiste em um modelo interno de mundo construído por meio da dinâmica do sistema.

Na prática, a memória (ou modelo de mundo) é representada pelas conexões da rede imunológica. Tal tipo de memória pode ser encaixado na categorização de Arkin (1998) como de longo prazo, já que conhecimentos obtidos da interação com o ambiente são empregado para a construção do modelo de mundo. Além disso, a rede tem um papel de adaptação em relação ao componente reativo, visto que ela ajustará o peso das regras do sistema, estimulando-as ou suprimindo-as de acordo com as condições do ambiente.

Pelo fato de o ILCN contar com o mesmo mecanismo de aprendizagem do SNA do Capítulo 3, ele também acaba por conter aprendizagem por meios evolutivos. Este mecanismo é responsável pela evolução das regras do sistema. Por outro lado, as conexões entre as regras (responsáveis pela definição da topologia da rede) são obtidas por outro processo evolutivo completamente independente. Conseqüentemente, a rede imuno-classificadora (ILCN) proposta aqui realiza evolução com aprendizagem, sendo que a aprendizagem se dá evolutivamente. Ou seja, ambos os modos de aprendizado citados na Seção 2.5 estão presentes e integrados nesta abordagem, situação única entre todas as frentes de pesquisa deste trabalho.

É difícil distinguir qual componente do sistema é responsável por o quê exatamente. Tanto as regras quanto as conexões determinam o comportamento do robô. Entretanto, pode-se dizer que as conexões atuam mais no papel de coordenação, chaveando entre comportamentos elementares. Portanto, a memória neste sistema não é um conjunto de informações armazenado em um dispositivo separado, acessado quando necessário, mas sim conhecimento implícito incorporado na arquitetura da rede, a qual atua ativamente na dinâmica de tomada de decisões do controlador.

Mais uma vez, nesta frente, a aplicação da solução proposta é genérica, podendo ser utilizada em qualquer problema de navegação de robôs que possa se configurar como desvio de obstáculos e captura de alvos. Contudo, existe um enfoque especial na resolução de problemas com mínimos locais, os quais podem ser considerados como estando em uma categoria de maior grau de dificuldade do que os problemas até então investigados.

Pelo fato de a dinâmica do ILCN, inspirada em princípios imunológicos, permitir a eliminação natural de elementos (regras) do sistema, haverá, sob a perspectiva da otimização, uma busca não só por

produzir comportamentos eficientes, mas também por minimizar a complexidade do sistema. Ao longo do capítulo, este aspecto será explorado mais contundentemente.

Igual às frentes anteriores, esta adota o mini-robô Khepera II. Ambos os tipos de experimentação são empregados: simulação computacional e real. Em especial, a simulação é utilizada para evolução e maturação do sistema, enquanto os testes reais servem para validação final do controlador.

5.3. Trabalhos Correlatos

A evolução de arquiteturas para sistemas cognitivos tem demonstrado ser eficaz na literatura (Parisi & Calabretta, 2001), com destaque para as redes bayesianas (Kim, Yoo, & Cho, 2005) e redes neurais artificiais (Nolfi, Floreano, Miglino, & Mondada, 1994). Entretanto, como característica que o difere dessas abordagens, o sistema proposto aqui é capaz de se adaptar em resposta a mudanças do ambiente, mesmo após ter encontrado uma arquitetura de rede bem-sucedida. Isto se deve ao fato de que a aprendizagem do controlador, ou seja, a evolução dos classificadores é contínua e cumulativa.

Em relação a abordagens não-conexionistas, existem algumas propostas na literatura que desenvolvem métodos específicos para detectar e superar mínimos locais em navegação de robôs. A maioria delas, entretanto, são do tipo *top-down*, i.e., os componentes básicos e até mesmo a estrutura inteira do sistema são concebidos pelo projetista (e.g. Lumelsky (1991), Krishna & Kalra (2000), Xu & Tso (1999)).

Também com base em sistemas classificadores com aprendizagem, Vasilyev (2002) propôs uma versão conexionista em que os classificadores são ligados de modo similar ao das redes neurais. Geralmente, em tais SCAs conexionistas, os classificadores assumem o papel de neurônios organizados em arquiteturas de múltiplas camadas, sendo os parâmetros do sistema ajustados principalmente por técnicas de aprendizagem por reforço (e.g. *bucket brigade*). Uma vantagem é a redução do número de regras, quando comparado a um SCA que não segue uma proposta conexionista.

Particularmente, no contexto dos sistemas imunológicos artificiais, Ishiguro et al. (1996) propuseram um controlador de robôs baseado na teoria da rede imunológica para tarefas de coleta de lixo. Entretanto, os anticorpos da rede, que representam os comportamentos básicos de navegação, e as conexões são incorporados pelo projetista, restringindo a flexibilidade do sistema. Visando obter as conexões da rede automaticamente, algoritmos evolutivos foram empregados em Watanabe et al.

(1998) e Michelan & Von Zuben (2002), aumentando os graus de liberdade dos controladores. Muito embora tais propostas tenham sido bem-sucedidas em tarefas de coleta de lixo, os comportamentos básicos (anticorpos) são ainda totalmente determinados pelos projetistas.

SCA e SIA têm sido comparados por alguns autores na literatura, havendo sido apontadas semelhanças e perspectivas de fusões teóricas [(Farmer, Packard, & Perelson, 1986), (Farmer, 1990), (Vargas, de Castro, & Von Zuben, 2002)]. Gaspar & Hirsbrunner (2002) foram os primeiros a realmente hibridizar as duas teorias. A proposta deles, chamada PICS, combinou seleção clonal e conceitos de memória imunológica com sistemas classificadores, em que cada indivíduo da população engloba um conjunto completo de regras (i.e. versão conhecida como Pittsburgh). Embora o PICS não tenha sido aplicado à robótica, os resultados de simulação indicaram que o sistema é capaz de memorizar boas estratégias de operação.

Explorando as características que se complementam nas duas teorias, Vargas et al. (2003) propuseram a CLARINET (do inglês *Classifier Immune Network*), um híbrido de sistema classificador tradicional com a rede imunológica. A CLARINET usa a dinâmica da rede imunológica como parte do processo de tomada de decisão, colocando a energia de cada classificador em função do nível de concentração dos anticorpos, o qual é definido pela dinâmica da rede. A evolução do sistema é parcialmente restrita, dado que os classificadores são divididos em grupos, e classificadores pertencentes a grupos diferentes não podem ser combinados. Além disso, a quantidade de classificadores que compõem a CLARINET é fixa. Embora a CLARINET tenha algumas limitações impostas pelas características mencionadas, ela mostrou resultados promissores em um conjunto de experimentos reativos.

O sistema aqui proposto, ILCN, concorda com a CLARINET no embasamento teórico, mas difere em aspectos essenciais. Primeiro, os classificadores são completamente evoluídos *on-line*, ou seja, enquanto o robô navega, e sem qualquer tipo de restrição ou critério de parada para o aprendizado. As conexões entre classificadores também são evoluídas *online*. Portanto, existem dois processos evolutivos, distintos e independentes, que ocorrem simultaneamente em busca de atingir os objetivos da navegação. Em termos de tomada de decisão, a dinâmica da rede é a única e exclusiva responsável. Outra inovação é o uso da meta-dinâmica da rede imunológica visando eliminar regras inúteis do sistema, produzindo redes mais parcimoniosas.

5.4. Aspectos Básicos da Teoria da Rede Imunológica

O sistema imunológico biológico dos animais vertebrados, especialmente os mamíferos, consiste de um sistema distribuído que opera por meio de mecanismos intrincados envolvendo um grande número de células e moléculas no organismo. O sistema imunológico tem o objetivo principal de proteger o corpo de qualquer invasor prejudicial e manter a integridade do organismo. Para tanto, ele apresenta habilidades cognitivas importantes, tais como aprendizagem, memória e adaptação (de Castro L. N., 2003).

Tentando investigar e entender os sistemas imunológicos biológicos, diversas teorias imunológicas foram formuladas. Baseando-se nelas, modelos computacionais têm sido propostos na literatura visando reproduzir as propriedades imunológicas que podem ser muito úteis na resolução de problemas computacionais complicados. Estes modelos são conhecidos como sistemas imunológicos artificiais (SIA)(de Castro & Timmis, 2002).

A teoria da rede imunológica é uma das abordagens de SIA. Basicamente, ela sugere que o sistema imune é composto por um conjunto de células e moléculas que interage dinamicamente através de suas propriedades físico-químicas. A dinâmica da rede define os mecanismos regulatórios que ajustam a concentração de cada componente do sistema. Há dois tipos de componentes neste modelo: anticorpos e antígenos. Anticorpos são responsáveis por reconhecer antígenos (patógenos) de acordo com a afinidade entre eles. Além disso, a teoria da rede propõe que os anticorpos também são capazes de reconhecer outros anticorpos. Reconhecer um anticorpo ou ser reconhecido é o que representa as interações da rede imunológica. Logo, o reconhecimento de um anticorpo por outro causa o estímulo do primeiro e a supressão do segundo, implicando em modificações na concentração deles (Farmer, Packard, & Perelson, 1986).

Como ilustrado na Figura 5.1, os anticorpos são os nós da rede, e as interações estimulantes ou inibidoras são representadas pelas setas. O processo de ajuste da concentração dos anticorpos, denominado dinâmica da rede, é governado por um sistema de equações diferenciais ordinárias acopladas que serão apresentadas nas seções a seguir.

Outra importante propriedade dos sistemas imunológicos é a meta-dinâmica, isto é, o sistema imunológico está constantemente criando novas células e moléculas, e eliminando aquelas que são muito antigas ou não estão sendo muito usadas. Esta contínua produção, recrutamento e morte de

células e moléculas imunológicas têm papel essencial no funcionamento do sistema imunológico (Varela, Coutinho, Dupire, & Paz, 1988). No modelo da rede imunológica adotado aqui, a meta-dinâmica atua quando da morte e automática eliminação dos anticorpos cuja concentração decai a zero.

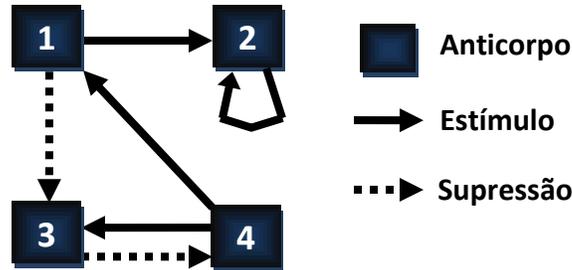


Figura 5.1: Exemplo de uma rede imunológica simples em que o anticorpo 1 estimula o anticorpo 2 e suprime o anticorpo 3; o anticorpo 2 se auto estimula; o anticorpo 3 é suprimido pelo anticorpo 1 e suprime o anticorpo 4; e o anticorpo 4 estimula os anticorpos 1 e 3.

5.5. Rede Imuno-classificadora com Aprendizagem como um Sistema de Navegação Autônomo

O sistema de navegação autônomo proposto aqui, denominado rede imuno-classificadora com aprendizagem (ILCN), é um híbrido de uma versão melhorada do sistema classificador com aprendizado (SCA) já descrito no Capítulo 3 e do modelo computacional da teoria da rede imunológica. Isto é, conceitos da teoria da rede imunológica foram trazidos para o SCA para criar uma rede de classificadores, ou rede de regras de inferência, caracterizando esta proposta como uma abordagem connexionista.

As conexões entre as regras, fundamento herdado da teoria da rede imunológica, são essenciais para o controlador connexionista. Elas introduzem uma enorme flexibilidade no sistema de navegação, por meio de algo que não existia no SCA do Capítulo 3: interações diretas entre classificadores. A rede, com suas múltiplas conexões, é capaz de representar e guardar conhecimentos valiosos adquiridos no curso de operação do sistema, assim como se supõe ocorra na memória imunológica biológica (de Castro & Timmis, 2002). A rede imunológica fundamenta a dinâmica do mecanismo de tomada de decisão, ao mesmo tempo em que o SCA toma conta do processo de produção e atualização da base de conhecimento.

É importante deixar clara a correspondência entre os componentes do sistema imunológico e o sistema de navegação connexionista proposto. Assim como mostra a Tabela 5.1, os estímulos sensoriais

detectados pelos sensores do robô são representados pelos antígenos no sistema imunológico. Cada anticorpo, por sua vez, se refere a um classificador da ILCN. Logo, a rede de anticorpos pode ser vista como uma rede de classificadores que reage em resposta à presença de estímulos sensoriais (antígenos). Além disso, uma conexão entre dois classificadores tem o mesmo papel da afinidade entre anticorpos nos sistemas imunológicos, da mesma forma que o grau de casamento entre os classificadores e os estímulos está associado à afinidade entre anticorpos e antígenos.

Tabela 5.1: Analogias entre o sistema imunológico biológico e o sistema híbrido proposto, ILCN.

Sistema Imunológico	Rede Imuno-classificadora com Aprendizagem
Organismo	Robô
Antígenos	Estímulos capturados pelos sensores do robô
Anticorpos	Classificador
Afinidade anticorpo-anticorpo	Conexão entre classificadores
Afinidade anticorpo-antígeno	Grau de similaridade com os estímulos

O sistema de navegação autônomo interage com o ambiente por meio dos sensores e atuadores do robô, sendo os últimos responsáveis por executarem as ações de controle a cada passo da navegação (iteração). Assim como no SNA original (Capítulo 3), este sistema pode ser dividido em duas partes: processos evolutivos e sistema de controle. A Figura 5.2 mostra a organização dos componentes do sistema de controle.

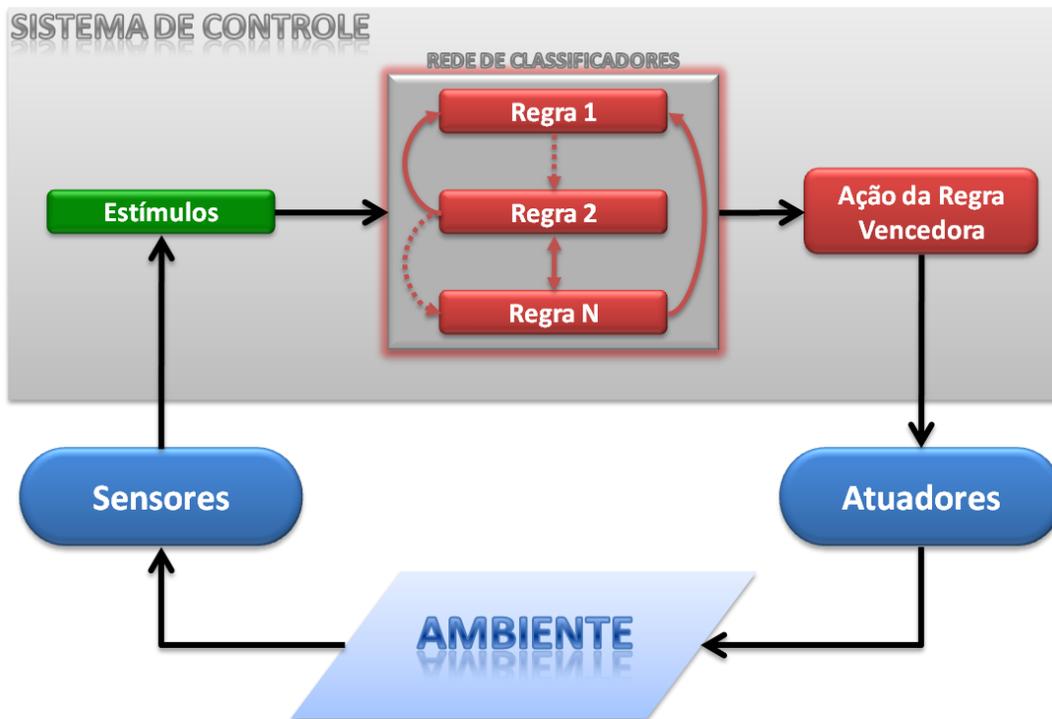


Figura 5.2: Componentes do sistema de controle da rede imuno-classificadora com aprendizagem.

A rede de classificadores (composta por regras e conexões entre elas) representa a base de conhecimento do sistema. Dado que a rede possui sua própria dinâmica interna, pode-se dizer que ela incorpora o mecanismo de tomada de decisão, diferentemente do SNA original. Logo, a rede recebe estímulos capturados pelos sensores de alvo e obstáculos e, de acordo com sua dinâmica interna, determina que classificador agirá no ambiente por meio de sua ação de controle. Este procedimento se repete toda vez que uma ação de controle é executada, constituindo um ciclo que somente é interrompido para dar lugar aos processos evolutivos, responsáveis por atualizar a população de classificadores e também as conexões da rede.

5.5.1. Classificadores e Conexões

Os indivíduos da população são representados por regras condição – ação, com mecanismo de inferência do tipo *modus-ponens*. O cromossomo de cada classificador, exemplificado na Figura 5.3, possui três partes: antecedente, conseqüente e conexões.

A parte antecedente é composta por dois vetores: distância a obstáculos (RO: vetor de inteiros) e intensidade luminosa do alvo (RA: vetor de inteiros). A parte conseqüente também é constituída de dois vetores: ajuste de direção (RD: vetor binário) e ajuste de velocidade (RV: vetor binário). A parte das conexões, por sua vez, é representada por um único vetor de inteiros: RC. Dado um classificador i , seu vetor de conexões (RC) contém os índices de todos os classificadores com os quais i está conectado. Conseqüentemente, o comprimento do vetor de conexões é variável.

Portanto, RO e RA são os antecedentes do classificador e contêm, respectivamente, seis e oito elementos. O primeiro corresponde ao número de sensores de proximidade, e o outro ao número de sensores de luminosidade do robô. Na parte conseqüente, o vetor RD possui nove elementos que codificam o ajuste de direção do robô. Os quatro bits menos significativos do vetor são convertidos em um valor absoluto, enquanto os cinco bits mais significativos são convertidos ao sinal do ajuste de direção (positivo ou negativo) por voto majoritário. No caso do vetor RV, se a maioria dos bits for zero, a velocidade deve ser reduzida de um valor constante; caso contrário, a velocidade é aumentada de um valor constante.

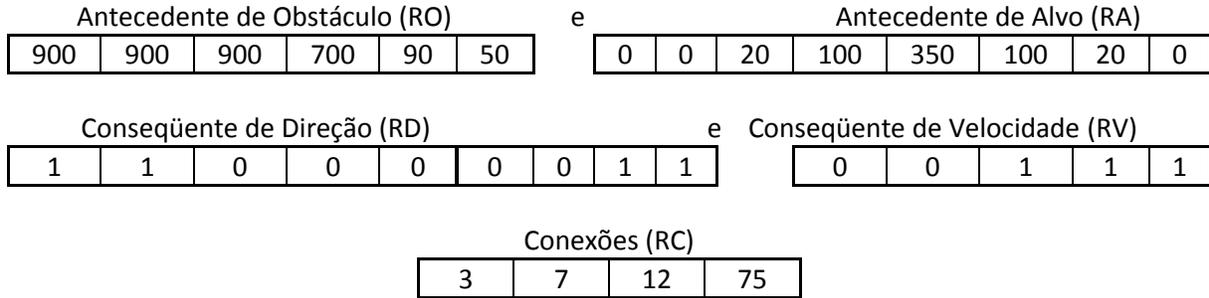


Figura 5.3: O cromossomo de uma regra em que as partes conseqüentes RD e RV determinam um ajuste de direção de 3° e aumento da velocidade, respectivamente. RC indica que este indivíduo está conectado aos classificadores 3, 7, 12 e 75.

Todos os vetores dos classificadores são inicialmente construídos aleatoriamente. No caso do vetor de conexões, a quantidade de conexões, e elas próprias, são escolhidas aleatoriamente. A quantidade de conexões pode variar de 1 a 10.

Há uma propriedade adicional associada a cada classificador: sua concentração. A concentração do classificador é usada pelo mecanismo de decisão como fator fundamental na definição do vencedor a cada ciclo sensorio-motor. As concentrações variam dinamicamente de acordo com a Equação 5.1, que corresponde originalmente à dinâmica da rede imunológica. Caso a concentração de um classificador chegue a zero, ele é automaticamente removido da população. Esta é uma das ações resultantes do fenômeno conhecido como meta-dinâmica imunológica, a qual faz com que o sistema se torne mais compacto.

5.5.2. Dinâmica da Rede para Tomada de Decisão

Este mecanismo é responsável por determinar o classificador que irá navegar o robô a cada iteração. Basicamente, o vencedor é o classificador com maior concentração a cada instante. Como já dito, a concentração a_i de cada classificador i é calculada por um sistema de equações diferenciais ordinárias, representadas pela Equação 5.1, que corresponde originalmente à dinâmica da rede imunológica. Como pode ser visto na Equação 5.1, existem dois fatores que têm maior influência na dinâmica da rede: a afinidade entre classificador e estímulo sensorial, e as conexões entre classificadores.

$$\frac{da_i}{dt} = \left(\alpha \sum_{j=1}^N m_{ij} a_j(t) - \beta \sum_{k=1}^N m_{ik} a_k(t) + \gamma M_i - h \right) a_i(t) \quad \text{Equação 5.1}$$

onde:

- N é o número de classificadores (anticorpos) da rede;

- M_i é o valor de casamento entre um classificador i e o estímulo sensorial atual (antígeno) dado pela Equação 5.2;
- m_{ij} representa o efeito da estimulação do classificador j pelo classificador i . Isto é, se o classificador j está conectado ao classificador i , então $m_{ij} = 1$, caso contrário, $m_{ij} = 0$;
- m_{ik} representa o efeito supressivo do classificador k no classificador i . Isto é, se o classificador i está conectado ao classificador k , então $m_{ik} = 1$, caso contrário, $m_{ik} = 0$;
- h é a taxa de morte natural dos classificadores;
- os coeficientes α , β e γ determinam a relevância de cada termo da equação;
- o método de discretização de Euler, com passo 1, é utilizado na implementação desta equação.

A computação de M_i , que é a afinidade entre classificador e estímulos, é feita através da codificação da distância aos obstáculos e intensidade do alvo, em dois vetores, EO e EA, respectivamente. EO e EA representam o antígeno, sendo que a sua similaridade M_i com cada classificador i é dada pela Equação 5.2:

$$M_i = \frac{\|RO_i - EO\|}{\max O} + \frac{\|RA_i - EA\|}{\max A} \quad \text{Equação 5.2:}$$

onde $\max O$ e $\max A$ são usados para normalização dos respectivos termos e $\| \cdot \|$ é a norma euclidiana.

A propósito, no mecanismo de decisão do SCA descrito no Capítulo 3, o classificador vencedor era aquele cujo casamento com o estímulo sensorial fosse o melhor dentre todos na população. No ILCN, o vencedor é aquele que possui a maior concentração no dado momento. Note, portanto, que a similaridade entre estímulo e classificador é apenas parte do processo de tomada de decisão no novo ILCN.

O Algoritmo 5.1 resume os principais passos do mecanismo de tomada de decisão que é executado a cada iteração e corresponde a um ciclo sensório-motor. Observe que o processo evolutivo não está incluído neste algoritmo, sendo explicado adiante.

Algoritmo 5.1: Pseudocódigo do mecanismo de tomada de decisão que navega o robô (ciclo sensório-motor).

```

Obter estímulos dos sensores do robô;
Atualizar a concentração de todos os classificadores baseados na Equação 5.1;
Eliminar todos os classificadores cuja concentração é nula;
Definir o classificador de maior concentração como o vencedor;
Aplicar a ação do classificador vencedor nos atuadores do robô.
    
```

5.5.3. Processos Evolutivos

No ILCN, dois processos evolutivos ocorrem de forma concorrente: uma micro-evolução, que atua nos classificadores, e uma macro-evolução que evolui a rede (conjunto de conexões) baseado no comportamento global do robô. Embora ambas as evoluções aconteçam em diferentes escalas de tempo, podendo explorar regiões distintas do espaço de busca, elas acabam por cooperar de uma maneira poderosa para evoluir totalmente os comportamentos básicos (classificadores) e também seu mecanismo de coordenação (rede).

É importante sublinhar que a evolução das redes é completamente independente da evolução dos classificadores, como deixa claro a Tabela 5.2. Note que cada processo evolutivo tem seu modo particular de funcionar e é operado em ocasiões não correlatas. Isto é, a evolução das redes é executada com frequência constante (número pré-definido de iterações), enquanto a evolução dos classificadores é disparada sempre que um dos três eventos a seguir ocorre: colisão em obstáculo, captura de alvo ou monotonia. O Algoritmo 5.2 sintetiza o processo evolutivo inteiro, indicando seus principais passos. Os detalhes completos de ambas as evoluções serão descritos a seguir.

Tabela 5.2: Propriedades dos processos evolutivos do sistema proposto.

	Evolução da Rede	Evolução dos Classificadores
Indivíduos	Redes	Classificadores
Fitness	Equação 5	Equações 3 e 4
Época	Número fixo de iterações	Colisão, captura e monotonia
Seleção	Melhor indivíduo	Roleta
Operadores genéticos	Mutação (2%)	Cruzamento (100%) e mutação (1%)
Critério de parada	Fitness mínimo	Não há

Algoritmo 5.2: Pseudocódigo do processo evolutivo completo.

```

Inicializa aleatoriamente os classificadores e conexões para todos os robôs;
enquanto melhor_fitness_obtido < fitness_desejado faça
  para i=1 até número_de_iterações faça
    para n=1 até número_de_robôs faça
      Navegar robôn (veja Algoritmo 5.1)
      se robôn colidiu em obstáculo então
        Evoluir classificadores do robôn por colisão (Seção 3.3.3.1);
      fim do se
      se robôn capturou alvo então
        Evoluir classificadores do robôn por captura (Seção 3.3.3.2);
      fim do se
      se robôn entrou em comportamento monótono então
        Evoluir classificadores do robôn por monotonia (Seção 3.3.3.3);
      fim do se
    fim do para
  fim do para
  robô_base = Encontrar melhor robô;
  Atualizar melhor_fitness_obtido;
  para n=1 até número_de_robôs exceto robô_base faça
    Copiar sistema de navegação do robô_base para robôn;
    Evoluir conexões da rede do robôn (Seção 5.5.3.2)
  fim do para
fim do enquanto
Salvar o melhor sistema de navegação encontrado;

```

5.5.3.1. Evolução dos Classificadores

Da mesma forma que no SNA relativo do Capítulo 3, a aprendizagem deste sistema se dá por um processo evolutivo que é responsável por aperfeiçoar a população de classificadores toda vez que um evento de colisão, captura ou monotonia é detectado. O diagrama apresentado na Figura 3.3 indica como se dá a evolução de classificadores. Os procedimentos utilizados em cada caso já foram devidamente explanados na Seção 3.3.3 e, portanto, serão suprimidos aqui.

Note que, muito embora a ocorrência de colisões possa significar que o sistema de navegação está falhando, as colisões são essenciais para a evolução do mesmo. Se o robô não colide, ele nunca aprenderá a evitar obstáculos, uma vez que o sistema não possui comportamentos inatos.

5.5.3.2. Evolução da Rede

O processo de evolução da arquitetura da rede é ilustrado na Figura 5.4. Inicialmente, um número arbitrário de robôs é inicializado, cada um com uma instância da rede imuno-classificadora (ILCN) independente, com classificadores e conexões produzidos aleatoriamente.

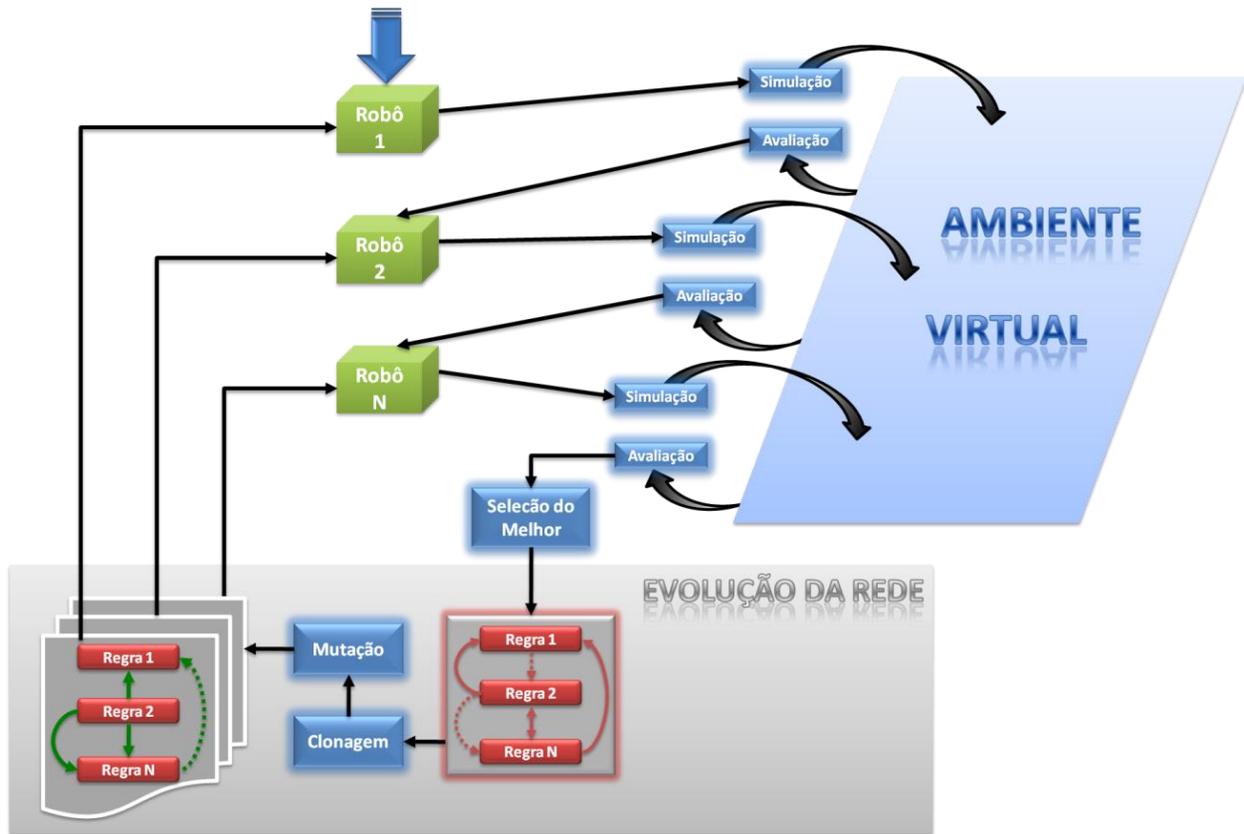


Figura 5.4: Processo de evolução da arquitetura da rede no sistema ILCN.

A seguir, realizam-se simulações com cada robô, deixando que ele navegue por um intervalo de tempo no ambiente, sob as mesmas condições iniciais. Ao longo das simulações, cada robô tem suas próprias experiências, sofrendo colisões, capturas e eventos de monotonia, que acionam a evolução dos classificadores conforme já discutido. Note que as conexões entre classificadores não mudam durante as simulações. Ao final de cada época, ou seja, quando todas as simulações de cada robô chegam ao fim, a evolução da arquitetura da rede tem lugar.

Em outras palavras, cada robô possui seu sistema e sua rede de classificadores. Baseado no seu desempenho global durante sua simulação, sua rede terá uma avaliação melhor ou pior que a dos outros robôs. Esta medida de avaliação, formulada matematicamente pela Equação 5.3, representa o *fitness* da rede que será usado no processo reprodutivo descrito a seguir.

$$Fitness_r = Capturas_r - Colisões_r$$

Equação 5.3

onde $Capturas_r$ e $Colisões_r$ indicam o número de capturas e colisões que ocorreram durante a simulação do robô r .

Com base no *fitness* calculado para cada robô, a rede imuno-classificadora do melhor deles é selecionada, enquanto as demais são descartadas. Assim como ilustrado pela Figura 5.4, o melhor sistema é clonado e replicado para todos os demais robôs, não sem antes fazer mutação nas conexões da rede (apenas as conexões são modificadas nesta etapa). A mutação consiste em inserir novas conexões aleatórias entre classificadores e remover conexões existentes, o que obedece a uma taxa de probabilidade de 2%. O processo de clonagem e mutação é repetido até que novas redes sejam geradas e transferidas para todos os robôs, exceto aquele cujo sistema obteve o melhor *fitness*, sendo este preservado. Finalmente, o ciclo de simulações se reinicia e o processo é repetido continuamente, até que um número máximo de épocas seja atingido ou um desempenho mínimo desejado (*fitness*) seja obtido.

Ao final da etapa de simulações e evolução das redes de classificadores, o melhor sistema de navegação encontrado é armazenado e empregado em experimentos posteriores por robôs reais ou virtuais. Um detalhe a se ressaltar é que não há operadores de cruzamento nesta evolução pela seguinte razão: o sistema atua como uma estrutura monolítica (conjunto de regras que se conectam entre si), não sendo plausível combinar subconjuntos de redes diferentes, mesmo que elas possam parecer semelhantes.

5.6. Experimentos e Resultados

Esta seção relata experimentos realizados no intuito de investigar o sistema de navegação autônomo conexionista (ILCN) e também analisar seu desempenho em relação aos resultados esperados. Todos os ambientes usados nos experimentos contêm uma configuração fechada e possuem obstáculos e alvos dispostos arbitrariamente, formando configurações não conhecidas pelos robôs. Tanto simulações computacionais quanto experimentos reais foram executados neste trabalho, um complementando o outro. Isto é, a evolução do sistema é feita em simulação e sua validação se dá com o robô real em ambientes reais.

No caso das simulações, os alvos têm que ser capturados conforme uma ordem fixa seqüencial. Quando o robô completa a seqüência de capturas, ela é reiniciada. Desta forma, há sempre apenas um alvo ativo no ambiente, o qual é desativado ao ser capturado, levando o próximo da lista a se tornar ativo. Para facilitar o entendimento, a seqüência de alvos é indicada nas figuras desta seção por meio de

rótulos numéricos. Nos ambientes reais, o alvo é, de fato, uma fonte de luz que deve ser alcançada pelo robô.

Os experimentos virtuais aconteceram em um simulador desenvolvido pelo autor, usando o modelo do robô Khepera II. Cada caso foi simulado 10 vezes para se chegar aos resultados compilados nesta seção. Alguns parâmetros do sistema e das simulações foram determinados empiricamente. O número inicial de classificadores (anticorpos) na rede é 100, todos criados aleatoriamente; a quantidade máxima de conexões por classificador vale 10; a taxa de cruzamento para evolução de classificadores é de 100% e a de mutação de 0,5%. Na evolução da rede, a taxa de mutação é de 2%. Os coeficientes da Equação 5.1 são: $\alpha=0,2$, $\beta=0,1$, $\gamma=0,5$, e $h=0,02$.

5.6.1. Experimentos Reativos

Este estudo de caso consiste de um ambiente contendo dois alvos, separados por um obstáculo central, que devem ser capturados ciclicamente (veja Figura 5.5). Note que a configuração do ambiente provoca um problema multi-objetivo, uma vez que o robô confronta situações conflitantes sempre que tenta capturar o alvo, o qual se encontra atrás do obstáculo central. Não há, no entanto, becos ou mínimos locais no ambiente. Sendo assim, um controlador puramente reativo pode resolver o problema. Foi exatamente disto que demonstrou ser capaz o SCA desenvolvido pelo autor em Cazangi et al. (2003), obtendo bons resultados em diversos ambientes reativos, inclusive este apresentado na Figura 5.5.

Embora o ILCN seja teoricamente mais sofisticado, a comparação com o SCA de Cazangi et al. (2003) é bastante válida. Para tanto, cada um dos dois sistemas passou por 10 simulações no ambiente da Figura 5.5. No caso do ILCN, cinco robôs foram utilizados na evolução da rede, sendo que a época da mesma foi de 10 mil iterações. Lembre-se que uma iteração é o mesmo que um ciclo sensorio-motor (leitura sensorial seguida por ação de controle). As simulações eram encerradas sempre que um robô capturava cinco alvos de forma consecutiva (sem colisões ou eventos de monotonia entre elas), comportamento considerado ótimo neste experimento.

A Tabela 5.3 apresenta os resultados obtidos por ambos os sistemas analisados. Observe que o SCA convergiu muito mais rápido do que o ILCN neste caso. As colunas de tempo de simulação, gerações dos classificadores e iterações mostram isto claramente. Em termos de iterações, pode-se ver que, no pior caso, o SCA convergiu em 21633, que é quase o mesmo valor da evolução mais rápida do ILCN (20000).

Tamanho disparidade pode ser explicada com base na complexidade dos sistemas. Note que, no ILCN, tanto os classificadores quanto a rede precisam ser evoluídos; já no SCA, apenas a evolução dos classificadores é suficiente para emergir um comportamento satisfatório no cenário reativo.

Tabela 5.3: Resultados obtidos no experimento reativo comparando o sistema classificador do Capítulo 3 com a rede imuno-classificadora após 10 simulações.

Original	Iterações	Gerações dos Classif.	Tempo de simulação (s)
Médio	13498.4	32,3	101,2
Mínimo	4598	10	44
Máximo	21633	48	176

Conexionista	Iterações	Gerações da Rede	Gerações dos Classif.	Classif. Número	Número de Conexões	Tempo de simulação (s)
Médio	46000	4,6	80,7	82	349,9	101,2
Mínimo	20000	2	41	74	260	44
Máximo	80000	8	139	90	457	176

Embora o ILCN tenha precisado, em média, de 80,7 gerações para a evolução de classificadores (contra 32,3 do outro sistema), ambos cumpriram os objetivos propostos. O melhor comportamento obtido para cada sistema pode ser visto na Figura 5.5.

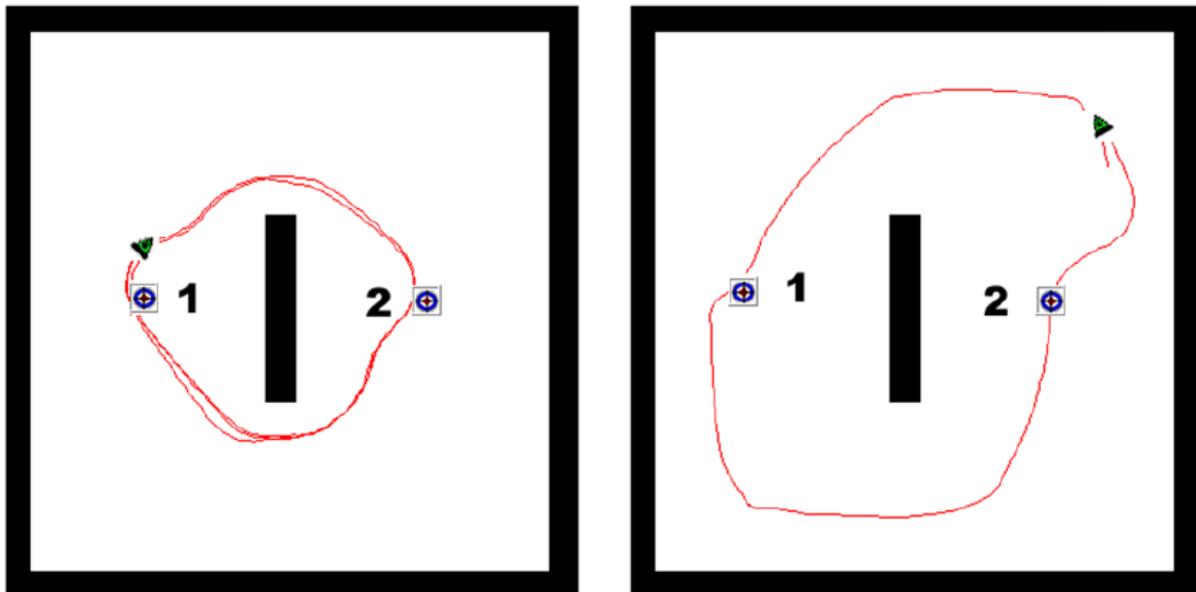


Figura 5.5: Comportamento final de navegação observado para o ILCN (à esquerda) e o SCA do Capítulo 3 (à direita) no ambiente reativo.

O melhor ILCN evoluído durante as simulações foi transferido para o robô Khepera II e colocado em um ambiente real análogo ao virtual, visto na Figura 5.6, visando sua validação. Assim como na simulação, o sistema também foi bem-sucedido no experimento real, indicando robustez em relação a diferentes ambientes.

Uma vantagem do sistema conexionista pode ser destacada: o número de classificadores na rede final evoluída foi, em média, 82, chegando a 74 em uma simulação. Ou seja, a meta-dinâmica da rede imunológica foi responsável por reduzir a população em até 36% do seu tamanho original. De qualquer forma, pode-se concluir que o SCA provou ser mais adequado para ambientes reativos, graças à sua menor complexidade.

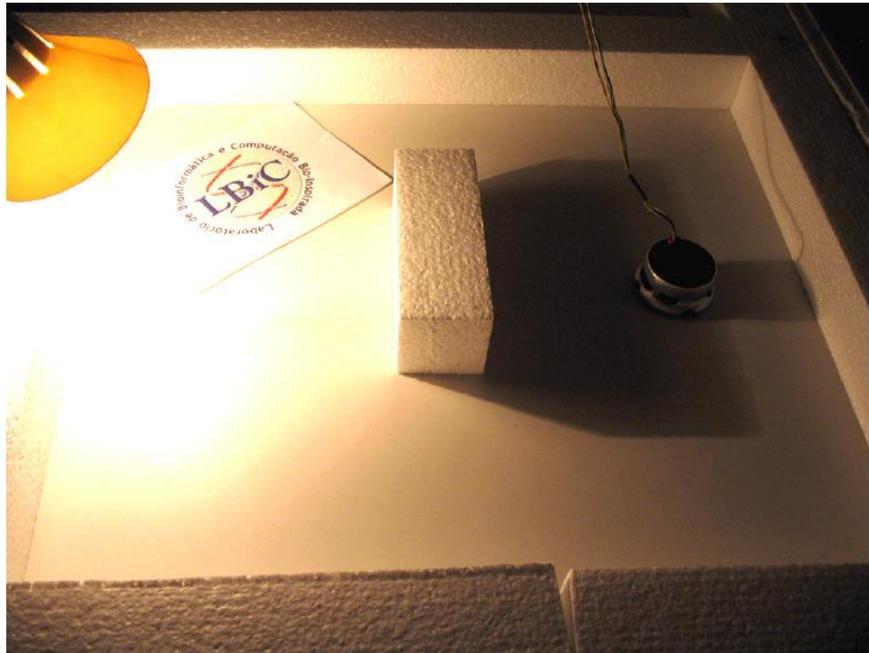


Figura 5.6: O melhor sistema ILCN evoluído em simulação foi transferido para o robô real para ser testado. Parte deste experimento pode ser visto no Vídeo 5.1.

5.6.2. Experimentos com Mínimos Locais

Nesta seção, a intenção é colocar o sistema proposto em ambientes com becos e mínimos locais, nos quais sistemas puramente reativos não funcionariam. Assim, espera-se mostrar o potencial do sistema conexionista em superar tais problemas.

Inicialmente, 10 simulações foram feitas com o ILCN no ambiente mostrado na Figura 5.7, que contém quatro alvos a serem capturados em seqüência. 10 robôs foram usados durante a evolução da rede, cuja época encerrava 10 mil iterações. O critério de parada para convergência é o mesmo do caso anterior: a evolução da rede termina quando ao menos cinco alvos são capturados consecutivamente.

Os resultados da simulação estão dispostos na Tabela 5.4. O destaque é o desempenho positivo alcançado pelo sistema em todas as 10 simulações, o que indica definitivamente que a rede imuno-classificadora conta com recursos eficientes para superar mínimos locais. O maior nível de dificuldade deste estudo de caso fica evidente ao se investigar os resultados obtidos (não foram realizados experimentos comparativos, pois o SCA não resolve problemas de navegação com mínimos locais). Em média, as evoluções produziram 36,4 gerações de redes (364 mil iterações em 28,29 minutos) e 545,7 gerações de classificadores. Outra observação é que o número total de eventos evolutivos (colisões + capturas + monotonia) do melhor robô por geração de rede foi 35.

Em relação à complexidade das redes evoluídas, de novo a habilidade natural da rede imunológica de eliminar anticorpos inúteis mostrou-se válida. Note que a quantidade de classificadores, que originalmente era 100, foi reduzida para 72,7, em média, após evoluir. Em termos de desempenho de navegação, medido pelo fitness geral (número de capturas subtraído pelo número de colisões), o valor médio foi de 10,8. O melhor sistema evoluído obteve fitness 16, ou seja, o robô foi capaz de capturar 1,6 alvos a cada mil iterações (considerando que o experimento levou 10 mil iterações).

Tabela 5.4: Resultados obtidos em 10 simulações no ambiente com becos do ILCN.

Conexionista	Gerações da Rede	Gerações dos Classif.	Fitness	Número de Classif.	Número de Conexões	Tempo de simulação (s)
Médio	36,4	545,7	10,8	72,7	264,8	1697,3
Mínimo	4	35	7	63	177	188
Máximo	91	1316	16	82	371	4171

A ilustração à esquerda da Figura 5.7 exhibe o comportamento final obtido pelo melhor ILCN evoluído dentre as 10 simulações. Analisando a trajetória do robô e sua velocidade a cada passo (indicada ao longo da trajetória), é possível compreender seu comportamento. Basicamente, o robô navega em mínima velocidade (1 u.v.) quase o tempo todo, visando desviar dos obstáculos, e acelera apenas quando chega perto dos alvos.

Para mostrar a relevância e o papel das conexões entre classificadores na superação de mínimos locais, outro teste foi executado. Dado o beco da Figura 5.8, duas simulações de curta duração (1000 iterações) foram realizadas. Em ambas, as condições iniciais foram as mesmas: um robô, controlado pelo melhor ILCN evoluído no experimento anterior, foi colocado na mesma posição inicial, devendo capturar o alvo 1 e, depois, o 2. A grande diferença foi que as conexões entre os classificadores foram desabilitadas no teste mostrado à direita da Figura 5.8, reduzindo o sistema a uma abordagem puramente reativa. O efeito é notável: enquanto o sistema conexionista passa pelo beco sem problemas, o sistema sem conexões fica preso virtualmente para sempre. O que aconteceu foi que o sistema perdeu informações indispensáveis incorporadas pelo padrão das conexões entre classificadores, tornando-o incapaz de ignorar a atratividade do alvo momentaneamente em favor de superar o beco.

Apesar de o desempenho do melhor ILCN evoluído ser considerado bom, ele está distante de ser uma solução ótima, uma vez que uma atitude de navegação em velocidade mais alta faria com que o robô capturasse mais alvos em menor tempo. Em busca desta otimização, simulações adicionais foram realizadas com um critério de parada diferente, que determina o fim da evolução apenas em caso de estagnação da curva de fitness. Como consequência, o melhor ILCN evoluído sob esta nova perspectiva conseguiu capturar alvos a uma taxa de 4 por mil iterações (fitness 40). Seu comportamento após a convergência é mostrado à direita da Figura 5.7 (veja também o Vídeo 5.2) e a arquitetura de sua rede é ilustrada na Figura 5.12. Examinando a trajetória neste caso, nota-se que o robô navega em máxima velocidade (3 u.v.) freqüentemente, freando apenas ao sair dos becos e em regiões perigosas, como quinas.

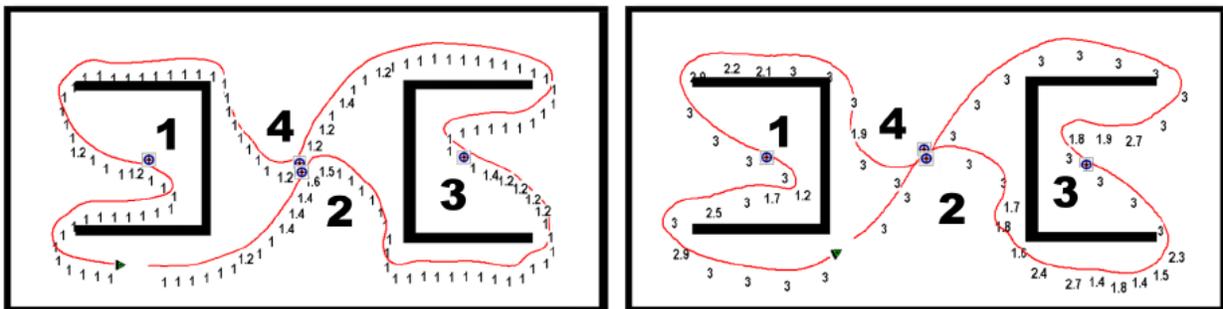


Figura 5.7: Comportamento de navegação do ILCN evoluído, com fitness 16 (à esquerda) e 40 (à direita).

Visando validar em ambientes reais a capacidade do sistema de escapar de mínimos locais, o sistema com fitness 40 foi transferido para o robô Khepera II e este colocado no ambiente com um beco

mostrado à esquerda da Figura 5.9. O robô inicia sua navegação dentro do beco, vai até seu fundo para então retornar e escapar em direção à fonte luminosa. Ao alcançar a região iluminada, representando a captura do alvo, o robô é pausado por alguns segundos e o ambiente é modificado. A alteração faz com que o robô esteja novamente dentro do beco e a lâmpada é movida para o lado oposto do ambiente. A seguir, o robô volta a navegar e, mais uma vez, consegue escapar do beco e alcançar o alvo. A modificação do ambiente durante o experimento se deu pois não havia espaço suficiente para colocar dois becos juntos. No entanto, não houve nenhuma mudança no sistema que controlava o robô. O Vídeo 5.3 mostra o experimento, sendo que uma seqüência de quadros retirada dele está disposta na Figura 5.10.

Embora o feito tenha sido realizado com maiores dificuldades do que na simulação, causadas especialmente pelos sensores extremamente ruidosos, a rede de classificadores apresentou um comportamento básico análogo ao obtido em simulação, levando o robô a cumprir seu objetivo. Além de servir para validar o sistema em condições reais, este resultado aponta outra propriedade importante: a capacidade de generalização de conhecimento. Ou seja, resultados semelhantes foram obtidos mesmo em topologias e sob condições ambientais distintas daquelas em que o sistema foi originalmente evoluído.

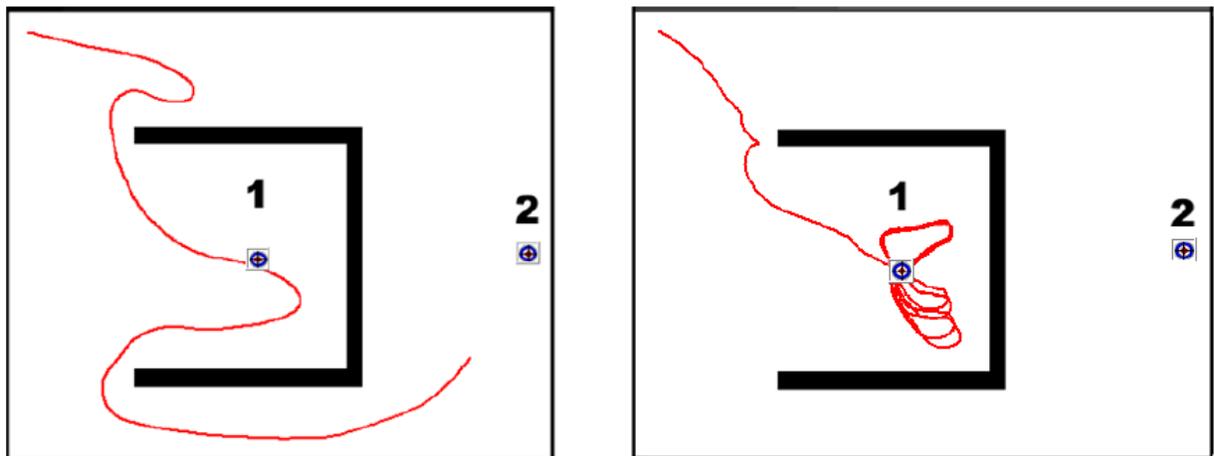


Figura 5.8: À esquerda, o ILCN evoluído consegue escapar do beco. À direita, o mesmo sistema, mas sem conexões entre classificadores, entra em um ciclo infinito por motivo do mínimo local.

O ILCN foi testado em um ambiente real ainda mais complicado, mostrado à direita da Figura 5.9. Desta vez, o beco foi montado em forma de um corredor longo, o que exige uma pró-atividade ainda maior do sistema visando sair de lá e depois buscar o alvo. O mesmo sistema evoluído foi responsável

por controlar o robô Khepera II neste teste. O Vídeo 5.4 e também a seqüência de quadros da Figura 5.11 mostram que, novamente, o sistema foi robusto e eficiente ao ser capaz de escapar do longo beco e atingir a fonte luminosa. Percebe-se que o robô vai até o fundo do beco para então navegar até sua saída, ignorando os estímulos atrativos (gradiente luminoso) da lâmpada por um tempo, e, quando fora do beco, o robô se encaminha para a região luminosa.

Embora não seja totalmente visível no Vídeo 5.3 e no Vídeo 5.4, o robô detecta o gradiente luminoso em qualquer parte do ambiente, até mesmo quando ele está dentro do beco. Este é um dos fatores que influencia o sistema na execução de estratégias de navegação que evite mínimos locais. Quer dizer, o comportamento de navegação do sistema está distribuído pelos classificadores e conexões, e a dinâmica da rede tem o papel de ignorar a atração causada pela fonte luminosa até que o robô escape do beco. Na prática, quando o robô fica preso no beco, ele vai até a frente do obstáculo, desvia dele e, então, mesmo detectando luz em seus sensores traseiros, o padrão das conexões da rede promove uma diminuição da concentração dos classificadores associados à captura de alvo. Assim, o robô pode se afastar do beco e depois ir até o alvo.

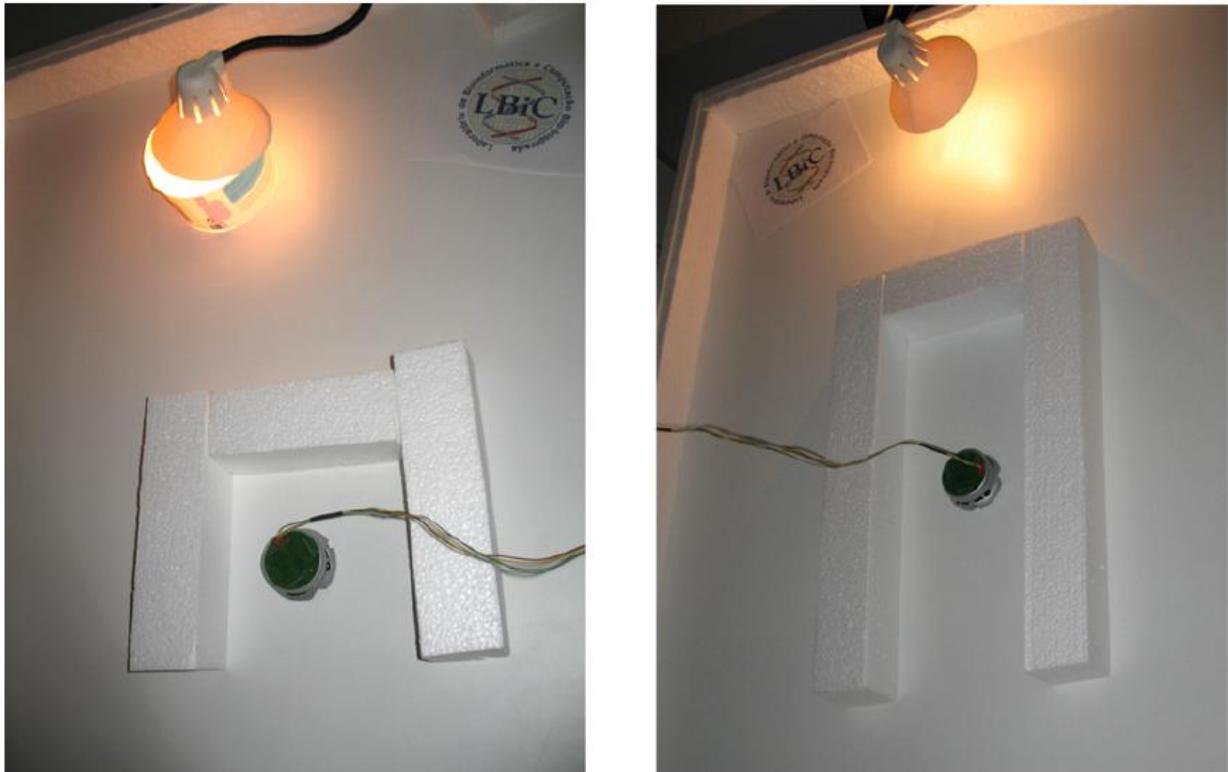


Figura 5.9: Ambientes montados para experimentação real da rede imuno-classificadora. No ambiente da esquerda se tem um beco curto e, à direita, um beco mais longo. A lâmpada é o alvo a ser capturado pelo robô.

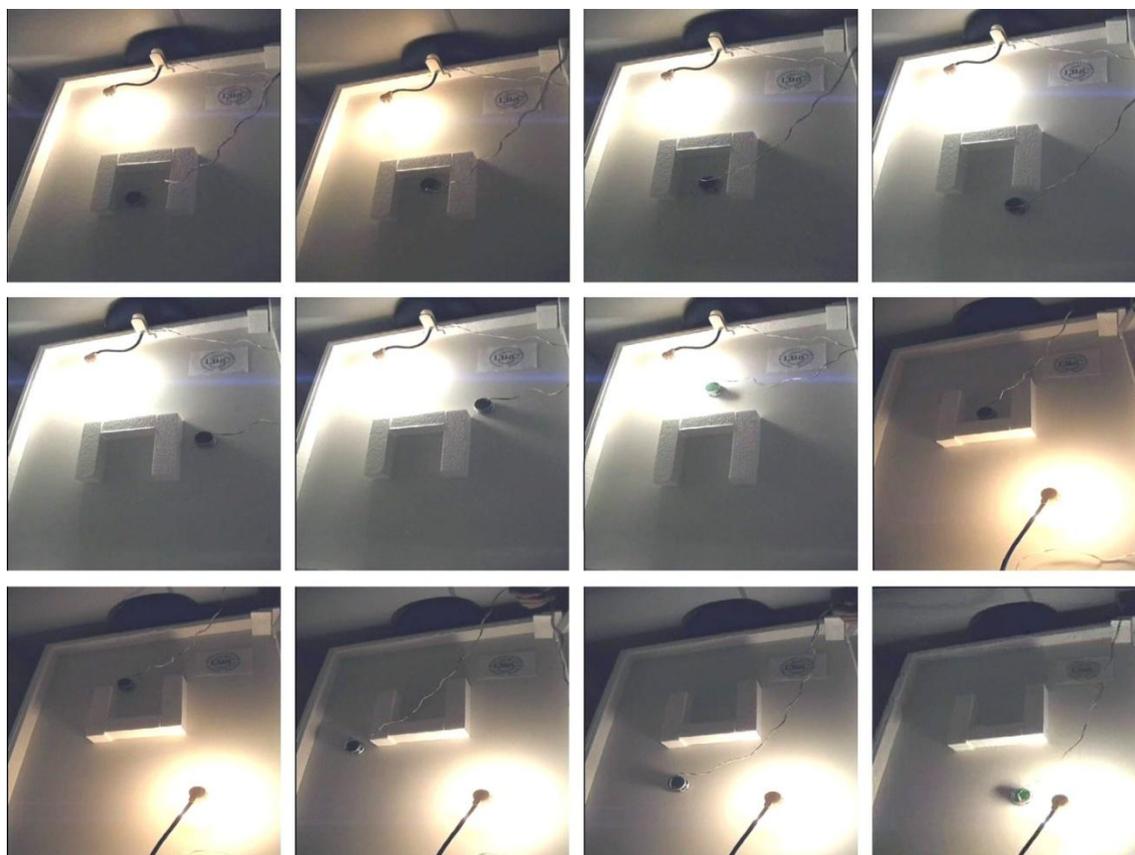


Figura 5.10: Seqüência de quadros retirada do Vídeo 5.3 que mostra o robô escapando por duas vezes do beco.

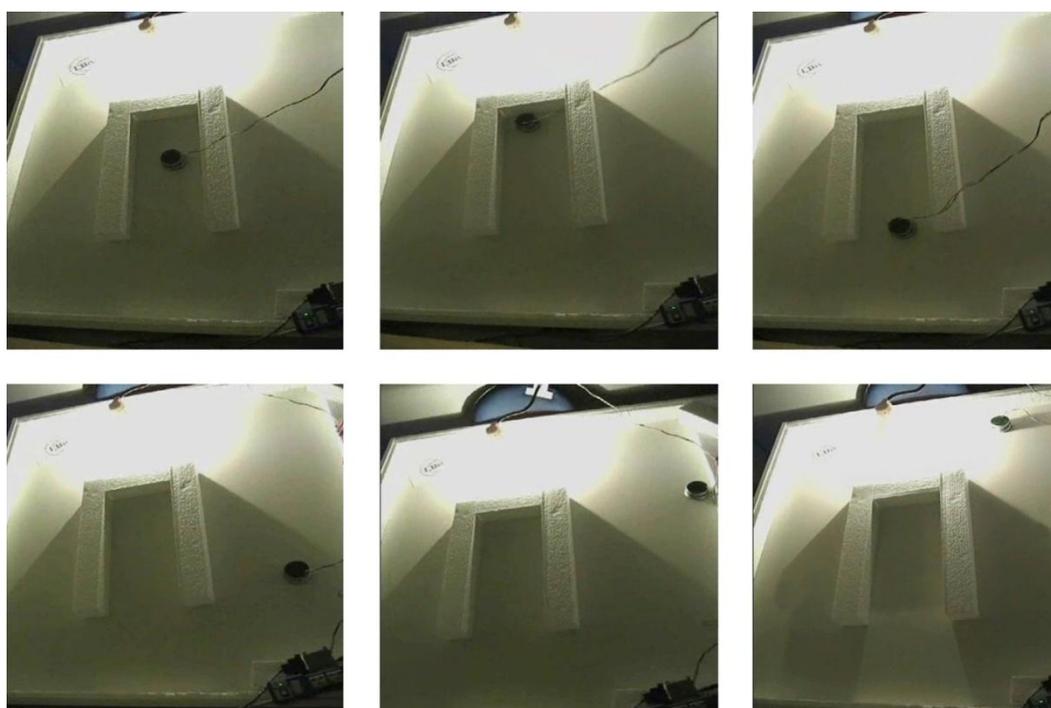


Figura 5.11: Seqüência de quadros proveniente do Vídeo 5.4 em que o robô escapa de um beco mais longo e atinge a fonte luminosa.

5.6.3. Análise da Rede

Assim como Bates & Elman (2002) afirmam, as representações e regras incorporadas em redes conexionistas são implícitas e altamente distribuídas, sendo muito difícil compreender exatamente o papel específico de cada conexão no todo. Entretanto, analisando globalmente a rede da Figura 5.12 é possível presumir algumas conclusões sobre sua arquitetura. Ela mostra a rede de classificadores que obteve melhor desempenho durante os testes (i.e., fitness igual a 40) e cujo comportamento é mostrado à direita da Figura 5.7.

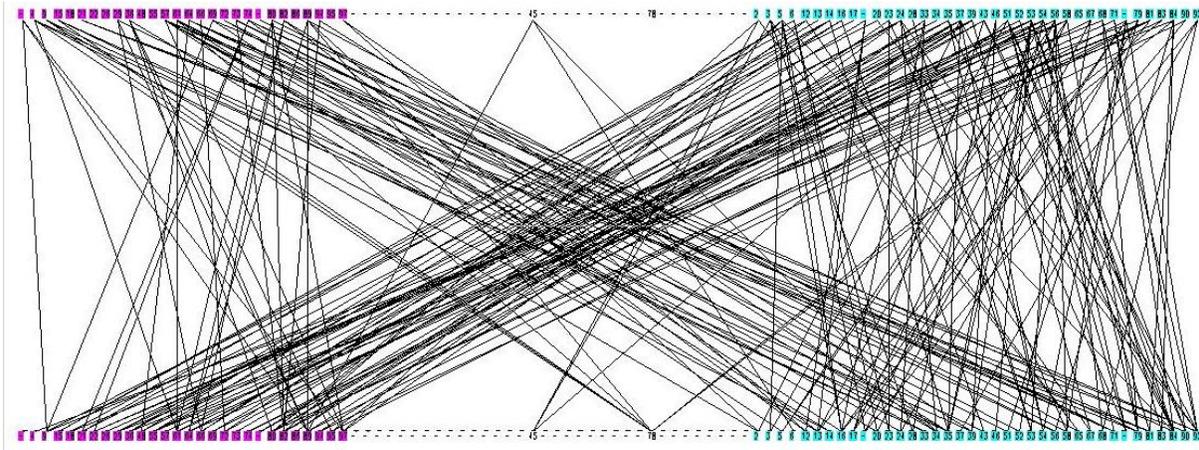


Figura 5.12: Arquitetura da rede evoluída do ILCN capaz de capturar 4 alvos a cada mil iterações. Há 80 classificadores e 337 conexões. Os nós acima e abaixo são os mesmos, sendo assim dispostos para melhor visualização.

Os 80 nós (classificadores) da rede estão agrupados em três classes, segundo tipo de comportamento: os 36 à esquerda são associados a desvio de obstáculos, os 40 à direita se associam à captura de alvos e os 4 restantes (no centro) não têm um comportamento específico aparente. Neste caso, o que determina o comportamento de um classificador é o último tipo de evolução de que ele fez parte (e.g., um classificador que participou por último de uma evolução disparada por um evento de colisão é classificado como constituinte do comportamento de desvio de obstáculos).

Observe que a rede é relativamente organizada e existe um equilíbrio entre o número de classificadores de ambos os comportamentos, havendo intra e inter conexões entre eles. Esta alta conectividade entre classificadores de diferentes tipos é consequência das diversas situações conflitantes impostas pela topologia do ambiente ao robô, também indicando que há intensa interação deles durante a operação do sistema. A ausência de nós na região central da rede da Figura 5.12 é resultado da eliminação automática de classificadores pouco úteis.

Na Figura 5.13, a mesma rede é mostrada sob uma perspectiva diferente. O tamanho do círculo que representa o nó é proporcional à sua relevância global para a rede, i.e., círculos maiores representam nós com maior número de conexões. Ao comparar os nós mais importantes da rede com os nós mais ativados quando esta rede controlava o robô da simulação mostrada no Vídeo 5.2 (após a convergência do sistema), a natureza distribuída do conhecimento presente no sistema fica evidenciada. Isto é, os classificadores que agiram mais freqüentemente (37, 29 e 67) não são os mais conectados (24 e 61). Na verdade, os classificadores mais usados são nós com baixo grau de conectividade. Portanto, os nós mais importantes para a dinâmica da rede não têm papel direto no processo de tomada de decisão, mas sua relevância para o sistema como um todo é significativa.

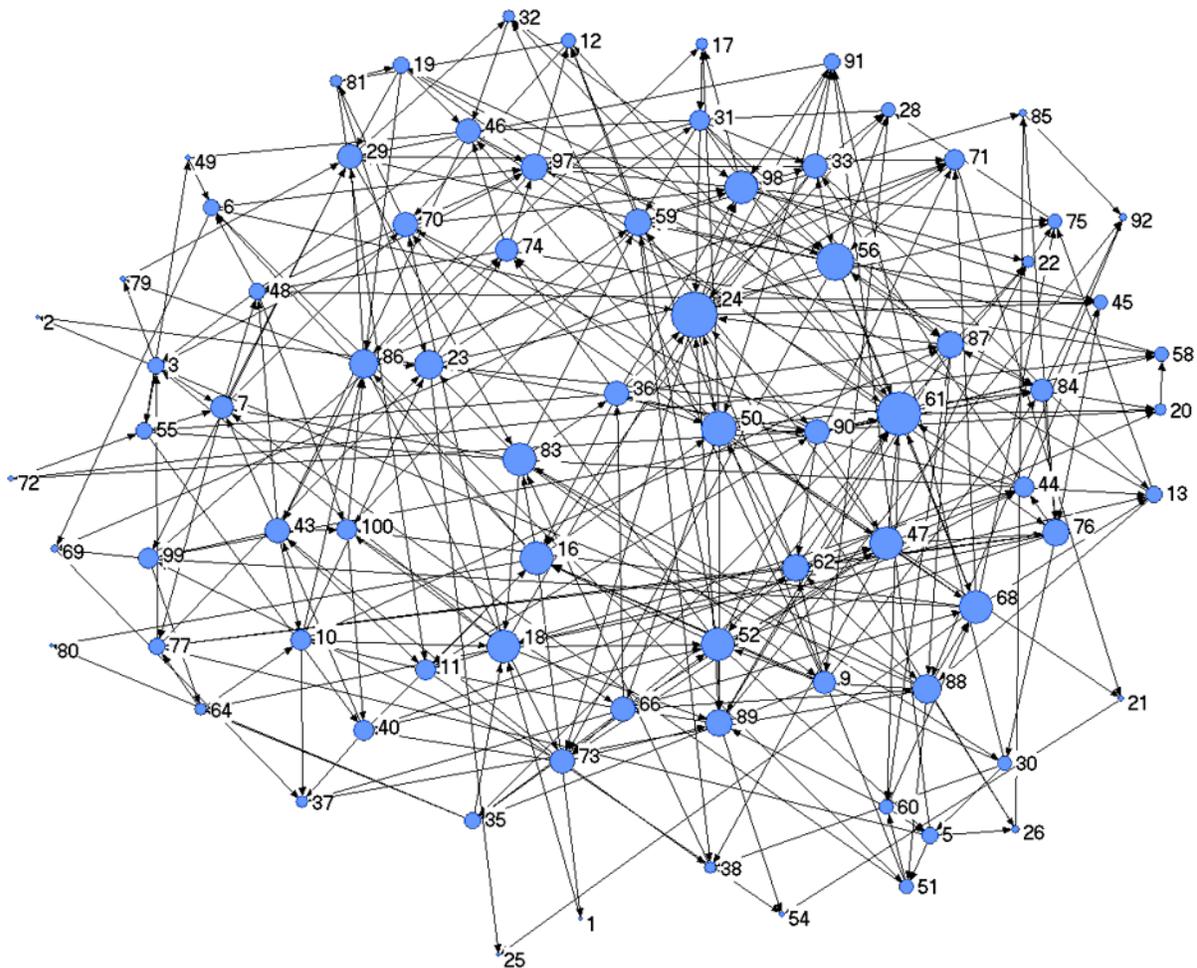


Figura 5.13: Mesma rede da figura anterior mostrada de uma forma diferente. Os nós com maior grau de conectividade são representados por círculos maiores. Os nós 37, 29 e 54 são os mais ativados durante a navegação do robô, ao mesmo tempo em que os nós 24 e 61 são os mais importantes globalmente. A posição dos nós na figura foi definida arbitrariamente pelo software de visualização utilizado (NetDraw®).

5.7. Conclusão

Inspirado em propriedades atrativas que emergem da dinâmica intrínseca dos modelos da rede imunológica, e aproveitando as habilidades dos sistemas classificadores com aprendizado (SCA) no gerenciamento do conhecimento, um sistema conexionista autônomo para navegação de robôs, chamado Rede Imuno-classificadora com aprendizagem (ILCN), é proposto visando combinar os atributos complementares dos sistemas originais. A hibridização é feita fundindo o sistema de navegação original apresentado no Capítulo 3, que se baseia em uma versão melhorada do SCA com o modelo da rede imunológica [(Jerne, 1974)(Farmer, Packard, & Perelson, 1986)]. O sistema é sintetizado por meio de dois processos evolutivos, um devotado à busca por um conjunto de classificadores, e o outro responsável pela configuração do padrão de conexões entre os classificadores. Resultados bem-sucedidos em problemas multi-objetivos desafiadores foram relatados neste capítulo, envolvendo situações de mínimos locais em que sistemas puramente reativos se mostram impotentes.

Especificamente, o ILCN foi aplicado a dois conjuntos de experimentos. O primeiro considerou um ambiente sem mínimo local. Os testes mostraram que, embora o ILCN seja mais lento que o sistema de navegação reativo do Capítulo 3 em relação à convergência, devido ao número de parâmetros a serem ajustados, ambos os sistemas foram capazes de aprender a desviar de obstáculos e capturar alvos eficientemente. O segundo conjunto de experimentos contou com um ambiente de maior grau de dificuldade, pois existiam dois becos dispostos de forma a causar mínimos locais. Os resultados obtidos confirmaram que, após a evolução, os robôs controlados pelo ILCN apresentaram comportamentos emergentes bastante eficazes em realizar as tarefas de navegação e, principalmente, escapar dos mínimos locais. Além disso, em todos os casos, a evolução dos sistemas conexionistas produziu soluções parcimoniosas, graças à eliminação automática de classificadores inúteis (característica herdada da meta-dinâmica imunológica).

Os experimentos executados visando a avaliação da proposta tiveram simulação computacional e testes em ambientes reais. A simulação foi usada principalmente para evoluir e ajustar os sistemas, enquanto os testes reais serviram para validar os controladores em robôs reais. O ILCN mostrou ser robusto e ter boa capacidade de generalização, uma vez que ele se deu bem em ambientes cuja topologia e condições eram diferentes daquelas encontradas onde o sistema foi originalmente evoluído.

Em termos de futuro, vislumbra-se a experimentação da rede imuno-classificadora em outros problemas de mínimo local com diferentes topologias. Adicionalmente, uma característica a ser melhor explorada é o processo de evolução da rede, visando encontrar melhores operadores e funções de fitness mais sofisticadas que possam tornar possível a obtenção de redes melhores em termos de parcimônia e eficiência. Uma análise mais avançada das redes também faz parte dos planos para que se tente entendê-las melhor, particularmente a relação entre a dinâmica da rede e suas capacidades cognitivas.

5.8. Descrição dos Vídeos

Vídeo 5.1: Parte do experimento em que o sistema conexionista (ILCN) é testado em um ambiente sem mínimos locais.

Vídeo 5.2: Simulação que mostra o comportamento do robô controlado por melhor rede de classificadores evoluída (fitness igual a 40) em um ambiente com dois becos. As barras mostradas na parte baixa do vídeo indicam a variação da concentração dos classificadores de acordo com a navegação do robô.

Vídeo 5.3: Experimento em que o sistema obtido via simulação é testado em um ambiente real com becos. Tão logo o robô alcança a fonte luminosa, a armadilha é invertida e a lâmpada movida para o lado contrário da arena.

Vídeo 5.4: Experimento em que o ambiente possui um beco longo, tornando mais difícil para que o robô controlado pela rede imuno-classificadora com aprendizagem escape dele.

Capítulo 6: Abordagem Evolutiva para Rastreamento Robótico de Alvos Móveis

6.1. Introdução

Até este ponto do trabalho, têm sido discutidos muitos problemas de navegação de robôs que supõem a operação em ambientes completamente desconhecidos *a priori*. Mas e quando há disponibilidade de conhecimento parcial ou total do mundo ao redor do robô, quais as implicações e como usá-lo? Neste capítulo, estas e outras questões serão investigadas em problemas práticos que fazem parte do cotidiano de um hospital, visando sua automatização por meio de soluções robóticas.

O ambiente hospitalar é complexo em vários aspectos, incluindo organização, logística e estrutura física. Diferentemente de outros tipos de instituições, entretanto, um hospital é um ambiente para cuidados com saúde e, portanto, falhas ou ineficiências no gerenciamento desta complexidade inerente podem levar ao prejuízo último: vidas humanas. Neste sentido, todo esforço cujo objetivo seja de melhorar os processos que ocorrem no ambiente hospitalar é bem vindo.

Juntamente com outros campos, a informática médica tem contribuído intensamente com a missão de produzir soluções para a área de saúde (Chen, Fuller, Friedman, & Hersh, 2005). Gerenciamento de dados, automação de tarefas e otimização de processos são algumas das suas principais metas. Abraçando as duas últimas linhas, a robótica tem um papel fundamental como a extensão física da informática aplicada à medicina. Embora braços robóticos cirúrgicos sejam os robôs mais comuns nos hospitais atualmente, seus compartes móveis também têm grande importância e estão no foco deste capítulo.

Existem inúmeras aplicações possíveis de robôs móveis em ambientes médicos, tais como teleconferência remota entre médicos e pacientes, e transporte de materiais infecciosos. Neste trabalho, em particular, trata-se do problema de rastreamento de alvos móveis por um robô autônomo em um hospital. O alvo é, na verdade, um dispositivo emissor de sinais por rádio frequência, chamado RFID (*radio frequency identifier*) tag, que pode ser carregado por uma pessoa (e.g., médico, enfermeira, paciente), ou acoplado a macas, cadeiras de roda, e qualquer outro tipo de equipamento. Este dispositivo transmite sinais para diversas antenas distribuídas pelo prédio, gerenciadas por um sistema de localização, sendo possível identificar, em tempo real, as coordenadas 3D da posição onde o objeto de interesse de encontra.

Como motivação, imagine um médico necessitando acesso imediato ao banco de dados do hospital sem que haja um terminal por perto; ou uma enfermeira em busca de provisões; ou mesmo um paciente procurando por uma área específica do hospital. Em todos os exemplos, o robô pode ser útil no auxílio a tais tarefas. Ao ser requisitada sua presença, o robô deve ir até a pessoa, que não necessariamente estará no mesmo local o tempo todo, e então executar a tarefa solicitada. Dado que a pessoa pode estar se movendo pelo prédio, o robô tem que ser capaz de rastreá-la e segui-la onde quer que ela vá.

Este problema pode ser visto como uma classe especial de planejamento de trajetórias. Em geral, o planejamento de trajetórias busca determinar uma rota partindo de um ponto inicial até um ponto final no espaço, de modo que nenhuma colisão em obstáculo ocorra. No caso de ambientes não estáticos, que exigem reajustes de trajetória a todo momento, o problema passa a se chamar planejamento de trajetórias *online* (Dudek & Jenkin, 2000).

Um vasto número de trabalhos sobre planejamento de trajetórias pode ser encontrado na literatura. A categoria dos algoritmos exatos é geralmente dedicada a encontrar sempre o caminho ótimo, desde que este exista. No entanto, a necessidade intensa de recursos computacionais e conseqüentes atrasos podem tornar tais algoritmos proibitivos em problemas complexos de tempo real (Hwang & Ahuja, 1992). Uma alternativa são os algoritmos heurísticos, que procuram encontrar não a solução ótima, mas sim boas soluções dentro do período de tempo disponível. Muito embora exista a probabilidade de não se chegar a resultados razoáveis, métodos heurísticos têm se mostrado bastante adequados a problemas de planejamento em tempo real (Capozzi, 2001). Alguns exemplos são métodos probabilísticos (e.g., Schulz et al. (2003) e Roy et al. (2003)), e campos de potencial artificiais (e.g., (Ge & Cui (2002) e Poty et al.(2004)).

Ao se lidar com planejamento de trajetórias on-line em ambientes dinâmicos, propriedades como robustez em face de incertezas e flexibilidade a mudanças são essenciais. Estas características podem ser encontradas em abordagens de computação evolutiva, graças às suas estratégias baseadas em população, que agem como repositório de soluções alternativas, assim permitindo uma exploração em paralelo do espaço de busca. Estas técnicas são, portanto, adaptativas e podem responder com brevidade a modificações inesperadas no ambiente, devido ao seu poderoso mecanismo de busca [(Michalewicz, 1996) e (Yang, Ong, & Jin, 2007)].

Com base nos benefícios da evolução artificial, alguns pesquisadores têm desenvolvido planejadores evolutivos de trajetórias. Por exemplo, Xiao et al. (1997) apresentam uma proposta que faz planejamento on-line e off-line, mas admitindo apenas ambientes estáticos. Em Elshamli et al. (2004), o trabalho anterior é estendido para ambientes simples com obstáculos móveis. Em Capozzi (2001), por sua vez, um sistema de planejamento que consegue lidar com obstáculos e alvos móveis é proposto.

Apesar dos resultados promissores apresentados pelos trabalhos de ordem evolutiva mencionados, todos eles (assim como a maioria dos planejadores de trajetória em geral) são desenvolvidos e testados apenas em ambientes fictícios, normalmente supondo que a dinâmica envolvida é conhecida *a priori*.

Neste capítulo, entretanto, um sistema de navegação evolutivo é proposto para controlar um robô autônomo que deve operar no ambiente de um hospital real, a saber, o *Washington Hospital Center* (o maior em Washington D.C., EUA). O sistema é desenvolvido para operação em tempo real, sendo dotado de mecanismos para realizar tanto planejamento de trajetórias global quanto local. A aplicação, como já dito, trata do rastreamento de pessoas e equipamentos, acoplados a dispositivos RFID, que estão em constante movimentação por todo o hospital.

Diversos experimentos são executados, em simulação e com o robô real, mostrando a eficiência do sistema em várias situações que requerem respostas em tempo real, adaptabilidade para lidar com a complexidade dos problemas envolvidos na navegação em um prédio real, em particular a dinâmica imprevisível de seus obstáculos e alvos.

6.2. Contextualização do Capítulo

A primeira grande diferença desta frente para as demais está na aplicação do sistema autônomo. Embora o problema teórico tratado, planejamento de trajetórias, seja genérico, a aplicação em si é bastante específica, voltada para o rastreamento de pessoas em hospitais. É importante também distinguir o tipo de navegação feito nas frentes de pesquisa anteriores em relação a esta. Basicamente, planejamento de trajetórias envolve a determinação do caminho completo da origem ao destino antes da sua execução, diferentemente da navegação passo-a-passo das abordagens anteriores.

Na verdade, o estilo diferente de navegação não é apenas uma questão de conceito, mas é resultado das características dos problemas e dos sistemas propostos. Quando se fala em navegação passo a passo, refere-se ao ciclo de sensoriamento, tomada de decisão e atuação. Planejar trajetórias

implica em tomar uma decisão (rota) que engloba muitos ou todos os passos de navegação de uma vez só. Pode-se dizer que o primeiro tipo de tomada de decisão é local, enquanto o segundo é global.

O processo de planejamento de uma trajetória global possui características deliberativas, já que contempla a antecipação de passos futuros e a idealização de uma solução completa com base em conhecimentos inatos. Não é necessário reafirmar aqui os perigos de se usar exclusivamente este tipo de estratégia. Uma forma robusta de atenuar tais riscos é introduzir adaptabilidade e flexibilidade ao processo de planejamento por meio da integração de camadas reativas. É exatamente nesta hibridização que se fundamenta o controlador desenvolvido neste capítulo.

O sistema proposto combina características de sistemas deliberativos e reativos, dado que são reunidas duas camadas: uma de planejamento de trajetórias globais e outra de planejamento de trajetórias locais. Na prática, a primeira atua antes da navegação, usando a planta estrutural do hospital, conhecida previamente, para gerar uma trajetória completa de origem a destino; a segunda, por sua vez, opera ao mesmo tempo em que o robô navega, re-planejando segmentos locais da rota global com base nos estímulos instantâneos capturados pelos sensores. Fica claro, portanto, que a camada deliberativa assume o papel de orientação da camada reativa, tendo a última, todavia, total liberdade de operação.

Ambos os subsistemas empregam versões análogas de um mesmo algoritmo evolutivo. A propósito, esta proposta é um caso típico de aprendizagem por evolução, em que os processos de aprendizado são desvinculados. Isto é, a aprendizagem se dá apenas durante a evolução da trajetória, e não há acúmulo de conhecimento entre uma evolução e outra. A aprendizagem é de curtíssimo prazo e consiste em, a partir de trajetórias candidatas aleatórias, produzir uma rota final factível. Além disso, há também mecanismos de auto-ajuste de parâmetros do processo evolutivo, que operam de acordo com as dificuldades encontradas.

O planejamento de trajetórias, tanto global quanto local, possui um forte caráter otimizador, ou seja, as trajetórias são evoluídas sempre visando satisfazer critérios como minimização do seu comprimento e número de obstruções. No caso das trajetórias locais em particular, há ainda uma busca por maximizar sua proximidade em relação à rota global usada como guia.

O robô utilizado é o PatrolBot, desenvolvido pela MobileRobots Inc., que pode ser visto na Figura 3 do Capítulo 2. Trata-se de um robô autônomo programável de uso geral que, por ser de tamanho

natural, permite uma interação mais fácil com as pessoas no ambiente. Como parte do processo de desenvolvimento do controlador, além dos experimentos reais no hospital, também foram feitas simulações computacionais envolvendo cenários fictícios.

6.3. Localização e Posicionamento por Rádio Frequência

O sistema de localização de dispositivos por rádio-freqüência (RFID), conhecidos como *tags*, é uma tecnologia desenvolvida pela empresa Parco Merged Media Corp., que faz o rastreamento de pessoas e objetos, tendo uma precisão média de 60 cm. Para tanto, existem centenas de antenas instaladas por todo o hospital, cobrindo uma área de 230 mil m², além de 2500 *tags* (transponders). As *tags* são do tipo ativo pois emitem sinais a uma freqüência de 1 Hz. As antenas e as *tags* se comunicam sem fio por meio de sinais de banda ultra-larga.

Em termos práticos, este sistema de localização é voltado à observação e estimação do posicionamento da *tags*, as quais podem estar acopladas a equipamentos quaisquer ou ser carregadas por pessoas. Logo, qualquer objeto de interesse pode ser monitorado e rastreado em qualquer região do prédio. Obviamente, há restrições e imprecisões no sistema, as quais serão discutidas neste e no próximo capítulo.

Este sistema de localização pode ser entendido de forma análoga ao GPS (sistema de posicionamento global). Embora utilizem tecnologias diferentes, os dois sistemas são capazes de estimar a localização de objetos e pessoas, sendo que o GPS funciona apenas ao ar livre (*outdoor*) e o sistema de rádio freqüência é empregado dentro de prédios (*indoor*).

6.4. Abordagem Evolutiva para Planejamento de Trajetórias e Rastreamento de Alvos Móveis

6.4.1. Visão Geral

O sistema proposto nesta frente de pesquisa pode ser dividido em três partes: planejamento de trajetória global, planejamento de trajetória local e controle de navegação. As duas primeiras partes são os componentes mais importantes do sistema, fomentando totalmente a terceira parte. Basicamente, eles operam assim: o módulo de planejamento global produz uma trajetória global completa (da origem ao destino), a qual é parcialmente refinada e ajustada pelo módulo de planejamento local, de acordo

com as informações instantâneas capturadas pelos sensores do robô. Um segmento de trajetória é produzido, sendo este prontamente navegado pelo robô. A Figura 6.1 sintetiza o esquema de funcionamento e a organização do sistema proposto.

Há duas entradas no módulo de planejamento global: o mapa do ambiente (planta do hospital conhecida *a priori*) e a posição atual do alvo (coordenadas da *tag* de interesse fornecidas pelo sistema de localização por radiofrequência). Além disso, toda vez que o robô é solicitado ou se ele já se encontra em modo de rastreamento, e ocorre de o alvo se mover, a evolução da trajetória global é disparada. Uma vez evoluída, esta é passada ao módulo de planejamento local, o qual também utiliza as leituras dos sensores do robô. Sempre que novos estímulos sensoriais são recebidos, o módulo local evolui o segmento da trajetória global compreendido pela posição atual do robô até o ponto de alcance máximo dos seus sensores. O trecho local evoluído é então submetido ao módulo de navegação para que este determine a velocidade e direção, passadas aos motores de modo a navegar o trecho. Conforme indicado na Figura 6.1, a evolução da trajetória local e a sua navegação compreendem um ciclo que se repete até que a trajetória global seja totalmente cumprida, ou seja, o alvo tenha sido alcançado.

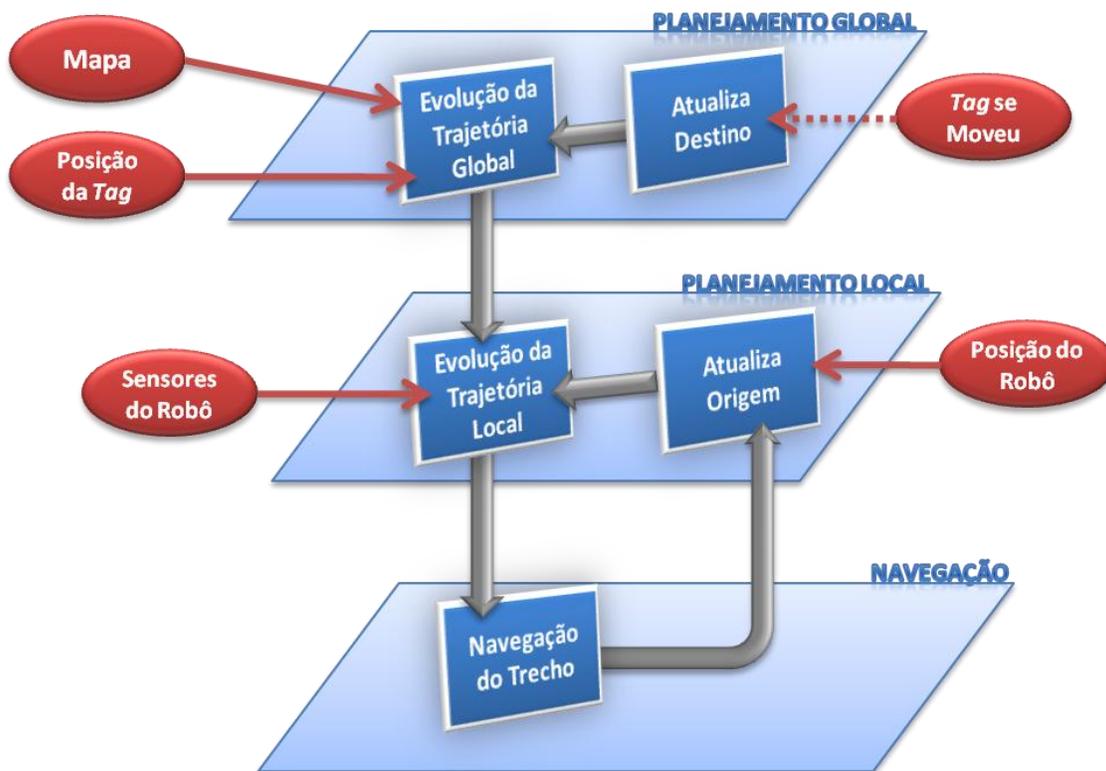


Figura 6.1: Diagrama que representa a organização e funcionamento da abordagem para rastreamento.

A geração e o ajuste das trajetórias feitos pelos subsistemas de planejamento global e local resultam de instâncias do mesmo processo evolutivo, as quais trabalham simultaneamente. Os procedimentos global e local diferem apenas no objetivo e configuração de seus parâmetros; a estrutura e a dinâmica são as mesmas. Por esta razão, a abordagem evolutiva será descrita como um todo nesta seção, sendo as particularidades de cada caso enfatizadas quando conveniente.

Assim como em algoritmos evolutivos tradicionais, esta proposta evolui uma população de soluções em potencial, com inicialização aleatória, por meio de um processo iterativo em que elas são avaliadas, selecionadas com base em seu fitness e geneticamente modificadas, até que ao menos uma solução apropriada surja.

Alguns poucos destaques merecem espaço aqui. Primeiro, um repositório de obstáculos é utilizado pelo sistema para evitar que regiões do mapa fora do contexto não sejam processadas. Isto é, de acordo com a região do mapa dentro do escopo da trajetória, utilizam-se apenas os obstáculos contidos naquela área (que pode aumentar ou diminuir). Além disso, a estratégia empregada pelo sistema evolutivo explora o espaço de busca aceitando a existência de soluções ineficazes e tentando corrigi-las. Por último, o alvo pode se mover a qualquer instante pelo fato de ser, é claro, independente do sistema de navegação. Nas próximas seções, cada parte do sistema será descrita em detalhes.

6.4.2. Codificação

Imagine uma trajetória, partindo de uma dada origem até um destino desejado, como uma seqüência de pontos interconectados, do modo mostrado pela Figura 6.2. Uma vez que o espaço de busca é representado por um mapa 2D, os pontos, chamados nós, podem ser indicados por um par ordenado $\langle x,y \rangle$. Logo, uma lista encadeada de pares ordenados pode representar adequadamente qualquer trajetória no problema tratado.

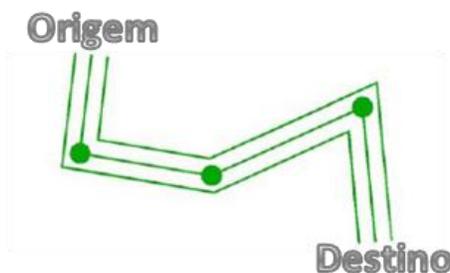


Figura 6.2: Exemplo de trajetória com três nós intermediários (juntas) e margens paralelas.

Portanto, cada indivíduo da população é expresso por uma lista ligada de nós $\langle x,y \rangle$ que codifica uma trajetória desde o início até o fim. O número de nós é variável, sendo ajustado durante o processo evolutivo de forma a se adaptar ao ambiente.

A codificação mencionada descreve trajetórias. Entretanto, algo a mais precisa ser considerado para realmente modelar uma rota factível: o tamanho do robô. Embora isto não seja necessariamente um componente da codificação, ele tem um papel crítico ao longo da evolução, particularmente na fase de avaliação (Seção 6.4.4). Em outras palavras, uma trajetória, mesmo que ótima, não é útil caso o tamanho do robô não tenha sido levado em consideração.

Como ilustrado na Figura 6.2, a solução adotada é adicionar pontos paralelos aos nós, sendo que a distância entre eles é igual ao diâmetro do robô, formando duas margens paralelas à trajetória planejada (linha central). Conseqüentemente, trajetórias cujas margens estão livres de obstruções garantirão espaço suficiente para o robô navegar.

Considerando que os pontos paralelos são obtidos meramente em função de variáveis conhecidas, eles não fazem parte da codificação dos indivíduos propriamente dita e, portanto, não são diretamente evoluídos. A evolução de trajetórias opera apenas nas coordenadas dos nós centrais.

6.4.3. Inicialização

Dados o local da origem e a posição do alvo, a trajetória ótima irrestrita é construída: uma linha reta entre eles. A seguir, ocorre o processo de inserção de nós na trajetória. No caso do planejamento global, para cada obstrução (i.e., intersecção entre a linha da trajetória e a linha que representa um obstáculo), um nó é introduzido naquele ponto exato (veja a Figura 6.3). Diferentemente, no caso local os nós são distribuídos uniformemente ao longo do segmento de trajetória.

Uma quantidade predefinida de indivíduos constitui a população. Todos têm o mesmo número de nós, cuja inicialização é feita tomando a linha reta original como referência e movendo os nós originais aleatoriamente. Para tanto, existem duas variáveis aleatórias: ângulo (0° a 359°) e raio (0 a $MaxRaioNo$).

Note que, durante o processo evolutivo, origem e destino não se alteram jamais, mesmo que os nós intermediários mudem de posição. Tanto a origem quanto o destino são comuns a todos os indivíduos, e suas coordenadas são modificadas apenas em resposta a movimentos reais do robô e do alvo sendo rastreado.

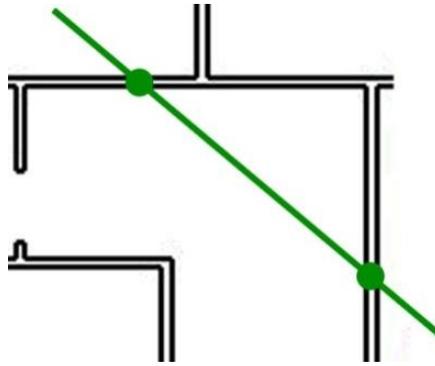


Figura 6.3: Ilustração que mostra como os nós são inicialmente inseridos para criar uma trajetória global.

6.4.4. Avaliação

O estágio de avaliação é responsável por guiar a evolução em direção a regiões mais promissoras do espaço de busca. Isto é feito atribuindo notas aos indivíduos, conforme a qualidade da trajetória que eles representam. Esta nota, chamada de fitness, torna possível identificar as melhores soluções, permitindo que suas boas características se propaguem para as próximas gerações.

Três fatores são levados em conta para medir o fitness dos indivíduos: número de obstruções, comprimento e quantidade de auto-cruzamentos. Dado que as obstruções degradam o grau de factibilidade da trajetória, elas são definitivamente o critério mais importante na hora da avaliação. Enquanto o comprimento da trajetória é importante em termos de otimização, os auto-cruzamentos (a trajetória cruza com ela mesma) deterioram a rota e, por isso, são penalizadas. A Equação 6.1, a seguir, indica a combinação linear dos critérios usados para computar o valor do fitness dos indivíduos, o qual se busca minimizar.

$$fitness(i) = n_obst(i) \times pen + len(i) + n_loops \times pen / 4 \quad \text{Equação 6.1}$$

onde i é o índice do indivíduo, n_obst é o número de obstruções, len representa o comprimento da trajetória, n_loops é o número de auto-cruzamentos, e pen é uma constante para penalização inicializada com valor igual ao comprimento da trajetória mais longa presente na população.

6.4.5. Seleção

O método de seleção adotado é o torneio. Basicamente, um subconjunto de indivíduos é escolhido aleatoriamente dentre todos da população e, dentre eles, os dois melhores são selecionados como genitores que se reproduzirão a seguir. O tamanho do torneio, ou seja, o número de indivíduos pegos

aleatoriamente no primeiro passo do processo, é predefinido como 20% do tamanho total da população.

6.4.6. Geração de Descendentes

Nesta fase, os genitores selecionados sofrem operações que criarão seus descendentes. A idéia é combinar boas trajetórias visando obter novas ainda melhores e também realizar alterações pontuais de forma a manter a diversidade da população. Neste sentido, dois operadores genéticos são aplicados aos indivíduos: cruzamento e mutação.

O cruzamento do tipo uniforme é executado primeiro, trocando, de acordo com uma taxa de probabilidade, nós entre os indivíduos genitores selecionados. Logo, dois genitores produzem sempre dois descendentes, havendo sempre a chance de que os filhos sejam clones dos pais, ou seja, não ocorra intercâmbio genético.

A seguir, o operador de mutação é aplicado aos novos indivíduos, obedecendo a uma probabilidade de mutação. A mutação consiste em deslocar um nó da sua localização atual para outra qualquer determinada aleatoriamente. Em particular, a direção do deslocamento pode variar de 0° a 359° , e a distância euclidiana máxima permitida é de *MaxRaioNo*.

Além disso, existe um mecanismo especial para ajustar os parâmetros do operador de mutação ao longo das gerações. Embora todos os indivíduos possuam as mesmas taxas e parâmetros inicialmente, eles podem variar e serem customizados para cada nó de cada indivíduo sob influência do seu desempenho. Por exemplo, se logo após uma operação de mutação ainda existirem obstruções no trecho entre dois nós adjacentes, ambos terão sua taxa de mutação e *MaxRaioNo* incrementados. Caso não haja obstruções entre nós, os valores dos parâmetros específicos daqueles nós são reduzidos até, no mínimo, seus valores originais. Esta estratégia, empregada apenas no módulo de planejamento global, tem o objetivo de concentrar a mutação em nós específicos associados a obstruções, assim permitindo que eles possam ser deslocados mais extensivamente.

Finalmente, após gerar os indivíduos descendentes, estes substituem seus genitores, formando uma nova geração. Um detalhe importante é que o melhor indivíduo de uma geração é sempre mantido na geração subsequente, estratégia esta conhecida como elitismo.

6.4.7. Reinício da Evolução

Algumas vezes durante a evolução, a população pode sofrer uma convergência prematura para um mínimo local, ficando estagnada. Uma vez que a evolução está, na verdade, fazendo planejamento de trajetórias on-line, cujo resultado deve ser obtido em tempo real, é temerário esperar por muito tempo para que o processo escape da estagnação por si só. Neste contexto, sempre que este tipo de situação é observado (fitness estável do melhor indivíduo) por mais do que *MaxGeracoesEstavel*, o processo evolutivo global é reiniciado, tomando como trajetória base o melhor indivíduo encontrado. Este artifício não se aplica ao módulo de planejamento local.

Antes de efetivamente reiniciar a evolução, o melhor indivíduo passa por duas operações: remoção de nós ineficientes e inserção de novos nós. Em resumo, a primeira operação é responsável por encontrar atalhos na própria trajetória, testando todas as possíveis conexões diretas entre dois nós e removendo aqueles cujos segmentos intermediários são inúteis. A segunda operação, por outro lado, procura por obstruções na trajetória e insere nós no seu exato ponto de intersecção.

Após o eventual encurtamento da trajetória pela eliminação de nós inúteis e o aumento de sua flexibilidade pela introdução de novos nós, o indivíduo está pronto para ser usado como semente no processo evolutivo sendo reiniciado. Semelhantemente ao que é feito na inicialização (Seção 6.4.3), uma população completamente nova é criada com base na trajetória referencial. Agora, a evolução pode continuar seguindo os mesmos passos que vêm sendo descritos.

6.4.8. Critérios de Parada

Há dois critérios de parada em efeito nesta abordagem evolutiva. O primeiro, e mais importante, indica quando uma trajetória satisfatória é encontrada e não há mais necessidade de melhora. Este critério funciona da seguinte forma: se o melhor indivíduo da população representando uma trajetória limpa (sem obstruções) surge e seu fitness permanece estável por $(MaxGeracoesEstavel \div 2)$ gerações, então a evolução é interrompida. A busca não pára imediatamente após uma trajetória limpa qualquer ser encontrada, visto que ainda pode ser possível minimizar seu comprimento. Ao final, o método de remoção de nós, explicado na Seção anterior, é aplicado à trajetória vencedora visando um último refinamento. A seguir, a trajetória está pronta para ser navegada pelo robô.

O segundo critério é uma salvaguarda que evita que a evolução seja executada para sempre: quando um certo número máximo de gerações (*MaxGeracoes*) é atingido, a evolução pára e a melhor trajetória é apresentada como resultado final. É claro que, quando há a satisfação deste critério, não há garantias de que a trajetória final seja livre de obstruções. De qualquer forma, há ainda chances de que o módulo de planejamento local encontre, com base nos estímulos sensoriais, caminhos alternativos. Portanto, mesmo que a trajetória global não seja totalmente limpa, enquanto o robô estiver navegando, segmentos da trajetória estarão sendo localmente reavaliados, em especial quando o robô se confrontar com obstáculos.

Note que a quantidade de vezes que a evolução é reiniciada não influi neste critério de parada, dado que o contador de gerações não é zerado durante o planejamento de uma trajetória.

6.4.9. Movimentos do Alvo

Uma vez que a dinâmica do alvo é desconhecida, ele pode se mover a qualquer momento e para qualquer lugar, independentemente do fato de o sistema de navegação estar em processo de evolução ou não. Graças à forma como o sistema é arquitetado, isto não é um problema. Caso o robô esteja parado e não evoluindo trajetórias no momento que as novas coordenadas do alvo são recebidas, é simplesmente uma questão de dar início à evolução global.

O outro caso a se considerar, quando a evolução está em atividade, é mais desafiador porque as boas soluções atuais podem, de repente, se tornar não tão boas ou mesmo inviáveis. Entretanto, lembre-se de que a posição do alvo é comum a todos os indivíduos da população e, conseqüentemente, a evolução não precisa nem sequer ser interrompida. De fato, os algoritmos evolutivos são naturalmente capazes de se readaptar e procurar por soluções alternativas. É obvio que, dependendo de quão drástico é o deslocamento do alvo, o sistema pode levar mais ou menos tempo para recuperar um bom desempenho.

6.4.10. Controle de Navegação

A partir do instante em que a trajetória está definida, o módulo de navegação é responsável por mover o robô de nó em nó da trajetória local. Existem duas variáveis de controle: velocidade e velocidade angular. Enquanto a primeira move o robô para frente ou para trás, a última gira o robô.

Rotações são necessárias quando o robô está sobre um nó e deve ajustar seu ângulo em direção ao próximo nó.

O controle de navegação é de baixo nível, no sentido de que atua diretamente nos motores do robô, fazendo-o apenas seguir os trechos indicados pela trajetória local. Este módulo é, até certo ponto, cego, porque não utiliza diretamente as informações sensoriais do robô. Isto não é um problema, pois o módulo de planejamento local pré-processa todos os estímulos externos, sintetizando-os na trajetória local, a qual contém todas as informações necessárias ao módulo de navegação de baixo nível.

A Figura 6.4 ilustra o processo de planejamento e execução de trajetórias por meio de um exemplo simples, mostrando o resultado da integração de cada parte do sistema. No exemplo, uma trajetória global é planejada da origem ao destino, contendo três nós intermediários. Isto se dá enquanto o robô se encontra no ponto de origem. A seguir, trajetórias locais passam a ser evoluídas com base na global e o robô começa a navegá-las. O caminho efetivamente navegado pelo robô pode ser diferente do planejado, como acontece no exemplo da figura. A trajetória local é planejada ao mesmo tempo em que o robô navega, utilizando as leituras sensoriais capturadas pelo robô, e, portanto, sua extensão se restringe ao alcance máximo dos sensores (ilustrado por uma semi-elipse na Figura 6.4). Há também três nós intermediários na trajetória local da ilustração, significando que segmentos parciais da trajetória global podem ser minuciosamente ajustados conforme a necessidade. Finalmente, caso o alvo de destino se mova, o trecho global ainda não navegado pelo robô é replanejado.

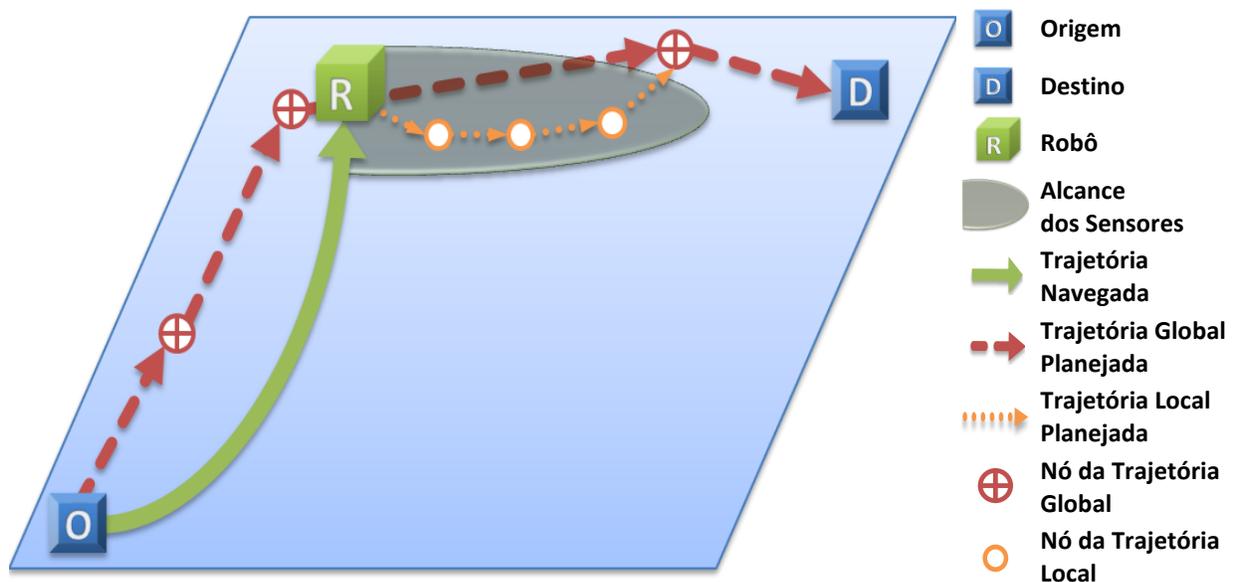


Figura 6.4: Ilustração que exemplifica como se dá o planejamento e navegação de trajetórias pelo robô.

6.4.11. Parâmetros

Os parâmetros do sistema foram determinados empiricamente por meio de diversos testes e simulações e estão apresentados na Tabela 6.1.

Tabela 6.1: Parâmetros do sistema evolutivo.

Parâmetro	Planejamento Global	Planejamento Local
Tamanho da população	20	10
Taxa de cruzamento	10%	10%
Taxa de mutação	20%	20%
MaxGeracoes	2 mil	300
MaxGeracoesEstavel	200	30
MaxRaioNo	4 mil	100

6.5. Experimentos e Resultados

Diversos experimentos foram feitos durante o desenvolvimento deste trabalho, buscando avaliar e validar a proposta. Eles serão apresentados nas seções que seguem, mostrando o sistema em operação. Os testes serão organizados em quatro níveis crescentes de complexidade: cenários estáticos, cenários dinâmicos, estudo de casos e testes reais

Os experimentos simulados foram feitos usando a planta do pavimento térreo de um hospital real. Também as trajetórias dos alvos (pessoas) utilizados foram obtidas a partir do próprio sistema de localização RFID. O computador utilizado foi um Intel Core Duo de 2 GHz e 2 GB de memória RAM.

Nas figuras apresentadas durante esta seção, o robô é representado por um círculo azul; o alvo é mostrado como um retângulo roxo; as trajetórias planejadas pelo sistema evolutivo são indicadas por três linhas paralelas; a trajetória efetivamente executada pelo robô aparece como uma seqüência de círculos; e, finalmente, os obstáculos são as linhas pretas.

6.5.1. Cenários Estáticos

Muito embora o principal objetivo deste trabalho seja o rastreamento de pessoas, avaliar o sistema de navegação em ambientes estáticos é importante como primeira fase de testes. Dois casos foram examinados em duas áreas diferentes do hospital. No primeiro, apresentado à esquerda da Figura 6.5, o sistema evolutivo teve que encontrar a trajetória global em uma região contendo 646 obstáculos (linhas). Em 10 simulações, o comprimento da rota mais curta evoluída (mostrada à esquerda da Figura 6.5) foi de 46,82 metros. Em média, o número de gerações necessárias até a conversão para o caminho final foi de 520,33, associado a um tempo médio gasto de 7,7 segundos

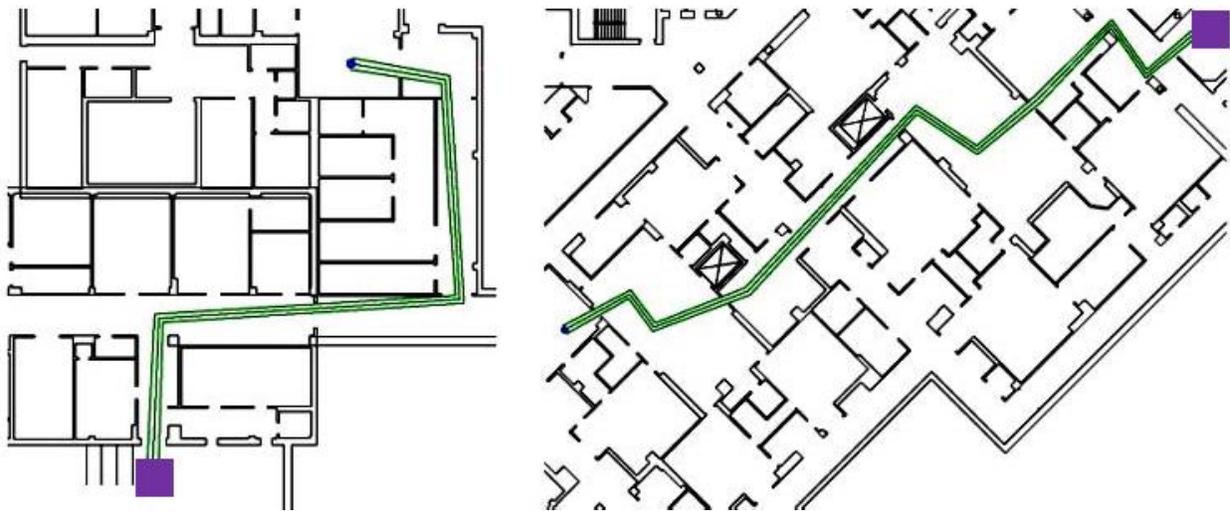


Figura 6.5: Evolução de trajetórias globais em dois cenários estáticos em um hospital.

O segundo experimento ocorreu em uma área maior, que possuía por volta de 1150 obstáculos. A maior complexidade na busca por trajetórias factíveis se refletiu nas seguintes estatísticas obtidas após 10 simulações. Levou em torno de 1328 gerações, em média, para convergir, o que representou uma demora média de 42,37 s. A Figura 6.5, à direita, mostra a melhor solução encontrada, cujo comprimento total foi de 75,19 m.

Observe que, apesar de as trajetórias evoluídas não serem suaves, isto não se caracteriza como um problema, pois o robô é capaz de girar sobre o próprio eixo. Além disso, os trechos das trajetórias que se encontram próximos dos obstáculos não representam risco de colisão para o robô devido ao fato de as margens paralelas garantirem uma distância segura.

6.5.2. Cenários Dinâmicos

Cenários estáticos são apenas casos teóricos em planejamento de trajetórias, dado que variáveis desconhecidas estão massivamente presente em problemas reais, especialmente em um hospital. Nesta seção, dois experimentos expõe o sistema de navegação a ambiente dinâmicos, que contêm obstáculos imprevistos e alvos móveis.

6.5.2.1. Desviando de Obstáculos Imprevistos

Obstáculos imprevistos são objetos desconhecidos *a priori* (i.e., obstáculos que não fazem parte do mapa global do ambiente) que são detectados apenas quando entram no campo de ação dos sensores do robô. Estes obstáculos podem causar problemas para o robô se ele não for capaz de reagir adequadamente quando eles surgirem. Pessoas andando e móveis são exemplos típicos.

Neste experimento, enquanto o robô estava navegando por uma trajetória recém evoluída, sete novos obstáculos foram surgindo pelo caminho à frente do robô, obstruindo sua passagem. Como eles aparecem abruptamente na frente do robô, é necessária uma resposta rápida para evitar a colisão. A Figura 6.6 mostra o cenário do experimento, sendo os obstáculos imprevistos representados pelas linhas que atravessam a trajetória planejada.

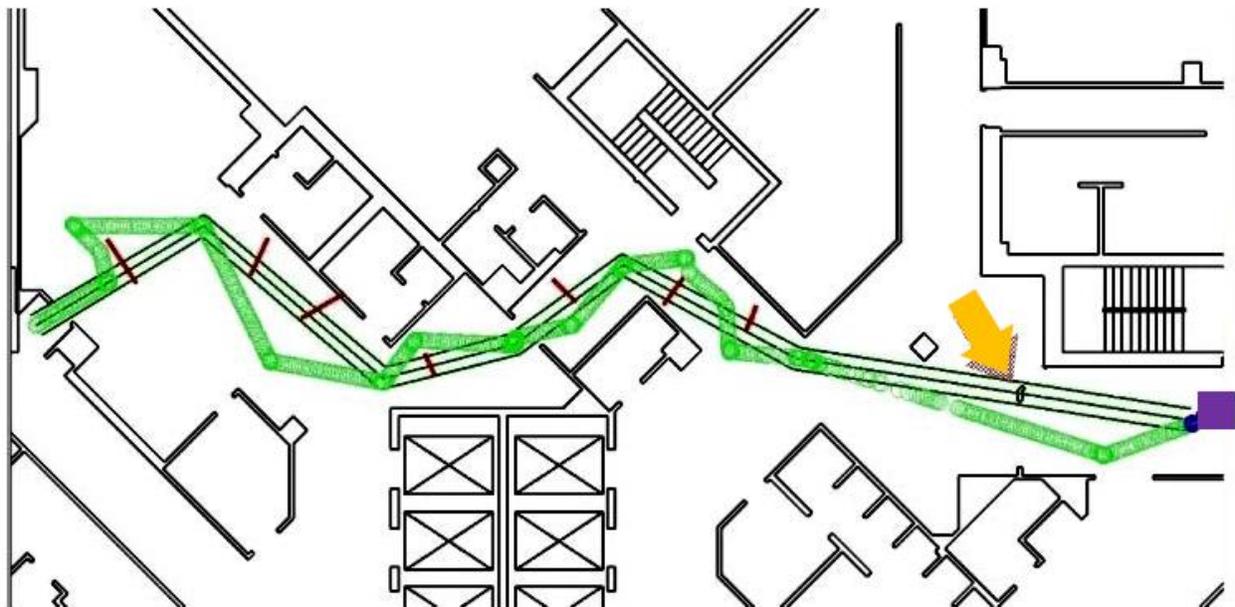


Figura 6.6: Experimento em ambiente dinâmico em que o robô desviou de obstáculos imprevistos (linhas que atravessam a trajetória planejada).

A trajetória global (linhas verdes paralelas) foi evoluída em 9,57 s e 893 gerações, medindo 40,33 m. Note que o robô teve que navegar fora da trajetória originalmente planejada para desviar dos obstáculos imprevistos. Embora a trajetória efetiva (seqüência de círculos) realizada pelo robô tenha sido 7,63 m mais longa que a original planejada, o módulo de planejamento local fez um bom trabalho encontrando rotas alternativas em resposta à presença de obstáculos que não eram esperados.

Além da importância de apresentar capacidade de desvio de obstáculos não presentes no mapa, é interessante investigar a questão da agilidade do sistema em tais situações. Para isso, o tempo gasto pelo sistema para produzir os desvios de rota foi computado toda vez que eles se fizeram necessários. Em média, o sistema evoluiu a trajetória local alternativa em apenas 337,4 milissegundos, o que mostra a eficiência do sistema frente a problemas de tempo real.

Um evento interessante pode ser percebido. O robô também desviou de uma coluna de concreto (indicada pela flecha) antes de chegar até o alvo. Uma vez que a coluna acabou ocasionalmente bem no meio da trajetória, não sendo interceptada pelas margens de segurança, ela foi ignorada pelo sistema de planejamento global. Entretanto, quando a coluna foi detectada pelos sensores do robô, imediatamente o sistema de planejamento local reagiu, prevenindo a colisão.

6.5.2.2. Rastreando Alvos Móveis

O objetivo deste experimento é mostrar a habilidade do sistema em readaptar a trajetória global em resposta a movimentos feitos pelo alvo sendo rastreado. Logo, os únicos obstáculos são aqueles existentes no mapa do ambiente. Os movimentos do alvo são eventos discretos, do ponto de vista do robô, uma vez que eles são informados pelo servidor do sistema de RFID em intervalos variáveis de tempo. 16 posições do alvo, previamente gravadas enquanto ele se movimentava, foram utilizadas nesta simulação.

A Figura 6.7 apresenta a configuração final do experimento, sendo os locais pelos quais o alvo passou rotulados (1 a 16) e ordenados temporalmente. A trajetória final efetiva do robô é representada pela seqüência de círculos. Note que todas as rotas evoluídas ao longo da navegação do robô são mostradas através das linhas laranja, sendo assim possível perceber como o sistema de navegação adaptou as trajetórias toda vez que o alvo se moveu. Sempre que uma nova posição do alvo é recebida pelo sistema, a evolução da trajetória global é disparada (se ela já não estiver acontecendo) para processar a atualização. Além disso, como o objetivo era ilustrar a capacidade de adaptação do sistema,

a velocidade máxima do robô foi intencionalmente limitada para evitar que o alvo fosse alcançado logo. Isto explica o porquê da maioria dos locais em que o alvo esteve não ter sido visitada pelo robô (1,2, 5, etc.).



Figura 6.7: Experimento em que o robô segue um alvo móvel.

A eficiência ao planejar caminhos atualizados em relação ao estado do ambiente não pode ser medida apenas pela capacidade de encontrar trajetórias factíveis, mas também precisa levar em conta o tempo consumido para fazê-lo. Neste sentido, o tempo foi computado em todos os 16 movimentos do alvo, resultando nas seguintes estatísticas. Em média, o sistema precisou de 1,79 s para evoluir uma trajetória global adaptada, cujo comprimento médio foi 9,6 m. Este resultado permite concluir que o sistema de navegação apresenta, ao menos em simulação, uma rapidez de reação razoável para perseguição de alvos em tempo real.

6.5.2.3. Estudo de Caso: Seguindo Paciente em um Corredor Ocupado

Agora que os temas desvio de obstáculos e rastreamento de alvos foram devidamente estudados em separado, é hora de examiná-los de forma conjunta em um mesmo experimento. Para tanto, uma situação real bastante comum em hospitais foi simulada: o robô tem que acompanhar um paciente por um corredor lotado de pessoas e outros obstáculos. O principal corredor do piso térreo do hospital foi escolhido para o teste, medindo 100 x 2,4 metros.

A Figura 6.8 apresenta seis cenas capturadas durante a simulação e, na parte de baixo, a trajetória completa realizada pelo robô, cujo comprimento total foi de 110,84 metros. Do início ao final do experimento, o sistema de localização RFID reportou 22 movimentos do paciente. Em média, o robô se manteve a uma distância de 6,69 m da pessoa. Esta distância, juntamente com as cenas da Figura 6.8, evidenciam as dificuldades causadas pelos obstáculos imprevisíveis confrontados pelo robô enquanto seguia o alvo.

De fato, a freqüente necessidade de readaptar a trajetória é a principal razão para justificar por que o robô não alcançou o paciente com maior antecedência. O sistema de navegação foi bem-sucedido, evoluindo trajetórias que fizeram com que o robô superasse todos os obstáculos sem colisões. Mesmo em situações mais complicadas, como os casos 2 e 4 da Figura 6.8, em que o corredor estava quase que totalmente bloqueado, o resultado foi positivo. Além do mais, observe que, após desviar dos obstáculos, o sistema de navegação tenta insistentemente retornar o robô à trajetória global planejada, sua referência.

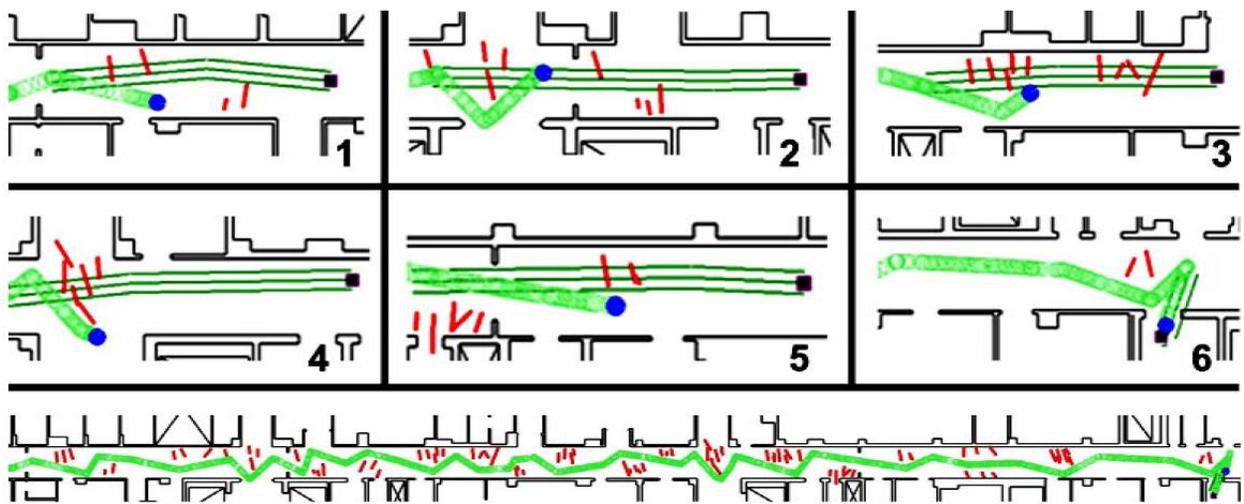


Figura 6.8: Simulação do robô (círculo) seguindo um paciente (retângulo) por um corredor lotado (obstáculos em vermelho) de 100 m de comprimento. As cenas numeradas foram capturadas ao longo do experimento, sendo que a imagem mais abaixo mostra a trajetória completa percorrida pelo robô.

Outra simulação foi executada, desta vez contando com obstáculos móveis. Ao invés de aparecer na frente do robô, como acontecia nos experimentos anteriores, os obstáculos são simulados mais realisticamente, existindo o tempo todo e se movendo de encontro ao robô. Os movimentos do paciente, registrados previamente, foram empregados novamente.

A Figura 6.9 mostra três cenas capturadas em momentos em que o robô desviava dos obstáculos móveis e, em baixo, a trajetória efetiva completa. É interessante que, embora os obstáculos se movendo em direção ao robô tenham aumentado o risco de colisões (e.g. caso 3), o alvo foi alcançado em apenas 30,62 m. Complementando as informações, a distância média do robô ao alvo foi de 3,6 m.

Uma possível justificativa para este comportamento surpreendentemente melhor recai sobre a capacidade do robô de detectar os obstáculos com maior antecedência, ou seja, desde que eles entram na zona de alcance dos sensores. Ao tomar conhecimento dos obstáculos antes, há mais tempo para um melhor planejamento, dado que o sistema, além de saber a posição do alvo, consegue também de certa forma incorporar sua dinâmica no processo de evolução de trajetórias. Portanto, trajetórias mais robustas às mudanças do ambiente são produzidas e executadas. Note que, no caso da Figura 6.8, os obstáculos surgem de repente.

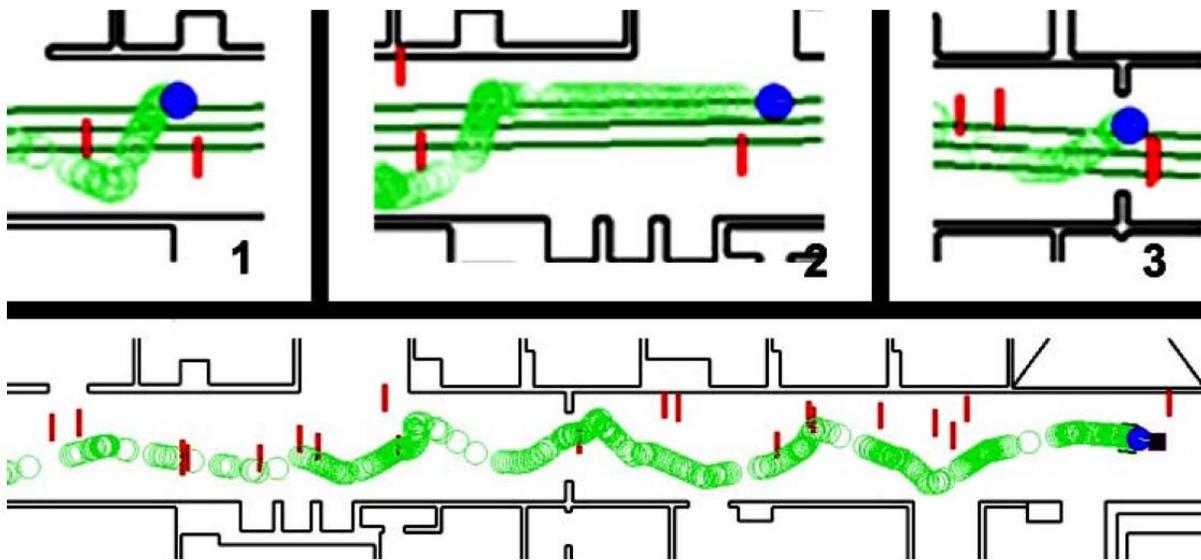


Figura 6.9: Simulação com obstáculos (linhas vermelhas) que se moviam em direção ao robô.

6.5.2.4. Estudo de Caso: Médico Visitando Quartos

O último estudo de caso simula um cenário rotineiro para médicos e enfermeiras: visitas e avaliações periódicas a pacientes internados no hospital. Em geral, o médico deve visitar uma série de pacientes em diversos quartos para checar o estado de saúde do indivíduo e atualizar estas informações no sistema de informações clínicas. O robô serve como um terminal móvel que dá acesso ao sistema de informações (ele possui um laptop embutido) e também carrega material médico necessário.

Claramente, a tarefa somente é possível se o robô for capaz de rastrear eficientemente o médico por todos os quartos.

O experimento é mostrado na Figura 6.10, sendo possível ver os 15 quartos visitados em seqüência pelo médico, e a sala usada como base pelo robô, onde ele fica estacionado. Propositalmente, as coordenadas do alvo eram enviadas para o robô apenas quando o médico já se encontrava no interior das salas. Desta forma, o problema de planejamento de trajetórias se torna mais complicado.

A Tabela 6.2 exibe os dados obtidos ao longo do experimento, incluindo a distância navegada pelo robô de quarto a quarto e o tempo total gasto em cada caso (considerando a evolução da trajetória e sua execução). Analisando a tabela, conclui-se que o médico teve que esperar, em média, 1 minuto e 12 segundos até que o robô chegasse. Visto que o médico estaria examinando o paciente enquanto o robô estava navegando até o quarto, este tempo de espera pode ser considerado aceitável.

Tabela 6.2: Dados do experimento da Seção 6.5.2.4, em que o robô segue um médico por 15 quartos de pacientes.

Quarto #	Comprimento da Trajetória (m)	Tempo Gasto (s)	Quarto #	Comprimento da Trajetória (m)	Tempo Gasto (s)
1	23	231	9	5,48	22
2	9,26	72	10	13,26	72
3	6,83	59	11	9,03	66
4	9,91	74	12	8,14	43
5	9,30	51	13	5,11	37
6	14,80	91	14	10,06	64
7	8,51	65	15	12,54	81
8	6,82	54	Média	10,14	72,13

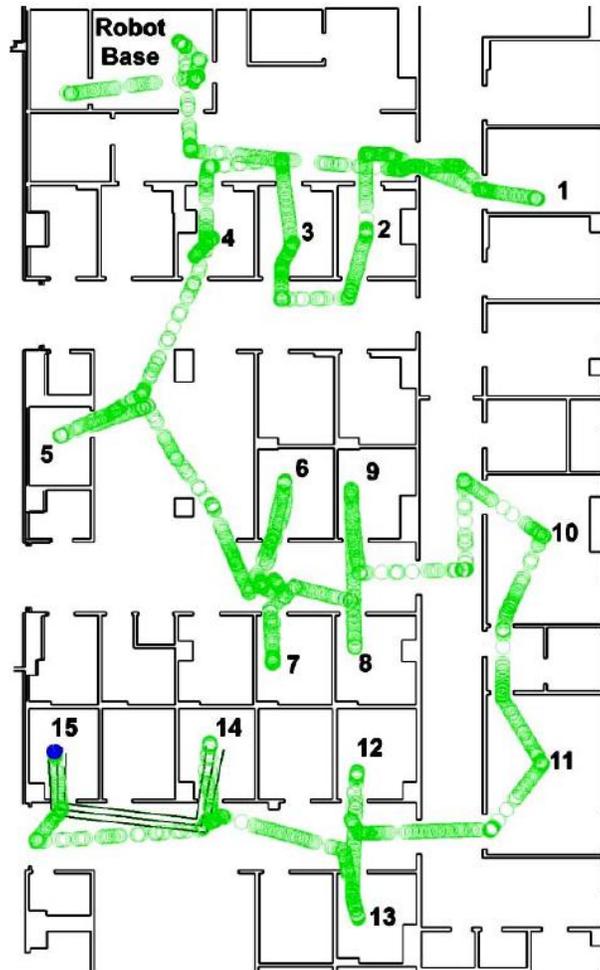


Figura 6.10: Simulação do robô rastreando um médico (não mostrado) por 16 quartos.

6.5.3. Testes Reais

O sistema proposto não foi apenas testado em simulação computacional; um extensivo conjunto de experimentos reais foi realizado utilizando o robô PatrolBot. Os testes foram feitos na área interna do hospital *Washington Hospital Center* em Washington D.C., EUA, especialmente no departamento de emergência, por ser o ambiente mais dinâmico do hospital.

Os experimentos reais são bem mais curtos e com tarefas mais simples que as simulações, devido à taxa de imprecisão do sistema de localização por rádio frequência, na faixa de dois metros em média. Tamaña quantidade de ruído torna difícil planejar trajetórias muito longas e, por isso, o rastreamento fica prejudicado. A empresa prestadora deste serviço se comprometeu a melhorar seus equipamentos para aumentar a precisão em um futuro próximo. Nos experimentos reais, o robô persegue uma pessoa que carrega consigo um dispositivo RFID (*tag*).

6.5.3.1. Primeiro Experimento

Este teste foi realizado em uma sala pequena, sendo que a pessoa se move para uma posição e espera até que o robô chegue lá para então se encaminhar até o próximo local. O experimento completo foi capturado no Vídeo 6.1, a partir do qual foi extraída a seqüência de cenas apresentada na Figura 6.11.

Na Figura 6.11 pode-se notar que o robô alcança a pessoa quatro vezes, realizando trajetórias com comprimento médio de 1,8 m. Embora não haja obstáculos entre a origem e o destino, o sistema demonstra ser capaz de seguir a pessoa com rapidez e boa precisão.



Figura 6.11: Robô seguindo uma pessoa em uma sala. A pessoa apenas se move quando o robô a alcança (Vídeo 6.1).

6.5.3.2. Segundo Experimento

O segundo experimento exige que o robô siga a pessoa (no caso o próprio *cameraman*) por um corredor estreito e distâncias maiores. Testa-se também a capacidade de desvio de obstáculos do sistema. A Figura 6.12 exibe uma seqüência de imagens capturadas durante este experimento, o qual pode ser visto por inteiro no Vídeo 4.2.

O robô inicia a perseguição no meio do corredor e vai até a pessoa com a câmera que está no final do corredor (cenas 1 e 2 da Figura 6.12). A seguir, a pessoa passa pelo robô no sentido oposto (a traseira do robô aparece na 3ª cena) e, enquanto caminha até o meio do corredor, colocam-se dois obstáculos que bloqueiam parcialmente o corredor. Imediatamente após girar e começar a navegar atrás da pessoa, trajetórias locais que levam o robô a desviar das latas de lixo são evoluídas, permitindo que ele

supere os obstáculos que estavam ausentes anteriormente (cenas 4 e 6). Adiante, a pessoa continua andando em direção ao extremo oposto do corredor, ao mesmo tempo em que um terceiro obstáculo é inserido e, novamente, o robô reage bem conseguindo desviar (cenas 7 e 8). Finalmente o robô alcança o destino final, chegando perto do câmera que está parado no final do corredor (última cena).

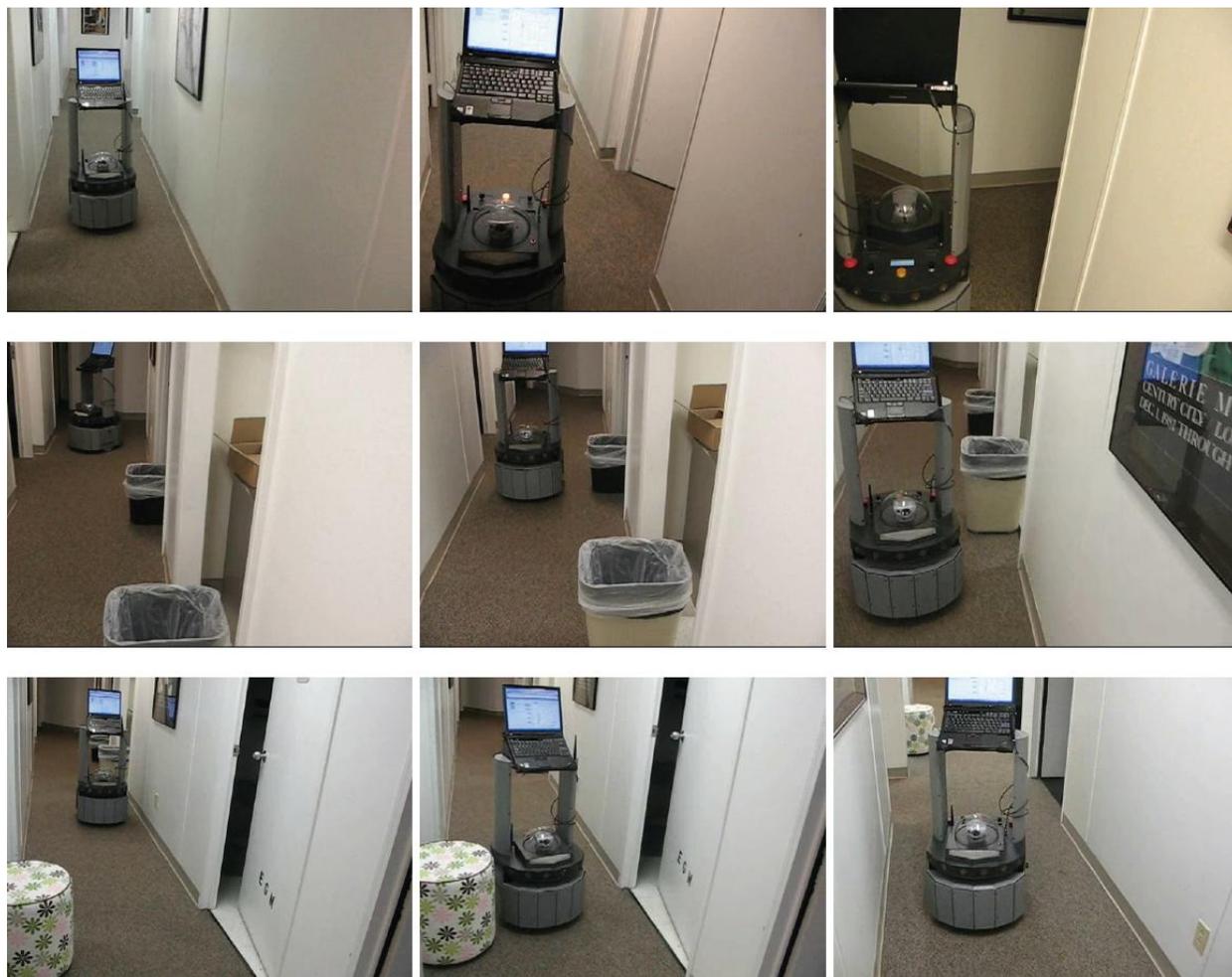


Figura 6.12: Experimento em que o robô rastreia o câmera por um corredor estreito, desviando de obstáculos (Vídeo 6.2).

6.5.3.3. Terceiro Experimento

Enquanto os dois experimentos reais anteriores foram feitos de maneira mais controlada, por exemplo com a introdução gradual de obstáculos, este foi executado nos corredores da sala de emergência do hospital em plena operação. O experimento teve como objetivo simplesmente fazer o robô seguir a pessoa com a *tag* por um corredor de aproximadamente 20 metros de comprimento, sempre no mesmo sentido.

A Figura 6.13, que é uma compilação do Vídeo 6.3 que registra o experimento, mostra como o teste se deu. Em resumo, o robô foi bem-sucedido na tarefa de rastrear a *tag* RFID. Um detalhe capturado na 2ª cena da seqüência ilustra bem como a dinâmica do ambiente traz complicações ao problema de navegação autônoma: ao sair de uma sala, um médico não percebeu a presença do robô, o que quase levou à colisão. Não fosse a agilidade do médico e a imediata reação do robô, parando de se movimentar, um acidente poderia ter ocorrido.



Figura 6.13: A pessoa carregando a tag é seguida continuamente pelo robô por um corredor de 20 metros (Vídeo 6.3).

6.5.3.4. Quarto Experimento

Este experimento é o mais complexo desta série, pois foi executado em meio a um período de grande movimentação de pessoas, macas e equipamentos nas cercanias da sala de emergência. Note, mais uma vez, que não houve nenhuma restrição ou orientação especial às pessoas presentes naquela área; a idéia era justamente testar o robô sob as verdadeiras condições daquele ambiente. Assim como no experimento anterior, o robô teve que rastrear a pessoa com a *tag* mas, desta feita a perseguição aconteceu com variações de direção, sentido e continuidade (em alguns momentos, a pessoa está andando e, em outros, parada).

O teste foi gravado no Vídeo 6.4, do qual se produziu a seqüência de imagens que compõe a Figura 6.14. Analisando a figura observa-se que, apesar da existência de diversos obstáculos (algo que, por sinal, dificulta a visualização de partes do experimento), o robô é capaz de seguir a *tag* com eficiência, indo e voltando no corredor sem provocar nenhuma colisão.



Figura 6.14: Experimento em que o robô faz rastreamento em meio a diversas pessoas e obstáculos no hospital (Vídeo 6.4).

É evidente que, em algumas situações, há um atraso no planejamento da trajetória (por exemplo, na cena 8 da Figura 6.14, o sistema leva em torno de 14 s) causado pela maior complexidade daquele cenário. Mesmo assim, considera-se melhor que o sistema demore um pouco mais para navegar, mas o faça com maior segurança.

Uma constatação importante, observada não apenas neste experimento mas também em outros, é que a pessoa sendo rastreada, quando dentro do alcance dos sensores do robô, é considerada como um obstáculo pelo sistema de navegação. Isto causa problemas em certas situações, muitas vezes levando o robô a replanejar sua trajetória desnecessariamente. Tipicamente, quando isto acontece, observa-se no robô um comportamento de girar no mesmo lugar ou até navegar brevemente no sentido contrário do alvo (veja o Vídeo 6.4 no instante 1min 50s). Esta questão deverá ser resolvida futuramente.

6.6. Conclusão

Este capítulo contribui com uma solução para um problema desafiador em robótica: rastreamento de alvos móveis em ambientes dinâmicos. A proposta consiste de um sistema de navegação evolutivo capaz de realizar planejamento global e local de trajetórias, além de controle autônomo do robô. O sistema foi arquitetado de forma a integrar informação global disponível *a priori* (mapa do ambiente) com estímulos locais capturados pelos sensores do robô, de modo a lidar com eventos não previsíveis.

Este sistema de navegação foi testado em um hospital, permitindo que o robô siga pessoas, sejam eles médicos, enfermeiros ou pacientes, e assista-os em variadas tarefas. Embora, à primeira vista possa parecer que robótica móvel não tem muito a ver com medicina, este trabalho mostra, no papel de um precursor, que existem aplicações viáveis que podem ser implementadas em hospitais em um futuro próximo, colocando os robôs para ajudarem aqueles que ajudam pessoas em ambientes de saúde.

Os experimentos discutidos neste capítulo foram realizados em um hospital real, a saber *Washington Hospital Center*, usando um sistema de localização por rádio frequência que permite saber a localização de pessoas e objetos. Os resultados obtidos mostram que o sistema de controle proposto foi bem-sucedido na tarefa de definir e implementar trajetórias em ambos os tipos de ambiente: estáticos e dinâmicos. Especialmente no último, o sistema mostrou ser adaptativo e rápido em face a obstáculos inesperados e alvos móveis.

Futuramente, pretende-se trabalhar na extensão do sistema, buscando tornar o planejamento de trajetórias mais rápido e robusto em relação a longas distâncias. Há planos também de integrar o processo de chamada do robô pelas pessoas através de celulares, pagers ou mesmo Internet. Outro passo adiante envolve o uso de informações adicionais, realizando a fusão de diversos sensores, como por exemplo a utilização de reconhecimento de faces para a tarefa de rastreamento (Chen & Birchfield, 2006).

6.7. Descrição dos Vídeos

Vídeo 6.1: Teste realizado em uma sala pequena, sendo que a pessoa se move para uma posição e espera até que o robô chegue lá para então se encaminhar até o próximo local. O robô alcança a pessoa quatro vezes.

Vídeo 6.2: O experimento exige que o robô siga a pessoa (no caso o próprio *cameraman*) por um corredor estreito. Testa-se a capacidade de desvio de obstáculos do sistema, já que vários obstáculos são colocados de forma surpresa no caminho do robô.

Vídeo 6.3: O experimento teve como objetivo de fazer o robô seguir a pessoa com a *tag* por um corredor de aproximadamente 20 metros de comprimento. Um detalhe que ilustra bem como a dinâmica do ambiente traz complicações ao problema de navegação autônoma pode ser visto: ao sair de uma sala, um médico não percebeu a presença do robô, o que quase levou à colisão.

Vídeo 6.4: Experimento executado em meio a uma grande movimentação de pessoas, macas e equipamentos nas cercanias da sala de emergência. O robô teve que rastrear a pessoa com a *tag*, a qual, em alguns está andando e, em outros, parada.

Capítulo 7: Plataforma para Inspeção de Ambientes e Coleta de Dados

7.1. Introdução

Planejamento consiste na determinação de um conjunto de procedimentos e ações visando a realização de determinado projeto ou objetivo. Quando a finalidade é obter um caminho factível de um local a outro, trata-se de planejamento de trajetórias, o qual foi o focado em detalhes no capítulo anterior. Mais do que apenas trajetos, o planejamento de muitos outros tipos de tarefas, contemplando variados graus de complexidade, é algo que permeia o dia-a-dia dos seres humanos em diversas atividades.

Planos que possuam múltiplas etapas a serem executadas, as quais podem estar associadas a outros sub-planos e tarefas, são conhecidos, especialmente na área de robótica, como missões. O planejamento de missões é um problema extremamente desafiador, uma vez que envolve a determinação de uma seqüência de ações que dependem, e muito, de eventos futuros. A idéia é, literalmente, tentar antecipar a “Lei de Murphy” de modo a descobrir o que pode, ou não, dar errado antes de tomar qualquer decisão. Idealmente, seria necessário procurar por todo o espaço de possíveis ações até encontrar uma seqüência de decisões futuras que garantiriam a maior compensação. Sendo isto claramente inviável, é necessária a utilização de heurísticas que sejam capazes de encontrar atalhos em direção a soluções mais promissoras (Capozzi, 2001).

O ato de planejar é, sem dúvida, deliberativo, já que tem uma intrínseca relação com a previsão de passos futuros. Para tanto, o controlador de um robô depende fortemente de conhecimentos a respeito do mundo externo em que atua: quanto mais precisas as informações, maior a chance de que as previsões estejam corretas e o plano seja realizado com sucesso. O mapa do ambiente e sua dinâmica são dois dos principais conhecimentos utilizados no planejamento de missões com robôs móveis.

Existem vários trabalhos que propõem planejadores de missões com robôs, sendo válido citar alguns. Deutsch et al. (1985) desenvolveram um sistema que utiliza a tradicional técnica de busca A* (Nilsson, 1986) para obtenção de trajetórias e um tipo de algoritmo de recozimento simulado que cuida dos objetivos gerais da missão, diferenciando claramente planos de curto e longo prazo. Visando transformar os objetivos de alto nível das missões em uma agenda de eventos discretos a serem postos

em prática, Chien (2000) concebeu um sistema bastante completo que usa diversas técnicas de IA, possuindo uma grande flexibilidade ao permitir re-planejamento em tempo real do plano de execução.

Uma modalidade ainda mais complexa de planejamento de missões é aquela que envolve a execução de tarefas por múltiplos robôs. Embora este tema não seja tratado neste capítulo, alguns trabalhos investem nesta frente. Em Chandler et al. (2000), por exemplo, os autores se dedicam a um problema de coordenação de veículos aéreos autônomos com a missão de atingir um alvo minimizando sua exposição aos radares. Zhang & Kim (1997), por sua vez, apresentam um planejador evolutivo visando aprendizagem em futebol de robôs.

De acordo com o levantamento feito por Capozzi (2001), boa parte dos planejadores encontrados na literatura obtém bons resultados quando construindo trajetórias. Os mesmos planejadores, no entanto, tendem a falhar seriamente quando adaptados para planejamento de missões. Segundo ele, a exceção são as abordagens bio-inspiradas, cujas características demonstram ser mais apropriadas para tal nível de dificuldade. Adotando princípios de computação evolutiva, também empregados no capítulo anterior, propõe-se, neste capítulo, um sistema de controle capaz de realizar ambos, planejamento de trajetória e missões, no contexto da aplicação detalhada a seguir.

O problema específico tratado neste capítulo consiste na inspeção sistemática de um ambiente por um robô autônomo (PatrolBot) com o intuito de coletar dados úteis a análises posteriores. Neste processo, o robô precisa visitar diversas regiões do ambiente, navegando entre elas e executando tarefas relacionadas à obtenção dos dados requisitados. Para tanto, o sistema de controle deve determinar os locais que o robô visitará, em que seqüência e quais trajetórias fará entre eles; estes elementos são, no problema tratado aqui, os fatores que constituem as missões a serem planejadas.

Este trabalho dá prosseguimento à proposta do capítulo anterior, empregando o robô inspetor para coletar dados das redes sem fio do Washington Hospital Center. Embora o robô tenha à sua disposição o mapa do hospital, a movimentação constante de pessoas e equipamentos exige que o sistema de controle seja capaz de adaptação e respostas reativas. Logo, o planejamento de missões e trajetórias precisa ser flexível e não deve levar em conta apenas o mapa do ambiente, mas também os estímulos locais capturados pelos sensores do robô.

Tendo em vista os desafios supracitados, é concebido um sistema autônomo que combina planejamento de longo prazo e adaptação a eventos recentes por meio de algoritmos evolutivos. Da

mesma forma que o sistema evolutivo do Capítulo 6, empregado nesta proposta como módulo de planejamento de trajetórias, este também confia na habilidade dos mecanismos evolutivos de encontrar soluções eficazes e diversas para a realização de planejamento de missões. Os detalhes do sistema, assim como os testes realizados, são apresentados ao longo das próximas seções.

7.2. Contextualização do Capítulo

Continuando a seqüência progressiva de frentes de pesquisa, apresenta-se agora aquela que pode ser considerada a com maior grau de deliberação dentre todas. O foco deste capítulo está fortemente direcionado à questão de planejamento. Não apenas planejamento de trajetórias, mas também planejamento de missões compostas por diversas atividades. O objetivo é desenvolver uma abordagem que se aproxime um pouco mais de uma habilidade que é inerente e fundamental a seres com alto nível cognitivo: proposição de planos.

Ao se notar evolução das linhas de pesquisa conforme a ordem em que têm sido colocadas, percebe-se que há uma progressão iniciando com abordagens mais reativas e concluindo com esta que se caracteriza por atuar de forma mais deliberativa (não puramente). A proposta deste capítulo é, de fato, um sistema híbrido que combina reatividade e deliberatividade. O sistema faz planejamento de missões, as quais são constituídas, dentre outras coisas, por trajetórias. Internamente, as etapas de planejamento de trajetórias são executadas exatamente pelo sistema proposto no capítulo anterior, do qual esta abordagem herda seu único componente reativo.

Existem, portanto, duas camadas deliberativas e uma reativa. O módulo de planejamento de trajetórias globais, que guia o módulo de trajetórias locais, será, por sua vez, orientado pelo módulo de planejamento de missões. Os dois módulos orientadores são totalmente dependentes de conhecimento inato: o mapa do ambiente. Esta forte dependência deixa evidente a importância da hibridização com componentes reativos. Isto é, mesmo que partes da missão planejada se tornem ineficazes por causa de modificações imprevistas no ambiente (em relação ao mapa original), o componente reativo flexibilizará a execução, adaptando a missão conforme as novas condições detectadas.

Da mesma forma que no capítulo anterior, esta abordagem também possui aprendizagem por evolução, sendo que esta é não cumulativa (i.e., eventos diferentes de evolução de missões ou trajetórias são independentes uns dos outros). Para que fiquem claras as diferenças entre formas de aprendizado, compare esta proposta com as três primeiras do trabalho (Capítulos 3, 4 e 5). Nelas, o

controlador aprende comportamentos duradouros de forma incremental; já aqui (e também no Capítulo 6) há o aprendizado de soluções para cenários específicos, as quais são consumidas e então descartadas.

Novamente, a aprendizagem evolutiva é intrinsecamente otimizadora. Além da otimização de trajetórias herdada da metodologia do capítulo anterior, o mecanismo de planejamento de missões também busca otimizar diversos critérios pertinentes, os quais serão descritos em detalhes nas próximas seções.

Embora o estudo e as idéias voltadas à proposição de planos tenham o objetivo de serem genéricos, o resultado da implementação é aplicado a um problema definido: coleta automatizada de dados pelo robô autônomo. Tanto simulações computacionais quanto experimentos reais são executados. Emprega-se o robô PatrolBot nesta frente de pesquisa e o ambiente real de testes também se encontra no Washington Hospital Center.

7.3. Sistema de Inspeção e Coleta de Dados

7.3.1. Visão Geral

O sistema está organizado em três camadas: planejamento da missão, planejamento da trajetória e execução. As camadas coincidem com as etapas do processo de inspeção e coleta de dados. Adicionalmente, após a execução do processo, uma etapa de análise e interpretação dos resultados é realizada. Cada camada é constituída, basicamente, por módulos que formam uma cadeia progressiva de sub-processos. O planejamento da missão é dividido em distribuição de pontos de coleta e evolução do itinerário de inspeção. A seguir, a trajetória global é evoluída na camada responsável por seu planejamento. A última camada, de execução, é ativada quando o robô está em operação, fazendo a evolução da trajetória local, navegação e coleta dos dados de interesse. O diagrama da Figura 7.1 apresenta a arquitetura da plataforma aqui proposta.

A primeira camada e seus módulos são o foco deste capítulo, sendo responsáveis pelo planejamento da missão de coleta, o que é feito por meio de algoritmos evolutivos. As camadas de planejamento da trajetória e execução são, essencialmente, reaproveitadas do sistema evolutivo de rastreamento proposto no capítulo anterior e, portanto, não serão discutidas novamente.

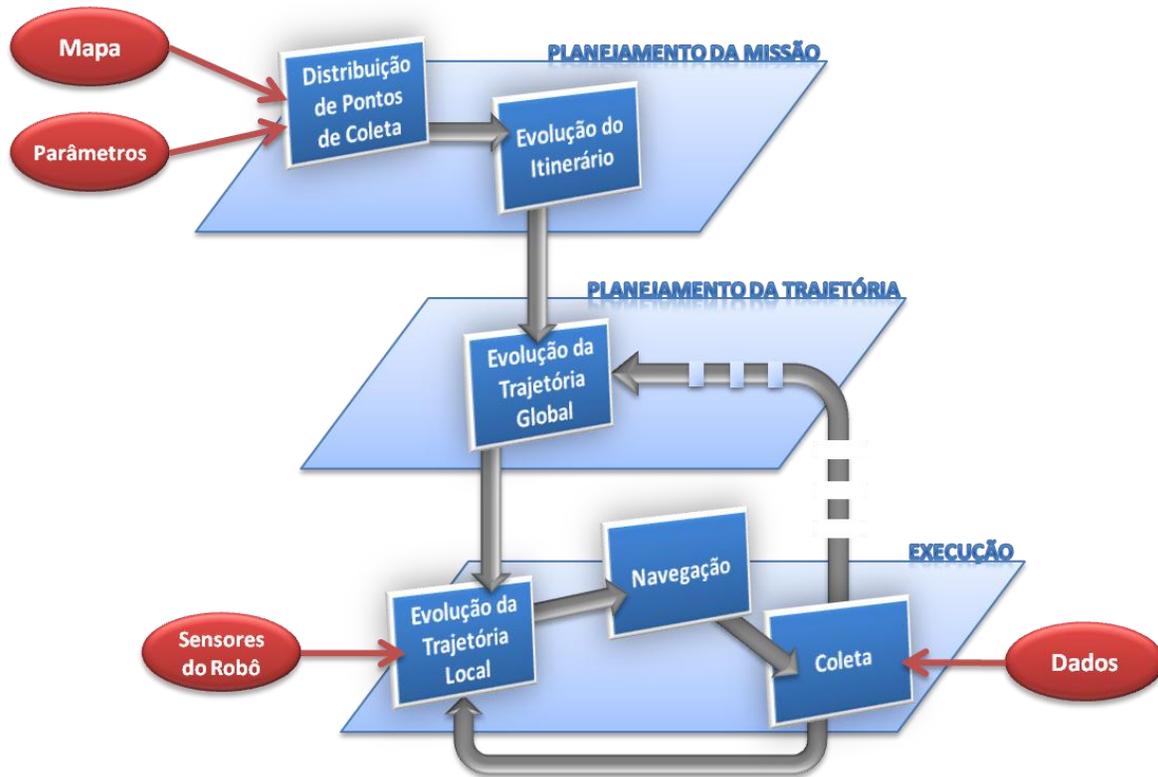


Figura 7.1: Arquitetura da plataforma de inspeção e coleta de dados, organizada em três camadas.

Dada a existência do mapa do ambiente de interesse, conhecido *a priori* pelo robô, pode-se, de forma resumida, delinear a seqüência lógica do processo da seguinte forma: configuram-se os parâmetros básicos da coleta; um algoritmo evolutivo realiza a distribuição dos pontos de coleta, buscando cobrir a área avaliada e evitar locais inacessíveis; outro algoritmo evolutivo busca o melhor itinerário para visitar todos os pontos recém distribuídos; as rotas são então planejadas e executadas pelo robô, sendo os dados coletados; finalmente, analisam-se os dados de acordo com os objetivos desejados. Veja nas seções a seguir o detalhamento de cada parte do processo.

Como a informação de interesse pode ser diversa, e.g, som, imagem ou ondas eletro-magnéticas, o processo de coleta pode usar variados tipos de sensores acoplados ao robô. Com isso, a plataforma pode ser utilizada em diferentes atividades de inspeção. Logo, ao longo da descrição do sistema, as referências feitas aos dados coletados serão genéricas.

7.3.2. Configuração

Esta é a primeira etapa do processo e é bastante simples. O usuário que deseja realizar o mapeamento configura o sistema entrando com um parâmetro geométrico e dois numéricos, respectivamente: área específica do mapa a ser coberta, densidade de pontos de coleta e tempo de coleta.

A área em que será feita a coleta de informações é, na verdade, um polígono composto por um conjunto de pontos (vértices) que representa a região do mapa a ser inspecionada. O usuário define a área inserindo vértices por meio da interface do programa, onde é possível ver o mapa graficamente e clicar sobre os pontos desejados para selecionar a área.

A densidade indica a quantidade de pontos de coleta que serão distribuídos na área selecionada. Já o tempo de coleta em cada ponto está diretamente relacionado ao número de amostras obtidas. Isto é, o robô ficará estacionado em cada ponto visitado por um certo período de tempo, registrando os dados relativos àquele local.

Além dos três parâmetros discutidos, pode haver parâmetros adicionais específicos relativos aos sensores utilizados. Suponha, por exemplo, que o robô esteja capturando fotos; uma configuração adicional poderia ser a quantidade de fotos tiradas em cada ponto do mapa ou mesmo algum nível de qualidade das imagens.

7.3.3. Distribuição dos Pontos de Coleta

A distribuição dos pontos a serem visitados pelo robô é feita por um algoritmo genético, tomando a área de cobertura selecionada como espaço de busca. O algoritmo evolui o conjunto completo de pontos, procurando manter uma densidade desejada e, ao mesmo tempo, evitar locais inacessíveis, onde há muitos obstáculos por exemplo. Ou seja, a idéia é usar uma abordagem evolutiva para obter uma boa distribuição de pontos que represente adequadamente a área de cobertura, segundo os parâmetros pré-definidos pelo usuário.

7.3.3.1. Codificação e Inicialização

Cada indivíduo da população representa uma solução completa, ou seja, contém todos os pontos de coleta de dados. Em termos de codificação, o cromossomo é representado por uma matriz de pares

ordenados $\langle x, y \rangle$, os quais correspondem aos pontos de coleta e suas respectivas coordenadas. Assim como se pode ver na Figura 7.4, cada elemento da matriz representa um ponto de coleta. Esta estrutura de dados é adotada para facilitar a análise computacional da distância entre um ponto e seus vizinhos e também as operações evolutivas de cruzamento.

A inicialização da população é feita de maneira aleatória a partir de um indivíduo referência ou semente. Neste indivíduo, os pontos são distribuídos uniformemente, respeitando perfeitamente a distância entre eles, a qual é proporcional à densidade desejada. Inicia-se a distribuição no ponto de coordenadas mínimas da área de cobertura (mínimo x e mínimo y entre todos os vértices do polígono que representa a área) e encerra-se ao ultrapassar o ponto máximo da área (máximo x e y entre todos os vértices). Observe que, se a área for diferente de um retângulo, haverá pontos fora da mesma. Embora eles façam parte da matriz, eles não serão utilizados em nenhum momento.

Tendo sido construída a matriz semente, gera-se a população de indivíduos realizando modificações aleatórias em cada ponto dela. Ou seja, cada indivíduo é uma cópia alterada da semente original. As modificações de cada ponto são feitas usando duas variáveis aleatórias: direção (0° a 359°) e distância (0 a $Raio_{Max}$). Note que $Raio_{Max}$ é um valor sempre menor que a distância padrão entre os nós (veja Tabela 7.1), evitando que haja pontos superpostos nos indivíduos da população inicial.

7.3.3.2. Avaliação

A avaliação dos indivíduos se dá por meio da função de fitness representada na Equação 7.1 e Equação 7.2, em que os seguintes componentes são considerados:

1. Número de pontos fora da área de cobertura ($NPtosFora$);
2. Número de pontos que se encontram sobre obstáculos do mapa ($NPtosObst$);
3. Distância mínima média dos pontos ao obstáculo mais próximo ($DistMinMed$);

$$fitness = \begin{cases} NPtosRuins + \frac{1}{DistMinMed} & \text{se } NPtosRuins > 0 \\ \frac{1}{DistMinMed} & , \text{ caso contrário} \end{cases} \quad \text{Equação 7.1}$$

$$NPtosRuins = NPtosFora + NPtosObst \quad \text{Equação 7.2}$$

7.3.3.3. Operadores Evolutivos

Após a avaliação da população, entram em cena os operadores evolutivos responsáveis por modificar indivíduos genitores produzindo descendentes que comporão a próxima geração. Os genitores são selecionados aos pares, cada par gerando dois novos indivíduos. O método de seleção utilizado é o torneio, em que uma certa proporção da população (veja na Tabela 7.1) é escolhida aleatoriamente, e os dois melhores são eleitos genitores.

Cada par de genitores é, em seguida, combinado. O operador de cruzamento uniforme atua realizando a troca de pontos entre os genitores. Por ser uniforme, todos os pontos têm chances iguais de intercâmbio (veja a taxa de cruzamento na Tabela 7.1), não havendo o conceito de ponto de cruzamento ou corte. Em termos de cromossomo, a permuta se dá pela troca dos elementos correspondentes (mesma linha e coluna) da matriz de um genitor com a do outro.

A mutação é o último operador aplicado sobre o cromossomo dos indivíduos, deslocando cada ponto aleatoriamente de acordo com certa probabilidade (veja Tabela 7.1). Este deslocamento toma uma direção aleatória (0° a 359°) e uma distância aleatória (0 a $Raio(i,l,c)_{Max}$) para determinar a nova posição de cada ponto de coleta que vai sofrer mutação.

Um detalhe importante é que a distância máxima da mutação é variável e específica para cada ponto de cada indivíduo da população. Por este motivo $Raio(i,l,c)_{Max}$ possui três índices, sendo i referente ao indivíduo, l à linha da matriz, e c à coluna. Inicialmente, $Raio(i,l,c)_{Max}$ de todos os pontos assume $Raio_{Max}$ (veja Tabela 7.1). Entretanto, ao longo da evolução o valor de $Raio(i,l,c)_{Max}$ é incrementado de $Raio_{Max}$ toda vez que o respectivo elemento sofre uma mutação e, mesmo assim, continua sobre um obstáculo ou fora da área de cobertura. Desta forma, permite-se que o operador de mutação seja mais abrangente em situações críticas.

7.3.3.4. Critérios de Parada

A evolução responsável por distribuir os pontos de coleta pára em duas situações. Primeiro, caso um número máximo de iterações (épocas) do processo evolutivo seja atingido (veja o valor na Tabela 7.1), a distribuição é terminada e a melhor solução, mesmo contendo pontos ineficazes, é adotada. Nesta situação, a responsabilidade de evitar regiões inacessíveis acaba sendo transferida para camadas subseqüentes do sistema (e.g., planejamento de trajetórias e navegação). O segundo critério é de estabilidade, ou seja, caso o fitness do melhor indivíduo da população mantenha-se estável por mais

que uma quantidade de iterações igual a 10% do máximo, encerra-se a evolução. Ao final, o melhor indivíduo é definido como o conjunto de pontos a ser visitado pelo robô durante a inspeção.

7.3.4. Evolução do Itinerário

Uma vez obtido o posicionamento dos locais de coleta, é necessário determinar a ordem na qual o robô passará por todos eles. Quando se menciona um grupo de lugares que devem ser visitados seqüencialmente, imediatamente recobra-se o problema do caixeiro viajante, abordado no Capítulo 4. O que faz este módulo do sistema é exatamente resolver cenários deste famoso problema da área de otimização combinatória, levando-se em conta algumas particularidades pertinentes a esta aplicação.

Por definição, o problema do caixeiro viajante consiste em, dada uma coleção de cidades, encontrar o itinerário mais curto para visitar todas as cidades uma vez e retornar à inicial. Especificamente neste trabalho, faz-se a analogia entre as cidades e os pontos de coleta, sendo que duas variações em relação à definição original são introduzidas:

1. Não há necessidade de o robô voltar ao primeiro ponto visitado ao final do circuito;
2. Os obstáculos presentes no mapa são considerados na determinação da ordem de visita aos pontos, caracterizando o problema como um caso de otimização com restrições.

A forma como as restrições são tratadas e todos os demais detalhes do processo evolutivo são apresentados nas próximas seções.

7.3.4.1. Codificação e Inicialização

A codificação empregada aqui armazena, para cada ponto, o ponto anterior e posterior a ser visitado. Ou seja, o cromossomo de um indivíduo é uma matriz com número de linhas igual ao número de pontos de coleta, e número de colunas igual a três: em primeiro o ponto de coleta, em segundo o elemento anterior da seqüência e, por último, o elemento posterior. A Figura 7.2 mostra como funciona a codificação e o que ela representa em termos de fenótipo.

Ponto de Coleta	1ª Conexão	2ª Conexão
A	F	C
B	E	G
C	A	E
D	G	F
E	C	B
F	D	A
G	B	D

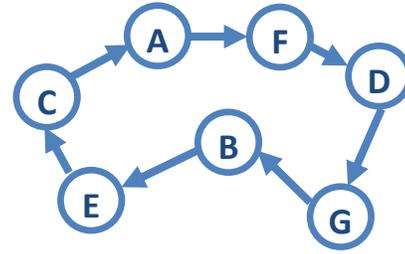


Figura 7.2: Um exemplo de indivíduo, seu genótipo e fenótipo.

A inicialização da população, embora seja feita aleatoriamente, dá chance maior de conexão entre locais mais próximos na tentativa de produzir uma geração inicial melhor. Em particular, dado um ponto, existe uma probabilidade de *ProbPtosProx* de que as suas conexões virão do conjunto dos pontos mais próximos a ele, e $(100-ProbPtosProx)$ de que a escolha das conexões será entre todos os pontos de coleta. O conjunto de pontos próximos é determinado para cada ponto no início do processo, incluindo os *NumPtosProx* de menor distância euclidiana.

Nenhuma inconsistência na construção dos indivíduos (itinerários) é aceita. Sempre que uma escolha é feita, remove-se o elemento escolhido do *pool* de opções para que não haja arcos repetidos no circuito. Obviamente, sempre que um ponto é conectado a outro, automaticamente a ligação do outro com o um é indicada no cromossomo. Como consequência das restrições citadas, proíbe-se a criação de conexões que fechem o circuito, a não ser que a ligação seja entre o último e o primeiro da seqüência.

7.3.4.2. Avaliação

O cálculo do fitness dos indivíduos é simples e intuitivo. Considerando que os pontos de coleta são fixos, assim como o mapa e seus obstáculos também o são, monta-se uma única vez, no início da evolução, a matriz de distâncias euclidianas entre todos os pontos. O detalhe importante é que todas as possíveis ligações entre dois pontos que cruzem com obstáculos do mapa têm seu respectivo valor na matriz de distâncias modificado para uma constante muito grande (*DistConexInfactive!*). Isto é, assume-se que a distância entre dois pontos cuja conexão não é factível, dada a existência de obstruções, é virtualmente infinita, penalizando itinerários com tais conexões e, conseqüentemente, forçando o algoritmo evolutivo a buscar soluções alternativas.

Vale ressaltar que a obtenção da seqüência de pontos a serem visitados irá determinar apenas a ordem pela qual o robô passará por ele. Embora nesta etapa se adote a linha reta que liga diretamente dois pontos como trajetória, representada pela distância euclidiana entre eles, a navegação efetiva do robô de um ponto a outro será controlada por um sistema especializado em planejamento de trajetórias, o mesmo proposto no Capítulo 6. Logo, espera-se que este módulo de ordenação dos pontos otimize a seqüência a ser visitada pelo robô conforme as informações conhecidas de antemão, visando realizar um planejamento teórico do processo de coleta de dados que, no entanto, pode terminar com resultados diferentes em virtude de circunstâncias encontradas no ambiente real.

7.3.4.3. Operadores Evolutivos

Após avaliar a população, é feita a seleção dos genitores pelo método do torneio, segundo uma proporção de 20% do número total de indivíduos. Isto é, 20% da população definidos aleatoriamente entram como candidatos e os dois melhores são selecionados como genitores. Diferentemente da etapa de distribuição dos pontos, aqui se adota uma estratégia mais conservadora em que, a cada geração, apenas dois descendentes são gerados a partir de operações genéticas sobre os dois melhores genitores selecionados. Os dois descendentes substituem os dois piores indivíduos presentes na população.

Para produzir os indivíduos descendentes, faz-se cruzamento e mutação. O cruzamento envolve os seguintes passos:

1. Identificam-se conexões comuns entre os dois genitores, as quais são herdadas pelos filhos;
2. As conexões divergentes são atribuídas aos filhos alternando o pai de origem (e.g., filho 1 recebe 1ª conexão divergente do pai 1, depois 2ª conexão divergente do pai 2, depois 3ª conexão divergente do pai 1, etc.). O mesmo é feito com o filho 2, mas iniciando com a 1ª conexão divergente do pai 2;
3. Caso haja conexões faltando nos filhos, elas são complementadas aleatoriamente com os pontos ausentes do respectivo circuito parcial, de forma a completar a seqüência adequadamente.

Para finalizar, executa-se o operador de mutação sobre o cromossomo dos descendentes recém criados, segundo uma determinada taxa de probabilidade (veja Tabela 7.1). A mutação simplesmente inverte a posição de dois pontos aleatórios, reconfigurando suas respectivas conexões nos cromossomos. Por exemplo: 1-2-3-4 → 1-4-3-2.

7.3.4.4. Critérios de Parada

Os critérios de parada são os mesmos usados no processo evolutivo de distribuição de pontos de coleta: número máximo de iterações (épocas) ou quantidade de iterações (10% do máximo) sem mudança no fitness do melhor indivíduo. Os parâmetros que configuram estes critérios são, entretanto, diferentes da outra evolução, podendo ser encontrados na Tabela 7.1.

7.3.5. Execução

Conforme as especificações estabelecidas pelo usuário, foram obtidos, até aqui, um conjunto de locais distribuídos de forma representativa, e a seqüência otimizada pela qual o robô os visitará. Em outras palavras, tem-se um planejamento de como realizar a coleta de dados: resta executá-lo. Para tanto, o robô deverá navegar até cada ponto e capturar as informações desejadas. Muito embora tenha havido, na etapa de ordenação dos pontos, a preocupação de se produzir um itinerário que evitasse obstáculos presentes no mapa, há diversas variantes intrínsecas à operação em ambiente e tempo real que, por serem imprevisíveis, exigem robustez e adaptabilidade no processo de navegação e coleta de dados.

Os requisitos supracitados coincidem com as características do sistema evolutivo de planejamento de trajetórias proposto no Capítulo 6. Naquele caso, o sistema é empregado em problemas de rastreamento e perseguição de alvos móveis. As sinergias entre ambos os cenários são evidentes, uma vez que a navegação do robô ocorre no mesmo tipo de ambiente dinâmico, e o robô deve executar trajetórias que o levem a pontos de destino.

Na verdade, quando se compara a tarefa de coleta de dados com a de rastreamento de alvos móveis, fica claro que a primeira é relativamente mais simples em termos de navegação, já que os pontos de destino são estáticos e, em geral, próximos uns dos outros (fator muito favorável ao sistema, como mostrado nos experimentos do Capítulo 6). Logo, o sistema será mais exigido no que tange o desvio de obstáculos inesperados, o planejamento de trajetórias alternativas e a adaptação em situações em que o local de coleta se encontra inacessível. Em todos estes quesitos, o sistema provou sua competência.

Motivado por todos os fatos discutidos, opta-se pela adoção do mesmo sistema evolutivo descrito na Seção 6.4 como controlador do robô na etapa de execução da coleta de dados. A única diferença a se ressaltar é que o robô, quando alcança o ponto de coleta, pára e aguarda o tempo necessário para que

os dados de interesse sejam obtidos (a coleta em si é um processo extremamente simples em que estímulos recebidos pelos sensores correspondentes são gravados). Só então a execução continua normalmente.

7.3.6. Interpretação dos Dados

Esta é uma etapa complementar do processo e pode ser bastante diversa, dependendo do tipo de dados coletados e objetivo da análise a ser feita. Em particular, neste trabalho, foi desenvolvida uma ferramenta de análise geográfica de informações, cujo resultado assuma a forma de mapas representativos das informações processadas. Obviamente, devido ao fato de a coleta de dados não incluir qualquer pós-processamento, a natureza bruta dos dados permite qualquer tipo de interpretação e análise *a posteriori*.

7.3.7. Parâmetros

Os parâmetros do sistema, apresentados na Tabela 7.1, foram determinados empiricamente por meio de diversos testes e simulações. Entre elas, foi feita uma análise de sensibilidade dos parâmetros, a qual é apresentada na Seção 7.4.1.

Tabela 7.1: Parâmetros do sistema evolutivo.

Nome	Distribuição dos pontos	Ordenação dos pontos
Tamanho da população	40	10000
Taxa de cruzamento	25%	--
Taxa de mutação	10%	10%
<i>Num. máximo iterações</i>	5 mil	2 milhões
<i>Raio máximo para mutação</i>	50 cm	--
<i>Proporção usada no torneio</i>	20%	10 indivíduos
<i>Chance dos pontos próximos (ProbPtosProx)</i>	--	90%
<i>Número de pontos próximos (NumPtosProx)</i>	--	10

7.4. Experimentos e Resultados

Como tradicionalmente vem sendo feito nas frentes de pesquisa, os experimentos que põem o sistema proposto à prova são divididos em duas categorias: virtual e real. O objetivo é testar o controlador e investigar sua validade para a aplicação considerada.

Note que, embora o planejamento evolutivo de trajetórias seja um integrante importante deste sistema, ele não é explorado em detalhes nesta frente, dada a extensa investigação feita a seu respeito no Capítulo 6. O foco aqui está no planejamento de missões, que são, neste trabalho, itinerários de inspeção e coleta de dados.

Embora a plataforma desenvolvida possua aplicabilidade genérica, nos experimentos a seguir ela será usada especificamente em situações de medição da qualidade do sinal transmitido por redes sem fio. Seja nas simulações ou nos testes reais, o objetivo é examinar autonomamente áreas que sejam servidas por tais tipos de redes, coletando dados que permitam mapear, de forma representativa, o nível do sinal em cada região. A frequência de coleta de amostras é, sempre, de uma por segundo.

As ilustrações e resultados apresentados a seguir adotam uma simbologia que requer algumas explicações. Nos mapas, os obstáculos (e.g., paredes) são representados por linhas negras espessas; o polígono delimitado pelas linhas marrons indica a área a ser inspecionada; círculos grandes pontilhados com quadrados pequenos no centro representam antenas ou torres virtuais de transmissão de sinal e seu alcance; já os círculos pequenos (interligados ou não) são os locais de coleta (nós) que compõem o itinerário planejado. Finalmente, faz-se uso freqüente de mapas de cores (conhecidos em inglês por *heat maps*) para apresentar graficamente os dados coletados pelo robô. As cores e suas variações de tom estão associadas à força ou precisão do sinal transmitido (zero é o melhor), obedecendo à escala da Figura 7.3.

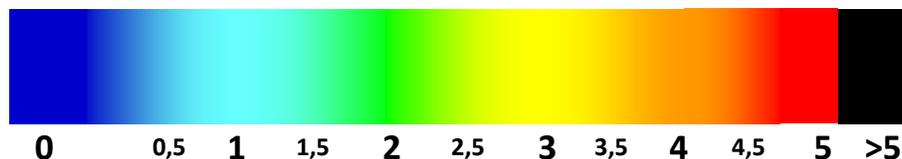


Figura 7.3: Escala de cores usada para indicar a qualidade do sinal medido, sendo zero (azul), o melhor resultado possível.

7.4.1. Análise Paramétrica

Inicialmente é feita uma avaliação da sensibilidade de alguns dos principais parâmetros envolvidos no processo de evolução de missões ou itinerários de inspeção, a saber: tamanho da população, taxa de mutação, tamanho do torneio, número de pontos próximos e chance deles serem escolhidos quando criando conexões. Os experimentos consistiam apenas nas etapas de distribuição de nós e evolução do itinerário, sendo o desempenho avaliado por meio de dois critérios: extensão total do circuito e número de gerações executadas. Cada experimento foi repetido cinco vezes, utilizando sempre o mesmo ambiente, mostrado na Figura 7.4.

Com base na análise desta seção e também em outros testes empíricos, foram definidos os melhores valores para cada parâmetro, os quais são adotados em todos os demais experimentos (veja a Tabela 7.1). Os resultados obtidos em relação a cada parâmetro são resumidos na Tabela 7.2, sendo destacadas as linhas com os valores de referência.

Examinando a Tabela 7.2, podem-se fazer algumas observações interessantes. Embora a taxa de mutação pareça não influenciar muito no comprimento do circuito, os valores maiores que 10% levam à necessidade de um maior número de gerações até a convergência. Em relação ao tamanho do torneio, ocorre o contrário: quanto maior seu valor, mais rápida é a convergência. Entretanto, esta se mostra prematura dado que as extensões médias foram piores. Em geral, não houve grandes discrepâncias de desempenho. Logo, os valores escolhidos foram aqueles com melhor compromisso fitness (extensão) versus número de gerações. Por último, vale destacar que, com as configurações de parâmetros selecionadas, conseguiu-se o melhor itinerário dentre todos os experimentos no ambiente tratado, o qual é exibido na Figura 7.4 e cuja extensão total é de 118,5 m.

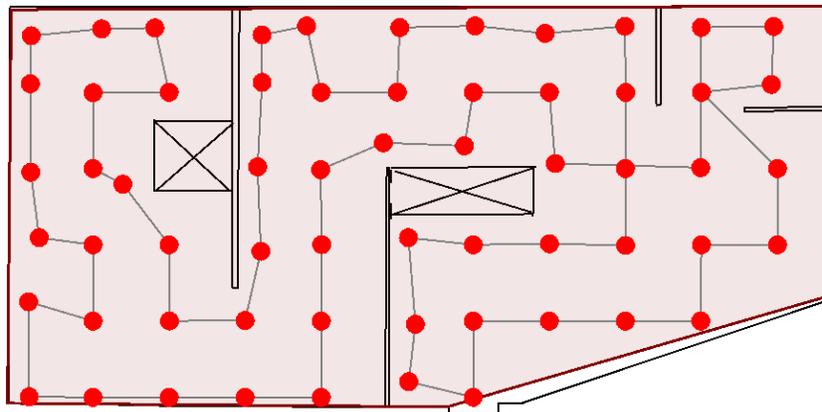


Figura 7.4: Ambiente utilizado para os experimentos de análise paramétrica. O itinerário mostrado foi o melhor obtido, totalizando 118,5 metros de extensão.

Tabela 7.2: Resultados da análise paramétrica realizada para o sistema proposto.

Parâmetro	Valores	Extensão Total Média (m)	Média de Gerações
Tamanho da população	5 mil	123,78	307368,3
	10 mil	121,42	338897,7
	15 mil	127,95	569073,0
Taxa de mutação (%)	10	121,42	338897,7
	20	122,26	424667,7
	30	122,83	393457,3
Tamanho do torneio	10	121,42	338897,7
	30	126,33	197323,7
	50	129,56	101704,0
Número de pontos próximos	3	122,93	281963,7
	7	120,21	367749,0
	10	121,42	338897,7
Porcentagem de chance do ponto próximo (%)	30	125,08	486829,7
	60	125,00	346839,3
	90	121,42	338897,7

7.4.2. Distribuição de Pontos de Coleta

Nesta seção, o objetivo é mostrar brevemente alguns exemplos de distribuição de pontos de coleta de dados. Como já dito, a forma como tais pontos são organizados determina parte do processo de inspeção e tem implicações na uniformidade da cobertura, assim como na representatividade dos dados coletados. O conjunto de pontos é obtido evolutivamente, de acordo com dois parâmetros escolhidos pelo usuário: área a ser inspecionada e densidade das amostras.

Dois casos são apresentados na Figura 7.5, ambos provenientes de simulações feitas em um ambiente de aproximadamente 162 m². Pode-se notar claramente o esforço, em geral bem-sucedido, do algoritmo para manter os pontos uniformemente distribuídos. Fica evidente também o acúmulo de nós nas bordas dos obstáculos, que é conseqüência da busca por posicionar os pontos fora de regiões obstruídas. Um possível aperfeiçoamento seria o uso de um mecanismo para eliminar pontos sobrepostos ou muito próximos.

A ilustração à esquerda da Figura 7.5 mostra o resultado da distribuição tendo em vista uma densidade desejada de 1 pto/m², enquanto, na direita, tem-se uma densidade de 3,9 pto/m². Isto quer dizer que, caso o usuário configure o robô para ficar capturando informações por 10 s em cada ponto, o

tempo total da inspeção seria, no mínimo, 27 min para o primeiro caso e 1 h 45 min para o segundo (não contando o tempo de planejamento e deslocamento).

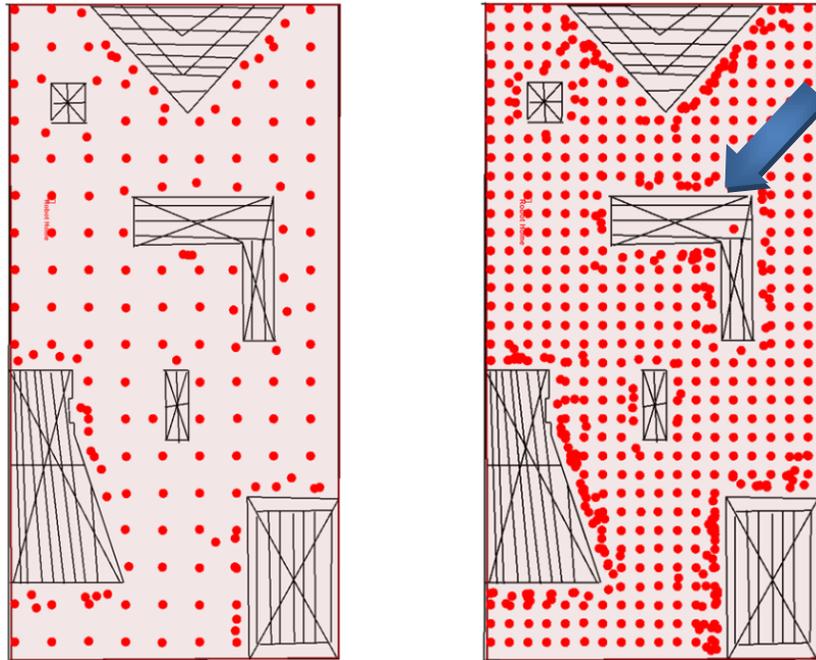


Figura 7.5: Simulações que mostram o resultado da distribuição evolutiva de locais de coleta com densidade de 1 pto/m², à esquerda, e 3,9 pto/m², à direita.

Uma pequena falha pode ser vista na figura da direita: um ponto acabou convergindo para um local inacessível (indicado pela seta), cercado por obstáculos. Embora isto seja um problema tanto para o planejamento do itinerário, quanto para a navegação, não há nenhum comprometimento sério. Graças à flexibilidade do planejador evolutivo de trajetórias, situações como esta podem ser superadas facilmente no momento da navegação, simplesmente ignorando lugares inatingíveis ao se detectar a inexistência de caminhos factíveis.

7.4.3. Simulações

Após ter examinado o sistema por partes, experimentos completos, realizados por meio de simulação computacional, são discutidos nesta seção. Nos cenários utilizados, assume-se a existência de antenas ou torres virtuais de transmissão de sinal, cuja qualidade é proporcional à proximidade do equipamento transmissor. Além disso, não há sinal em regiões fora do alcance da antena (indicado pelos círculos pontilhados).

A primeira simulação foi feita no ambiente fictício mostrado na Figura 7.6, o qual contém duas antenas virtuais de raio de alcance igual a 5 m. Dada a densidade escolhida de 0,2 pto/m², o sistema evolutivo acabou por distribuir 42 pontos de coleta, e produziu um itinerário de inspeção cuja extensão total é de 107,4 m (veja-os na Figura 7.6). Como se pode notar, o itinerário evoluído resolve com antecedência problemas de desvio de obstáculos estáticos que o robô encontraria ao navegar pelo ambiente, já que, em seu planejamento, conexões entre pontos sem obstrução são favorecidas. O cruzamento de trajetórias visto no canto superior direito é uma consequência da propriedade mencionada; embora seja bem sabido que cruzamentos em circuitos são prejudiciais em termos de otimalidade, este caso é totalmente justificável.

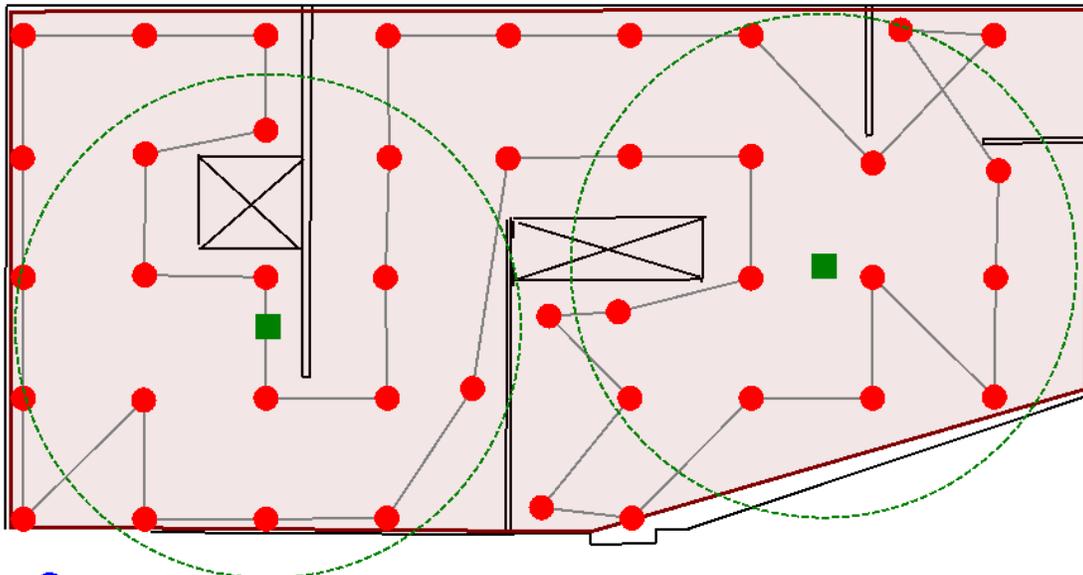


Figura 7.6: Ambiente da primeira simulação com duas antenas virtuais, cujo raio de alcance é igual a 5 metros.

Uma vez que a missão foi configurada com a exigência de que o robô ficasse apenas um segundo em cada ponto de coleta, o tempo total para executá-la foi de 18 minutos. O resultado da simulação é apresentado como um mapa de cores na Figura 7.7, ficando claro que o robô seguiu à risca o itinerário planejado. Ao se analisar o mapa compilado a respeito do sinal da rede, é possível perceber facilmente as regiões sem sinal (em preto) e aquelas com melhor qualidade, ou seja, cuja distância da antena é menor que 1 metro (em tons de azul). Note que todas as amostras colhidas, seja com o robô parado sobre os pontos de referência, seja enquanto o robô está navegando, são exibidas no mapa.

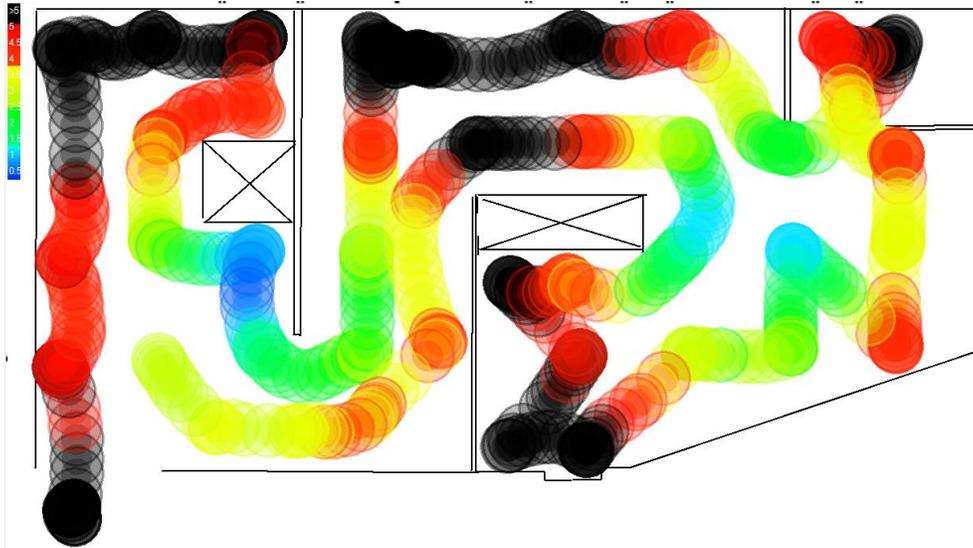


Figura 7.7: Mapa com os resultados da coleta de dados feita na primeira simulação.

Na segunda simulação, foi utilizada parte da planta de um prédio real (hospital) de 356 m² de área, a qual possui 15 salas conforme pode ser visto à esquerda da Figura 7.8. Quatro torres de transmissão virtuais foram dispostas, cada qual com raio de cobertura de 5 m. Considerando que nesta inspeção se desejava coletar dados com uma densidade de amostragem de 0,35 pto/m², o sistema gerou um conjunto de 125 nós, arranjados na conformação mostrada à esquerda da Figura 7.8. Com base nestes nós, o sistema evoluiu um itinerário de inspeção de comprimento total igual a 252 m, sem nenhuma obstrução. Obviamente que a ausência de obstruções tem seu preço: o circuito apresenta diversos cruzamentos que, de fato, o tornam mais longo, porém factível.

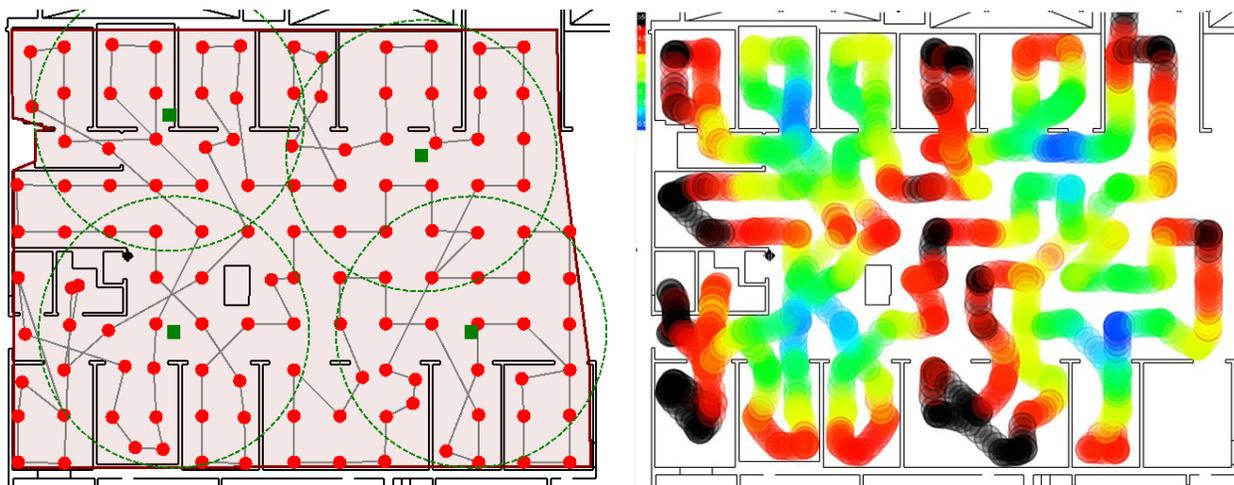


Figura 7.8: À esquerda, ambiente com quatro antenas de 5 m de alcance e o itinerário planejado para a segunda simulação. À direita, o mapa da qualidade do sinal obtido após a inspeção com o robô.

O robô levou 48 minutos para executar a missão planejada, da qual resultou o mapa exibido à direita da Figura 7.8 (todas as amostras capturadas são mostradas). A maior densidade de pontos utilizada nesta simulação, quando comparada à anterior, permitiu uma representação com maior resolução dos dados coletados. É possível identificar nitidamente a força do sinal em cada região do mapa e, inclusive, estimar a exata localização das antenas, caso esta fosse desconhecida. Verifica-se que o robô ignorou alguns pontos na parte de baixo do mapa, pois eles eram inacessíveis por estarem muito próximos da parede, e, em especial, deixou de visitar a sala no canto inferior direito, já que o sistema não conseguiu obter uma trajetória que passasse pela porta dentro da quantidade máxima de iterações do algoritmo evolutivo (veja *MaxGeracoes* na Tabela 6.1). Estas decisões foram tomadas pelo módulo de planejamento de trajetórias em tempo de execução, mostrando a flexibilidade e robustez do sistema frente a situações inusitadas.

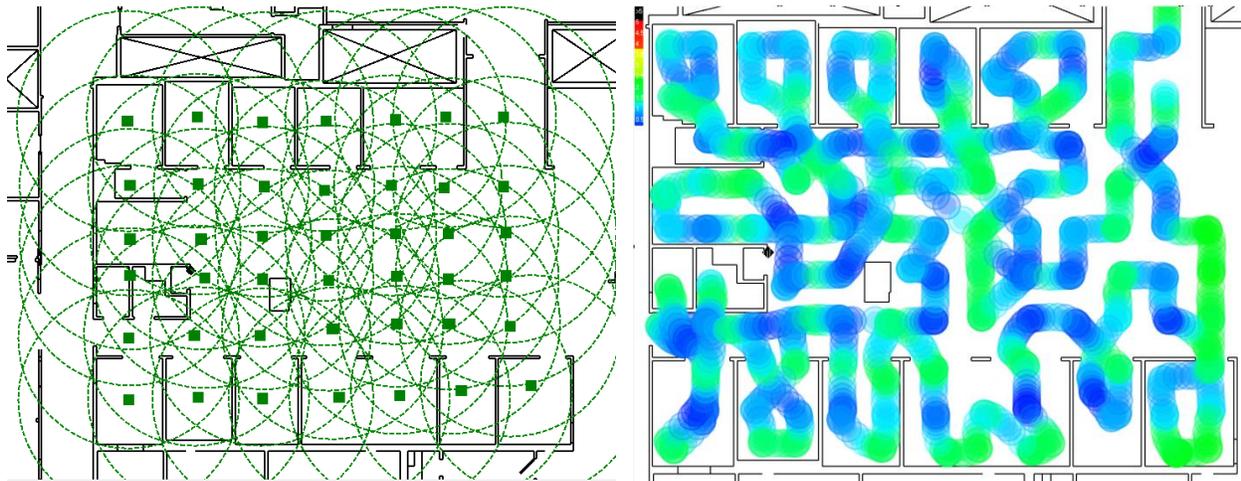


Figura 7.9: Mapa do sinal com excelente qualidade (distância das antenas sempre menor que 2 m) (direita), produzido a partir da simulação de coleta de dados realizada em um ambiente com alta densidade de transmissores (esquerda).

Uma das aplicações mais importantes da plataforma proposta se dá no sentido de modelar as condições do ambiente inspecionado e, com base nos resultados, propor aprimoramentos no sistema investigado. Justamente com este objetivo, uma terceira simulação foi feita no mesmo ambiente usado anteriormente, mas desta vez supondo uma alta concentração de antenas virtuais. Como indica a planta à esquerda da Figura 7.9, 42 antenas (de 5 m de alcance) foram colocadas bem próximas umas das outras, visando elevar a qualidade do sinal da rede hipotética.

Apenas pela comparação visual do mapa de cores resultante da simulação com 42 antenas (à direita da Figura 7.9) com o mapa da simulação com 4 antenas (à direita da Figura 7.8), já é possível notar a drástica melhoria. Mais precisamente, a quantidade de amostras com distância menor que 2 m

aumentou de 17% para 96,5%, ao mesmo tempo em que a proporção de amostras com valor maior que 2 m caiu de 83% para 3,5%. Finalmente, nenhuma região sem sinal foi detectada, ao passo que, no caso das 4 antenas, 13,4% das amostras manifestaram esse problema.

7.4.4. Experimentos Reais

Nos experimentos a seguir, a plataforma de inspeção e coleta de dados é usada em aplicações reais, especificamente para medir e mapear a precisão de um sistema de localização de objetos e pessoas por rádio frequência instalado no Washington Hospital Center. O sistema a ser avaliado é o mesmo utilizado na frente dedicada a rastreamento evolutivo (Capítulo 6), sendo ele constituído por uma rede de antenas para comunicação sem fio via sinais de banda ultra-larga.

Diferentemente das simulações anteriores, esta rede é voltada à observação e estimação do posicionamento de *tags* (transponders). Como mencionado no capítulo anterior, existe uma quantidade considerável de ruído na determinação da localização dos dispositivos RFID, o que leva a imprecisões. O robô é empregado, portanto, com o objetivo de identificar áreas problemáticas, possibilitando a introdução de ajustes que levem à elevação da acurácia da rede.

Para fazer este mapeamento, duas *tags* foram acopladas ao robô, sendo assim possível comparar suas coordenadas (estimadas pelo sistema de localização) com as coordenadas do robô (referência). Como fisicamente as *tags* e o robô se encontram no mesmo lugar, qualquer discrepância observada é considerada como erro de estimação. Imagens de como as *tags* foram montadas no robô podem ser vistas na Figura 7.10 e Vídeo 7.1.

Os experimentos de inspeção foram executados na sala de emergência do hospital (543 m²), a qual é considerada modelo, pois dispõe das tecnologias mais modernas existentes na área médica, incluindo o sistema de localização de objetos. Enquanto o primeiro teste foi feito poucos dias antes da inauguração das premissas, o segundo ocorreu dois meses mais tarde, permitindo que a rede fosse re-configurada de acordo com os resultados iniciais. Pelo fato do primeiro teste ter sido realizado em um ambiente sem pacientes e com poucos funcionários, como se nota na Figura 7.10, foi possível inspecionar um número maior de áreas, em particular quartos de pacientes, do que no segundo experimento. De qualquer forma, a maior parte das áreas não reservadas foi coberta em ambos os casos, possibilitando comparações relevantes dos resultados.



Figura 7.10: Cenas não seqüenciais capturadas durante o processo de inspeção e coleta de dados realizado na sala de emergência, as quais fazem parte do Vídeo 7.1.

Ambas as missões foram planejadas para ter pontos de coleta distribuídos com densidade $0,25 \text{ pts/m}^2$ e requisitando que o robô ficasse parado em cada um deles por pelo menos 30 segundos, visando coletar uma quantidade representativa de amostras. Dos 136 pontos distribuídos pelo sistema evolutivo no primeiro itinerário, 86 foram efetivamente visitados; já no segundo itinerário, a razão foi de 61 visitados para 101 planejados. Os mapas com os resultados são mostrados a seguir, respeitando a mesma escala de cores da Figura 7.3.

A primeira análise comparativa é exibida na Figura 7.11, estando à esquerda os resultados da primeira inspeção. As cores representam o erro médio da posição estimada das duas *tags* em relação à posição do robô. Neste caso, foram utilizadas apenas amostras de locais onde o robô esteve por, no mínimo, 5 s. Isto é, o erro em cada ponto é calculado pela média da distância medida em, pelo menos, 10 amostras ($2 \text{ tags} \times 5 \text{ s}$). O tamanho dos círculos aqui é fixo e relativo à densidade de cobertura desejada, ao passo que os traços que partem dos centros indicam a exata posição de cada amostra.

Pode-se constatar nos gráficos (c) e (d) da Figura 7.11, algo que é notório nos mapas correspondentes (a) e (b): as melhorias impostas à rede de banda ultra-larga promoveram uma redução expressiva do erro absoluto (distância das *tags* para o robô) de localização. Um exemplo específico é a substituição de uma antena defeituosa que foi identificada na região do círculo negro no corredor central (veja (a)), a qual estimava coordenadas totalmente equivocadas, causando um erro médio maior que 5 m.

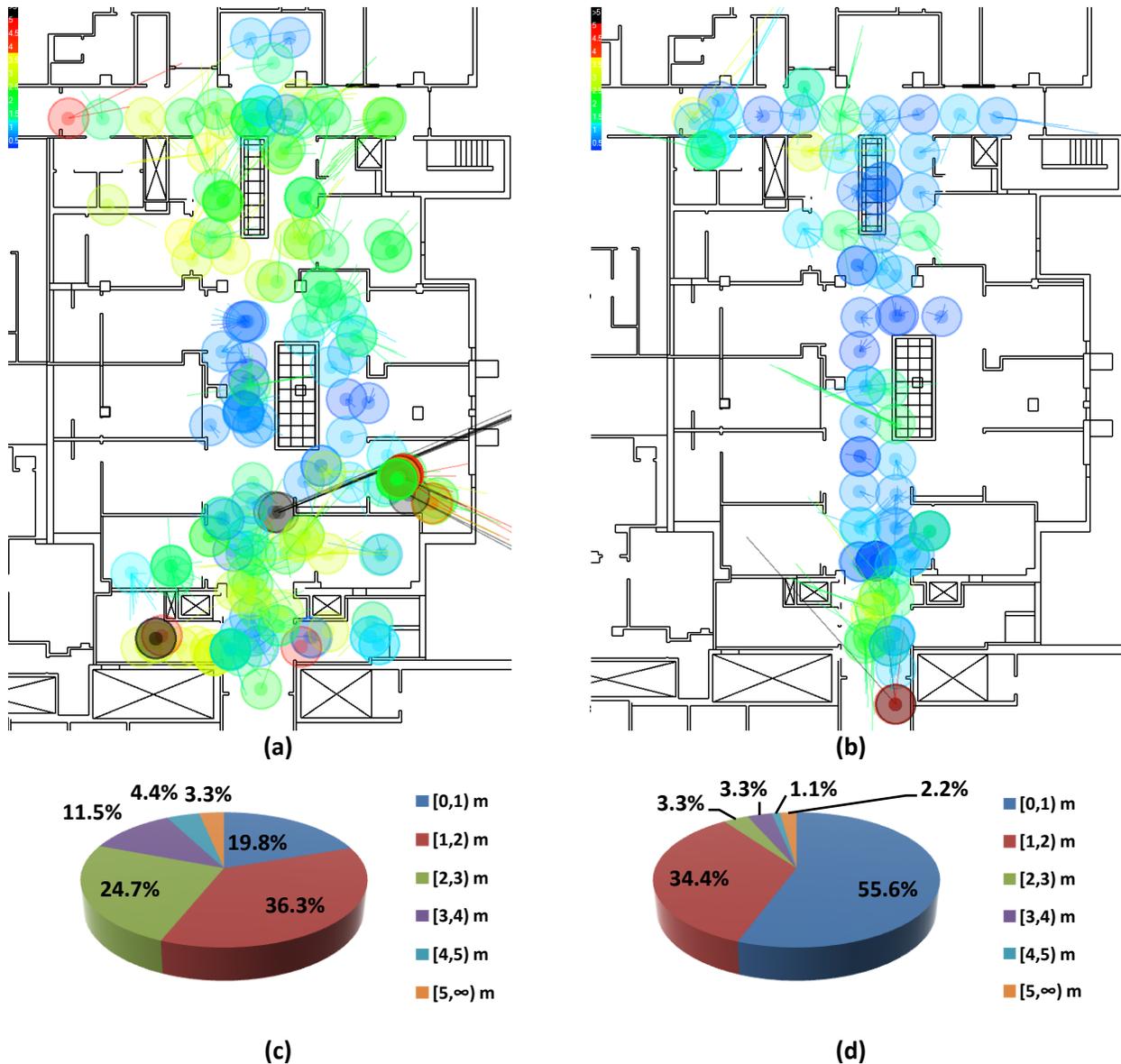


Figura 7.11: Resultados relativos ao erro absoluto nos dois experimentos reais, sendo considerados aqui apenas os pontos onde o robô ficou parado por 5 segundos ou mais. Em (a) e (b) mostram-se mapas dos dados coletados no primeiro e segundo experimentos, respectivamente. O gráfico (d) mostra a melhora acentuada da qualidade dos resultados em comparação ao gráfico do primeiro teste (c).

A Figura 7.12 apresenta outra análise dos resultados das duas inspeções na sala de emergência. Desta vez, todas as amostras coletadas foram compiladas para compor os gráficos mostrados: tanto as medidas feitas com o robô momentaneamente estático, quanto as obtidas com o robô em movimento. Ainda que estas não tenham a mesma representatividade estatística daquelas, assim fica mais fácil visualizar o itinerário completo efetuado pelo robô em (a) e (b). Os gráficos em (c) e (d) confirmam a melhoria na precisão do sistema de localização, propiciada pelo trabalho automatizado do robô.

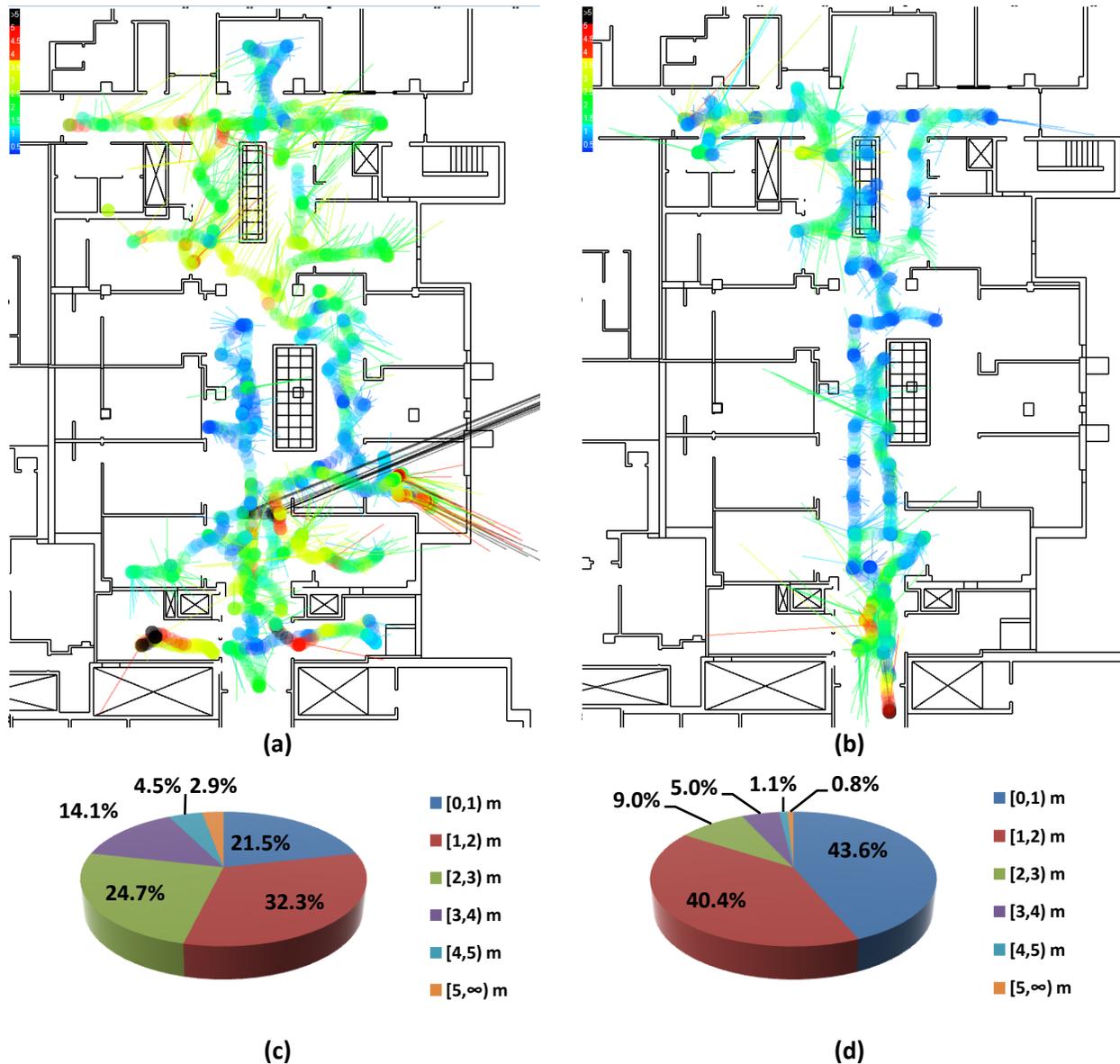


Figura 7.12: Resultados referentes ao erro absoluto nas duas inspeções na sala de emergência, sendo exibidas todas as amostras coletadas, inclusive com o robô em movimento. (a) e (b) representam as amostras em forma de mapa para o primeiro e segundo experimentos, respectivamente. Novamente nota-se um maior número de regiões com menor nível de erro médio no gráfico (d), correspondente ao segundo teste, do que no gráfico do primeiro teste (c).

É fato que o grau de redução do erro médio absoluto nos gráficos da Figura 7.12 acaba sendo levemente menor do que nas estatísticas da Figura 7.11. Isto se justifica devido à computação dos dados recolhidos com o robô em movimento, situação que traz maiores dificuldades para estimar a posição das *tags* e, conseqüentemente, introduz uma taxa mais alta de ruído em tais amostras. Contudo, é possível observar outros detalhes significativos, como a transição do sinal entre regiões do mapa. Por vezes a variação é gradativa, principalmente em (b); já em (a), diversas ocorrências de transições abruptas são perceptíveis.

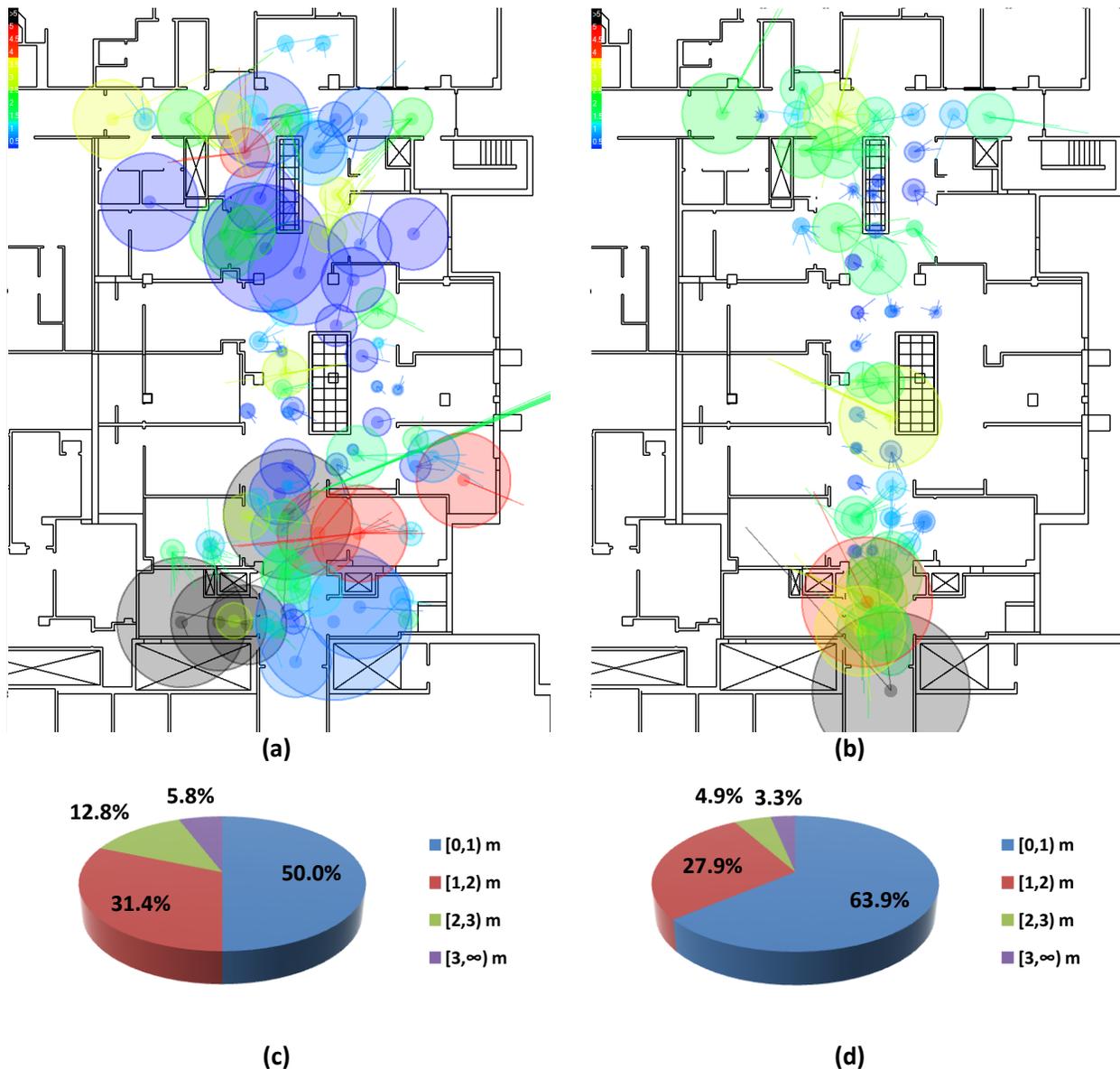


Figura 7.13: Resultados que indicam o desvio padrão do erro relativo (entre tags) nas duas inspeções, tendo sido usados apenas dados dos locais em que o robô esteve por 30 s ou mais. Os mapas da primeira e segunda coletas são apresentados em (a) e (b), respectivamente. Percebe-se nos gráficos (c), do primeiro experimento, e (d), do segundo, que há um aumento de 80,4% para 91,8% no número de regiões com desvio padrão menor que 2 metros.

O último estudo, sintetizado na Figura 7.13, investiga outros aspectos do problema. Discute-se aqui o desvio padrão do chamado erro relativo de localização. Uma vez que havia duas *tags* sendo monitoradas e elas estavam na mesma posição (ignorada a altura), a distância entre elas determina o erro relativo. O desvio padrão, calculado usando apenas dados com 30 ou mais amostras (i.e., locais onde o robô esteve parado por 30 s ou mais), é representado pelo tamanho dos círculos em (a) e (b) (as cores continuam se referindo à média do erro relativo). Mais uma vez fica evidente o progresso do sistema. Por exemplo, a quantidade de regiões cuja variação do erro entre *tags* é menor que 2 m aumentou de 81,4% para 91,8%.

7.5. Conclusão

Encerrando as frentes de pesquisa contidas neste trabalho, propôs-se, neste capítulo, um sistema evolutivo capaz da realização de planejamento não apenas no nível de trajetórias, mas também de missões. Dentre todas as frentes, esta é a que explora o maior espectro de camadas dentro da arquitetura de um controlador, contemplando desde a reatividade presente no módulo de navegação do robô até graus mais altos de deliberação com os módulos de planejamento de curto e longo prazo.

No problema tratado, o robô é utilizado como uma ferramenta autônoma para inspeção de ambientes, coleta de dados e avaliação de resultados. Especificamente, as missões de inspeção se resumem a itinerários compostos por pontos de coleta, os quais são planejados por meio de técnicas de computação evolutiva. Além disso, a execução das missões também é feita por um sistema evolutivo, o mesmo desenvolvido no Capítulo 6.

Observe que os algoritmos evolutivos utilizados aqui são relativamente simples se comparados ao estado da arte na resolução dos sub-problemas específicos abordados. A intenção não é obter a solução ótima, mas sim uma boa solução em tempo adequado, que resulte em melhoras na execução do planejamento evoluído. Obviamente que existem diversas alternativas muito mais sofisticadas na literatura para problemas como o do caixeiro viajante, por exemplo. Entretanto, julga-se bastante útil a abordagem simplificada apresentada neste trabalho, dadas a aplicabilidade e a praticidade em situações diversas.

Outro assunto interessante a se discutir é a questão de em que nível ou camada de um sistema híbrido se deve alocar cada tipo de decisão a ser tomada. Em particular, nesta proposta, um exemplo claro são as decisões em relação ao problema de desvio de obstáculos estáticos presentes no mapa

conhecido *a priori*. Há duas alternativas: criar itinerários que já contemplem conexões livres entre pontos (sem obstrução) ou deixar que o robô, quando navegando, tente desviar dos obstáculos que porventura encontre em seu caminho.

A opção de tomar tais decisões na camada mais alta (planejamento de missões) traz vantagens associadas ao aproveitamento de informações conhecidas *a priori*, para criar planos que facilitem a etapa de execução. Neste trabalho, se usa o mapa do ambiente para planejar o itinerário do robô, visando antecipar e eliminar possíveis rotas que cruzem com obstáculos. Embora se gaste maior esforço e tempo para planejar a missão, isto facilita o trabalho da camada de planejamento de trajetórias que atua em tempo de execução. Ou seja, espera-se obter uma maior fluidez na navegação do robô, o que é benéfico para uma coleta de dados mais uniforme. O aspecto negativo desta abordagem é que ela tende a criar itinerários com cruzamentos e, portanto, mais longos, prejudicando a minimização das trajetórias.

Por outro lado, é recomendável vincular a tomada deste tipo de decisão à camada de execução se as informações inatas disponíveis não sejam confiáveis ou precisas. Assim, evita-se planejar missões cujos detalhes possam estar equivocados ou inválidos, incumbindo o robô de apenas reagir quando preciso. Caso esta opção tivesse sido adotada neste trabalho, no entanto, a camada de planejamento de trajetórias ficaria sobrecarregada desnecessariamente, já que os mapas utilizados são de boa qualidade. Possivelmente, o robô ainda levaria mais tempo na execução da inspeção, do qual, boa parte seria consumida com o robô parado planejando trajetórias factíveis.

Com relação aos experimentos e resultados obtidos, considera-se que eles foram bem-sucedidos no intuito de mostrar a utilidade da proposta. Enquanto as simulações visaram explorar aspectos mais intrínsecos ao sistema, os experimentos reais focaram nas implicações e benefícios práticos do uso de um robô autônomo inteligente em atividades de inspeção. Os maiores desafios foram, sem dúvida, as inspeções do sistema de localização por rádio-frequência instalado na sala de emergência do hospital em Washington D.C. Ao mesmo tempo, a maior realização foi conseguir, por meio da plataforma implementada, detectar as regiões problemáticas de forma a permitir significativas melhorias no sistema.

As perspectivas futuras são inúmeras. Dado o caráter genérico da plataforma de inspeção, projeta-se testá-la para outras atividades como avaliação do sinal de provedoras de celular, medição de níveis de ruído sonoro e captura de vídeos ou imagens. Além disso, a composição modular do sistema de

controle permite vislumbrar sua expansão para lidar com missões ainda mais complexas. Isto é, enquanto uma camada do sistema cuida das trajetórias e outra dos itinerários dentro de uma área restrita, poderiam ser adicionadas camadas progressivamente mais altas para planejar seqüências de departamentos a serem visitados, depois circuitos de prédios, cidades, e assim por diante.

7.6. Descrição dos Vídeos

Vídeo 7.1: Edição com várias cenas filmadas durante o processo de inspeção e coleta de dados realizado na sala de emergência do Washington Hospital Center. O objetivo era avaliar a qualidade e acurácia do sistema de localização por radiofrequência lá instalado.

Capítulo 8: Conclusão

8.1. Visão Geral

Entre as tecnologias consideradas do futuro, a robótica é uma das mais esperadas e antecipadas. Mais do que especulações, existem sinais claros de que os robôs estão cada vez mais presentes nos diversos segmentos da sociedade, trazendo benefícios antes impensáveis. A popularização dos robôs está em curso e, embora ainda haja enormes obstáculos pela frente, sua sina tende a ser muito semelhante àquela dos computadores pessoais, os quais são, em 2008, tão numerosos quanto 1/6 da população total do planeta.

A grande limitação que desafia os pesquisadores é que os robôs atuais são insuficientemente inteligentes para atender as demandas mais complexas. Por outro lado, na natureza, há abundância de seres capazes de tomar decisões e manter sua integridade mesmo sujeitos a condições desfavoráveis e não-antecipáveis. Estes seres biológicos com inteligência e autonomia extraordinárias são resultados de outro fenômeno natural impressionante, responsável pela sua geração: a evolução das espécies.

Motivado e guiado pelos fundamentos da computação bio-inspirada e da robótica evolutiva, este trabalho propõe o desenvolvimento de cinco sistemas de controle inteligente para robôs móveis autônomos, os quais são empregados em problemas relevantes da área. Todas as frentes contemplam fases de experimentação dos controladores, tanto em modo de simulação como em ambientes reais, buscando mostrar a validade e aplicabilidade das propostas.

Tal qual nos processos de engenharia em geral, na elaboração de sistemas para controle de robôs também existe uma diversidade de parâmetros e aspectos que regem as etapas de projeto, desenvolvimento e experimentação. Muito além de apenas tomar decisões e implementá-las na forma de controladores, este trabalho também objetiva investigar tais aspectos, teórica e praticamente, com o intuito de avaliar suas implicações nos comportamentos e resultados finais.

8.2. Revisão e discussão

As principais características das pesquisas de sistemas de navegação autônomos investigadas ao longo deste trabalho são: arquitetura, modo de experimentação, aprendizagem, coletividade e

otimização. O papel de cada uma delas, em cada frente específica, foi tratado nos seus respectivos capítulos. No entanto, há aspectos que merecem destaque.

Das cinco frentes, duas são classificadas como reativas (Capítulos 3 e 4) e três possuem arquitetura híbrida (Capítulos 5, 6 e 7). A ordem dos capítulos foi definida de forma a coincidir com a seqüência progressiva desta propriedade dos sistemas. Isto é, a cada capítulo, novos elementos são introduzidos na arquitetura dos sistemas como resposta aos desafios adicionais inerentes aos respectivos problemas tratados.

Na primeira frente (Capítulo 3), opta-se por um sistema puramente reativo, dada a total ausência de conhecimentos a respeito do ambiente em que o robô navega. O mesmo sistema reativo é adaptado na segunda frente (Capítulo 4) para controlar múltiplos robôs operando de forma cooperativa indireta. Embora os ambientes também sejam desconhecidos, o sistema é mais avançado pois recebe o incremento de um mecanismo evolutivo de comunicação por estigmergia, permitindo que os robôs colaborem entre si no cumprimento de seus objetivos.

O uso do próprio ambiente como meio de comunicação indireta (fenômeno inspirado no comportamento das formigas) é um grande benefício pelo fato de disponibilizar uma fonte adicional de informações voltadas para a tomada de decisão. Obviamente que a necessidade de aprender como usar tais informações (feromônios virtuais) implica em uma exigência extra ao sistema, a qual é compensada pelos ganhos expressivos de desempenho presentes nos resultados.

É sabido que controladores reativos têm grandes dificuldades com problemas de mínimo local. E foi justamente este o desafio enfrentado na terceira frente (Capítulo 5): os ambientes utilizados eram, além de desconhecidos, cheios de becos e armadilhas. Para lidar com este novo desafio e, ao mesmo tempo, manter o desempenho visto anteriormente, propôs-se a integração do sistema classificador reativo a um modelo de rede imunológica, promovendo assim uma vinculação direta entre múltiplos modos de comportamento. Esta arquitetura híbrida mostrou possuir um balanço ideal de reatividade e deliberação, visto que conferiu ao robô a capacidade de detectar e escapar de situações de mínimo local, sem perder em adaptabilidade. É claro que as vantagens têm seu custo: a maior complexidade do sistema acaba por requerer maior esforço computacional para evoluí-lo e executá-lo.

Nas duas frentes de pesquisa restantes, uma novidade surge: os problemas tratados contêm informações disponíveis *a priori*. Tais tipos de problema são comuns quando o robô opera em regiões

urbanas ou dentro de prédios cujo mapa é total ou parcialmente conhecido. O desafio consiste, então, em tirar proveito dos subsídios fornecidos sem perder autonomia para lidar com situações potencialmente inesperadas. Para tanto, as soluções adotadas são sistemas híbridos que combinam camadas reativas e deliberativas.

O sistema do Capítulo 6, utilizado para rastreamento de pessoas ou equipamentos móveis, investe em planejamento evolutivo de trajetórias de uma forma inusitada: tanto a camada deliberativa quanto a reativa empregam o mesmo planejador (algoritmo evolutivo). Enquanto a primeira usa apenas conhecimentos inatos (mapa) para produzir uma trajetória de referência completa, a segunda a atualiza gerando trechos parciais ajustados às condições atuais do ambiente (detectáveis pelos sensores do robô), o que permite que o robô navegue com segurança.

Por fim, um passo maior é dado ao se propor um problema que requer, além de planejamento de trajetórias, planejamento de missões (Capítulo 7). Trata-se de usar o robô para inspeção de ambientes, missão esta formada por múltiplas tarefas, como o estabelecimento dos locais a serem visitados, a determinação do itinerário e sua execução. Mais uma vez, opta-se por uma abordagem evolutiva, modular e voltada ao reuso, reaproveitando o planejador de trajetórias do capítulo anterior e acoplando a ele uma terceira camada cujo papel é de deliberação da missão. Uma das grandes virtudes deste sistema é o modo como o planejador de missão trabalha as informações pré-conhecidas visando antecipar dificuldades (obstáculos) e facilitar a tarefa do planejador de trajetórias e, conseqüentemente, a navegação do robô.

Ainda em relação à arquitetura do sistema, ressalta-se a capacidade de tolerância a falhas, atributo altamente desejável em robótica autônoma. Controladores reativos são, em geral, intrinsecamente mais robustos por não dependerem de fatores externos (e.g. mapa). Ao se lidar com componentes deliberativos, entretanto, as preocupações devem ser maiores. A filosofia adotada ao longo deste trabalho foi de combinar deliberação e reatividade visando justamente dar uma maior confiabilidade ao sistema. A modularidade dos controladores concebidos também colabora nesta direção. Um bom exemplo é a plataforma de inspeção e coleta de dados, em que há uma cadeia de componentes organizados de modo a produzir redundância. Ainda que o itinerário da missão planejada contenha caminhos ineficazes (causado por um mapa desatualizado, por exemplo), o módulo de planejamento de trajetórias globais tentará construir uma rota válida; caso este também falhe, o subsistema de

trajetórias locais utilizará aquilo que tem disponível, ou seja, os estímulos captados pelos sensores do robô, para navegá-lo até seu destino.

Um aspecto essencial, que talvez até passe despercebido, é a capacidade de aprendizagem e adaptação presente nos sistemas autônomos. Um robô conseguir desviar de um obstáculo é um grande feito. Aprender a fazê-lo, porém, é algo extremamente valioso. Boa parte da robustez e autonomia dos controladores estudados se deve à habilidade de aprender com a experiência ou se adaptar a situações inusitadas. Mesmo que um robô resolva problemas, aprender a solucioná-lo e ser capaz de se adaptar a suas variantes são grandes virtudes de uma máquina que se deseja chamar de inteligente.

Muitos dos bons resultados obtidos ao longo da pesquisa se devem à possibilidade da realização de simulações computacionais. Em particular pela presença de algoritmos evolutivos em todas as propostas, reconhecidamente exigentes de poder de processamento, este trabalho vem confirmar a relevância deste modo virtual de experimentação. Obviamente que é premente a necessidade de testar e validar controladores robóticos no ambiente real para o qual eles foram desenvolvidos. Defende-se aqui a utilização conjunta e complementar dos dois modos sempre que possível, tendo em vista seus benefícios.

Outra evidência prática em favor das simulações computacionais pôde ser observada ao longo desta tese. Nas seções referentes a resultados e experimentos, existem tabelas com grande quantidade de dados somente para os casos simulados. Isto é, a agilidade e praticidade da simulação tornam possível coletar informações relevantes em intervalos de tempo aceitáveis. Por vezes, alguns experimentos podem levar a conclusões equivocadas caso uma variada gama de testes não seja executada.

Em relação à robótica coletiva, apenas uma frente de pesquisa (Capítulo 4) explorou esta vertente de enorme potencial (aqui se restringiu a experimentos em ambientes virtuais). Embora existam muitos outros aspectos envolvidos no projeto de sistema multi-robôs, a comunicação foi o principal investigado. Os resultados mostraram que mais de um robô operando em conjunto, mesmo que simples e se comunicando primitivamente, são capazes de fazer emergir comportamentos complexos. A robótica coletiva seria, sem dúvida, extremamente conveniente nas missões de inspeção do Capítulo 7, sendo esta possibilidade deixada como um tema para investigação futura.

8.3. Considerações Finais

É verdade que o estado atual da robótica está longe de ser como apostam os especialistas ou insinuam os filmes e livros de ficção científica. Mesmo os mais sofisticados robôs atuais apresentam comportamentos incipientes quando comparados a animais e bebês. No entanto, cada pequeno avanço conseguido nesta área deve ser avaliado como um passo importante. Afinal de contas, o que se espera de um robô é, nada mais, nada menos, que ele interaja com o mundo de forma tão ou mais competente que a empregada por seres biológicos.

Nesta tese, uma série de avanços foi relatada. A começar por um robô capaz de, em um ambiente completamente desconhecido, aprender a encontrar locais de interesse se esquivando de obstáculos. Depois, não um, mas múltiplos robôs, além de enfrentarem o mesmo desafio anterior, o fazem encurtando caminho (entre os locais de interesse), pois cooperam entre si para tanto. O próximo robô concebido é capaz dos mesmos feitos do primeiro, porém ele também possui a habilidade de aprender a escapar de labirintos e becos sem saída, algo de que, embora pareça simples, insetos como as moscas são incapazes. O robô seguinte, por sua vez, detém a competência de acompanhar pessoas dentro de ambientes como hospitais, cheios de gente e obstáculos, evitando colisões. Finalmente, o último robô desenvolve a capacidade de vistoriar as dependências de um prédio com eficiência, escolhendo e visitando os lugares de maior interesse para recolher informações pertinentes. O grau de importância das contribuições representadas pelos robôs mencionados e sistemas que os controlam é amplificada por um fato que merece destaque: eles operam de forma autônoma, ou seja, sem qualquer auxílio ou intervenção externa.

É difícil prever quando e como os robôs se tornarão realmente comuns e úteis tal qual os eletrodomésticos e computadores são hoje. De qualquer forma, para os cientistas e empreendedores que dedicam suas carreiras ao avanço da robótica e áreas correlatas, isto pouco importa. É, na verdade, do trabalho contínuo e progressivo de todos eles, da soma de esforços, como os desta pesquisa e de tantas outras sendo realizadas, que se originarão os alicerces dos robôs do futuro.

Referências Bibliográficas

- Aguirre, L. A. (Ed.). (2007). *Enciclopédia de Automática - Controle & Automação*. São Paulo: Editora Blucher.
- Arai, T., Pagello, E., & Parker, L. (2002). Guest editorial, advances in multi-robot systems. *18*, pp. 655–661. *IEEE Transactions on Robotics and Automation*.
- Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press.
- Balch, T., & Arkin, R. C. (1994). Communication in reactive multiagent robotic systems *Autonomous Robots*, *1*, pp. 27–52.
- Bates, E., & Elman, J. (2002). Connectionism and the study of change. In M. Johnson (Ed.), *Brain Development and Cognition: A Reader* (2a. ed.). Oxford: Blackwell Publishers.
- Beckers, R., Holland, O. E., & Deneubourg, J. L. (1994). From local actions to global tasks: Stigmergy and collective robotics. *International Workshop on the Synthesis and Simulation of Living Systems*. Artificial Life IV.
- Bekey, G. A. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*. MIT Press.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press.
- Bonabeau, E., Theraulaz, G., Deneubourg, J. L., Aron, S., & Camazine, S. (1997). Selforganization in social insects. *Trends in Ecology and Evolution*, *12*, 188-193.
- Booker, L. B., Goldberg, D. E., & Holland, J. H. (1989). Classifier Systems and Genetic Algorithms. *Artificial Intelligence*, *40*, 235-282.
- Brooks, R. A. (1991). Intelligence without reason. *In Proc of the 1991 Int Joint Conf on Artificial Intelligence*, (pp. 569–595).
- Brooks, R., & Arkin, R. (1986). Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, *2* (1).
- Caetano, F. H., Klaus, J., & Zara, F. J. (2002). *Ants: Biology and Anatomy (in Portuguese)*. Editora da UNESP.
- Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Self-Organization in Biological Systems* Princeton. Princeton University Press.
- Cao, Y., Fukunaga, A., & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *4*, pp. 1-23. *Autonomous Robots*.

- Capozzi, B. (2001). Evolution-Based Path Planning and Management for Autonomous Vehicles. *Ph.D. Thesis*. University of Washington.
- Cazangi, R. R. (2004). Uma Proposta Evolutiva para Controle Inteligente em Navegação Autônoma de Robôs. *Dissertação de Mestrado*.
- Cazangi, R. R., Von Zuben, F. J., & Figueiredo, M. F. (2003). A classifier system in real applications for robot navigation. *Proceedings of the 2003 Congress on Evolutionary Computation*. vol. 1, pp. 574–580. Canberra, Australia: IEEE Press.
- Chandler, P. R., Rasmussen, S., & Pachter, M. (2000). UAV cooperative path planning. *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Denver, CO.
- Chen, H., Fuller, S., Friedman, C., & Hersh, W. (Eds.). (2005). *Medical Informatics: knowledge management and data mining in biomedicine*. Springer.
- Chen, Z., & Birchfield, S. (2006). *Autonomous Person Following Using a Vision-Based Mobile Robot*. University of Clemson.
- Chien, S. (2000). Aspen - automated space mission operations using automated planning and scheduling. *SpaceOps 2000*. Toulouse, France.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* (1st ed.). London: John Murray.
- Dawkins, R. (1997). *Climbing Mount Improbable*. W W Norton & Co Inc.
- de Castro, L. N. (2003). Immune cognition, micro-evolution, and a personal account on immune engineering. *S.E.E.D. Journal*, vol. 3 (no. 3), 134–155.
- de Castro, L. N., & Timmis, J. I. (2002). *Artificial Immune Systems: A New Computational Intelligence Approach*. London: Springer-Verlag.
- de Castro, L. N., & Von Zuben, F. J. (Eds.). (2005). *Recent Developments in Biologically Inspired Computing*. Idea Group Inc.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley.
- Deneubourg, J. L., Aron, S., Goss, S., & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3, 159-168.
- Deneubourg, J. L., Goss, S., Franks, N., SendovaFranks, A., Detrain, C., & Chretien, L. (1991). *The dynamic of collective sorting robot-like ants and antlike robots*. In JA Meyer, SW.
- Deneubourg, J.-L., Pasteels, J. M., & Verhaeghe, J. C. (1983). Probabilistic behaviour in ants: A strategy of errors. *Journal of Theoretical Biology*, 105, 259-271.

- Deutsch, O., Harrison, J., & Adams, M. (1985). Heuristically-guided planning for mission control/decision support. *AIAA Conference on Guidance, Control and Navigation*, (pp. 86–395). Snowmass, CO.
- Ding, Y., He, Y., & Jiang, J. (2003). Multi-Robot Cooperation Method Based on the Ant Algorithm. *Proceedings of the 2003 IEEE*. Indianapolis, USA: Swarm Intelligence Symposium.
- Dorf, R. C., & Bishop, R. H. (2008). *Modern Control Systems*. Pearson Prentice Hall.
- Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. Cambridge, MA: MIT Press/Bradford Books.
- Drexler, E. K. (1987). *Engines of Creation*. Anchor.
- Drogoul, A., & Ferber, J. (1992). From Tom Thumb to the Dockers: Some experiments with foraging robots. *Second International Conference on Simulation of Adaptive Behavior*, (pp. 451-459). Honolulu, HI.
- Dudek, G., & Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press.
- Edelen, M. R. (2003). Swarm Intelligence and Stigmergy: Robotic Implementation of Foraging Behavior. *MSc Thesis*. University of Maryland.
- Elshamli, A., Hussein, A., & Areibi, S. (2004). Genetic Algorithm for Dynamic Path Planning. *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, (pp. 677–680).
- Ermentrout, G., & Edelstein-Keshet, L. (1993). Cellular automata approaches to biological modeling. *Journal of Theoretical Biology*, 160, 97-133.
- Eysenck, H. (2000). *Intelligence: A New Look*. Transaction Publisher.
- Farmer, J. (1990). A rosetta stone for connectionism. *Physica D*, vol. 42 (no. 1–3), pp. 153–187.
- Farmer, J., Packard, N., & Perelson, A. (1986). The immune system, adaptation, and machine learning. *Physica*, vol. 22D, 187–204.
- Figueiredo, M. (1999). Navegação Autônoma de Robôs. *VII Escola de Informática da SBC - Regional Sul*, (pp. 74 - 106).
- Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, vol. 28, pp. 3–72.
- Gaspar, A., & Hirsbrunner, B. (2002). From optimization to learning in learning in changing environments: The Pittsburgh immune classifier system. *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS)*, (pp. 190–199).
- Ge, S., & Cui, Y. (2002). Dynamic Motion Planning for Mobile Robots Using Potential Field Method. *Autonomous Robots*, vol. 13 (n. 3), 207–222.

- Grasse, P. P. (1959). *La reconstruction du nid et les coordinations inter-individuelle chez bellicoitermes natalenis et cubitermes sp la theorie de la stigmergie: Essai d'interpretation des termites constructeurs* (Vol. 6). Insectes Sociaux.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
- Hershberg, U., & Efroni, S. (2001). The immune system and other cognitive systems. *Complexity* , vol. 6 (no. 5), pp. 14–21.
- Holland, J. (1975). *Adaptation in natural and artificial systems* (1ts ed.). The University of Michigan Press.
- Holland, J. (1986). Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. In R. Michalsky, C. J, & M. T, *Machine Intelligence II*. Eds. Morgan Kaufmann.
- Holland, J. (1995). *Hidden Order*. Addison-Wesley Reading.
- Holland, O., & Melhuish, C. (1999). Stigmergy, self-organization and sorting in collective robotics *Artificial Life.*, 5, pp. 173-202.
- Holldobler, B., & Wilson, E. O. (1990). *The Ants, Cambridge, Mass.* Harvard University Press.
- Hwang, Y. K., & Ahuja, N. (1992). Gross Motion Planning - A Survey. *ACM Computing Survey* , vol. 124 (no.3), pp. 219–291.
- Ishiguro, A., Watanabe, Y., Kondo, T., & Uchikawa, Y. (1996). Decentralized consensus-making mechanisms based on immune system. Application to a behavior arbitration of an autonomous mobile robot. *Internationa- tional Conference on Evolutionary Computation*, (pp. 82-87).
- Jerne, N. K. (1974). Towards a network theory of the immune system. *Ann. Immunol* , vol. 125C, 373–389.
- Kim, K. -J., Yoo, J. -O., & Cho, S. -B. (2005). Robust inference of Bayesian networks using speciated evolution and ensemble. *ISMIS* , 92–101.
- Krishna, K. M., & Kalra, K. (2000). Solving the local minima problem for a mobile robot by classification of spatio-temporal sensory sequences. *Journal of Robotic Systems* , vol. 17, 549–564.
- Kube, C. R., & Bonabeau, E. (2000). Cooperative Transport By Ants and Robots. *Robotics and Autonomous Systems*, (pp. 85-101).
- Kuo, B. C., & Golnaraghi, F. (2003). *Automatic Control Systems*. John Wiley & Sons.
- Lumelsky, V. (1991). A comparative study on the path length performance of maze-searching and robot motion planning algorithms. *Robotics and Automation*. vol. 7, pp. 57–66. IEEE Trans.

- Lyons, D. M., & Hendriks, A. J. (1995). Planning as incremental adaptation of a reactive system. *Robotics and Autonomous Systems*, 14 (34), pp. 255-288.
- Matarić, M. J. (1992). Interaction and Intelligent Behavior. *PhD Thesis*. MIT.
- Mayley, G. (1997). Guiding or hiding: Explorations into the effects of learning on the rate of evolution. In P. H. Harvey (Ed.), *Fourth European Conference on Artificial Life*. Cambridge: MIT Press.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.
- Michelan, R., & Von Zuben, F. J. (2002). Decentralized control system for autonomous navigation based on an evolved artificial immune network. *Proceedings of the 2002, Congress on Evolutionary Computation CEC2002*, (pp. 1021–1026).
- Nicolis, S. C., & Deneubourg, J.-L. (1999). Emerging patterns and food recruitment in ants: an analytical study. *Journal of Theoretical Biology*, 198, 575-592.
- Nilsson, N. J. (1986). *Principles of artificial intelligence*. Morgan Kaufmann.
- Nolfi, S., & Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. MIT Press.
- Nolfi, S., Floreano, D., Miglino, O., & Mondada, F. (1994). How to evolve autonomous robots: Different approaches in evolutionary robotics. *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems ArtificialLifeIV*, (pp. 190-197).
- Norman, D., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior. In R. Davidson, G. Schwartz, & D. Shapiro (Eds.), *Consciousness and Self Regulation: Advances in Research and Theory* (Vol. 4, pp. 1-18). New York: Plenum.
- Otis, A. B., Fenn, W., & Rahn, H. (1950). Mechanics of Breathing in Man. *Journal of Applied Physiology* (2), 592-607.
- Parisi, D., & Calabretta, R. (2001). *Evolutionary connectionism and mind/brain modularit*. Institute of Psychology, National Research Council; Rome, Italy, Tech. Rep. NSAL 01-01, 2001.
- Passino, K. (2005). *Biomimicry for Optimization, Control, and Automation*. Springer.
- Pasteels, J. M., Deneubourg, J.-L., & Goss, S. (1987). Self-organization mechanisms in ant societies (I): Trail recruitment to newly discovered food sources. *Experientia Supplementum*, 54, 155-175.
- Pfeifer, R., & Scheier, C. (1999). *Understanding Intelligence*. MIT Press.
- Poty, A., Melchior, P., & Oustaloup, A. (2004). Dynamic path planning for mobile robots using fractional potential field. *Proceedings of the First International Symposium on Control, Communications and Signal Processing*, (pp. 557–561).

- Ram, A., Arkin, R. C., K, M., & Clark, R. J. (1997). Case-based reactive navigation: a method for on-line selection and adaptation of reactive robotic control parameters. *Systems, Man, and Cybernetics*. vol. 27, pp. 376–394. IEEE Trans.
- Ranganathan, A., & Koenig, S. (2003). A reactive robot architecture with planning on demand. *Intelligent Robots and Systems*. 2, pp. 1462-1468. IEEE.
- Reinelt, G. (1990). *TSPLIB - A t.s.p. library*. Technical Report 250, Universität Augsburg, Institut für Mathematik, Augsburg.
- Resnick, M. (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Par-allel Microworlds*. London: MIT Press.
- Roy, N., Gordon, G., & Thrun, S. (2003). Planning under Uncertainty for Reliable Health Care Robotics”, in The Fourth International Conference on Field and Service Robots., (pp. 1–10).
- Rybski, P. E., Larson, A., Veeraraghavan, H., LaPoint, M., & Gini, M. (2004). Communication Strategies in Multi-Robot Search and Retrieval. *7th International Symposium on Distributed Autonomous Robotic Systems* (pp. 301-310). Toulouse, France: Proceedings of DARS'04.
- Sauter, J. A., Matthews, R., Parunak, H. V., & Brueckner, S. (2002). Evolving adaptive pheromone path planning mechanisms In. *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems* (pp. 434–440). Bologna, Italy: ACM Press.
- Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: Detection, search, and attention. *Psychological Review* , 84, 1-66.
- Schulz, D., Burgrad, W., Fox, D., & Cremers, A. (2003). People tracking with mobile robots using sample-based joint probabilistic data association filters. *International Journal of Robotics Research* , vol. 22 (no. 2), 99–16.
- Smolensky, P. (1987). *On the proper treatment of connectionism*. University of Colorado, Dept. of Computer Science, Boulder, CO, Tech. Rep. CU-CS-377-87.
- Sugawara, K., & Watanabe, T. (2002). Swarming robots - Foraging behavior of simple multirobot system. *International Conference on Intelligent Robots and Systems* (pp. 2702-2707). Lausanne, Switzerland: IEEE/RSJ.
- Sutton, R., & Barto, A. (1998). *Reinforcement Learning: An Introduction*. Cambridge, United States of America: MIT Press.
- Svennebring, J., & Koenig, S. (2002). *Building Terrain-Covering Ant Robots*. College of Computing, Georgia Institute of Technology, Atlanta, USA: Technical Report GIT-COGSCI-2002/10.
- Tambe, M., Adibi, J., Al-Onaizan, Y., Kaminka, G. A., & Muslea, I. (1999). Building agent teams using an explicit teamwork model and learning Artificial Intelligence. vol 110, pp. 215–239.

- Varela, F., Coutinho, A., Dupire, B., & Paz, N. (1988). Cognitive networks Immune and neural and otherwise. *Theoretical Immunology: Part Two, SFI Studies in the Sciences of Complexity*, vol. 2, 359–371.
- Vargas, P. A., de Castro, L. N., Michelan, R., & Von Zuben, F. J. (2003). An immune learning classifier network for autonomous navigation. *Proceedings of the Second International Conference on Artificial Immune Systems*, (pp. 69–80).
- Vargas, P., de Castro, L., & Von Zuben, F. J. (2002). Mapping artificial immune systems into learning classifier systems. *IWLCS*, (pp. 163–186).
- Vasilyev, A. (2002). Autonomous agent control using connectionist XCS classifier system. *Transport and Telecommunication*, vol. 3, pp. 56–63.
- Vaughan, R., Stoy, K., Sukhatme, G., & Mataric, M. (2002). Lost: Localization-space trails for robot teams. *IEEE Transactions on Robotics and Automation*, vol 18, pp. 796–812.
- Von Zuben, F. J., & Cazangi, R. R. (2007). Computação Bioinspirada em Controle de Processos. In: L. A. Aguirre (Ed.), *Enciclopédia de Automática - Controle & Automação* (1 ed., Vol. 3, pp. 237-282). São Paulo: Editora Blucher.
- Wagner, I. A., & Bruckstein, A. M. (1997). Cooperative Cleaners: A Case of Distributed Ant- Robotics. In *In Communications, Computation, Control, and Signal Processing: A Tribute to Thomas Kailat* (pp. 289 – 308). Kluwer Academic Publishers, The Netherlands.
- Wang, L., Tan, K. C., & Chew, C. M. (2006). *Evolutionary Robotics: From Algorithms to Implementations*. World Scientific.
- Watanabe, Y., Ishiguro, A., Shirai, Y., & Uchikawa, Y. (1998). Emergent construction of behavior arbitration mechanism based on the immune system. *Proceedings of the 1998* (pp. 481–486). IEEE International Conference on Evolutionary Computation.
- Wurr, A. (2003). Robotic Team Navigation in Complex Environments Using Stigmergic Clues. *MSc Thesis*. University of Manitoba.
- Xiao, J., Michalewicz, Z., Zhang, L., & Trojanowski, K. (1997). Adaptive evolutionary planner/navigator for mobile robots. *IEEE Transaction in Evolutionary Computation*, vol. 1, pp. 18–28.
- Xu, W. L., & Tso, S. K. (1999). Sensor-based fuzzy reactive navigation of a mobile robot through local target switching. *Systems, Man, and Cybernetics, Part C*. vol. 29, pp. 451–459. IEEE Trans.
- Yang, S., Ong, Y., & Jin, Y. (. (2007). *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer.
- Zhang, B.-T., & Kim, S.-H. (1997). An evolutionary method for active learning of mobile robot path planning. *Proceedings of 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, (pp. 312–317). Monterey, CA.

Ziemke, T. (2003). On the Role of Robot Simulations in Embodied Cognitive Science. *AISB Journal*, 1 (4).

Índice de Trabalhos Citados

Aguirre (2007)	6
Arai, Pagello & Parker (2002).....	32, 59, 64
Arkin (1998).....	8, 16, 17, 18, 19, 20, 21, 29, 33, 102
Balch & Arkin (1994)	60, 65
Bates & Elman (2002).....	100, 123
Beckers, Holland & Deneubourg (1994)	67
Bekey (2005)	5, 6, 7, 17, 19, 20, 29, 30, 31, 34
Bonabeau, Dorigo & Theraulaz (1999).....	11, 50, 59, 64, 66, 77
Bonabeau, Theraulaz, Deneubourg, Aron & Camazine (1997).....	62
Booker, Goldberg & Holland (1989).....	10, 73
Brooks (1991)	65
Brooks & Robust (1986)	17
Caetano, Klaus & Zara (2002).....	63, 83
Camazine, Deneubourg, Franks, Sneyd, Theraulaz & Bonabeau (2001).....	60, 62, 64, 66
Cao, Fukunaga & Kahng (1997).....	65
Capozzi (2001)	130, 131, 157, 158
Cazangi (2004).....	37, 39, 52
Cazangi, Von Zuben & Figueiredo (2003).....	115
Chandler, Rasmussen & Pachter (2000)	158
Chen, Fuller, Friedman & Hersh (2005)	129
Chen & Birchfield (2006).....	155
Chien (2000).....	158
Darwin (1859)	10, 30
Dawkins (1997)	9
de Castro (2003).....	11, 101, 105
de Castro & Timmis (2002)	11, 33, 63, 101, 105, 106

de Castro & Von Zuben (2005).....	8
Deb (2001).....	99
Deneubourg, Goss, Franks, SendovaFranks, Detrain & Chretien (1991)	67
Deneubourg, Aron, Goss & Pasteels (1990).....	33, 66
Deneubourg, Pasteels & Verhaeghe (1983).....	64, 88
Deutsch, Harrison & Adams (1985).....	157
Ding, He & Jiang (2003).....	65
Dorf & Bishop (2008)	6
Dorigo & Stützle (2004).....	92
Drexler (1987)	24
Drogoul & Ferber (1992)	66
Dudek & Jenkin (2000).....	5, 6, 19, 130
Edelen (2003)	32, 63, 66
Elshamli, Hussein & Areibi (2004).....	131
Ermentrout & Edelstein-Keshet (1993).....	66
Eysenck (2000)	20
Farmer (1990)	100, 101, 104
Farmer, Packard & Perelson (1986).....	101, 104, 105, 125
Figueiredo (1999).....	7
Fodor & Pylyshyn (1988).....	99
Gaspar & Hirsbrunner (2002).....	104
Ge & Cui (2002).....	130
Grasse (1959)	60, 63
Haykin (1998)	29
Hershberg & Efroni (2001).....	101
Holland (1975).....	10, 29
Holland (1986).....	38, 57, 100
Holland (1995).....	10

Holland & Melhuish (1999).....	60, 67
Holldobler & Wilson (1990)	63, 64
Hwang & Ahuja (1992).....	130
Ishiguro, Watanabe, Kondo & Uchikawa (1996).....	103
Jerne (1974)	101, 125
Kim, Yoo & Cho (2005).....	103
Krishna & Kalra (2000)	99, 103
Kube & Bonabeau (2000).....	67
Kuo & Golnaraghi (2003).....	7
Lumelsky (1991)	103
Lyons & Hendriks (1995)	19
Matarić (1992)	19
Mayley (1997)	31
Michalewicz (1996).....	9, 47, 130
Michelan & Von Zuben (2002)	104
Nicolis & Deneubourg (1999).....	66
Nilsson (1986)	157
Nolfi & Floreano (2000)	9, 23, 30, 31
Nolfi, Floreano, Miglino & Mondada (1994).....	103
Norman & Shallice (1986).....	20
Otis, Fenn, & Rahn (1950).....	34
Parisi & Calabretta (2001).....	103
Passino (2005).....	7, 30
Pasteels, Deneubourg & Goss (1987)	64
Pfeifer & Scheier (1999).....	9, 23, 29
Poty, Melchior & Oustaloup (2004).....	130
Ram, Arkin & Clark (1997).....	99
Ranganathan & Koenig (2003).....	20

Reinelt (1990).....	93
Resnick (1994).....	63
Roy, Gordon & Thrun (2003).....	130
Rybski, Larson, Veeraraghavan, LaPoint & Gini (2004).....	67
Sauter, Matthews, Parunak & Brueckner (2002).....	67
Schneider & Shiffrin (1977).....	20
Schulz, Burgrad, Fox & Cremers (2003)	130
Smolensky (1987).....	99
Sugawara & Watanabe (2002).....	66
Sutton & Barto (1998).....	29
Svennebring & Koenig (2002)	62
Tambe, Adibi, Al-Onaizan, Kaminka & Muslea (1999).....	65
Varela, Coutinho, Dupire & Paz (1988).....	106
Vargas, de Castro, Michelan & Von Zuben (2003).....	104
Vargas, de Castro & Von Zuben (2002).....	104
Vasilyev (2002).....	103
Vaughan, Stoy, Sukhatme & Mataric (2002).....	66
Von Zuben & Cazangi (2007).....	7
Wagner & Bruckstein (1997).....	65
Wang, Tan & Chew (2006)	5
Watanabe, Ishiguro, Shirai & Uchikawa (1998)	103
Wurr (2003).....	68
Xiao, Michalewicz, Zhang & Trojanowski (1997)	131
Xu & Tso (1999).....	103
Yang, Ong & Jin (2007).....	130
Zhang & Kim (1997).....	158
Ziemke (2003)	23