

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

**Sistemas Inteligentes de Navegação Autônoma: Uma Abordagem
Modular e Hierárquica Com Novos Mecanismos de Memória e
Aprendizagem**

PAULO ROBERTO CRESTANI JUNIOR

Orientador: PROF. DR. FERNANDO JOSÉ VON ZUBEN
(DCA – FEEC – Unicamp)

Co-Orientador: PROF. DR. MAURÍCIO FERNANDES FIGUEIREDO
(DIN – UEM)

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Banca Examinadora: Prof. Dr. Ricardo Ribeiro Gudwin (DCA/FEEC/Unicamp)
Prof. Dr. Marconi Kolm Madrid (DSCE/FEEC/Unicamp)
Profa. Dra. Roseli Aparecida Francelin Romero (ICMC/USP-SC)

Campinas
Estado de São Paulo – Brasil
Dezembro de 2001

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

C865s

Crestani Junior, Paulo Roberto

Sistemas inteligentes de navegação autônoma: uma abordagem modular e hierárquica com novos mecanismos de memória e aprendizagem / Paulo Roberto Crestani Junior.--Campinas, SP: [s.n.], 2001.

Orientadores: Fernando José Von Zuben e Maurício Fernandes Figueiredo.

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Robôs móveis. 2. Redes neurais (Computação). 3. Lógica difusa. 4. Algoritmos genéticos. I. Von Zuben, Fernando José. II. Figueiredo, Maurício Fernandes. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Resumo

Este trabalho envolve o desenvolvimento de métodos, modelos e ferramentas que suportem o tratamento conjunto de controle e informação na simulação computacional de sistemas de navegação autônoma dotados de comportamento inteligente. O interesse científico no estudo de sistemas de navegação em ambientes desconhecidos e sujeitos ao atendimento de múltiplos objetivos é motivado basicamente pelo seu evidente potencial em aplicações industriais e pelo fato de demandarem a implementação de estratégias de solução complexas. A proposta básica desta dissertação é implementar novos mecanismos de memória e aprendizagem para superar uma série de limitações verificadas em um sistema de navegação já proposto na literatura, que apresenta uma arquitetura neuro-nebulosa modular e hierárquica, de modo que módulos devotados a comportamentos inatos são coordenados por um módulo caracterizado pela presença de memória e capacidade de aprendizado. A introdução destes novos mecanismos permite uma exploração mais abrangente da informação disponível no ambiente de navegação, a qual é utilizada para melhorar o desempenho do robô, possibilitando a introdução de controle sobre sua velocidade de deslocamento e, por consequência, simulações em ambientes mais complexos contendo obstáculos móveis e múltiplos robôs. O programa desenvolvido para simulação computacional de ambientes virtuais para sistemas de navegação também é parte integrante dos resultados obtidos.

Abstract

This work comprehends the development of methods, models and tools that support the joint treatment of control and information in the computational simulation of autonomous navigation systems with intelligent behavior. The scientific interest in the study of multiobjective oriented navigation systems in unknown environments is mainly motivated by its clear potential in industrial applications and also by the fact that they require the implementation of complex solution strategies. The basic proposal of this dissertation is the creation of new mechanisms of memory and learning to overcome a series of limitations verified in a system previously presented in the literature, composed of a modular and hierarchical neurofuzzy architecture. In this architecture, a module characterized by the presence of memory and learning capability coordinates inborn behavioral modules. The introduction of these new mechanisms permits a broader exploration of the environment information, which is used to improve the robot performance with the introduction of velocity control and, as a result, the execution of simulations in more complex environments, containing mobile obstacles and multiple robots. The program developed for the computational simulation of virtual environments devoted to navigation systems is also part of the obtained results.

Agradecimentos

Agradeço ao professor Fernando J. Von Zuben pela orientação sempre oportuna e pelo incentivo; ao professor Maurício Figueiredo pela valorosa co-orientação e ajuda.

Agradeço ao professor Luiz Carlos Kretly pela elucidação de conceitos e termos técnicos.

Agradeço à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo suporte financeiro (processo número 00/04991-1).

Agradeço à Universidade Estadual de Campinas e particularmente à Faculdade de Engenharia Elétrica e de Computação pela oportunidade de realizar este trabalho.

Por fim, agradeço a meus pais pelo apoio decisivo e a Deus por permitir a conclusão de mais esta etapa de minha vida.

Dedico esta dissertação a meus pais
e a meu irmão.

Índice

CAPÍTULO 1	1
INTRODUÇÃO.....	1
1.1 CARACTERIZAÇÃO DO PROBLEMA.....	1
1.2 HISTÓRICO	2
1.3 METODOLOGIA A SER ESTENDIDA	6
1.4 EXTENSÕES JUNTO À METODOLOGIA ORIGINAL.....	9
CAPÍTULO 2	13
FUNDAMENTOS TEÓRICOS VINCULADOS A FERRAMENTAS DE INTELIGÊNCIA COMPUTACIONAL, APRENDIZADO E SELEÇÃO DE GRUPOS NEURAIS.....	13
2.1 INTRODUÇÃO	13
2.2 REDES NEURAIS ARTIFICIAIS	14
2.2.1 <i>Histórico e visão geral</i>	14
2.2.2 <i>Modelo de aprendizagem de Hebb</i>	19
2.2.3 <i>Modelo de aprendizagem por retropropagação</i>	19
2.2.4 <i>Aplicações de redes neurais artificiais</i>	22
2.3 CONJUNTOS, INFERÊNCIA E SISTEMAS NEBULOSOS	22
2.3.1 <i>Histórico e visão geral</i>	22
2.3.2 <i>Inferência Nebulosa</i>	26
2.3.3 <i>Sistemas Nebulosos</i>	28
2.4 ALGORITMOS GENÉTICOS	30
2.4.1 <i>Histórico e visão geral</i>	30
2.4.2 <i>Operador de recombinação</i>	32
2.4.3 <i>Operador de mutação</i>	33
2.4.4 <i>Critério de avaliação</i>	34
2.4.5 <i>Operador de seleção</i>	35
2.5 CONSIDERAÇÕES SOBRE APRENDIZADO E COMPORTAMENTO	35
2.6 APRENDIZAGEM POR REFORÇO	39
2.6.1 <i>Histórico e visão geral</i>	39
2.6.2 <i>Aprendizagem por diferença temporal</i>	42
2.6.3 <i>Comentários conclusivos para aprendizado por reforço</i>	44
2.7 TEORIA DA SELEÇÃO DE GRUPOS NEURAIS.....	45
2.8 VISÃO SINTÉTICA DO CAPÍTULO 2	46
CAPÍTULO 3	49
SISTEMA DE NAVEGAÇÃO AUTÔNOMO BASEADO EM REDES NEURAIS NEBULOSAS	49
3.1 INTRODUÇÃO	49
3.2 MODELO DO ROBÔ	49

3.3 O CONTROLADOR AUTÔNOMO	50
3.3.1 <i>Arquitetura do controlador</i>	50
3.3.2 <i>Aprendizagem do Módulo de Coordenação</i>	56
3.3.3 <i>Introdução do Valor Acumulado de Ativação Sináptica</i>	58
3.4 SIMULAÇÕES E TESTES	62
3.5 VISÃO SINTÉTICA DO CAPÍTULO 3	69
CAPÍTULO 4	71
O MECANISMO DE CONTROLE DE VELOCIDADE	71
4.1 INTRODUÇÃO	71
4.2 MODIFICAÇÃO DO SENSOR DE RASTREAMENTO DE ALVOS	73
4.3 EXTENSÕES NO MODELO DO CONTROLADOR AUTÔNOMO	75
4.4 SITUAÇÕES E COMPORTAMENTOS ASSOCIADOS	81
4.5 SIMULAÇÕES E TESTES.....	82
4.5.1 <i>Comparação entre modelos</i>	82
4.5.2 <i>Manipulação de ruído nos sensores</i>	90
4.6 VISÃO SINTÉTICA DO CAPÍTULO 4	97
CAPÍTULO 5	99
ESTUDO DE COMPORTAMENTOS INATOS	99
5.1 INTRODUÇÃO	99
5.2 EVOLUÇÃO DE COMPORTAMENTOS DE DESVIO DE OBSTÁCULOS E BUSCA DE ALVOS..	101
5.3 RESULTADOS PARA O MBA	106
5.4 RESULTADOS PARA O MDO	111
5.4 VISÃO SINTÉTICA DO CAPÍTULO 5	115
CAPÍTULO 6	117
APRENDIZAGEM POR RECORDAÇÃO.....	117
6.1 INTRODUÇÃO	117
6.2 A MEMÓRIA DE EVENTOS CRÍTICOS	118
6.3 CRIAÇÃO DE RELAÇÕES ENTRE O MC E A MEC	121
6.4 RECUPERAÇÃO DE ESTÍMULOS CRÍTICOS	125
6.5 SIMULAÇÕES E TESTES	127
6.6 VISÃO SINTÉTICA DO CAPÍTULO 6	133
CAPÍTULO 7	135
COMPORTAMENTOS COMPLEXOS EM AMBIENTES DINÂMICOS	135
7.1 INTRODUÇÃO	135
7.2 INTRODUÇÃO DE OBSTÁCULOS MÓVEIS.....	139
7.3 SIMULAÇÕES E TESTES	140
7.3.1 <i>Simulações com obstáculos móveis</i>	140
7.3.2 <i>Simulações com múltiplos robôs</i>	147
7.4 VISÃO SINTÉTICA DO CAPÍTULO 7	153

CAPÍTULO 8	157
CONSIDERAÇÕES FINAIS	157
8.1 RECAPITULAÇÃO GERAL E CONSIDERAÇÕES ADICIONAIS	157
8.2 AUTONOMIA	160
8.3 APRENDIZAGEM POR REFORÇO.....	162
APÊNDICE A.....	169
SISTEMA DE SIMULAÇÃO COMPUTACIONAL	169
REFERÊNCIAS BIBLIOGRÁFICAS.....	181
ÍNDICE REMISSIVO	191

Lista de figuras

Figura 1.1 – Diagrama simples de um mecanismo de estímulo-resposta.....	3
Figura 1.2 – Situação conflitante entre busca de alvo e desvio de obstáculo.....	7
Figura 1.3 – Arquitetura do sistema de navegação autônomo baseado em redes neurais nebulosas.....	8
Figura 2.1 – Modelo simplificado de neurônio biológico.....	15
Figura 2.2 – Modelo de neurônio artificial.....	16
Figura 2.3 – Funções de ativação não-lineares: (a) função sinal; (b) função sigmoide.....	17
Figura 2.4 – Estrutura de uma rede neural do tipo perceptron multicamadas.....	18
Figura 2.5 – Neurônios adjacentes e sinapse modificada segundo a lei de Hebb.....	19
Figura 2.6 – Modelo de treinamento supervisionado.....	20
Figura 2.7 – Duas funções de pertinência de conjuntos nebulosos representando pessoas de estatura elevada: (a) representação convencional; (b) representação nebulosa.....	24
Figura 2.8 – Operadores básicos de conjuntos nebulosos.....	26
Figura 2.9 – Organização modular de um sistema nebuloso.....	29
Figura 2.10 – Funções de pertinência típicas utilizadas em sistemas nebulosos: (a) curva de Gauss; (b) função triangular.....	30
Figura 2.11 – Diagrama apresentando uma proposta de algoritmo genético.....	32
Figura 2.12 – Dois exemplos de recombinação: (a) simples de um ponto; (b) uniforme.....	33
Figura 2.13 – Dois exemplos de mutação: (a) binária; (b) indutiva.....	34
Figura 3.1 – Modelo do veículo e seus sensores de obstáculo e alvo.....	50
Figura 3.2 – Arquitetura do sistema de navegação autônomo baseado em redes neurais nebulosas.....	51
Figura 3.3 – Arquitetura da rede neural nebulosa, onde a entrada da rede é dada por um padrão unidimensional de distâncias a obstáculos.....	53
Figura 3.4 – Modelo de neurônio nebuloso.....	54
Figura 3.5 – Novo modelo de neurônio nebuloso com VAAS.....	60
Figura 3.6 – Simulação de navegação com o controlador autônomo baseado em redes neurais nebulosas.....	63
Figura 3.7 – Seqüência de valores para os vetores \bar{w}_1 e \bar{s}_1 durante processo de aprendizagem do neurônio 1 do MC.....	64
Figura 3.8 – Resultados comparativos entre controladores: mesma condição inicial e a) sem VAAS; b) com VAAS.....	66
Figura 3.9 – Simulação comparativa em um novo ambiente para os controladores: mesma condição inicial e a) sem VAAS; b) com VAAS.....	67
Figura 3.10 – Resposta do MC sem VAAS: a) antes da primeira colisão; b) antes da segunda colisão.....	68
Figura 3.11 – Resposta do MC com VAAS: a) antes da primeira colisão; b) antes da primeira captura.....	69
Figura 4.1 – Dois modelos de sensores de alvos: a) Modelo utilizado em Figueiredo (1997); b) Modelo com múltiplas antenas, utilizado neste trabalho.....	74
Figura 4.2 – Nova arquitetura do controlador autônomo.....	76
Figura 4.3 – Nova arquitetura do Módulo de Coordenação.....	77
Figura 4.4 – Dois sistemas de navegação distintos: a) velocidade constante = 0,2 ud./i.; b) velocidade variável; c) velocidade constante = 0,7 ud./i.....	83
Figura 4.5 – Padrões sinápticos registrados após a 1ª e 8ª colisões para três neurônios.....	86

Figura 4.6 – Registro da velocidade desenvolvida pelo robô durante a navegação	87
Figura 4.7 – Relação entre risco de colisão e queda da velocidade de deslocamento	87
Figura 4.8 - Situações em que o controlador com velocidade variável leva vantagem sobre o controlador com velocidade constante.....	88
Figura 4.9 – Alvos encostados a obstáculo: a) robô com velocidade variável; b) robô com velocidade constante	89
Figura 4.10 – Três simulações considerando diferentes parâmetros de ruído nos sensores de obstáculos: a) $\rho = 0\%$, $\delta = 0$ ud.; b) $\rho = 10\%$, $\delta = 5$ ud.; c) $\rho = 30\%$, $\delta = 5$ ud.....	92
Figura 4.11 – Colisão lateral: a) $\rho = 0\%$; $\delta = 0$ ud.; b) $\rho = 10\%$, $\delta = 5$ ud.;.....	93
c) $\rho = 30\%$, $\delta = 5$ ud.	93
Figura 4.12 – Quatro situações considerando diferentes parâmetros de ruído para os sensores de alvos: a) $\rho = 0$, $\delta' = 0$; b) $\rho = 10\%$, $\delta' = 0,05$; c) $\rho = 15\%$, $\delta' = 0,05$;.....	94
d) $\rho = 10\%$, $\delta' = 0,04$	94
Figura 4.13 – Alvo atingido frontalmente: a) $\rho = 0\%$, $\delta' = 0,05$; b) $\rho = 15\%$, $\delta' = 0,05$	95
Figura 4.14 – Estudo dos efeitos do ruído nos sensores de obstáculos e nos sensores de alvos: a) sem ruído; b) $\rho = 10\%$ para sensores de alvos, $\rho = 30\%$ para sensores de obstáculos, $\delta = 5$ ud., $\delta' = 0,04$	96
Figura 5.1 – Arquitetura dos módulos inatos MBA e MDO.....	102
Figura 5.2 – Evolução do MBA com posições iniciais fixas: a) ambiente de evolução;....	107
b) Curvas evolutivas; c) navegação pós-evolutiva.....	107
Figura 5.3 – Simulação com Módulo de Busca de Alvos evoluído.....	108
Figura 5.4 – Curvas evolutivas considerando variações aleatórias na topologia do ambiente de treinamento durante a evolução.....	109
Figura 5.5 – Simulações considerando variações aleatórias na topologia do ambiente de treinamento durante a evolução	110
Figura 5.6 – Simulações com MBA evoluído e introdução de perturbações na entrada: (a) $\rho=10\%$, $\delta=180^\circ$; (b) $\rho=50\%$, $\delta=85^\circ$; (c) $\rho=100\%$, $\delta=20^\circ$	111
Figura 5.7 – Evolução do MDO: (a) primeira simulação após a evolução; (b) evolução com Módulo de Coordenação treinado; (c) Curvas evolutivas do MDO	112
Figura 5.8 – Simulações com o MDO evoluído	113
Figura 5.9 – Perturbação nos sensores de obstáculos: (a) $\rho = 0\%$ e $\delta = 0$ u.d.; (b) $\rho = 10\%$ e $\delta = 10$ u.d.; (c) $\rho = 15\%$ e $\delta = 5$ u.d.	113
Figura 5.10 – Simulação com MDO e MBA evoluídos: (a) curvas de avaliação do MDO; (b) curvas de avaliação do MBA, (c) navegação	114
Figura 6.1 – Esquema parcial do controlador autônomo considerando a MEC	119
Figura 6.2 – Novo modelo de neurônio do MC	120
Figura 6.3 – Modelo de neurônio da MEC	121
Figura 6.4 – Estímulos após uma colisão frontal: a) registrados por neurônios do MC; b) por um neurônio da MEC	123
Figura 6.5 – Relações sinápticas entre o MC e a MEC após diversas colisões	125
Figura 6.6 – Comparação entre controladores: a) sem aprendizagem por recordação; b) com aprendizagem por recordação	128
Figura 6.7 – Robô sem aprendizagem por recordação em percurso longo.....	130
Figura 6.8 – Robô com aprendizagem por recordação em percurso longo	131
Figura 6.9 – Três configurações de um neurônio do MC: a) após sua criação;.....	133
b) após segunda aprendizagem por colisão; c) após aprendizagem por recordação.....	133

Figura 7.1 – Simulações com obstáculos móveis lentos.....	141
Figura 7.2 – Simulação com obstáculos móveis rápidos	142
Figura 7.3 – Simulação com obstáculos de movimento aleatório rápido	143
Figura 7.4 – Simulação com obstáculo rápido atraído pelo robô	144
Figura 7.5 – Simulação com obstáculo lento, atraído pelo robô.....	145
Figura 7.6 – Simulação no ambiente <i>prisão-aberta</i>	146
Figura 7.7 – Múltiplos robôs navegando em formação	147
Figura 7.8 – Robôs com velocidades constantes iguais a 0,6 ud./i.: a) período de aprendizagem com 19 colisões; b) colisões entre robôs treinados.....	148
Figura 7.9 – Robôs com velocidades variáveis: a) período de aprendizagem com 73 colisões; b) navegação sem colisões entre robôs treinados.....	150
Figura 7.10 – Desempenhos do sistema frente a mínimos locais: a) sucesso com um robô; b) fracasso com um robô; c) múltiplos robôs dos quais um obtém sucesso	151
Figura 7.11 – Simulação com seis robôs em ambiente com mínimo local do qual três conseguem escapar.....	153
Figura 8.1 – Generalização do conhecimento: a) ambiente de treinamento;.....	165
b) navegação com robô treinado em (a).....	165
Figura 8.2 – Influência da topologia sobre a navegação: a) corredores estreitos;	166
b) espaços alternados	166
Figura A.1 – Ambiente de navegação em procedimento de simulação.....	170
Figura A.2 – Diagrama de blocos do Ambiente de Simulação.....	171
Figura A.3 – Edição das propriedades de um obstáculo	172
Figura A.4 – Configuração dos módulos de nível hierárquico inferior: a) MDO; b) MBA	172
Figura A.5 – Propriedades do robô: a) propriedades padrão; b) quantidade e distribuição dos sensores de obstáculos; c) quantidade dos sensores de alvos;.....	173
d) manipulação de ruído nos sensores	173
Figura A.6 – Representação gráfica da simulação.....	176
Figura A.7 – Dados numéricos sobre a simulação.....	176
Figura A.8 – Quadro de observações sobre a simulação	177
Figura A.9 – Dois momentos de aprendizagem após captura para um neurônio do	177
Módulo de Coordenação	177
Figura A.10 - Dois momentos de aprendizagem de desvio em um neurônio do	178
Módulo de Coordenação	178
Figura A.11 – Resposta do controlador antes de colisão	178
Figura A.12 – Resposta do controlador antes de captura de alvo.....	179

Capítulo 1

Introdução

1.1 Caracterização do problema

O desenvolvimento de novas tecnologias segue diretamente as aspirações da humanidade por critérios progressivamente mais rígidos de desempenho (FIGUEIREDO, 1997). Sistemas capazes de automatizar processos em que a presença humana não seja essencial apresentam, portanto, um apelo muito interessante do ponto de vista científico. Neste contexto, um sistema autônomo pode ser caracterizado pela capacidade que possui de executar a tarefa para a qual foi projetado sem a interferência de qualquer mecanismo de controle externo (FABRO, 1996).

SMITHERS (1997) apresenta uma definição ainda mais abrangente para o termo “autonomia”, quando aplicado a um sistema de controle artificial. De acordo com esta definição, “autonomia” se refere não apenas à capacidade de ação e decisão independente de influências externas, mas sobretudo à capacidade de autoprodução do mecanismo que gera a ação. Em outras palavras, um sistema autônomo deve ser capaz de sintetizar suas próprias leis de ação sobre o ambiente.

Particularmente, sistemas de navegação autônomos devem ser capazes de definir uma seqüência de ações a serem tomadas por robôs móveis, dotados de um conjunto limitado de sensores, quando expostos a um ambiente externo suposto desconhecido e tendo que atender simultaneamente a um elenco de objetivos previamente especificados (FIGUEIREDO, 1999). Os veículos auto-guiados resultantes criam grandes expectativas quanto às possibilidades de aplicação prática, inclusive industriais. Muito embora seres vivos com capacidade motora não encontrem grandes dificuldades em gerar trajetórias eficazes e tomar decisões nos mais variados ambientes, dotar sistemas artificiais com tais habilidades não é uma tarefa elementar (COLOMBETTI *et al.*, 1996).

A pesquisa em torno do desenvolvimento de sistemas de navegação autônomos é intensa e apresenta diversas abordagens. Propostas baseadas em Inteligência Artificial têm apresentado resultados promissores, sendo que as propriedades que permitem caracterizá-los como sistemas autônomos inteligentes são: o emprego de técnicas de aprendizado na aquisição de conhecimento e adaptação de comportamento a partir da interação com o ambiente; a capacidade de encontrar soluções factíveis, e muitas vezes ótimas, que atendam simultaneamente a múltiplos objetivos, possivelmente conflitantes, utilizando o conhecimento adquirido; e a habilidade para operar em condições adversas, tais como: ausência de um conjunto de informações completo para um planejamento prévio de seu comportamento, imprevisibilidade na interação com o ambiente, suposto ser de topologia desconhecida, ruído nos sensores e atuadores.

1.2 Histórico

Os primeiros experimentos envolvendo mecanismos de navegação para robôs móveis datam dos anos 60. O robô Shakey, desenvolvido no *Stanford Research Institute* (NILSSON, 1969) é um dos membros desta primeira geração. Projetado para navegar em um ambiente com objetos demarcados especialmente para auxiliar o sistema de sensoriamento, Shakey tomava suas decisões de navegação através de processos de inferência simbólica e as leituras sensoriais eram utilizadas para a construção e manutenção de seu modelo interno do ambiente de navegação.

Assim como Shakey, outros robôs com sistemas de navegação baseados em regras de inferência simbólica foram concebidos nas primeiras décadas de pesquisa na área (ARKIN, 1998), como o Hilare em 1977 no *Laboratoire d'Automatique et d'Analyse des Systèmes*, na França (GIRALT *et al.*, 1984), o Cart, em Stanford (MORAVEC, 1977) e o Rover na *Carnegie-Mellon University* (MORAVEC, 1983). O método de raciocínio destes primeiros sistemas era deliberativo, pois se baseava na construção de uma representação interna do mundo, utilizada posteriormente no planejamento do melhor caminho de navegação para que o seu objetivo fosse atingido.

Os principais problemas apresentados por esta abordagem são o custo de manutenção da representação interna do mundo e a quantidade de informação necessária para construí-la. Especialmente quando se trata de um sistema baseado em regras de inferência simbólica, uma base expressiva de regras deve ser mantida para que as mais diversas situações com as quais o sistema venha a se deparar sejam tratadas. Em grande parte dos problemas, dentre os quais a navegação de robôs, a codificação de uma base razoavelmente completa exigiria um grande esforço de implementação, sem garantia de sucesso.

Com base nestas limitações de sistemas baseados em regras, quando aplicados ao controle de robôs móveis, uma nova arquitetura para controladores autônomos foi desenvolvida em meados nos anos 80 no *Massachusetts Institute of Technology* (BROOKS, 1986). Batizada por seu criador Ronald Brooks de *Subsumption Architecture*, esta nova técnica baseia-se na exploração dos comportamentos que devem ser exibidos pelo sistema em determinadas situações caracterizadas pelas leituras sensoriais, e a resposta do sistema deve ser uma consequência imediata do comportamento assumido, ao invés de basear-se em uma cadeia de inferências simbólicas. A Figura 1.1 ilustra como um estímulo se converte em resposta diretamente através de um comportamento de acordo com esta abordagem.

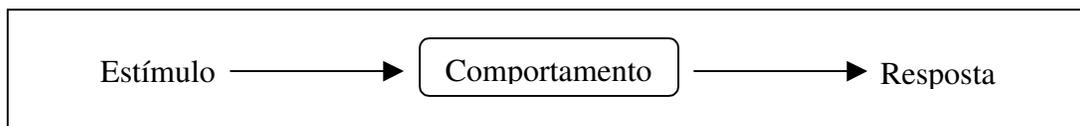


Figura 1.1 – Diagrama simples de um mecanismo de estímulo-resposta

Na Arquitetura de Assunção, tarefas simples são realizadas através de comportamentos simples (e.g. navegar em frente, virar para a esquerda, virar para a direita). Tarefas mais elaboradas requerem uma conjunção de comportamentos simples com comportamentos mais especializados (e.g. seguir uma parede, buscar um alvo). Neste caso, através de uma estrutura de processamento que realiza o acoplamento dos vários estímulos e comportamentos associados, um comportamento pode ter seu estímulo de entrada inibido ou sua resposta suprimida por outro comportamento com mais prioridade em determinada situação.

O modelo criado por Brooks inclui-se em uma classe de sistemas denominados sistemas baseados em comportamentos, que ao contrário dos sistemas deliberativos, não necessitam de uma representação interna do mundo com o qual interagem, pois suas respostas são expressões imediatas de comportamentos, que no caso da Arquitetura de Assunção, correspondem a estruturas pré-programadas pelo projetista. MATARIC (2001) apresenta uma distinção controversa entre modelos baseados em comportamento e modelos reativos. ARKIN (1998) por sua vez, considera modelos baseados em comportamentos como uma classe de modelos reativos.

Mataric também considera que sistemas híbridos sejam aqueles que possuam características deliberativas e características reativas. Neste sentido, restringe a definição de sistemas híbridos no que se refere a sistemas de inteligência computacional e desconsidera outros modelos apresentados na literatura (KASABOV, 1996; VENTURA & MARTINEZ, 1998; MCGARRY *et al.*, 1999).

Diversos outros modelos de robótica baseados em comportamentos têm sido propostos (ALBUS *et al.*, 1982, ARKIN, 1987; KAEHLING & ROSENSCHEIN, 1991; MAES, 1990; CONNELL, 1989; FIRBY, 1989; ROSENBLATT & PAYTON, 1989, ZELTZER & JOHNSON, 1991; KAHN, 1991, GRUPEN & HENDERSON, 1990, ANDERSON & DONATH, 1991; LIM, 1994; SANBORN, 1988; BEKEY & TOMOVIC, 1986; MILLER, 1995; FUKUDA & KUBOTA, 1999), carregando graus diferentes de embasamento em teorias de comportamento natural.

Uma característica marcante dos trabalhos desenvolvidos nesta linha de pesquisa corresponde à importância dada à implementação dos modelos propostos em ambientes reais, como critério de validação de resultados. Esta particularidade, que chega a se tornar uma obsessão em alguns casos, é em parte compreensível pelo fato de que uma das principais motivações para o surgimento de modelos baseados em comportamento tenha se devido a críticas sobre o fraco relacionamento entre os modelos de representação de conhecimento da inteligência artificial simbólica e seus ambientes físicos.

Por um lado, é um fato um tanto óbvio que resultados obtidos em uma simulação computacional não possam ser completamente estendidos a situações reais. Em geral, as diferenças entre estes dois casos decorrem do fato de ser muito difícil capturar em um modelo simulado toda a riqueza de detalhes de um ambiente real, o que compreende entre outras coisas, a presença de ruído nos sensores, atuação de forças físicas, etc. Neste sentido, o custo de implementação de um ambiente de simulação que capture todas as características de um ambiente real não justifica sua implementação em detrimento de um experimento prático.

Por outro lado, em sistemas de engenharia, freqüentemente a reprodução minuciosa de todos os detalhes de um sistema físico não é crucial para que conclusões preliminares de boa qualidade possam ser inferidas a seu respeito. Isto explica o fato do uso de protótipos e simuladores ser tão disseminado, sendo usados desde teste estrutural de componentes de aeronaves até o treinamento de astronautas para missões espaciais.

Não obstante, os custos de uma implementação prática nestes casos, sem o embasamento dos resultados preliminares de uma simulação, são pouco atraentes. No que diz respeito a modelos conceituais de sistemas de navegação autônomos, muitas vezes simulações também servem como base para conclusões que levem a boas previsões sobre o comportamento do sistema real equivalente. Além disso, se um sistema não for robusto o suficiente para tolerar, com poucas modificações, a migração de uma simulação para uma implementação prática, sua capacidade de autonomia merece ser revista.

Particularmente no que se refere ainda à abordagem de assunção proposta por Brooks, uma das principais limitações de seus primeiros modelos era a ausência de um mecanismo de aprendizagem que conferisse ao sistema adaptabilidade a mudanças ambientais e geração automática de comportamentos conforme eram necessários. Esta deficiência parece solucionada em modelos mais recentes MATARIC (2001). Nesta linha, as redes neurais artificiais oferecem potencialidades no que se refere à aprendizagem (VERSHURE, 1998), sistemas nebulosos revelam grande praticidade na representação de regras heurísticas (LI, 1994), e algoritmos genéticos proporcionam excelente robustez (DONNART & MEYER,

1996). Estas características têm levado as abordagens baseadas em comportamentos a se tornarem cada vez mais híbridas (LI *et al.*, 1997; IWAKOSHI *et al.*, 1998; CARVALHO & SOUZA, 1998; FLOREANO & MONDADA, 1998; MCFETRIDGE & IBRAHIM, 1998; CHUNG *et al.*, 2000; SONG & SHEEN, 2000)

1.3 Metodologia a ser estendida

Seguindo estes passos, o objetivo básico deste trabalho é desenvolver novos mecanismos de memória e aprendizagem junto a um sistema de navegação autônomo inteligente baseado em comportamentos que represente uma extensão da proposta apresentada por FIGUEIREDO (1997), desenvolvida de modo a permitir a navegação em ambientes desconhecidos, e com o propósito de atender simultaneamente a múltiplos objetivos.

A abordagem de FIGUEIREDO (1997) é inovadora e eficaz, por ser concebida com base em técnicas avançadas de redes neurais artificiais e sistemas nebulosos, capazes de realizar processamento híbrido, com etapas de natureza numérica e simbólica. A arquitetura do sistema é hierárquica e dividida em três módulos principais: dois devotados à tomada de decisão e um terceiro à coordenação. Este trabalho defende a manutenção desta arquitetura básica, mas propõe novas formas de operação interna dos módulos, e extensões junto às etapas de coordenação e aprendizado. A motivação está na necessidade de superar limitações importantes já verificadas na operação do sistema autônomo resultante da implementação da abordagem.

A estratégia de navegação da proposta de FIGUEIREDO (1997) é desenvolvida à medida que as interações com o ambiente vão se sucedendo. Inicialmente, o sistema apresenta apenas comportamentos instintivos, os quais são progressivamente refinados com a aprendizagem desencadeada pelos diferentes estímulos que recebe do ambiente. Esta proposta se baseia na Teoria de Seleção de Grupos Neurais (EDELMAN, 1987), que considera que conhecimentos instintivos para a execução de uma determinada tarefa estejam presentes em seus modelos antes mesmo de iniciada a sua interação com o ambiente.

A arquitetura utilizada é modular e hierárquica. Em um primeiro nível, dois módulos distintos atuam na tomada de decisão, sendo um deles responsável pela busca de alvos e o outro responsável pelo desvio de obstáculos. Entretanto, as respostas produzidas por tais repertórios podem ser conflitantes. Por exemplo, considerando-se o ambiente ilustrado na Figura 1.2, o módulo de busca de alvos determinaria que o robô deveria avançar na direção do alvo, enquanto que o módulo de desvio de obstáculos determinaria que o robô deveria deslocar-se na direção contrária à do alvo.

Para tratar este e outros eventuais conflitos, um terceiro módulo, situado em um nível hierárquico superior, realiza automaticamente a tarefa de coordenação sobre os demais, ponderando a influência de cada um dos dois módulos de tomada de decisão, de acordo com a situação com que o robô se depara. Inicialmente, o módulo de coordenação não possui conhecimento, o qual vai sendo adquirido por aprendizado, à medida que o robô interage com o ambiente, e sempre se levando em conta o elenco de objetivos a serem atendidos.

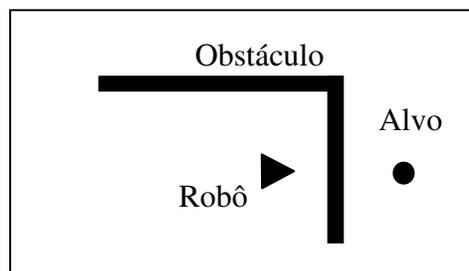


Figura 1.2 – Situação conflitante entre busca de alvo e desvio de obstáculo

A Figura 1.3 ilustra a arquitetura do sistema autônomo de navegação. A estrutura dos módulos é descrita em detalhes em OLIVEIRA *et al.* (1994), e envolve redes neurais nebulosas. O número de neurônios no módulo de coordenação é determinado durante a navegação, pois a cada colisão um processo de aprendizagem é realizado, dando origem a um novo neurônio. O número de sinapses dos neurônios da primeira camada varia de acordo com o número de sensores de obstáculo.

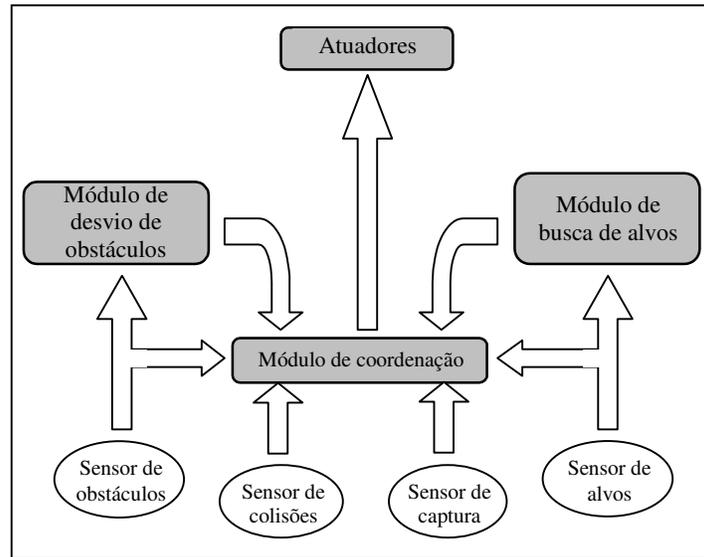


Figura 1.3 – Arquitetura do sistema de navegação autônomo baseado em redes neurais nebulosas

A estratégia de aprendizagem adotada para o módulo de coordenação obedece à teoria do condicionamento operante, freqüentemente observada em sistemas biológicos (THORNDIKE, 1911). Ao colidir, o robô manifesta-se instintivamente, alterando sua direção em 180°. Neste momento, o sinal transmitido pelo sensor de colisão fará o papel de um agente de punição positiva, conforme será apresentado no capítulo 2, que tenderá a fortalecer a relação estímulo-resposta determinada pelos sinais dos sensores de obstáculo e pelo sinal do sensor de colisão.

Após algumas colisões, o módulo de desvio de obstáculos passa a associar classes de padrões capturados pelos sensores de obstáculos a situações de colisão iminente, gerando, então, comportamentos correspondentes ao desvio de obstáculo. De maneira análoga, a cada vez que um alvo é capturado, um reforço positivo da relação entre os sinais dos sensores de alvo e o sinal do sensor de captura ocorre junto ao módulo de busca de alvos. Quando há uma colisão ou uma captura de alvo, os neurônios do módulo de coordenação têm suas intensidades sinápticas devidamente ajustadas.

1.4 Extensões junto à metodologia original

No modelo de FIGUEIREDO (1997), o Sistema de Navegação Autônomo do robô utiliza seu mecanismo de aprendizagem apenas em duas situações: quando ocorre uma colisão, os neurônios do módulo de coordenação responsáveis pela ponderação da ação de desvio de obstáculos são punidos; quando ocorre uma captura de alvo, o neurônio do módulo de coordenação responsável pela ponderação da ação de busca de um alvo é recompensado.

Com isto, as estratégias de recompensa e punição são manipuladas de maneira desequilibrada pelo sistema. Os neurônios de ponderação da ação de desvio nunca são recompensados por uma manobra bem sucedida do robô, restando-lhes apenas as punições decorrentes de uma colisão. Com o passar do tempo, o excesso de punições leva o robô a desenvolver uma espécie de aversão excessiva à aproximação de obstáculos. Por outro lado, o neurônio responsável pela ponderação da ação de busca de alvos nunca é punido por ter influenciado em uma decisão que leva o robô a uma colisão, recebendo apenas a recompensa por cada alvo capturado. O excesso de recompensas concedidas a cada captura de um alvo reforça sobremaneira a influência do neurônio na decisão final do módulo de coordenação. Ocorre então que, mesmo diante de um obstáculo, a resposta dos neurônios de ponderação da ação de desvio já não é suficiente para evitar uma colisão.

Para eliminar estas deficiências, é preciso que o mecanismo de aprendizagem do robô distribua de maneira mais apropriada, tanto as punições quanto as recompensas, sobre as sinapses do módulo de coordenação. Trata-se de resolver um problema de designação de créditos. Em outras palavras, a punição ou recompensa deve tornar-se de algum modo proporcional à responsabilidade de cada neurônio ao longo do processo de tomada de decisão. Para tanto, propõe-se uma memória de ativação, ou seja, um meio de indicar a ocorrência de uma ativação de determinada sinapse instantes após esta ter ocorrido. Neste trabalho, esta memória é modelada incorporando-se uma nova propriedade sináptica, denominada valor acumulado de ativação sináptica (VAAS).

Outra questão fundamental é decidir que tipo de propriedades do sistema autônomo devem ser projetadas previamente (propriedades inatas) e quais delas devem ser aprendidas por

interação com o ambiente e/ou outros sistemas autônomos (propriedades não-inatas). Uma vez definido o conjunto de propriedades não-inatas, é necessário definir como implementá-las por processo de aprendizagem ou aquisição de conhecimento. Estes aspectos serão tratados explicitamente neste trabalho.

A atuação conjunta na definição da direção e velocidade de deslocamento também é fundamental para fornecer flexibilidade ao sistema de navegação de modo a permitir uma operação mais refinada em ambientes estáticos, e uma operação bem-sucedida em ambientes com obstáculos móveis e/ou com múltiplos robôs. Isto irá requerer a inclusão de uma saída adicional de atuação, já que o modelo original de Figueiredo (1997) não considerava a possibilidade de variar a velocidade.

Mais um aspecto evidente é o fato de que os processos de aprendizagem do robô se encontram muito restritos a situações excessivamente bem definidas, quais sejam: colisões e capturas de alvos. Em sistemas biológicos evoluídos (e.g. seres humanos), a aquisição de conhecimento acompanha a interação constante do indivíduo com o ambiente que o cerca, não se limitando a uma aprendizagem condicionada através de estímulos hedônicos primitivos, (e.g. aqueles que garantam sobrevivência e reprodução). A aquisição de conhecimento pode ocorrer, por exemplo, quando refletimos e raciocinamos utilizando um processo dedutivo (CUSSINS, 1990).

Experimentos em neurologia indicam que a aquisição de novos conhecimentos por parte de um indivíduo possui íntima relação com o grau de familiarização deste indivíduo ao estímulo a ele apresentado. Em outras palavras, os processos de aprendizagem seriam disparados mais intensamente mediante a exposição do indivíduo a eventos novos e tenderiam a se tornar menos intensos à medida que estes eventos se tornassem monótonos ou familiares (MILLER *et al.*, 1991).

Neste sentido, um outro avanço a ser incorporado junto ao sistema proposto por FIGUEIREDO (1997) consiste na expansão das situações que disparem a aprendizagem do sistema, dotando o robô de capacidade de aprender com o ambiente sem a necessidade de

uma colisão ou uma captura. Este processo de aprendizagem exige que o robô reconheça as situações que lhe oferecem perigo e que levam ao atendimento de objetivos previamente definidos. Estas situações correspondem àquelas que levaram a colisões e a capturas de alvo, respectivamente. Para tanto, este trabalho propõe a especificação de uma memória de eventos críticos (MEC).

O que se pretende com a MEC é fazer com que o robô antecipe a possibilidade de colisão ou captura de alvo, e assim possa indicar uma seqüência de ações mais eficazes em cada caso. Pela lei de HEBB (1949), é possível fortalecer as correlações existentes entre eventos e ativações de neurônios. Este é o princípio que será explorado aqui na concepção da MEC. É importante notar que todos os processos de aprendizagem propostos por FIGUEIREDO (1997) são mantidos, de forma que a aprendizagem sem colisão não os substituirá, mas apenas os complementarará.

Por fim, é interessante se considerar que em um determinado ambiente, podem co-existir diversos robôs, iguais ou diferentes em suas habilidades e capacidades sensoriais. Com isso, além de se estudar as características particulares que regem o funcionamento de cada sistema de navegação (um implementado em cada robô), pode-se analisar o comportamento destes integrados ao ambiente. Assim, o ambiente pode ser entendido como uma sociedade de sistemas autônomos, que pode apresentar comportamentos sofisticados tais como competição e/ou cooperação, de forma deliberada ou não. Por exemplo, BEER (1995) adotou uma perspectiva de sistemas dinâmicos para mostrar que a interação do ambiente com um ou mais robôs pode ser descrita na forma de um sistema de equações diferenciais acopladas.

Os demais capítulos desta dissertação encontram-se organizados da seguinte maneira: o capítulo 2 é dedicado à apresentação dos fundamentos teóricos necessários para a compreensão da proposta deste trabalho. No capítulo 3, o sistema de navegação proposto por FIGUEIREDO (1997) é apresentado, juntamente com a extensão referente ao VAAS. O capítulo 4 introduz o mecanismo de controle de velocidade, e considera mudanças junto à arquitetura original do sistema apresentada no capítulo 3. Um estudo sobre as estruturas

inatas que compõem o sistema de navegação do robô é desenvolvido ao longo do capítulo 5. A especificação da MEC e do mecanismo de aprendizagem por recordação são realizados ao longo do capítulo 6. No capítulo 7, considera-se a ocorrência de obstáculos móveis e de múltiplos robôs em um ambiente de navegação. Por fim, o capítulo 8 é reservado para discussão e considerações finais do trabalho. No Apêndice A, serão descritas as principais funções do sistema de simulação utilizado para a geração dos resultados apresentados ao longo do texto.

Capítulo 2

Fundamentos Teóricos vinculados a ferramentas de inteligência computacional, aprendizado e seleção de grupos neurais

2.1 Introdução

Conforme apresentado no capítulo 1, o estado atual da pesquisa sobre navegação autônoma de robôs móveis alimenta um fluxo de pesquisa considerável, sendo que este fluxo acompanha diretamente a sofisticação dos recursos computacionais disponíveis. Se de um lado a capacidade de processamento é fator importante no tratamento de problemas complexos, tais como a navegação de robôs, de outro lado o desenvolvimento de modelos lógico-matemáticos capazes de otimizar e fornecer flexibilidade ao seu processo de solução é fundamental.

Neste sentido, sistemas biológicos têm servido ao longo das últimas décadas como inspiração para a concepção de diversos mecanismos de otimização. A motivação para esta inspiração está na forma como tais sistemas encontram soluções eficientes e robustas para problemas de grande complexidade, forma esta freqüentemente associada à capacidade de aquisição de conhecimento extraído do próprio problema. Esta forma de aquisição de conhecimento, ou aprendizagem, promove uma adaptação que possibilita diminuição considerável nos custos associados à caracterização a priori do problema em questão.

Mecanismos de processamento de informação que atendam a estas condições são denominados Sistemas Inteligentes, os quais podem apresentar diversos graus de afinidade com seu modelo de inspiração encontrado na natureza, variando desde uma simples motivação até a uma reprodução tão fiel quanto possível das características originais da versão natural. Dentre as classes de Sistemas Inteligentes, destaca-se no âmbito deste trabalho a Inteligência Computacional, que contempla algumas das principais técnicas

empregadas na implementação do sistema autônomo proposto no presente trabalho: Redes Neurais Artificiais, Algoritmos Genéticos e Sistemas Nebulosos.

Este capítulo apresenta os fundamentos necessários para o entendimento de algumas destas técnicas de Inteligência Computacional. Também serão consideradas algumas idéias e teorias oriundas de estudos em psicologia e em biologia, utilizadas como subsídio para o modelo de navegação autônoma apresentado no capítulo 3.

2.2 Redes Neurais Artificiais

2.2.1 Histórico e visão geral

A forma de processamento realizada pelo cérebro humano é notadamente diferente daquela realizada por uma máquina digital construída a partir de uma arquitetura do tipo Von Newman, em que a memória e a unidade de processamento de dados correspondem a duas entidades distintas e o processamento é realizado serialmente. No cérebro, o processamento de dados é essencialmente paralelo, ocorrendo simultaneamente de forma distribuída nas células nervosas, estimadas em aproximadamente 10 bilhões. Além disso, a memória também é distribuída entre as células nervosas, localizando-se nas sinapses, estimadas em aproximadamente 60 trilhões no córtex humano (HAYKIN, 1999).

A Figura 2.1 ilustra uma célula nervosa humana. Existem dezenas de tipos diferentes de neurônios catalogados na literatura, mas, em geral, um neurônio apresenta: um corpo celular; alguns milhares de dendritos, através dos quais a célula recebe estímulos elétricos provindos de células nervosas adjacentes; um axônio, através do qual a célula dispara um estímulo elétrico sob certas condições; e sinapses, que são comunicações inter-celulares presentes nos dendritos e nas terminações das ramificações do axônio. O *disparo* produzido por um neurônio ocorre após a atividade elétrica interna da célula exceder um determinado limiar de ativação, gerado pela carga recebida a partir dos dendritos. Quando isto ocorre, uma série de pulsos elétricos percorre o axônio e vai estimular as células pós-sinápticas conectadas à célula que efetuou o disparo.

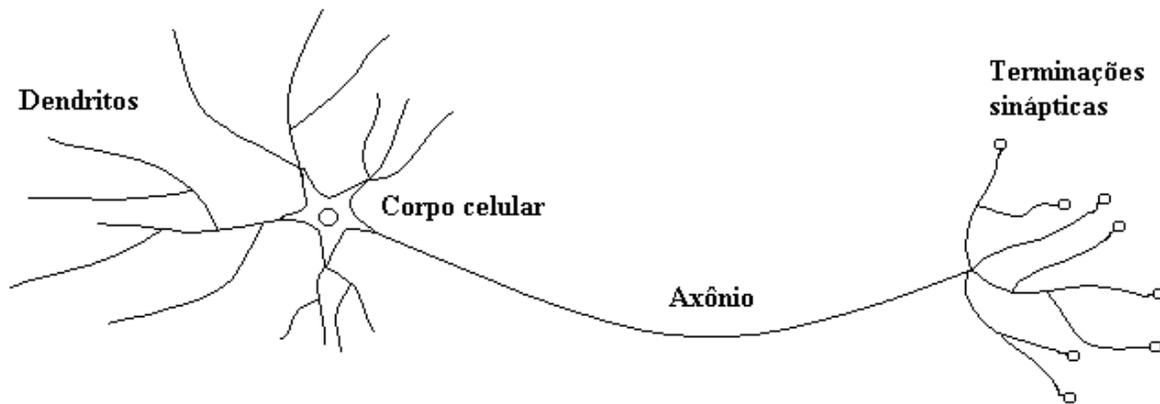


Figura 2.1 – Modelo simplificado de neurônio biológico

A velocidade de processamento no interior de uma célula nervosa é algo em torno de seis ordens de magnitude menor que a velocidade de uma porta lógica de silício, entretanto, devido à complexa rede de conexões entre as células nervosas, o cérebro é capaz de executar atividades que nenhuma máquina jamais pôde realizar sequer de forma aproximada. Mesmo tarefas simples para o sistema nervoso humano, como a coordenação motora, são desafios ainda não solucionados em máquinas digitais.

A capacidade de aprendizagem é outra característica marcante do sistema nervoso humano. Ao adquirir conhecimento a partir de suas experiências com o ambiente que o envolve, o sistema é capaz de melhorar seu desempenho ao longo da vida do indivíduo e aumentar suas chances de sobrevivência no meio que o cerca. Além disto, é capaz de adaptar-se a novas situações ou a novas condições ambientais e mesmo moldar-se para realizar novas funções as quais requeiram um processo de aprendizagem interativa (MENDELECK, 1995).

Considerando as características notáveis do mecanismo de processamento nervoso, o psiquiatra W. McCulloch e o matemático W. Pitts propuseram, em 1943, um trabalho pioneiro que marcou o início das pesquisas na área das Redes Neurais Artificiais (MCCULLOC & PITTS, 1943). Em seu artigo, os autores propõem o primeiro modelo matemático de neurônio artificial (Figura 2.2). Com um número suficiente de unidades básicas e conexões sinápticas adequadamente ajustadas, os pesquisadores sugeriram que uma rede assim formada poderia sintetizar, em princípio, qualquer função computável (HAYKIN, 1999).

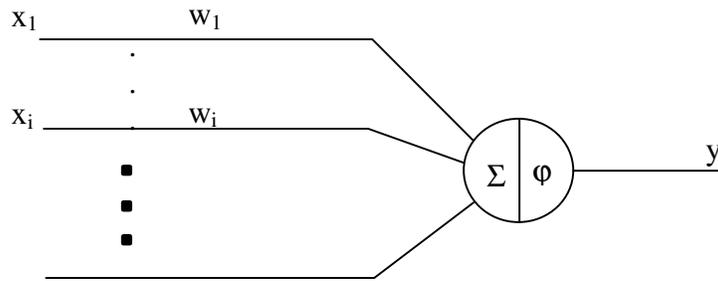


Figura 2.2 – Modelo de neurônio artificial

De acordo com o modelo da Figura 2.2, $\vec{x} = [x_1, x_2, \dots, x_N]^T$ equivale ao vetor de entradas ou estímulos externos apresentados ao neurônio e $\vec{w} = [w_1, w_2, \dots, w_N]^T$ corresponde ao vetor de pesos sinápticos, que correspondem à memória do neurônio. O processamento neural ocorre da seguinte maneira: quando as entradas \vec{x} são apresentadas ao neurônio, alguma função de medida determina a similaridade entre os vetores \vec{x} e \vec{w} . A medida normalmente utilizada é a distância Euclidiana, dada pela equação (2.1).

$$d(\vec{x}, \vec{w}) = \|\vec{x} - \vec{w}\|_2 = \left[\sum_{k=1}^N (x_k - w_k)^2 \right]^{1/2} \quad (2.1)$$

Outra medida de similaridade bastante utilizada é o produto interno entre os vetores \vec{x} e \vec{w} , dado pela equação (2.2). Uma vez comparados os vetores de pesos e de entradas, a saída y é obtida através da aplicação da função ϕ ao resultado u da comparação, conforme a equação (2.3). A função de ativação ϕ pode ser linear ou não-linear. Normalmente, um limiar de ativação é estabelecido para que o disparo do neurônio só seja significativo caso sua ativação interna exceda o limiar. Duas funções de ativação não-lineares típicas são a função sinal, dada pela equação (2.4), e a função sigmoideal, dada pela equação (2.5), ilustradas na Figura 2.3.

$$(\vec{x}, \vec{w}) = \vec{x}^T \vec{w} = \sum_{k=1}^N x_k w_k \quad (2.2)$$

$$y = \varphi(u) \quad (2.3)$$

$$\varphi(u) = \begin{cases} 0 & \text{se } u < 0 \\ 1/2 & \text{se } u = 0 \\ 1 & \text{se } u > 0 \end{cases} \quad (2.4)$$

$$\varphi(u) = \frac{1}{(1 + e^{-\beta u})} \quad (2.5)$$

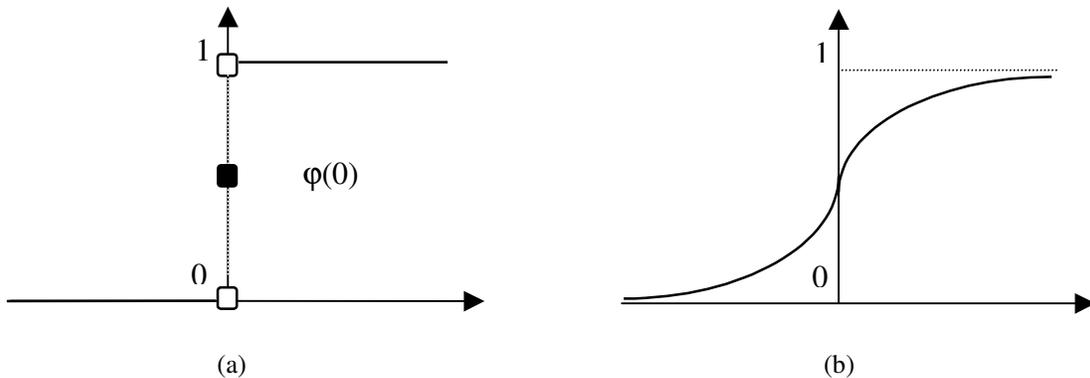


Figura 2.3 – Funções de ativação não-lineares: (a) função sinal; (b) função sigmoidal

O modelo de McCulloch e Pitts, apesar de capturar algumas características típicas do processamento neural biológico, ainda carecia de um mecanismo de aprendizagem que viabilizasse a incorporação de conhecimento e a melhoria no desempenho da rede, de forma automática. Neste sentido, a primeira proposta concreta e bem sucedida de um método de aprendizagem para estruturas neurais foi apresentada por D. Hebb, em 1949 (HEBB, 1949), a qual hoje leva seu nome.

Outro importante passo na pesquisa em redes neurais artificiais foi a proposta de ROSENBLATT (1958), que introduziu um novo modelo de reconhecimento de padrões, batizado de perceptron, juntamente com um método de aprendizagem supervisionada. A organização em camadas dos neurônios que compõem um perceptron conduz à rede neural do tipo perceptron multicamadas, amplamente utilizada em tarefas de reconhecimento e classificação de padrões, mas isto somente após a concepção do algoritmo de

retropropagação, a ser discutido mais adiante. Basicamente, a arquitetura de uma rede multicamadas apresenta a forma ilustrada na Figura 2.4.

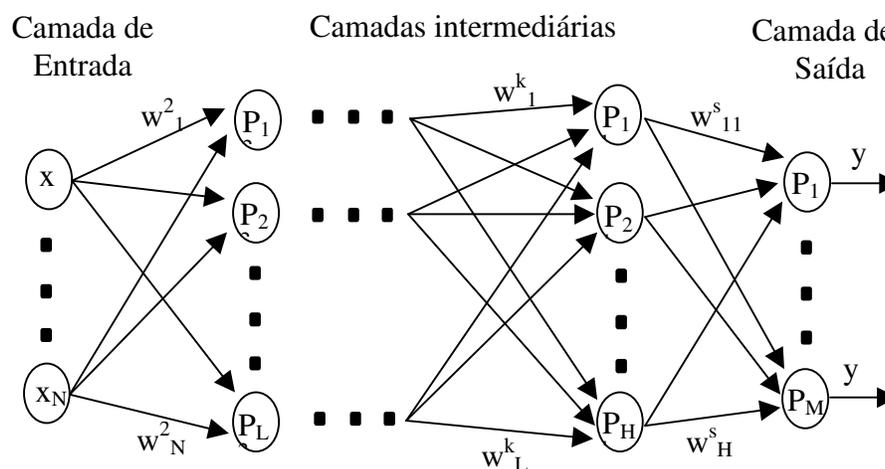


Figura 2.4 – Estrutura de uma rede neural do tipo perceptron multicamadas

Na rede neural multicamadas, o número de camadas, bem como o número de neurônios por camada, podem variar e é determinado pelo projetista. Além disto, embora normalmente as conexões liguem neurônios de camadas consecutivas, podem ocorrer conexões entre camadas não consecutivas, de forma que, em geral, a arquitetura de uma rede multicamadas é muito versátil e adaptável.

MINSKY & PAPERT (1969) apontaram uma limitação do perceptron de uma única camada que impossibilitava a implementação da função Ou-exclusivo. Graças às suas considerações, a pesquisa sobre redes neurais sofreu uma drástica desaceleração durante os anos 70. Somente em 1986, Rumelhart, Hinton e Williams (RUMELHART *et al.*; 1986) reascendem as pesquisas solucionando o problema apontado por Minsky através da proposição do algoritmo de retropropagação para o treinamento de redes multicamadas. Um pouco antes, KOHONEN (1982) formalizara um importante método de aprendizagem não supervisionada que chamou de mapas auto-organizáveis. Também em 1982, J. Hopfield (HOPFIELD; 1982) utilizou-se da idéia de funções de energia para criar um modelo de rede neural recorrente, que hoje leva seu nome, capaz de armazenar e posteriormente recuperar padrões, mesmo sujeitos a perturbações.

2.2.2 Modelo de aprendizagem de Hebb

O postulado de aprendizagem de Hebb é a mais antiga e mais famosa lei de aprendizado de um mecanismo neural (HAYKIN, 1999). Segundo esta lei, quando duas células adjacentes disparam repetidamente e de forma sincronizada, a sinapse que as une tem sua intensidade reforçada. Ao contrário, se as mesmas duas células adjacentes, disparam repetidamente de forma assíncrona, a sinapse que as une tem sua intensidade reduzida.

A Figura 2.5 ilustra o mecanismo de aprendizagem de Hebb considerando dois neurônios N_i^k e $N_j^{(k+1)}$ e a sinapse $w^{(k+1)}_{ij}$ que os une, a qual é reforçada ou enfraquecida dependendo da sincronia entre os disparos dos dois neurônios.

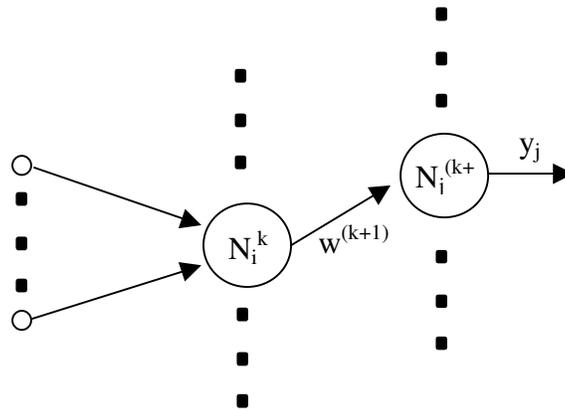


Figura 2.5 – Neurônios adjacentes e sinapse modificada segundo a lei de Hebb

2.2.3 Modelo de aprendizagem por retropropagação

O modelo de aprendizagem por retropropagação é o método mais conhecido de treinamento supervisionado de redes neurais artificiais. Diz-se que o método é supervisionado, pois a aprendizagem é dirigida por um conjunto de treinamento composto por uma série de n padrões do tipo (\vec{x}, \vec{y}^d) , onde \vec{x} corresponde a um vetor de entradas e \vec{y}^d corresponde ao vetor de saídas desejadas associado ao vetor de entradas \vec{x} .

A partir da aplicação do vetor de entradas \vec{x} à rede neural, a função de erro $E(\vec{w})$ determina uma relação entre os vetores de saída \vec{y} da rede e a saída desejada \vec{y}^d , cuja

forma é dada pela equação (2.6), onde $\| \cdot \|_2$ é a norma Euclidiana e N é o número de padrões de entrada-saída. O objetivo do processo de aprendizagem é então minimizar a função de erro da rede através do ajuste conveniente dos pesos sinápticos de seus neurônios. A Figura 2.6 ilustra o modelo de treinamento supervisionado.

$$E(\bar{w}) = \frac{1}{2} \sum_{l=1}^N \|y_l^d - y_l\|_2^2 \quad (2.6)$$

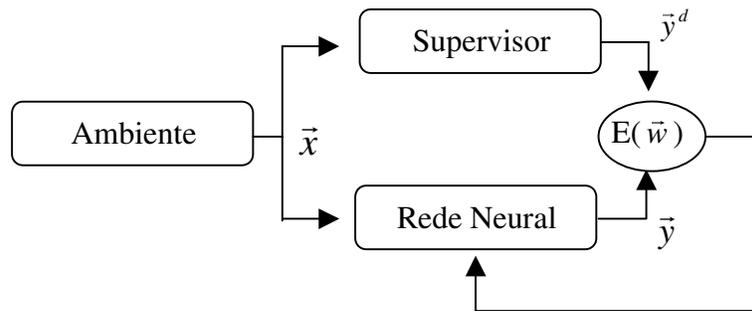


Figura 2.6 – Modelo de treinamento supervisionado

O método de aprendizagem recebe o nome retropropagação devido ao fato de o sinal do erro $E(\bar{w})$ ser propagado a partir da saída da rede para todos os seus neurônios, percorrendo o caminho inverso daquele realizado pelo processamento do sinal de entrada. Diversos são os mecanismos de ajuste dos pesos sinápticos de uma rede neural multicamadas. Dentre eles, o mais utilizado é o método do gradiente.

Este método prevê o ajuste dos pesos da rede na direção contrária àquela dada pelo vetor gradiente da função de erro. Este método necessita da informação das derivadas parciais de primeira ordem da função de erro com relação aos pesos de cada neurônio e das funções de ativação dos neurônios da rede. Sendo assim, tanto a função de erro $E(\bar{w})$ quanto a função de ativação $\varphi(u_j)$ do neurônio j devem ser contínuas e diferenciáveis no domínio do conjunto de treinamento. A expressão geral de ajuste de um peso sináptico w_{ij} no instante (t+1) é dada pela equação (2.7).

$$w_{ij}(t+1) = w_{ij}(t) - \alpha(\nabla w_{ij}(t)) \quad (2.7)$$

Nesta equação, α corresponde ao coeficiente de aprendizagem, que dita a velocidade do processo de aprendizagem, e ∇w_{ij} corresponde ao componente do vetor gradiente em relação ao peso sináptico w_{ij} . Quando α é muito grande, há o risco de o processo de treinamento não convergir para uma solução, e se for muito pequeno, a convergência pode se tornar excessivamente lenta. Considerando o ajuste de pesos sinápticos através do método do gradiente, uma rede com três camadas, a camada de saída com n neurônios e N padrões no conjunto de treinamento, tem-se que o componente do vetor gradiente com relação ao peso w^s_{ij} de um neurônio da camada de saída é dado pela equação (2.8).

$$\nabla E(w^s_{ij}) = \frac{\partial E}{\partial w^s_{ij}} = \sum_{l=1}^N (\varphi(u^s_{li}(\vec{w})) - y^d_{li}) \frac{dg}{du^s_{li}} x^s_{lij} \quad (2.8)$$

$$i = 1, 2, \dots, p; j = 1, 2, \dots, n$$

Para os pesos de neurônios da e -ésima camada intermediária, o componente do gradiente na direção do peso w^e_{ij} é dado pela equação (2.9). Para redes com q camadas intermediárias, um componente do gradiente para a camada t é encontrado através de sucessivas aplicações da regra da cadeia para derivações junto $E(\vec{w})$ com relação a seus componentes, encontrando-se a derivada parcial em relação ao peso que se deseja ajustar.

$$\nabla E(w^e_{jk}) = \frac{\partial E}{\partial w^e_{jk}} = \sum_{l=1}^N \left[\sum_{i=1}^n (\varphi(u^s_{li}(\vec{w})) - y^d_{li}) \frac{dg}{du^s_{li}} w^s_{ij} \right] \frac{dg}{du^e_{lj}} x^e_{ljk} \quad (2.9)$$

O processo de aprendizagem deve prosseguir até que o critério de parada seja atingido. Normalmente, este critério implica o estabelecimento de um erro mínimo ϵ . Deve ser observado, porém, que quando o treinamento se estende por um período demasiadamente longo, há o risco de a rede perder sua característica de generalização (GEMAN *et al.*, 1992).

2.2.4 Aplicações de redes neurais artificiais

Dentre os diversos tipos de arquiteturas de redes neurais propostos em literatura, muitos dos quais não apresentados aqui, cada qual comporta características particulares para um determinado tipo de aplicação. FIGUEIREDO (1997) aponta sete classes de problemas para os quais boas soluções podem ser encontradas com a ajuda de modelos neurais.

A *capacidade de aproximação* permite a utilização de determinadas redes em tarefas como identificação de sistemas ou estimação de funções. O perceptron multicamadas, por exemplo, tem capacidade de aproximação universal de funções. A *capacidade de associação* permite a recuperação de padrões prototípicos a partir dos próprios padrões degradados, como no caso das redes de Hopfield. A *capacidade de classificação e de agrupamento*, são características do perceptron e das redes de Kohonen, além de tarefas como *controle, predição e otimização*, bem suportadas pelos modelos neurais discutidos anteriormente.

2.3 Conjuntos, Inferência e Sistemas Nebulosos

2.3.1 Histórico e visão geral

O conceito de conjuntos nebulosos, formalizado por ZADEH (1965) tornou-se a base da lógica nebulosa, aplicável a problemas que apresentam características inerentemente imprecisas, para os quais os mecanismos de inferência vigentes até então se mostravam pouco atraentes. Desde sua concepção, a teoria dos conjuntos nebulosos tem sido largamente utilizada nos mais diversos problemas de controle e automação (FUKUDA & KUBOTA, 1999; AGUIRRE & GONZÁLEZ, 2000), sistemas de apoio à decisão (KUO & XUE, 1999), modelagem (KASABOV, 1996; YANG *et al.*, 2000), representação do conhecimento (HIROTA & PEDRYCZ, 1999) e processamento de sinais (PLATANIOTIS *et al.*, 1999).

A fusão de conceitos extraídos da teoria dos conjuntos nebulosos com outras técnicas de inteligência computacional (e.g. sistemas especialistas, redes neurais artificiais, computação evolutiva) também tem apresentado grande sucesso (KASABOV, 1996; LI *et al.*, 1997;

McFETRIDGE & IBRAHIM, 1998; KUO & XUE, 1999; YANG *et al.*, 2000; CHUNG *et al.*, 2000; SONG & SHEEN, 2000; ALIEV *et al.* 2001).

Um conjunto nebuloso é caracterizado por uma função de pertinência $N(x)$ que relaciona elementos de um domínio ou universo de discurso U e o intervalo $[0, 1]$ na forma expressa pela equação (2.11), ou alternativamente como um conjunto de pares ordenados na forma $N = \{(x, N(x)) | x \in U\}$. Um conjunto nebuloso corresponde portanto a uma generalização do conceito de conjunto (GOMIDE & PEDRYCZ, 1998), uma vez que a função de pertinência nebulosa admite graus de pertinência de um objeto ao conjunto a ele associado.

$$N(x): U \rightarrow [0,1] \quad (2.11)$$

A Figura 2.8 apresenta no plano cartesiano dois conjuntos, A e B , semanticamente idênticos, cujos elementos são pessoas de estatura elevada, sendo que A não é um conjunto nebuloso, ao passo que B o é. O exemplo deixa clara a flexibilidade de representação de um conjunto nebuloso, uma vez que leva em consideração a possível diversidade de gradações de estatura de um grupo de pessoas, ao passo que na representação convencional, uma pessoa com 180cm pertence ao conjunto de pessoas altas, ao passo que uma pessoa com 179cm não pertence, embora a diferença entre ambas seja mínima levando-se em consideração o universo de discurso utilizado.

A teoria clássica de conjuntos permite boas representações quando uma caracterização bem definida das restrições e propriedades a serem observadas encontra-se disponível. Considerando ainda o conjunto da Figura 2.7(a), caso a condição de pertinência ao conjunto fosse o indivíduo apresentar *estatura acima de 180 cm* ao invés de simplesmente *estatura elevada*, a representação observada estaria correta e plausível. A condição *estatura elevada* é muito imprecisa e intuitiva, não havendo um ponto exato de divisão entre o verdadeiro e o falso para representá-la.

A maioria das situações reais, isto é, aquelas que ocorrem fora de um ambiente controlado de simulação computacional, são caracterizadas por definições imprecisas. Mais ainda, o

mecanismo de raciocínio de um ser humano é fortemente baseado em associações entre objetos e significados freqüentemente imprecisos e inexatos. Neste caso, a representação nebulosa não apenas apresenta maior credibilidade como também maior facilidade de interpretação.

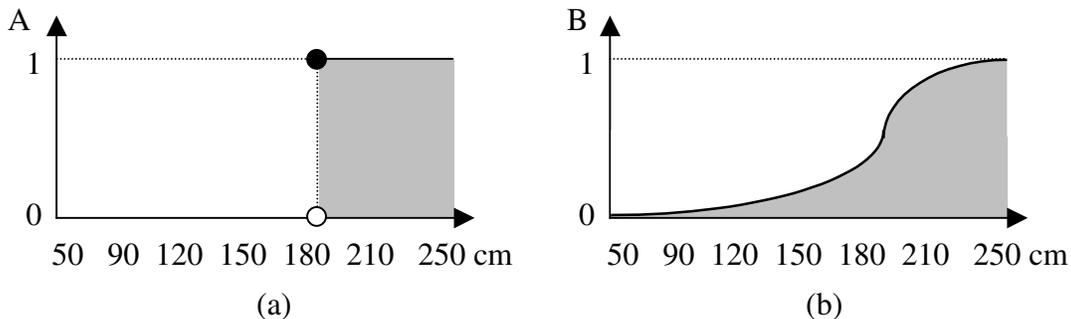


Figura 2.7 – Duas funções de pertinência de conjuntos nebulosos representando pessoas de estatura elevada: (a) representação convencional; (b) representação nebulosa

Todos os operadores aplicáveis a um conjunto clássico também são aplicáveis a um conjunto nebuloso, sendo que os principais dentre estes operadores são definidos por duas classes de funções de duas variáveis, denominadas t-norma e s-norma. As funções pertencentes a estas classes devem satisfazer a um conjunto de propriedades. Dado que x , y , z , u e v sejam elementos do universo de discurso para os quais as funções $S(\dots)$, pertencente à classe das s-normas, $T(\dots)$ pertencente à classe das t-normas e $Q(\dots)$, pertencente a ambas, sejam definidas, então as propriedades (i) a (iv) a seguir se aplicam (YAGER & FILEV, 1994):

- i) Comutatividade: $Q(x, y) = Q(y, x)$.
- ii) Associatividade: $Q(x, T(y, z)) = Q(T(x, y), z)$.
- iii) Monotonicidade: Se $x \leq u$ e $y \leq v \rightarrow Q(x, y) \leq Q(u, v)$.
- iv) Condições de contorno: $S(0, z) = z$, $S(1, z) = 1$;
 $T(0, z) = 0$, $T(1, z) = z$.

Considerando os conjuntos nebulosos $A(x)$ e $B(x)$, o operador nebuloso de união, dado pela função $\max(A(x), B(x))$, é um exemplo de s-norma, sendo formalmente definido pela

equação (2.12). O operador nebuloso de intersecção, dado pela função $\min(A(x),B(x))$, é um exemplo de t-norma, sendo formalmente definido pela equação (2.13). O operador nebuloso de complemento é dado pela equação (2.14). A equação (2.15) apresenta a t-norma produto, representada por t_p .

$$A \cup B = \max(A(x), B(x)) = \begin{cases} A(x) \rightarrow A(x) \geq B(x); \\ B(x) \rightarrow B(x) > A(x). \end{cases} \quad (2.12)$$

$$A \cap B = \min(A(x), B(x)) = \begin{cases} A(x) \rightarrow A(x) \leq B(x); \\ B(x) \rightarrow B(x) < A(x). \end{cases} \quad (2.13)$$

$$\sim A = 1 - A \quad (2.14)$$

$$A(x) t_p B(x) = A(x) \cdot B(x) \quad (2.15)$$

Os operadores de medida de possibilidade e medida de necessidade relacionam dois conjuntos nebulosos $A(x)$ e $B(x)$. O operador de medida de possibilidade avalia o quanto os dois conjuntos se aproximam um do outro, ao passo que o operador de medida de necessidade avalia o quanto um conjunto está contido no outro. Dada a função $\sup[A(x)]$, definida como o valor verdade máximo de um conjunto nebuloso, e dada a função $\inf[A(x)]$, definida como o valor verdade mínimo de um conjunto nebuloso, os operadores de medida de possibilidade e medida de necessidade são dados pelas equações (2.16) e (2.17) respectivamente. A Figura 2.8 ilustra os operadores nebulosos básicos, apresentados acima.

$$Poss(A, B) = \sup[\min(A(x), B(x))] \quad (2.16)$$

$$Nec(A, B) = \inf[\max(A(x), 1 - B(x))] \quad (2.17)$$

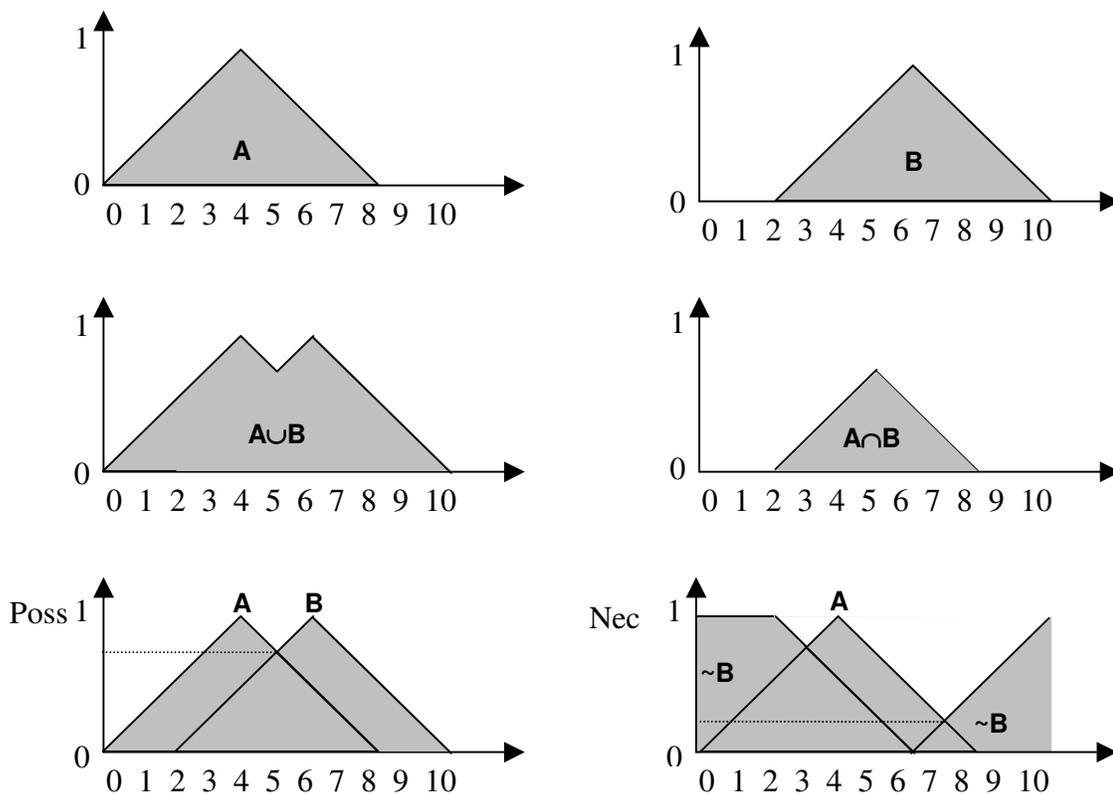


Figura 2.8 – Operadores básicos de conjuntos nebulosos

2.3.2 Inferência Nebulosa

A inferência nebulosa, desenvolvida a partir da teoria dos conjuntos nebulosos, confere um tratamento formal a expressões em que termos lingüísticos vagos ou imprecisos (e.g. médio, grande, pequeno) são empregados na descrição de conhecimento (FIGUEIREDO, 1997). Estas expressões são, em muitos casos, descritas por regras do tipo *se-então*, e.g., “Se a probabilidade de estiagem é alta, então a expectativa de produtividade é baixa”. A representação desta informação pode se dar através da regra nebulosa apresentada pela expressão (2.18).

$$\text{Se } X \text{ é } A \text{ então } Y \text{ é } B \quad (2.18)$$

Nesta expressão, X e Y correspondem a variáveis nebulosas, que por sua vez correspondem aos atributos “probabilidade de estiagem” e “expectativa de produtividade”, ao passo que

os conjuntos nebulosos A e B são conceitos associados a estes atributos. Uma característica marcante da inferência nebulosa é sua capacidade de representar raciocínio aproximado, na medida em que não há a necessidade de que um fato ou um conjunto de fatos concorde exatamente com a expressão, mas se um fato implica uma conclusão, então fatos semelhantes devem implicar conclusões semelhantes.

Um dos mecanismos de inferência nebulosa mais utilizados, conhecido como *modus ponens* apresenta a forma da expressão (2.19). A premissa 1 corresponde a um fato com M variáveis X_i , ao passo que a premissa 2 corresponde a uma regra.

$$\begin{array}{ll}
 \text{Premissa 1:} & X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \text{ e } \dots X_M \text{ é } A_M \\
 \text{Premissa 2:} & \text{Se } X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \text{ e } \dots X_M \text{ é } A_M \text{ então } Y \text{ é } B \\
 \text{Conclusão:} & Y \text{ é } B
 \end{array} \tag{2.19}$$

Um dos procedimentos de implementação do mecanismo de inferência nebulosa apresentado acima, visando obter o conjunto nebuloso B, compreende três etapas: comparação, agregação e conclusão (FIGUEIREDO, 1997; YAGER & FILEV, 1994).

Na comparação, o fato da premissa 1 é comparado separadamente em cada variável de acordo com a equação (2.20), onde R_i é o resultado da i-ésima comparação e S e T são respectivamente uma s-norma e uma t-norma no universo de discurso da i-ésima variável.

$$R_i = S \left(T(A_i(x), A_i(x)) \right) \tag{2.20}$$

A agregação dos antecedentes toma os resultados individuais obtidos da equação (2.20) conjuntamente e gera o valor C, conforme a equação (2.21).

$$C = \prod_{i=1}^M (R_i) = T(R_M, T(R_{M-1}, T(\dots, T(R_2, R_1)))) \tag{2.21}$$

Por fim, a conclusão promove a geração do conjunto nebuloso B a partir do resultado da agregação C. O operador O, que geralmente corresponde à t-norma min, pode ser expresso pela equação (2.22).

$$B(y) = O_{y \in U_y} (C, B(y)) \quad (2.22)$$

Para o caso de múltiplas regras, uma nova etapa deve ser incorporada ao mecanismo, correspondendo à agregação das regras, que corresponde à aplicação de uma s-norma S, conforme apresentada na equação (2.23).

$$B(y) = \sum_{i=1}^N (B^i(y)) \quad (2.23)$$

2.3.3 Sistemas Nebulosos

Desenvolvidos a partir da teoria dos conjuntos nebulosos e da inferência nebulosa, os sistemas nebulosos são importantes ferramentas de inferência, com múltiplas aplicações na medida em que possuem a capacidade de inferir conclusões a partir das entradas apresentadas (FIGUEIREDO, 1997). A Figura 2.9 apresenta os componentes básicos de um sistema nebuloso.

A partir das leituras sensoriais obtidas do ambiente, a interface de entrada promove a transformação, quando necessário, dos valores obtidos em conjuntos nebulosos. A base de regras contém regras nebulosas do tipo *se-então* discutidas anteriormente e a base de dados contém informações sobre parâmetros e outros módulos do sistema. A partir destas duas bases, o módulo de inferência realiza o processamento do sinal de entrada, gerando as conclusões pertinentes. Estas conclusões, sob a forma de conjuntos nebulosos, são transformadas em sinais compatíveis com o ambiente pelo módulo de saída.

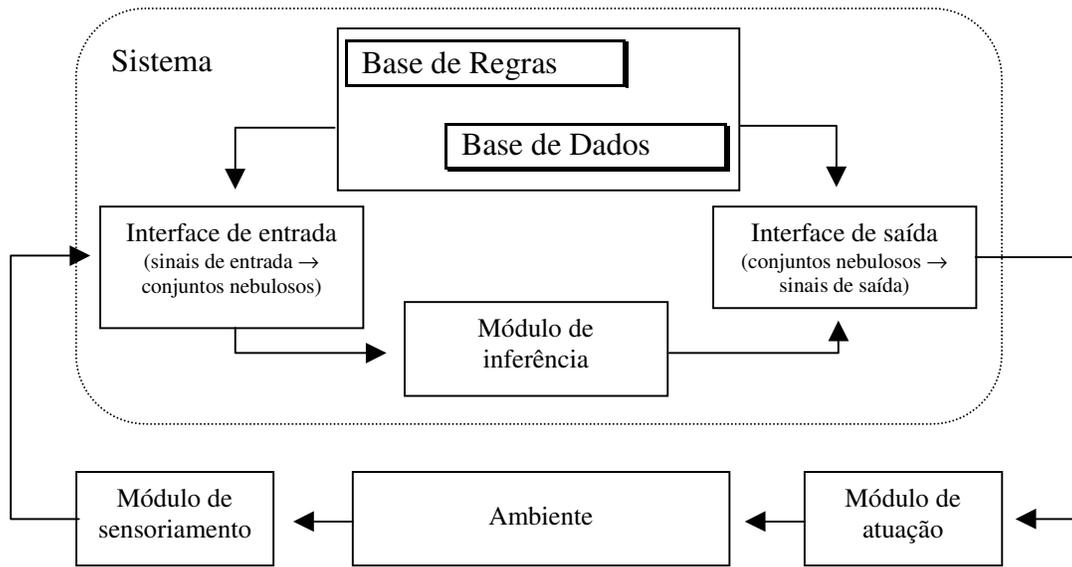


Figura 2.9 – Organização modular de um sistema nebuloso

O processamento de implementação da base de regras faz uso de funções de pertinência $A(x) \in U$, geralmente definidas por um valor modal μ e por uma dispersão δ , satisfazendo às propriedades de normalidade e simetria, conforme expresso a seguir:

Normalidade: $A(\mu) = 1$;

Simetria: $\forall (u \wedge v) \in U$ se $|z - u| = |z - v|$ então $A(u) = A(v)$.

Dois exemplos de funções típicas que apresentam estas propriedades são as curvas de Gauss (Figura 2.11(a)) e as funções triangulares (Figura 2.11(b)). Um exemplo de um processo de transformação do sinal de entrada $x \in \mathfrak{R}$ em um conjunto nebuloso corresponde à aplicação de x como valor modal de uma função de pertinência, que pode assumir diversas formas, como na Figura 2.10, considerando uma certa dispersão δ . O processo de geração da saída $\bar{y} \in \mathfrak{R}$ a partir do conjunto nebuloso B geralmente corresponde à utilização do método centro de área, dado pela equação (2.24).

$$\bar{y} = \frac{\sum_{k=1}^K y_k B(y_k)}{\sum_{k=1}^K B(y_k)} \quad (2.24)$$

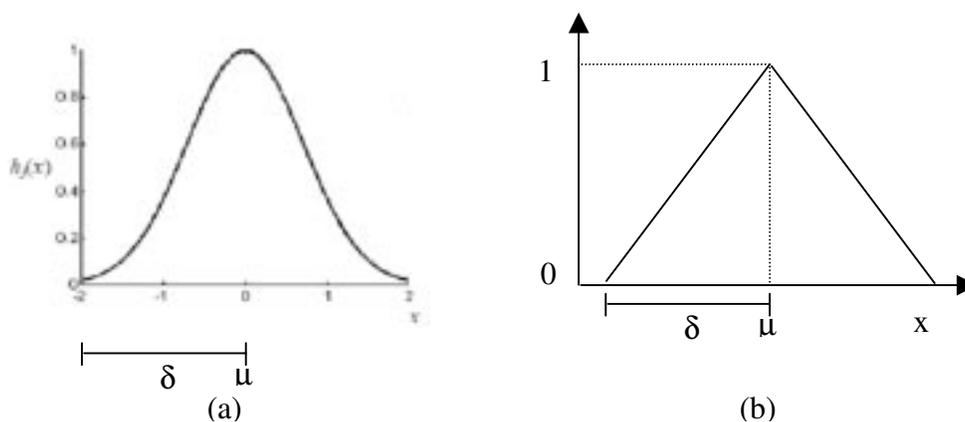


Figura 2.10 – Funções de pertinência típicas utilizadas em sistemas nebulosos: (a) curva de Gauss; (b) função triangular

2.4 Algoritmos Genéticos

2.4.1 Histórico e visão geral

Assim como no caso das redes neurais artificiais, os algoritmos genéticos correspondem a uma técnica computacional fortemente inspirada em características de sistemas biológicos. Particularmente, a base de inspiração dos algoritmos genéticos encontra-se na teoria da evolução das espécies, originalmente proposta por DARWIN (1859) e fundamentada em mecanismos de seleção em que o indivíduo mais bem adaptado ao meio possui chances maiores de deixar descendentes para futuras gerações.

Estes mecanismos de seleção, associados a pequenas variações nas características físicas dos indivíduos de mesma espécie, produziram então, ao longo de gerações sucessivas, indivíduos progressivamente mais bem adaptados ao seu meio ambiente. Com o advento da moderna genética, pôde-se verificar que aquelas pequenas variações físicas dos indivíduos decorriam de erros aleatórios de cópia do DNA no momento da reprodução (mutações), e no caso de indivíduos com reprodução sexuada, da combinação das contribuições genéticas dos pais na formação da herança do filho (recombinação).

As primeiras tentativas de aplicação da teoria da evolução de Darwin à concepção de sistemas computacionais decorreram dos estudos pioneiros realizados nos anos 50 e 60

(ANDERSON, 1953; FRIEDBERG, 1958; FRASER, 1959; BREMERMAN, 1962), os quais visavam a concepção de ferramentas para a solução de problemas de engenharia. Ao longo dos anos 60 e início dos anos 70, HOLLAND (1975) e seu grupo na universidade de Michigan desenvolvem os algoritmos genéticos, com o objetivo de estudar formalmente os fenômenos de adaptação e importá-los para ambientes computacionais.

Algoritmos genéticos trabalham com a evolução paralela de soluções candidatas a um determinado problema. A representação das soluções candidatas é feita através de cromossomos ou listas de atributos, que reduzem a um conjunto de parâmetros, cada qual correspondendo a um gene. Os genes podem assumir qualquer codificação, sendo normalmente utilizadas as codificações numéricas em base binária ou decimal. Além disso, um gene pode ser caracterizado por um único valor numérico (e.g. 0 ou 1) ou por uma seqüência de valores.

Uma vez definido o tipo de codificação das soluções candidatas, uma população inicial de indivíduos é gerada e os valores dos genes são atribuídos aleatoriamente de acordo com o tipo de codificação utilizada. A geração aleatória da população inicial de cromossomos tem como finalidade produzir diversidade entre os indivíduos e estimular a exploração de todo o espaço de busca pelas soluções. Outros mecanismos de geração inicial têm sido propostos com a mesma finalidade (SANTOS *et al.* 2001).

A Figura 2.11 apresenta o diagrama de uma proposta de algoritmo genético. Após o bloco de geração da população inicial, tem início o ciclo de gerações do algoritmo, sendo que cada ciclo corresponde a uma geração de indivíduos. De acordo com o algoritmo, o operador de recombinação é aplicado à população com probabilidade percentual p_r e o operador de mutação é aplicado com probabilidade percentual p_m , sendo que $N \in \mathfrak{R}$ e $L \in \mathfrak{R}$ correspondem a valores gerados aleatoriamente no intervalo $[0, 100]$ a cada ciclo. Após a aplicação dos operadores genéticos, cada indivíduo da população, que representa uma solução candidata do problema, é avaliado e o critério de parada do algoritmo é testado. Este critério pode corresponder a um número limite de gerações, ou à obtenção de uma solução considerada aceitável para o problema, ou a uma conjunção de ambos os critérios.

Caso não tenha sido satisfeito o critério de parada, o operador de seleção é aplicado à população, de forma a escolher, de acordo com algum critério, os indivíduos que participarão da próxima geração do algoritmo.

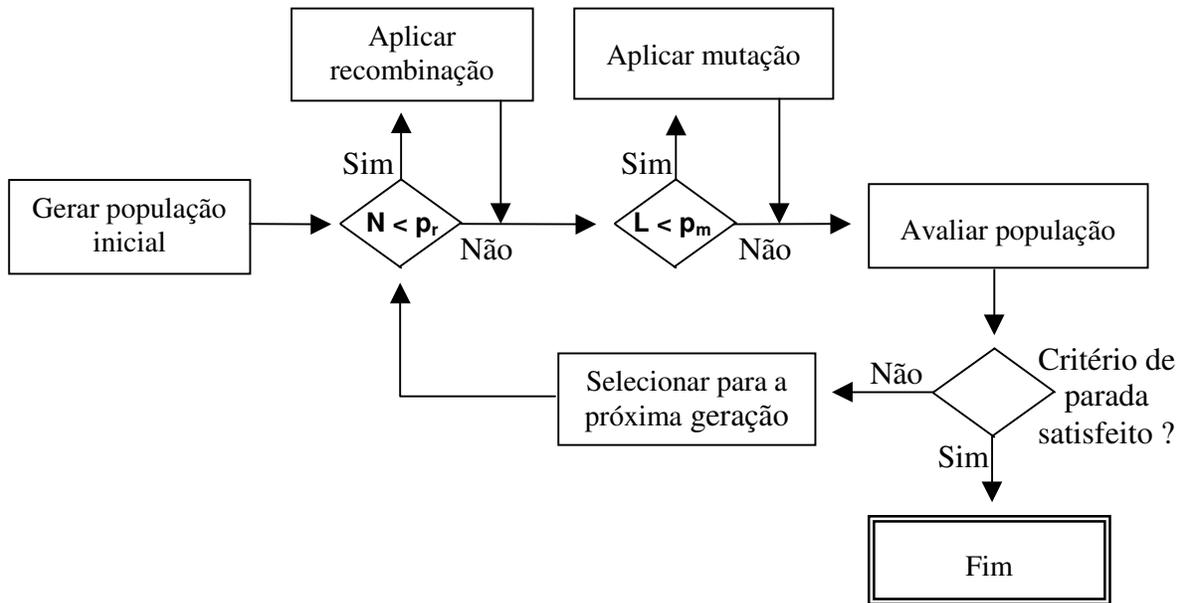


Figura 2.11 – Diagrama apresentando uma proposta de algoritmo genético

2.4.2 Operador de recombinação

O operador de recombinação promove o compartilhamento do material genético entre dois cromossomos da população. Dependendo da forma como o operador é implementado, o resultado de sua aplicação pode ser a criação de um único cromossomo filho contendo parte da carga genética de cada um dos pais, ou a simples troca do material genético entre os cromossomos-pais. Pode-se ainda escolher entre manter ou não os cromossomos pais na população após a realização da recombinação. A motivação básica da implementação de um operador de recombinação, a exemplo do que ocorre em alguns organismos biológicos, é manter a variabilidade genética na população (BOOKER, 1993).

De acordo com a implementação do operador de recombinação proposta por HOLLAND (1975), uma vez escolhidos aleatoriamente dois cromossomos a serem cruzados, um ou mais pontos de corte, no mesmo locus em ambos os cromossomos, são escolhidos também aleatoriamente e os segmentos dos cromossomos-pais são trocados. Em uma outra implementação, conhecida como recombinação uniforme, não há segmentação nos

cromossomos antes da troca, mas cada um dos genes dos dois cromossomos a serem cruzados é selecionado individualmente de acordo com a mesma probabilidade p_r . A Figura 2.12(a) apresenta graficamente a recombinação simples de um ponto, ao passo que a Figura 2.12(b) apresenta a recombinação uniforme, ambas aplicadas a cromossomos binários.

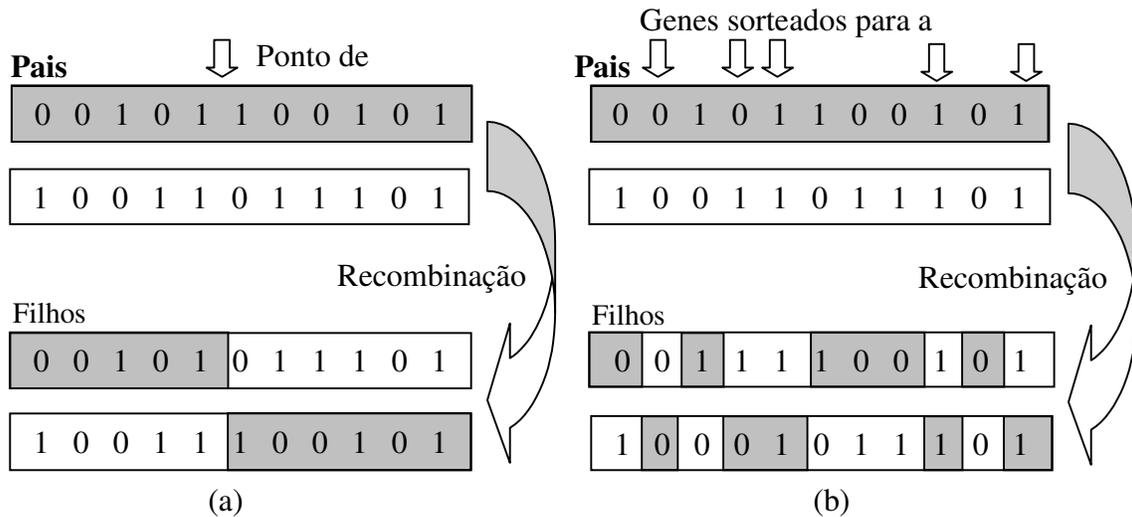


Figura 2.12 – Dois exemplos de recombinação: (a) simples de um ponto; (b) uniforme

2.4.3 Operador de mutação

O operador de mutação introduz pequenas variações genéticas nos cromossomos, mais especificamente nos valores de seus genes. Assim como o operador de recombinação, o operador de mutação visa manter a variabilidade genética na população e além disso, permitir a exploração de novas alternativas dentro do espaço de busca pelas soluções candidatas. Diferentemente do operador de recombinação, a mutação é sempre pontual, aplicando-se a um gene de um determinado cromossomo de acordo com a probabilidade p_m .

Em vetores binários, a mutação de um gene consiste na troca de seu valor binário (e.g. de 0 para 1). Para vetores que apresentem codificação de seus genes através de números reais, uma variação do operador clássico de mutação, conhecida como mutação indutiva, prevê a adição ou a subtração de um valor $\kappa \in \mathfrak{R}$ ao gene a ser modificado. Uma atenção especial neste caso deve ser dada a κ , pois caso seja muito grande, pode influir negativamente no processo de convergência do algoritmo e caso seja muito pequeno, pode apresentar efeito muito reduzido na preservação da diversidade populacional. A Figura 2.13 apresenta o

operador de mutação, em (a), aplicado a um cromossomo binário e em (b), aplicado a um cromossomo de codificação real.

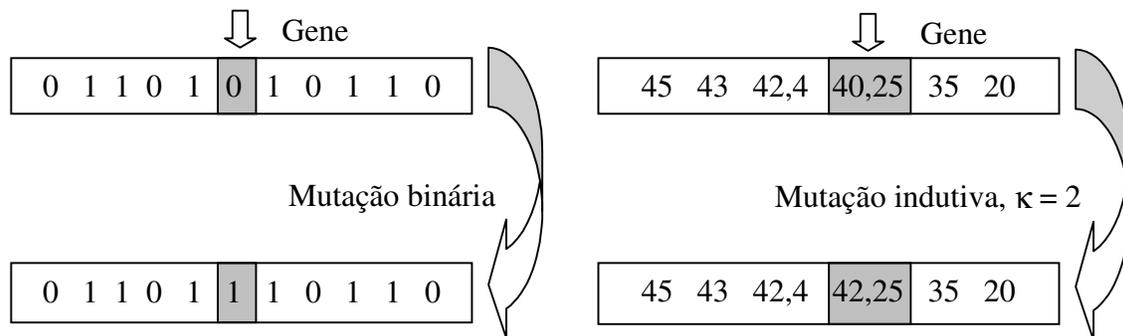


Figura 2.13 – Dois exemplos de mutação: (a) binária; (b) indutiva

2.4.4 Critério de avaliação

Os algoritmos genéticos são métodos genéricos de tratamento de problemas, exigindo muito pouca informação sobre o mesmo. Assim, embora o critério de avaliação da população seja um ponto crítico do algoritmo e dependa da natureza do problema em questão e da codificação utilizada, basta que uma função de avaliação $\varphi(\dots)$, também denominada função-objetivo seja definida no espaço de busca. Considerações quanto à sua continuidade ou diferenciabilidade em todo o espaço não são necessárias.

A superfície de adaptação definida pela função-objetivo pode conter, e geralmente contém, máximos locais. Cromossomos que representem máximos locais podem se disseminar rapidamente por toda a população e travar o processo evolutivo, sem que boas soluções tenham sido obtidas. Por isto, tanto na geração da população inicial quanto ao longo de todo o processo evolutivo, a manutenção da diversidade populacional é fundamental.

Em geral, embora nem sempre haja garantia de se encontrar a solução exata do problema, um algoritmo genético é capaz de encontrar boas aproximações para a mesma. Neste sentido, caso exista um método mais específico de resolução do problema em questão, a aplicação de alternativas evolutivas se torna injustificável, até pela quantidade de recursos computacionais que exige. No entanto, levando-se em consideração a quantidade de problemas tratáveis por métodos específicos considerando aqueles que ainda carecem de

métodos dedicados, como os sujeitos a explosões combinatórias, o campo de aplicação dos algoritmos genéticos é muito amplo.

2.4.5 Operador de seleção

O principal objetivo do operador de seleção é privilegiar a ocorrência de boas soluções na população. Não obstante, nenhum poder de criação de novas soluções lhe é conferido. Seu único papel é tomar para a próxima geração de cromossomos aqueles que representam boas soluções, descartando soluções ruins ou não tão boas, sem, contudo, reduzir drasticamente a diversidade. Há diversos tipos de seleção que atendem, em linhas gerais, a este mesmo conjunto de requisitos.

O tipo mais simples de seleção é conhecido como *método da roleta*, em que a probabilidade de escolha de um indivíduo para a próxima geração é proporcional ao valor relativo de sua avaliação frente à avaliação dos demais indivíduos da população. Outro método conhecido como *elitista*, seleciona os melhores indivíduos da população e os envia para a geração seguinte sem a realização de sorteio como ocorre com o método da roleta. A vantagem do método elitista é que não há o risco de perda da melhor solução.

Outros mecanismos de seleção procuram levar em consideração também o nível de diversidade populacional. Por exemplo, o método de *seleção por diversidade* leva em consideração os indivíduos mais diversos da população a partir do melhor indivíduo. O método de seleção *bi-classista* toma os $p\%$ melhores indivíduos e os $(100-p)\%$ piores indivíduos da população. Para maiores detalhes sobre operadores de seleção, veja (BÄCK *et al.*, 2000).

2.5 Considerações sobre aprendizado e comportamento

A história evolutiva, ou *filogênese* de uma espécie age de forma decisiva não apenas na determinação de alguns comportamentos inatos, por vezes denominados *instintos*, mas também de forma íntima sobre o aprendizado dos indivíduos da espécie, uma vez que

aspectos fisiológicos e anatômicos muitas vezes impõem barreiras ao aprendizado de determinados tipos de habilidades e predisõem o aprendizado de outras.

Determinados traços comportamentais são tão característicos de uma espécie quanto seus traços anatômicos (BAUM, 1999). Neste sentido, os *reflexos* são produto da seleção natural, e, portanto, características inatas da espécie que a manifesta. Sempre associados à manutenção das condições físicas e promoção da sobrevivência e reprodução do indivíduo, são reações simples e fortemente identificadas com um estímulo externo (e.g. espirro, excitação sexual).

Além dos reflexos, os *padrões fixos de ação*, que correspondem a formas mais elaboradas de comportamento, também correspondem a estruturas inatas do indivíduo, pois são intimamente relacionados à sua espécie, não necessitando de aprendizagem para se desenvolverem (e.g. movimentos elaborados do ritual de acasalamento de algumas espécies animais).

Muitas das teorias que relacionam estes comportamentos inatos com a aprendizagem de novas habilidades levam em consideração a ocorrência de dois tipos de eventos, chamados agentes de reforço e agentes de punição (BAUM, 1999). Neste sentido, eventos que, durante a filogênese, aumentaram a aptidão do indivíduo por estarem presentes, são chamados de agentes de reforço e tendem a fortalecer o comportamento a eles associados, (e.g. alimento, abrigo, sexo). Ao contrário, eventos que, durante a filogênese, aumentam a aptidão por estarem ausentes, são chamados de agentes de punição e tendem a suprimir o comportamento a eles associados (e.g. se uma pessoa pegar uma brasa no fogo e queimar sua mão, então será menos provável que em uma outra oportunidade ela volte a tentar pegar uma brasa no fogo).

PAVLOV (1927) afirma que um animal possui a capacidade de responder a um determinado estímulo de forma incondicionada. Este estímulo específico, dito não-condicionado, gera uma resposta específica não-condicionada. Entretanto, de acordo com Pavlov um segundo estímulo pode ser utilizado para a mesma resposta, desde que ocorra um treinamento para

tanto. O treinamento consiste em apresentar o segundo estímulo na presença do estímulo não-condicionado que irá gerar a resposta não-condicionada, isto repetidas vezes. O resultado é que o segundo estímulo, agora chamado de condicionado, passa a gerar uma resposta semelhante à resposta não-condicionada.

THORNDIKE (1911) considera o aprendizado da relação estímulo-resposta à luz de um agente de reforço. Em sua *lei da prontidão*, afirma que a satisfação é o resultado da execução de uma ação para a qual o sistema nervoso estava preparado, ao passo que a frustração pode ser o resultado da não realização de uma ação para a qual o sistema estava preparado ou da realização de uma ação para a qual o sistema não estava preparado. Se por um lado o reforço fortalece as relações de estímulo-resposta, para Thorndike, a frustração nem sempre as enfraquece e o desuso nunca as enfraquece. Além disso, Thorndike considera que estímulos semelhantes possam levar à mesma resposta por parte do sistema nervoso (MENDELECK, 1995).

SKINNER (1953) compreende o aprendizado como um conjunto de dois comportamentos: o comportamento respondente e o comportamento operante. O comportamento respondente segue essencialmente a mesma idéia do condicionamento clássico de Pavlov e o termo respondente significa que a relação *ambiente-comportamento*, aprendida pelo ser, é uma resposta a um estímulo causal fornecido ao mesmo (e.g. um sinal sonoro apresentado antes da apresentação do alimento a um cão). Por outro lado, o comportamento operante ou instrumental, originalmente apresentado por Thorndike, corresponde a aquele explicitado pelo estado do próprio ser interagindo com o meio (MENDELECK, 1995), independentemente da apresentação de um estímulo causal (e.g. o pressionar de uma alavanca por um rato libera o alimento). O termo “instrumental” indica que a resposta do ser, fruto da sua interação com o ambiente, age como um instrumento para produzir o estímulo causal e o termo “operante” indica que a resposta do ser opera sobre o ambiente para produzir o estímulo causal (DONAHOE & PALMER, 1994).

Skinner também salienta a importância do reforço e da experiência repetitiva para o processo de aprendizagem da relação estímulo-resposta, e afirma que qualquer resposta

seguida do reforço de um estímulo tende a ser aprendida. Além disso, este estímulo pode ser externo (gerado por um professor) ou interno (vontade de satisfazer a uma necessidade).

Assim, diferentemente do comportamento respondente, que relaciona dois estímulos (e.g. o som de uma campainha e a apresentação do alimento simultaneamente), o comportamento operante relaciona um estímulo e uma ação (e.g. o estímulo dado pelo alimento quando alcançado reforça a ação de busca pelo mesmo). Esta relação entre comportamento e consequência pode ser positiva quando o comportamento aumenta a probabilidade de ocorrência da consequência, ou negativa, quando o comportamento reduz a probabilidade de ocorrência da consequência. A partir destes dois tipos de relação ação-consequência, aliados a dois tipos de consequência (reforço e punição), BAUM (1999) enumera quatro tipos de relações que podem gerar aprendizagem operante: o reforço positivo, o reforço negativo, a punição positiva e a punição negativa.

Segundo Baum, tem-se um *reforço positivo* quando a relação tende a manter a ação, que por sua vez torna mais provável o reforço (e.g. caçar para comer), *reforço negativo*, quando a relação tende a manter a ação, que torna menos provável a punição (e.g. ir ao médico ao ficar doente), *punição positiva*, quando a relação tende a suprimir a ação na medida em que esta torna a punição mais provável (e.g. pegar uma brasa no fogo e queimar a mão) e *punição negativa*, quando a relação tende a suprimir a ação na medida em que esta torna o reforço menos provável (e.g. relação entre fazer barulho durante a caçada e pegar o alimento).

Uma última consideração interessante sobre os agentes de reforço é que sua ação pode ter efeito variado dependendo da circunstância. Assim, se um reforço é aplicado sucessivas vezes, pode levar à saturação, chegando mesmo a se tornar uma punição (e.g. a prática regular de exercícios físicos é saudável para o corpo, mas quando esta prática se torna excessiva para os limites físicos do indivíduo pode levar à morte). Por outro lado, um reforço se torna mais expressivo após um período de privação (e.g. quando se está com muita fome, a comida parece mais saborosa).

2.6 Aprendizagem por Reforço

2.6.1 Histórico e visão geral

Os conceitos de punição e recompensa, originalmente concebidos para descrever mecanismos de aprendizagem de comportamentos em sistemas biológicos, conforme descrito acima, foram gradativamente incorporados à engenharia e à computação ao longo dos últimos cinquenta anos, sob diversos aspectos, dando origem a uma classe de mecanismos de aprendizagem denominada Aprendizagem por Reforço. A principal idéia por trás deste tipo de aprendizagem é permitir que o conhecimento seja incorporado através de interação direta do aprendiz com o substrato de aprendizagem, por meio de experiências que levarão a satisfações e a frustrações.

Na aprendizagem por reforço, diferentemente da aprendizagem supervisionada, não existe a indicação direta do grau de adequação das atitudes do aprendiz frente àquelas que lhe seriam esperadas. Neste caso, o que existe são sugestões, dadas pelos sinais de reforço e punição, a manutenções ou mudanças de determinados comportamentos frente às expectativas do próprio aprendiz. Logo, a aprendizagem por reforço é desencadeada pela presença ativa de eventos oriundos de estruturas inatas do aprendiz.

Ao longo do desenvolvimento das pesquisas envolvendo aprendizagem de máquina, os conceitos de aprendizagem por reforço e aprendizagem supervisionada sempre estiveram muito próximos, freqüentemente se confundindo um com o outro. Em parte devido a esta confusão, estudos genuinamente dedicados à aprendizagem por reforço são raros nos anos 60 e 70 (SUTTON & BARTO, 1998). Dentre os precursores desta área encontram-se MINSKY (1954) e FARLEY & CLARK (1954), com trabalhos dedicados à aplicação de aprendizagem por reforço a modelos de redes neurais, outra área de pesquisa incipiente na época.

Conceitos de busca heurística, advindos da inteligência artificial clássica, também contribuíram na implementação de técnicas de aprendizagem de máquina por reforço (MICHIE, 1961; MICHIE & CHAMBERS, 1968) e influenciaram o desenvolvimento de uma nuance muito em voga nos dias atuais, desenvolvida sobretudo a partir dos anos 80, que

considera problemas cujos estados possuam uma propriedade especial, denominada Propriedade de Markov. Entende-se por estado o sinal, ou conjunto de sinais, obtido do ambiente (e.g. via leitura sensorial) em um determinado instante. De acordo com esta propriedade, toda a informação relevante para a escolha da próxima ação do sistema deve estar contida no sinal de seu estado atual, que por sua vez resume toda a história de estímulos passados do sistema. Devido a isto, esta propriedade também é referida como Propriedade Livre de Trajetória (SUTTON & BARTO, 1998). Problemas cujos estados obedecem à Propriedade de Markov recebem o nome de Processos de Decisão de Markov (PDM).

Formalmente, considerando-se um número finito de estados s e valores de reforço r , a resposta do ambiente e o reforço obtido no instante $t+1$ frente à ação a tomada pelo sistema no instante t , assumindo-se um ambiente genérico, dependem de toda a história prévia do sistema, conforme a distribuição de probabilidades expressa na equação (2.25). Porém, no caso de um ambiente que possua a Propriedade de Markov, a probabilidade de resposta pode ser obtida imediatamente a partir do estado atual, conforme a equação (2.26).

$$P_r \{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0\} \quad (2.25)$$

$$P_r \{s_{t+1} = s', r_{t+1} = r | s_t, a_t\} \quad (2.26)$$

Freqüentemente, trabalhos aplicados a PDMs denominam a aprendizagem por reforço alternativamente como aprendizagem por tentativa e erro. De fato, aprendizagem nestes casos consiste na seleção por tentativa e erro, em cada estado do sistema, da ação que retorne o maior reforço, visando a satisfação de um determinado objetivo. Neste caso, o sinal de reforço enfatiza as melhores alternativas encontradas e o sistema tem constantemente a ambição de maximizar o reforço ao longo das escolhas.

PDMs são definidos por seus conjuntos de estados e ações. Assim, considerando o estado s e a ação a , a probabilidade de escolha do próximo estado s' é dada pela expressão (2.27), e assumindo que s' seja escolhido, o valor esperado para o reforço r é dado pela expressão

(2.28). Em particular estas expressões definem PDMs finitos, ou seja, aqueles que possuem um número finito de estados e transições.

$$P_{ss'}^a = P_r \{s_{t+1} = s' | s_t = s, a_t = a\} \quad (2.27)$$

$$R_{ss'}^a = E \{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \quad (2.28)$$

A maioria dos PDMs utilizam-se de funções especiais, denominadas *funções de valor* que correspondem à política de busca heurística utilizada para percorrer os estados do sistema, atendendo ao critério de maximização da soma dos sinais de reforço, recebidos em cada estado. Uma política π corresponde a um mapeamento do estado $s \in S$ e da ação $a \in A(s)$ à probabilidade $\pi(s,a)$ de se tomar a ação a dado que o estado atual seja s . As equações (2.29) e (2.30) apresentam duas políticas de busca, a primeira (função de valor de estado) considerando apenas o estado atual e a segunda (função de valor de ação) considerando o estado e a ação atuais. O fator γ é a probabilidade associada ao reforço r .

$$V^\pi(s) = E_\pi \{R_t | s_t = s\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right\} \quad (2.29)$$

$$Q^\pi(s, a) = E_\pi \{R_t | s_t = s, a_t = a\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right\} \quad (2.30)$$

PDMs permitem uma resposta do sistema baseada apenas na informação do estado atual do mesmo. Entretanto, a dinâmica apresentada pelas equações (2.27) e (2.28) requer o armazenamento explícito da informação de cada estado, necessária para a escolha do próximo estado e do reforço resultante. Isto implica a criação, ao longo das iterações do sistema, de um modelo de representação interno baseado no conhecimento específico sobre o problema.

De fato, este tipo de tratamento tem se mostrado eficaz em diversos tipos de problemas apresentados na literatura (BRAGA & ARAÚJO, 1998; FOX *et al.*, 1998; LITTMAN, 2001) , mas restringe muito o escopo de ação do sistema a ambientes estáticos ou muito pouco sujeitos a mudanças, visto que uma mudança ambiental implica a necessidade de reorganização do modelo de representação interno.

2.6.2 Aprendizagem por diferença temporal

A aprendizagem por reforço em PDMs ocorre por meio da atualização da probabilidade de escolha de uma ação ou de um estado, mediante o reforço associado à escolha realizada anteriormente. Neste sentido, a Aprendizagem por Diferenças Temporais (ADT) constitui uma das classes mais conhecidas de algoritmos de aprendizagem em PDMs. Entretanto, ao invés de estimar o reforço total a ser recebido, dada uma política π , os métodos de ADT restringem sua estimativa exclusivamente ao reforço recebido do estado posterior ao estado atual.

Assim, dada a estimativa V_t para a função de valor de estado V^π , a forma mais simples de implementação de ADT é conhecida como algoritmo TD(0) e é apresentada no Algoritmo 2.1. Os algoritmos apresentados nesta seção constam em (SUTTON & BARTO, 1998). Como pode ser observada, a política é iterativamente atualizada através de uma fração do reforço recebido pela escolha do estado s' .

```
Inicialize  $V(s)$  arbitrariamente,  $\pi$  para a política a ser avaliada
Repita (Para cada episódio)
  Inicialize  $s$ 
  Repita (para cada passo do episódio)
     $a \leftarrow$  ação dada por  $\pi$  para  $s$ 
    Tome a ação  $a$ ; observe o reforço  $r$  e o próximo estado  $s'$ 
     $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$ 
     $s \leftarrow s'$ 
  Até que  $s$  seja um estado terminal
```

Algoritmo 2.1 – Algoritmo TD(0)

Em PDMs, entende-se por *episódios* eventuais etapas ou subdivisões que uma tarefa possa apresentar e que sejam separadas por estados críticos do sistema (e.g. o transporte de determinado material por um robô pode ser dividido nos seguintes episódios: chegar ao local onde se encontra o material, pegar o material, levar o material para outro local, deixar o material, voltar para o ponto inicial).

Outro algoritmo de ADT, denominado Sarsa, trabalha com a estimativa $Q(s_t, a_t)$ da função de valor de ação $Q^\pi(s, a)$. Essencialmente, o processo é o mesmo que o utilizado pelo algoritmo TD(0), exceto que ao invés de se considerar transições de estado para estado, considera-se transições entre pares *estado-ação* conforme apresentado no Algoritmo 2.2.

Um último algoritmo, digno de nota por sua importância dentro da classe de algoritmos de ADT, é denominado *Q-learning* desenvolvido por WATKINS (1989). Este algoritmo apresenta como distinção o fato de que a função de valor de ação Q aproxima diretamente a função de valor de ação ótima Q^* , entendendo-se por ótima aquela função que retorna maior reforço ao longo do percurso pelos estados do sistema que todas as demais, conforme a equação (2.31). O algoritmo 2.3 apresenta o *Q-learning*.

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) \quad (2.31)$$

```

Inicialize  $V(s, a)$  arbitrariamente
Repita (Para cada episódio)
  Inicialize  $s$ 
  Escolha  $a$  de  $s$  usando uma política derivada de  $Q$ 
  Repita (para cada passo do episódio)
    Tome a ação  $a$ ; observe o reforço  $r$  e o próximo estado  $s'$ 
    Escolha  $a'$  de  $s'$  usando uma política derivada de  $Q$ 
     $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma Q(s', a') - Q(s, a)]$ 
     $s \leftarrow s'$ ;  $a \leftarrow a'$ ;
  Até que  $s$  seja um estado terminal
  
```

Algoritmo 2.2 – Algoritmo Sarsa

```

Inicialize  $V(s, a)$  arbitrariamente
Repita (Para cada episódio)
  Inicialize  $s$ 
  Repita (para cada passo do episódio)
    Escolha  $a'$  de  $s'$  usando uma política derivada de  $Q$ 
    Tome a ação  $a$ ; observe o reforço  $r$  e o próximo estado  $s'$ 
     $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ 
     $s \leftarrow s'$ ;
  Até que  $s$  seja um estado terminal

```

Algoritmo 2.3 – Algoritmo *Q-learning*

2.6.3 Comentários conclusivos para aprendizado por reforço

A teoria de aprendizagem de máquina por reforço, inspirada em teorias de comportamento animal, constitui uma importante fonte de inspiração para a proposição de sistemas inteligentes aplicados a problemas de decisão, em que a única fonte de informação disponível seja o próprio ambiente de atuação do sistema. O campo de abrangência de sistemas dotados de capacidade de aprendizado com base nestas restrições é muito amplo.

Pesquisas relacionadas ao tema têm sido observadas desde os anos 50, mas apenas a partir dos anos 80 é que uma corrente de pesquisa ganhou força e passou a se identificar com o nome Aprendizagem por Reforço. Esta corrente, que estuda os Processos de Decisão de Markov, foi brevemente abordada acima e sobre ela algumas considerações adicionais merecem destaque.

Em princípio, trata-se de uma abordagem diferenciada e interessante dos processos de busca heurística da inteligência computacional clássica, na medida em que incorpora à busca a capacidade de adaptação a mudanças do ambiente, introduzida pelo mecanismo de reforço das probabilidades de escolha dos caminhos mais promissores dentro do espaço de estados do problema. No entanto, como já foi observado, os PDMs carregam muitas das restrições observadas nos métodos da inteligência computacional clássica, dentre os quais restrição de ação a ambientes específicos, ou seja pouca capacidade de generalização do conhecimento aprendido para sua aplicação em outros problemas afins. Além disso, um modelo de representação interno é necessário para guardar informação sobre os estados.

Este modelo pode demandar uma quantidade considerável de recursos computacionais dependendo do problema.

Não obstante a popularidade dos métodos dedicados ao tratamento dos PDMs, outras abordagens têm se desenvolvido considerando a utilização de aprendizagem por reforço, como por exemplo os Sistemas Classificadores, desenvolvidos por HOLLAND (1986) que utilizam um algoritmo genético para a definição de regras e um algoritmo de atribuição de crédito às melhores dentre as regras geradas, denominado *Bucket Brigade*.

FIGUEIREDO (1997) também propõe um modelo de aprendizagem por reforço aplicado a redes neurais nebulosas para o controle de um veículo autônomo, fundamentado na teoria do condicionamento operante, que será apresentado no capítulo 3. Particularmente, a abordagem de aprendizagem por reforço introduzida por FIGUEIREDO (1997) será mantida e estendida neste trabalho.

2.7 Teoria da Seleção de Grupos Neurais

A Teoria da Seleção de Grupos Neurais de EDELMAN (1987), baseia-se em aspectos da teoria da seleção natural de Darwin para explicar questões sobre o comportamento e o funcionamento cerebral (FABRO, 1996). Edelman aponta para interações competitivas entre neurônios na formação seletiva de conexões e agrupamentos, tendendo a uma especialização progressiva nas funções das estruturas neurais envolvidas. Em sua teoria, considera três fases de organização do sistema nervoso, correspondendo à fase de seleção por desenvolvimento, à fase de seleção por experiência e à fase de formação de mapeamentos reentrantes.

Na fase de *seleção por desenvolvimento*, a organização cerebral é construída através de sucessivos processos de divisão e seleção de grupos de neurônios, determinados por fatores filogenéticos. Como resultado, grupos neurais primários são formados, sendo responsáveis pela manifestação de comportamentos inatos no indivíduo (e.g. busca pelo alimento, defesa) e pela regulação de suas funções biológicas básicas (e.g. contrações musculares

involuntárias). Esta fase estabelece os pré-requisitos para o aprendizado do indivíduo. Por isto, muitas das estruturas neurais nela criadas apresentam grande plasticidade.

A fase seguinte corresponde à *seleção por experiência*. Nesta fase, impera a aprendizagem através de interações do indivíduo com o meio que o cerca. Através das experiências com o ambiente externo, as conexões sinápticas dos grupos neurais parcialmente desenvolvidos na fase de seleção por desenvolvimento são agora reforçadas ou enfraquecidas. Com isto, são formados os grupos neurais secundários, que correspondem a estruturas mais especializadas e adaptadas a características do ambiente, e portanto, com menor plasticidade quando comparados aos grupos primários. Em geral, estes dois primeiros grupos neurais são relacionados à formação de mapas de percepção sobre o ambiente.

Na fase de *formação de mapeamentos reentrantes* ocorre a interconexão de diversos repertórios neurais com funcionalidades diferentes. Duas propriedades importantes que emergem desta fase são a capacidade de coordenação de tarefas e a formação de mapeamentos globais, o que implica o desenvolvimento de funções mais complexas que aquelas realizadas pelos grupos primário e secundário. Além disto, por estarem fortemente condicionadas ao ambiente externo, estas estruturas apresentam constante adaptação a mudanças ambientais, correspondendo tal adaptação a ajustes relacionados ao comportamento adquirido pela estrutura, e não a mudanças em seu padrão comportamental.

2.8 Visão sintética do capítulo 2

Neste capítulo foram apresentados, de forma sucinta, os principais assuntos relacionados ao tema do presente trabalho. Dentre as técnicas de inteligência computacional, as redes neurais artificiais são particularmente poderosas para processamento paralelo de informações e adaptabilidade a mudanças, freqüentes em problemas dinâmicos; os algoritmos genéticos proporcionam robustez e qualidade na solução de problemas para os quais se possui pouca informação disponível; a teoria dos conjuntos nebulosos torna possível o tratamento de imprecisões e raciocínio aproximado; e os mecanismos de aprendizagem por reforço constituem uma ferramenta importante de treinamento na ausência de um supervisor.

Algumas considerações sobre comportamento e aprendizagem segundo a ótica da psicologia e da biologia também foram apresentadas. Em particular, as teorias apresentadas neste capítulo fundamentam o modelo proposto neste trabalho, e a ser apresentado nos capítulos seguintes.

Capítulo 3

Sistema de Navegação Autônomo Baseado em Redes Neurais Nebulosas

3.1 Introdução

O sistema de navegação proposto em FIGUEIREDO (1997) baseia-se na Teoria de Seleção de Grupos Neurais (EDELMAN, 1987), que considera que conhecimentos específicos para uma determinada tarefa estejam presentes em seus modelos de solução antes mesmo de iniciada sua execução.

Ao longo do capítulo 3, uma breve apresentação daquele sistema que serviu de base para o desenvolvimento deste trabalho será considerada, com ênfase especial dada à estrutura responsável pela coordenação dos movimentos do robô, bem como aos mecanismos de aprendizagem utilizados por esta estrutura.

3.2 Modelo do robô

O modelo do veículo adotado é apresentado na Figura 3.1. Cada sensor de obstáculo determina a distância entre o robô e o obstáculo mais próximo situado na direção do sensor (ângulo α). O sensor de alvos determina o ângulo β , referenciado no ambiente de navegação como o desvio de trajetória, desvio este que pode ser positivo ou negativo, variando de -180° a $+180^\circ$. Além disso, assume-se um sensor de captura que determina quando um alvo foi atingido e sensores de colisão determinam quando o robô colidiu contra um obstáculo. Em particular, estes dois últimos sensores são entidades lógicas e por isto não aparecem na figura

O posicionamento do robô no ambiente de navegação é realizado exclusivamente com o auxílio dos sensores supracitados, e o sistema é completamente reativo, de forma que nenhum mecanismo de mapeamento global é utilizado no planejamento da trajetória do veículo. Assim, não há necessidade da manutenção de quaisquer estruturas de

representação internas na forma de mapas e as informações sensoriais se convertem diretamente no ângulo de ajuste das rodas com base no comportamento gerado pelo controlador. A extensão para o controle de velocidade será apresentada no capítulo 4.

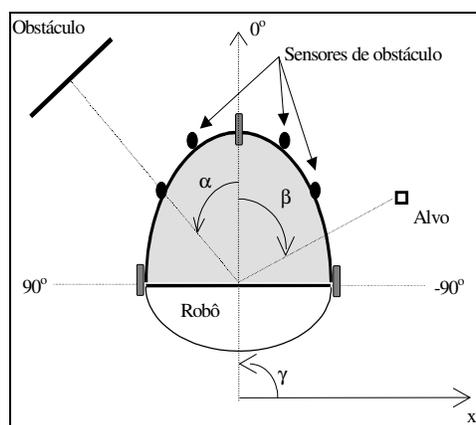


Figura 3.1 – Modelo do veículo e seus sensores de obstáculo e alvo

O modelo do veículo descrito acima é utilizado em um ambiente de simulação cuja principal função é a de prover características de um ambiente de navegação real (físico) através de software, oferecendo uma série de facilidades tanto na montagem do ambiente de navegação como na configuração dos sensores do veículo a ser testado. Este ambiente é apresentado no Apêndice 1.

3.3 O controlador autônomo

3.3.1 Arquitetura do controlador

A arquitetura utilizada é modular e hierárquica, conforme a Figura 3.2. Em um primeiro nível, dois módulos distintos atuam na tomada de decisão, sendo um Módulo de Busca de Alvos (MBA), e um Módulo de Desvio de Obstáculos (MDO). Entretanto, as respostas produzidas por tais repertórios podem ser conflitantes. Por exemplo, considerando-se o ambiente ilustrado na Figura 1.2 (página 7), o MBA determinaria que o robô deveria avançar na direção do alvo, enquanto que o MDO determinaria que o robô deveria deslocar-se na direção contrária à do alvo.

Para tratar este e outros eventuais conflitos, um terceiro módulo, denominado Módulo de Coordenação (MC) situado em um nível hierárquico superior, realiza automaticamente a tarefa de coordenação sobre os demais, ponderando a influência de cada um dos dois módulos de tomada de decisão, de acordo com a situação com que o robô se depara. Inicialmente, o Módulo de Coordenação não possui conhecimento, o qual vai sendo adquirido por aprendizado à medida que o robô interage com o ambiente, sempre se levando em conta o elenco de objetivos a serem atendidos.

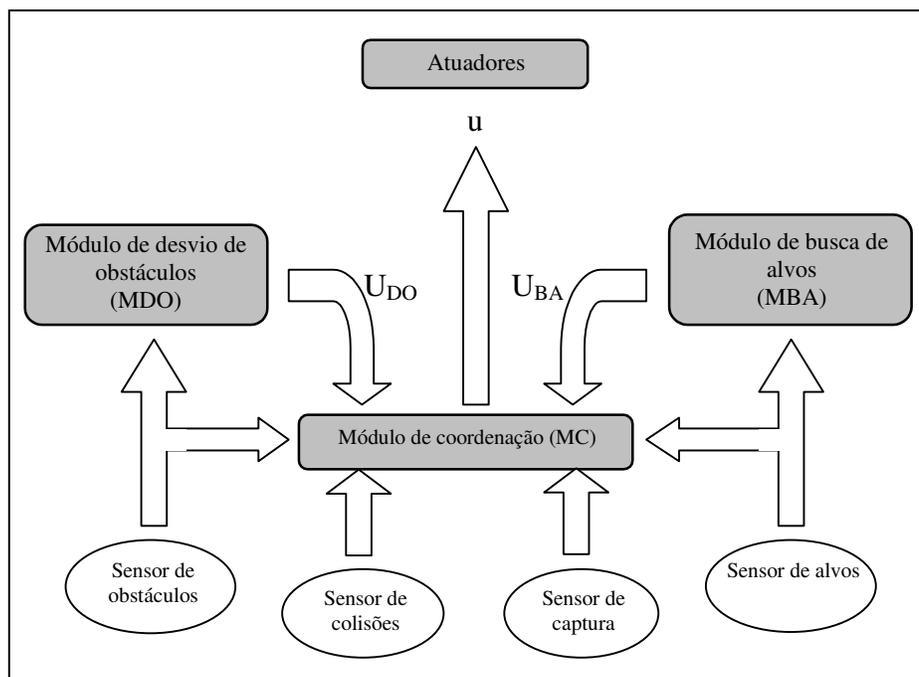


Figura 3.2 – Arquitetura do sistema de navegação autônomo baseado em redes neurais nebulosas

O MDO e o MBA são tidos como estruturas inatas do controlador autônomo, pois se encontram aptos a produzirem sinais de desvio de obstáculos e de busca de alvos antes do início da navegação. No capítulo 5, modelos conexionistas treinados por algoritmos genéticos são considerados de maneira alternativa para estes módulos. Estes módulos instintivos também podem ser descritos a partir de um conjunto reduzido de regras de inferência do tipo: Se <antecedente> então <conseqüente>. Neste sentido, este trabalho, a exceção do capítulo 5, apresenta o MDO e o MBA descritos através do Algoritmo 3.1 e do Algoritmo 3.2, respectivamente, em que os valores de ajuste de direção foram obtidos empiricamente.

O MC compreende uma rede neural nebulosa não-recorrente, conectada aos sensores de obstáculos, e um neurônio nebuloso que processa o sinal proveniente do sensor de alvos. A arquitetura da rede neural nebulosa é apresentada na Figura 3.3. O número de neurônios da primeira camada é determinado durante a navegação, pois a cada colisão um processo de aprendizagem é realizado, dando origem a um novo neurônio. O número de sinapses dos neurônios da primeira camada varia de acordo com o número de sensores de obstáculos.

```
Encontre o sensor de obstáculos que indique o obstáculo mais próximo do robô
Se a distância indicada pelo sensor for menor que 50 ud, então
    Se o obstáculo estiver à esquerda do robô, então
        Virar 10° para a direita
    Senão,
        Virar 10° para a esquerda
Fim
Fim
```

Algoritmo 3.1 – MDO baseado em regras, onde ud representa unidade de distância

```
Encontre o sensor de alvos que registre com maior intensidade o sinal de um alvo
Se não há sinal de alvos ou se o desvio do alvo com relação à direção de navegação for de  $\pm 1^\circ$ , então
    A direção de deslocamento se mantém inalterada
Senão, se o alvo estiver à direita do robô, então
    Se o desvio for menor que  $5^\circ$ , então
        Vira para a direita  $1^\circ$ 
    Senão, se o desvio for menor que  $10^\circ$ , então
        Virar para a direita  $5^\circ$ 
    Senão,
        Virar para a direita  $10^\circ$ 
Fim
Senão, se o alvo estiver à esquerda do robô, então
    Se o desvio for menor que  $5^\circ$ , então
        Vira para a esquerda  $1^\circ$ 
    Senão, se o desvio for menor que  $10^\circ$ , então
        Virar para a esquerda  $5^\circ$ 
    Senão,
        Virar para a esquerda  $10^\circ$ 
Fim
Fim
```

Algoritmo 3.2 – MBA baseado em regras

A notação utilizada para a descrição da arquitetura da rede neural nebulosa, bem como dos algoritmos de aprendizagem utilizados é a seguinte:

- Número de sensores de obstáculos : N ;
- Número de colisões : c ;
- Número de neurônios na primeira camada : Q ;
- Vetor de distâncias a obstáculos : $v = (v_1 \dots v_k \dots v_N)$;
- Peso sináptico de comparação (entrada i e neurônio j) : w_{ij} ;
- Peso sináptico de ponderação (entrada i e neurônio j) : s_{ij} .

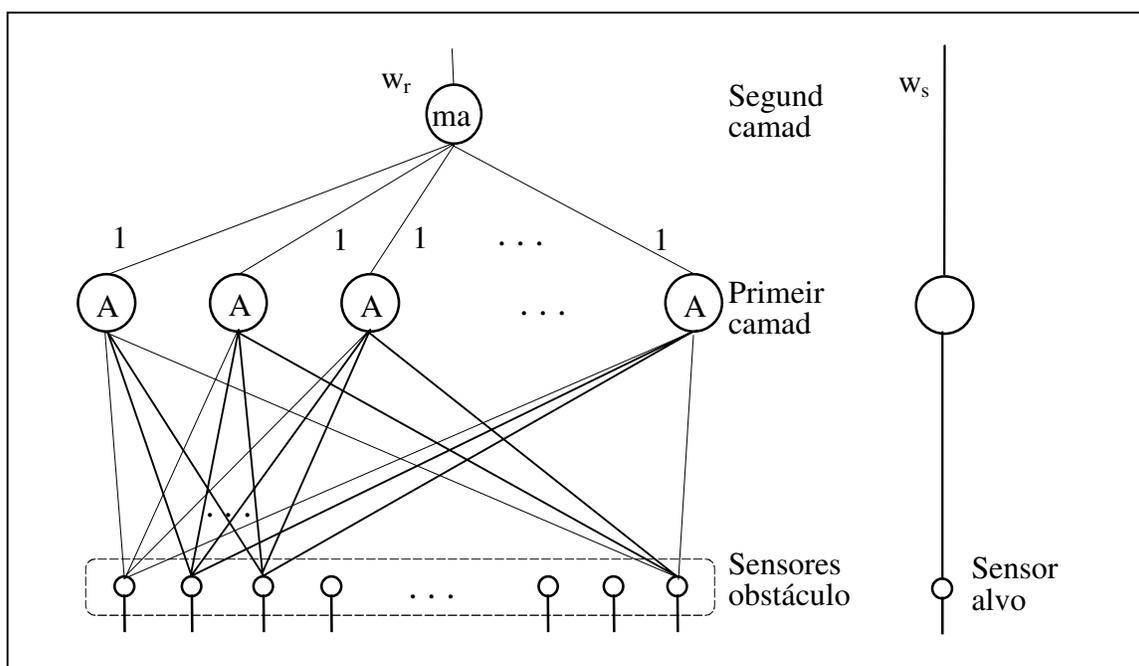


Figura 3.3 – Arquitetura da rede neural nebulosa, onde a entrada da rede é dada por um padrão unidimensional de distâncias a obstáculos

De acordo com o modelo de neurônio ilustrado pela Figura 3.4, as entradas x_i e os pesos sinápticos w_i , $i = 1, \dots, N$, são parâmetros dos conjuntos nebulosos $\Omega_{x_i}(x)$ e $\Phi_{w_i}(x)$, respectivamente. Conforme a equação (3.1), que representa o j -ésimo neurônio da primeira camada da rede durante o processamento sináptico, estes conjuntos nebulosos são comparados via um operador monotônico $\Delta(\cdot)$ e o valor encontrado na operação é

ponderado pelo peso sináptico s_i através da t-norma $T_2(\cdot)$. O peso sináptico w_i representa a memória com relação ao padrão induzido pelas respectivas entradas, enquanto que s_i representa um reforço à modificação da entrada pela sinapse. Após o processamento sináptico, um operador de agregação nebuloso $A(\cdot)$ agrega as entradas e a função de ativação $f(\cdot)$ determina a saída do neurônio.

$$y_j = f\left(\mathbf{A}\left(T_2\left(\Delta(\Omega_{x_i}(x), \Phi_{w_i}(x)), s_i\right)\right)\right) \quad (3.1)$$

Em particular para a equação (3.1), $\Delta(\cdot)$ corresponde ao operador nebuloso de medida de possibilidade conforme a equação (2.16), T_2 é a t-norma produto de acordo com a equação (2.15); $A(\cdot)$ é uma s-norma máximo equivalente à equação (2.12) aplicada a n operandos; e $f(\cdot)$ é a função identidade, dada por $f(x) = x$. Além disso, $\Omega_{x_i}(x)$ e $\Phi_{w_i}(x)$ possuem perfis triangulares (isósceles) e a largura da base é fixa e igual a 3 nos resultados apresentados por FIGUEIREDO (1997).

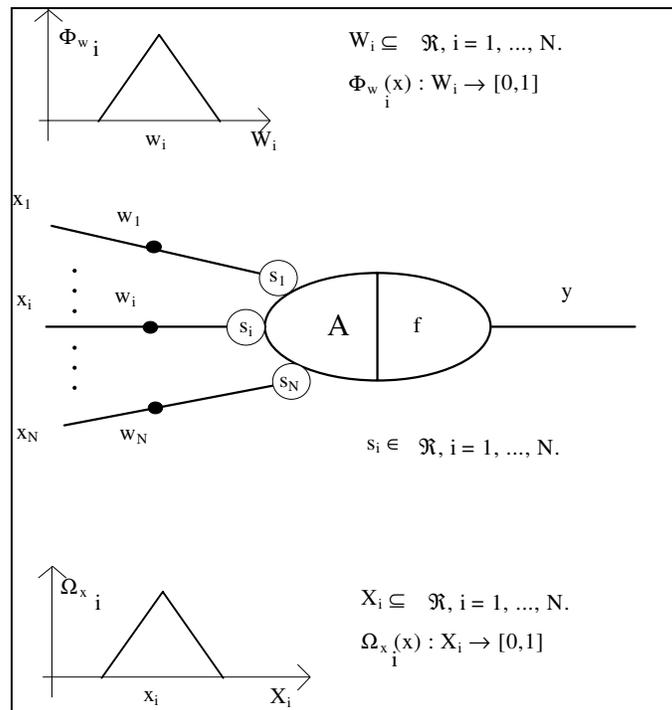


Figura 3.4 – Modelo de neurônio nebuloso

A saída do único neurônio da segunda camada da rede neural nebulosa que compõe o MC é dada pela equação (3.2). A função de agregação deste neurônio é dada pela s-norma máximo $S(\cdot)$ e a função de ativação é a função identidade. A saída Wr , que corresponde ao neurônio mais ativo da camada intermediária, tem por função ponderar o sinal de desvio de obstáculos produzido pelo MDO.

$$Wr = \mathcal{S}_{j=1}^q (y_j) \quad (3.2)$$

Outro componente do MC do sistema autônomo de FIGUEIREDO (1997) é o neurônio nebuloso responsável pela ponderação do sinal de busca de alvos produzido pelo MBA. Este neurônio é semelhante a aquele do modelo apresentado na Figura 3.4, exceto pelo fato de possuir apenas uma entrada v_1 proveniente do sensor de alvos e portanto apenas dois componentes sinápticos: w_1 e s_1 . Sua saída é dada pela equação (3.3). Nos resultados apresentados, utilizou para os conjuntos nebulosos $\Omega_{v_1}(x)$ e $\Phi_{w_1}(x)$ dispersão igual a 10.

$$Ws = S_1(\Delta(T(\Omega_{v_1}(x), \Phi_{w_1}(x)), s_1)) \quad (3.3)$$

A saída u do controlador é determinada pelo único neurônio motor existente no modelo, sendo representada pela expressão (3.4). Nesta expressão, Wr e Ws correspondem às saídas do MC, sendo que Wr corresponde ao sinal de ponderação da saída U_{DO} produzida pelo MDO, ao passo que Ws corresponde ao sinal de ponderação da saída U_{BA} produzida pelo MBA. O sinal u corresponde ao ajuste incremental na direção de deslocamento do robô. Além disso, a forma como Wr age sobre o sinal U_{BA} faz com que, antes de o MC ter sido treinado, e portanto não possuindo neurônios, o robô manifeste apenas o comportamento de busca pelo alvo, com Wr e Ws iguais a zero.

$$u = (1 - Wr) \cdot U_{BA} + (1 - Ws) \cdot Wr \cdot U_{DO} \quad (3.4)$$

3.3.2 Aprendizagem do Módulo de Coordenação

A estratégia de aprendizagem adotada para o MC obedece à teoria do condicionamento operante, freqüentemente observada em sistemas biológicos (THORNDIKE, 1911). Ao colidir, o robô manifesta-se instintivamente, alterando sua direção em 180°. Neste momento, o sinal transmitido pelo sensor de colisão fará o papel de um agente de punição positiva, que foi apresentado no capítulo 2. Tenderá a fortalecer a relação estímulo-resposta determinada pelos sinais dos sensores de obstáculo e pelo sinal do sensor de colisão.

A cada colisão, o MC tem os pesos sinápticos de seus neurônios ajustados. Após algumas colisões, o módulo associa classes de padrões capturados pelos sensores de obstáculos a situações de colisão iminente, gerando, então, comportamentos correspondentes ao desvio de obstáculo. De maneira análoga, a cada vez que um alvo é atingido, um ciclo de aprendizagem também ocorre.

Desta forma, os pesos sinápticos w_i e s_i dos neurônios do MC são ajustados em duas situações específicas do processo de navegação, que serão analisadas separadamente a seguir. Quando há uma colisão, um neurônio é criado, o qual memoriza o padrão unidimensional de distâncias a obstáculos que resultaram da colisão. Da mesma forma, todos os neurônios já existentes no MC e responsáveis pelo desvio de obstáculos têm seus pesos sinápticos modificados em função destes valores, conforme o Algoritmo 3.3.

O resultado da aprendizagem correspondente ao Algoritmo 3.3 é que os valores pequenos de distâncias a obstáculos são associados a situações de colisão. Quanto mais estes valores se repetem ao longo da navegação, tanto mais sua importância é reforçada através dos pesos sinápticos s_i . Os valores utilizados por FIGUEIREDO (1997) para as constantes START e LIMIAR durante a execução do algoritmo correspondem respectivamente a 0,1 e 0,5. Estes valores foram, por conveniência, adotados neste trabalho e embora influam na velocidade da aprendizagem, não interferem qualitativamente nos resultados obtidos.

1 - Inicialização: Inicialize o número de colisões e o número de neurônios na primeira camada: $c(0) = 0$ e $N(0) = 0$.

2 - Ajuste: Se uma colisão ocorre, então faça:

- Se $c(t) = 0$ então

Adicione o primeiro neurônio na primeira camada da rede neural nebulosa e inicialize seus pesos sinápticos:

$$c(t+1) = c(t) + 1$$

$$Q(t+1) = Q(t) + 1$$

Para $i = 1$ até N e $j = Q(t+1)$ faça:

$$s_{ij}(t+1) = \text{START}$$

$$w_{ij}(t+1) = v_i$$

Fim

(Um neurônio relativo à segunda camada também deve ser considerado para a arquitetura da rede neural nebulosa).

Senão

Para $j = 1$ até $Q(t)$ faça: (para cada neurônio da primeira camada)

$$\alpha = c(t) - j + 1$$

Para $i = 1$ até N faça: (para cada sinapse do neurônio)

$$\text{SIMIL} = 0$$

Para $k = 1$ até N faça: (para cada entrada do neurônio)

$$\text{AUX} = \Delta(\Phi_{w_{ij}}(x), \Omega_{v_k}(x))$$

Se $\text{SIMIL} < \text{AUX}$ então

$$\text{SIMIL} = \text{AUX}$$

$$\text{COMP} = k$$

Fim

Fim

Se $\text{SIMIL} > \text{LIMIAR}$ então execute os ajustes:

$$w_{ij}(t+1) = w_{ij}(t) + (V_{\text{COMP}}(t) - w_{ij}(t)) / (1 + s_{ij}(t))$$

$$s_{ij}(t+1) = s_{ij}(t) + (\text{SIMIL} - s_{ij}(t)) / (\alpha + 1)$$

Fim

Fim

Fim

$$C(t+1) = c(t) + 1$$

$$Q(t+1) = Q(t) + 1$$

Para $i = 1$ até N e $j = Q(t+1)$ inicialize os pesos sinápticos do novo neurônio:

$$s_{ij}(t+1) = \text{START}$$

$$w_{ij}(t+1) = v_i$$

Fim

Fim

3 - Retorne para 2.

Algoritmo 3.3 – Ajuste sináptico mediante uma colisão

Outra situação que sujeita o MC a um processo de aprendizagem corresponde à captura de um alvo. Neste caso, da mesma forma que os neurônios responsáveis pela ponderação do sinal de desvio de obstáculos associavam padrões de distância a situações de colisão, agora o neurônio responsável pela ponderação do sinal de busca de alvos associa padrões de desvio na direção de navegação do robô com relação à direção e sentido do alvo como sendo sinais de que o robô esteja caminhando na direção e sentido corretos.

O Algoritmo 3.4 corresponde ao mecanismo de aprendizagem deste neurônio nebuloso. Diferentemente do que ocorre como a aprendizagem vista no Algoritmo 3.3, aqui não há adição de neurônios a cada aprendizagem. Neste caso, um único neurônio é criado quando o primeiro alvo é atingido. A partir de então, futuras aplicações do Algoritmo 3.4 ao MC restringir-se-ão ao ajuste dos pesos sinápticos daquele neurônio.

1 - Inicialização: Inicialize os pesos w_1 e s_1 .

$s_1 = 0$

$w_1 = 0$

2 - Ajuste: Se o veículo alcança o alvo, então faça:

$\beta = \Delta(\Phi_{w_1}(w), \Omega_{v_1}(x))$

Se $\beta > 0$ então:

$s_1(t+1) = s_1(t) + (1 - s_1(t)) / (\beta + 1)$

Fim

Prossiga em 3.

3 - Retorne para 2.

Algoritmo 3.4 – Aprendizagem mediante uma captura

3.3.3 Introdução do Valor Acumulado de Ativação Sináptica

Conforme apresentado acima, no modelo de FIGUEIREDO (1997), o sistema de navegação autônomo do robô utiliza o mecanismo de aprendizagem do MC apenas em duas situações: quando ocorre uma colisão, os neurônios do módulo de coordenação responsáveis pela ponderação da ação de desvio de obstáculos são punidos; quando um alvo é atingido, o

neurônio do módulo de coordenação responsável pela ponderação da ação de busca de um alvo é recompensado.

Com isto, as estratégias de recompensa e punição são manipuladas de maneira desequilibrada pelo sistema. Os neurônios de ponderação da ação de desvio nunca são recompensados por uma manobra bem sucedida do robô, restando-lhes apenas as punições decorrentes de uma colisão. Com o passar do tempo, o excesso de punições leva o robô a desenvolver uma espécie de aversão excessiva à aproximação de obstáculos.

Por outro lado, o neurônio responsável pela ponderação da ação de busca de alvos nunca é punido por ter influenciado em uma decisão que leva o robô a uma colisão, recebendo apenas a recompensa pelo alvo atingido. O excesso de recompensas concedidas a cada captura de um alvo reforça sobremaneira a influência do neurônio na decisão final do módulo de coordenação. Ocorre então que, mesmo diante de um obstáculo, a resposta dos neurônios de ponderação da ação de desvio já não é suficiente para evitar uma colisão.

Para eliminar estas deficiências, é preciso que o mecanismo de aprendizagem do robô distribua de maneira mais uniforme, tanto as punições quanto as recompensas, sobre as sinapses do MC. Trata-se de resolver um problema de atribuição de créditos. Em outras palavras, a punição ou recompensa deve tornar-se de algum modo proporcional à responsabilidade de cada neurônio ao longo do processo de tomada de decisão. Para tanto, introduziu-se ao modelo anteriormente apresentado uma memória de ativação, ou seja, um meio de indicar a ocorrência de uma ativação de determinada sinapse instantes após esta ter ocorrido. A modelagem desta memória incorpora uma nova propriedade sináptica, denominada valor acumulado de ativação sináptica (VAAS).

A Figura 3.5 apresenta o novo modelo de neurônio em que o peso sináptico r_i representa o VAAS. As demais propriedades deste novo modelo obedecem às mesmas definições feitas para o modelo da Figura 3.4. Com esta nova propriedade, a cada vez que a i -ésima sinapse do neurônio se torna ativa, esta tem parte do seu valor acumulado no peso r_i . Durante um período de ausência de ativação, o VAAS sofre um decremento segundo uma função não

linear. O valor de VAAS não influi na saída do neurônio, participando apenas dos ajustes dos pesos sinápticos w_i e s_i como será visto a seguir. Entretanto, diferentemente do que ocorre com os pesos sinápticos w_i e s_i , o peso sináptico r_i é ajustado a cada vez que o valor de ativação $\varphi_i(t)$ da sinapse ultrapassar o valor de $r_i(t)$, e não apenas em situações específicas. A forma de ajuste dos pesos sinápticos r_i é tal como mostra a equação (3.5).

$$r_i(t+1) = r_i(t) + \begin{cases} \eta \cdot (\varphi_i(t) - r_i(t)), & \text{se } \varphi_i(t) > r_i(t) \\ -\mu \cdot r_i(t) & \text{caso contrário} \end{cases} \quad (3.5)$$

Na equação (3.5) nota-se dois fatores de aprendizagem distintos. O fator η é denominado fator de ganho por determinar a intensidade do valor acumulado a cada ativação da sinapse. Por outro lado, o fator μ é denominado fator de perda, por determinar a intensidade com que o valor acumulado decai a cada iteração em que a sinapse permanece inativa. A ativação da i -ésima sinapse do neurônio no instante t é dada por $\varphi_i(t)$ e é expressa pela t-norma T_2 da equação (3.1). Na equação (3.1), no entanto, a t-norma é aplicada a todas as sinapses do neurônio e a s-norma máximo $A(\cdot)$ faz a agregação destes valores..

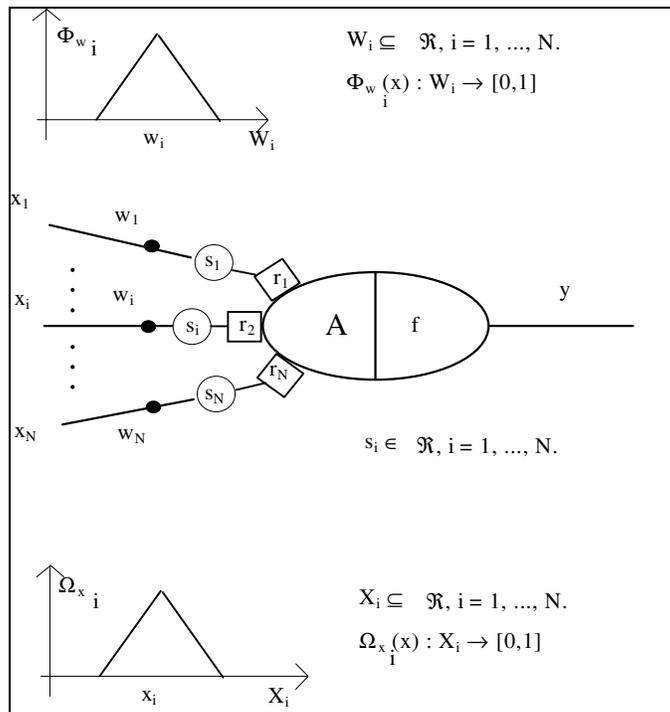


Figura 3.5 – Novo modelo de neurônio nebuloso com VAAS

O VAAS é utilizado para auxiliar mecanismos complementares de aprendizagem, a serem aplicados conjuntamente com os mecanismos apresentados pelos algoritmos 3.3 e 3.4. Assim, em virtude de uma colisão, além do Algoritmo 3.3, um outro processo de aprendizagem ocorre no MC, desta vez sobre o neurônio responsável pela ponderação do sinal do MBA. A finalidade deste mecanismo é apurar a responsabilidade do neurônio de ponderação do sinal de busca de alvos na colisão do robô, ou seja, verificar se a saída deste neurônio afetou a decisão do MC impedindo-o de fazer com que o robô se desviasse do obstáculo.

Esta análise da responsabilidade do neurônio de ponderação do sinal de busca de alvos em uma colisão é realizada através da análise do VAAS deste neurônio. Caso o VAAS esteja elevado, isto indica que o neurônio esteve muito ativo pouco antes da colisão ocorrer, interferindo no processo de decisão do módulo. Assim, o neurônio é punido, tendo seu peso sináptico s_1 inibido proporcionalmente à intensidade de seu VAAS, como na equação (3.6).

$$s_1(t+1) = s_1(t) - r_1 \cdot s_1(t) \quad (3.6)$$

De forma análoga, a cada vez que um alvo é atingido, um processo complementar de aprendizagem passa a ser desencadeado sobre os neurônios de ponderação do sinal de desvio de obstáculo do MDO. Assim como durante uma colisão apurava-se a influência do neurônio de ponderação do sinal de busca naquela decisão, semelhante fato ocorre quando de uma captura. Entretanto, o que se procura verificar agora é a influência dos neurônios de desvio na captura do alvo, de forma que aqueles que mais tiverem colaborado com o processo sejam recompensados.

Agora, os VAAS das sinapses de todos os neurônios de ponderação do sinal de desvio são verificados. Um valor alto indica que a sinapse colaborou bastante para a captura do alvo, tendo então seu peso s reforçado na proporção de sua colaboração. O Algoritmo 3.5 expressa a aprendizagem dos neurônios de desvio com base na análise de seus VAAS.

Para cada neurônio nebuloso de ponderação do sinal de desvio de obstáculos N'_j faça:

Para cada peso sináptico s_{ij} faça:

$$s_{ij}(t+1) = s_{ij}(t) + r_{ij} \cdot (1 - s_{ij}(t))$$

Fim

Fim

Algoritmo 3.5 – Aprendizagem com VAAS após alvo ter sido atingido

3.4 Simulações e testes

Simulações realizadas com o sistema de navegação autônoma demonstraram que o robô, apesar de inicialmente não possuir qualquer capacidade de coordenação sobre os movimentos que lhe possam guiar através de um ambiente desconhecido, consegue com o decorrer da navegação, mediante processos de aprendizagem, adquirir tal capacidade, deslocando-se através do ambiente sem colidir contra os obstáculos e sempre buscando o alvo. Estas simulações não contemplam ruído nos sinais dos sensores.

Uma destas simulações é apresentada na Figura 3.6. Neste e nos demais exemplos de simulação apresentados ao longo deste trabalho, convencionou-se que os locais de colisão sejam representados por pontos vermelhos, os alvos atingidos por pontos amarelos, os alvos não atingidos por pontos azuis, os obstáculos por retângulos pretos, a trajetória por uma linha contínua preta e o robô por um triângulo isósceles preto. Embora nenhuma escala tenha sido apresentada nas figuras que ilustram ambientes de navegação, pode-se utilizar como referência a largura do robô, que equivale a 20 unidades de deslocamento (ud.). Além disso, durante as simulações, os alvos são apresentados seqüencialmente, um a um ao robô, sendo que somente quando um alvo é atingido o próximo é introduzido no ambiente e os obstáculos podem ser justapostos para formarem ambientes complexos. Por fim, salvo quando mencionado em contrário, nas simulações realizadas utilizou-se 37 sensores de obstáculos e 1 sensor de alvo. A partir do capítulo 4, conforme será convenientemente descrito, a arquitetura dos sensores de alvos será modificada.

Conforme a Figura 3.6, uma vez tendo adquirido conhecimento necessário para desviar das paredes do ambiente, o robô passa a deslocar-se com sucesso na direção do alvo. A figura

mostra que após ter vencido a fase de aprendizagem, representada pela série de colisões que ocorre logo no início da navegação, o robô passa a percorrer o ambiente sem colidir, e termina por atingir todos os alvos. A velocidade do robô é constante e igual a 0,2 unidades de deslocamento por iteração (ud./i.) e a largura do robô equivale a 16 unidades de deslocamento (ud.).

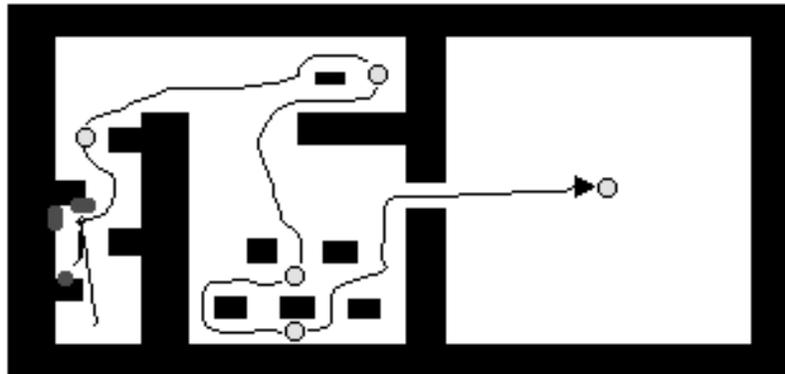


Figura 3.6 – Simulação de navegação com o controlador autônomo baseado em redes neurais nebulosas

A Figura 3.7 ilustra como ocorre o processo de aprendizagem dentro de um neurônio da rede neural nebulosa do MC após algumas colisões. Cada gráfico da figura representa um passo de aprendizagem, o que significa uma colisão no ambiente. O neurônio em questão é o primeiro neurônio da rede, criado por ocasião da primeira colisão do robô, sendo que os demais neurônios não são analisados aqui. As barras amarelas correspondem aos valores registrados pelos pesos sinápticos w_i do neurônio, ao passo que os pontos vermelhos representam seus pesos s_i .

É interessante notar da representação gráfica dos pesos sinápticos w_i , os quais capturam os sinais dos sensores de obstáculos no momento de uma colisão, que a implementação destes sensores pressupõe uma situação ideal na qual todos são capazes de registrar a distância do obstáculo à sua direção. Em uma situação real, é provável que nem todos os sinais emitidos pelos sensores retornem a eles, o que acarretaria perda na leitura daquele sensor. O retorno do sinal emitido pelo sensor depende, dentre outras coisas, da textura do obstáculo e o ângulo de incidência, no obstáculo, do sinal emitido.

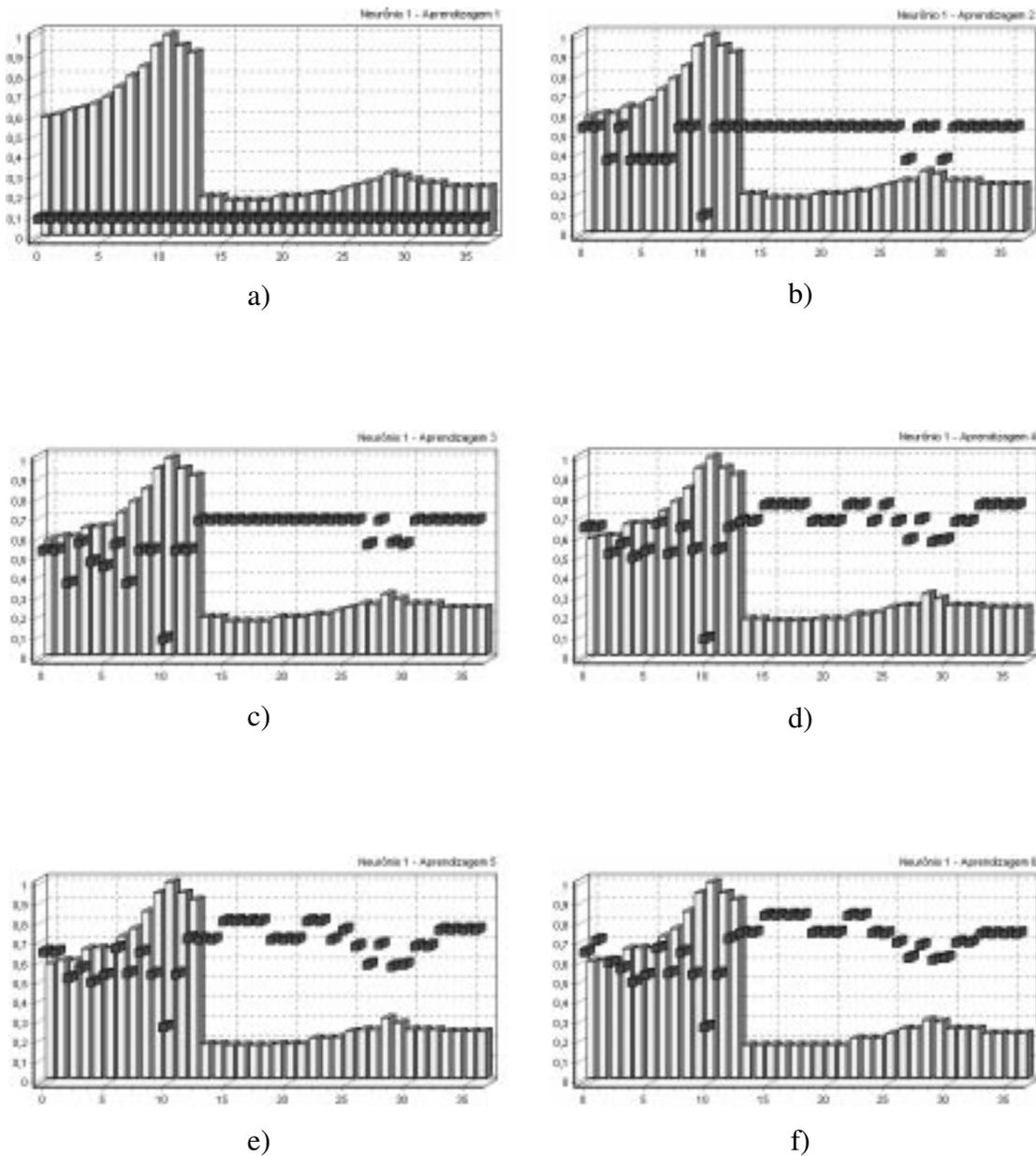


Figura 3.7 – Seqüência de valores para os vetores \vec{w}_1 e \vec{s}_1 durante processo de aprendizagem do neurônio 1 do MC

Não obstante, o fato de se estar assumindo uma situação ideal para a representação dos sensores de forma alguma deve interferir na eficácia do processo de aprendizagem caso ele seja estendido para uma situação prática. Isto pode ser afirmado pelo fato de que o que realmente importa durante o condicionamento operante realizado no controlador autônomo não corresponde a uma codificação particular dos sinais dos sensores, mas sim à formação

de um padrão sensorial, qualquer que seja ele, que diferencie uma situação de colisão iminente de uma situação de navegação sem riscos de colisão. Assim, o fato de este padrão não contemplar as leituras de todos os sensores não invalida a aprendizagem, desde que este padrão se repita de forma aproximadamente igual em outras situações similares à que gerou a aprendizagem, como é de se esperar que ocorra mesmo em um ambiente ruidoso.

A situação (a) apresenta o neurônio logo após sua criação. Os pesos sinápticos w_i , $i = 1, 2, \dots, 37$, capturam a configuração dos sensores de obstáculo, sendo os pesos s_i iniciados com o valor START, como no Algoritmo 3.3. Ao longo das próximas colisões, os valores dos pesos w_i variam pouco, dado que sua função é capturar um padrão de distâncias que deve ser mantido pelo robô. Em particular, o padrão capturado pelo neurônio da Figura 3.7 indica uma colisão lateral à direita, considerando que o peso w_{37} esteja ligado ao sensor mais à direita do robô. Por outro lado, os pesos s_i sofrem uma grande variação a cada nova aprendizagem, pois sua função é a de reforçar as sinapses que registram situações que ocorrem com frequência em cada colisão.

A variação dos pesos sinápticos s_i ao longo dos processos de aprendizagem, visa dar mais expressividade a distâncias pequenas de obstáculos registradas nos pesos w_i . Esta variação pode ser observada ao longo dos ajustes sinápticos no neurônio apresentado. Seis colisões podem ser identificadas na Figura 3.6, o que indica que além do neurônio apresentado nos gráficos a – f, outros seis neurônios foram criados ao longo desta simulação, sendo um a cada colisão.

A criação do conceito de VAAS, assim como descrito anteriormente, está associada a um problema identificado no processo de aprendizagem do modelo de FIGUEIREDO (1997) que prejudicava o desempenho do robô depois de determinado período de navegação. Inicialmente o robô começava sua navegação na direção do alvo sem estratégias de coordenação previamente desenvolvidas. Após algumas colisões, o MC sintetizava tais estratégias e passava a desviar-se de obstáculos de forma que as colisões não mais ocorriam durante a navegação. A partir de então, a rede neural nebulosa destinada a ponderar o sinal de desvio de obstáculos não mais sofria aprendizagem.

Entretanto, o mesmo não ocorria com o neurônio nebuloso destinado à ponderação do sinal de busca de alvos, uma vez que a cada alvo capturado, esta era reforçada de algum modo. Como resultado, em algum momento da simulação o sinal de ponderação de busca de alvos se tornaria forte a tal ponto que se sobreporia ao sinal de ponderação de desvio, mesmo diante de uma situação de colisão iminente. Esta situação demonstra um retrocesso na estratégia de navegação do robô, uma vez que este tenha se tornado instável como no início da navegação.

A Figura 3.8 apresenta um exemplo comparativo entre o sistema proposto por FIGUEIREDO (1997) (a) e o sistema com os neurônios do MC acrescentados do VAAS (b). Em ambos os casos, os módulos de coordenação já haviam sido treinados e uma saturação do neurônio que pondera o sinal do MBA havia sido induzida. Em (a), a ativação excessiva do neurônio nebuloso ponderador do sinal do MBA é responsável por três colisões. Em (b), a ativação excessiva do mesmo neurônio nebuloso também é a causa da única colisão existente. No entanto, neste último caso, a colisão desencadeia um processo de aprendizado capaz de determinar as responsabilidades e agir no sentido de evitar a repetição deste comportamento indesejável.

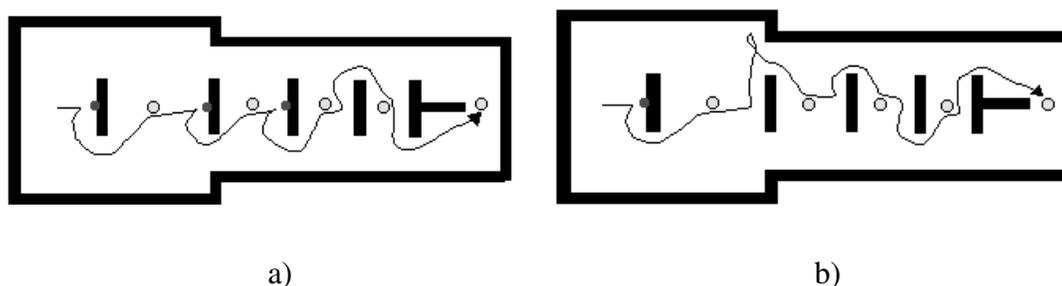


Figura 3.8 – Resultados comparativos entre controladores: mesma condição inicial e a) sem VAAS; b) com VAAS

A Figura 3.9 apresenta outro exemplo comparativo parecido com o da Figura 3.8, agora em um ambiente diferente. Assim como no exemplo anterior, os dois controladores já haviam sido treinados antes das simulações ilustradas. No caso das simulações da Figura 3.9, os robôs foram treinados para se desviarem de obstáculos e então colocados, numa segunda

fase de treinamento, em um ambiente sem obstáculos, mas com 25 alvos. O objetivo desta segunda fase de treinamento era saturar o neurônio de ponderação do sinal do MBA.

Como pode ser observado, na situação (a), o controlador não dotado do recurso de VAAS dá mais importância para a tarefa de atingir o alvo do que a de desviar-se dos obstáculos, mesmo quando sua aproximação aos obstáculos se torna perigosa. Com isto, duas colisões ocorrem. Após atingir o segundo alvo, no entanto, consegue restabelecer seus parâmetros de coordenação e desviar-se dos obstáculos, atingindo o terceiro alvo sem colidir uma terceira vez. No entanto, o fato de o robô ter conseguido se livrar da terceira colisão na situação (a) não implica que tenha resolvido o problema da saturação do neurônio ponderador do sinal do MBA, conforme será visto adiante.

Na situação (b), uma navegação muito similar àquela realizada na situação (a) ocorre. Agora porém, o controlador apresenta o VAAS nas sinapses do MC. Ocorre que após a primeira colisão, fruto da saturação do neurônio ponderador do MBA, este é punido e não há a ocorrência de uma segunda colisão. Pode-se concluir portanto, que o VAAS embora não elimine a possibilidade de colisões causadas pela saturação da aprendizagem por alvos atingidos, é capaz, no entanto, de reduzir o número de suas ocorrências no modelo de controlador apresentado por FIGUEIREDO (1997).

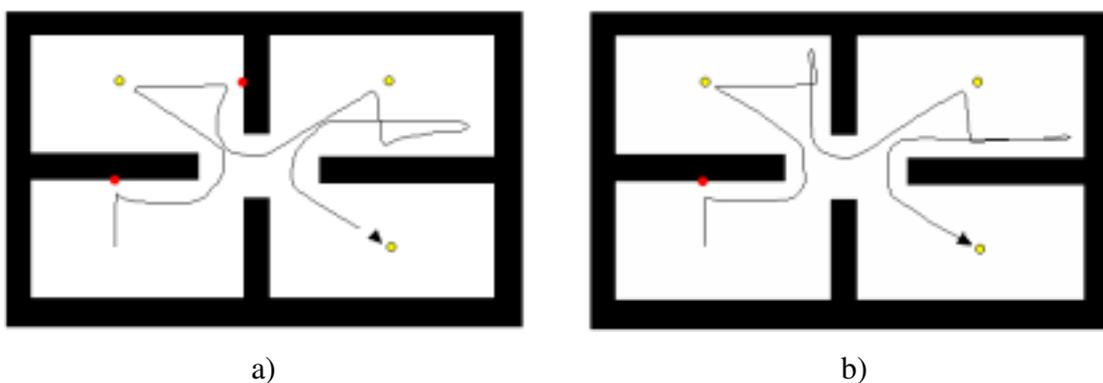


Figura 3.9 – Simulação comparativa em um novo ambiente para os controladores: mesma condição inicial e a) sem VAAS; b) com VAAS

Os gráficos apresentados na Figura 3.10 ilustram as ações do MC na simulação da Figura 3.9 (a), momentos antes das colisões observadas naquela simulação. As barras amarelas

representam as distâncias do robô com relação ao obstáculo mais próximo a cada iteração. A série vermelha representa o sinal W_r e a série azul representa o sinal W_s ao longo das iterações. São registradas em cada gráfico as 50 iterações que precedem a colisão, sendo que a última iteração apresentada registra a iteração imediatamente anterior à colisão.

O gráfico apresentado em (a), na Figura 3.10, corresponde ao registro da primeira colisão. Observa-se neste gráfico que o sinal W_s é excessivamente forte e mesmo que W_r também se torne forte conforme o obstáculo se torne cada vez mais próximo, não há tempo para uma reação efetiva de desvio e o robô colide. Antes da segunda colisão (b), uma situação muito parecida com a da primeira ocorre. Entretanto, agora W_r está mais forte devido à primeira colisão. Na próxima vez que o robô se aproxima de um obstáculo, W_r é ainda mais forte e a colisão é evitada. No entanto, W_s ainda continua forte, pois o neurônio que o produz não é punido após as colisões.

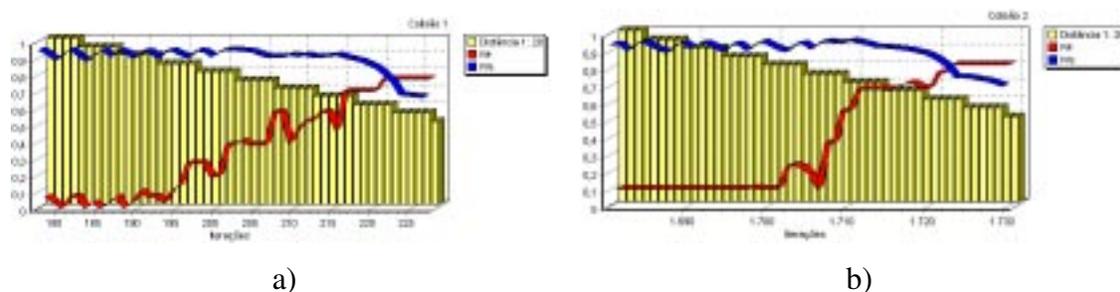


Figura 3.10 – Resposta do MC sem VAAS: a) antes da primeira colisão; b) antes da segunda colisão

Os gráficos apresentados na Figura 3.11 ilustram as ações do MC na simulação da Figura 3.9 (b), momentos antes da primeira colisão e do primeiro alvo ser atingido. O gráfico em (a) é análogo aos apresentados na Figura 3.10. Nota-se que antes da primeira colisão, o sinal W_s estava forte, assim como no caso da Figura 3.10 (a). Com isto a primeira colisão não pode ser evitada. No entanto, após esta colisão, o neurônio que produz W_s é punido e seu sinal é enfraquecido, não levando à ocorrência de outras colisões.

Assim, como uma segunda colisão não é observada nesta simulação, utilizou-se o gráfico apresentado em (b) para comprovar que o sinal W_s realmente se enfraquece com a punição

dirigida pelo VAAS. Este gráfico apresenta a ativação do MC momentos antes de um alvo ser atingido. No caso apresentado, trata-se do primeiro alvo atingido na simulação da Figura 3.9 (b). A série vermelha e a série azul mantêm o mesmo significado que no gráfico em (a), mas as barras amarelas neste gráfico representam o ângulo de desvio, em graus, do robô com relação ao alvo em cada iteração, de acordo com a escala apresentada.

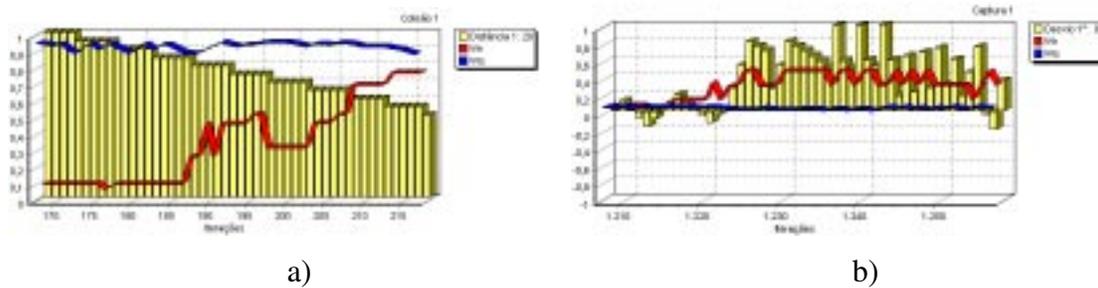


Figura 3.11 – Resposta do MC com VAAS: a) antes da primeira colisão; b) antes da primeira captura

Fica evidente pelo gráfico (b) da Figura 3.11 que W_s foi enfraquecido em decorrência da punição aplicada, após a primeira colisão, ao neurônio que o produz, uma vez que o sinal tem intensidade de menos de 0,2 mesmo estando o robô indo diretamente ao encontro do alvo. O sinal W_r apresenta alguma ativação neste gráfico e conforme os mecanismos de aprendizagem apresentados neste capítulo, os neurônios que o produzem são recompensados de acordo com seu VAAS após o alvo ter sido atingido.

3.5 Visão sintética do capítulo 3

Constitui o objetivo central deste trabalho a concepção de um sistema de controle que permita a um robô móvel deslocar-se através de ambientes desconhecidos visando atingir objetivos determinados, representados no presente trabalho por alvos a serem atingidos, sem que haja a ocorrência de colisões contra eventuais obstáculos que interfiram na trajetória do robô, e que seja independente de influências externas.

Neste capítulo, o sistema de FIGUEIREDO (1997) foi apresentado, o qual servirá de base às propostas a serem apresentadas ao longo dos próximos capítulos. Também foi apresentado neste capítulo uma contribuição original, que consiste no acréscimo de um parâmetro às

sinapses dos neurônios do Módulo de Coordenação do controlador autônomo. Sua função é corrigir uma anomalia presente no mecanismo de aprendizagem do sistema original que o levava a uma situação de instabilidade com o decorrer das simulações, conforme apresentado nas Figuras 3.8 (a) e 3.9 (a).

Em decorrência deste aperfeiçoamento, resultados de simulações apontam não apenas um aumento da robustez do sistema, mas também uma maior velocidade de adaptação às condições adversas que levavam à instabilidade no sistema original. O sistema se tornou capaz de reconhecer tal instabilidade e eliminá-la logo após sua identificação. Isto equivale a dizer que o controlador autônomo é capaz de reconhecer uma deficiência em si mesmo e corrigir-se.

Capítulo 4

O Mecanismo de Controle de Velocidade

4.1 Introdução

Diversas vantagens podem ser enumeradas em um modelo que considera o papel da velocidade no problema da navegação autônoma: maior flexibilidade nas manobras de desvio de obstáculos com a diminuição da velocidade, o que permite uma navegação mais segura; melhor aproveitamento de condições favoráveis à aproximação do alvo, como trechos do percurso sem obstáculos, permitindo um aumento da velocidade; e possibilidade de controlar a aproximação do alvo (CRESTANI *et al.*, 2001).

Dentre os modelos de navegação autônoma propostos na literatura, é comum supor que a velocidade seja um parâmetro constante (LI *et al.*; 1997; TSCHICHOLD-GÜRMAN, 1997; EPSTEIN, 1998; BRAGA & ARAÚJO, 1998; FOX & BURGARD, 1998; IWAKOSHI *et al.* 1998; JERBIC *et al.*, 1999; LEBETEL *et al.*, 2000; FARIA & ROMERO, 2000; YANG & MENG; 2000; JIM & GILES, 2001; SANTOS *et al.*, 2001), não sendo levada em consideração pelo mecanismo de controle. Nestes modelos, apenas a direção de deslocamento do robô é manipulada.

Embora alguns modelos apresentem mecanismos de variação de velocidade (NISHIMURA & IKEGAMI, 1997; NOGUCHI & TERAQ, 1997; FREIRE *et al.*, 1998; NOLFI & FLOREANO, 1999; LEE, 1999; SONG & SHEEN, 2000), freqüentemente ela não é enfatizada como um parâmetro central do desempenho da navegação do robô. Em alguns casos, a variação da velocidade é meramente utilizada como uma forma de controle direcional por meio de sua aplicação individual nas rodas. Além disso, há casos em que a política de controle da velocidade do robô não é sintetizada pelo sistema, mas imposta pelo projetista (AGUIRRE & GONZÁLEZ, 2000), o que impede o emprego do termo “autônomo” ao sistema, conforme SMITHERS (1997).

Neste sentido, FLOREANO & MONDADA (1996, 1998), descrevem um modelo interessante de sistema de controle para robôs móveis baseado em uma rede neural treinada por um algoritmo genético, sendo a velocidade de deslocamento do robô um dos parâmetros considerados na função de avaliação. Entretanto, embora o robô aprenda a controlar sua velocidade, o treinamento evolutivo é lento, de acordo com os resultados descritos. Além disso, nem mesmo os indivíduos mais adaptados conseguem atingir velocidades próximas da máxima permitida.

A idéia básica por trás de um mecanismo de controle de velocidade deve ser a de induzir o robô a aumentar a sua velocidade sempre que o ambiente assim o permitir, ou seja, quando estiver distante de obstáculos ou de um alvo. Por outro lado, conforme identifica a proximidade de um obstáculo, a velocidade do robô deve ser diminuída, de forma que este tenha mais tempo de ajustar sua direção de deslocamento, desviando seu curso com maior segurança.

O mesmo deve valer quando o robô nota a aproximação de um alvo, uma vez que não é desejável que o alvo seja atingido a uma grande velocidade. Esta condição é adotada pois geralmente um alvo está associado a um local onde alguma tarefa deve ser executada. Sabe-se, portanto, que o controle da velocidade deve ser função da motivação pela realização de seu objetivo de atingir o alvo e da cautela quanto à colisão contra obstáculos e aproximação do alvo.

Para que este controle de velocidade seja possível, é necessário que o controlador seja capaz de identificar a aproximação ou o distanciamento de um obstáculo, ou de um alvo, com relação ao robô, e além disso reconhecer o significado desta informação no contexto dos objetivos do sistema. Uma vez que o controlador possua estas habilidades, controlar adequadamente a velocidade de deslocamento do robô deixa de ser uma tarefa complexa.

KUBOTA *et al.* (2001) levam em consideração estas idéias ao proporem um modelo de controlador baseado em algoritmos genéticos e aprendizagem de regras nebulosas. O controlador reativo compreende um conjunto de regras nebulosas que relacionam estímulos

sensoriais e sinais de atuação sobre a direção e a velocidade do robô. O algoritmo genético promove a adaptação das regras a situações de navegação gerais e um mecanismo de aprendizagem por interação com o ambiente realiza um ajuste refinado das regras durante a navegação do robô.

No modelo de controlador apresentado no capítulo 3, o sistema de controle autônomo baseado em redes neurais nebulosas controlava a direção de deslocamento do robô baseando-se na distância do mesmo com relação a um obstáculo, e na direção relativa do alvo a ser atingido. Portanto, o modelo já incorpora a noção de distância do robô a um obstáculo. Por outro lado, a única medida sensorial disponível ao sistema com relação ao alvo corresponde à sua direção relativa ao robô. Torna-se claro desta forma, que o sensor dedicado ao rastreamento do alvo no ambiente deve ser adaptado de forma a permitir que sua distância relativa ao robô também possa ser medida.

4.2 Modificação do sensor de rastreamento de alvos

O sensor de rastreamento de alvos utilizado no modelo de controlador de FIGUEIREDO (1997) fornecia uma única medida angular que indicava a direção do alvo no ambiente com relação ao robô. Deste fato segue que o neurônio do MC responsável pela ponderação do sinal de busca de alvos possui apenas uma única sinapse. Para a criação de um modelo em que não apenas a direção relativa do alvo, mas também sua distância relativa ao robô seja considerada, utilizou-se um novo conceito de sensor de alvos.

Os sensores de obstáculos são distribuídos ao longo da parte frontal do robô e detectam a distância do robô com relação ao obstáculo posicionado na direção de cada sensor. De forma análoga, os novos sensores de alvos correspondem a um conjunto de antenas (ou a uma única antena giratória) posicionadas em torno do robô para determinar o ângulo de desvio entre a direção de deslocamento do robô e a direção do alvo no ambiente. A existência de múltiplos sensores de obstáculos e de alvos permite a formação de padrões de sensoriamento vetoriais, os quais fornecem informações pontuais (relativas a cada sensor) e também relacionais (relativas ao conjunto ordenado de sensores).

Supõe-se, portanto, que o alvo possui anexado a si um emissor de ondas eletromagnéticas a serem detectadas pelos sensores de alvos do robô. A partir da conjunção das medidas sensoriais de todas as antenas, pode-se detectar a direção relativa de um ou mais alvos em qualquer ponto do ambiente, mesmo que alguns destes alvos estejam ocultos por um obstáculo. Além disto, a variação na intensidade do sinal captado pelas antenas indica se o alvo se aproxima ou se distancia do robô durante seu deslocamento. A Figura 4.1 ilustra em (a) o modelo até então utilizado para o sensor de alvos, e em (b) o modelo utilizado a partir de agora.

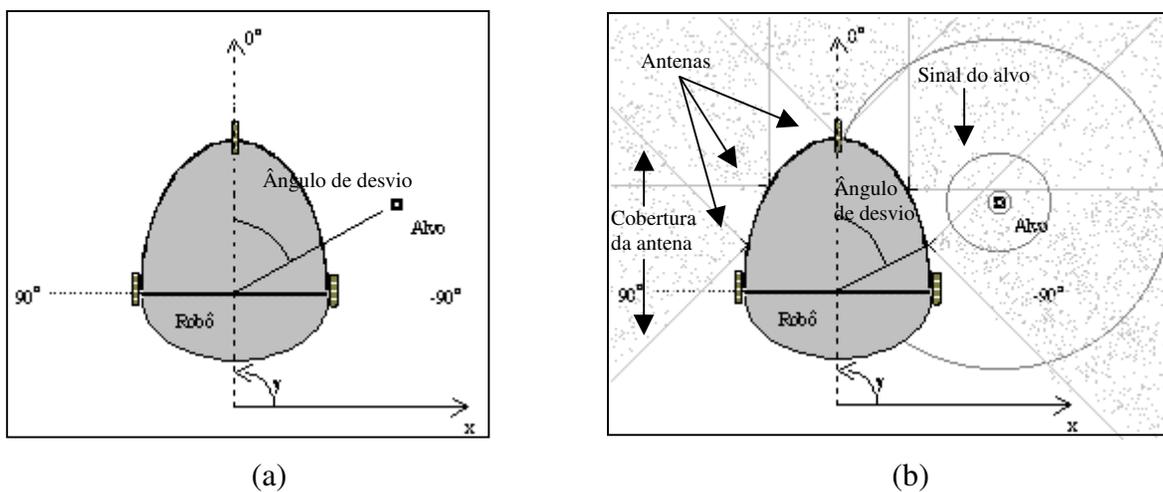


Figura 4.1 – Dois modelos de sensores de alvos: a) Modelo utilizado em Figueiredo (1997); b) Modelo com múltiplas antenas, utilizado neste trabalho.

Outra vantagem apresentada por este novo aparato sensorial, com relação ao modelo anterior, é que agora se tem um conjunto de informações sensoriais, ao invés de uma única informação. Além disso, no que se refere ao mecanismo de controle de velocidade apresentado, abre-se caminho a modificações essenciais na arquitetura do controlador autônomo apresentado no capítulo 3, conforme será descrito a seguir. O número de sensores de alvo utilizado nas simulações apresentadas de agora em diante neste trabalho é 37.

4.3 Extensões no modelo do controlador autônomo

Diversas modificações no modelo original do controlador autônomo do robô, além daquela realizada sobre seu aparato sensorial, devem ser consideradas para a implantação do mecanismo de controle de velocidade. Estas modificações contemplam tanto a arquitetura quanto os mecanismos de aprendizagem do MC. Em primeiro lugar, as duas ações que devem ser realizadas pelo controlador, ou seja, o ajuste na direção e na velocidade de deslocamento do robô, são de natureza distinta, o que implica a criação de um novo neurônio motor com ação exclusiva sobre a velocidade do robô, o qual será apresentado em maiores detalhes mais adiante.

Além disso, assim como no controlador apresentado no capítulo 3 havia dois módulos inatos dedicados ao ajuste direcional (MBA e MDO) e que respondiam pelos dois comportamentos antagônicos de busca de alvos e desvio de obstáculos, os quais eram coordenados por um módulo não-inato situado em um nível hierárquico superior (MC), agora também vai haver dois módulos inatos dedicados ao ajuste da velocidade do robô e que devem ser incorporados à nova arquitetura do controlador, os quais também respondem por comportamentos antagônicos a serem coordenados pelo MC.

Estes dois novos módulos são denominados Módulo de Incremento de Velocidade (MIV), que produz o sinal U_{IV} de incremento na velocidade atual da navegação e Módulo de Decremento de Velocidade (MDV), que produz o sinal U_{DV} de decremento de velocidade. O novo modelo arquitetural do controlador autônomo, apresentado na Figura 4.2, é portanto uma extensão daquele apresentado na Figura 3.2. Os dois novos módulos inatos incorporados à arquitetura do controlador são estimulados por um sensor de velocidade, que também foi concebido para a nova arquitetura.

Exatamente por serem antagônicos, estes dois sinais devem ser ponderados de acordo com a situação em que o robô se encontra exposto, da mesma forma como ocorre com os sinais U_{BA} e U_{DO} gerados por MBA e MDO, respectivamente, tal como apresentado no capítulo 3. Além disso, há uma estreita relação entre os comportamentos dos módulos MIV e MBA, bem como entre os comportamentos dos módulos MDV e MDO.

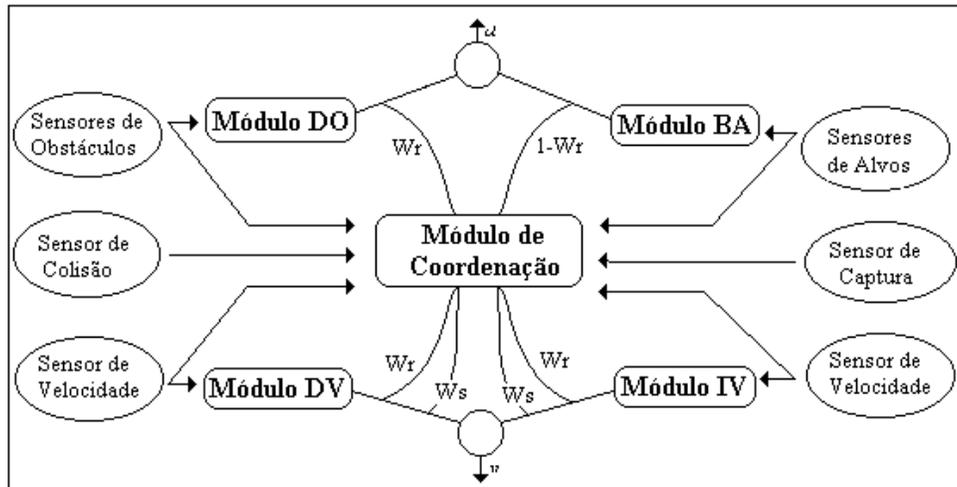


Figura 4.2 – Nova arquitetura do controlador autônomo

Quando o robô percebe uma situação de risco de colisão, os comportamentos de diminuição de velocidade e de desvio de obstáculo devem se manifestar ao mesmo tempo. Da mesma forma, quando o robô se vê distante de obstáculos e do alvo, os comportamentos de aumento de velocidade e manutenção da trajetória na direção do alvo também devem se manifestar ao mesmo tempo. Isto sugere que o MC tem grandes chances de ser utilizado com sucesso também para coordenar as ações dos dois novos módulos inatos.

Para que isto seja possível, a arquitetura do MC também deve ser estendida de forma a considerar o papel da velocidade em cada aprendizagem. Da mesma forma como associa padrões de distâncias a obstáculos com situações de colisão iminente, velocidades muito altas seguidas de uma colisão devem condicionar no robô o comportamento de diminuição da velocidade diante da aproximação de um obstáculo e o aumento da mesma, caso contrário. Uma situação semelhante deve ocorrer quando o robô atinge um alvo, exceto pelo fato de que se espera que o robô aprenda a diminuir sua velocidade sem contudo desviar-se do alvo enquanto se aproxima. A Figura 4.3 apresenta a nova arquitetura do MC, como uma extensão daquela apresentada na Figura 3.3.

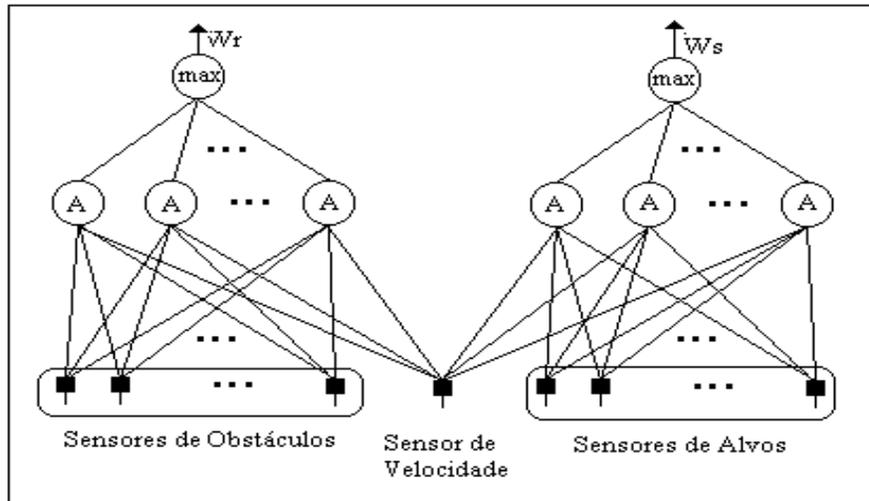


Figura 4.3 – Nova arquitetura do Módulo de Coordenação

A velocidade é incorporada a este novo modelo através de uma sinapse extra em cada um dos neurônios da rede, estimulada pelo sensor de velocidade do robô. Além disso, com a modificação realizada sobre a natureza do sensor de alvos do robô, a porção da rede responsável pela produção do sinal W_s , que antes era constituída por um único neurônio nebuloso com uma única sinapse, passa a ser composta por diversos neurônios nebulosos ligados aos novos sensores de alvos, de forma análoga ao que ocorre com a porção responsável pela produção do sinal W_r .

Esta modificação confere maior simetria e uniformidade aos mecanismos de aprendizagem do MC. Além disso, o aprendizado da diminuição da velocidade do robô quando da aproximação de um alvo é realizado de forma gradual, na medida em que diversos padrões de direção e velocidade são incorporados ao MC através da criação de neurônios conforme os alvos são atingidos.

O Algoritmo 3.1 e o Algoritmo 3.3, que apresentam respectivamente processos de aprendizagem mediante uma colisão e uma captura, dos neurônios da rede neural nebulosa, responsáveis pelo sinal W_r , não sofrem modificações com a incorporação da sinapse ligada ao sensor de velocidade, já que a aprendizagem desta sinapse é realizada essencialmente da mesma forma, salvo a natureza do sinal que registra (i.e. velocidade de deslocamento).

Portanto, para aqueles algoritmos basta a aplicação do mesmo processo à nova sinapse. Por outro lado, os processos de aprendizagem do que era o único neurônio responsável pelo sinal W_s , mediante uma captura ou uma colisão, sofreram alterações consideráveis, devido às modificações apresentadas na Figura 4.3. Neste sentido, o Algoritmo 4.1 apresenta o mecanismo de aprendizagem da rede neural nebulosa que produz W_s . Este algoritmo é análogo ao Algoritmo 3.1, analogia esta que segue da simetria conseguida com a nova arquitetura do MC. Para o Algoritmo 4.1 vale a seguinte nomenclatura:

- Número de sensores de alvos : N_A ;
- Número de capturas : c_A ;
- Número de neurônios na primeira camada : Q_A ;
- Vetor de sinais captados por cada sensor (antena) : $v' = (v'_1 \dots v'_k \dots v'_N)$;
- Peso sináptico de comparação (entrada i e neurônio j) : w'_{ij} ;
- Peso sináptico de ponderação (entrada i e neurônio j) : s'_{ij} ;
- Peso sináptico VAAS (entrada i e neurônio j) : r'_{ij} .

1 - Inicialização: Inicialize o número de capturas e o número de neurônios na primeira camada: $c_A(0) = 0$ e $N_A(0) = 0$.

2 - Ajuste: Se uma captura ocorre, então faça:

- Se $c_A(t) = 0$ então

Adicione o primeiro neurônio na primeira camada da rede neural nebulosa e inicialize seus pesos sinápticos

$$c_A(t+1) = c_A(t) + 1$$

$$Q_A(t+1) = Q_A(t) + 1$$

Para $i = 1$ até N_A e $j = Q_A(t+1)$ faça:

$$s'_{ij}(t+1) = START$$

$$w'_{Aij}(t+1) = v'_{Ai}$$

Fim

(Um neurônio relativo à segunda camada também deve ser considerado para a arquitetura da rede neural nebulosa).

Senão

Para $j = 1$ até $Q_A(t)$ faça: (para cada neurônio da primeira camada)

$$\alpha = c_A(t) - j + 1$$

Para $i = 1$ até N_A faça: (para cada sinapse do neurônio)

$$SIMIL = 0$$

Para $k = 1$ até N_A faça: (para cada entrada do neurônio)

$$AUX = \Delta(\Omega v'_k(x), \Phi w'_{ij}(x))$$

Se $SIMIL < AUX$ então

$$SIMIL = AUX$$

$$COMP = k$$

Fim

Fim

Se $SIMIL > LIMIAR$ então execute os ajustes:

$$w'_{ij}(t+1) = w'_{ij}(t) + (v'_{COMP}(t) - w'_{ij}(t)) / (1 + s'_{ij}(t))$$

$$s'_{ij}(t+1) = s'_{ij}(t) + (SIMIL - s'_{ij}(t)) / (\alpha + 1)$$

Fim

Fim

Fim

$$c_A(t+1) = c_A(t) + 1$$

$$Q_A(t+1) = Q_A(t) + 1$$

Para $i = 1$ até N_A e $j = Q_A(t+1)$ inicialize os pesos sinápticos do novo neurônio:

$$s'_{ij}(t+1) = START$$

$$w'_{ij}(t+1) = v'_i$$

Fim

Fim

3 - Retorne para 2.

Algoritmo 4.1 – Mecanismo aperfeiçoado de aprendizagem por captura

Na aprendizagem após uma colisão, em que se utiliza o VAAS para determinar a responsabilidade dos neurônios que produzem o sinal W_s , e puni-los se for o caso, a qual antes era representada pela expressão (3.6), passa a assumir a forma do Algoritmo 4.2, que por sua vez é análogo ao Algoritmo 3.3 apresentado no capítulo 3.

Para cada neurônio nebuloso de ponderação do sinal de busca de alvos N''_j faça:
 Para cada peso sináptico s'_{ij} faça:

$$s'_{ij}(t+1) = s'_{ij}(t) + (1 - s'_{ij}(t)) \cdot r'_{ij}$$

 Fim
 Fim

Algoritmo 4.2 – Aprendizagem com VAAS aperfeiçoada mediante uma colisão

Além dos mecanismos de aprendizagem, os mecanismos de ativação do MC também sofreram modificações para acomodar a contribuição da velocidade de deslocamento do robô. Com isto, a função de saída f que corresponde ao sinal de saída y do j -ésimo neurônio da rede nebulosa que gera W_r , antes dada pela equação (3.1), passa a assumir a forma de f_1 apresentada na equação (4.1).

$$y_j = f_1 \left(\left(\sum_{i=1}^N A(T_2(\Delta(\Omega_{x_i}(x), \Phi_{w_i}(x)), s_i)) + T_2(\Delta(\Omega_{x_{(N+1)}}(x), \Phi_{w_{(N+1)}}(x)), s_{(N+1)}) \right) / 2 \right) \quad (4.1)$$

A expressão (4.1), representa a média das contribuições dos sensores de obstáculos e de velocidade. A primeira contribuição é fornecida pela função de agregação $A(.)$ e a segunda contribuição é dada pela resposta da sinapse $(N+1)$ ao estímulo apresentado pelo sensor de velocidade do robô. $A(.)$ e $T_2(.)$ são as mesmas da equação (3.1).

De maneira semelhante, a função de saída f_2 que corresponde ao sinal y' do j -ésimo neurônio, passa a substituir a função S_1 apresentada na equação (3.3), tomando a forma expressa pela equação (4.2). Embora esta expressão apresente a mesma interpretação que aquela dada pela equação (4.1), o fato de a natureza dos sensores envolvidos no cálculo de (4.2) ser distinta e ainda de as situações em que os padrões sinápticos são memorizados

pelos neurônios serem diversas, confere ao sinal Ws uma interpretação completamente diferente daquela imputada ao sinal Wr . Por fim, a equação (4.3) expressa o sinal Ws no novo modelo, onde $S_A(\cdot)$ é a t-norma máximo.

$$y'_j = f_2 \left(\left(\bigwedge_{i=1}^{N_A} (T_2(\Delta(\Omega_{x'_i}(x), \Phi_{w'_i}(x)), s'_i)) + T_2(\Delta(\Omega_{x'(N+1)}(x), \Phi_{w'(N+1)}(x)), s_{(N+1)})) \right) / 2 \right) \quad (4.2)$$

$$Ws = \bigwedge_{j=1}^{Q_A} (y'_j) \quad (4.3)$$

As modificações realizadas sobre o controlador nebuloso apresentado no capítulo 3 prevêm, além do neurônio motor de ajuste de direção, responsável pela saída u , dada pela equação (3.4), a incorporação de um segundo neurônio motor, cuja saída v corresponde a um ajuste da velocidade de deslocamento do robô (conforme Figura 4.1) e cuja saída é dada pela equação (4.4), onde $S_m(\cdot)$ é a s-norma máximo.

$$v = U_{IV} (1 - S_m(Wr, Ws)) + U_{DV} (S_m(Wr, Ws)) \quad (4.4)$$

4.4 Situações e comportamentos associados

Quando o robô colide, seu mecanismo de aprendizagem é disparado e um reflexo instantâneo faz com que sua direção de navegação se altere em 180 graus. Tendo colidido diversas vezes, o robô passa a associar determinadas situações a uma condição de risco de colisão iminente. Com isto, ao aproximar-se de um obstáculo, o sinal Wr da rede neural de coordenação passa a assumir valores progressivamente mais altos, conseqüentemente dando maior importância às saídas dos módulos DV e DO. Em termos práticos, isto faz com que o robô reduza sua velocidade e assuma uma nova direção de deslocamento que viabilize o desvio do obstáculo.

De maneira análoga, quando um alvo é atingido outro processo de aprendizagem é disparado. Assumindo-se que o robô já tenha atingido diversos alvos, algumas situações

similares serão tidas como um indício de que a direção de deslocamento esteja correta. Quando o robô se desloca na direção do alvo, o sinal W_s se torna progressivamente mais forte, visto que a distância do alvo com relação ao robô diminui. Quando W_s se torna muito forte, passa a ter uma interpretação negativa para o neurônio motor de velocidade, levando a uma queda da mesma.

Devido ao raciocínio induzido no MC e à natureza do sinal dos sensores de alvos, W_s só se torna forte o suficiente para reduzir a velocidade do robô quando este já se encontra próximo do alvo. Em princípio, esta redução na velocidade é pequena, mas conforme novos alvos vão sendo atingidos e a rede vai sendo composta por novos neurônios, a velocidade tende a diminuir cada vez mais a cada nova aproximação. Por outro lado, quando o robô não detecta a proximidade de um alvo nem a de um obstáculo, há uma tendência natural de que este assuma uma velocidade mais elevada.

4.5 Simulações e Testes

4.5.1 Comparação entre modelos

Os primeiros testes realizados sobre o novo modelo do controlador não apresentavam ruído nos sinais dos sensores. Nestas condições, confirmaram a capacidade do MC em aprender a manipular a velocidade de deslocamento do robô, diminuindo-a em situações de risco de colisão e aumentando-a em situações que não ofereçam tal risco. Neste trabalho, a faixa de variação permitida para a velocidade compreende o intervalo de 0,1 ud./i. a 1 ud./i. Além disso, as taxas de aceleração e desaceleração podem ser diferentes e são determinadas pela equação (4.4). Nas comparações entre modelos com e sem variação de velocidade, ambos possuem VAAS.

A simulação apresentada na Figura 4.4 (a) foi realizada com o sistema de controle descrito no capítulo 3, com velocidade constante e igual a 0,2 ud./i. Neste caso, utilizou-se um robô cujo MC ainda não havia sido treinado em simulações anteriores, por isto, as colisões e os alvos atingidos representam as primeiras fases da aprendizagem. Na Figura 4.4 (a), foram

necessárias 5 colisões antes que o robô atingisse o alvo, a distância percorrida pelo robô foi de 696 ud. e o tempo de simulação foi de 6 minutos e 19 segundos.

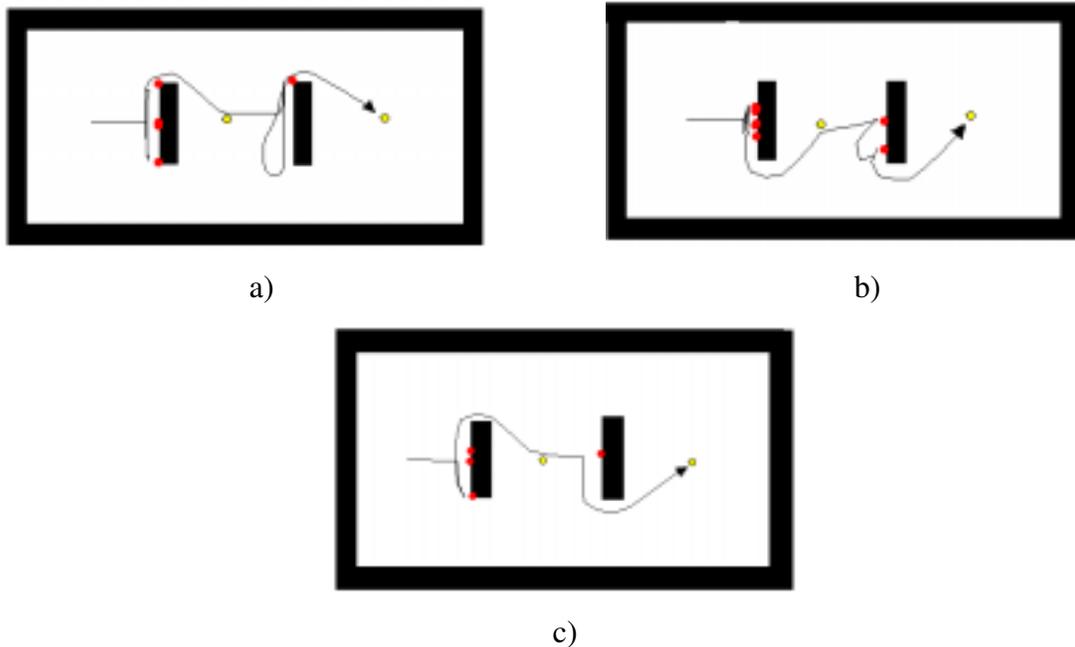


Figura 4.4 – Dois sistemas de navegação distintos: a) velocidade constante = 0,2 ud./i.; b) velocidade variável; c) velocidade constante = 0,7 ud./i.

A Figura 4.4 (b) apresenta uma simulação realizada com o novo modelo de controlador, descrito neste capítulo. Foram utilizados o mesmo ambiente de simulação e os mesmos recursos computacionais da simulação representada pela Figura 4.4 (a). Além disso, as mesmas condições iniciais foram respeitadas: o robô parte do mesmo ponto e com as mesmas direção e velocidade iniciais de deslocamento. Finalmente, também neste caso, o MC não havia passado por situações de treinamento antes do início da simulação.

Agora, foram necessárias 8 colisões antes que o robô atingisse o alvo. A velocidade média desenvolvida pelo robô foi de 0,7 ud./i. A distância total percorrida foi de 552,23 ud. O tempo de simulação foi de 1 minuto e 32 segundos. É interessante notar que foi necessário um número maior de colisões, e portanto de neurônios no MC, para que o robô conseguisse atingir o alvo. Este fato certamente é reflexo da maior complexidade do controle que é estendido também para a velocidade. Entretanto, esta maior exigência em termos de

recursos computacionais e de tempo de aprendizagem é compensada no ganho de desempenho apresentado pelo sistema.

Neste sentido, a diferença mais marcante entre as duas simulações da Figura 4.4 (a) e (b) corresponde ao tempo necessário para que o robô atingisse o alvo. Com velocidade constante, este tempo foi mais de quatro vezes maior que com o modelo de robô com velocidade variável. Esta situação pode ser óbvia, considerando-se que a velocidade média desenvolvida pelo robô com velocidade variável foi quase quatro vezes maior que as 0,2 unidades de deslocamento por iteração conseguida com o modelo de velocidade constante, mas não deixa de refletir uma vantagem do novo sistema, que se aproveita de situações seguras para aumentar sua velocidade. A distância percorrida pelo robô, que é uma medida alternativa ao tempo computacional, também evidencia um ganho de desempenho por parte do novo sistema.

Uma nova simulação com o controlador com velocidade constante é observada na Figura 4.4 (c), agora com velocidade de 0,7 ud./i.. Esta velocidade corresponde à velocidade média desenvolvida pelo robô com velocidade variável na simulação apresentada na Figura 4.4 (b). Também agora foram respeitadas as mesmas condições observadas nas duas simulações anteriores.

O tempo de navegação antes que o robô atingisse o alvo foi de 1 minuto e 22 segundos. Um tempo menor que o apresentado na simulação com velocidade variável, ao passo que foram necessárias apenas 4 colisões para que o robô atingisse o alvo, ou seja, a metade do número de colisões da primeira simulação. Além disso, a distância da trajetória descrita pelo robô nesta última simulação foi de 523 unidades de deslocamento, contra os 552,23 da simulação com velocidade variável.

Em todos os aspectos, então, o robô com velocidade variável foi superado pelo robô com velocidade constante igual a 0,7 ud./i. Por outro lado, considerando-se que 0,7 ud./i. corresponda a uma velocidade próxima da máxima permitida, o robô não respeita a condição de se atingir o alvo com velocidade baixa, além de poder encontrar dificuldades

em realizar manobras seguras em ambientes que requeiram percursos sinuosos, conforme será apresentado mais adiante.

A Figura 4.5 ilustra como o conhecimento extraído do ambiente ao longo das aprendizagens realizadas mediante colisões é representado no novo modelo de neurônio. Os gráficos ilustram neurônios do controlador após a simulação da Figura 4.4 (b). Embora o MC possuísse 8 neurônios após o fim daquela simulação, a figura apresenta apenas os três primeiros, incorporados após a primeira, a segunda e a terceira colisões, respectivamente. Os gráficos possuem o mesmo significado que aqueles apresentados na Figura 3.7, ou seja, os gráficos (a₁) e (a₂), (b₁) e (b₂), e (c₁) e (c₂) correspondem às sinapses dos neurônios 1, 2 e 3 após a primeira, e oitava colisões, respectivamente. De acordo com o algoritmo que incorpora neurônios a cada colisão, os neurônios 1, 2 e 3 sofreram 8, 7 e 6 ajustes sinápticos ao final de 8 colisões, respectivamente.

O ponto interessante a ser notado nestes gráficos é a forma como o robô aprende a manipular sua velocidade. A sinapse que registra a velocidade do robô no momento de uma colisão se encontra na extremidade direita de cada gráfico (38^a sinapse). Na primeira aprendizagem (a₁), a velocidade registrada pelo peso sináptico w_{38} era alta, pois não havia um mecanismo de coordenação desenvolvido. Entretanto, ao longo das próximas aprendizagens, o robô aprende a diminuir sua velocidade ao se aproximar dos obstáculos, conforme as situações (b₁) e (c₁), que apontam uma diminuição gradual do valor registrado por seu peso w_{38} .

Ao final da simulação, o peso sináptico s_{38} que age sobre o valor da velocidade registrada no primeiro neurônio (situação (a₂)), foi razoavelmente bem reforçado, pois nas primeiras duas colisões houve velocidades elevadas. Entretanto, a partir do segundo neurônio, as velocidades baixas passaram a ser verificadas com maior frequência durante as colisões. Por isto, o peso s_{38} , associado à velocidade elevada registrada no segundo neurônio, reforçou pouco o valor do peso w_{38} (situação (b₂)). Por outro lado, o peso s_{38} associado ao neurônio 3 (situação (c₂)) reforçou bastante o peso w_{38} , já que este registrava uma velocidade consideravelmente mais baixa que aquela registrada nos dois primeiros

neurônios. Isto comprova que o robô passa a desenvolver velocidades baixas ao se aproximar de obstáculos.

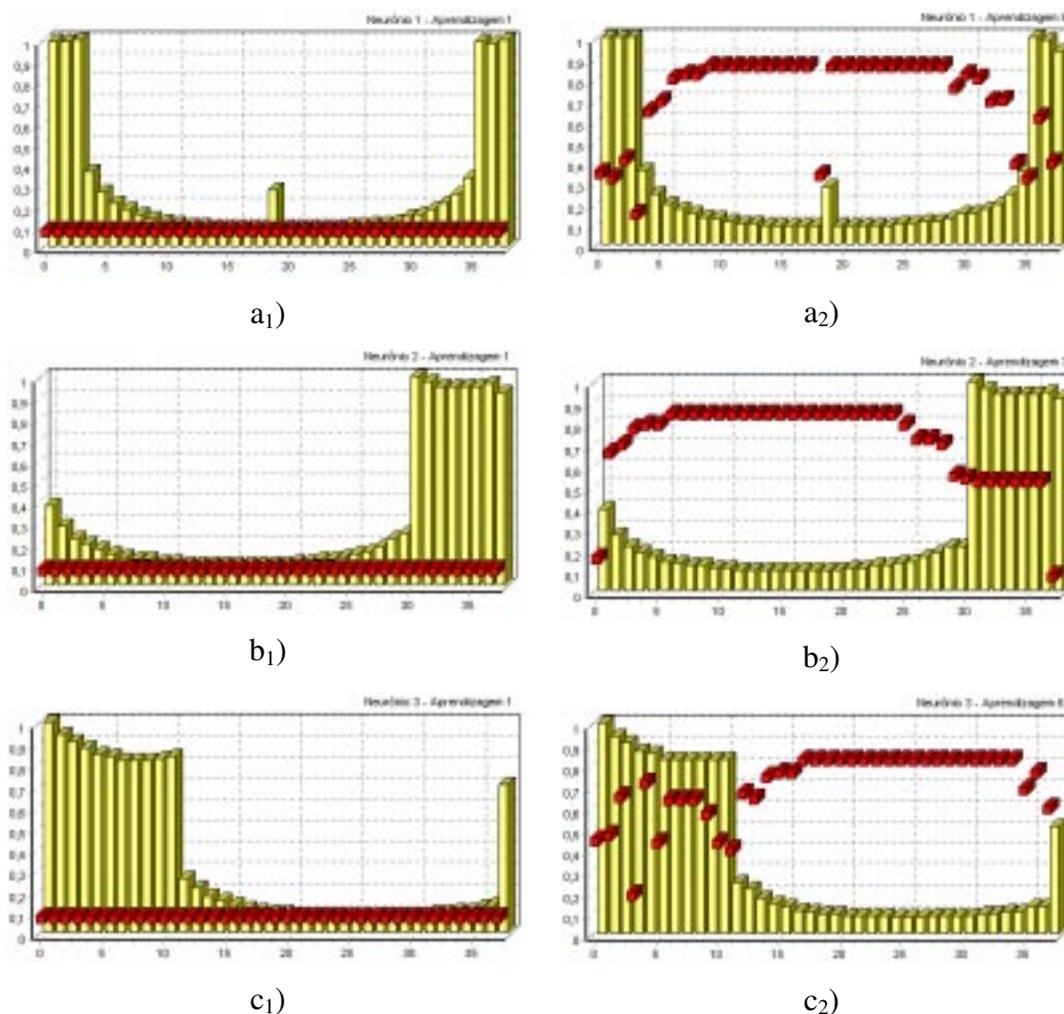


Figura 4.5 – Padrões sinápticos registrados após a 1^a e 8^a colisões para três neurônios

A forma como o robô controla sua velocidade ao longo da navegação é demonstrada na simulação da Figura 4.6. Antes de ser submetido a este ambiente, o robô foi treinado para ser capaz de desenvolver comportamentos de aceleração e desaceleração. Nesta simulação, a trajetória do robô é representada pelos valores de sua velocidade ao longo da navegação. Entre dois valores consecutivos de velocidade apresentados na figura, há um intervalo de 25 iterações do robô com o ambiente em que os valores da velocidade não são apresentados, tanto devido a questões de desempenho computacional quanto por clareza na representação gráfica.

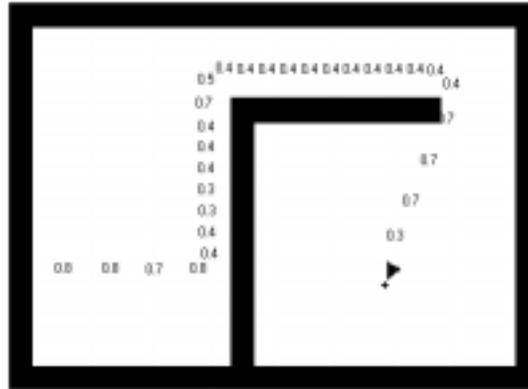


Figura 4.6 – Registro da velocidade desenvolvida pelo robô durante a navegação

Nota-se que, a partir do início da navegação e até encontrar o primeiro obstáculo, o robô desenvolve uma velocidade elevada, que chega a 0,8 ud./i.. Entretanto, quando se aproxima do obstáculo, passa a tangenciá-lo com velocidade abaixo de 0,5 ud./i. Finalmente, quando se livra dos obstáculos, dirige-se para o alvo novamente com velocidade elevada, até que torna a desacelerar quando o alvo se torna muito próximo.

O gráfico da Figura 4.7, que apresenta cinquenta iterações que antecedem uma colisão contra obstáculos do ambiente, ilustra como a queda na velocidade de deslocamento do robô está associada ao reconhecimento de uma situação de risco de colisão por parte do MC. As barras amarelas representam as distâncias do robô, em cada iteração com o ambiente, com relação ao obstáculo mais próximo. Pode-se observar que o robô se aproxima do obstáculo a cada iteração, e à medida que esta distância se torna muito pequena, o sinal W_s (série vermelha) aumenta. Ao mesmo tempo, a velocidade do robô (série verde) cai.

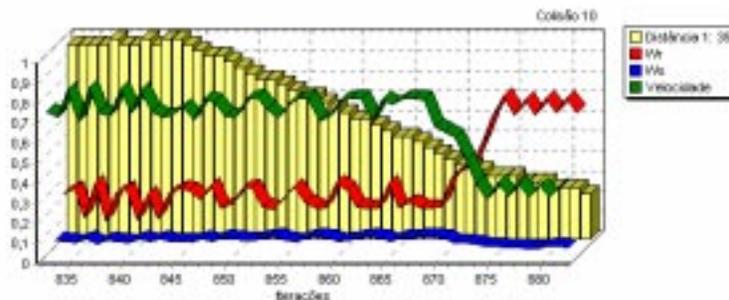


Figura 4.7 – Relação entre risco de colisão e queda da velocidade de deslocamento

Conforme visto na Figura 4.4 (c), há algumas situações em que o controlador com velocidade constante leva vantagem sobre o controlador com velocidade variável, especialmente quando se despreza a necessidade de se aproximar do alvo com velocidade baixa. Por outro lado, na Figura 4.8 pode ser observado que o desempenho do sistema com velocidade variável (simulações (a₁), (b₁) e (c₁)) é muito superior ao do sistema com velocidade constante (simulações (a₂), (b₂) e (c₂)), que nem sequer consegue ultrapassar a porta e atingir o alvo.

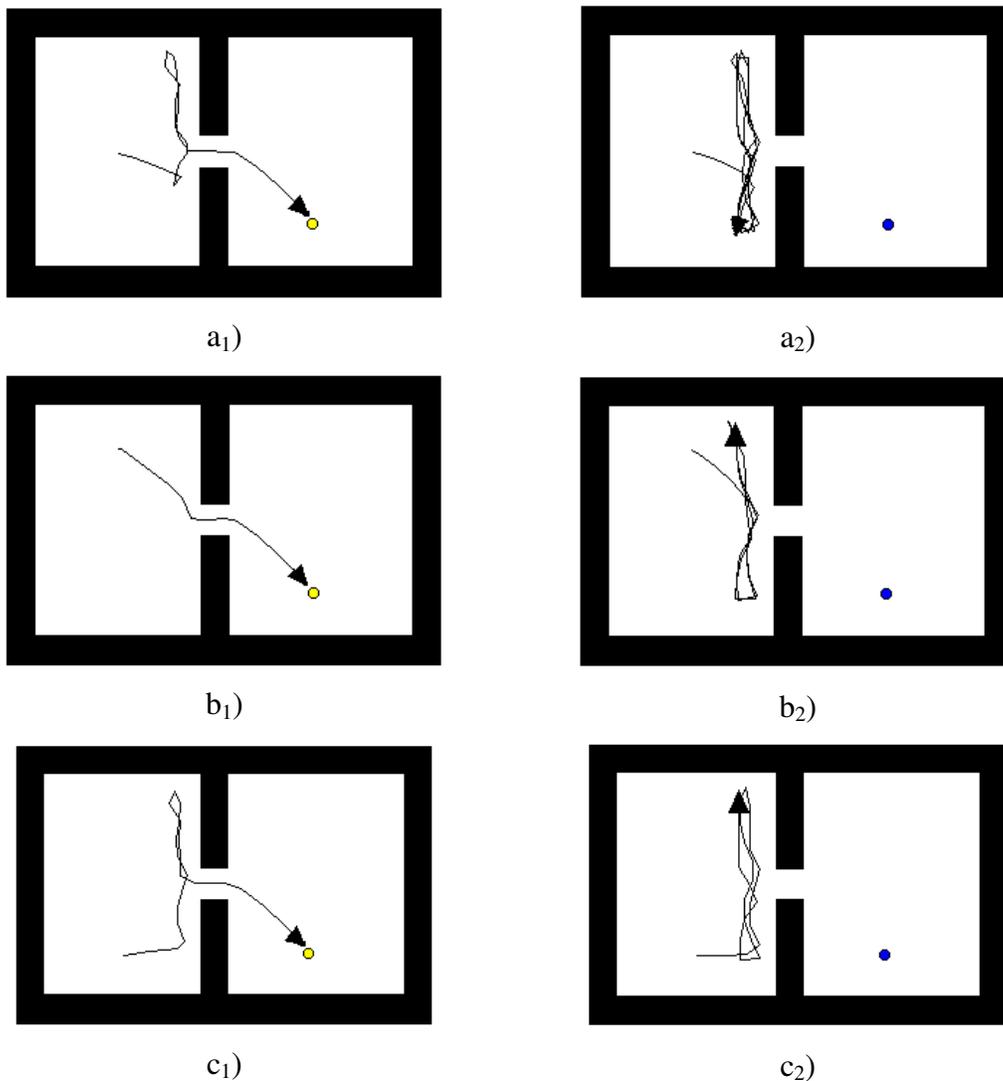


Figura 4.8 - Situações em que o controlador com velocidade variável leva vantagem sobre o controlador com velocidade constante

Como o robô com velocidade variável desenvolveu uma velocidade média de 0,6 ud./i. durante das navegações (a₁), (b₁) e (c₁), esta foi também a velocidade utilizada no robô com velocidade constante. Certamente, o fato de possuir velocidade elevada impede o robô com velocidade constante de ultrapassar o vão. Em todos os casos observados, os robôs já haviam sido treinados e o vão possui 20 ud. de largura, enquanto os robôs apresentam 16 ud. de largura.

Uma outra situação em que a superioridade do mecanismo de controle de velocidade se torna evidente é apresentada na Figura 4.9. Em (a), uma simulação com o robô com velocidade variável e em (b) uma simulação no mesmo ambiente considerando o robô com velocidade constante. Em ambos os casos os robôs passaram por treinamentos prévios em ambientes construídos para tal, de forma que, ao final do treinamento que antecedeu as simulações da Figura 4.9, o MC de cada robô possuísse 15 neurônios inseridos após colisões e 10 neurônios inseridos mediante alvos atingidos.

Neste ambiente, os alvos se encontram encostados a um obstáculo, o que torna a tarefa de atingi-los um desafio, já que os robôs são treinados para se manterem afastados de obstáculos. Em (a), o robô com velocidade variável consegue se aproximar até atingir ambos os alvos, mas o robô com velocidade constante, em (b), não consegue atingir o primeiro alvo e ainda passa para outro compartimento do ambiente, sendo que de lá não consegue mais retornar para onde o alvo se encontra.

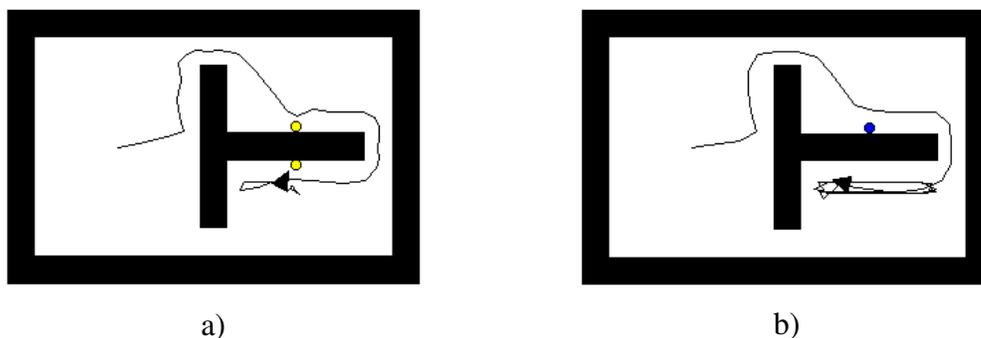


Figura 4.9 – Alvos encostados a obstáculo: a) robô com velocidade variável; b) robô com velocidade constante

O fato de o robô da situação (a) ter conseguido se aproximar dos obstáculos o suficiente para atingir os alvos se deve à forma como o novo mecanismo de aprendizagem age no MC. Quando o robô colide contra um obstáculo o mecanismo de aprendizagem associa não apenas as distâncias a obstáculos, mas também a velocidade do robô a situações de risco de colisão. Logo, a velocidade passa a ser um parâmetro de medida do risco de colisão. Se o obstáculo está próximo e a velocidade está alta, o risco também é alto.

Como o robô diminui a velocidade ao aproximar-se do obstáculo, diminui também o risco de colisão e ele se aproxima ainda mais. Como no robô da situação (b) apenas as distâncias a obstáculo indicam o risco da situação, o robô não se aventura a passeios muito próximos ao obstáculo.

4.5.2 Manipulação de ruído nos sensores

Até o momento, o controlador autônomo apresentado neste capítulo e no capítulo anterior foi sujeito a diversas topologias de ambiente, tendo apresentado bons resultados quanto à navegação autônoma do robô móvel apresentado no capítulo 1. Entretanto, em todos os casos estudados, os sensores de obstáculos, bem como os sensores de alvos não estavam sujeitos a nenhum tipo de ruído. Considerando que em situações reais de navegação as leituras sensoriais apresentam ruído, alguns casos particulares de ruído serão considerados nas próximas simulações com o intuito de verificar a robustez do controlador. Os casos de ruído apresentados não contemplam falhas em determinados sensores, mas apenas variações em seu sinal. Além disso, em todas as simulações apresentadas nesta seção utilizou-se robôs com velocidade variável.

Inicialmente, será examinada a incorporação de ruído apenas nos sensores de obstáculo. Neste caso, assume-se uma probabilidade ρ de ocorrência de ruído em um determinado sensor, ao passo que a ocorrência de ruído em um sensor implica uma variação absoluta de δ ud. na leitura correta. Esta variação possui 50% de probabilidade de ser positiva, e por conseguinte, 50% de probabilidade de ser negativa. Tanto ρ quanto δ são determinados a priori no sistema de simulação (vide Apêndice A).

Na Figura 4.10, três simulações diferentes realizadas sobre o mesmo ambiente ilustram os efeitos do ruído para a aprendizagem e posterior gerenciamento da navegação por parte do controlador. Em todos os casos, o MC do controlador encontrava-se desprovido de neurônios e portanto incapaz de coordenar os movimentos do robô a caminho do alvo. Por isto, em todas as três simulações, uma primeira fase de aprendizagem com a presença de diversas colisões pode ser observada. No entanto, o número de colisões necessárias para a aprendizagem em cada caso foi diferente de acordo com o nível de ruído empregado.

A situação (a), apresenta uma simulação em que nenhum ruído é introduzido nos sensores. Com isto, dez colisões são necessárias para que o MC aprenda a coordenar os movimentos do robô e o resto da navegação ocorre de forma eficiente e livre de novas colisões. Na situação (b), ρ é definido como 10% e δ como 5 ud. Nestas condições, 13 colisões são necessárias para que o MC passe a coordenar corretamente os movimentos do robô e o leve para o alvo. Na situação (c), ρ é definido como 30% e δ é mantido como 5 ud. Agora, o número de colisões necessárias para que o MC coordene coerentemente os movimentos do robô é ainda maior, sendo igual a quinze.

É interessante observar que, na presença de ruído, além de o número de colisões necessárias para que o controlador aja corretamente sobre os atuadores ser maior, a trajetória descrita pelo robô passa a ser mais defensiva. O robô passa a se afastar mais dos obstáculos enquanto se desloca (situações (b) e (c)), ao passo que na ausência de ruído (situação (a)), uma maior aproximação é verificada. Em alguns casos de excesso de ruído, o robô pode mesmo se negar a atravessar passagens estreitas, como as da Figura 4.8, após o treinamento.

A medida adotada para δ (5 ud) é constante ao longo de todo o percurso, não variando conforme a distância medida pelo sensor. Considerando que o ambiente apresentado pela Figura 4.10 possui dimensões de 300 ud. (base) por 200 ud. (altura), a medida de variação adotada é expressiva, especialmente porque, conforme apresentado anteriormente, o controlador considera mais fortemente pequenas distâncias a obstáculos na ponderação do sinal do MDO, uma vez que são estas pequenas distâncias que representam situações de

risco de colisão. Estas pequenas distâncias, obtidas após colisões geralmente são da ordem de 5 ud.

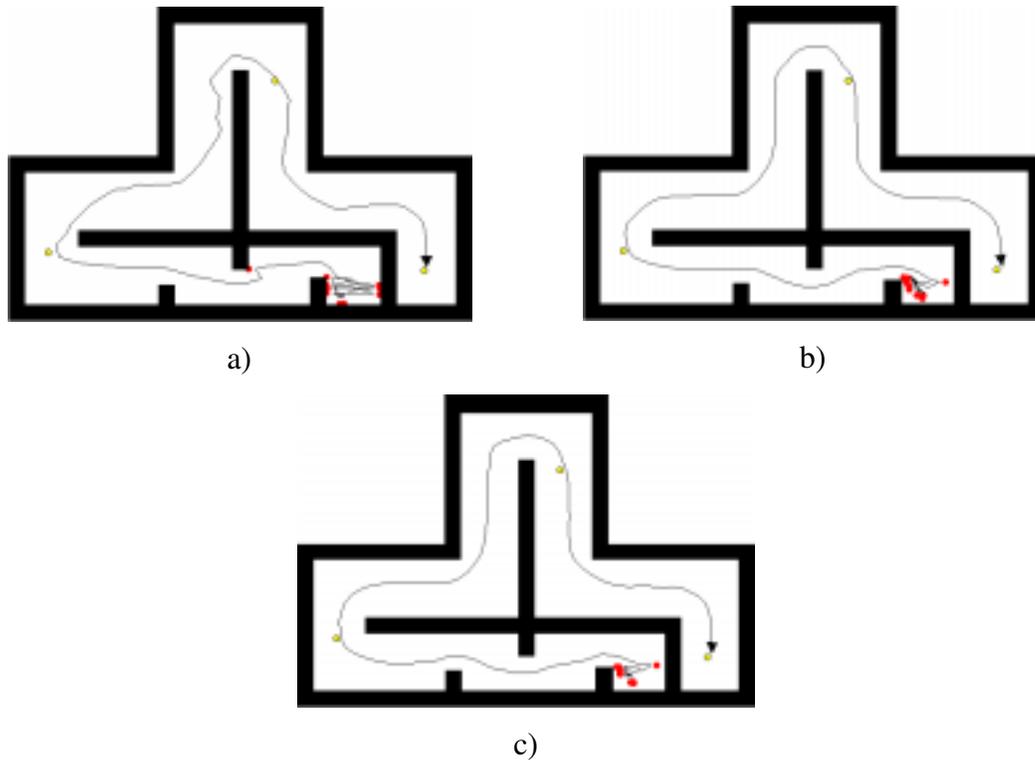


Figura 4.10 – Três simulações considerando diferentes parâmetros de ruído nos sensores de obstáculos: a) $\rho = 0\%$, $\delta = 0$ ud.; b) $\rho = 10\%$, $\delta = 5$ ud.; c) $\rho = 30\%$, $\delta = 5$ ud

A Figura 4.11 ilustra como o ruído é agregado aos pesos sinápticos de um neurônio do MC. Em todos os casos, por se tratar da aprendizagem que deu origem ao neurônio, todos os termos do vetor de pesos \vec{s} são iguais a START (vide Algoritmo 3.3). Em (a), um neurônio representando uma colisão em que os dados dos sensores são obtidos sem ruído é apresentado. Em (b), observa-se a representação de uma colisão contra o mesmo ponto, com as leituras dos sensores sujeitas a 10% de ruído. Finalmente em (c) a mesma situação com os sensores sujeitos a 30% de ruído. Em todos os casos δ é igual a 5 ud.

Pode-se observar nas três situações apresentadas pela Figura 4.11 que em todas elas os vetores de pesos sinápticos \vec{w} preservam sua similaridade apesar das variações provocadas pelo ruído. Estas variações também estão presentes quando o robô estiver diante de

situações parecidas às apresentadas pela figura. Neste caso, a rede neural nebulosa que compõe o MC foi projetada para suportar tais variações, que são solucionadas pela aplicação do operador nebuloso de medida de possibilidade à comparação dos vetores de pesos sinápticos \vec{w} .

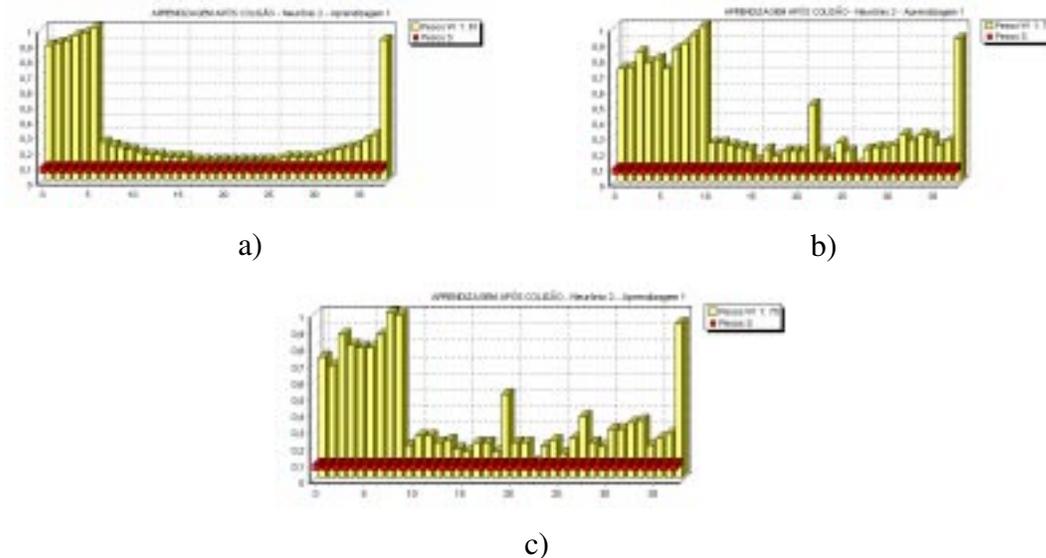


Figura 4.11 – Colisão lateral: a) $\rho = 0\%$; $\delta = 0$ ud.; b) $\rho = 10\%$, $\delta = 5$ ud.;
 c) $\rho = 30\%$, $\delta = 5$ ud.

Uma vez tendo sido estudados os efeitos do ruído sobre os sensores de obstáculo, a Figura 4.12 apresenta situações em que se considera o efeito do ruído apenas sobre os sensores de alvos. Agora, a medida δ' representa uma variação na intensidade do sinal recebido do alvo. Este sinal é normalizado e se restringe ao intervalo $[0, 1]$. Nas simulações apresentadas pela Figura 4.12 utilizou-se δ' igual a 0,05, exceto para o caso (d). Em todos os casos considera-se controladores que ainda não tenham passado por processos de aprendizagem.

No caso (a), nenhum ruído é incorporado aos sensores e o robô percorre seu caminho normalmente em direção aos alvos. No caso (b), utiliza-se ρ igual a 10% e o robô realiza uma trajetória com alguma sinuosidade enquanto atinge os alvos. No caso (c), utiliza-se ρ

igual a 15% e com isto, a trajetória descrita pelo robô é muito sinuosa. Finalmente, no caso (d) utilizou-se ρ igual a 10% e δ' igual a 0,04.

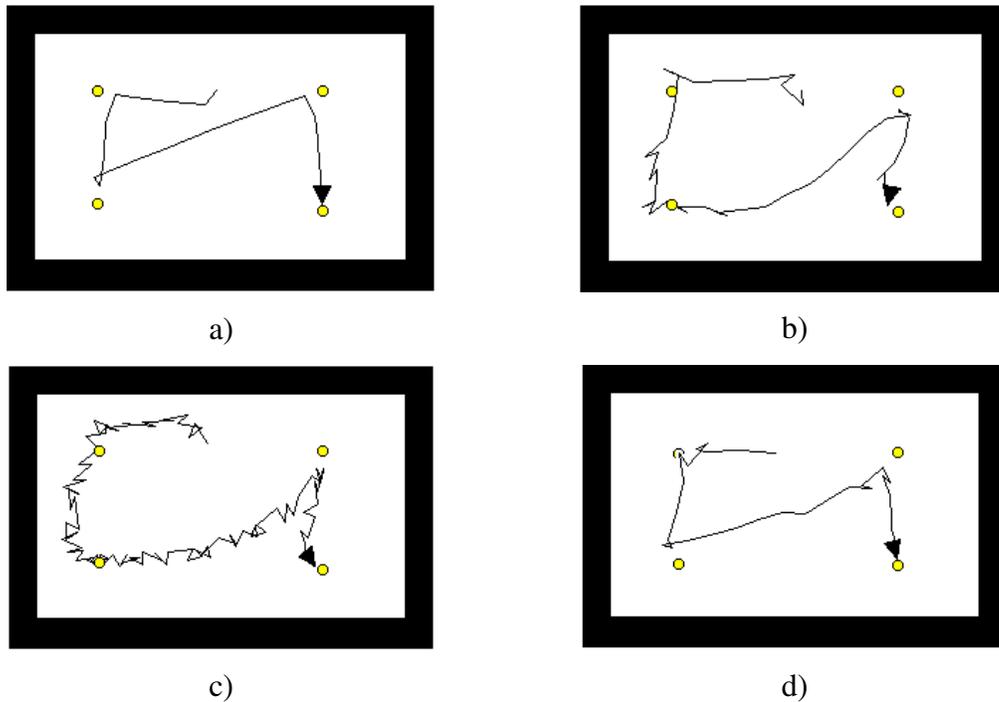


Figura 4.12 – Quatro situações considerando diferentes parâmetros de ruído para os sensores de alvos: a) $\rho = 0$, $\delta' = 0$; b) $\rho = 10\%$, $\delta' = 0,05$; c) $\rho = 15\%$, $\delta' = 0,05$; d) $\rho = 10\%$, $\delta' = 0,04$

Observou-se ao longo dos experimentos que pequenas variações em δ' são responsáveis por mudanças significativas no comportamento de coordenação do controlador. Além disso, o MC tolera menos a ocorrência de ruído nos sensores de alvos que nos sensores de obstáculos. Neste sentido, probabilidades acima de 15% de ruído com $\delta' = 0,05$ inviabilizam a busca pelo alvo, de forma que o robô passa a desenvolver uma trajetória que não o leva em direção ao alvo.

A Figura 4.13 apresenta duas representações, nos pesos sinápticos de dois neurônios do MC, de aprendizagens após alvos atingidos. Em (a) não há a ocorrência de ruído e em (b) há a ocorrência de ruído com ρ igual a 15% e δ' igual a 0,05. Cada barra amarela nos gráficos representa um sensor (antena). A barra mais alta representa o sensor que captura

com mais intensidade o sinal do alvo. Portanto, o alvo provavelmente se encontra na direção deste sensor. Conforme o ângulo de desvio do sensor com relação ao alvo aumenta, a intensidade do sinal diminui. Para sensores com desvios superiores a 90° o sinal do alvo é nulo.

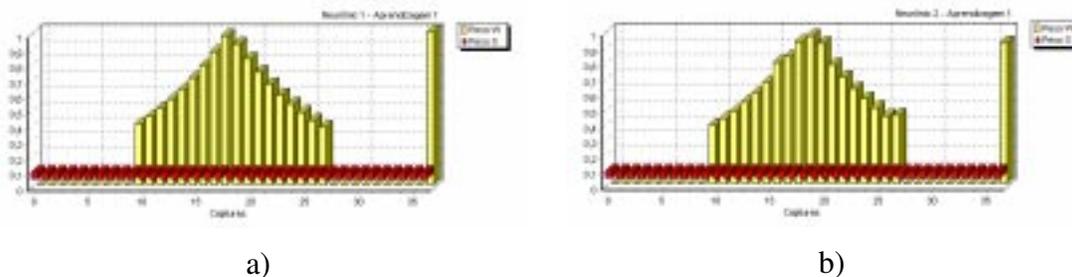


Figura 4.13 – Alvo atingido frontalmente: a) $\rho = 0\%$, $\delta' = 0,05$; b) $\rho = 15\%$, $\delta' = 0,05$

Como pode ser observado, as diferenças entre as duas representações (a) e (b) são muito pequenas. Não obstante, da forma como o MBA interpreta os sensores de alvos, estas pequenas diferenças causam alterações consideráveis de comportamento, como pode ser observado na Figura 4.12 entre as situações (a) e (c).

Assim como observado nas representações neurais apresentadas na Figura 4.13, na grande maioria das vezes em que um alvo é atingido pelo robô, este o faz frontalmente, uma vez que se dirige diretamente ao alvo. Com isto, o padrão do sinal do alvo em quase todas as capturas será aproximadamente o mesmo. Assim, em princípio parece um contra-senso que um novo neurônio seja incorporado ao MC após cada captura de alvo, o que caracterizaria um excesso desnecessário de redundância.

Entretanto, com a incorporação do mecanismo de controle de velocidade, o padrão registrado de busca de alvos passou a ser composto por dois componentes: o sinal do alvo e a velocidade do robô. Assim como ocorre com os obstáculos, conforme a Figura 4.5, também no caso dos alvos o robô deve aprender gradualmente a controlar sua velocidade quando da aproximação de um alvo, e isto ocorre através do registro de diversos padrões compostos do sinal do alvo e do sinal da velocidade. Uma alternativa para reduzir o excesso

de neurônios da rede é, uma vez atingido o nível esperado de controle de velocidade, eliminar aqueles neurônios que desempenharam o papel intermediário de formação deste refinamento do controle.

Para terminar o estudo de casos de simulação com ruído, a Figura 4.14 apresenta duas situações, sendo que em (a) o robô navega em um ambiente livre de interferências, ao passo que em (b), incorporou-se ruído tanto nos sensores de alvos quanto nos sensores de obstáculos. Nos sensores de obstáculos ρ é igual a 30% δ é igual a 5 ud. e nos sensores de alvos ρ é igual a 10% e δ' é igual a 0,04.

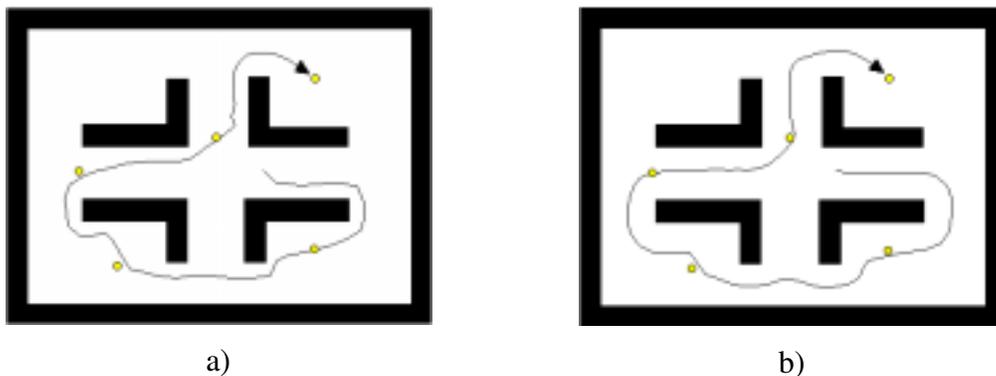


Figura 4.14 – Estudo dos efeitos do ruído nos sensores de obstáculos e nos sensores de alvos: a) sem ruído; b) $\rho = 10\%$ para sensores de alvos, $\rho = 30\%$ para sensores de obstáculos, $\delta = 5$ ud., $\delta' = 0,04$

Em primeiro lugar, observa-se que na simulação com ruído (b), de forma similar às simulações com ruído nos sensores de obstáculos apresentadas na Figura 4.11, o robô desenvolve uma trajetória mais afastada dos obstáculos que na simulação sem ruído (a). Outro aspecto interessante a ser notado é que a pressão exercida pelos obstáculos sobre o robô e a boa tolerância dos sensores de obstáculos a ruído parece ter contribuído para diminuir os efeitos do ruído sobre os sensores de alvos, fazendo com que o robô não descreva uma trajetória sinuosa.

4.6 Visão sintética do capítulo 4

Neste capítulo, a arquitetura de controlador autônomo para navegação de robôs móveis apresentada no capítulo 3 foi estendida de forma a apresentar um novo comportamento. Além da direção de deslocamento, o ajuste da velocidade do robô também passa a ser tarefa do controlador autônomo, no modelo apresentado.

Dentre as principais vantagens deste novo modelo, destacam-se: maior flexibilidade nas manobras, o que possibilita economia de tempo em ambientes abertos e controle mais refinado na aproximação de obstáculos, redução de velocidade na aproximação de alvos e, por fim, a possibilidade de realização de simulações com múltiplos robôs em um mesmo ambiente, o que abre espaço para novos estudos relacionados a comportamentos coletivos dos robôs. Esta última característica será explorada em maiores detalhes no capítulo 7.

O novo modelo de controlador pressupõe, no entanto, maior disponibilidade de recursos computacionais, tais como memória e processamento, quando comparado a modelos de controle de navegação mais simples (EPSTEIN, 1998; FARIA & ROMERO, 2000; FREIRE *et al.*, 1998). Além disso, apesar de a velocidade nos experimentos apresentados neste capítulo ter sido normalizada no intervalo $[0, 1]$, observou-se em experimentos não apresentados aqui que quanto maior a velocidade máxima permitida, maior também é a dificuldade de controle sobre os movimentos do robô, e maior será o período de aprendizagem do controlador.

Por fim, como pode ser observado no modelo da Figura 4.3, apesar de a arquitetura do MC apresentar duas redes neurais distintas destinadas a ponderar a importância dos sinais do MBA e do MDO a cada passo da navegação, o sensor de velocidade age como um elo comum entre estas duas redes neurais. Com isto, em certas situações a velocidade se torna um parâmetro ambíguo para o controlador.

Sabe-se que o robô deve aprender a diminuir sua velocidade ao se aproximar de um obstáculo e que velocidades baixas acabam sendo reconhecidas como indícios da proximidade de obstáculos, e portanto de situações de risco de colisão. Por outro lado,

velocidades baixas também podem ser associadas à aproximação de um alvo. Esta ambigüidade pode levar o MC a tomar decisões contraditórias em algumas situações, ao passo que o que se espera do MC é justamente eliminar as contradições do MDO e do MBA.

Para solucionar este problema, alguns ajustes ainda devem ser considerados na arquitetura do MC, de forma que esta simetria introduzida pelo sensor de velocidade seja eliminada da aprendizagem durante a navegação.

Capítulo 5

Estudo de Comportamentos Inatos

5.1 Introdução

Um dos pontos mais críticos do problema da navegação autônoma de robôs consiste no tratamento de ambientes e situações previamente desconhecidos. Neste caso, qualquer esforço na construção de um modelo de representação de conhecimento deve ser cuidadosamente considerado, na medida em que impõe ao sistema restrições nem sempre válidas ou suficientes ao problema. Devido a este fato, modelos de inferência simbólica, propostos pela Inteligência Artificial Clássica, costumam apresentar deficiências na construção de veículos auto-guiados (BROOKS, 1987).

Assim, ao invés de se tentar antecipar o conhecimento do sistema frente às situações a que este possa ser submetido, uma alternativa interessante é capacitá-lo a construir suas próprias estratégias de ação sobre o ambiente em que atuará. Neste sentido, o estudo apresentado a seguir demonstra ser possível dotar um robô da habilidade de sintetizar comportamentos básicos para a navegação, quais sejam: desvio de obstáculos e busca de alvos.

O desenvolvimento destes comportamentos deve se dar através da aquisição de conhecimento a partir do próprio ambiente, eliminando assim qualquer forma de conhecimento arbitrário pré-definido, juntamente com seus malefícios, dentre os quais se destacam: dificuldade na determinação de uma base de conhecimento completa, no sentido de satisfazer a todas as necessidades do sistema em todas as situações possíveis, e grau de aplicação do conhecimento, considerando cada uma destas situações de navegação possíveis. Por isto, no caso particular de um ambiente desconhecido, foco de interesse deste trabalho, a chance de um mecanismo de controle arbitrário obter sucesso na tarefa de navegação é reduzida.

Em particular, no que se refere à geração automática de conhecimento a partir de interações com o ambiente, técnicas de evolução genética têm sido aplicadas de forma crescente à construção de mecanismos de controle para robôs móveis (URZELAI & FLOREANO; 2001). Alguns modelos utilizam a codificação dos cromossomos como representação direta dos comportamentos a serem evoluídos (FLOREANO & MONDADA, 1998; LEE, 1999; SANTOS *et al.*, 2001). Entretanto, em abordagens híbridas que se aproveitam, por exemplo, do poder de representação dos sistemas nebulosos (IWAKOSHI *et al.*, 1998) e da flexibilidade das redes neurais artificiais (KUBOTA *et al.*, 2001) a codificação dos cromossomos freqüentemente corresponde a parâmetros indiretos de controle (e.g. pesos sinápticos de uma rede neural e termos de uma regra nebulosa).

Quanto ao potencial representado pela computação evolutiva no tratamento de problemas complexos, como a navegação autônoma de robôs móveis, HUSBANDS *et al.* (1997) apontam para o surgimento de uma nova área de pesquisa sobre robôs móveis, a qual, segundo os autores, vem sendo referida como robótica evolutiva. Em seu artigo, uma rede neural com três camadas é treinada geneticamente para controlar um robô móvel utilizando um algoritmo evolutivo, sendo que cada cromossomo codifica os parâmetros de toda a rede. As vantagens de se evoluir uma rede neural, segundo os autores, procede, dentre outras coisas, do fato de seu espaço de estados ser mais suave do que o de programas de controle explícitos e do fato de os parâmetros do treinamento serem de baixo nível, o que evita a incorporação de conhecimento sofisticado por parte do projetista.

Outra questão levantada por HUSBANDS *et al.* (1997) se refere à utilização de implementações práticas em detrimento de simulações computacionais para robôs móveis. Embora mencionem o fato de que comportamentos adquiridos durante uma evolução simulada, em que o ruído é cuidadosamente considerado, se aproximem daqueles adquiridos em implementações reais, os autores preferem o uso de implementações práticas. Não obstante, o ambiente de implementação é cuidadosamente construído para o robô, no que se refere à iluminação e à coloração dos obstáculos, uma vez que o mecanismo de sensoriamento do robô é implementado por meio de uma câmara digital.

Uma outra aplicação interessante de técnicas evolutivas à navegação de robôs móveis é apresentada por KUBOTA *et al.* (2001). Neste trabalho, um controlador nebuloso é evoluído, considerando como saídas o ajuste na direção e na velocidade do robô de forma semelhante à apresentada no capítulo 3. Além disto, cada cromossomo representa uma única regra de inferência e, por conseguinte, uma parte do controlador.

Com base nestas considerações sobre mecanismos evolutivos empregados a controladores de navegação de robôs, este capítulo concentra-se na síntese, através de evolução genética, de estruturas inatas capazes de gerar comportamentos de desvio de obstáculos e de busca de alvos, representadas respectivamente pelo Módulo de Desvio de Obstáculos (MDO) e pelo Módulo de Busca de Alvos (MBA), responsáveis pelo ajuste na direção de deslocamento do robô e apresentados no capítulo 3. O Módulo de Incremento de Velocidade e o Módulo de Decremento de Velocidade não serão considerados devido à simplicidade de suas ações, que consistem na emissão, a cada passo do robô, de sinais de incremento e de decremento da velocidade de deslocamento aos neurônios motores do controlador, independentemente de parâmetros externos ou de parâmetros do próprio controlador.

5.2 Evolução de comportamentos de desvio de obstáculos e busca de alvos

A abordagem utilizada para a síntese de conhecimento inato no MDO e no MBA consiste na aplicação de técnicas de evolução genética aos módulos, de forma que sua ação sobre o robô se torne coesa com relação à tarefa de navegação autônoma. Porém, antes de apresentar a abordagem evolutiva utilizada, convém introduzir uma breve descrição do MDO e do MBA, apresentados no capítulo 3. A Figura 5.1 apresenta a arquitetura geral destes módulos.

A arquitetura apresentada consiste em uma representação neural para um conjunto de h regras de inferência nebulosa. Cada uma destas regras possui a forma $A_i \rightarrow C_i$, $i \in \aleph$, $A_i \in \mathfrak{R}$, $C_i \in \mathfrak{R}$, em que A_i é o antecedente e C_i é o conseqüente da i -ésima regra nebulosa. Além disso, tanto o antecedente quanto o conseqüente codificam medidas angulares em radianos. No caso do antecedente, esta medida pode ser a direção do alvo com relação ao robô (MBA) ou a direção do obstáculo mais próximo do robô com

relação ao mesmo (MDO). No caso do conseqüente, a medida é o ajuste na direção de deslocamento do robô, tanto para o MBA quanto para o MDO.

Ainda de acordo com o modelo da Figura 5.1, o vetor de entradas \vec{x} corresponde aos sinais dos sensores de obstáculos no caso do MDO e dos sensores de alvos no caso do MBA. Este vetor sofre um pré-processamento resultando no valor P , que codifica uma medida angular em radianos, cujo significado é idêntico àquele apresentado anteriormente para o antecedente de uma regra nebulosa. O valor de P serve então como entrada para a rede neural nebulosa composta pelo conjunto das h regras nebulosas.

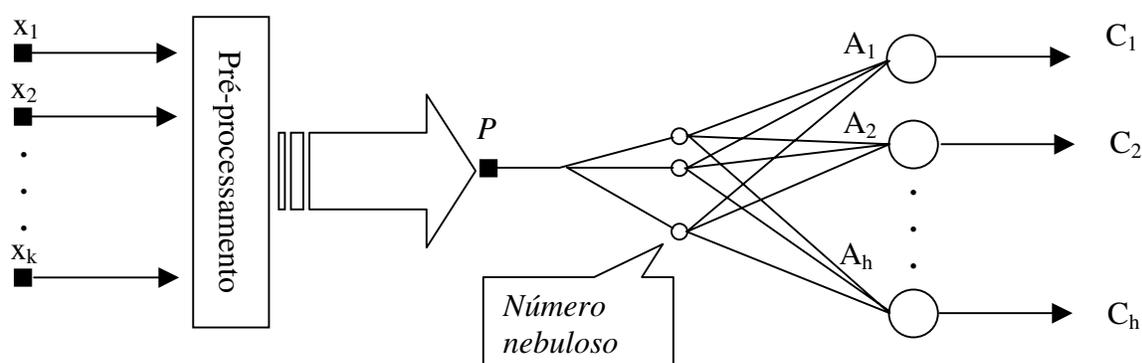


Figura 5.1 – Arquitetura dos módulos inatos MBA e MDO

O pré-processamento do vetor de entradas \vec{x} , conforme apresentado pela Figura 5.1, implica assumir para P o valor do ângulo do sensor que detectou o alvo, no caso do MBA, ou o ângulo do sensor que detectou o obstáculo mais próximo do robô, no caso do MDO. A Figura 3.1 identifica este ângulo como α no caso do MDO e como β , no caso do MBA. Caso o pré-processamento não fosse utilizado na evolução, cada regra possuiria não apenas um, mas k antecedentes. Com isto, considerando uma rede com n regras, o número de genes em cada cromossomo, supondo-se regras com um único antecedente, é $2n$, ao passo que supondo-se regras com k antecedentes, com $k > 1$, o número de genes em cada cromossomo sobe para $(k+1)n$. Quanto maior o número de genes de um cromossomo, tanto maior será o tempo necessário para que a evolução leve ao resultado esperado.

Após ser transformado em um número nebuloso, P é comparado via o operador de medida de possibilidade, conforme a equação (2.16), a cada um dos elementos do vetor de antecedentes \bar{A} , também transformados em números nebulosos, sendo que o antecedente A_j mais similar a P irá levar ao disparo do j -ésimo neurônio. A transformação de P e de cada um dos antecedentes A_j em números nebulosos é feita através de sua aplicação a uma função de pertinência triangular, como a apresentada na Figura 2.11(b). Assim, o valor modal da função é dado pelo valor de P ou por um dos antecedentes A_j , e a dispersão da função, é $\pi/4$. No caso do MBA, o valor da dispersão se justifica pelo fato de que cada sensor (antena) varre um espaço de $\pi/2$ rad. No caso do MDO, o valor $\pi/4$ é adotado empiricamente por questão de compatibilidade com o MBA.

A função de ativação $q(A_j, P)$ do j -ésimo neurônio da camada de saída é dada pela equação (5.1), em que $Poss(A_j, P)$ é o operador de medida de possibilidade, conforme a equação (2.16) e h é o número de neurônios na camada de saída. A função de ativação gera, portanto, o conseqüente C_j da j -ésima regra nebulosa.

$$q(A_j, P) = \begin{cases} C_j, & \text{caso } Poss(A_j, P) > Poss(A_i, P), \forall i \in \mathfrak{K}, i \leq h, i \neq j \\ 0 & \text{caso contrário} \end{cases} \quad (5.1)$$

Em termos gerais, o treinamento evolutivo visa simular a adaptação adquirida pelos módulos inatos ao longo de gerações de robôs dedicados à tarefa de desvio de obstáculos e de busca de alvos. Isto implica a aplicação de um algoritmo genético sobre uma população de indivíduos em que cada um represente um conjunto de h regras de inferência nebulosa. Assim, apesar de o resultado da evolução ser um conjunto de regras de inferência agrupadas em uma estrutura neural nebulosa, estas regras não são arbitradas por um especialista.

A população inicial de regras é gerada aleatoriamente, respeitadas as seguintes restrições: O universo de discurso de um antecedente A_j se limita ao intervalo $[-\pi, \pi]$; o universo de discurso de um conseqüente C_j se limita ao intervalo $[-\pi/18, \pi/18]$. O intervalo de valores assumidos por um conseqüente é mais restrito que no caso de um antecedente, pelo fato de

um conseqüente representar variações na direção de deslocamento do robô, variações estas sujeitas a limitações físicas em um ambiente real.

Para avaliar o desempenho de cada indivíduo que compõe a população de regras nebulosas, realiza-se uma série de simulações de navegação, ao longo das quais diversos critérios de desempenho são analisados. Estes critérios variam de acordo com o módulo que esteja sendo evoluído. No caso do MDO, importa que o robô não colida contra os obstáculos do ambiente enquanto navega, ao passo que no caso do MBA, importa que todos os alvos apresentados no ambiente sejam atingidos e que o robô tente descrever a trajetória mais curta possível.

A equação (5.2) é utilizada na avaliação de um indivíduo do MDO em evolução, após o final de uma simulação. Conforme a equação, Γ é uma constante arbitrária utilizada apenas para garantir que a avaliação seja sempre maior que zero. Nos resultados apresentados neste trabalho utilizou-se empiricamente $\Gamma = 1000$. ∂_m é a distância média com que o robô permanece em relação ao obstáculo mais próximo a ele durante a simulação, e c é o número de colisões ocorridas durante a simulação. As constantes α_1 e α_2 podem ser utilizadas para ponderar a importância dos termos aos quais se aplicam, mas nos resultados apresentados na seção 5.3, ambas têm valor igual a 1.

$$A_{MDO} = \Gamma + \partial_m \cdot \alpha_1 - c \cdot \alpha_2 \quad (5.2)$$

A equação (5.3) é utilizada na avaliação de um indivíduo do MBA em evolução, após o final de uma simulação. Neste caso, ϕ_m é o desvio médio da direção de deslocamento do robô em relação à direção do alvo durante a simulação, a é o número de alvos atingidos e Δs é a diferença entre o número máximo de passos permitidos e o número de passos realizados pelo robô durante uma simulação. Nos resultados apresentados na seção 5.3, supõe-se empiricamente que o número máximo de passos permitidos em uma simulação com o MBA sendo evoluído é 750. Caso o robô consiga atingir todos os alvos antes que este número máximo de passos seja alcançado, a simulação é interrompida e Δs assume valor maior que zero, caso contrário, a simulação é interrompida quando o número máximo

de passos for atingido e Δs assume valor zero. As constantes β_1 e β_2 ponderam a importância dos termos aos quais são aplicadas na equação, e nos resultados apresentados em 5.3, são respectivamente 100 e 5, valores estes obtidos empiricamente.

$$A_{MBA} = (\pi - \phi_m) \cdot \beta_1 + a \cdot \beta_2 + \Delta_s \quad (5.3)$$

Para a evolução de ambos os módulos inatos discutidos neste capítulo, utilizou-se cromossomos que codificam 5 regras nebulosas, resultando num total de 10 genes em cada cromossomo. Esta representação compacta da informação necessária para a construção dos módulos inatos, através de um conjunto reduzido de regras, foi possível graças ao pré-processamento realizado sobre os sinais dos sensores do robô, de acordo com o mostrado na Figura 5.1, que produz uma informação de alto nível sobre o panorama do ambiente, a qual irá compor o antecedente das regras, e pode ser generalizada para uma série de situações de navegação diferentes.

No entanto, apesar do espaço de busca do algoritmo genético ser pequeno, devido ao fato de ser necessário submeter o robô a uma navegação no ambiente de simulação antes que cada cromossomo possa ser avaliado, o tempo para a evolução completa dos módulos inatos é um fator crítico. Para resultados obtidos em simulação, utilizou-se populações compostas por 50 cromossomos, sendo que o número de gerações foi limitado a 50. Como será apresentado, estes valores são suficientes para que o robô desenvolva habilidades de desvio de obstáculo e busca de alvos. Em uma máquina com processador de 166MHz foram necessários pouco mais de cinco minutos para a evolução do MBA e aproximadamente quatro horas e meia para a evolução do MDO. Esta diferença expressiva de tempo de evolução entre os dois módulos se deve ao fato da rotina de sensoriamento de obstáculos ser mais complexa que a rotina de sensoriamento de alvos.

Em particular, para evoluir o MBA, utilizou-se um ambiente sem obstáculos, contendo cinco alvos, de acordo com a Figura 5.2. A cada geração a posição inicial dos robôs era fixada, o mesmo valendo para as posições dos alvos. No entanto, alguns testes foram realizados variando-se aleatoriamente a posição inicial dos robôs e as posições dos alvos

para gerações diferentes. Alguns dos principais resultados obtidos nestes testes são apresentados na seção 5.3.

Para evoluir o MDO, utilizou-se um ambiente fechado contendo um obstáculo central, mas sem alvos. Este ambiente é apresentado na Figura 5.7. Da mesma forma que no caso da evolução do MBA, agora também as posições iniciais dos robôs dirigidos por diferentes cromossomos é mantida fixa ao longo de uma geração, sendo sujeita a variações aleatórias para gerações diferentes.

Em ambos os processos evolutivos, utilizou-se seleção elitista com probabilidade de 10%, o que significa que os 10% melhores cromossomos de uma geração são automaticamente selecionados para a próxima geração. Além disso, utilizou-se recombinação simples (crossover de um ponto) com 20% de probabilidade e mutação com 5% de probabilidade para a manutenção da diversidade populacional. A escolha dos cromossomos a serem recombinados é feita por meio do método *roullet wheel*, um mecanismo de sorteio em que indivíduos mais bem avaliados possuem maior chance de serem escolhidos (MICHALEWICZ, 1996). A mutação utilizada é do tipo indutiva, dado que os genes assumem valores reais, o que implica que pequenas variações são aplicadas ao gene que sofre mutação, sendo da ordem de $0,5 \pi$ rad para antecedentes e 0,02 rad para conseqüentes.

5.3 Resultados para o MBA

Os primeiros testes com o mecanismo descrito na seção 5.2 foram feitos sobre o MBA. A Figura 5.2 apresenta em (a) o ambiente utilizado para a realização da evolução e em (b) o gráfico que ilustra a curva evolutiva da avaliação do melhor indivíduo da população a cada geração do algoritmo genético, representada em vermelho, e a curva evolutiva da avaliação média da população, representada em verde. Como pode ser observado, o ambiente utilizado para a evolução do comportamento de busca de alvos não possui quaisquer obstáculos. A velocidade do robô em todas as simulações apresentadas nesta seção possuem velocidade constante e igual a 0,5 unidade de deslocamento por iteração.

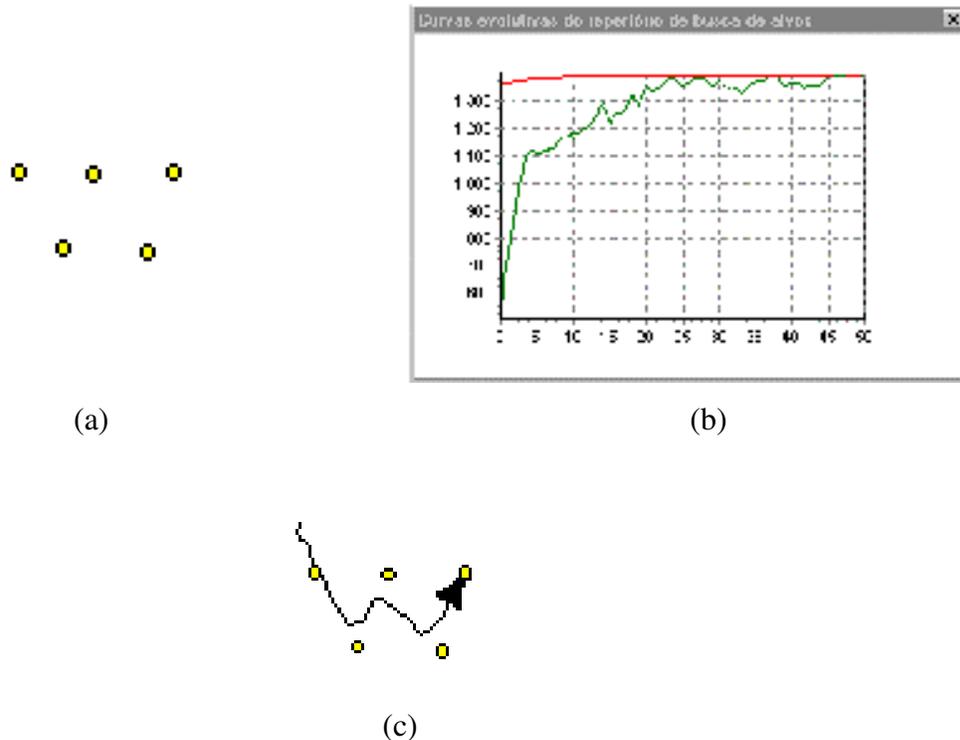


Figura 5.2 – Evolução do MBA com posições iniciais fixas: a) ambiente de evolução; b) Curvas evolutivas; c) navegação pós-evolutiva

A simulação apresentada na Figura 5.2 (c) foi realizada logo após o robô ter tido seu MBA evoluído. Neste caso, o MDO utilizado era controlado pelo Algoritmo 3.1. Durante este processo evolutivo, tanto a posição inicial do robô quanto a posição inicial dos alvos foi invariável ao longo de todas as gerações. Por isto, devido à utilização de seleção elitista, conforme mencionado na seção 5.2, pode-se observar na Figura 5.2 (b) que o melhor indivíduo é sempre mantido na população.

Um aspecto interessante da evolução apresentada pelas curvas do gráfico em (b) é que, logo no início da evolução, a avaliação do melhor indivíduo já era muito boa, mesmo tendo sido os genes de cromossomos da população gerados aleatoriamente. Este fato se explica por duas razões. Em primeiro lugar o tamanho dos cromossomos é pequeno, pois foram utilizadas 5 regras o que faz com que cada cromossomo possua apenas 10 genes. Em segundo lugar, utilizou-se um número relativamente alto de indivíduos na população considerando-se cromossomos com 10 genes. Assim, a probabilidade de que um indivíduo bom esteja presente logo nas primeiras gerações da população é considerável. Assim,

reduzindo-se o número de indivíduos da população a avaliação do melhor indivíduo se tornará pior nas primeiras gerações.

A Figura 5.3 ilustra como o MBA resultante de evolução genética pode ser utilizado para controlar a direção de deslocamento do robô mesmo em um ambiente completamente diferente daquele utilizado durante a evolução. Pode-se observar, portanto, na Figura 5.3, que o robô, além de atingir todos os alvos do ambiente, ainda dirige-se sempre na direção do alvo a ser atingido e descreve um percurso curto dentro do ambiente apresentado.

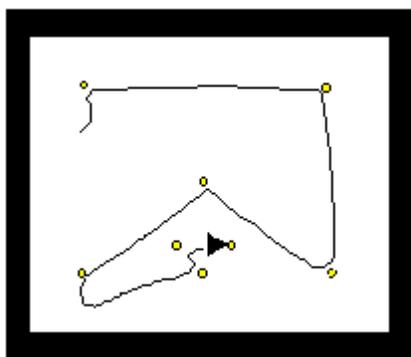


Figura 5.3 – Simulação com Módulo de Busca de Alvos evoluído

Ainda com relação ao processo evolutivo do MBA, algumas simulações foram realizadas considerando os efeitos de variações aleatórias na posição inicial dos robôs e nas posições dos alvos no ambiente a cada geração. Estes efeitos são apresentados nas duas próximas figuras. A Figura 5.4 apresenta dois gráficos contendo as curtas evolutivas do MBA, considerando: (a) variações apenas na posição inicial dos robôs a cada geração; e (b) variações tanto na posição dos robôs quanto nas posições dos alvos a cada geração.

A característica mais interessante a ser observada nos gráficos da Figura 5.4 é a variação, com freqüentes quedas, na avaliação do melhor indivíduo da população. Este fato é explicado justamente pelas variações nas condições iniciais de navegação a cada geração. Em (a), como a posição inicial dos robôs varia de uma geração para outra, pode ocorrer que, em determinada geração, o robô seja iniciado mais longe do primeiro alvo que na geração anterior. Como o número de passos necessários para que todos os alvos sejam atingidos é um termo da função de avaliação, quanto mais longe o robô for iniciado do alvo

menor será sua avaliação. Por isto, mesmo com uma estratégia de seleção elitista, variações de avaliações como as apresentadas nos gráficos da Figura 5.4 devem ocorrer, o que não implica que a estratégia elitista não esteja selecionando os indivíduos mais aptos daquela geração e que não haja progressos no processo evolutivo.



(a)



(b)

Figura 5.4 – Curvas evolutivas considerando variações aleatórias na topologia do ambiente de treinamento durante a evolução

Na Figura 5.5 (a) e (b) são apresentadas simulações realizadas com os MBA evoluídos, cujas curvas de avaliação são mostradas na Figura 5.4 (a) e (b), respectivamente. Como pode ser observado, o desempenho pós-evolutivo dos robôs é praticamente o mesmo, quando apenas as posições iniciais dos robôs são modificadas entre gerações, bem como quando tanto a posição inicial dos robôs quanto às posições dos alvos são modificadas. Desempenho semelhante também foi verificado na Figura 5.3, quando nenhuma variação foi imposta às condições iniciais das simulações ao longo das gerações.

Nas simulações apresentadas nas figuras 5.2, 5.3 e 5.5, nenhum tipo de ruído nos sensores de alvos havia sido considerado. Agora porém, faz-se conveniente a realização de simulações considerando algum tipo de perturbação na entrada apresentada ao MBA. Neste sentido o ruído utilizado nas simulações apresentadas na Figura 5.6 é definido pelos parâmetros ρ e δ . O parâmetro ρ determina a probabilidade de ocorrência de uma perturbação na entrada do MBA e o parâmetro δ determina a intensidade desta perturbação,

que consiste em uma variação na direção real com que o alvo é percebido pelos sensores de alvos do robô.

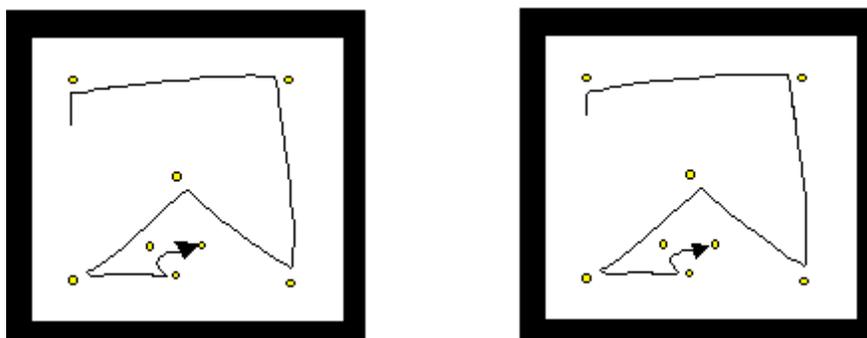


Figura 5.5 – Simulações considerando variações aleatórias na topologia do ambiente de treinamento durante a evolução

As três simulações apresentadas na Figura 5.6 foram realizadas com diferentes valores para a probabilidade ρ e para a perturbação δ . Em cada uma das simulações, a intensidade da perturbação corresponde ao máximo que permite ao robô navegar através do ambiente e atingir os alvos de acordo com a probabilidade de perturbação utilizada, o que significa que perturbações com intensidade além da utilizada, dada a probabilidade de perturbação aplicada, fazem com que o robô perca sua capacidade dirigir-se na direção dos alvos.

Assim, (a), utilizou-se probabilidade de perturbação de 10%. Com isto, o MBA suportou variações da entrada da ordem de 180°. Isto implica dizer que toda vez que uma perturbação era aplicada à entrada do MBA, o alvo era percebido na direção oposta à sua posição real. Como isto ocorria em apenas 10% das vezes, o robô conseguiu lidar com o problema.

Na simulação apresentada em (b), utilizou-se probabilidade de perturbação de 50%. Com isto, a variação máxima suportada pelo MBA para a direção com que o alvo era percebido caiu para 85°. Ainda assim, em alguns momentos, o robô enfrentou certa dificuldade de se dirigir na direção do alvo. Finalmente em (c), utilizou-se probabilidade de perturbação de 100%, de forma que toda a entrada do MBA sofria perturbação. Neste caso, variações

maiores que 20° eram suficientes para que o robô perdesse sua capacidade de navegação através do ambiente.

Ainda com respeito à simulação (c) da Figura 5.6, o fato de o robô suportar variações de 20° na direção do alvo, com 100% de probabilidade e ainda assim conseguir atingir todos os alvos do ambiente é explicado pela utilização do operador nebuloso de medida de possibilidade à entrada P do MBA e aos pesos A_j de antecedentes da rede mostrada na Figura 5.1. Graças à utilização de um operador nebuloso, a sensibilidade da rede a degradações do valor real da entrada pode ser controlada através da dispersão da função de pertinência utilizada para os conjuntos nebulosos. Como já foi dito, no caso dos módulos evoluídos neste capítulo, o valor desta dispersão é de $\pi/4$.

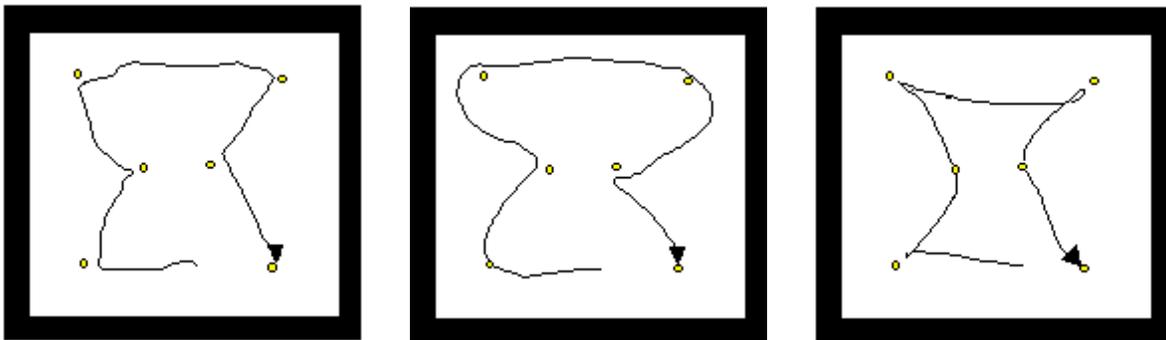


Figura 5.6 – Simulações com MBA evoluído e introdução de perturbações na entrada: (a) $\rho=10\%$, $\delta=180^\circ$; (b) $\rho=50\%$, $\delta=85^\circ$; (c) $\rho=100\%$, $\delta=20^\circ$

5.4 Resultados para o MDO

A evolução do MDO ocorreu segundo critérios semelhantes a aqueles utilizados para a evolução do MBA. O mecanismo de avaliação dos cromossomos do MDO foi apresentado na seção 5.2, tendo sido utilizada a equação (5.2) para a atribuição de crédito aos mesmos. O ambiente utilizado durante a evolução do MDO não possuía alvos e é apresentado pela Figura 5.7. Em (a), a primeira simulação após o término do processo evolutivo. Neste caso um alvo é introduzido no ambiente para que o robô navegue com o objetivo de atingi-lo. Isto facilita a visualização da trajetória descrita pelo robô. Durante a evolução, entretanto, bastava que o robô permanecesse distante dos obstáculos e não colidisse contra estes durante as navegações, o que dispensava a presença de um alvo.

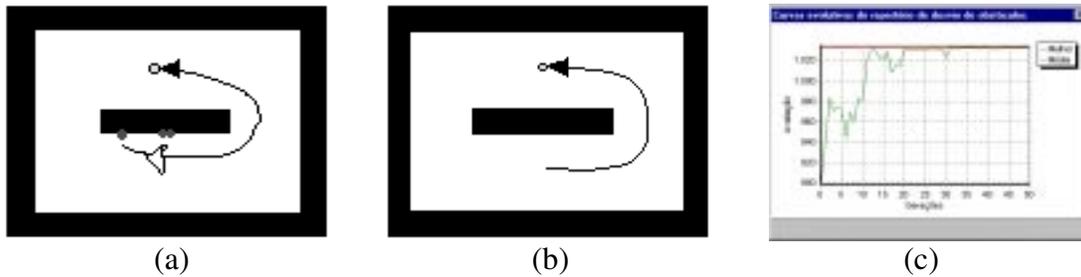


Figura 5.7 – Evolução do MDO: (a) primeira simulação após a evolução; (b) evolução com Módulo de Coordenação treinado; (c) Curvas evolutivas do MDO

Em (b), é apresentada uma segunda simulação, no mesmo ambiente que em (a) e partindo o robô do mesmo ponto. Agora porém, o Módulo de Coordenação (MC), treinado na situação (a), permite que o robô atinja o alvo sem colidir contra o ambiente. É importante frisar que o treinamento do Módulo de Coordenação não tem qualquer relação com a evolução dos módulos inatos, conforme apresentado neste capítulo. O treinamento do Módulo de Coordenação segue o descrito nos capítulos 3 e 4, sendo que seu desempenho será consistente caso os módulos inatos MDO e MBA estejam desempenhando devidamente o seu papel.

As curvas evolutivas da avaliação do melhor indivíduo e da avaliação média populacional são apresentadas em (c). Embora os robôs estejam sujeitos a variações aleatórias em suas posições iniciais de simulação a cada geração, não se observa na curva evolutiva do melhor indivíduo da Figura 5.7 (c) variações como as observadas nas curvas de melhor indivíduo da Figura 5.4. Isto se deve ao fato do ambiente utilizado para a evolução do MDO ser fechado e a distância média do robô com relação às paredes ser aproximadamente a mesma independentemente de onde o robô seja iniciado dentro do ambiente.

Algumas simulações foram realizadas, com o MDO evoluído, em ambientes diferentes daquele utilizado durante o processo evolutivo. A Figura 5.8 apresenta duas destas simulações. Em ambos os casos, o MBA é controlado pelo Algoritmo 3.2. Além disso, os Módulos de Coordenação já haviam passado por processos de aprendizagem, conforme descrito nos capítulos 3 e 4. Isto explica apenas parte do fato de nenhuma colisão ter sido

probabilidade de perturbação, não mais que 5 u.d. de perturbação foram suportados pelo MDO.

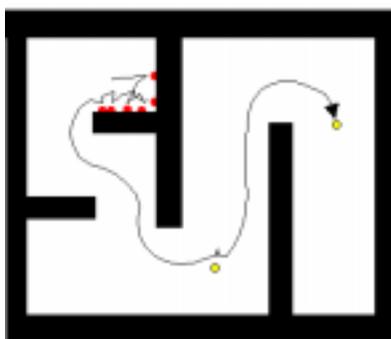
Por fim, a Figura 5.10 (c) apresenta uma simulação em que tanto o MDO quando o MBA são obtidos através de processos evolutivos, cujas curvas de avaliação são dadas respectivamente pelos gráficos apresentados em (a) e em (b). No início da navegação algumas colisões indicam o processo de aprendizagem do MC que, uma vez treinado, passa a coordenar os movimentos do robô e o leva a atingir os dois alvos do ambiente. Na simulação apresentada na Figura 5.10 (c), o robô atingiu o mais alto grau de autonomia dentre todas as simulações apresentadas neste trabalho, uma vez que todo o seu controle dispensa o auxílio de um especialista.



(a)



(b)



(c)

Figura 5.10 – Simulação com MDO e MBA evoluídos: (a) curvas de avaliação do MDO; (b) curvas de avaliação do MBA, (c) navegação

5.4 Visão sintética do capítulo 5

O intuito do estudo apresentado neste capítulo foi deixar que o robô incorporasse o conhecimento necessário para a realização de suas tarefas, ampliando assim seu grau de autonomia. Esta abordagem, que carrega uma inspiração biológica, parece ainda mais coerente quando se pretende atacar problemas sobre os quais não se possui conhecimento prévio suficiente.

A principal motivação deste capítulo não foi a de gerar um mecanismo de controle eficaz para os módulos inatos, fato este que poderia ter se dado simplesmente com a utilização da base de regras de avaliação, como no caso dos algoritmos 3.1 e 3.2 amplamente utilizados em simulações ao longo deste trabalho. O propósito aqui foi, antes, demonstrar que é possível que este mecanismo de controle seja construído pelo próprio controlador, sem interferências externas, o que confere mais autonomia ao sistema (SMITHERS, 1997).

Diversas simulações foram realizadas com dois módulos inatos responsáveis por comportamentos instintivos básicos de navegação autônoma, correspondendo ao desvio de obstáculos e à busca de alvos. Dentre estas simulações, aquelas consideradas mais relevantes foram incluídas na seção de resultados. Neste sentido, o texto apresenta alguns resultados promissores, considerando-se os sistemas de controle obtidos via evolução.

Além disso, embora a utilização de sistemas baseados em regras para a tarefa de navegação possua uma série de riscos e limitações previsíveis, a utilização de técnicas de inferência nebulosa mostrou ser uma boa alternativa para a exploração de seus pontos fortes e melhoria de sua robustez. Esta consideração é especialmente importante no que se refere à presença de ruídos nos sensores de obstáculos e de alvos. Neste caso, algumas simulações apontam para uma boa tolerância do controlador a variações nos sinais de entrada.

Capítulo 6

Aprendizagem por Recordação

6.1 Introdução

Como ficou comprovado no capítulo 3, a criação do conceito de valor acumulado de ativação sináptica (VAAS) promoveu uma melhoria considerável junto ao mecanismo original de aprendizagem do sistema de navegação autônoma, proposto por FIGUEIREDO (1997). Verificou-se também que a essência da proposta então apresentada consiste no balanceamento dos processos de aquisição de conhecimento, que até então ocorriam de forma desorganizada. Entretanto, mesmo com este aperfeiçoamento, os processos de aprendizagem do controlador ainda se encontram muito restritos a situações excessivamente bem definidas, quais sejam: colisões e capturas de alvos.

Neste sentido, tais processos de aprendizagem diferem claramente daqueles que ocorrem em sistemas biológicos evoluídos (e.g. seres humanos). Nestes sistemas, não parece haver uma definição clara e precisa dos eventos que devam levar à aquisição de conhecimento por parte do indivíduo. Ao contrário, a aquisição de conhecimentos acompanha a interação constante do indivíduo com o ambiente que o cerca e sabe-se que é disparada devido a estímulos do próprio ambiente. Entretanto, aparentemente não há um mecanismo que defina com exatidão a natureza de tais estímulos e as ocasiões em que ocorrem.

O objetivo principal do mecanismo de aprendizagem proposto nesta seção é, a partir de inspiração biológica, dotar o robô da capacidade de aprender mediante da interação com o ambiente sem a necessidade de uma colisão ou uma captura. Este processo de aprendizagem exige dois passos: em um primeiro momento, o robô deve identificar as situações que lhe oferecem perigo. Estas situações correspondem àquelas que levaram a colisões no passado. Em um segundo momento, se o robô conseguir superar a dificuldade em que se encontrava, deve reconhecer o fato e premiar os neurônios do MC responsáveis pelo bom desempenho do sistema.

A partir disto, a capacidade do robô em reconhecer situações de risco implica que os processos de aprendizagem baseados na teoria do condicionamento operante devem ser preservados, de forma que, em um primeiro momento, a aprendizagem do robô se dê exclusivamente por colisões e capturas de alvos, até que o robô esteja apto a reconhecer as situações de risco do ambiente. Mesmo após o desenvolvimento da habilidade de coordenação dos comportamentos inatos, a aprendizagem por colisão ainda perdurará caso uma colisão venha a ocorrer. Pelo fato de o controlador aprender baseado no reconhecimento de eventos passados, este mecanismo de aprendizagem é dito Aprendizagem por Recordação.

6.2 A Memória de Eventos Críticos

Para a realização da aprendizagem por recordação, um novo módulo neural deve ser introduzido no modelo do sistema de navegação autônomo apresentado no capítulo 4. Este módulo, cuja arquitetura será descrita mais adiante, foi denominado Memória de Eventos Críticos (MEC). Este nome possui estreita relação com sua função no sistema, qual seja, levar ao MC estímulos semelhantes aos produzidos pelos sensores de colisão. Como a incorporação desta nova estrutura neural afeta todas as demais estruturas do sistema, o que segue é uma descrição do Sistema de Navegação Autônomo Baseado em Redes Neurais Nebulosas com a presença da MEC.

Um esquema genérico e simplificado do sistema é apresentado na Figura 6.1. Como a proposta de aperfeiçoamento a ser apresentada agora contempla apenas os mecanismos de aprendizagem dos neurônios do MC responsáveis pelo sinal W_r , apresentados nos capítulos 3 e 4. Os neurônios responsáveis pelo sinal W_s , bem como os módulos inatos não são considerados na figura, uma vez que o mecanismo de aprendizagem por recordação apresentado aqui utiliza a recordação de uma colisão para realizar apenas a aprendizagem dos neurônios responsáveis pelo sinal W_r . Ainda com o intuito de simplificar o entendimento do esquema, a ilustração traz as ligações de apenas um neurônio do MC com cada neurônio da MEC. Na realidade porém, cada neurônio representado no MC liga-se através de sinapses a todos os neurônios da MEC.

A função do Módulo de Sensores de Colisão (MSC) é simples. A cada colisão os neurônios deste módulo enviam estímulos à MEC correspondendo à intensidade com que cada um percebe a colisão. Assim, se o robô colide frontalmente, o neurônio deste módulo posicionado na parte frontal do robô irá gerar o estímulo mais forte, sendo que os neurônios de sua periferia enviam estímulos proporcionais à distância com que estão em relação ao neurônio frontal que efetivamente detectou a colisão. Cada neurônio do MSC possui uma única ligação com um neurônio da MEC.

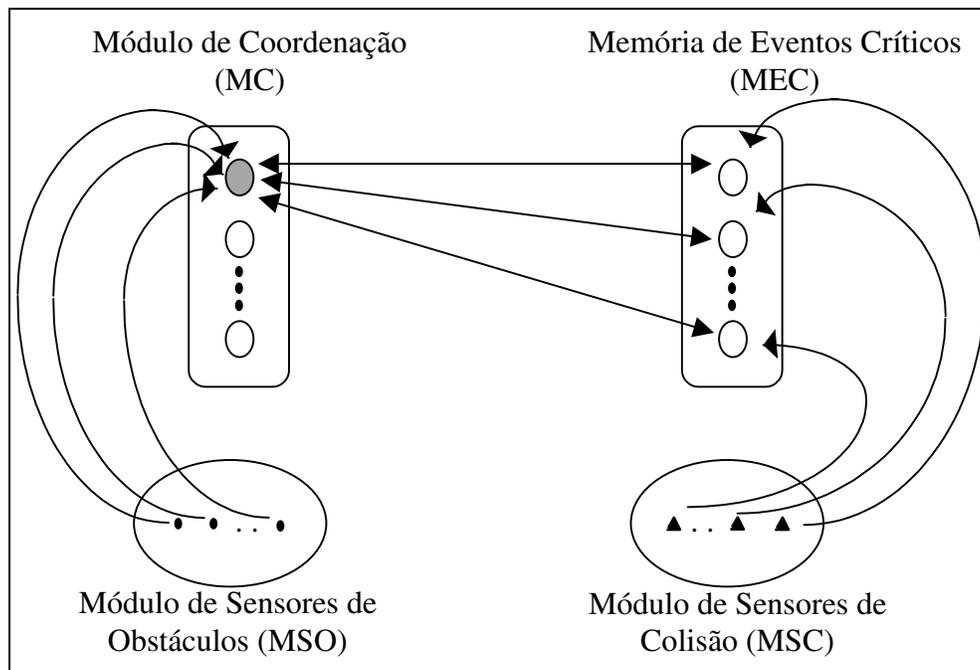


Figura 6.1 – Esquema parcial do controlador autônomo considerando a MEC

A função do Módulo de Sensores de Obstáculos (MSO) é similar àquela do MSC, pois quando há uma colisão seus neurônios capturam os estímulos do ambiente e os encaminham ao MC. Entretanto, a natureza dos estímulos produzidos é diferente, pois no caso do MSO corresponde ao panorama do ambiente que cerca o robô naquele momento, expresso pelo vetor de distâncias do robô com relação aos obstáculos que o cercam, conforme apresentado no capítulo 3. Este panorama é registrado pelos neurônios do MC. Além disso, a todo momento o MSO produz tais estímulos durante a navegação, e não apenas durante uma colisão. Quanto à topologia, cada neurônio do MSO se liga a todos os neurônios do MC, conforme apresentado pela Figura 3.3, embora nem todas estas ligações tenham sido representadas na Figura 6.1.

A estrutura chave do controlador é o Módulo de Coordenação (MC), descrito nos capítulos 3 e 4, cuja função é coordenar os movimentos do robô para que este atinja seu objetivo sem colidir contra os obstáculos do ambiente. O mecanismo de aprendizagem por recordação visa o ajuste conveniente dos pesos deste módulo neural. Assim, a MEC, que consiste no principal item da proposta de aperfeiçoamento aqui apresentada, não tem ação direta sobre as decisões de navegação do robô. As ligações sinápticas que convergem para o MC provêm do MSO e da MEC. As ligações sinápticas entre o MC e a MEC são particularmente interessantes e serão analisadas separadamente mais adiante. Além disso, o modelo apresentado na Figura 6.1 leva à formalização de um novo modelo de neurônio no MC, mais particularmente aplicado aos neurônios responsáveis pelo sinal Wr . Este novo modelo é esboçado na Figura 6.2.

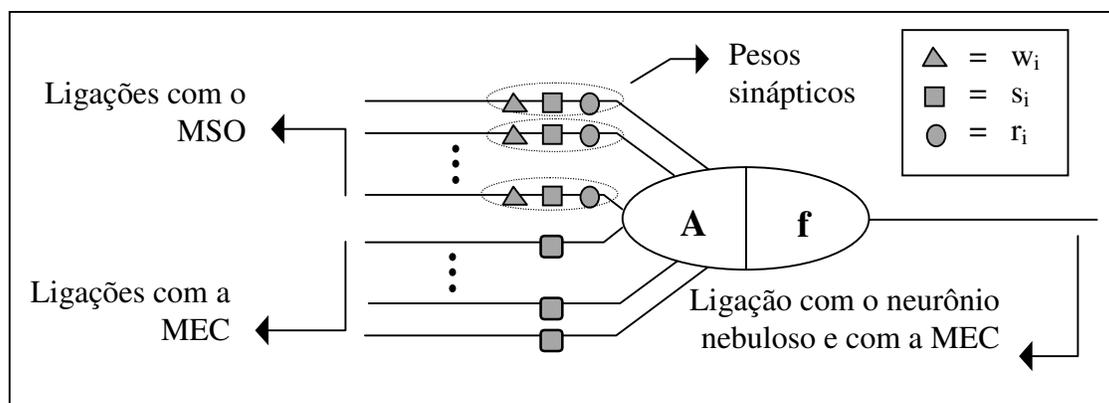


Figura 6.2 – Novo modelo de neurônio do MC

A função da MEC é reproduzir comportamentos de uma colisão sem que esta ocorra. A denominação “Memória de Eventos Críticos” se deve exatamente a esta peculiaridade. Enquanto o robô ainda não possui os conhecimentos necessários para evitar colisões, a cada colisão esta rede retém os estímulos advindos do MSC para a situação da colisão, de forma que quando o robô se aproxima de uma situação similar esta rede tem a capacidade de reproduzir a percepção causada pela colisão. Este recurso será fundamental para que o robô possa aprender sem colidir. O modelo de um neurônio da memória de eventos críticos é esboçado na Figura 6.3.

O processo de aprendizagem por recordação implica a premiação dos neurônios da rede neural nebulosa do MC responsáveis pelo sinal Wr e que contribuam para o sucesso da navegação do robô. Este tipo de premiação já ocorre quando um alvo é capturado, como descrito anteriormente. O desafio agora é estender esta premiação para toda a navegação e não apenas para capturas de alvos. Entretanto, para que isto seja feito, o robô precisa saber quando se livrou de uma situação de perigo, e portanto realizou uma manobra bem sucedida, e que neurônios foram os responsáveis pela manobra realizada.

Primeiramente, para saber que obteve êxito em uma situação difícil o robô deve identificar quando se encontra em tal situação. Ora, o problema de identificar situações de perigo de colisão foi solucionado com a MEC. Entretanto, todo o processo de previsão do sucesso de uma manobra está intimamente relacionado não à MEC em si, mas às ligações sinápticas entre a MEC e o MC. A natureza destas ligações e a forma como são estabelecidas serão descritas a seguir.

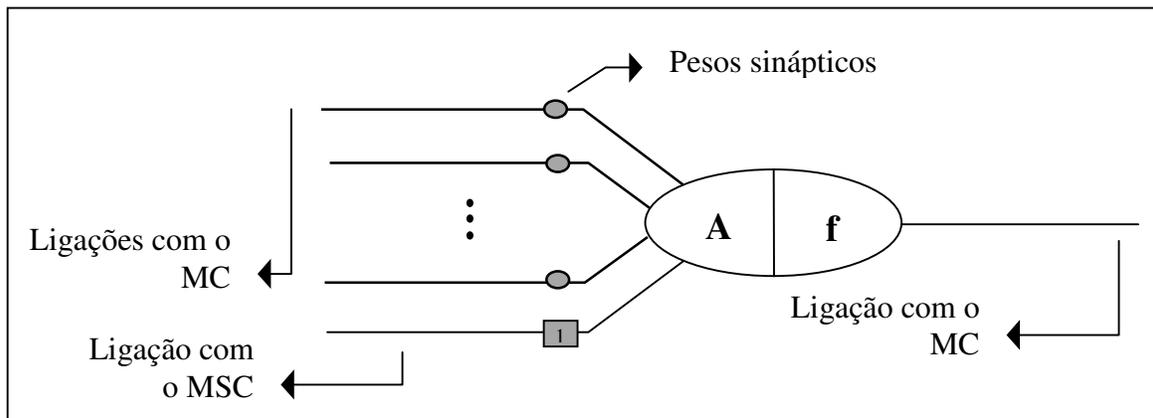


Figura 6.3 – Modelo de neurônio da MEC

6.3 Criação de relações entre o MC e a MEC

Diferentemente do que ocorre com os neurônios do MC, que são “criados” ao longo da navegação conforme a ocorrência de colisões, assume-se que todos os neurônios da MEC sejam estruturas inatas do robô, sendo que o número destes neurônios é igual ao número de neurônios do MSC. Ocorre então que a cada colisão um neurônio do MC é criado e então são estabelecidas ligações entre este neurônio e todos os neurônios da MEC, respeitando a seguinte política: o neurônio do MC recém criado, recebe estímulos de uma ligação

sináptica de cada neurônio da MEC; além disso, sua saída também é direcionada para todos os neurônios da MEC, conforme a Figura 6.1. Para tanto, supõe-se que os valores dos pesos sinápticos das ligações entre o MSC e a MEC possuam valores constantes e iguais a 1.

Em princípio, os neurônios do MC e da MEC estão totalmente interligados. No entanto, quando um neurônio do MC, bem como suas ligações com a MEC são estabelecidos, não expressam uma relação clara entre ambos os módulos. Esta relação vai então sendo estabelecida aos poucos ao longo das colisões sucessivas do robô. Neste estágio, há um reforço seletivo das ligações sinápticas entre o MC e a MEC, que ocorre segundo a lei de aprendizagem proposta por HEBB (1949).

Os neurônios da MEC só recebem estímulos dos sensores mediante uma colisão contra algum obstáculo. Recebendo estes estímulos, aqueles neurônios que mais sofreram a ação da colisão irão disparar. Ao mesmo tempo, algum dos neurônios do MC, que recebem continuamente sinais do MSO, também irá disparar. Pela lei de Hebb, conforme apresentado no capítulo 2, quando dois neurônios adjacentes disparam consecutivamente, há um reforço das ligações sinápticas que os comunicam. Neste caso, com a ocorrência de sucessivas colisões, verifica-se o estabelecimento de relações entre alguns neurônios do MC com alguns neurônios da MEC. Estas relações podem ser melhor compreendidas graficamente, como mostra a Figura 6.4.

Os gráficos apresentados na figura são esquemas que representam estimativas aproximadas baseadas em simulações reais, porém não apresentando os valores numéricos de cada peso sináptico. Não obstante, servem perfeitamente para a compreensão das relações que se estabelecem nas ligações entre a MEC e o MC. Estes gráficos apresentam reações diferentes a um mesmo evento: uma colisão frontal, conclusão deduzida da observação do panorama apresentado em (a).

Ainda com relação ao gráfico observado em (a), tem-se o registro da configuração dos n pesos sinápticos w_i do neurônio, responsáveis pela captura das distâncias do robô ao obstáculo contra o qual este colidiu. Nota-se que a menor distância aparece no centro do

gráfico, correspondendo ao peso sináptico ligado ao neurônio sensor situado na parte frontal do robô. É fundamental que se note que em (a) são registrados os pesos sinápticos de um único neurônio do MC.

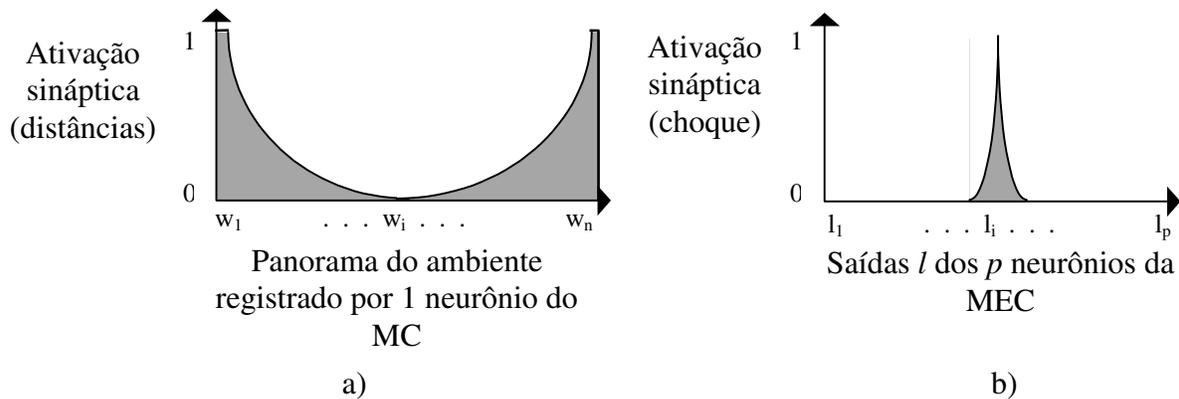


Figura 6.4 – Estímulos após uma colisão frontal: a) registrados por neurônios do MC; b) por um neurônio da MEC

O gráfico em (b) representa a mesma situação do gráfico em (a), porém não mais em termos de distâncias a obstáculos, mas em termos de intensidade do choque sofrido. O que se verifica então em (b) é que, em um ponto do robô (parte frontal), o choque é percebido com mais intensidade. Nas periferias deste ponto a sensação vai se atenuando até desaparecer por completo. Além disso, em (b), o gráfico representa o agrupamento dos disparos de todos os p neurônios da MEC.

Assim, cada neurônio da MEC emite a sensação de um único ponto do robô. Esta característica é expressa no modelo do neurônio da Figura 6.3. Embora este neurônio contenha diversas sinapses, apenas uma provém do MSC. Além disto esta sinapse tem peso constante e igual a 1, não definido por experiência de navegação, o que indica que o sinal proveniente do sensor não é armazenado, mas serve apenas para efetuar o disparo do neurônio.

Uma vez tendo sido compreendida a relação entre os neurônios do MC e da MEC durante uma colisão, torna-se mais simples a compreensão do reforço seletivo das ligações sinápticas entre estas mesmas redes neurais, a que aludia o texto anteriormente. Ocorre que

uma colisão sempre gera o disparo de um único neurônio do MC, ao passo que na MEC diversos neurônios irão disparar, com maior ou menor intensidade. Com isto, como visto pela lei de Hebb, as ligações entre estes neurônios irão se fortalecer. Ainda pela lei de Hebb, se os neurônios não dispararem consecutivamente, suas ligações, ao contrário, tendem a se enfraquecer.

Assim, após várias colisões, as ligações entre o MC e a MEC passam a estabelecer relações relevantes entre situações de colisão (MC) e a sensação de choque que aquela situação provoca (MEC). Os dois módulos continuam totalmente interligados, mas algumas destas ligações não mais possuem papel na arquitetura do controlador, por terem sido enfraquecidas, ao passo que outras foram fortalecidas, tornando-se muito atuantes. A Figura 6.5 ilustra hipoteticamente como ficariam as ligações entre o MC e a MEC após o reforço seletivo das sinapses.

Esta figura, considera apenas as ligações entre a MEC e o MC, sendo que as demais estruturas do controlador são ocultadas no modelo. Nota-se que o robô possui apenas 4 sensores de colisão para maior simplicidade. Também para simplificar a análise, apresenta-se apenas uma ligação entre um neurônio do MC e um neurônio da MEC. No entanto o que ocorre realmente é uma ligação duplamente direcionada entre o MC e a MEC, assim como na Figura 6.1.

Esta situação é representada pelas setas com duas pontas na Figura 6.5. Outra consideração importante a ser feita é que as ligações fortalecidas são representadas por linhas cheias, ao passo que as ligações enfraquecidas são representadas por linhas pontilhadas. Por fim, os neurônios da MEC são agrupados de acordo com a distribuição espacial dos sensores à direita ou à esquerda do robô.

De acordo com a Figura 6.5, é possível verificar que a primeira colisão foi frontal, dado que o primeiro neurônio do MC se liga mais fortemente aos dois neurônios centrais da MEC, justamente aqueles ligados aos sensores mais frontais do robô. Já a segunda colisão foi lateral e ocorreu mais precisamente do lado direito, pois as ligações mais fortalecidas do

segundo neurônio do MC são aquelas para com neurônios da MEC ligados a sensores do lado direito do robô. Pelos mesmos motivos, enfim, podemos dizer que a última colisão se deu do lado esquerdo do robô.

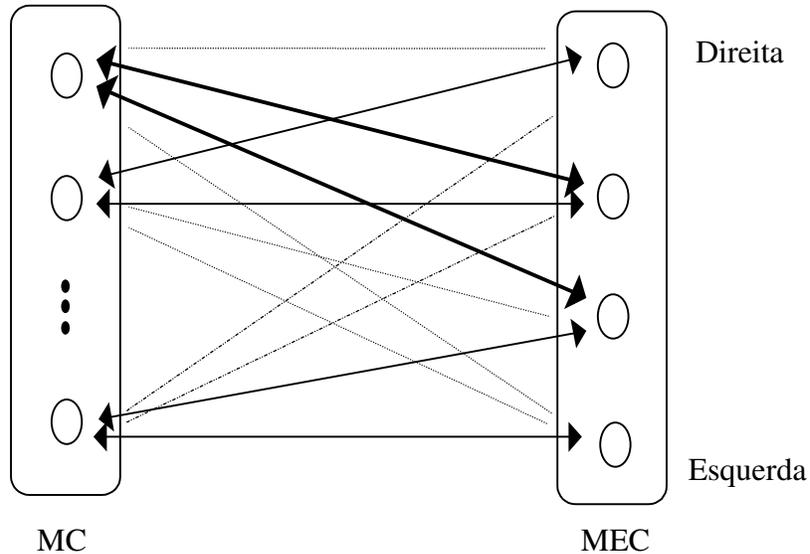


Figura 6.5 – Relações sinápticas entre o MC e a MEC após diversas colisões

6.4 Recuperação de estímulos críticos

Após a formação das relações apresentadas, mediante colisões, o robô estará então apto a proceder a aprendizagem da mesma maneira como era feita anteriormente (por reforço dos pesos sinápticos do MC), mesmo na ausência de colisões contra os obstáculos. Quando o robô se aproxima de alguma situação de risco de colisão, um dos neurônios do MC irá disparar alertando o robô para aquela situação. Seu sinal então estimula o neurônio nebuloso para a produção do sinal W_r . A diferença agora é que o sinal deste neurônio do MC também estimula os neurônios da MEC, com os quais este estabeleceu uma relação. Este estímulo faz com que os neurônios da MEC, por sua vez, devolvam um estímulo de lembrança de uma colisão ao neurônio do MC. Este estímulo permanece até que o robô consiga efetuar uma manobra que o livre da situação de risco de colisão. Quando isto acontecer, o estímulo então cessa.

Esta recordação de um choque é o ponto mais importante da teoria que vem sendo desenvolvida ao longo desta seção, com respeito à criação de um método de aprendizagem

sem colisão, de forma que tudo o que foi dito até aqui tem por finalidade a criação deste comportamento. Como mencionado anteriormente, no início do capítulo, em um primeiro momento o robô deve reconhecer as situações que lhe oferecem perigo, o que é feito quando a lembrança de um choque está ativa. Em um segundo momento, se o robô conseguir superar a dificuldade em que se encontrava, deve reconhecer tal fato para que possa realizar convenientemente a aprendizagem, o que ocorre quando a lembrança de um choque cessa.

Se a recordação de um choque cessar devido a uma manobra bem sucedida, o neurônio do MC que reconheceu a situação de risco terá suas sinapses reforçadas. Entretanto, se houver uma colisão os mecanismos de aprendizagem utilizados serão aqueles apresentados nos capítulos 3 e 4, desenvolvidos por FIGUEIREDO (1997) e aperfeiçoados por CRESTANI (1999). É importante notar que todos os processos de aprendizagem anteriores são mantidos, de forma que a aprendizagem sem colisão não os substitui, mas apenas os complementa.

Já se sabe então qual neurônio deve ter seus pesos sinápticos reforçados e em que situações. Além disso, sabe-se que cada neurônio do MC possui diversas sinapses, relativas aos neurônios do MSO. Cada uma destas sinapses deve ser premiada na mesma proporção com que tenha contribuído para a manobra que retirou o robô da situação de risco. Assim, uma boa ferramenta para determinar as sinapses que devem ser mais ou menos reforçadas é o VAAS, descrito no capítulo 3. O reforço então para cada sinapse é realizado de acordo com o Algoritmo 4.2.

Há ainda uma última consideração a ser feita sobre a estrutura então descrita. Os neurônios do MC disparam constantemente, pois recebem constantemente sinais do MSO durante a navegação, de forma a controlar os movimentos do robô a cada instante. Assim, não é possível que a memória de eventos críticos tenha seus neurônios disparados a cada estímulo recebido do MC. Caso isto ocorresse, esta memória estaria indicando constantemente uma sensação de choque, mesmo em situações fora de risco de colisão. Para evitar este tipo de

comportamento, é preciso estabelecer-se um limiar ξ nos neurônios da MEC, sendo que somente acima deste limiar os mesmos disparem emitindo a lembrança de choque.

6.5 Simulações e testes

A descrição da MEC e da aprendizagem por recordação, realizada acima tem, como principal motivação a apresentação clara e coesa do mecanismo de aprendizagem introduzido neste capítulo. A formalização segue os princípios da lei de aprendizagem de Hebb, o que deve servir para enfatizar sua inspiração biológica. Não obstante, na implementação do conceito de aprendizagem por recordação uma estrutura equivalente, porém computacionalmente menos custosa, foi criada.

De acordo com esta implementação alternativa, a MEC e suas ligações com o MC não necessitam ser explicitamente definidas. A invés disto, assume-se que quando o sinal W_r ultrapassar o limiar $\xi = 0,25$, as sinapses que tiverem contribuído para a ativação recebem o estímulo que lhes seria dado pela MEC. Este estímulo consiste em marcar as sinapses que devem ter sua importância reforçada no momento em que W_r voltar a se tornar inferior a ξ . Uma vez que isto ocorrer, as sinapses são então convenientemente reforçadas segundo seu VAAS, de forma semelhante à apresentada pela equação (3.6).

Nas simulações realizadas com o intuito de comparar os modelos com e sem aprendizagem por recordação, utilizou-se controladores com velocidade variável, de acordo com o descrito no capítulo 4. Além disso, o MDO e o MBA são implementados na forma dos algoritmos 3.1 e 3.2, respectivamente. Por fim, os sensores de obstáculos e de alvos não estão sujeitos a ruído ao longo das simulações.

A Figura 6.6 apresenta uma comparação interessante entre um robô que não possui mecanismo de aprendizagem por recordação (situação (a)) e um robô que o possui (situação (b)). Em ambos os casos, os robôs ainda não haviam sido treinados para coordenarem seus movimentos. Além disso, o ambiente foi concebido para alternar pontos de dificuldade de navegação com espaços de mais fácil locomoção. Com isto, pretendia-se valorizar a ação

do mecanismo de aprendizagem por recordação assim que o robô saísse de um ponto de dificuldade.

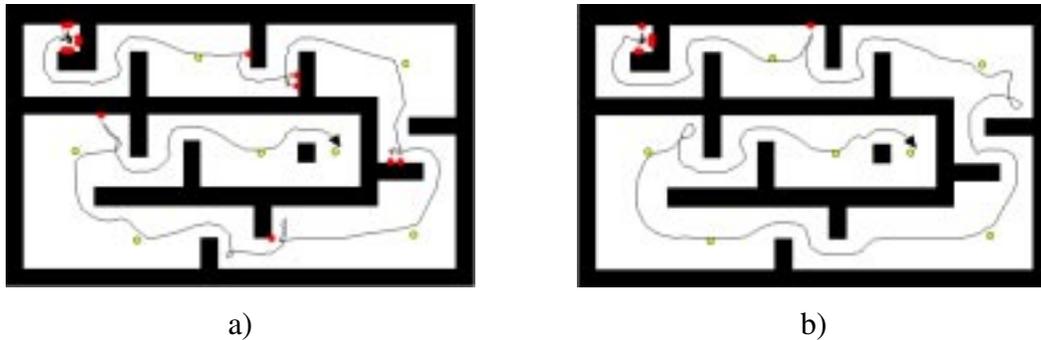


Figura 6.6 – Comparação entre controladores: a) sem aprendizagem por recordação; b) com aprendizagem por recordação

Como resultado, na simulação (a), em que o robô não possuía o mecanismo de aprendizagem por recordação implementado, foram observadas 16 colisões, ao passo que na situação (b) apenas 10 colisões ocorreram. Por outro lado, o mecanismo de aprendizagem por recordação foi utilizado 92 vezes na simulação (b). Com isto, a importância das sinapses mais relevantes do MC foi enfatizada e houve menos necessidade de neurônios para que o robô coordenasse seus movimentos.

Contudo, apesar de a Figura 6.6 apontar para um melhor desempenho do MC com aprendizagem por recordação, este mecanismo possui um efeito colateral que merece ser comentado. Conforme as sinapses do MC são reforçadas, este passa a dar maior importância ao sinal do MDO que ao sinal do MBA. Ao longo da navegação, os sucessivos reforços que o mecanismo de aprendizagem proporciona ao MC faz com que o robô se torne cada vez mais contrário à aproximação de obstáculos.

Com o passar do tempo o robô pode desenvolver algumas anomalias semelhantes às apresentadas no capítulo 4 sobre o controlador sem controle de velocidade (Figuras 4.7 e 4.8). Na Figura 6.6 (b) pode-se observar em alguns trechos que o robô procura descrever uma trajetória afastada das paredes do ambiente, procurando se locomover pelo meio dos corredores. Outro elemento das simulações apresentadas, que ilustra a maior cautela

adotada pelo robô com aprendizagem por recordação, é sua velocidade média, que foi de 0,4 ud./i., contra 0,5 ud./i. do robô sem aprendizagem por recordação.

Aparentemente, esta diferença de velocidade média é muito pequena para se inferir algo significativo a respeito dos desempenhos dos robôs. Mesmo assim, já foi suficiente para que o robô da simulação (a) concluísse sua navegação em 3921 passos, sendo que o robô da simulação (b) necessitou de 5036 passos para chegar ao último alvo. Em termos de tempo de navegação isto representa uma diferença de dois minutos e dezessete segundos entre o robô da simulação (a) que concluiu o percurso em 7 minutos e 31 segundos e o robô da simulação (b), que levou 9 minutos e 48 segundos para chegar até o fim, considerando os mesmos recursos computacionais.

Levando-se em conta a distância percorrida pelos dois robôs ao final da simulação, as diferenças entre os dois modelos diminuí bastante. Neste caso, a trajetória do robô com aprendizagem por recordação teve um comprimento de 2056 ud., sendo que a trajetória do robô sem aprendizagem por recordação teve um comprimento de 2062 ud., ligeiramente mais comprida. Esta diferença pouco expressiva na simulação apresentada levanta a suspeita de que o novo mecanismo de aprendizagem não tenha influência forte sobre o comprimento do percurso final em ambientes pequenos, sendo que os dois fatores principais de distinção entre um modelo e outro se referem à segurança e ao tempo de realização do percurso. A confirmação destas idéias carece, no entanto, de um número maior de simulações.

Como o mecanismo de aprendizagem por recordação age constantemente, mesmo após o robô ter diminuído a frequência de suas colisões durante a navegação, simulações com percurso mais longo tendem a ressaltar as diferenças entre modelos de controladores com e sem capacidade de aprender sem a necessidade de uma colisão. Neste sentido, as duas próximas figuras apresentam simulações em um ambiente que além de possuir um percurso extenso, apresenta ainda obstáculos dispostos de forma que o robô deva descrever uma trajetória bastante sinuosa ao longo da navegação, alternando pontos mais estreitos com regiões de maior liberdade de movimento.

Na Figura 6.7, é apresentada uma simulação neste ambiente com um modelo de controlador que não apresenta aprendizagem por recordação. Os onze alvos dispostos ao longo do ambiente foram atingidos, mas foram necessárias 38 colisões antes que o MC conseguisse coordenar os movimentos do robô adequadamente, tendo sido necessário percorrer quase a metade do trajeto para isto. Do início ao fim do percurso, o robô teve de dar 20823 passos e percorrer uma distância de 8747 ud., tendo apresentado uma velocidade média de 0,4 ud./i.

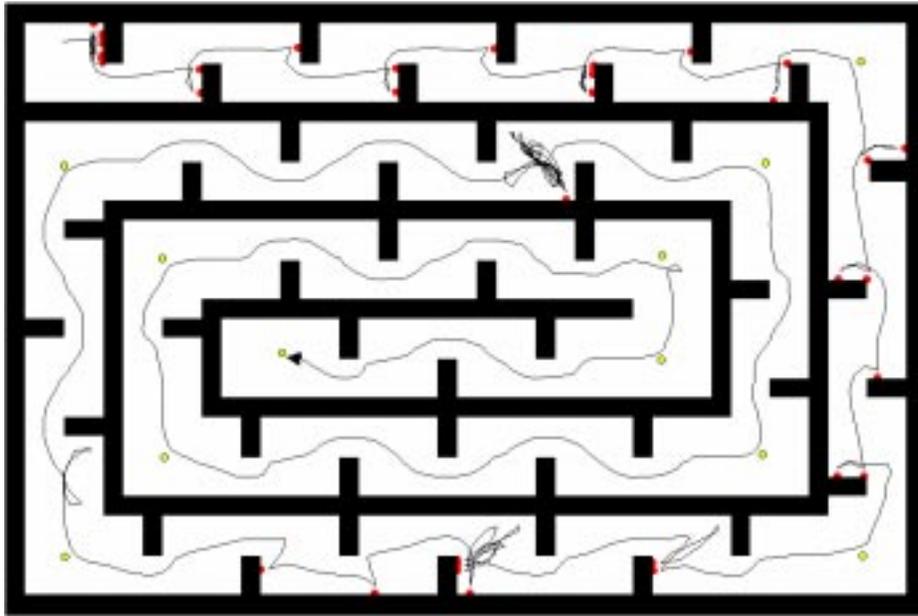


Figura 6.7 – Robô sem aprendizagem por recordação em percurso longo

Na Figura 6.8, uma simulação no mesmo ambiente da Figura 6.7 é realizada, exceto que o modelo de controlador utilizado agora possuía aprendizagem por recordação. A diferença visível de desempenho que aponta para a superioridade do modelo da Figura 6.8 pode ser expressa em números. Os mesmos onze alvos foram atingidos ao custo de apenas 15 colisões, tendo em sua maior parte ocorrido logo no início da navegação. Por outro lado, o mecanismo de aprendizagem por recordação foi utilizado 319 vezes.

Outra questão interessante observada nestas duas simulações é que a velocidade média do modelo da Figura 6.8 foi de 0,5 ud., ficando acima do modelo da Figura 6.7, diferentemente do que ocorreu com as simulações da Figura 6.6. Neste caso, a topologia do ambiente

influenciou o resultado final, uma vez que o grande número de colisões do modelo sem aprendizagem por recordação fez com que este assumisse uma postura defensiva, a partir da metade do percurso, reduzindo bastante sua velocidade média. O mesmo não acontece com o outro modelo de controlador.

Como resultado da velocidade média mais elevada, foram necessários 14856 passos para que o robô da Figura 6.8 percorresse todo o ambiente. Um número bem menor que o apresentado pelo robô da Figura 6.7. Outra diferença se refere ao comprimento do percurso. Diferentemente da Figura 6.6, em que esta diferença é pouco relevante, aqui o robô da Figura 6.8 percorreu 6774 ud., ou seja, o trajeto foi 22% mais curto que o do robô sem aprendizagem por recordação, o que certamente se deveu também ao grande número de colisões, que por sua vez foram influenciadas pela topologia do ambiente.

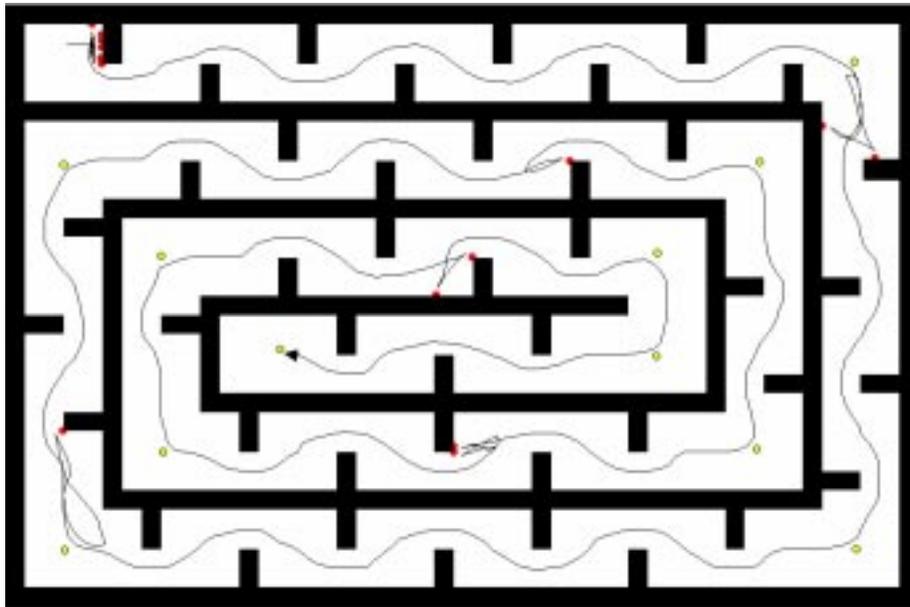


Figura 6.8 – Robô com aprendizagem por recordação em percurso longo

A forma como o mecanismo de aprendizagem por recordação age nas sinapses de um neurônio do MC pode ser observada e comparada com o mecanismo de aprendizagem após colisão, através da Figura 6.9. Em (a), é apresentada a configuração dos vetores de pesos sinápticos \vec{w} e \vec{s} logo após uma colisão ter ocorrido, o que pode ser verificado pelo fato de

os componentes do vetor \vec{s} possuírem o valor da constante START (igual a 0,1, conforme Figueiredo (1997)).

Na situação (b), o mesmo neurônio é observado logo após ter passado por um segundo processo de aprendizagem por colisão. Como já foi descrito em capítulos anteriores, este processo altera muito pouco o vetor \vec{w} , enquanto que o vetor \vec{s} é fortemente modificado para ressaltar a importância dos componentes mais significativos de \vec{w} . Estes componentes mais significativos correspondem a medidas baixas de distâncias observadas em diversas colisões.

A situação (c) apresenta a influência do novo processo de aprendizagem sobre as sinapses de um neurônio do MC. Este processo age exclusivamente sobre pesos do vetor \vec{s} , diferentemente da aprendizagem por colisão, que embora praticamente não modifique o padrão do vetor \vec{w} , age sobre ele quando necessário. A aprendizagem por recordação também é um tanto sutil com relação ao vetor \vec{s} . Apenas algumas sinapses selecionadas durante a navegação têm sua expressividade ressaltada. Na situação (c), apenas os pesos s_7 e s_{38} são reforçados com relação à situação (b).

Na realidade, a intensidade do ajuste sináptico realizado pelo processo de aprendizagem por recordação pode variar de acordo com o tempo a que o robô se sujeitou a uma situação de risco. Se este tempo se prolongar e diversas situações forem apresentadas ao robô, possivelmente um número grande de sinapses serão marcadas para serem reforçadas assim que o robô se livrar da situação de risco.

Disto surge um problema para se implementar este mecanismo de aprendizagem. Caso o risco de colisão se prolongue muito, pode ser que as sinapses que tenham agido logo no início deste período não tenham contribuído efetivamente para que o robô se livrasse do perigo caso isto venha a ocorrer. Decidir que sinapses devem ser reforçadas pelo mecanismo é um ponto central e ainda não totalmente solucionado.

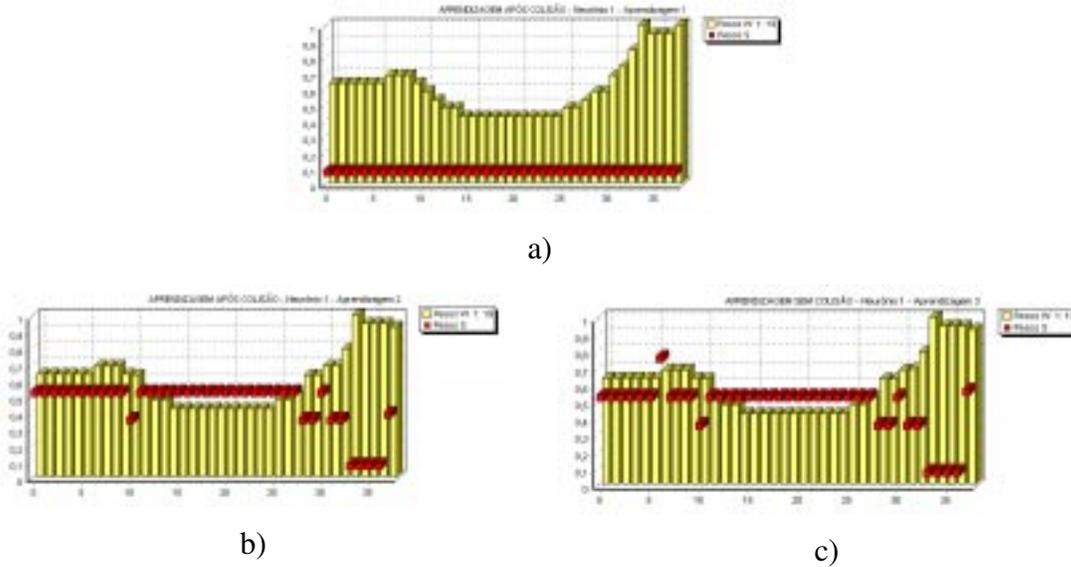


Figura 6.9 – Três configurações de um neurônio do MC: a) após sua criação; b) após segunda aprendizagem por colisão; c) após aprendizagem por recordação

6.6 Visão sintética do capítulo 6

Neste capítulo, um mecanismo de aprendizagem que age pela recordação da ocorrência de uma colisão foi formalizado, levando-se em consideração um modelo de memória intitulado Memória de Eventos Críticos (MEC) e fundamentando-se em princípios da lei de aprendizagem de Hebb. A implementação prática da estrutura formalizada considerou um modelo equivalente, porém computacionalmente menos custoso, por não necessitar da definição explícita da MEC e de suas ligações com o MC.

Simulações realizadas com o modelo implementado, em comparação com um controlador que não possuía aprendizagem por recordação, evidenciaram as vantagens do novo modelo de aprendizagem, especialmente em ambientes com topologias mais complexas. Os benefícios do novo modelo de aprendizagem também podem ser melhor observados conforme o comprimento do percurso se torna maior. Em todos os testes realizados com o modelo de aprendizagem por recordação, utilizou-se limiar ξ igual a 0,25. Este limiar foi determinado empiricamente.

Com tudo isto, o processo formalizado apresenta um ponto em aberto. A aprendizagem por recordação é realizada sobre o neurônio tão logo o sinal do MC caia abaixo do limiar ξ .

Entretanto, devido a imprecisões do sistema, o robô poderia aparentemente desviar-se provisoriamente de um obstáculo, fazendo com que o sinal do MC ficasse abaixo do limiar, sem contudo ter se livrado realmente da situação de risco, de forma que no instante seguinte o robô voltaria a se aproximar do obstáculo. Mesmo assim, o modelo atual aplicaria a premiação às sinapses dos neurônios do MC como se elas tivessem conduzido o robô a uma situação segura.

Obviamente os testes apresentados acima comprovam que este mecanismo, mesmo que carente de maior refinamento quanto à decisão de premiar as sinapses do MC, já apresenta bons resultados. Entretanto, como pôde ser observado nas simulações realizadas com o modelo de aprendizagem por recordação, em cada simulação um número muito grande de aprendizagens deste tipo são realizadas, muitas das quais de forma equivocada. O reconhecimento de situações relevantes de aplicação do mecanismo de aprendizagem proposto neste capítulo pode portanto, levar a resultados ainda mais expressivos.

Capítulo 7

Comportamentos Complexos em Ambientes Dinâmicos

7.1 Introdução

Até o presente momento, todos os desenvolvimentos realizados sobre o controlador de navegação autônomo do robô tiveram seus testes feitos sobre ambientes com obstáculos estáticos e onde apenas um robô era considerado a cada simulação. De fato, este método de avaliação de desempenho possibilitou a aquisição de resultados interessantes, que levaram à proposição de novas metodologias para o controlador autônomo. Entretanto, a simulação em ambientes com características dinâmicas, ou seja, sujeitos a mudanças topológicas no decorrer das simulações, abre novas perspectivas de avaliação.

Neste sentido, a possibilidade de se considerar múltiplos robôs navegando em um mesmo ambiente de simulação pode levar à observação de comportamentos complexos ainda não considerados em estágios anteriores do projeto, tendendo mesmo a novas propostas de aperfeiçoamento sobre a arquitetura já existente. Em um ambiente real, pode ser desejável, por exemplo, que vários robôs trabalhem ao mesmo tempo no desenvolvimento de uma tarefa, por vezes cooperando, outras vezes competindo entre si para a realização de seu trabalho.

Questões sobre os efeitos da divisão de uma tarefa global em sub-tarefas, sobre o comportamento dos indivíduos, bem como o estudo de comportamento social, não podem ser respondidas a partir de meras especulações. Mesmo o uso de simulações computacionais pode não ser suficiente para que respostas conclusivas para tais questões sejam encontradas. No entanto, as simulações podem muito bem indicar pistas sobre os caminhos a serem seguidos por uma implementação real.

Obviamente, a simples inclusão de diversos robôs em um mesmo espaço de trabalho não implicará que estes desenvolvam um comportamento social. De fato, este tipo de comportamento vem sendo amplamente estudado, principalmente em pesquisas ligadas à

área de Vida Artificial. Neste sentido, os trabalhos mais promissores relacionados ao tema baseiam-se no estudo de colônias de insetos (GRASSÈ, 1959). O interesse por este tipo de agrupamento social se deve ao elevado nível de organização e complexidade das tarefas realizadas em uma colônia, ao passo que cada indivíduo, observado separadamente, só é capaz de apresentar comportamentos simples.

Além do fato de que cada inseto representa um agente relativamente simples dentro da colônia, outro atrativo oferecido por estes “modelos” a projetistas de sistemas inteligentes corresponde ao grande número de estudos já realizados sobre o tema na biologia. THERAULAZ & BONABEAU (1999) apresentam uma breve história dos estudos realizados sobre estas colônias em biologia, enfatizando o trabalho de GRASSÈ (1959), que introduziu o termo *sinergia* para explicar como uma agregação de comportamentos simples resulta no comportamento complexo de uma colônia. De fato, insetos sociais não são capazes de sobreviver fora do contexto de sua colônia.

Grassè teve seu trabalho baseado na observação da construção de ninhos de cupins e observou que o ambiente de trabalho de um indivíduo se relacionava diretamente com seu comportamento. Neste sentido, modificações no ambiente resultariam posteriormente em modificações no comportamento dos indivíduos. Grassè observou que cada fase da construção do ninho dispara um comportamento, de forma que a conclusão de uma etapa da construção representa um estímulo para que uma nova fase seja iniciada.

Desta forma, o que se observa é a existência de uma comunicação indireta entre os indivíduos, membros da colônia, codificada na alteração realizada pelo indivíduo em seu ambiente, a qual servirá de estímulo para os demais indivíduos, dirigindo assim, de forma inconsciente, o plano global do ninho. Entretanto, é preciso reconhecer que insetos sociais possuem formas de reconhecimento do “eu” e do “outro”, não obstante o fato de o ambiente ter papel fundamental nas relações de trabalho.

KARSAI (1999) apresenta um estudo sobre a construção de ninhos de vespas *Polistes*, baseado nos princípios de sinergia apresentados por Grassè. Ressalta o autor algumas

diferenças básicas entre sistemas normalmente concebidos por mãos humanas e sistemas biológicos. Enquanto aqueles são hierárquicos e centralizados, estes são altamente distribuídos e robustos. Esta robustez se deve provavelmente à quantidade de redundâncias apresentadas nas unidades básicas de sistemas biológicos de natureza distribuída, unidades estas capazes de especialização, organização e desenvolvimento de comportamentos complexos.

Diversos trabalhos têm sido propostos na linha de sistemas baseados em propriedades de insetos sociais (THERAULAZ & BONABEAU, 1995; CAMPOS *et al.*, 2000) e modelos de presa-predador (NISHIMURA & IKEGAMI, 1997; NOLFI & FLOREANO, 1999; JIM & GILES, 2001) para simular comportamentos sociais em agentes artificiais. CASTELFRANCHI (1998) apresenta um trabalho voltado para a formalização de alguns fundamentos de relacionamentos sociais em agentes artificiais, descrevendo relações progressivamente mais complexas, desde ações simples orientadas a objetivos, passando por princípios de coordenação e delegação de tarefas, até organização e estrutura sociais. CASTELFRANCHI (2001) descreve e discute a teoria das funções sociais, e BURNS & GOMOLIŃSKA (2001) estudam mecanismos sócio-cognitivos à luz da teoria dos jogos.

No campo da robótica baseada em comportamento, MATARIC (2001) apresenta um estudo de desenvolvimento de mecanismos de cooperação em robôs móveis através da aplicação da teoria de aprendizagem por reforço em Processos de Decisão de Markov à arquitetura de robótica baseada em comportamentos. Os robôs recebem reforço por atingir objetivos e sub-objetivos e ainda por colaborarem na obtenção dos objetivos de outros robôs, através de compartilhamento de reforço. Segundo o autor, isto estimula o aparecimento de comportamento social. O trabalho é essencialmente conceitual e não apresenta resultados práticos, exceto aqueles revistos pela bibliografia.

Mataric apresenta também uma discussão sobre uma forma de aprendizagem muito interessante em sistemas de colaboração: a aprendizagem por observação, também dita aprendizagem por imitação. Neste tipo de aprendizagem, bem a propósito do que ocorre em uma aprendizagem supervisionada (e.g. treinamento supervisionado de uma rede neural),

um robô pode ser levado a aprender um comportamento através da simples observação das conseqüências da ação do outro (i.e. um outro robô ou mesmo um ser humano), ou através da imitação ativa das ações do professor.

HOLLAND & MELHUIH (1999) apresentam uma implementação do princípio da sinergia a um grupo de robôs reais, dotados de comportamento extremamente simplificado e sem capacidade de qualquer tipo de representação. O trabalho em grupo, dirigido por estímulos do ambiente, gera resultados comparáveis aos observados por GRASSÈ (1959) com os cupins construtores. É interessante ressaltar neste trabalho o baixo nível cognitivo dos agentes envolvidos, que não possuem qualquer mecanismo de reconhecimento do “eu” e do “outro”, como nos estudos envolvendo comportamentos sociais.

Conforme observa ARKIN (1998), a criação de times de robôs levanta uma série de expectativas quanto ao potencial que oferece. Dentre os pontos positivos pode-se considerar: melhoria de desempenho, caso a tarefa a ser realizada possa ser decomposta, permitindo a exploração do paralelismo oferecido por múltiplos agentes; capacitação para tarefas que seriam impossibilitadas considerando-se um único robô; ação distribuída, possibilitando trabalhos em diversos locais simultaneamente; e maior tolerância a falhas advinda da redundância promovida por múltiplos robôs.

No entanto, apesar dos pontos positivos, Arkin enumera alguns problemas que devem ser tratados cuidadosamente quando do advento de múltiplos robôs, dentre os quais: interferências na locomoção e mesmo colisões entre robôs; custo elevado dependendo do tipo de comunicação adotada entre os indivíduos; problemas de coordenação referentes à incerteza de um robô quanto às intenções dos demais; custo de implantação do time, advindo da aquisição e manutenção de um maior número de robôs.

A partir destas considerações, a inclusão de múltiplos robôs no sistema de simulação apresentado no capítulo 4 vem possibilitar uma ampla gama de considerações sobre comportamentos complexos envolvendo múltiplos robôs autônomos. No entanto, pelo fato de o sistema de controle autônomo então apresentado não permitir nenhuma forma de

comunicação explícita e ainda devido ao fato de um robô não ser capaz de reconhecer o outro como um indivíduo semelhante a si, deve se ter cuidado ao interpretar as simulações apresentadas neste capítulo sob o ponto de vista de relacionamentos sociais. Ainda assim, uma série de conclusões muito interessantes podem ser extraídas, principalmente em relação ao desempenho do sistema apresentado no capítulo 4.

7.2 Introdução de obstáculos móveis

A inclusão de obstáculos móveis no sistema de simulação descrito no Apêndice A é feita de forma parecida como a inclusão de obstáculos imóveis. Em princípio, todos os obstáculos inseridos no ambiente são imóveis. Após a inserção de um obstáculo no ambiente, este pode então ser animado, seguindo um dos quatro padrões de movimento permitidos pelo sistema. Desta forma, o movimento de um obstáculo pode ser: vertical, horizontal, aleatório ou atraído pelo robô.

Obstáculos com movimento vertical seguem uma trajetória periódica vertical. Obstáculos com movimento horizontal seguem uma trajetória periódica horizontal. Obstáculos com movimento aleatório também podem descrever trajetórias periódicas. Estes obstáculos estão sujeitos a mudanças aleatórias em sua direção e/ou sentido de movimento. Neste caso, a aleatoriedade das mudanças é determinada pelo usuário, e pode variar de 0% a 100%, correspondendo esta medida à probabilidade de ocorrer uma mudança de direção e/ou sentido do deslocamento do obstáculo a cada passo da simulação. Obstáculos com movimento atrativo são os mais interessantes, pois são capazes de simular o padrão de ataque de um predador. Este tipo de obstáculo oferece ao robô uma dificuldade extra para aproximação de um alvo.

Em todos os casos descritos anteriormente, os obstáculos têm seus movimentos restritos à área do ambiente, delimitada pelos demais objetos presentes e pelos extremos do mesmo. Isto significa que, ao colidir contra outros obstáculos, contra alvos, contra robôs ou contra as extremidades do ambiente, seu sentido ou mesmo sua direção de movimento são modificados. No caso dos movimentos vertical e horizontal, apenas o sentido do

movimento é invertido quando uma colisão ocorre. Já no caso dos movimentos aleatório e atrativo, a direção do movimento também é sujeita a alterações.

Quando um obstáculo é móvel, sua velocidade de deslocamento também pode ser determinada. A velocidade de deslocamento dos obstáculos pode assumir valores no intervalo [1,20], não seguindo a mesma escala que a velocidade de deslocamento dos robôs e não sendo expressa em unidades de deslocamento por iteração (ud/i), mas se referindo ao número de chamadas ao procedimento de movimentação de obstáculo em determinado intervalo de tempo e está sujeita a variações dependendo ao número de obstáculos móveis no ambiente. Na realidade, a escala aplicada ao aumento da velocidade segue uma progressão geométrica de base 2. Esta escala de variação permite a introdução de obstáculos mais rápidos ou mais lentos que os robôs do ambiente, aumentando o domínio de possibilidades de teste.

7.3 Simulações e testes

A seguir, são apresentadas duas séries de simulações em ambientes dinâmicos utilizando-se o sistema de controle autônomo descrito nos capítulos 3 e 4. A menos que seja explicitamente considerada outra configuração para o controlador, de forma geral assume-se para as simulações seguintes que o controlador apresenta controle de velocidade, que o MDO e o MBA sejam dirigidos pelos algoritmos 3.1 e 3.2, respectivamente, e que o mecanismo de aprendizagem por recordação não esteja em uso. Além disso, nenhum tipo de ruído é considerado nos sensores.

7.3.1 Simulações com obstáculos móveis

Nesta seção, é apresentada uma série de simulações envolvendo o controlador autônomo do robô, supondo um ambiente com obstáculos móveis e imóveis. O intuito destas simulações é a busca por uma habilidade ou por um comportamento ainda não verificado em outras simulações cujos ambientes lhe apresentassem menos chance de explorá-las. O que se pretende é sujeitar o robô a situações que exponham os limites de sua capacidade de navegação, de forma que suas limitações e potencialidades se tornem evidentes. Este estudo

pode levar a novas concepções de arquitetura, memória e aprendizagem para o modelo em questão.

Em todas as simulações a serem apresentadas com obstáculos móveis, assume-se um robô previamente treinado para a tarefa de navegação. Além disso, os obstáculos móveis são identificados por possuírem cor cinza, e sua forma de deslocamento pode ser deduzida através dos rótulos neles contidos, quais sejam: (H) para movimento horizontal, (V) para movimento vertical, (R) para movimento aleatório e (S) para movimento atraído pelo robô.

A Figura 7.1 apresenta duas destas simulações. Em ambas, utilizou-se obstáculos móveis com velocidade de deslocamento baixa, não ultrapassando a metade da velocidade de deslocamento do robô. Para estes casos, nota-se a habilidade do robô em desviar-se de seu curso de forma a escapar do obstáculo.



Figura 7.1 – Simulações com obstáculos móveis lentos

Para testar o desempenho do robô frente a situações mais complexas, foi realizada uma simulação em que a velocidade de deslocamento dos obstáculos móveis era maior que a metade da velocidade do robô. Três momentos desta simulação foram extraídos e compõem a Figura 7.2. O robô encontra grande dificuldade em transpor o primeiro obstáculo, colidindo várias vezes contra ele, e como pode ser observado em (a), tem sua passagem obstruída pelo segundo obstáculo. Neste caso consegue desviar-se do obstáculo até que sua passagem seja liberada em (b). Por fim, consegue transpor o terceiro obstáculo sem grandes dificuldades e atingir o alvo em (c).

Tanto na simulação da Figura 7.1, quanto naquela da Figura 7.2, o que se verifica é que o robô possui dificuldades em lidar com obstáculos móveis. Isto ocorre pois, ao longo das colisões que sofre, o controlador aprende a diminuir sua velocidade mediante a aproximação de um obstáculo. Isto é coerente com a linha de raciocínio apresentada no capítulo 4, que leva em consideração apenas obstáculos imóveis. No que se refere, porém, a um obstáculo móvel, o comportamento esperado do controlador é que aumente a velocidade do robô ao notar uma aproximação por iniciativa do obstáculo, reconhecendo o fato como uma perseguição. No entanto, com os mecanismos de memória que possui, o robô não é capaz de diferenciar um obstáculo móvel de um imóvel.

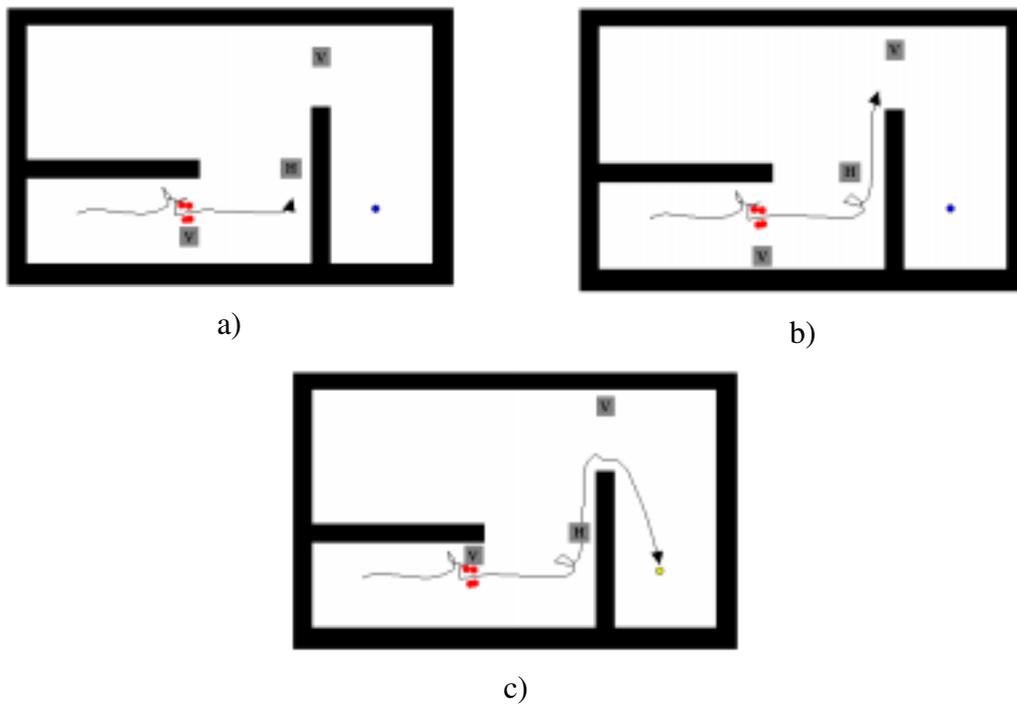


Figura 7.2 – Simulação com obstáculos móveis rápidos

Para dificultar ainda mais a tarefa do robô, construiu-se um ambiente com dois compartimentos e seis obstáculos com movimento aleatório. Novamente, a velocidade dos obstáculos é maior que a metade da velocidade do robô e o fato de o robô diminuir sua velocidade ao se aproximar de um obstáculo complica ainda mais sua situação. A aleatoriedade dos movimentos de cada obstáculo é de 10%, o que implica ser esta a probabilidade de mudança em sua direção e/ou sentido a cada movimento realizado.

A Figura 7.3 apresenta três situações da simulação supracitada. Em (a), nota-se que o robô ainda não havia colidido nem uma vez, mas que um dos obstáculos móveis obstrui sua passagem para o compartimento em que se encontra o alvo. Com isto, a colisão é inevitável. Após diversas colisões, o robô consegue empurrar o obstáculo móvel, abrindo caminho para o compartimento do alvo em (b), e a tarefa é concluída sem maiores problemas em (c).

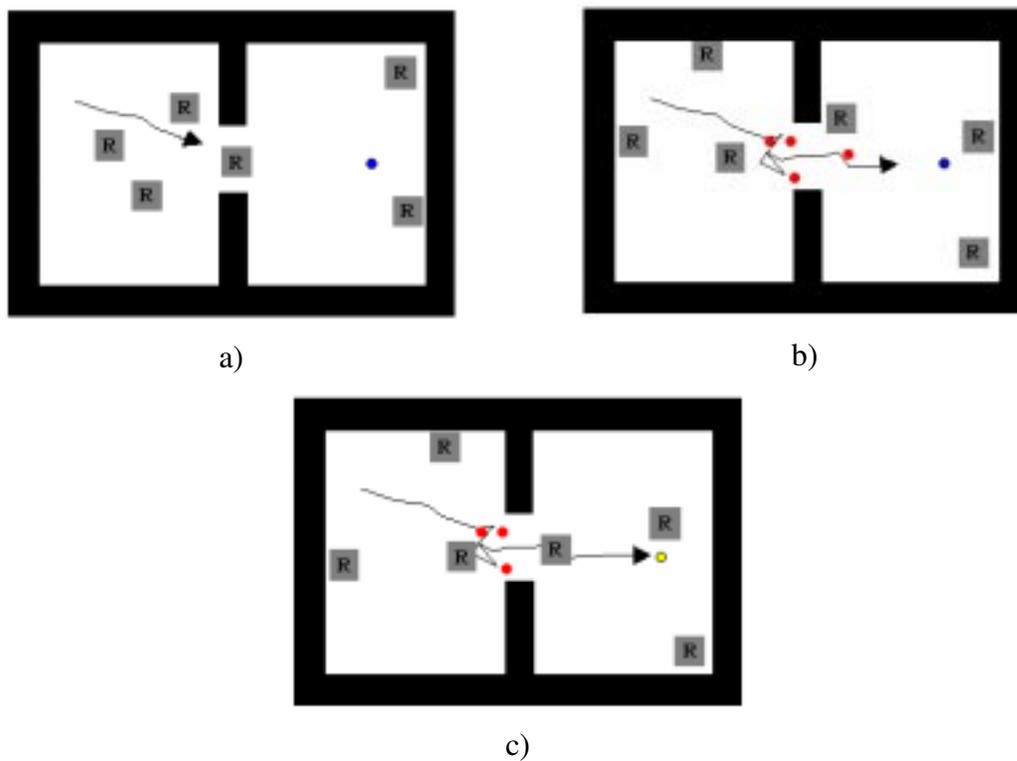


Figura 7.3 – Simulação com obstáculos de movimento aleatório rápido

A palavra “empurrar” é utilizada, pois o robô literalmente empurra o obstáculo a cada vez que colide, já que é característica dos obstáculos móveis mudarem sua direção e/ou sentido de deslocamento ao encontrarem alguma resistência no ambiente. Entretanto, o robô não adquire nenhum conhecimento quanto a esta sua potencial habilidade enquanto a realiza e nem é capaz de percebê-la como uma habilidade.

Embora o sistema de simulação utilizado não tenha sido projetado para implementar diretamente simulações que caracterizem uma relação presa-predador, com o auxílio de obstáculos móveis atraídos pelo robô pode-se esboçar este tipo de relação. Neste caso, o robô pode ser visto como presa, enquanto que o obstáculo móvel pode ser visto como predador. Esta interpretação é válida, pois se aplica apenas ao estudo de características de movimento do robô, não se estendendo a outros aspectos de comportamento social.

As duas simulações seguintes, com obstáculos atraídos pelo robô, exploram situações de perseguição. Agora, além de buscar o alvo, o robô deve fugir do obstáculo que o persegue. Na primeira simulação, a velocidade do obstáculo é maior que a do robô. As duas situações extraídas desta simulação (Figura 7.4) mostram como o robô não tem tempo suficiente para realizar uma manobra de desvio, o que é agravado pelo fato de ele reduzir sua velocidade quando da aproximação.

Desta forma, colide contra o obstáculo em (b). A partir de então, a cada colisão, sua velocidade se torna mais baixa diante do obstáculo, quando o desejável neste caso seria que a velocidade aumentasse, caracterizando uma manobra de fuga. Como isto não ocorre, o robô nunca consegue se livrar do obstáculo que o persegue, e por conseqüência, nunca atinge o alvo.

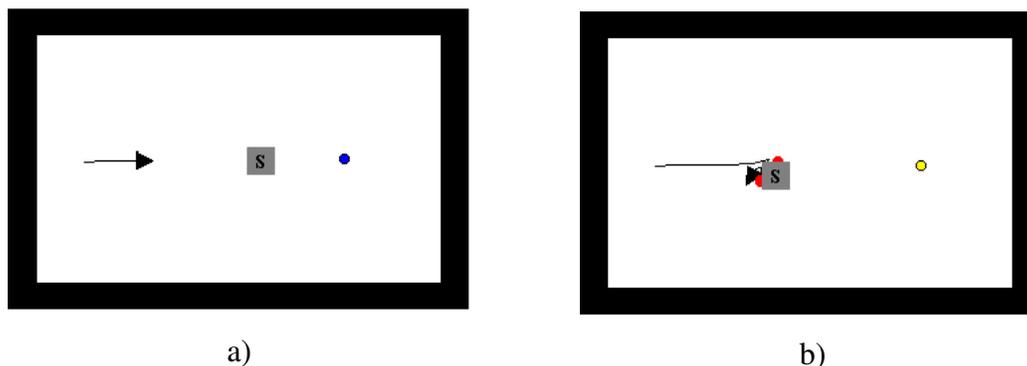


Figura 7.4 – Simulação com obstáculo rápido atraído pelo robô

Na segunda simulação com obstáculos móveis atraídos pelo robô (Figura 7.5), a velocidade de deslocamento dos obstáculos é menor que a metade da velocidade do robô. Com isto, contorno do obstáculo que se desloca em sua direção é facilmente realizado e o alvo é atingido sem problemas, mesmo diminuindo sua velocidade ao se aproximar do obstáculo. Isto pode ser observado nas três situações extraídas da simulação.

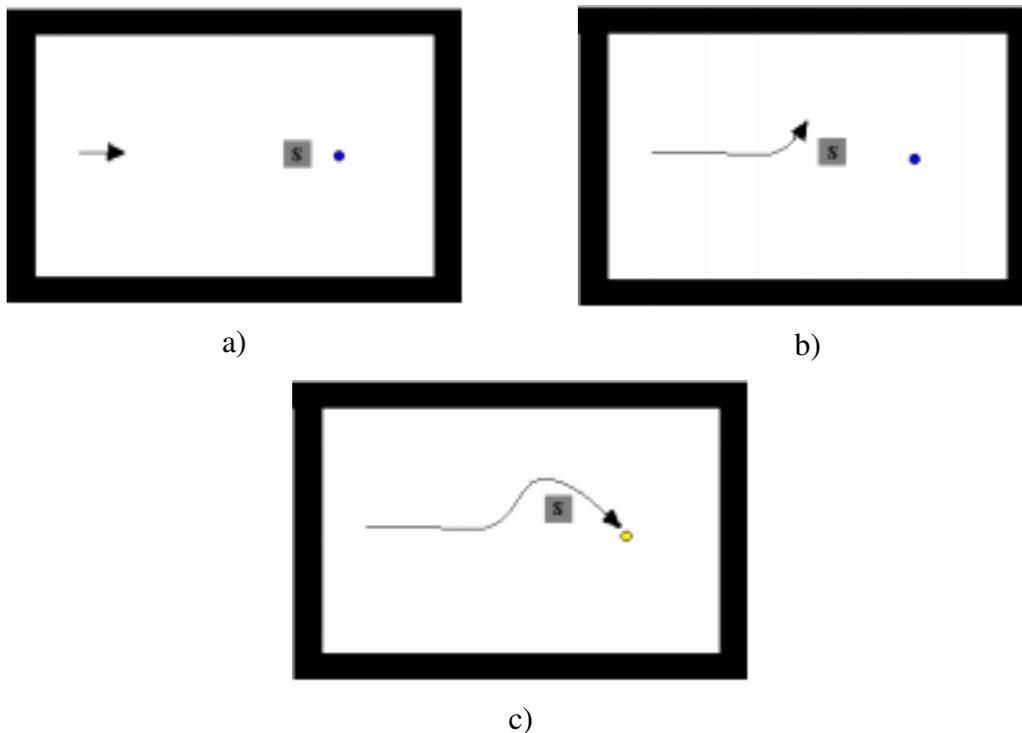


Figura 7.5 – Simulação com obstáculo lento, atraído pelo robô

A simulação seguinte utiliza obstáculos móveis atraídos pelo robô para a construção de um ambiente muito interessante e que sujeita o robô a situações que exigem capacidades básicas de previsão de manobras. Este ambiente, por ser peculiar, foi batizado de *prisão aberta*, dado que, apesar de o robô não conseguir escapar do ambiente, inicialmente sua passagem encontra-se aberta, e o fato de ele estar preso se deve apenas a suas próprias limitações, como é apresentado na Figura 7.6. Antes dos comentários relativos à simulação, uma breve descrição do ambiente e de suas características principais se faz necessária.

Observando a situação (a) da Figura 7.6, os quatro obstáculos internos são imóveis e constituem as paredes da prisão. Os quatro obstáculos externos, são obstáculos móveis

atraídos por robôs. Nesta simulação, sua velocidade é maior que a metade da velocidade do robô que se encontra inserido na prisão. Na parte externa da prisão, nos cantos do ambiente, três alvos são posicionados.

Além disso, um segundo robô é posicionado do lado de fora da prisão. Este segundo robô é responsável pela dinâmica da simulação, pois à medida que captura um alvo, faz com que outro alvo seja exibido, levando o robô que se encontra dentro do ambiente a dirigir-se para este novo alvo, o que promove uma modificação na configuração do ambiente conforme será descrito a seguir.

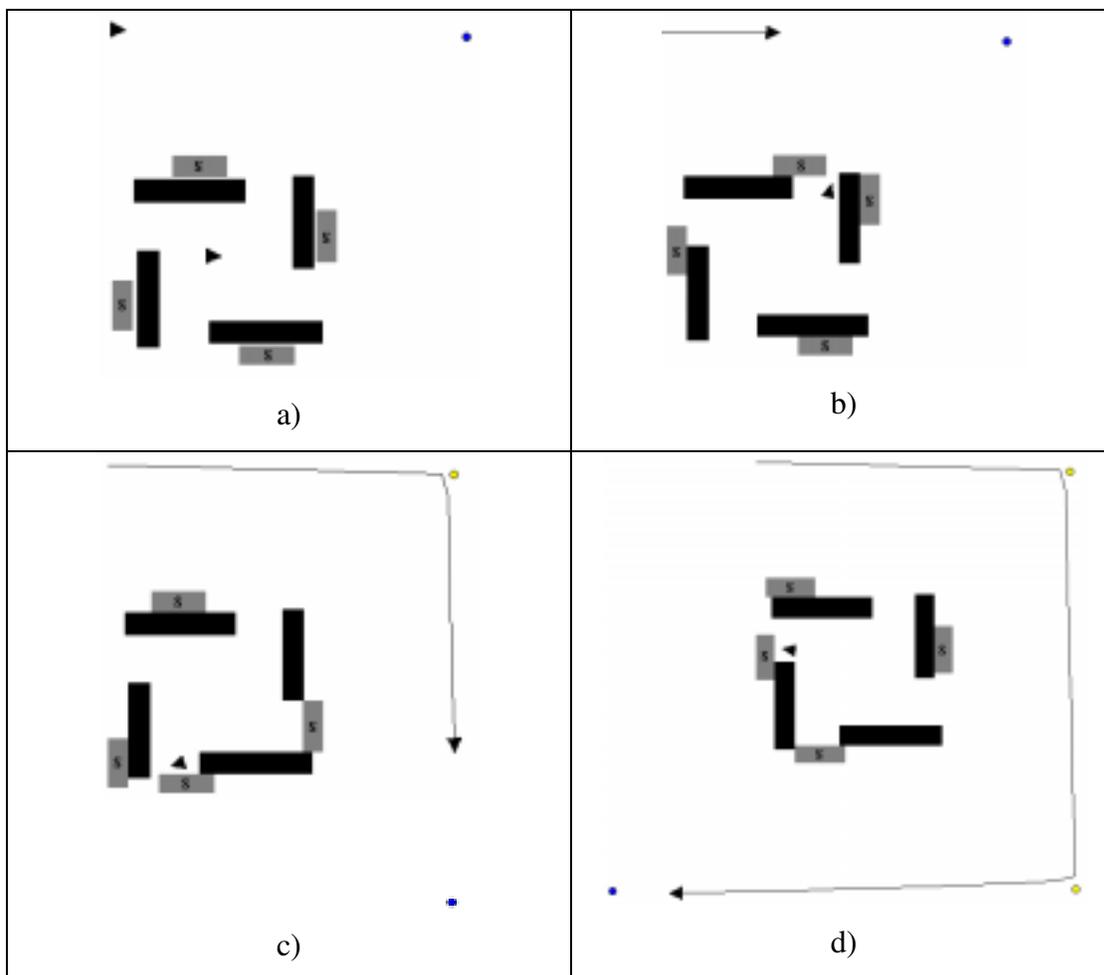


Figura 7.6 – Simulação no ambiente *prisão-aberta*

Uma característica dos obstáculos atraídos por robôs, é que sua atração se dá apenas pelo robô mais próximo a eles. Logo, para não interferir nos movimentos destes obstáculos,

tanto os alvos quanto o robô externo à prisão são posicionados a uma distância maior que a máxima distância que o robô interno possa apresentar com relação a qualquer dos obstáculos móveis.

Antes do início da simulação (a), todas as portas da prisão estão abertas. Quando a simulação começa, os dois robôs se dirigem na direção do alvo. Entretanto, conforme o robô interno à prisão se aproxima de uma das saídas, o obstáculo móvel por ele atraído se transforma em uma porta que se fecha, impedindo sua passagem, de forma que ele deve retroceder (b). Por medida de clareza, nem a trajetória, nem as colisões do robô interno são apresentadas, embora algumas colisões tenham ocorrido. Após a captura do primeiro alvo pelo robô externo, o robô interno também é guiado para o alvo seguinte (c). Novamente, os obstáculos por ele atraídos se transformam em portas que impedem sua passagem, o mesmo acontecendo com relação ao último alvo (d).

7.3.2 Simulações com múltiplos robôs

A Figura 7.7 apresenta uma simulação em que quatro robôs previamente treinados em conjunto foram colocados em um mesmo ambiente de navegação. Foram utilizados 20 alvos para o treinamento e, embora submetidos às mesmas condições iniciais, o número de neurônios de cada robô ao final do treinamento foi diferente, variando de dezesseis a vinte e dois. Esta diferença pode estar ligada a questões tão sutis quanto à posição de um robô com relação aos demais no momento em que um novo alvo é apresentado como pode também refletir colisões entre dois robôs, dado que cada robô representa um obstáculo aos demais.

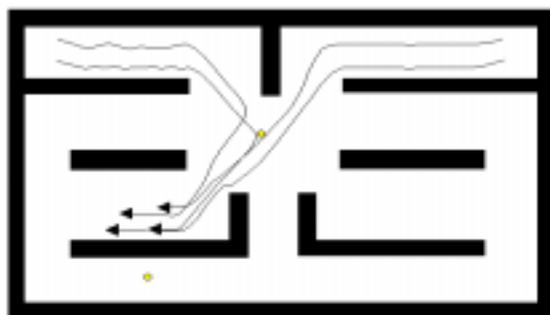


Figura 7.7 – Múltiplos robôs navegando em formação

Simulações como a apresentada na Figura 7.7 se tornam mais interessantes quando os robôs têm controle de suas velocidades. Caso contrário, sua habilidade de manobras é reduzida, acarretando um número indesejado de colisões entre os robôs após a fase de treinamento, o que complicaria a navegação. Para ilustrar este fato, a Figura 7.8 apresenta duas simulações em um ambiente contendo quatro robôs cujas velocidades de deslocamento são constantes e iguais a 0,6 ud./i.

A situação (a) representa a fase de treinamento do MC. Os quatro robôs são posicionados inicialmente, dois a dois, nos lados esquerdo e direito do ambiente, próximos às áreas onde as colisões são observadas. Nesta fase de aprendizado, as colisões de todos os robôs somam dezenove, sendo que a partir de então, nenhuma outra colisão é observada no ambiente e os robôs passam a desenvolver trajetórias cíclicas em torno do ambiente sem conseguir atravessar as passagens que os separam do alvo. Estas passagens possuem largura de 20 ud. cada, sendo que os robôs possuem largura de 16 ud. cada.

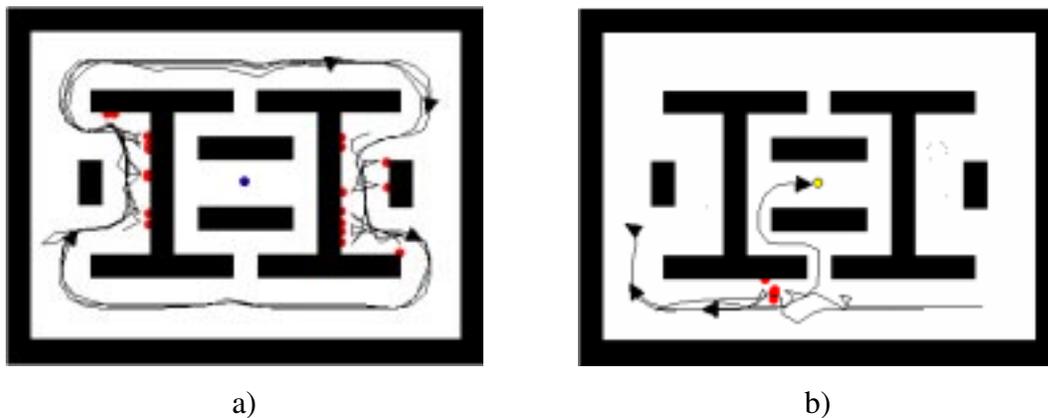


Figura 7.8 – Robôs com velocidades constantes iguais a 0,6 ud./i.: a) período de aprendizagem com 19 colisões; b) colisões entre robôs treinados

Em uma segunda simulação (situação (b)), os mesmos quatro robôs treinados na situação (a) são inicialmente agrupados dois a dois em fila indiana, sendo que uma fila é posta de frente para a outra e os robôs de cada fila se locomovem em direções opostas. Como pode ser observado, o fato de os robôs estarem treinados não evita que algumas colisões ocorram entre eles. Outro fato interessante é que, uma vez pressionado pelos demais, um dos robôs é

empurrado para a passagem que dá acesso ao alvo e consegue atingi-lo, mesmo que de forma ocasional.

Diferentemente da Figura 7.8, em que os robôs possuem velocidades constantes, na Figura 7.9 os robôs são capazes de controlar suas próprias velocidades. Para verificar a eficácia deste controle do sistema frente ao caso em que tal controle está ausente, o mesmo ambiente e as mesmas condições iniciais da Figura 7.8 foram observados em ambas as situações apresentadas agora. Assim, o que se observa em (a) é o período de aprendizagem dos quatro robôs.

Observa-se que conforme dito no capítulo 4, o controle de velocidade envolve um número maior de conhecimento. Neste caso, as colisões de todos os robôs para que as estratégias de coordenação fossem desenvolvidas, quando somadas, totalizam setenta e três, um número muito maior quando comparado ao da simulação com velocidade constante. Por outro lado um dos robôs consegue atravessar as passagens estreitas e atingir o alvo no momento em que um segundo robô já entrava pela mesma passagem.

Além disso, o que se observa na situação (b), em que os robôs já se encontram treinados, é que embora naveguem em direções opostas, conseguem controlar com sucesso suas velocidades de forma que nenhuma colisão entre eles ocorra, o que confirma o que foi dito anteriormente sobre os benefícios do controle de velocidade sobre navegações com múltiplos robôs. Deve-se reparar também que, embora o número de colisões para a aprendizagem seja maior, estas tendem a praticamente desaparecer após esta fase, ao passo que as colisões entre robôs observadas na Figura 7.8 (b) tendem a ocorrer indefinidamente, sempre que os robôs estiverem em rota de colisão.

Um outro ponto, que já foi observado nas simulações apresentadas, é que até o momento não há nenhum mecanismo explícito nos robôs de reconhecimento mútuo para apresentação de comportamentos de colaboração durante a navegação. O sistema de controle autônomo utilizado nas simulações deste capítulo é o mesmo apresentado e descrito nos capítulos 3 e

4. Entretanto, há indícios de que o processo de aprendizagem se torne mais eficaz quando diversos robôs são treinados simultaneamente em um mesmo ambiente.

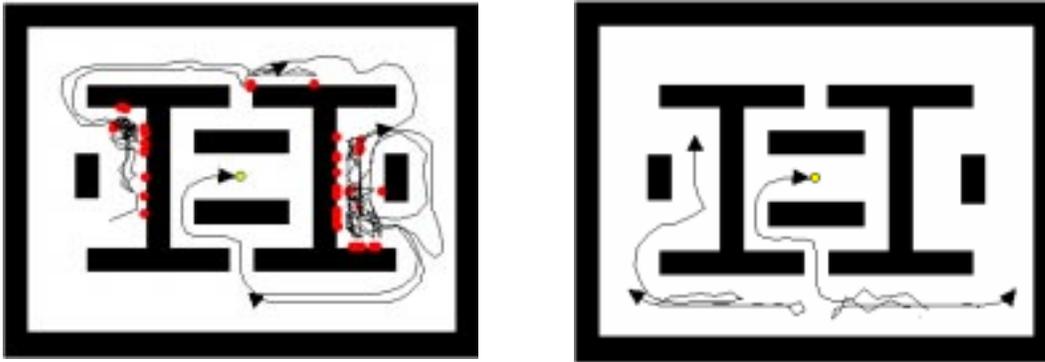


Figura 7.9 – Robôs com velocidades variáveis: a) período de aprendizagem com 73 colisões; b) navegação sem colisões entre robôs treinados

Uma razão para isto é que quando colidem repetidas vezes entre si, os robôs aceleram o processo de treinamento. Além disso, como todos se movimentam na direção do alvo, acabam exercendo o papel de guias uns dos outros. Embora sejam competidores na busca do alvo, colaboram indiretamente para se manterem no caminho certo. Retomando à Figura 7.7, pode-se observar os robôs caminhando em formação rumo ao alvo. De certo modo, esta formação oferece benefícios a todos os seus membros, embora não haja nenhuma intenção explícita para tal.

Colaboração não explícita pode ser encontrada em diversos sistemas biológicos. Em particular, processos de co-evolução natural fornecem exemplos interessantes deste tipo de efeito. DAWKINS (1998) apresenta o caso das vespas e dos figos. Dentre todas as espécies de vespas de figo, cada uma possui sua própria espécie de figo, a qual poliniza e da qual extrai seu alimento. Ambas as espécies (vespa e figo) evoluíram e se especializaram através de mutações e seleções e terminaram por colaborarem mutuamente de forma a dependerem uma da outra para a sobrevivência, mas sem que jamais houvesse uma pretensão explícita do figo em alimentar a vespa ou uma pretensão explícita da vespa em permitir a procriação do figo.

Além da Figura 7.7, outra demonstração de como os robôs colaboram indiretamente na obtenção de seus objetivos individuais está presente na Figura 7.10, que compara a navegação com múltiplos robôs e a navegação com um único robô. O controlador apresentado no capítulo 4 é capaz de superar máximos locais¹ que não apresentem um grau de dificuldade muito elevado, conforme pode ser observado na situação (a), em que um robô inicialmente desprovido de treinamento consegue, após algumas colisões, contornar o beco sem saída e atingir o alvo.

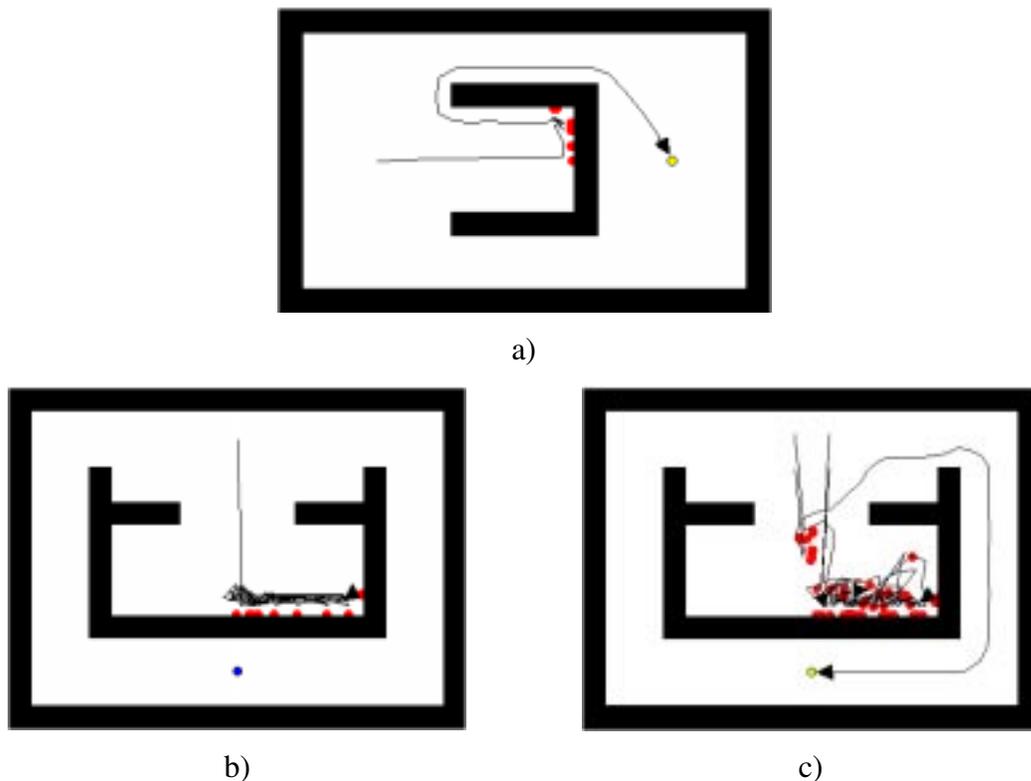


Figura 7.10 – Desempenhos do sistema frente a mínimos locais: a) sucesso com um robô; b) fracasso com um robô; c) múltiplos robôs dos quais um obtém sucesso

No entanto, quando a dificuldade do máximo local se torna demasiadamente desafiadora, o sistema, por ser reativo, fica preso pelo obstáculo sem poder avançar em uma direção alternativa que o levaria para o alvo, o que pode ser observado na situação (b) com um robô

¹ Por máximo local entende-se, neste trabalho, o ponto do ambiente mais próximo do objetivo do robô que pode ser atingido por uma estratégia de maximização gradiente, mas que ainda não é o objetivo do robô.

inicialmente sem qualquer treinamento. Por outro lado, quando quatro robôs são colocados no mesmo ambiente da situação (b) e sob as mesmas condições iniciais, o resultado é outro. Da mesma forma como na Figura 7.8 (b), em que um dos robôs é pressionado pelos demais para dentro da passagem estreita, na Figura 7.10 (c) um dos robôs é pressionado para fora do compartimento que prende os demais e consegue atingir o alvo.

Diante desta situação, parece muito interessante capacitar os robôs com a habilidade de observar o comportamento de seus semelhantes e aprender a imitar suas ações. Caso este mecanismo estivesse implementado na situação (c) da Figura 7.10, os demais robôs que ficaram presos pelo mínimo local poderiam tentar seguir o robô que conseguiu escapar, e eventualmente todos os robôs poderiam escapar do compartimento. A situação (c) atenta também para o grande número de colisões entre os robôs durante a fase de treinamento.

Por fim, na Figura 7.11, um número maior de robôs (seis) são dispostos no mesmo ambiente utilizado nas simulações da Figura 7.10 (b) e (c), sendo que desta vez os robôs já haviam sido previamente treinados em conjunto. Algumas poucas colisões esporádicas entre os robôs devido às dimensões reduzidas do ambiente não são apresentadas para facilitar a visualização. Em (a), pode-se observar o momento em que o primeiro robô a escapar do compartimento é empurrado para fora. Em (b), este robô já pode ser observado fora do compartimento e um segundo robô segue o mesmo caminho.

Nas simulações em (c) e em (d), as trajetórias descritas pelos robôs também são ocultadas para facilitar a visualização. Em (c), os dois primeiros robôs a escaparem do mínimo local já se aproximam do alvo e um terceiro robô pode ser observado sendo pressionado para fora do mínimo local. Em (d), o primeiro robô a escapar da armadilha atinge o alvo. Neste ponto, o terceiro robô a escapar da armadilha já se aventurava para fora do ambiente.

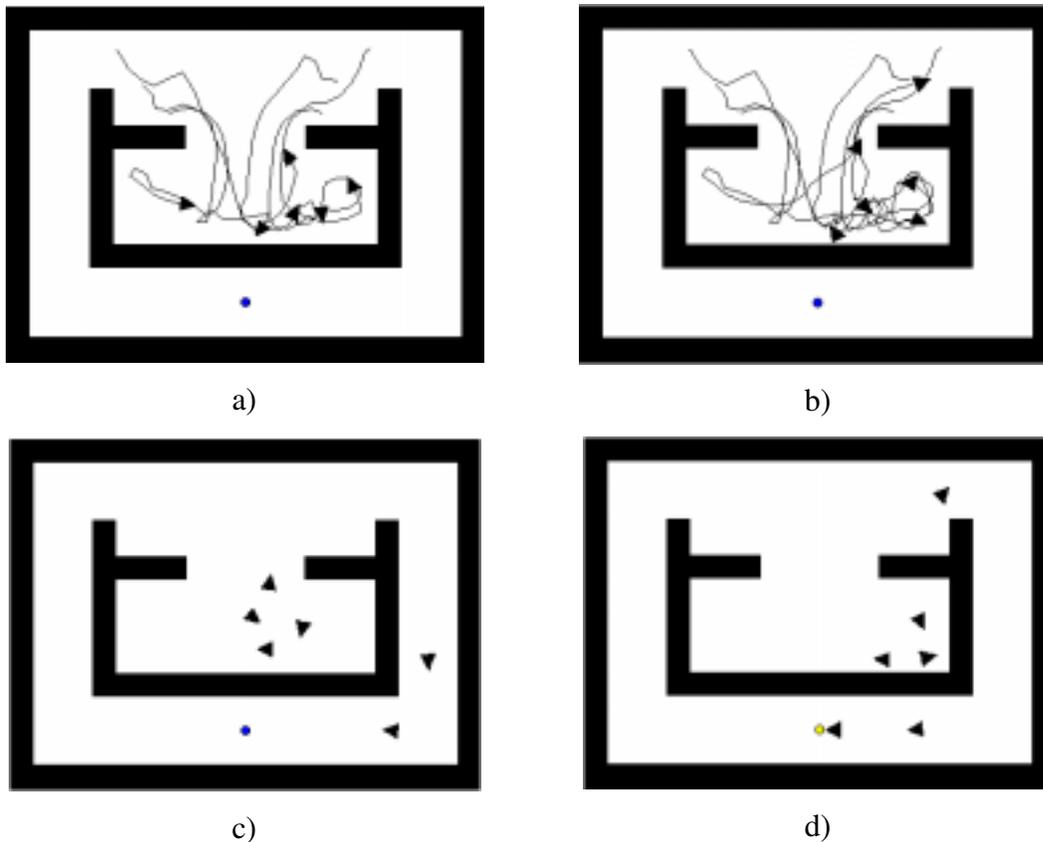


Figura 7.11 – Simulação com seis robôs em ambiente com mínimo local do qual três conseguem escapar

7.4 Visão sintética do capítulo 7

A partir das simulações realizadas com obstáculos móveis, conclui-se que o robô não consegue reconhecer uma situação em que o obstáculo, por vezes, aja deliberadamente com o sentido de evitar que o alvo seja atingido. Isto porque todo o treinamento modelado nos capítulos 3 e 4 leva em consideração apenas obstáculos estáticos. Os casos de simulações com múltiplos robôs são diferentes, pois embora cada robô seja um obstáculo móvel para os demais, não há nenhuma relação intencional por parte dos robôs em prejudicar o desempenho dos demais. Se bem que esta seria uma característica muito interessante, se desenvolvida.

O grande problema verificado em todas as simulações com obstáculos móveis é que o robô diminui sua velocidade ao se aproximar de um obstáculo. Uma solução para este problema seria desenvolver nos robôs, através de aprendizagem, um comportamento de fuga em que

a velocidade de deslocamento se tornasse maior quando uma aproximação lateral ou traseira fosse detectada. Entretanto, não basta que um obstáculo seja notado através dos sensores destas regiões do robô. É preciso ainda que o controlador seja capaz de estimar a ocorrência de movimento realizado pelo obstáculo.

Com este tipo de habilidade, o robô poderia, uma vez tendo reconhecido uma ameaça móvel, diminuir sua velocidade, como já é capaz de fazer, evitando uma colisão frontal, e acelerando quando é perseguido. Além disso, em situações em que os obstáculos não possuam o intuito de perseguir o robô, como no caso dos obstáculos com movimento vertical, horizontal e aleatório, mas também como no caso do problema da prisão aberta, muitas vezes é preciso estimar o momento exato de aumentar a velocidade de forma a aproveitar as oportunidades de transposição. Também este tipo de problema poderia ser resolvido com um comportamento como o descrito no parágrafo anterior.

O grande problema em se desenvolver este tipo de comportamento consiste na necessidade de se reconhecer movimentos, o qual pode ser realizado através da utilização de técnicas analíticas. Entretanto, a abordagem utilizada até aqui na concepção do controlador autônomo de navegação compreende o desenvolvimento de comportamentos emergentes complexos a partir da interação de estruturas relativamente simples. Esta abordagem parece concordar com a organização observada em sistemas biológicos. Com isto, ao invés do desenvolvimento explícito de técnicas de reconhecimento de movimento, uma alternativa mais interessante e mais robusta seria a concepção de uma memória de curta duração, capaz de extração de padrões de movimento.

A utilização de uma memória levanta uma questão muito criticada por pesquisadores adeptos à robótica baseada em comportamentos, que geralmente vêm com maus olhos algumas formas de representação do conhecimento. Não obstante, os resultados apresentados pelas simulações anteriores deixam claro que um modelo puramente reativo e baseado em comportamentos está restrito a tarefas simples e que não contemplam todas as possibilidades verificadas em ambientes naturais.

Algumas vantagens de sistemas reativos frente aos modelos de inferência simbólica, fundamentados nas teorias da Inteligência Artificial Clássica, têm sido amplamente apresentadas desde que os primeiros modelos começaram a aparecer com mais ênfase, em meados dos anos 80. Entretanto, negar o poder de formas de representação do conhecimento diretamente relacionadas ao ambiente de ação do agente é negar em parte a base biológica que inspirou a própria abordagem comportamental em robótica, uma vez que a maioria dos organismos biológicos evoluídos apresenta algum mecanismo de representação de conhecimento.

As simulações com múltiplos robôs superaram as expectativas anteriores, por apresentarem a emergência de comportamentos que não haviam sido previstos na arquitetura do controlador, como: diminuição no tempo de aprendizagem de cada robô, e por conseqüência, diminuição do tempo de captura dos alvos; formação de agrupamentos que lembram padrões de ataque organizados, como os que ocorrem em algumas espécies de animais predadores.

Mais ainda, foi apresentado em simulações que o desempenho de navegações com múltiplos robôs é melhorado quando se considera o controle de velocidade proposto no capítulo 4 e que em ambientes com máximos locais, a colaboração indireta dos robôs pode fazer com que um dos indivíduos consiga superar a dificuldade, fato improvável para um único indivíduo. Neste caso, a implementação de mecanismos para reconhecimento e imitação de atitudes entre os robôs se torna interessante.

Capítulo 8

Considerações Finais

8.1 Recapitulação geral e considerações adicionais

Ao longo dos capítulos que compõem este texto, um modelo de controlador autônomo modular e hierárquico baseado em redes neurais nebulosas foi apresentado e descrito sob diversos aspectos. O trabalho aqui apresentado consiste de uma extensão daquele proposto por FIGUEIREDO (1997) e o desenvolvimento desta extensão foi motivado, em parte, por limitações encontradas no modelo de FIGUEIREDO (1997), e em parte por perspectivas de resultados promissores advindas da observação da fisiologia de sistemas biológicos. Estas perspectivas acabaram por se confirmar, como atestam os resultados expressos pelas simulações que foram apresentadas.

Assim, visto que o modelo de FIGUEIREDO (1997) não possuía um controle dedicado à velocidade de deslocamento do robô, este controle foi acrescentado à estrutura do controlador e proposto no capítulo 4. Também uma visão evolutiva de síntese dos módulos inatos de busca de alvos e desvio de obstáculos foram apresentadas no capítulo 5, a qual evidenciou a robustez proporcionada pelos algoritmos genéticos. Por outro lado, sua utilização evidenciou também a dificuldade na realização da atividade de atribuição de créditos de maneira satisfatória aos cromossomos da população, quando se trata de avaliar a navegação de um robô móvel.

O mecanismo de Aprendizagem por Recordação, descrito no capítulo 6, foi proposto com base na capacidade que sistemas biológicos evoluídos possuem de reforçarem suas crenças sobre determinado conceito simplesmente através da lembrança de sua ocorrência, sem a necessidade da observação ou presença física do objeto ou conceito que se esteja recordando. Simulações com o modelo de Aprendizagem por Recordação apontaram para um melhor desempenho no que se refere ao número de colisões necessárias para que o robô aprenda a coordenar adequadamente seus movimentos.

Além da motivação biológica, a Aprendizagem por Recordação complementa em parte o processo de aprendizagem que ocorre após um alvo ser atingido. Uma vez que o robô deve sempre buscar novos alvos no ambiente, o processo de aprendizagem disparado pelo fato de se atingir um alvo se repete durante toda a navegação, em detrimento da aprendizagem após uma colisão, que cessa tão logo o robô aprenda a coordenar seus movimentos de forma a não mais colidir contra obstáculos. Esta desigualdade faz com que após diversos alvos atingidos, o MC passe a dar mais importância à busca de alvos que ao desvio de obstáculos. Este problema foi apresentado no capítulo 3 e parcialmente solucionado com a introdução da propriedade sináptica VAAS.

Entretanto, o VAAS não elimina a causa do problema, que é a saturação dos neurônios do MC que ponderam o sinal de busca de alvos. O que o VAAS faz é tratar o problema, uma vez identificado, mas não é capaz de identificá-lo antes da ocorrência de sua consequência, que é a colisão. O mecanismo de Aprendizagem por Recordação, a exemplo da aprendizagem após um alvo ser atingido, também permanece atuante no MC mesmo quando as colisões já não mais ocorrerem. Desta forma, o reforço nos neurônios de ponderação do sinal de busca de alvos também é seguido pelo reforço nos neurônios de ponderação do sinal de desvio de obstáculos.

Esta forma de reforço de ambos os lados pode, no entanto, ter ainda um aspecto negativo. Conforme visto no capítulo 2, quando algumas considerações sobre reforço e punição em sistemas biológicos foi feita, observou-se que em alguns casos o excesso de reforço pode levar a uma saturação e, nestas condições, o reforço pode mesmo agir como uma punição. No caso do excesso de aprendizagem mediante alvos atingidos, as consequências negativas foram apresentadas. No caso do excesso de reforço das sinapses dos neurônios de ponderação do sinal de desvio, a consequência negativa é o robô se tornar excessivamente cauteloso e evitar atravessar portas ou corredores, o que pode comprometer seu desempenho ou impedir o atendimento de objetivos.

Embora estes casos de saturação não tenham sido tratados no âmbito deste trabalho, uma possível solução para o problema é reduzir progressivamente o reforço ao longo do tempo,

terminando por extingui-lo completamente. Esta solução, embora careça de comprovação prática, parece ser uma boa alternativa. Seguindo o que foi dito sobre reforço e punição no capítulo 2, a privação prolongada do reforço evidencia sua importância. Esta característica pode ser utilizada, por exemplo, na concepção de um mecanismo de enfraquecimento das sinapses menos utilizadas em determinado momento da navegação, de forma a enfatizar a importância de um eventual reforço nas mesmas quando forem novamente necessárias.

Por fim, o capítulo 7 não teve como meta a agregação de novas extensões ao modelo de controlador apresentado nos capítulos 3 e 4, mas sim testar este modelo em ambientes que apresentassem características dinâmicas, como obstáculos móveis e diversos robôs navegando simultaneamente em um mesmo ambiente. Neste caso, o principal interesse do capítulo se concentrou na identificação das virtudes e deficiências do modelo frente a condições mais árduas de navegação.

Quanto à inclusão de obstáculos móveis, ficou evidente a necessidade de uma adaptação no mecanismo de controle de velocidade do robô. Este mecanismo foi originalmente proposto para navegações em ambientes estáticos e, por isso, o robô aprende a diminuir sua velocidade ao se aproximar de obstáculos. No caso de obstáculos móveis, que podem inclusive se deslocar na direção do robô, seria necessário um comportamento de fuga. Entretanto, a implementação deste comportamento requer do robô a habilidade de reconhecer quando fugir e quando diminuir a velocidade diante de um obstáculo, o que implica alguma forma de identificar movimentos e que, por sua vez, implica a capacidade de se estabelecer relações espaço-temporais com o ambiente.

No que se refere a múltiplos robôs, observou-se que a navegação coletiva pode ajudar na superação de algumas dificuldades impostas pelo ambiente. A implementação de um modelo de percepção e comunicação entre robôs tornaria a tarefa de comunicação conjunta muito mais interessante, pois abriria espaço para o estudo de comportamentos coletivos coordenados. Um exemplo deste tipo de comportamento se dá através de manifestações sociais. Entretanto, este tipo de comportamento requer determinadas especializações por parte dos membros da sociedade e, neste sentido, o controlador apresentado neste texto

ainda carece de desenvolvimento, uma vez que nenhum dos robôs apresentados nas simulações do capítulo 7 possuíam especialização, pois foram implementados independentemente para desempenhar tarefas equivalentes.

Tanto no que se refere ao reconhecimento de movimento em um obstáculo, quanto ao estabelecimento de comunicação e especialização entre robôs, é possível que soluções possam ser implementadas através de modelos matemáticos sem engajamento biológico. No entanto, a exemplo do que foi feito com o MDO e com o MBA no capítulo 5, este trabalho se preocupa em manter uma aproximação com características de sistemas biológicos conexionistas por entender que estes sistemas são fontes seguras de inspiração para a proposição de sistemas complexos artificiais autônomos.

8.2 Autonomia

É sabido que a teoria de controle clássico exhibe limitações no tratamento de problemas de controle autônomo, definidos em ambientes complexos e caracterizados por elevados graus de incerteza (ANTSAKLIS *et al.*, 1996; FIGUEIREDO, 1999). Além disso, segundo SMITHERS (1997), embora o termo autonomia venha sendo empregado amplamente na literatura com os mais diversos significados, o simples fato de um robô possuir capacidade de locomoção sem a presença de um piloto não faz dele uma máquina autônoma.

Ao contrário, na visão de autores como SMITHERS (1997), FIGUEIREDO (1999) e STEELS (1995), este tipo de comportamento caracteriza apenas o controle automático, sendo que um comportamento efetivamente autônomo está relacionado diretamente à aquisição, por parte do próprio agente, do conhecimento necessário para a construção de seu comportamento. Neste sentido, para SMITHERS (1997), a introdução de conhecimento em um determinado sistema por meio de uma fonte externa implica perda de autonomia. Com isto, segundo esta visão, autonomia está diretamente ligada ao sentido de identidade do agente autônomo, que pode ser distinguido dentre outros semelhantes. A capacidade de possuir propósitos, ou motivações, também é típica de sistemas autônomos. No caso do sistema apresentado neste trabalho, a busca por alvos e o desvio de obstáculos caracterizam propósitos manifestos do modelo.

FIGUEIREDO (1999) entende ser possível a determinação de conhecimento a priori em um agente sem que sua autonomia seja prejudicada. De fato, em sistemas biológicos evoluídos, a filogênese se encarrega de trazer para um indivíduo de determinada espécie uma carga de características físicas e comportamentais particulares àquela espécie. Entretanto, o grau de liberdade para o aprendizado de novos comportamentos, a partir de interações com o ambiente, varia bastante de espécie para espécie.

Há espécies em que o indivíduo está fortemente limitado a desenvolver comportamentos inatos (e.g. insetos) tendo pouca capacidade de aprendizado. Entretanto, em outras espécies, a influência de seus comportamentos inatos é notada em menor grau no indivíduo, sendo-lhe dada maior margem para o aprendizado de novos comportamentos (e.g. seres humanos). Pode-se dizer que tanto os insetos quanto seres humanos são organismos essencialmente autônomos. Entretanto, FIGUEIREDO (1999) também entende que a presença de comportamentos inatos, agindo em diferentes graus para diferentes espécies, torna plausível a argumentação de que sistemas não possam ser diferenciados claramente entre autônomos e não-autônomos, mas que graus de autonomia devam ser considerados. Neste sentido, por exemplo, insetos sociais como formigas ou abelhas têm menos autonomia para sobreviverem individualmente na natureza que insetos não-sociais como a mosca.

Por fim, uma característica negativa apontada por FIGUEIREDO (1999) com relação à autonomia, apesar de seus muitos pontos positivos, é que ela torna inevitável a ocorrência de tentativa e erro, na medida em que boa parte das capacidades que podem ser desenvolvidas por um sistema autônomo (natural ou artificial) é a ele incorporada por aprendizagem mediante sua interação com o meio. Neste sentido, sistemas de controle clássicos, projetados a partir da definição precisa da tarefa a ser executada, podem ser definidos para atender seus objetivos com custo garantido. No entanto, perdem a capacidade de adaptação a mudanças nas condições iniciais de projeto.

De tudo o que foi considerado acima, pode-se dizer que o controlador proposto ao longo deste trabalho apresenta muitas das características de um sistema autônomo. Da forma

como foram apresentados no capítulo 5, o Módulo de Desvio de Obstáculos e o Módulo de Busca de Alvos, responsáveis por comportamentos inatos, foram sintetizados de forma automática. O Módulo de Coordenação, por sua vez, é capaz de adquirir suas próprias estratégias de coordenação de movimentos através de aprendizagem. As estruturas que compõem estes módulos (i.e. topologias das redes neurais nebulosas) podem ser consideradas parte de uma informação inata herdada, que inclui as características físicas dos sensores e dos atuadores e a forma do robô.

Através das simulações apresentadas ao longo dos diversos capítulos, pôde-se notar que o controlador autônomo proposto é dotado de adaptação frente a ambientes e situações desconhecidos, e que por isto está sujeito a eventuais e inevitáveis falhas, representadas por colisões contra obstáculos quando o conhecimento adquirido pelo sistema ao longo de suas interações com o ambiente não permite tratar convenientemente uma destas situações desconhecidas. Contudo, apesar da ocorrência de colisões, associadas a processos de tentativa e erro, o sistema na grande maioria das vezes suporta bem condições adversas de navegação, como foi notado ao se introduzir ruído nos sensores no robô nos capítulos 4 e 5.

8.3 Aprendizagem por reforço

A forma como o Módulo de Coordenação aprende a ponderar de maneira coerente os comportamentos de desvio de obstáculos e de busca de alvos se dá por meio de mecanismos de aprendizagem por reforço. Segundo esta técnica de aprendizagem, não há a figura do supervisor, que avalia quão correta foi a ação do sistema frente a uma determinada situação no ambiente. Neste caso, a maneira como as próprias necessidades do sistema são satisfeitas em cada passo do processo de interação com o ambiente fornece os indícios necessários para a avaliação de seu desempenho, levando à punição ou ao reforço de um determinado comportamento.

Contudo, apesar de apresentar estas características, a forma de implementação do mecanismo de aprendizagem do MC difere sob diversos aspectos da teoria de aprendizagem por reforço tradicional, vinculada a Processos de Decisão e Markov (PDM) e

baseados em processos de busca heurística (SUTTON & BARTO, 1998), como foi apresentado no capítulo 2.

Diversos modelos de controladores para navegação de robôs móveis seguindo técnicas de aprendizagem por reforço deste gênero têm sido propostos na literatura. Nestes controladores, os estados do sistema representam o conjunto de informações sensoriais em determinado momento da navegação, sendo que em cada estado é preciso decidir por uma ação dentre um conjunto de ações possíveis de serem tomadas, como por exemplo ajuste na direção e velocidade do robô, tal como é feito pelo controlador proposto neste trabalho. Frequentemente, o número de estados possíveis em um ambiente é muito elevado, especialmente porque um mesmo ponto do ambiente pode representar diversos estados, dependendo da direção de navegação do robô.

Uma vez escolhida a ação, o estado e a ação a ele associada são armazenados. Isto sendo feito a cada passo, o controlador termina por construir uma representação interna do ambiente com base nos estados visitados. Quanto mais vezes um mesmo estado é visitado, tanto mais o sistema é capaz de reforçar sua segurança quanto à melhor ação a ser tomada para aquele estado, uma vez que ações que tenham levado a resultados ruins passam a ter menos probabilidade de serem novamente escolhidas.

Sistemas desta natureza são um meio termo entre as abordagens puramente reativas e as abordagens puramente deliberativas. Podem ser considerados reativos na medida em que têm sua ação diretamente associada à informação sensorial corrente. Por outro lado, por necessitarem de uma estrutura de representação que lhes indique, por meio de uma busca a estados subseqüentes, a ação atual que lhes proporcionará melhor desempenho em médio e longo prazo, tais sistemas também podem ser considerados como deliberativos em um certo grau.

Exatamente pelo fato de sistemas de aprendizagem por reforço baseados em mecanismos de busca heurística terem seu desempenho melhorado com o passar do tempo, a partir da agregação de conhecimento e refinamentos sobre seu modelo de representação do

ambiente, alguns autores falam em navegação em ambiente inicialmente desconhecido (BRAGA & ARAÚJO, 1998). De fato, trabalhos apresentados na área costumam empregar exemplos de navegação considerando sempre um determinado ambiente de navegação (BRAGA & ARAÚJO, 1998; SUTTON *et al.*, 1999; FOX *et al.*, 1998), o que mostra que o conhecimento adquirido por um sistema de controle de navegação em um ambiente tem poucas chances de ser estendido a outros ambientes, a menos que um grande esforço de adaptação da estrutura interna de representação do sistema à nova topologia seja realizado.

Por outro lado, dado um ambiente sujeito a poucas alterações em suas condições topológicas, sistemas de controle de navegação baseados em aprendizagem por reforço mediante processos de busca heurística podem apresentar resultados melhores que aqueles observados por sistemas reativos sob as mesmas condições. Este melhor desempenho pode variar, desde a realização de percursos mais curtos até a transposição de obstáculos difíceis para sistemas reativos, como aquele apresentado pela Figura 7.10 (b). Não é difícil se prever, no entanto, que o mesmo ganho de desempenho não deve ser observado em ambientes dinâmicos como os apresentados no capítulo 7, para os quais se considerava a ocorrência de obstáculos móveis ou de múltiplos robôs.

O sistema de controle proposto neste trabalho também se baseia na representação de estados e ações, assim como nos modelos de aprendizagem por reforço discutidos acima. Além disso, no modelo proposto, os estados também representam o conjunto de informações sensoriais do robô em determinado momento, e os estados e ações mais promissores têm sua expressividade reforçada ao longo da navegação.

A primeira grande diferença entre os dois modelos consiste na ausência de necessidade de armazenamento de todos os estados do sistema por parte do MC para que sua decisão seja realizada. Ao contrário, apenas estados considerados críticos para o desempenho do robô, ou seja, aqueles que acarretaram colisões ou alvos atingidos, são armazenados nos neurônios do MC. Além disso, não há um número discreto de ações dentre as quais uma será escolhida. Uma ação neste caso corresponde a um valor real assumido no intervalo [0, 1] e nenhum processo de busca por soluções é realizado.

Com isto, o modelo apresentado neste trabalho perde a capacidade de previsão dos métodos baseados em busca heurística, que confere a capacidade aos robôs de transpor obstáculos difíceis. Voltando à Figura 7.10 (b), pode-se observar como o robô não consegue transpor sozinho o obstáculo. Entretanto, as vantagens apresentadas por esta nova abordagem se exprimem tanto pela substancial redução da quantidade de informação a ser armazenada, quando pelo caráter genérico que este conhecimento adquire, podendo ser facilmente aplicado a ambientes com topologias totalmente diferentes daquela sobre a qual o MC do controlador foi treinado. A Figura 8.1 ilustra este fato, apresentando em (a) o ambiente de treinamento de um robô e em (b) um ambiente totalmente diferente do primeiro, mas considerando-se o robô treinado em (a). Neste caso, o robô consegue atingir o alvo sem colidir e sem nunca antes ter navegado pelo ambiente (b), entretanto, a trajetória descrita em (b) não é a mais curta possível.

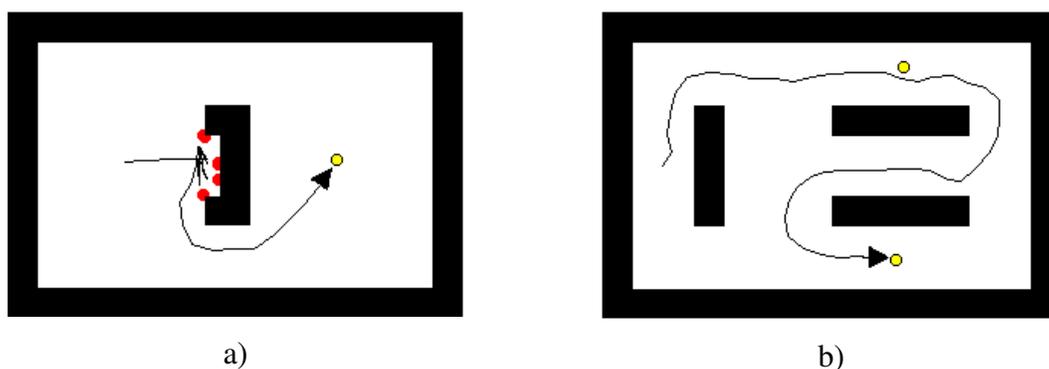


Figura 8.1 – Generalização do conhecimento: a) ambiente de treinamento; b) navegação com robô treinado em (a)

Alguns resultados de publicações apresentam simulações em ambientes que possuem apenas corredores estreitos e nenhum espaço amplo por onde o robô possa se locomover com maior liberdade. Neste caso, a pressão exercida pelas paredes dos corredores faz com que o robô se dirija com maior facilidade através do ambiente. Por isto, neste trabalho optou-se preferencialmente por ambientes com alternâncias entre espaços abertos, corredores e portas. A Figura 8.2 (a) ilustra como a utilização de corredores estreitos permite a construção de ambientes que guiam o robô, tendo havido 10 colisões, ao passo que em (b) o robô encontra dificuldades de locomoção, tendo colidido 12 vezes.

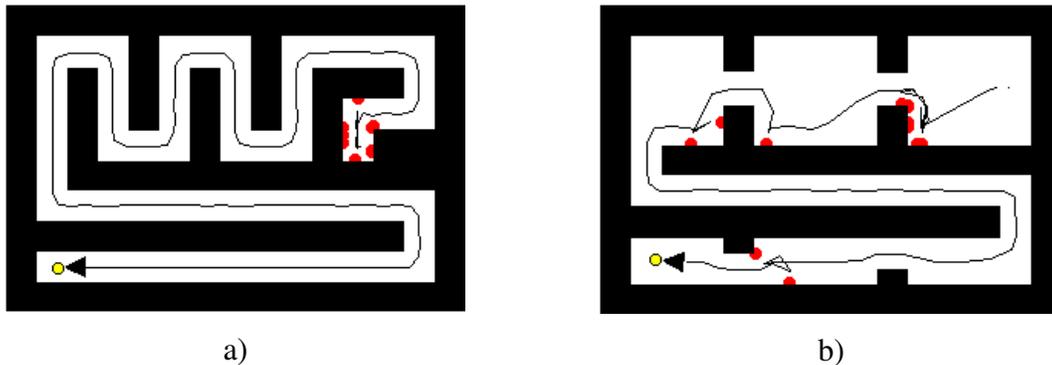


Figura 8.2 – Influência da topologia sobre a navegação: a) corredores estreitos; b) espaços alternados

Finalmente, ao longo dos capítulos que compõem este trabalho, apresentou-se um mecanismo de controle inteligente para a navegação de robôs móveis, baseado no trabalho de FIGUEIREDO (1997) e tendo como interesse central a utilização de mecanismos de aprendizagem que permitissem que uma quantidade reduzida de conhecimento inicial fosse necessária para a solução do problema. Esta abordagem pode, em princípio, apresentar algumas desvantagens com relação a mecanismos de controle dedicados à navegação em ambientes específicos e de topologias bem definidas. Contudo, ao possuir capacidade de ter seu desempenho melhorado com base na própria experiência, o controlador proposto passa a apresentar perspectivas muito interessantes no que diz respeito à realização de tarefas mais complexas para as quais uma definição formal seja difícil, senão impossível de ser proposta. Algumas contribuições originais foram apresentadas: introdução de um mecanismo de controle de velocidade, balanceamento e ampliação dos processos de aprendizagem verificados no trabalho original.

Com base no que foi apresentado neste trabalho, e no que se refere às perspectivas de futuros aperfeiçoamentos do modelo, bem como do desenvolvimento de linhas de investigação até o momento inexploradas, alguns comentários se fazem pertinentes. Em particular, a iniciativa de considerar múltiplos robôs em um mesmo ambiente de navegação abre espaço para um vasto campo de exploração na robótica móvel. Conforme já mencionado, neste trabalho os robôs não foram projetados para desenvolver qualquer tipo

de comportamento de colaboração ou competição explícita. Mesmo assim, estes comportamentos puderam ser implicitamente verificados.

O desenvolvimento de técnicas de aprendizado que levem em consideração a presença de outros indivíduos semelhantes no ambiente de aprendizagem pode levar à geração de comportamentos até então desconhecidos. Seria muito interessante observar a emergência de especializações dentro de um grupo de robôs, onde cada qual se adaptasse a uma determinada atividade, colaborando na solução de um problema complexo demais para um único robô. Neste momento, afirmações sobre comportamento social poderiam se tornar válidos para organismos artificiais.

Entretanto, para ser capaz de levar em consideração a presença de um indivíduo semelhante a si durante um processo de aprendizado, cada robô deve em primeiro lugar ser capaz de reconhecer-se como indivíduo. O reconhecimento de si mesmo como uma entidade que engloba todos os mecanismos que o compõem é uma tarefa extremamente complexa, dado o nível de subjetividade envolvido em tal relação. No entanto, é possível que alguma forma de auto-conhecimento e auto-preservação possa ser atribuída a mecanismos de aprendizagem e às estruturas criadas por estes mecanismos. Em particular, o mecanismo de correção de saturações em neurônios do Módulo de Coordenação através da utilização do Valor Acumulado de Ativação Sináptica, descrito no capítulo 3, constitui uma forma de reconhecimento de uma anomalia do sistema realizada pelo próprio sistema seguida pela recuperação de sua estabilidade. Esta atividade constitui uma forma primitiva de auto-conhecimento e aponta para caminhos que podem levar à expansão deste conceito.

Por fim, constitui parte essencial das perspectivas de desenvolvimentos futuros, a implementação em um robô real, dos conceitos apresentados neste trabalho. Só então toda a teoria apresentada durante os capítulos deste trabalho poderá ser definitivamente comprovada. Tal implementação certamente constituirá um grande desafio dadas as decisões que devem ser feitas quanto à escolha de parâmetros, números e tipo de sensores e adaptações do mecanismo de aprendizagem, embora o modelo do controlador se mantenha inalterado.

Apêndice A

Sistema de Simulação Computacional

Uma das principais características oferecidas pelo sistema de simulação computacional de ambientes de navegação é sua flexibilidade. Tendo sido projetado em linguagem Delphi 5.0 para ser executado em sistema operacional Windows, permite que a topologia do ambiente virtual possa ser rapidamente definida através da inserção de uma série de obstáculos e alvos, cujas posições são determinadas pelo usuário. Além disso, modificações na configuração do ambiente também podem ser realizadas com facilidade. Por medida de simplicidade, a geometria dos obstáculos é restrita a formas retangulares. Além disso, apesar da possibilidade de inclusão de múltiplos alvos no ambiente de navegação, apenas um alvo é habilitado por vez durante uma simulação.

O número de obstáculos e de alvos que podem compor um ambiente de navegação é limitado apenas pelos recursos computacionais disponíveis e pelas dimensões do ambiente. Na primeira versão do sistema, cada simulação só poderia ser realizada com um único robô. Atualmente esta restrição deixou de existir, conforme apresentado no Capítulo 7. Assim, da mesma forma como ocorre com obstáculos e alvos, o número de robôs suportados em uma mesma simulação é limitado apenas pela disponibilidade de recursos computacionais.

A Figura A.1 apresenta um processo de navegação em andamento no sistema de simulação, onde se pode notar, por exemplo, os obstáculos (polígonos pretos), a trajetória descrita pelo robô (linha preta), o ponto onde o robô colidiu contra um obstáculo (ponto vermelho), um alvo já capturado (ponto amarelo), as linhas que simulam os sensores de obstáculo (linhas pontilhadas pretas), o próximo alvo a ser capturado (ponto azul) e o menu de contexto com as opções de exibição de recursos gráficos.

Além de recursos gráficos, o sistema oferece ao usuário uma série de informações numéricas relativas à simulação em curso, que compreendem posição, direção e velocidade de deslocamento do robô, número de colisões e alvos já atingidos, direção relativa ao alvo,

distância percorrida, duração da navegação, número de iterações do robô com o ambiente, número de neurônios no Módulo de Coordenação, número de aprendizagens por recordação, além de uma série de sinais dedicados à monitoração, durante a navegação, das estruturas do Módulo de Coordenação descritas no Capítulo 3, tais como saídas dos neurônios motores e sinais W_r e W_s

Em particular, os sinais W_r e W_s também podem ser visualizados graficamente para facilitar o acompanhamento da resposta do Módulo de Coordenação frente às entradas sensoriais. Ainda na Figura A.1, pode-se observar o gráfico de saídas do Módulo de Coordenação no canto superior esquerdo, em que a série em vermelho corresponde ao sinal W_r e a série em azul corresponde ao sinal W_s .

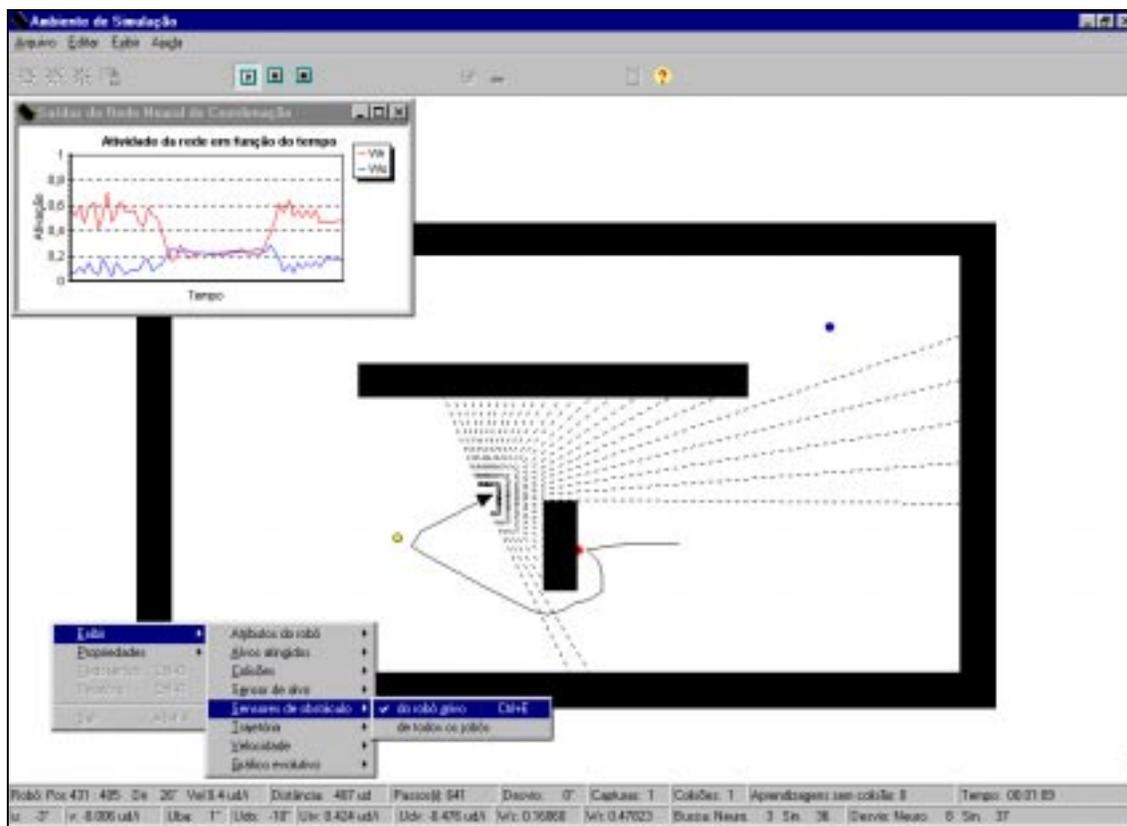


Figura A.1 – Ambiente de navegação em procedimento de simulação

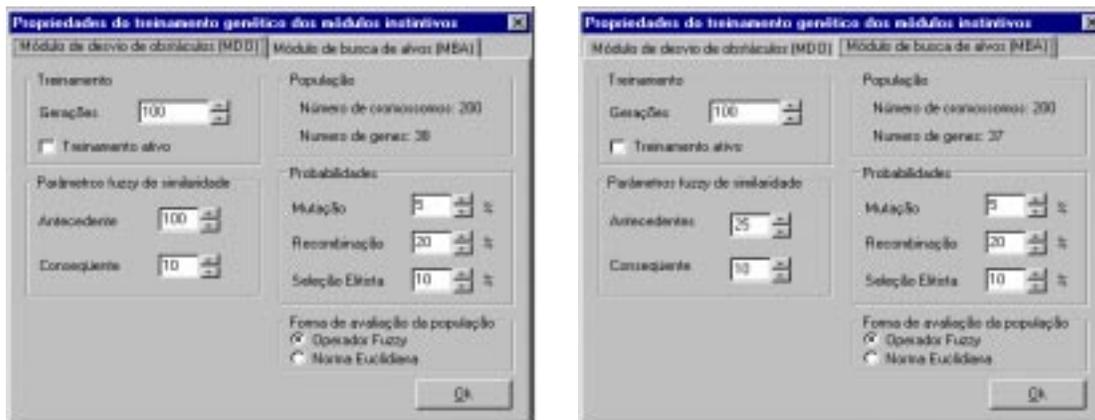
Não obstante estes recursos, as principais ferramentas que o ambiente de simulação oferece são relativas à análise de resultados. Isto ocorre porque, durante uma simulação, os

movimento aleatório. A caixa de diálogo apresenta as propriedades do obstáculo selecionado. Além do tipo de movimento, o usuário pode editar a velocidade do obstáculo, o grau de aleatoriedade do movimento (no caso de movimento aleatório) conforme discutido no capítulo 7 e as dimensões do obstáculo.



Figura A.3 – Edição das propriedades de um obstáculo

A seção de inicialização oferece também a oportunidade de configuração do Módulo de Desvio de Obstáculos (Figura A.4(a)) e do Módulo de Busca de Alvos (Figura A.4(b)), apresentados no Capítulo 3 como responsáveis pelos sinais de ajuste na direção de deslocamento do robô. A configuração destes módulos implica decidir se os mesmos serão modelados através de regras heurísticas pré-definidas do tipo **Se <Antecedentes> Então <Consequente>**, onde os antecedentes correspondem às leituras sensoriais e o consequente ao sinal de ajuste direcional, ou se as regras serão sintetizadas através de um processo de evolução genética.

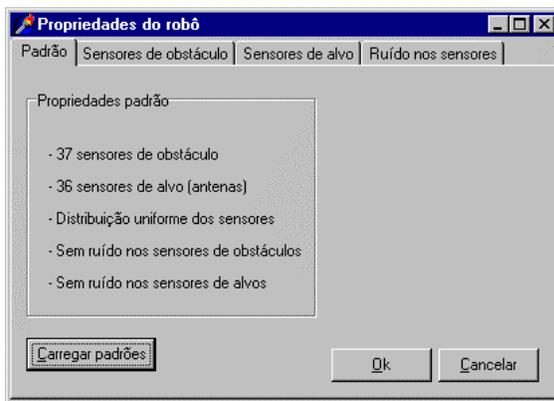


a)

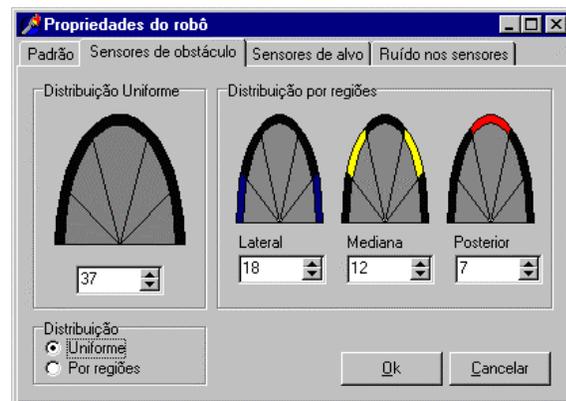
b)

Figura A.4 – Configuração dos módulos de nível hierárquico inferior: a) MDO; b) MBA

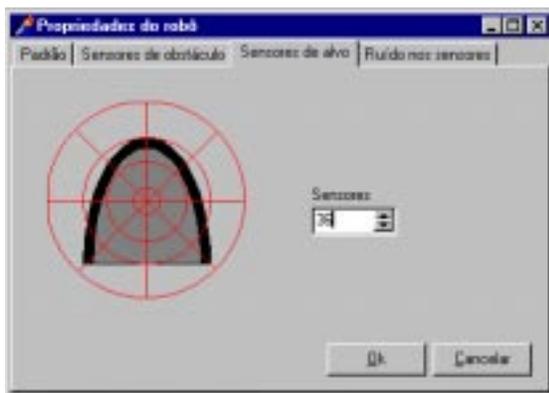
O processo de inserção do robô no ambiente de simulação é particularmente interessante, pois o usuário é capaz de determinar rapidamente o número e a distribuição dos sensores de obstáculos (Figura A.5(b)) e o número dos sensores de alvos (Figura A.5(c)) em torno do robô, o que possibilita a avaliação da relação custo/benefício de diversas configurações, com uma margem de confiabilidade segura, antes de uma implementação prática. Além disso, há a possibilidade de introdução de ruído nos sensores (Figura A.5(d)). Um próximo passo é o posicionamento do robô em um ponto qualquer do ambiente (exceto sobre obstáculos) e a determinação de sua direção inicial de deslocamento.



a)



b)



c)



d)

Figura A.5 – Propriedades do robô: a) propriedades padrão; b) quantidade e distribuição dos sensores de obstáculos; c) quantidade dos sensores de alvos; d) manipulação de ruído nos sensores

A modelagem de ruído nos sensores de obstáculos e de alvos é feita através de dois parâmetros. De acordo com a Figura A.5 (d), tanto para os sensores de obstáculos quanto

para os sensores de alvos, o campo *probabilidade* indica a possibilidade percentual de ocorrência de ruído em um determinado sensor e o campo *variação média* indica o quanto a ocorrência do ruído altera o sinal do sensor. No caso dos sensores de obstáculos a variação máxima permitida é de 25 e é medida em unidades de deslocamento (ud.) e no caso dos sensores de alvos a variação máxima é de 1.

Na seção de *Inicialização* também é possível salvar em arquivo a configuração do ambiente de navegação, a organização do Módulo de Coordenação de cada robô e o relatório de simulação que será apresentado a diante. Uma vez tendo sido salvas, estas informações podem ser utilizadas em futuras investigações, como por exemplo, o teste de características diferentes do controlador em um mesmo ambiente, a comparação do desempenho de controladores entre simulações diferentes e sujeitos a condições diferentes e mesmo a reutilização do Módulo de Coordenação em diversas simulações.

Após a construção do ambiente de acordo com as especificações do usuário e a inserção do robô neste ambiente, termina a seção de *Inicialização*. A partir de então, inicia-se o ciclo de navegação do robô. Como pode ser observado na Figura A.2, a primeira seção deste ciclo corresponde ao *Sensoriamento*. Isto ocorre pois para iniciar o processo de tomada de decisão o robô deve primeiramente situar-se no espaço em que se encontra. Nesta seção, cada sensor de obstáculo registra a distância do robô com relação ao obstáculo em sua direção, além de que, cada o sensor de alvo registra a intensidade do sinal emitido pelo alvo em seu campo de sensibilidade, sendo que, a partir da agregação das informações de todos os sensores de alvos, a direção do alvo com relação ao robô é determinada.

Após a seção de *Sensoriamento* o robô inicia a *Navegação*, onde o controlador autônomo utiliza os dados dos sensores de obstáculo, de alvo, de colisão e de captura do robô para produzir uma nova direção e uma nova velocidade de navegação. Terminada a seção de *Navegação*, o sistema inicia a seção de *Atualização*, onde a posição do robô é alterada no ambiente. O robô então avança na direção estabelecida pela seção de Navegação, de acordo com a velocidade estabelecida.

Duas situações distintas podem ocorrer após a *Atualização*: caso não haja colisão e nenhum alvo tenha sido atingido, o sistema inicia imediatamente a seção de *Sensoriamento* de forma a iniciar um novo ciclo de navegação. Entretanto, caso o robô colida com algum obstáculo ou algum alvo seja atingido, inicia-se a seção de *Aprendizagem*.

Na seção de *Aprendizagem*, o robô utiliza algoritmos de aprendizagem não supervisionada (Capítulos 3 e 4), para realizar o treinamento da ação de coordenação dos comportamentos de busca de alvos e desvio de obstáculos. A partir da *Aprendizagem*, duas situações podem ocorrer: caso o robô tenha sofrido uma colisão, a seção de *Reflexo* é executada. Entretanto, se o robô passou por um processo de aprendizagem pelo fato de ter atingido um alvo, então o ciclo de navegação continua na seção de *Sensoriamento*.

A seção de *Reflexo* ocorre após uma colisão do robô contra um obstáculo. A esta ação corresponde uma resposta instintiva por parte do robô, que consiste na mudança de direção em 180° com relação à direção no momento da colisão. Após a seção de *Reflexo*, o robô finalmente continua o ciclo de navegação a partir da seção de *Sensoriamento*.

O ciclo de navegação pode ser interrompido a qualquer momento. Neste caso, o sistema se torna apto à execução da seção de *Relatório*. Ao longo de toda a simulação, diversos dados considerados importantes para a análise do desempenho do robô são coletados. Na seção de *Relatório* estes dados são organizados em forma de gráficos, que apresentam por exemplo a evolução da aprendizagem do robô, bem como o registro das decisões tomadas pelo controlador diante de situações críticas.

Uma das informações apresentadas no relatório de simulação se refere ao próprio evento, representado de acordo com o ilustrado na Figura A.6. No caso da simulação representada por esta figura, dois robôs foram introduzidos no mesmo ambiente de navegação. Como ambos já haviam sido treinados, nenhuma colisão ocorreu. É interessante notar na Figura A.6, que é possível discriminar o robô a ser representado na ilustração. Além disso, o robô escolhido para ser representado será aquele ao qual as informações dos demais quadros do relatório irão se referir.

Além dos recursos gráficos, dados numéricos referentes à simulação constam do relatório, como é apresentado na Figura A.7. Este painel é interessante na medida que permite um confronto de informações precisas de simulações diferentes, dando margem a uma análise apurada de desempenho entre sistemas de controle. Ainda mais, estes valores podem ser registrados ao longo de sucessivas simulações com o mesmo robô, o que permite traçar um perfil da evolução do mecanismo de aprendizagem utilizado.



Figura A.6 – Representação gráfica da simulação

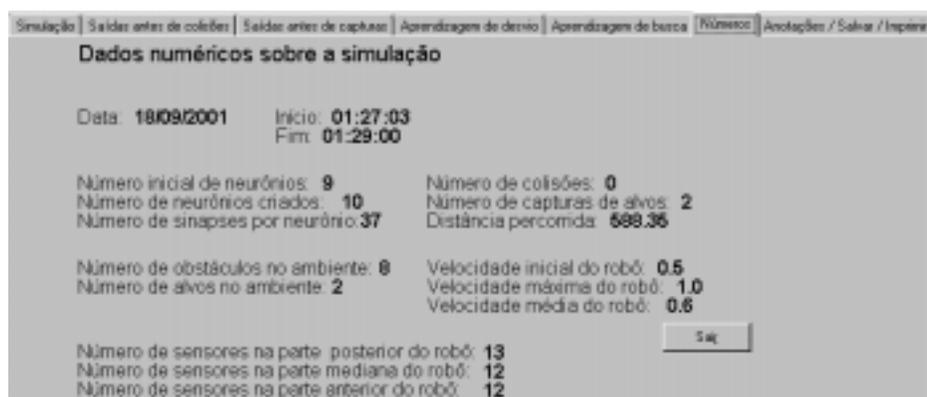


Figura A.7 – Dados numéricos sobre a simulação

Um outro recurso interessante do relatório de navegação é o quadro de observações apresentado pela Figura A.8. Através deste quadro, pode-se registrar os aspectos mais relevantes da simulação realizada, de forma que, com esta informação se mantenha salva junto com os demais parâmetros de avaliação da simulação. Com isto, pode-se ter uma interpretação precisa do foco de interesse da situação.

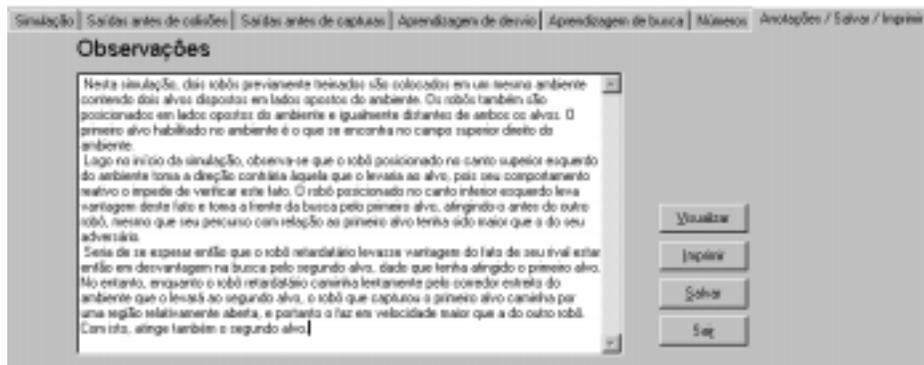


Figura A.8 – Quadro de observações sobre a simulação

Os neurônios do Módulo de Coordenação também são representados no relatório de simulação. Esta representação, ajuda sobremaneira a compreensão dos mecanismos de aprendizagem implementados naquele módulo, bem como a observação de suas falhas. A Figura A.9 apresenta duas representações após capturas para um neurônio de um dos robôs da simulação da Figura A.6. Em (a) tem-se o neurônio recém inserido, pela captura do primeiro alvo. As barras amarelas representam os pesos sinápticos w_{i1} , $i = 1, \dots, 71$ ligados aos sensores de alvos e os pontos vermelhos representam os pesos sinápticos s_{i1} (vide Capítulo 4). Em (b), observa-se o reforço dado pelos pesos sinápticos s_{i1} aos pesos w_{i1} cuja relevância se mostra maior ao longo de ambas as capturas.

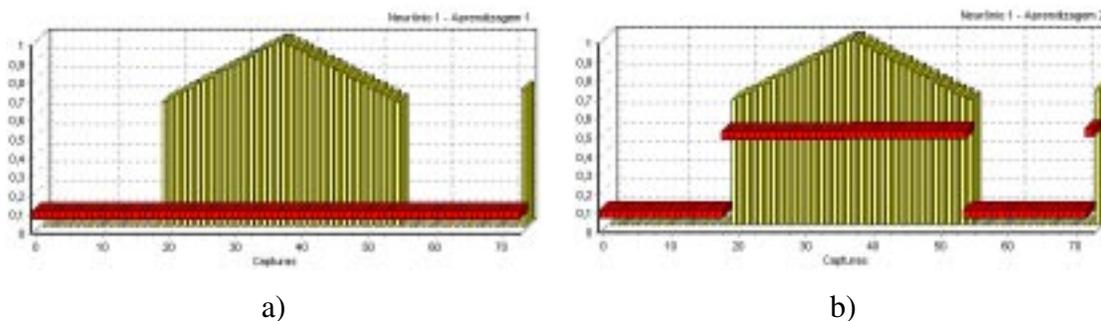


Figura A.9 – Dois momentos de aprendizagem após captura para um neurônio do Módulo de Coordenação

De forma análoga ao observado na Figura A.9, a Figura A.10 apresenta dois momentos de aprendizagem de desvio de obstáculos de um neurônio do Módulo de Coordenação. As representações não mais se referem à simulação da Figura A.6. As barras amarelas se

referem agora aos pesos sinápticos w_{j1} , $j = 1, \dots, 37$ ligados aos sensores de obstáculos e os pontos vermelhos se referem aos pesos sinápticos s_{j1} (vide Capítulo 3). Em (a), o neurônio recém inserido; em (b), a representação de uma aprendizagem por recordação, rotulada na figura como “Aprendizagem Sem Colisão”.

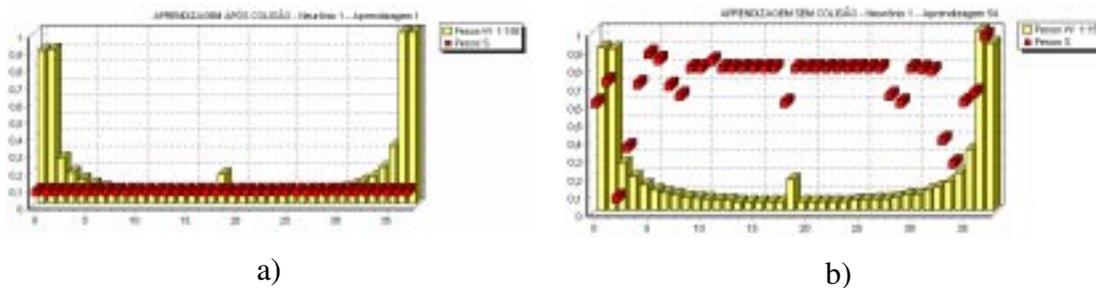


Figura A.10 - Dois momentos de aprendizagem de desvio em um neurônio do Módulo de Coordenação

Outra informação interessante sobre o desempenho do sistema é fornecida pelo gráfico apresentado na Figura A.11. Este gráfico verifica a resposta do controlador diante da aproximação de um obstáculo em uma situação que resultou em colisão. A linha vermelha representa o sinal W_r do controlador, a linha azul representa o sinal W_s e a linha verde representa a velocidade do robô. A distância do robô com relação ao obstáculo mais próximo é representada no gráfico pelas barras amarelas. Na figura nota-se que, conforme esta distância diminui, o sinal de desvio de obstáculos aumenta e a velocidade diminui.

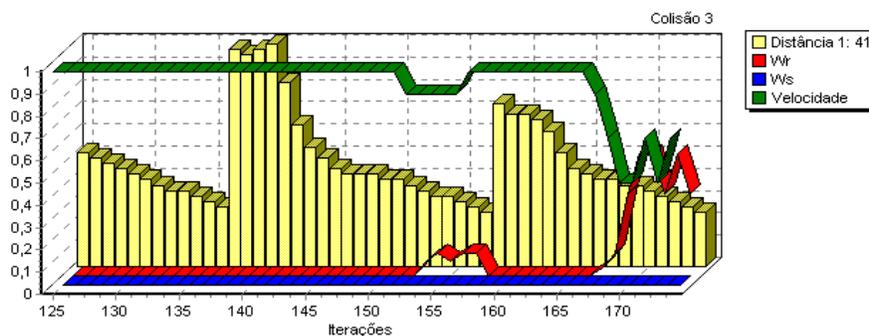


Figura A.11 – Resposta do controlador antes de colisão

A Figura A.12 apresenta uma situação semelhante àquela apresentada na Figura A.11, entretanto, agora a resposta do controlador é verificada diante da aproximação de um alvo.

As interpretações dadas às séries do gráfico são as mesmas, exceto pelas barras, que agora representam ângulos de desvio da direção de navegação do robô com relação à direção do alvo. Agora, à medida que o robô se dirige para o alvo, o sinal de busca se torna mais forte e, quando o alvo se encontra muito próximo, a velocidade cai. Este comportamento é analisado no Capítulo 4.

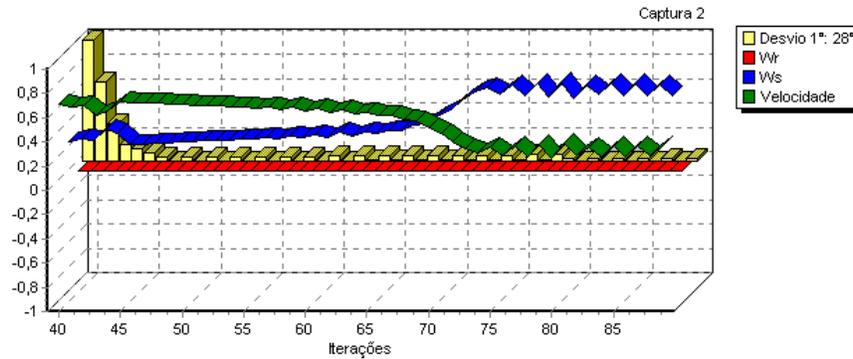


Figura A.12 – Resposta do controlador antes de captura de alvo

Referências Bibliográficas

- AGUIRRE, E. & GONZÁLEZ, A., 2000, “Fuzzy behaviors for mobile robot navigation: design, coordination and fusion”, *International Journal of Approximate Reasoning*, Vol. 25, pp. 255 – 289.
- ALBUS, J. S., MCLEAN, C., BARBERA, A. J., FITZGERALD, M. L., 1982, “An architecture for real-time sensory-interactive control of robots in a manufacturing environment”, IV IFAC/IFIP Symposium on Information Control Problems in Manufacturing Technology, Gaithersburg, MD.
- ALIEV, R. A., FAZLOLLAHI, B., VAHIDOV, R. M., 2001, “Genetic Algorithm-based learning of neural networks. Part 1: feed-forward neural networks”, *Fuzzy sets and systems*, Vol. 118, pp. 351 – 358.
- ANDERSON, 1953, “Recent advances in finding the best operating conditions”, *Journal of American Statistic Association*, Vol. 48, pp. 789 – 798.
- ANDERSON, T. & DONATH, M., 1991, “Animal behavior as a paradigm for developing robot autonomy”, em *Designing Autonomous Agents*, ed. P. Maes, MIT Press, Cambridge, MA, pp. 145 – 168.
- ANTSAKLIS, P., LEMMON, M., STIVER, J., 1996, “Learning to be autonomous”, em GUPTA, M. & SINHA N., (Eds.), *Intelligent Control Systems*, IEEE Press, pp. 28 – 62, NY, EUA.
- ARKIN, R. C., 1987, “Motor schema based navigation for a mobile robot: an approach to programming by behavior”, *Proceedings of the IEEE Conference on Robotics and Automation*, Raleigh, NC, pp. 264 – 271.
- ARKIN, R. C., 1998, “Behavior-based robotics”, MIT Press, Cambridge, MA.
- BÄCK, T, FOGEL, D. B., MICHALEWICZ, T, “Evolutionary computation 1 – basic algorithms and operators, IOP Publishing Ltd., Bristol, UK, 2000.
- BAUM, W. M., 1999, “Compreender o behaviorismo – ciência, comportamento e cultura”, pp. 69 – 81, tradução para o português Maria Tereza Araújo, Maria Amélia Matos, Gerson Yukio Tomanari, Editora Artes Médicas do Sul Ltda., Porto Alegre.
- BEER, R.D., 1995, “A dynamical systems perspective on agent-environment interaction”, *Artificial Intelligence*, vol. 72, pp. 173-215.

- BEKEY, G. & TOMOVIC, R., 1986, “Robot control by reflex actions”, IEEE International Conference on Robotics and Automation, San Francisco, CA, pp. 240 – 247, abril.
- BOOKER, L. B., 1993, “Recombination distributions for genetic algorithms *Foundations of Genetic Algorithms 2* ed. L. Whitley, San Mateo, CA, Morgan Kaufman, 1993: em BÄCK, T, FOGEL, D. B., MICHALEWICZ, T, “Evolutionary computation 1 – basic algorithms and operators, IOP Publishing Ltd., Bristol, UK, 2000.
- BRAGA, A. P. S. & ARAÚJO, F. R., 1998, “Navegação em ambiente fechado e inicialmente desconhecido utilizando aprendizagem por reforço”, XII Congresso da Sociedade Brasileira de Automática, Uberlândia, MG.
- BREMERMAN, 1962, “Optimization through evolution and recombination”, em M. C. Yovits, G. T. Jacobi & D. G. Goldstein editors, Self-organizing Systems, Spartan, Washington D. C., pp. 93 – 106.
- BROOKS, R. A., 1986, “A robust layered control system for a mobile robot”, IEEE Journal of Robotics and Automation, Vol. RA-2, Num. 1, pp. 14 – 23.
- BROOKS, R. A., 1987, “Planning is just a way of avoiding figuring out what to do next”, MIT Artificial Intelligence Laboratory Working Paper 303, Cambridge MA, in Cambrian Intelligence, The MIT Press, Cambridge, MA, 1998.
- BURNS, R. T., GOMOLIŃSKA, A., 2001, “Socio-cognitive mechanisms of belief change. Applications of generalized game theory to belief revision, social fabrication, and self-fulfilling prophesy”, Journal of Cognitive Systems Research, Vol. 2, pp. 39 – 54.
- CAMPOS, M., BONABEAU, E., THERAULAZ, G., DENEUBOURG, J.-L., 2000, “Dynamic Scheduling and Division of Labor in Social Insects”, Adaptive Behavior, Vol. 8, pp. 83 – 96.
- CARVALHO, J. A. D., SOUZA, J. S. S., “Sistema inteligente de navegação para robôs móveis autônomos”, XII Congresso Brasileiro de Automática, Uberlândia, MG.
- CASTELFRANCHI, C., 1998, “Modelling social action for AI agents”, Artificial Intelligence, Num. 103, pp. 157 – 182.
- CHUNG, I.-F., LIN, C. J., LIN, C.-T. (2000), “A GA-based fuzzy adaptive learning control network”, Fuzzy sets and systems, Vol. 112, pp. 65 – 84.

- COLOMBETTI, M., DORIGO, M., BORGHI, 1996, G. “Behavior Analysis and Training - A Methodology for Behavior Engineering”, IEEE Transactions on Systems, man, and Cybernetics, Part B, Cybernetics, vol. 26, nº 3, pp. 365-380, junho.
- CONNEL, J., 1989, “A colony architecture for an artificial creature”, Relatório Técnico Num. 1151, MIT AI Laboratory, agosto.
- CRESTANI, P. R. C., 1999, “Aperfeiçoamento de sistema de navegação autônomo baseado em redes neurais nebulosas”, Trabalho de Graduação, Departamento de Informática, Universidade Estadual de Maringá, dezembro.
- CRESTANI, P. R. J., FIGUEIREDO, M. F., VON ZUBEN, F. J., 2001, “Controlador neural nebuloso autônomo de direção e velocidade para robôs móveis”, V Simpósio Brasileiro de Automação Inteligente, novembro, Canela- RS.
- CUSSINS, A., 1990, “The connectionist construction of concepts”, in M. Boden (ed.) The Philosophy of Artificial Intelligence, Oxford University Press, pp. 368-440.
- DARWIN, C. 1859, “The origin of species”, John Murray, Penguin Classics, 1985.
- DAWKINS, R., 1998, “A escalada do monte improvável – uma defesa da teoria da evolução”, título original: Climbing mont Improbable, tradução de Suzana Sturlini Colto, Companhia das Letras, São Paulo.
- DONAHOE, J. W. & PALMER, D. C., 1994, “Learning and complex behavior”, Allyn and Bacon, Needham Heights, Massachusetts.
- DONNART, J.Y. & MEYER, J.A, 1996,. “Learning Reactive and Planning Rules in a Motivationally Autonomous Animat”, IEEE Transactions on Systems, Man, and Cybernetics-PartB: Cybernetics, vol. 26, no. 3, pp. 381-395, junho.
- EDELMAN, G., 1987, “Neural Darwinism: The theory of neuronal group selection”, Basic books, New York, EUA.
- EPSTEIN, S. L., 1998; “Pragmatic Navigation: Reactivity, Heuristics, and Search”; Artificial Intelligence; Vol. 100; pp. 275-321.
- FABRO, J. A., 1996, “Grupos neurais e sistemas nebulosos: aplicação à navegação autônoma”, tese de mestrado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, Campinas, SP.
- FARIA, G. & ROMERO, R. A. F., 2000, “Incorporating fuzzy logic to reinforcement learning”, IEEE 2000, SCE - ICMC – USP, São Carlos, Brasil.

- FARLEY, B. G. & CLARK, W. A., 1954, "Simulation of self-organizing systems by digital computer", IRE Transactions on Information Theory, Vol. 4, pp. 76 – 84.
- FIGUEIREDO M., 1997, "Redes neurais nebulosas aplicadas em problemas de modelagem e controle autônomo", tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, Campinas, SP.
- FIGUEIREDO, M., 1999, "Navegação Autônoma de Robôs", VII Escola de Informática da SBC - Regional Sul, pp. 74 - 106, maio.
- FIRBY, R. J., 1989, "Adaptive execution in complex dynamic worlds", Ph.D. Thesis, Technical Report YALEU/CSD/RR #672, Yale University, New Haven, CT.
- FLOREANO, D. & MONDADA, F., 1996, "Evolution of homing navigation in a real mobile robot; IEEE Transactions on Systems, Man and Cybernetics, Vol. XX, pp. 1-13.
- FLOREANO, D., MONDADA, F., 1998, "Evolutionary neurocontrollers for autonomous mobile robots", Neural Networks, Vol. 11, pp. 1461 – 1478.
- FOX, D., BURGARD W., THRUN S., 1998, "Active Markov localization for mobile robots", Journal of Robotics and Autonomous Systems, Preprint, março.
- FRASER, 1959, "Simulation of genetic systems by automatic digital computers", Australian Journal of Biological Science, Vol. 10, pp. 484 – 499.
- FREIRE, E. O.; BASTOS, T. F. F.; FREITAS, R. A., SCHNEEBELI, H. A.; SARCINELLI, M. F., 1998, "Um sistema de sensoriamento externo para robôs móveis com controle baseado em agentes", XII Congresso Brasileiro de Automática, Uberlândia, MG.
- FRIEDBERG, 1958, "A learning machine: part I", IBM Journal, Jan., pp. 2 – 13, 1958.
- FUKUDA, T. & KUBOTA, N., 1999, "An intelligent robotic system based on a fuzzy approach", Proceedings of the IEEE, Special Issue – Computational Intelligence, Vol. 87, nº 9, pp. 1448 – 1470, setembro.
- GEMAN, S., BIENENSTOCK, E., DOURSAT, R., 1992, "Neural networks and the bias/variance dilemma", Neural Computation, vol. 4, no. 1, pp. 1-58.
- GIRALT, G., CHATILA, R., VAISSET, M., 1984, "An integrated navigation and motion control system for autonomous multisensory mobile robots", First International Symposium on Robotics Research, pp. 191 – 214.
- GOMIDE F. & PEDRYCZ, W., 1998, "An introduction to fuzzy sets: analyze and design", The MIT Press, Cambridge, MA.

- GRASSÉ, P.-P., 1959, “La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cubitermes sp.* La théorie de la stigmergie: Essai d’interprétation du comportement des termites constructeurs”, *Insectes Sociaux*, Paris, Masson, em THERAULAZ, G. & BONABEAU, E., 1999, “A brief history of stigmergy”, *Artificial Life*, Vol. 5, pp. 97 – 116.
- GRUPEN, R. & HENDERSON, T., 1990, “Autochthonous behaviors – mapping perception to action”, em *Traditional and Non-Traditional Robotic Sensors*, ed. T. Henderson, NATO ASI Series, Vol. F-63, Springer-Verlag, Berlin, pp. 285 – 311.
- HAYKIN, S., 1999, “Neural networks: a comprehensive foundation”, Prentice Hall, New Jersey.
- HEBB, D. O., 1949, “The organization of behavior: a neuropsychological theory, New York: Wiley.
- HIROTA, K & PEDRYCZ, W., 1999, “Fuzzy computing for data mining”, *Proceedings of the IEEE, Special Issue – Computational Intelligence*, Vol. 87, n° 9, pp. 1575 – 1600, setembro.
- HOLLAND, J. H., 1975, “Adaptation in natural and artificial systems”, The University of Michigan Press, 1st. ed.
- HOLLAND, J. H., 1986, “Escaping brittleness: the possibility of general-purpose learning algorithms applied to rule-based systems”, em Michalski, R. S., Carbonell, J.G, Mitchell, T. M., *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, pp. 593 – 623, Morgan Kaufmann, San Mateo, CA.
- HOLLAND, O., MELHUISH, C., 1999, “Stigmergy, Self-Organization, and Sorting in Collective Robotics” *Artificial Life*, Vol. 5, pp. 173 – 202.
- HOPFIELD, J. J., 1982, “Neural networks and physical systems with emergent collective computational abilities”, *Proceedings of the National Academy of Sciences, USA*, vol. 79, pp. 2554 – 2558.
- HUSBANDS P., HARVEY I., CLIFF, D., MILLER G., 1997, “Artificial evolution: a new path for artificial intelligence?”, *Brain and Cognition*, Vol. 34, pp. 130 – 159, Academic Press.
- IWAKOSHI, Y., FURUHASHI, T., UCHIKAWA, Y., 1998, “A fuzzy classifier system for evolutionary learning of robot behaviors”, *Applied Mathematics and Computation*, Vol. 91, pp. 73 – 81.

- JERBIC, B, GROLINGER, K, VRANJES, B, 1999, “Autonomous agent based on reinforcement learning and adaptive shadowed network”, *Artificial Intelligence in Engineering*, Vol. 13, pp. 141 – 157.
- JIM, Kam-Chuen & GILES, C. L., 2001, “Talking helps: evolving communicating agents for the predator-prey pursuit problem”, *artificial life*, Vol. 6, n° 3, pp. 237 – 254, The MIT Press, Cambridge, MA.
- KAELBLING, L. & ROSENSCHEIN, S., 1991, “Action and planning in embedded agents”, em *Designing Autonomous Agents*, ed. P. Maes, MIT Press, Cambridge, MA, pp. 35 – 48.
- KAHN, P., 1991, “Specification and control of behavioral robot programs”, *Proceedings of the SPIE Sensor Fusion IV*, Boston, MA, novembro.
- KARSAI, I., 1999, “Decentralized control of construction behavior in paper wasps: an overview of the stigmergy approach”, *Artificial Life*, Vol. 5, pp. 117 – 136.
- KASABOV, N. K., 1996, “Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems”, *Fuzzy Sets and Systems*, Vol. 82, pp. 135 – 149.
- KOHONEN, T., 1982, “Self-organized formation of topologically correct feature maps”, *Biological Cybernetics*, vol. 43, pp. 59 – 69.
- KUO, R. J., XUE, K. C., 1999, “Fuzzy neural networks with application to sales forecasting”, *Fuzzy Sets and Systems*, Vol. 108, pp. 123 – 143, Elsevier Science Inc.
- KUBOTA, N. MORIOKA, T., KOJIMA F., FUKUDA T., 2001, “Learning of mobile robots using perception-based genetic algorithm”, *Measurement*, Vol. 29, pp. 237 – 248.
- LEBETEL, O., BESSIÈRE, P., DIARD, J., MAZER, E., 2000, “Bayesian robots programming”, *Les Cahiers du Laboratoire Leibniz*, n° 1; pp. 1-45, Grenoble, França; maio.
- LEE, Wei-Po, 1999, “Evolving complex robot behaviors”, *Information Sciences*, Vol. 121, pp. 1 – 25.
- LI, W., MA, C., WAHL, F. M., 1997 “A neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments”, *Fuzzy Sets and Systems*, Vol. 87, pp. 133 – 140.
- LIM, W., 1994, “An agent-based approach for programming mobile robots”, *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, pp. 3584 – 3589.

- LITTMAN, M. L., 2001, "Value-function reinforcement learning in Markov games", *Journal of Cognitive Systems Research*, Vol. 2, pp. 55 – 66.
- MAES, P., 1990, "Situated agents can have goals", *Robotics and Autonomous Systems*, Vol. 6, pp. 49 – 70.
- MATARIC, M. J., 2001, "Learning in behavior-based multi-robot systems: policies, models, and other agents", *Journal of Cognitive Systems Research*, Vol. 2, pp. 81 – 93.
- MCCULLOC, W. & PITTS, W., 1943, "A logical calculus of the ideas imminent in nervous activity", *Bull. Math. Biophys.*, vol. 5, pp. 115 – 133.
- MCFERTRIDGE, L., IBRAHIM, M. Y., 1998, "New technique of mobile robot navigation using a hybrid adaptive fuzzy potential field approach", *Computers ind. Engng*, Vol. 35, pp. 471 – 474.
- MCGARRY, K., WERMTER, S., MACINTYRE, J., 1999, "Hybrid neural systems: from simple coupling to fully integrated neural networks", *Neural Computing Surveys*, Vol. 2. pp. 62 – 93.
- MENDELECK, A., 1995, "Um modelo conexionista para a geração de movimentos voluntários em ambiente desestruturado", Tese de doutorado, Faculdade de Engenharia Mecânica, Unicamp, Campinas, SP.
- MILLER, D., 1995, "Experiences looking into niches", *Working Notes*, 1995 AAAI Spring Symposium: Lessons Learned from Implemented Software Architectures for Physical Agents, Palo Alto, CA, pp. 141 – 145, março.
- MILLER, E., LI, L., DESIMONE, R., 1991, "A neural mechanism for working and recognition memory in inferior temporal cortex", *Science*, vol. 254, pp. 1377-1379, novembro.
- MINSKY, M. L., 1954, "Theory of neural-analog reinforcement systems and it's application to the brain-model problem", Ph.D. thesis, Princeton University.
- MINSKY, M. L. & PAPERT, S. A., 1969, "Perceptrons", Cambridge, MA, MIT Press.
- MORAVEC, H., 1977, "Towards automatic visual obstacle avoidance", *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, agosto, p. 584.
- MORAVEC, R., 1983, "The Stanford Cart and the CMU Rover", *Proceedings of the IEEE*, Vol. 71, n° 7, pp. 872 – 884.

- NILSSON, N., 1969, “A mobile automaton: an application of artificial intelligence techniques”, Proceedings of the International Joint Conference on Artificial Intelligence, Washington D.C., May, reimpresso em Autonomous Mobile Robots, Vol. 2, eds, S. Iyengar and A. Elfes, IEEE Computer Society Press, Los Alamitos, 1991, pp. 233 – 244.
- NISHIMURA, S. I. & IKEGAMI, T., 1997, “Emergence of collective strategies in a prey-predator game model”, Artificial Life; Vol. 3, nº 4, pp. 243 – 260, The MIT Press, Cambridge, MA.
- NOGUSHI, N. & TERAOKA, H., 1997, “Path planning of an agricultural mobile robot by neural network and genetic algorithm”, Computers and Electronics in Agriculture, Vol. 18, pp. 187 – 204.
- NOLFI, S. & FLOREANO, D., 1999, “Coevolving predator and prey robots: do ‘arms races’ arise in artificial evolution ? ”, Artificial Life, Vol. 4, nº 4, pp. 311 – 335, The MIT Press, Cambridge, MA.
- OLIVEIRA, M., FIGUEIREDO, M., GOMIDE, F., 1994, “A neurofuzzy approach to autonomous control”, Proceedings of The 3rd International Conference on Fuzzy Logic, Neural Networks and Soft Computing, pp. 597-598, Iizuka, Japão.
- PAVLOV, I. P., 1927, “Conditioned Reflexes”, Oxford University Press, London, England.
- PLATANIOTIS, K. N., ANDROUTSOS, D., VENETSANOPOULOS, A. N., 1999 “Adaptive fuzzy systems for multichannel signal processing”, Proceedings of the IEEE, Special Issue – Computational Intelligence, Vol. 87, nº 9, pp. 1601 – 1622, setembro.
- ROSENBLATT, F., 1958, “The Perceptron: a probabilistic model for information storage and organization in the brain”, Psychological Review, Vol. 65, pp. 386 – 408.
- ROSENBLATT, J. & PAYTON, D., 1989, “A fine-grained alternative to the subsumption architecture for mobile robot control” Proceedings of the International Joint Conference in Neural Networks, June, pp. 317 – 323.
- RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J., 1986, “Learning representations of back-propagation errors”, Nature, Vol. 323, pp. 533 – 536.
- SANBORN, J. C., 1988, “A model of reaction for planning in dynamic environments”, Tese de mestrado, Departamento de Ciência da Computação, University of Maryland, College Park, maio.

- SANTOS, J., DURO, R. J., BECERRA, J. A., CRESPO, J. L., BELLAS, F., 2001, "Considerations in the application of evolution to the generation of robot controllers", *Information Sciences*, Vol. 133, pp. 127 – 148.
- SKINNER, B. F., 1953, "Science and human behavior", New York.
- SMITHERS, M., 1997, "Autonomy in robots and other agents", *Brain and Cognition*, Vol. 34., pp. 88 – 106, Academic Press.
- SONG, Kai-Tai, SHEEN, Liang-Huang, 2000, "Heuristic fuzzy-neuro network and it's application to reactive navigation of a mobile robot", *Fuzzy Sets and Systems* Vol. 110, pp. 331 – 340.
- STEELS, L., 1995, "When are robots intelligent autonomous agents?", *Journal of Robotics and Autonomous Systems*, Vol. 15, pp. 3 - 9.
- SUTTON, R.S. & BARTO, A.G. "Reinforcement learning: an introduction", MIT Press, 1998.
- SUTTON, R. S., PRECUP, D., SINGH, S., 1999, "Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning", *Artificial Intelligence*, Vol. 112, pp. 181 – 211, Elsevier Science Inc.
- THERAULAZ, G. & BONABEAU, E., 1999, "A brief history of stigmergy", *Artificial Life*, Vol. 5, pp. 97 – 116.
- THORNDIKE, E. L., 1911, "Animal intelligence", Hafner, Darien, CT.
- TSCHICHOLD-GÜRMAN, N., 1997, "The neural network model RuleNet and its application to mobile robot navigation", *Fuzzy Sets and Systems*, Vol. 85; pp. 287 – 303.
- URZELAI, J., FLOREANO, D., 2001, "Evolution of adaptive synapses: robots with fast adaptive behavior in new environments" *Evolutionary Computation*, Vol. 9, pp. 495-524.
- VENTURA, D. & MARTINEZ T., 1998, "Optimal control using a neural/evolutionary hybrid system", *WCCI*, pp. 1036 – 1041.
- VERSHURE, P.M.F.J., 1998, "A bottom up approach towards the acquisition and expression of sequential representations applied to a behaving real-world device: distributed adaptive control III", *Neural Networks*, 11, pp. 1531-1549.
- WATKINS, C. J. C. H., 1989, "Learning from delayed rewards", Ph. D. thesis, Cambridge University.

- YAGER, R. R., & FILEV, D. P., 1994, "Essential of fuzzy modeling and control", John Wiley & Sons Inc., New York, EUA.
- YANG, S. X. & MENG, M., 2000, "An efficient neural network approach to dynamic robot motion planning", Neural Networks, Vol. 13, pp. 143 – 148; Elsevier Science Inc.
- YANG, Y., XU, X., ZHANG, W., 2000, "Design neural networks based fuzzy logic", Fuzzy Sets and Systems, Vol. 114, pp. 325 – 238.
- ZADEH, L. A., 1965, "Fuzzy Sets", Information and Control, 8: 338-53.
- ZELTZER, D. & JOHNSON, M., 1991, "Motor planning: an architecture for specifying and controlling the behavior of virtual actors", Journal of Visualization and Computer Animation, Vol. 2, pp. 74 – 80.

Índice Remissivo

AGUIRRE & GONZÁLEZ, 2000	22, 71
ALBUS <i>et al.</i> , 1982.....	4
ALIEV <i>et al.</i> , 2001.....	23
ANDERSON, 1953.....	31
ANDERSON & DONATH, 1991	4
ANTSAKLIS <i>et al.</i> , 1996.....	159
ARKIN, 1987	4
ARKIN, 1998.....	2, 4, 138
BÄCK <i>et al.</i> , 2000	35
BAUM, 1999	36, 38
BRAGA & ARAÚJO, 1998.....	42, 71, 163
BEER, 1995.....	11
BEKEY & TOMOVIC, 1986.....	4
BOOKER, 1993	32
BREMERMANN, 1962	31
BROOKS, 1986	3
BROOKS, 1987	99
BURNS & GOMOLIŃSKA, 2001.....	137
CAMPOS <i>et al.</i> , 2000.....	137
CARVALHO & SOUZA, 1998	6
CASTELFRANCHI, 1998	137
CASTELFRANCHI, 2001	137
CHUNG <i>et al.</i> , 2000.....	6, 23
COLOMBETTI <i>et al.</i> , 1996.....	1
CONNEL, 1989	4
CRESTANI, 1999	126
CRESTANI <i>et al.</i> , 2001	71
CUSSINS, 1990.....	10
DARWIN, 1859.....	30

DAWKINS, 1998.....	150
DONAHOE & PALMER, 1994.....	37
DONNART & MEYER, 1996.....	5
EDELMAN, 1987.....	6, 45, 49
EPSTEIN, 1998.....	71, 97
FABRO, 1996.....	1, 45
FARIA & ROMERO, 2000.....	71, 97
FARLEY & CLARK, 1954.....	39
FIGUEIREDO, 1997.....	1, 6, 9, 10, 11, 22, 26, 27, 28, 45, 49, 55, 56, 58
.....	65, 66, 69, 73, 74, 117, 126, 156, 165
FIGUEIREDO, 1999.....	1, 159, 160, 161
FIRBY, 1989.....	4
FLOREANO & MONDADA, 1996.....	72
FLOREANO & MONDADA, 1998.....	6, 72, 100
FOX & BURGARD, 1998.....	71
FOX <i>et al.</i> , 1998.....	42, 163
FRASER, 1959.....	31
FREIRE <i>et al.</i> , 1998.....	71, 97
FRIEDBERG, 1958.....	31
FUKUDA & KUBOTA, 1999.....	4, 22
GEMAN <i>et al.</i> , 1992.....	21
GIRALT <i>et al.</i> , 1984.....	2
GOMIDE & PEDRYCZ, 1998.....	23
GRASSÈ, 1959.....	136, 138
GRUPEN & HENDERSON, 1990.....	4
HAYKIN, 1999.....	14, 15, 19
HIROTA & PEDRYCZ, 1999.....	22
HEBB, 1949.....	11, 17, 121
HOLLAND, 1975.....	31, 32
HOLLAND, 1986.....	45
HOLLAND & MELHUIH, 1999.....	138

HOPFIELD, 1982.....	18
HUSBANDS <i>et al.</i> , 1997.....	100
IWAKOSHI <i>et al.</i> , 1998	6, 71, 100
JERBIC <i>et al.</i> , 1999.....	71
JIM & GILES, 2001.....	71, 137
KAELBLING & ROSENSCHEIN, 1991	4
KAHN, 1991.....	4
KARSAI, 1999	136
KASABOV, 1996.....	4, 22
KOHONEN, 1982	18
KUBOTA <i>et al.</i> , 2001.....	72, 100, 101
KUO & XUE, 1999	22, 23
LEBETEL <i>et al.</i> , 2000	71
LEE, 1999	71, 100
LI, 1994.....	5
LI <i>et al.</i> , 1997	6, 22, 71
LIM, 1994	4
LITTMAN, 2001	42
MAES, 1990.....	4
MATARIC, 2001	4, 5, 137
MCCULLOC & PITTS, 1943.....	15
McFETRIDGE & IBRAHIM, 1998	6, 23
MCGARRY <i>et al.</i> , 1999	4
MENDELECK, 1995	15, 36
MICHALEWICZ, 1996	106
MICHIE, 1961.....	39
MICHIE & CHAMBERS, 1968.....	39
MILLER, 1995.....	4
MILLER <i>et al.</i> , 1991	10
MINSKY, 1954	39
MINSKY & PAPERT, 1969	18

MORAVEC, 1977	2
MORAVEC, 1983	2
NILSSON, 1969.....	2
NISHIMURA & IKEGAMI, 1997	71, 137
NOGUCHI & TERA0, 1997	71
NOLFI & FLOREANO, 1999	71, 137
OLIVEIRA <i>et al.</i> , 1994.....	7
PAVLOV, 1927	36
PLATAN10TIS <i>et al.</i> , 1999	22
ROSENBLATT, 1958	17
ROSENBLATT & PAYTON, 1989	4
RUMELHART <i>et al.</i> , 1986	18
SANBORN, 1988.....	4
SANTOS <i>et al.</i> , 2001.....	31, 71, 100
SKINNER, 1953	37
SMITHERS, 1997	1, 71, 115, 159, 160
SONG & SHEEN, 2000.....	6, 23, 71
STEELS, 1995.....	160
SUTTON & BARTO, 1998	39, 40, 42, 162
SUTTON <i>et al.</i> , 1999.....	163
THERAULAZ & BONABEAU, 1995	137
THERAULAZ & BONABEAU, 1999	136
THORNDIKE, 1911.....	8, 36, 56
TSCHICHOLD-GÜRMAN, 1997	71
URZELAI & FLOREANO, 2001	100
VENTURA & MARTINEZ, 1998.....	4
VERSHURE, 1998	5
WATKINS, 1989	43
YAGER & FILEV, 1994.....	24, 27
YANG <i>et al.</i> , 2000.....	22, 23
YANG & MENG, 2000.....	71

ZADEH, 1965	22
ZELTZER & JOHNSON, 1991	4

