

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

**Sobre cognição, adaptação e homeostase: uma análise
e síntese de ferramentas computacionais bio-
inspiradas aplicadas à navegação autônoma de robôs**

Autor: Renan Cipriano Moioli

Orientador: Prof. Dr. Fernando José Von Zuben

Co-orientador: Dra. Patrícia Amâncio Vargas

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Banca Examinadora

Fernando José Von Zuben, Dr. DCA/FEEC/Unicamp
Christiano Lyra Filho, Dr. DENSIS/FEEC/Unicamp
Maurício Fernandes Figueiredo, Dr. DC/UFSCAR

Campinas, SP
Setembro/2008

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

M728s Moiola, Renan Cipriano
Sobre cognição, adaptação e homeostase: uma análise e
síntese de ferramentas computacionais bio-inspiradas
aplicadas à navegação autônoma de robôs / Renan Cipriano
Moioli. --Campinas, SP: [s.n.], 2008.

Orientadores: Fernando José Von Zuben, Patrícia
Amâncio Vargas

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

1. Inteligência artificial. 2. Homeostase. 3. Redes neurais
(Computação). 3. Computação evolutiva. 4. Robôs móveis.
5. Sistemas inteligentes de controle. I. Von Zuben, Fernando
José. II. Vargas, Patrícia Amâncio. III. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica e
de Computação. IV. Título.

Título em Inglês: On cognition, adaptation and homeostasis: analysis and synthesis of
bio-inspired computational tools applied to robot autonomous
navigation

Palavras-chave em Inglês: Learning classifier system, Artificial neural networks,
Artificial homeostatic systems, Evolutionary robotics,
Reactive and non reactive behaviour

Área de concentração: Engenharia de Computação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Christiano Lyra Filho, Mauricio Fernandes Figueiredo

Data da defesa: 10/09/2008

Programa de Pós-Graduação: Engenharia Elétrica

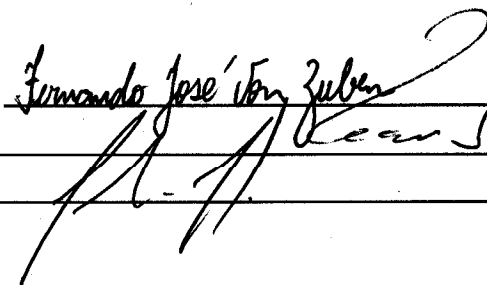
COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Renan Cipriano Moiolli

Data da Defesa: 10 de setembro de 2008

Título da Tese: "Sobre Cognição, Adaptação e Homeostase: Uma Análise e Síntese de Ferramentas Computacionais Bio-Inspiradas Aplicadas à Navegação Autônoma de Robôs"

Prof. Dr. Fernando José Von Zuben (Presidente):



Prof. Dr. Maurício Fernandes Figueiredo: _____

Prof. Dr. Christiano Lyra Filho: _____

Resumo

Este trabalho tem como objetivos principais estudar, desenvolver e aplicar duas ferramentas computacionais bio-inspiradas em navegação autônoma de robôs. A primeira delas é representada pelos Sistemas Classificadores com Aprendizado, sendo que utilizou-se uma versão da proposta original, baseada em energia, e uma versão baseada em precisão. Adicionalmente, apresenta-se uma análise do processo de evolução das regras de inferência e da população final obtida. A segunda ferramenta trata de um modelo denominado sistema homeostático artificial evolutivo, composto por duas redes neurais artificiais recorrentes do tipo NSGasNets e um sistema endócrino artificial. O ajuste dos parâmetros do sistema é feito por meio de evolução, reduzindo-se a necessidade de codificação e parametrização a priori. São feitas análises de suas peculiaridades e de sua capacidade de adaptação. A motivação das duas propostas está no emprego conjunto de evolução e aprendizado, etapas consideradas fundamentais para a síntese de sistemas complexos adaptativos e modelagem computacional de processos cognitivos. Os experimentos visando validar as propostas envolvem simulação computacional em ambientes virtuais e implementações em um robô real do tipo Khepera II.

Palavras-chave: sistemas classificadores com aprendizado; redes neurais artificiais; sistemas homeostáticos artificiais; robótica evolutiva; comportamento reativo e não-reativo.

Abstract

The objectives of this work are to study, develop and apply two bio-inspired computational tools in robot autonomous navigation. The first tool is represented by Learning Classifier Systems, using the strength-based and the accuracy-based models. Additionally, the rule evolution mechanisms and the final evolved populations are analyzed. The second tool is a model called evolutionary artificial homeostatic system, composed of two NSGasNet recurrent artificial neural networks and an artificial endocrine system. The parameters adjustment is made by means of evolution, reducing the necessity of a priori coding and parametrization. Analysis of the system's peculiarities and its adaptation capability are made. The motivation of both proposals is on the concurrent use of evolution and learning, steps considered fundamental for the synthesis of complex adaptive systems and the computational modeling of cognitive processes. The experiments, which aim to validate both proposals, involve computational simulation in virtual environments and implementations on real Khepera II robots.

Keywords: learning classifier systems; artificial neural networks; artificial homeostatic systems; evolutionary robotics; reactive and non reactive behaviour.

*Aos meus pais, Valdir e Iracelis, e
meu irmão, Bruno*

*Agora que sinto amor
Tenho interesse nos perfumes.
Nunca antes me interessou que uma flor tivesse cheiro.
Agora sinto o perfume das flores como se visse uma coisa nova.
Sei bem que elas cheiravam, como sei que existia.
São coisas que se sabem por fora.
Mas agora sei com a respiração da parte de trás da cabeça.
Hoje as flores sabem-me bem num paladar que se cheira.
Hoje às vezes acordo e cheiro antes de ver.*

Alberto Caeiro

Agradecimentos

À Força que existe entre todos nós, nos une e nos faz felizes.

Agradeço a toda a minha família, pessoas muito boas, que sempre me trataram com muito amor. Em especial, meus pais, Valdir e Iracelis, meu irmão, Bruno (Xinho), e minha vó, Edna. Tenho muita sorte de ter vocês na minha vida.

Aos meus orientadores, Prof. Dr. Fernando José Von Zuben e Dra. Patrícia Amâncio Vargas. Sou muito grato a tudo que pude aprender com vocês, ao exemplo de profissional e pessoa que vocês são. Muito obrigado pela confiança e pela motivação.

À Universidade de Sussex, ao Centre for Computational Neuroscience and Robotics e ao Prof. Phil Husbands, pelo apoio no desenvolvimento de parte deste trabalho.

À minha Bela, pela companhia, amor e paciência. São muitos os momentos inesquecíveis ao seu lado.

Aos meus amigos Pedrão, Pher, Thalitão, Carol, Rafa, Free, Xandy, Mucin, Dida, Maribrava, Plínio, Maurício, Gutão, Tati, Primo, Trovão, Boy, Mendigo, Piu, Renato, Pira, Guerra, Kiipper, Lila, Zollezzy, Homer, Digão, Aninha, Má, Filgas, ACS, Or, Giu, Fran, Diva, Léo, Dwidle, PH, Rafinha.

Aos Marimbondos, por serem minha família em Campinas. Muito sucesso a todos vocês.

Aos amigos do LBiC, com um abraço especial para Titi, Pablito, Will, Hamiltão, Gregs, Renatinho, Maestro Eurípedes, Napa, Hêlder, Michel e Mari. O trabalho fica mais alegre com a companhia de vocês.

Aos amigos do futebol, Rafinha, Luís, Baiano, e aos times que tive o prazer de integrar, Pôneis 02, LEU e Pós-Feec.

Aos mestres que pude conhecer na Unicamp, em especial Prof. Léo Pini, Prof. Christiano e Profa. Sônia, Prof. Rafael, Prof. Gomide, Prof. Pedro Peres, Prof. Ivanil, Prof. Romis, Profa. Salette, Prof. Ary.

À Unicamp, à FEEC e seus funcionários.

À CAPES e ao CNPq, pelo apoio financeiro.

A todos que estiveram comigo nessa caminhada.

Sumário

Trabalhos Publicados Pelo Autor	xv
Lista de Figuras	xvii
Lista de Tabelas	xxi
1 Introdução	1
1.1 Computação bio-inspirada	1
1.2 Cognição, inteligência artificial e robótica	2
1.3 Objetivos e organização do texto	6
2 Sistemas de inteligência computacional a serem adotados	9
2.1 Computação Evolutiva	9
2.1.1 Algoritmos Genéticos	10
2.2 Redes Neurais Artificiais	15
2.3 GasNets	18
2.3.1 Dinâmica e Treinamento da GasNet	19
2.3.2 Algoritmo Evolutivo	23
2.3.3 NSGasNets	28
3 Investigando Sistemas Classificadores com Aprendizado em tarefas robóticas reativas e não-reativas	33
3.1 Introdução	34
3.2 Fundamentos de Sistemas Classificadores com Aprendizado	36

3.2.1	SCAO	37
3.2.2	XCS	39
3.3	Trabalhos Relacionados	44
3.4	Experimentos	48
3.4.1	Experimento 1	49
3.4.2	Experimento 2	54
3.5	Discussão	61
4	Sistema Homeostático Artificial Evolutivo	65
4.1	Homeostase	65
4.2	Sistema Endócrino Artificial - SEA	69
4.3	Sistema Homeostático Artificial Evolutivo	72
4.3.1	Descrição e justificativa do modelo	72
4.3.2	Procedimentos Experimentais	75
4.3.3	Experimentos	82
4.3.4	Discussão	99
5	Conclusão	103
5.1	Análise das principais contribuições da tese	103
5.2	Perspectivas futuras e propostas de extensão	105
5.2.1	Sistemas classificadores com aprendizado	105
5.2.2	Sistema homeostático artificial evolutivo	106
5.3	Considerações finais	107
A	Robô Khepera II	109
A.1	Introdução	109
A.2	Robô Khepera II	109
B	Simuladores	113
B.1	Introdução	113
B.2	Simulador WSU Khepera Robot Simulator	113

B.2.1 Interfaces	115
B.3 Simulador KIKS	116
Referências Bibliográficas	118
Índice Remissivo de Autores	133

Trabalhos Publicados pelo Autor

1. Moioli, R.C., Vargas, P. A., Von Zuben, F. J., Husbands, P. “Evolving an Artificial Homeostatic System”. G. Zaverucha and A. Loureiro da Costa (Eds.): 19th Brazilian Symposium on Artificial Intelligence, SBIA 2008, Salvador, Brasil, pp. 278–288, LNAI 5249, Springer-Verlag Berlin Heidelberg, 2008.
2. Moioli, R.C., Vargas, P. A., Von Zuben, F. J., Husbands, P. “Towards the evolution of an artificial homeostatic system”. 2008 IEEE Congress on Evolutionary Computation (CEC 2008), pages 4024–4031, 2008.
3. Moioli, R.C., Vargas, P. A., Von Zuben, F. J. “Analyzing Learning Classifier Systems in Reactive and Non-Reactive Robotic Tasks”. In Jaume Bacardit, Ester Bernado-Mansilla, Martin V. Butz, Tim Kovacs, Xavier Llorca, Keiki Takadama (Eds.) Present and Future of Learning Classifier Systems. Revised Selected Papers of the 9th and 10th editions of the International Workshop on Learning Classifier Systems, Springer-Verlag, LNCS vol. 4998, 2008
4. Vargas, P.A., Moioli, R.C., de Castro, L.N., Timmis, J., Neal, M., Von Zuben, F.J. “Artificial Homeostatic System: a Novel Approach”. Proceedings of the VIIIth European Conference on Artificial Life, Canterbury, Kent (UK), 2005

Lista de Figuras

2.1	Principais etapas de um algoritmo genético	11
2.2	Representação genética de um cromossomo em codificação binária. Os genes podem ter tamanho variável, e cada locus (posição) no gene possui um alelo (valor) com base na codificação adotada	12
2.3	Relação genótipo-fenótipo (reproduzido de Barra (2007), com permissão do autor). Cada campo do genótipo codificado possui uma interpretação na formação da proposta de solução.	12
2.4	Crossover de um ponto (a) e crossover uniforme (b) (reproduzido de Barra (2007), com permissão do autor)	14
2.5	Operador de mutação (reproduzido de Barra (2007), com permissão do autor)	15
2.6	Modelo de um neurônio artificial	17
2.7	Rede neural artificial do tipo MLP totalmente conectada.	17
2.8	Modelo de uma rede GasNet composta por 6 nós. As conexões sinápticas e gasosas estão ilustradas. Linhas cheias representam conexões elétricas excitatórias e linhas pontilhadas representam conexões elétricas inibitórias. Note que o nó 3 emite uma nuvem de gás e pode modular os nós 2 e 4.	18
2.9	Função $T(t)$ (a) e função $H(x)$ (b). Note que a função $T(t)$ possui um máximo e satura com valor 1.	20
2.10	Comportamento das curvas $\tanh(kx)$ para diversos valores de k	22

2.11	GasNet obtida para resolução do problema de geração de padrões. A rede está limitada a um espaço euclidiano bidimensional de dimensões 100×100 , e só emite gás tipo 1, com o raio máximo de ação do gás de cada neurônio simbolizado pela cor vermelha.	26
2.12	Curvas relativas a algumas das variáveis relevantes para a compreensão da dinâmica da rede GasNet. Estão representados comportamentos obtidos em 100 passos do algoritmo.	27
2.13	Destaque da saída do nó 1, que representa a solução proposta pela GasNet evoluída para o problema de gerador de padrões.	28
2.14	Curvas relativas a algumas das variáveis importantes para a compreensão da dinâmica da rede NSGasNet. São apresentados comportamentos obtidos em 100 passos do algoritmo.	31
2.15	Destaque da saída do nó 1, que representa a solução proposta pela NSGasNet evoluída para o problema de gerador de padrões.	32
3.1	Interação do Sistema Classificador com o ambiente.	37
3.2	kovacs	42
3.3	Resultados dos experimentos simulados, incluindo a arena (a), a trajetória do SCA (b), e a trajetória do XCS (c).	50
3.4	Resultados do Experimento Real.	54
3.5	Ambiente de simulação para o Experimento 2 (adaptado de Jakobi (1998)) (a) e implementação do experimento utilizando o simulador WSU (b). A cada rodada de simulação, o robô é posicionado aleatoriamente na posição 1 ou 6 de um dos T-Maze, um com lâmpada à direita e outro com lâmpada à esquerda.	56
4.1	Arquitetura do Sistema Endócrino Artificial (SEA). Aqui estão representados seus três módulos principais: módulo de nível hormonal (NH), módulo controlador de produção hormonal (CPH), e módulo glândula endócrina (GE).	70
4.2	Visão geral do Sistema Homeostático Artificial Evolutivo (inspirado em Vargas et al. (2005))	73

4.3	Variação de δ em função das leituras do sensor de luz mais estimulado	74
4.4	Operação do algoritmo genético distribuído. Em cada aplicação, um conjunto de soluções é formado a partir do conjunto de toda a população. Os indivíduos escolhidos são ordenados, e os de maior <i>fitness</i> (cor preta) têm maior chance de reprodução. Por sua vez, os de menor <i>fitness</i> (cor cinza) têm maior chance de substituição. O novo indivíduo gerado é representado pelo quadrado hachurado.	78
4.5	Cenário de treinamento para o comportamento de desvio de obstáculos (a); e para o comportamento de fototaxia (b)	79
4.6	Evolução do <i>fitness</i> : desvio de obstáculos (a), fototaxia (b) e SEA (c)	83
4.7	Redes evoluídas: desvio de obstáculos N1 (a) e fototaxia N2 (b). Estão representados os neurônios de saída relativos ao motor esquerdo (ME) e motor direito (MD), e também os neurônios relativos aos sensores de entrada, sensor esquerdo (SE), sensor direito (SD), sensor frontal (SF) e sensor traseiro (ST).	84
4.8	Evolução do Sistema Homeostático Artificial (SHAE). Trajetória do melhor indivíduo evoluído após 300 iterações (a); e dinâmica das variáveis para $\beta = 0,01$ (b).	86
4.9	Destaque da dinâmica das variáveis para $\beta = 0,01$	87
4.10	Trajetória e dinâmica das variáveis para $\beta = 0,001$. A seta indica o momento em que o <i>EI</i> supera $\theta = 52,0$, e o robô começa a procurar a fonte luminosa (<i>EE</i>). Em muitos outros momentos houve aproximação da fonte luminosa pela simples natureza exploratória da trajetória do robô, sem a intenção de realizar recarga de bateria.	89
4.11	Trajetória e dinâmica das variáveis para $\beta = 0,02$. Todo ciclo de alta do nível hormonal (<i>NH</i>) é seguido por aproximação da fonte luminosa (<i>EE</i>).	90
4.12	Destaque da dinâmica das variáveis para $\beta = 0,02$. A seta indica o momento em que o sistema é controlado com predominância da rede de fototaxia.	91
4.13	Trajetória (a) e dinâmica das variáveis (b) para $0,005 \leq \beta \leq 0,015$	92
4.14	Trajetória (a) e dinâmica das variáveis (b) para perturbação de velocidade (a velocidade dobra). As setas indicam os momentos em que o comportamento predominante é o de fototaxia.	94

4.15	Trajectoria (a) e dinamica das variáveis (b) para SHAE sem receptor. O ponto A indica o momento em que o robô “morre”, após permanecer girando sobre o próprio eixo (comportamento da rede de fototaxia na ausência de luz) por muitas iterações.	96
4.16	Trajectoria (a) e dinamica das variáveis (b) para SHAE-R. O ponto A indica o momento em que o hormônio começa a ser produzido pela segunda vez, e o ponto B indica o momento em que o robô aproxima-se da fonte luminosa, simbolizando a recarga de suas baterias.	97
4.17	Trajectoria do robô real controlado pelo SHAE (a) e trajetória do robô real controlado pelo SHAE-R (b). Em ambos os casos, o ponto A indica o momento em que o robô chaveia seu comportamento para fototaxia. No segundo caso, o ponto B indica o momento em que o robô encontra a fonte luminosa e consegue manter seu estado interno dentro do limite.	98
4.18	Comparação entre a distância média percorrida pelo robô em cada configuração experimental.	100
4.19	Dinâmica das variáveis considerando apenas a rede N1 e $\beta = 0,02$. A seta indica o momento em que o robô “morre”.	101
A.1	Mini-robô Khepera II (adaptado de www.k-team.com)	110
A.2	Esquemático do robô, com a disposição dos sensores infra-vermelhos e das rodas . . .	110
B.1	Ambiente de simulação do Simulador WSU Khepera Robot Simulator.	114
B.2	Interface para auxílio em experimentos com robô real e o simulador WSU.	115
B.3	Interface para auxílio às simulações do Capítulo 3.	116
B.4	Ambiente de simulação do Simulador KIKS.	117

Lista de Tabelas

3.1	Descrição dos parâmetros - SCAO	40
3.2	Descrição dos parâmetros - XCS	43
3.3	Situações Possíveis	48
3.4	Resultados do SCA para o Experimento 1. A tabela apresenta as 5 regras com mais energia dentre as 45 regras presentes na população.	50
3.5	Parâmetros do Experimento 1 referentes ao SCA	51
3.6	Resultados do XCS para o Experimento 1	51
3.7	Parâmetros do XCS para o Experimento 1	52
3.8	Resultados do SCA para o Experimento 2	57
3.9	Resultados do XCS para o Experimento 2	60

Capítulo 1

Introdução

Resumo - Neste capítulo, apresentam-se os conceitos básicos e motivações para o desenvolvimento do trabalho. As duas primeiras seções trazem os conceitos relativos à computação bio-inspirada, cognição, inteligência artificial e robótica, que são as linhas centrais no desenvolvimento desta tese. Faz-se uma breve introdução sobre cada tema, correlacionando-os. Em seguida, apresentam-se os objetivos e a estruturação do texto do trabalho.

1.1 Computação bio-inspirada

A observação da natureza parece guiar, de certa forma, o pensamento criativo e o trabalho humano. É inegável a inspiração biológica em criações como o avião, o sonar, as nadadeiras aquáticas, o velcro, entre outras. Documentos mostram que Turing e von Neumann, considerados pais da computação e precursores da ciência da complexidade, buscaram inspiração em fenômenos naturais presentes em sistemas biológicos, procurando associá-los à computação (Jefress, 1967; Copeland & Proudfoot, 2004). É possível compreender essa tentativa de vinculação recorrendo-se ao fato de que o estudo de sistemas complexos é intrínseco à biologia (Hall & Halliday, 1998; Purves et al., 2001).

Os computadores e as tarefas que eles desempenham estão se tornando cada vez mais avançados. Um sistema computacional moderno deve ser capaz não apenas de processar dados, mas ter competência em adaptar e otimizar suas ações (Michalewicz et al., 2006). Características próprias e

presentes em sistemas biológicos, como tendência à descentralização, capacidade de adaptação, escalabilidade, auto-organização e robustez, sugerem a viabilidade de unir biologia à computação (Hall & Halliday, 2004).

A progressiva busca por inspiração na biologia como metáfora para a criação de algoritmos computacionais voltados à solução de problemas é denominada hoje de computação bio-inspirada ou computação biologicamente inspirada (Sipper, 2002; de Castro & Von Zuben, 2004; Hall & Halliday, 2004).

Diversas ferramentas de computação bio-inspirada têm sido aplicadas com sucesso a problemas considerados complexos. Destacam-se as redes neurais artificiais (Haykin, 1999), os sistemas endócrinos artificiais (Neal & Timmis, 2003; Vargas et al., 2005), os algoritmos evolutivos e sistemas classificadores com aprendizado (Holland, 1975; Bäck et al., 2000a,b), os sistemas imunológicos artificiais (de Castro & Timmis, 2002), os algoritmos inspirados no comportamento coletivo e social de insetos e animais (Bonabeau et al., 1999; Kennedy et al., 2001) e os algoritmos baseados em processos de crescimento e desenvolvimento de organismos (Kumar & Bentley, 2003).

Além do estudo de cada ferramenta bio-inspirada e seus potenciais, existe uma tendência ao desenvolvimento de propostas que combinem as funcionalidades de múltiplas ferramentas. De fato, estudos biológicos comprovam que existe interconexão e dependência entre o sistema imunológico, o sistema nervoso e o sistema endócrino na espécie humana (Besendovsky & Del Rey, 1996). Tal interação mostra-se fundamental para a manutenção do equilíbrio interno do organismo.

A interação do agente com o ambiente e sua própria dinâmica interna podem resultar em comportamentos ricos e complexos. A ciência cognitiva busca compreender a forma como esses comportamentos complexos se manifestam, bem como estabelecer modelos de análise que possibilitem uma melhor interpretação e previsão dos resultados. A próxima seção explora os conceitos relativos a cognição, inteligência artificial e robótica no contexto da computação bio-inspirada.

1.2 Cognição, inteligência artificial e robótica

O conceito de *inteligência artificial* normalmente está relacionado a abordagens que buscam compreender ou criar sistemas capazes de expressar inteligência. Múltiplas definições formais podem ser

encontradas na literatura (Schank, 1987). Embora não exista uma definição única e suficientemente completa, a maioria procura vincular a inteligência artificial a formas de reproduzir em computador, em algum grau de fidelidade, o processo de raciocínio e o comportamento dos seres humanos através de computação (Schalkoff, 1990; Russell & Norvig, 2003).

Após a expansão do conceito de inteligência artificial, diversas linhas de pesquisa foram derivadas e compõem subáreas de atuação. A ciência cognitiva ou ciência cognitiva incorporada (Pfeifer & Scheier, 1999) busca investigar os princípios da inteligência, possuindo um vínculo muito forte com outras áreas do conhecimento, como a psicologia e a biologia.

A ciência cognitiva atua na compreensão da interação *mente-corpo-ambiente*, na determinação de comportamentos e de fenômenos complexos, como categorização, aprendizado e memória, empregando principalmente teorias de sistemas dinâmicos e de sistemas auto-organizáveis. Argumenta-se que a auto-organização de processos dinâmicos internos ao agente é o ponto crucial para o entendimento e reprodução de sistemas cognitivos (Barandiaran & Moreno, 2006; Montebelli et al., 2008).

A definição de inteligência, por sua vez, é um ponto polêmico. Não cabe ao presente trabalho discutir as motivações por trás de cada interpretação, mas sim estabelecer uma visão que será adotada como padrão na sequência do texto. Portanto, aborda-se inteligência como sendo uma relação entre o corpo, a mente e o ambiente no qual o indivíduo está inserido. De acordo com Pfeifer & Scheier (1999), o grau de inteligência de um indivíduo está diretamente relacionado à forma e ao refinamento com que ele é capaz de representar as situações em que esteja envolvido, determinar objetivos específicos a partir de objetivos genéricos, tomar decisões e modificar seu comportamento a partir da interação com o ambiente externo. Desta forma, busca-se o desenvolvimento de ferramentas computacionais bio-inspiradas que apresentem algum grau de inteligência em seu processamento de informação.

O estudo da robótica é considerado muito relevante para o desenvolvimento de novas tecnologias e da sociedade (Andrew, 1975; Bekey, 2005). A razão para a escolha de experimentos com robôs autônomos fundamenta-se nos argumentos empregados por Pfeifer & Scheier (1999) e Boden (2006), que apontaram a robótica autônoma móvel como um paradigma apropriado para se estudar os princípios da cognição. De acordo com Ross e colaboradores (Ross et al., 2003), um sistema cognitivo deve possuir aprendizado contínuo e ser capaz de alterar seu comportamento através da experiência,

sendo essas características essenciais para o desenvolvimento perene do sistema. Agentes robóticos autônomos são considerados agentes capazes não apenas de interagir com um ambiente, mas também de aprender e de se adaptar, gerando suas próprias regras de comportamento em tempo real de execução. A motivação de utilizá-los reside no fato de que estes agentes podem apresentar comportamentos emergentes, isto é, comportamentos que surgem da interação robô-ambiente, sem terem sido programados a priori pelo projetista.

Essas manifestações de comportamento não se restringem a apenas um único robô. Nolfi & Floreano (2000), Crestani (2001), Bowling & Veloso (2003), Izquierdo (2004) e Cazangi et al. (2005), dentre outros estudiosos do assunto, mostram que processos de navegação autônoma envolvendo múltiplos robôs podem promover a emergência de comportamentos organizados que não são expressos quando estes robôs estão isolados dos demais.

Paralelamente ao estudo de processos cognitivos, a robótica autônoma móvel também se destaca por sua vasta aplicação prática em ambientes hostis e de difícil acesso, por exemplo, robôs combatentes de incêndio, resgatadores de vítimas, exploradores submarinos e espaciais, desarmadores de bombas. Esses cenários são desafiadores por representarem uma aplicação crítica em tempo real com diversos fatores complicantes: múltiplos sensores sujeitos a ruído, ambientes dinâmicos e desconhecidos e critérios de desempenho possivelmente conflitantes a serem satisfeitos simultaneamente.

O projeto de controladores para a navegação autônoma de robôs é um problema dinâmico e de grande complexidade (Figueiredo, 1999). A programação direta pelo projetista nem sempre produz resultados satisfatórios, visto que o número de variáveis do ambiente e do robô a serem tratadas é elevado e pouco domínio se tem acerca de suas inter-relações. Para o desenvolvimento de sistemas robóticos robustos e eficientes, técnicas de aprendizado de máquina aparecem como uma opção atrativa e inovadora, levando o robô a solucionar determinada tarefa por meio da adaptação de seu comportamento, sem intervenção externa.

Algoritmos de aprendizado de máquina podem possuir formas diferentes de aprendizado, ou seja, a forma como o sistema adquire e incorpora o conhecimento pode variar segundo alguns paradigmas de aprendizagem (Pfeifer & Scheier, 1999). Os mais observados na área de robótica cognitiva são brevemente descritos a seguir:

Aprendizado Supervisionado: caracteriza-se por uma aprendizagem através da comparação com exemplos fornecidos externamente por um sistema supervisor, ou seja, existe um conhecimento prévio sobre a saída esperada para cada opção de entrada. Como consequência, o sistema, após o período de aprendizagem, não é capaz de adaptar-se a mudanças no mapeamento entrada-saída e, portanto, deixa de expressar a capacidade de aprender com a experiência. Conclui-se que técnicas de aprendizado supervisionado, isoladas, são de pouco interesse para o desenvolvimento de sistemas complexos (Pfeifer & Scheier, 1999).

Aprendizado Não-Supervisionado: também conhecido como Aprendizado Auto-Organizado, possui como principal característica a aprendizagem por correlação entre os dados, não havendo um sistema supervisor externo (Pfeifer & Scheier, 1999). Dessa forma, os dados de entrada são utilizados pelo sistema para determinar padrões ou categorias, e cada novo dado de entrada é associado a uma classe ou padrão já determinado.

Aprendizado Por Reforço: caracteriza-se por um mapeamento entrada-saída de tal forma que um sinal de reforço (função de recompensa) seja maximizado. A cada passo, ou a cada série de passos, o sistema recebe um sinal com base na performance de suas decisões, não se conhecendo a priori qual a melhor opção a cada situação encontrada no problema (Sutton & Barto, 1998).

Aprendizado Evolutivo: a abordagem evolutiva pode ser considerada uma forma de aprendizado, como destaca Nolfi & Floreano (2000). É uma abordagem que implementa auto-organização, porém com um nível de supervisão inferior a outras técnicas, uma vez que a função de desempenho (*fitness*) apenas fornece uma medida relativa de quão boa uma proposta de solução é para o problema.

De acordo com Pfeifer & Scheier (1999), um sistema que apresente aprendizagem, mais especificamente aprendizagem robótica, deve possuir as seguintes características:

- capacidade de atenuar ruídos;
- rápida convergência;
- aprendizado *on-line*, ou seja, aprendizado em tempo de execução;
- aprendizado incremental, isto é, o processo de aprendizagem nunca deve ser interrompido;
- tratabilidade, ou capacidade de execução em tempo real;

- o sistema deve utilizar apenas informações obtidas pelos sensores do robô, sem incorporar nenhum conhecimento externo ao agente.

Considerando os paradigmas de aprendizado de máquina apresentados e as características que um sistema de aprendizagem robótico deve possuir, percebe-se uma tendência da literatura em utilizar principalmente variações de técnicas de aprendizado por reforço e aprendizado evolutivo, justamente por elas apresentarem intrinsecamente as melhores perspectivas para a robótica cognitiva (Arkin, 1998; Nolfi & Floreano, 2000). Nota-se uma contínua busca para que um agente robótico seja capaz de adaptar-se a uma série de eventos imprevistos, em diversos níveis de atuação.

Em 1952, Ashby propôs uma teoria para explicar como um sistema ou organismo se auto-regula visando manter a homeostase (Ashby, 1952). A homeostase pode ser definida como um estado de equilíbrio interno do organismo e de suas variáveis essenciais ou de seu meio. Desta forma, a inspiração biológica aliada à teoria apresentada por Ashby motivaram a aplicação da homeostase na síntese de sistemas autônomos em robótica móvel (Di Paolo, 2000; Dyke & Harvey, 2005; Vargas et al., 2005; Moiola et al., 2008), com o objetivo de desenvolver controladores que apresentem adaptação.

A próxima seção descreve os objetivos e contribuições desta tese, bem como a estruturação do texto.

1.3 Objetivos e organização do texto

Os objetivos principais deste trabalho são estudar, desenvolver e aplicar ferramentas computacionais bio-inspiradas em navegação autônoma de robôs, visando compreender melhor os mecanismos de aprendizado, memória e adaptação. Mais especificamente, o trabalho propõe:

1. empregar modelos e extensões de Sistemas Classificadores com Aprendizado (*Learning Classifier Systems*, LCS, na literatura em inglês). Originalmente propostos por Holland (1975), os sistemas classificadores com aprendizado têm se destacado na aplicação em robótica autônoma móvel e sistemas cognitivos. Estes sistemas são capazes de combinar evolução e aprendizado de forma integrada, sendo considerados modelos apropriados para a síntese de sistemas complexos adaptativos (Holland, 1995) e modelos de processos de inferência em sistemas cognitivos (Holland et al., 1986; Booker et al., 1989);

2. um novo modelo de sistema homeostático artificial evolutivo, composto por redes neurais artificiais recorrentes do tipo NSGasNet (Vargas et al., 2007) e um sistema endócrino artificial. Esta etapa do trabalho enquadra-se no contexto da robótica evolutiva, uma técnica para a criação de robôs autônomos inspirada no princípio da evolução das espécies de Darwin, que promove a seleção e reprodução dos indivíduos mais adaptados ao ambiente. A robótica evolutiva é uma área em forte ascendência, com diversos trabalhos bem sucedidos relatados na literatura (Nolfi & Floreano, 2000).

Esta tese divide-se em 5 capítulos e 2 apêndices.

O Capítulo 2 introduz e detalha as ferramentas de inteligência computacional a serem utilizadas nos demais capítulos: computação evolutiva e redes neurais artificiais. São contemplados principalmente os algoritmos genéticos, utilizados como base para os sistemas classificadores com aprendizado e para a evolução de redes neurais artificiais, e as redes neurais artificiais recorrentes do tipo NSGasNet, as quais representam um novo modelo proposto por Vargas et al. (2007), que já expressa por si só algum tipo de homeostase e emprega evolução.

O Capítulo 3 concentra-se no estudo de sistemas classificadores com aprendizado aplicados a tarefas robóticas reativas e não-reativas. Faz-se um estudo do processo de evolução dos indivíduos e da população final obtida. Para o desenvolvimento do estudo, utilizou-se uma versão do sistema classificador com aprendizado original (Holland, 1975) e uma versão do sistema classificador baseado em precisão, XCS (*eXtend Classifier System*, na literatura em inglês) (Wilson, 1995). Empregam-se também mecanismos que permitem aos Sistemas Classificadores lidarem com tarefas que demandem para sua resolução algum tipo de comportamento não-reativo.

No Capítulo 4, propõe-se um modelo de sistema homeostático artificial evolutivo. Este sistema é inspirado no trabalho de Vargas et al. (2005), e é formado por redes neurais artificiais recorrentes do tipo NSGasNet e um sistema endócrino artificial. O ajuste dos parâmetros do sistema é feito por meio de evolução, reduzindo-se a necessidade de codificação e parametrização a priori. Para a análise do modelo, de suas peculiaridades e de sua capacidade de adaptação, são realizados experimentos simulados com um robô do tipo Khepera II.

O Capítulo 5 apresenta uma discussão geral sobre os resultados obtidos nos capítulos anteriores e reforça as principais contribuições da dissertação. Também apresentam-se perspectivas futuras tanto

para a área de robótica cognitiva como para o emprego das ferramentas desenvolvidas em outras áreas de possível atuação.

Por fim, o Apêndice A descreve o robô Khepera II, utilizado nos experimentos simulados e reais, e o Apêndice B comenta sobre os simuladores escolhidos.

Capítulo 2

Sistemas de inteligência computacional a serem adotados

Resumo - Este capítulo fornece as bases de duas ferramentas computacionais bio-inspiradas, definidas dentro do contexto de sistemas de inteligência computacional. A descrição inicia-se com os fundamentos da Computação Evolutiva, focando nas características dos Algoritmos Genéticos. Em seguida, as Redes Neurais Artificiais serão descritas através de seus principais componentes e características, incluindo a abordagem utilizada nesse trabalho, denominada NSGasNet.

2.1 Computação Evolutiva

Com base nas observações de Charles Darwin e sua subsequente proposta da Teoria da Evolução e Seleção Natural (Darwin, 1859), as ciências biológicas encontraram alternativas para explicar a forma como populações surgem, interagem, reproduzem-se e adaptam-se ao ambiente. De acordo com essas hipóteses, um indivíduo melhor adaptado às condições em que está exposto tende a obter vantagem reprodutiva e de sobrevivência, propagando, desta forma, seus genes pelas gerações futuras. Assim, a população como um todo tende a possuir uma parte expressiva de sua carga genética relativa aos indivíduos melhor adaptados.

A observação da forma como os indivíduos de uma população interagem e o modo como são selecionados e transmitem suas características à geração seguinte, adaptando-se a ambientes dinâmicos, inspirou o desenvolvimento de algoritmos computacionais baseados na Teoria de Evolução e Seleção Natural.

A Computação Evolutiva (CE) (Holland, 1975; Schwefel, 1981; Goldberg, 1989; Michalewicz, 1996), que incorpora propostas alternativas de Algoritmos Evolutivos, caracteriza-se basicamente por manipular uma população de soluções candidatas e promover a reprodução (proposta de novas soluções com base nas soluções atuais), sujeita a variações aleatórias nas características de cada indivíduo da nova geração. O ambiente que exerce pressão sobre os indivíduos é geralmente associado às demandas de um problema, sendo que a aptidão de um indivíduo está diretamente vinculada ao grau de atendimento das demandas, razão pela qual cada indivíduo recebe a denominação de solução candidata. Na presença de indivíduos com aptidões diferentes, mecanismos de seleção que privilegiam os indivíduos mais aptos e reprodução com variações aleatórias, a limitação do tamanho da população tende a produzir um aumento médio da aptidão da população ao longo das gerações como resultado da competição sob recursos limitados.

Dessa forma, a Computação Evolutiva é uma técnica bio-inspirada de busca iterativa e paralela, adequada a situações em que o número de possibilidades é computacionalmente proibitivo para técnicas de busca e otimização tradicionais, além de ser capaz de tratar problemas multimodais ou com ausência de diferenciabilidade do critério de otimalidade (Michalewicz & Fogel, 2000).

Dentro da Computação Evolutiva, destacam-se os Algoritmos Genéticos (AG) (Holland, 1975), as Estratégias Evolutivas (Rechenberg, 1973; Schwefel, 1965), a Programação Evolutiva (Fogel, 1962) e a Programação Genética (Koza, 1992). O foco do presente trabalho será nos Algoritmos Genéticos.

2.1.1 Algoritmos Genéticos

John Holland (Holland, 1975) foi o pioneiro a propor uma formalização matemática para os Algoritmos Genéticos. Holland inspirou-se na forma como indivíduos reproduzem-se e transmitem sua carga genética para a geração seguinte, criando mecanismos específicos para a codificação e manipulação de soluções. A Figura 2.1 ilustra as principais etapas de um algoritmo genético: geração da população inicial, avaliação, seleção e reprodução com variação genética.

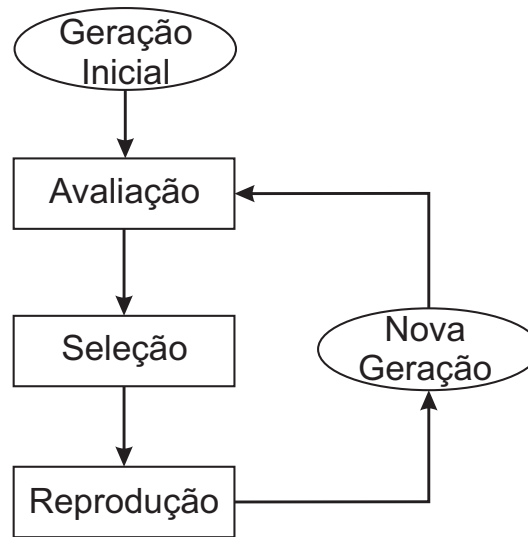


Fig. 2.1: Principais etapas de um algoritmo genético

Geração da população inicial e codificação dos indivíduos

Um algoritmo genético inicia-se com uma população inicial, que deve conter propostas de solução para o problema. Cada execução do algoritmo culmina na proposta de uma nova população, chamada de nova geração.

A codificação de uma proposta de solução, que faz parte da população a cada geração, é específica para o tipo de problema, e denominada genótipo. Um genótipo pode ser representado por um vetor de atributos, e cada posição deste vetor é denominada locus. Cada locus pode assumir um valor (alelo) de acordo com o tipo de codificação adotada, normalmente binária, inteira ou em ponto flutuante. A etapa de codificação é decisiva, pois irá influir diretamente na natureza do espaço de exploração, na complexidade da busca e no número de soluções presentes. A Figura 2.2 ilustra uma representação genética de um cromossomo em codificação binária.

O fenótipo de uma proposta de solução pode ser considerado como o conjunto observável de propriedades e características dessa solução, delineadas pelo tipo de problema a ser tratado. Por exemplo, em um problema que visa otimizar as características de uma antena, o fenótipo poderia ser representado pelo conjunto global de aspectos construtivos da antena, e o genótipo seria formado por campos relativos a cada parte constituinte da antena (Figura 2.3).

Existe uma necessidade de conversão do código expresso pelo genótipo para a característica física observável. Essa conversão é ditada pela natureza do problema. No mesmo caso da antena, supondo-

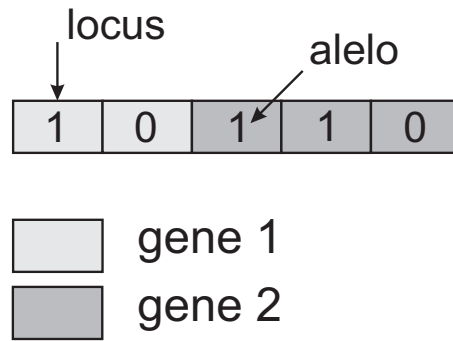


Fig. 2.2: Representação genética de um cromossomo em codificação binária. Os genes podem ter tamanho variável, e cada locus (posição) no gene possui um alelo (valor) com base na codificação adotada

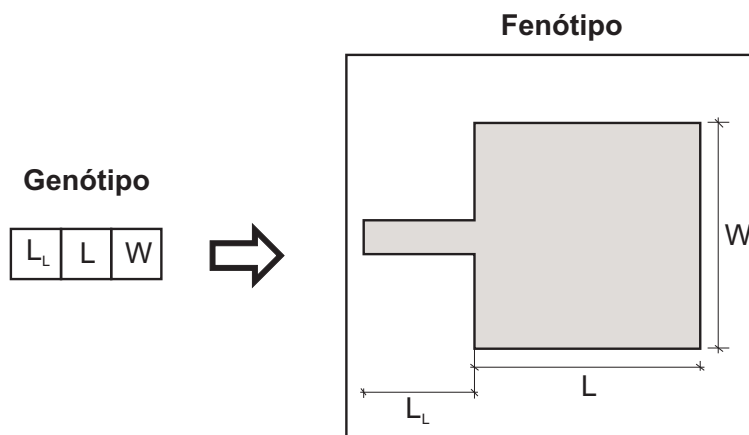


Fig. 2.3: Relação genótipo-fenótipo (reproduzido de Barra (2007), com permissão do autor). Cada campo do genótipo codificado possui uma interpretação na formação da proposta de solução.

se uma codificação binária, o valor de um campo do genótipo correspondente a “1” poderia indicar “*forma redonda*”, enquanto que um valor “0” poderia indicar “*forma quadrada*”.

Avaliação dos indivíduos

A cada ciclo de aplicação do algoritmo genético, uma proposta de solução que consta na população (indivíduo) é avaliada de acordo com o desempenho de suas características fenotípicas na resolução do problema. Essa avaliação é normalmente feita com base em uma função de desempenho, chamada *função de fitness*, que mede o grau relativo de aptidão do indivíduo. A função de *fitness* deve guiar o processo evolutivo na direção da solução esperada para o problema, procurando diferenciar cada proposta de solução de maneira relativa, e assim criando um diferencial de qualidade dentre as soluções. Por exemplo, para um problema do tipo *caixeiro viajante* (Applegate et al., 2006),

em que se busca o percurso de comprimento mínimo, a função de *fitness* poderia ser definida como o inverso da distância associada a cada percurso.

Seleção com base no valor de fitness

A próxima etapa, seleção, é feita levando-se em conta o valor de *fitness* obtido por cada indivíduo. Os métodos mais usuais de seleção são (Fogel, 1994; Michalewicz, 1996; Miller & Goldberg, 1996):

- roleta (*roulette wheel*, na literatura em inglês): obtém-se a soma de todos os valores de *fitness* da população e a probabilidade de escolha de um indivíduo é proporcional ao valor de seu *fitness* em relação a essa somatória;
- elitista: a cada aplicação do algoritmo, o melhor indivíduo da população é escolhido junto com algum outro membro da população para reprodução. O critério de escolha deste outro membro pode variar, bem como a forma como o indivíduo resultante do processo reprodutivo será inserido na população. Mantém-se o indivíduo de maior *fitness* na geração seguinte, evitando-se que bons indivíduos desapareçam da população evoluída devido aos operadores genéticos;
- torneio: selecionam-se aleatoriamente alguns indivíduos da população, e dentre esses escolhidos, escolhe-se um ou dois de maior *fitness* para integrar o processo reprodutivo. O tamanho do conjunto de indivíduos escolhidos pode variar. Essa abordagem é mais robusta frente ao problema de ausência de diversidade (um bom indivíduo espalha-se rapidamente pela população);
- diversidade: escolhem-se os indivíduos mais diversos da população;
- bi-classista: selecionam-se os $P\%$ melhores indivíduos e os $(100 - P)\%$ piores.

Reprodução com mutação gênica

No passo seguinte, referente à reprodução, operadores de recombinação gênica (*crossover*, na literatura em inglês) e mutação são aplicados junto aos pais selecionados, visando obter filhos mais adaptados e que herdem características que tornam seus pais boas propostas para determinada situação do problema.

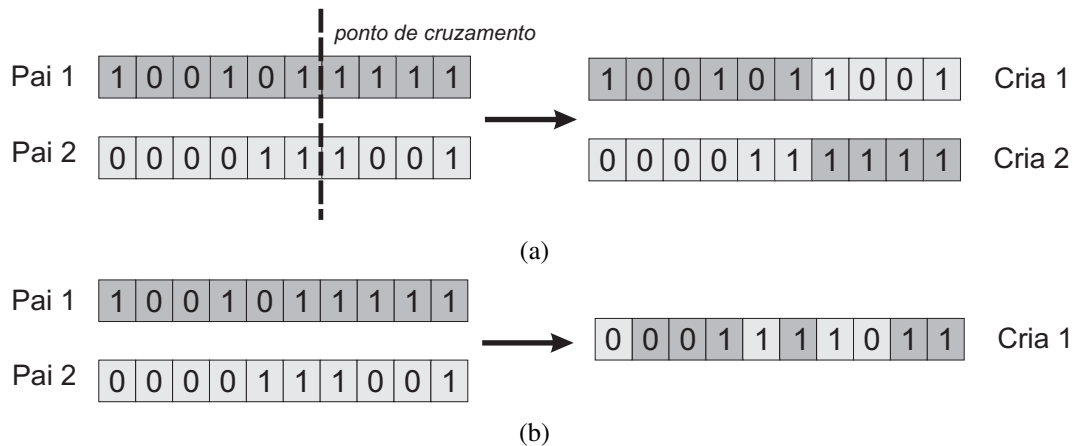


Fig. 2.4: Crossover de um ponto (a) e crossover uniforme (b) (reproduzido de Barra (2007), com permissão do autor)

O operador de *crossover* combina o conteúdo genético de dois pais para a geração de um ou mais filhos. As formas de combinação mais usuais são (Holland, 1975; Bäck et al., 2000a,b):

- um ponto (Figura 2.4(a)): escolhe-se um ponto de corte no genótipo dos dois pais, e geram-se novos filhos a partir da troca entre os segmentos cortados;
- múltiplos pontos: semelhante ao operador anterior, porém neste caso mais de um ponto é escolhido para corte e permutação;
- uniforme (Figura 2.4(b)): ao invés de pontos de corte, cada posição do genótipo do filho é determinado por um campo de um dos dois pais, escolhido aleatoriamente. A composição é, portanto, feita campo a campo, e não mais através de trechos dos genótipos dos pais. Essa estratégia é indicada quando a ordem dos elementos do vetor (campos) não é relevante.

Os operadores de mutação visam perturbar uma posição aleatória do genótipo, mantendo a diversidade (Holland, 1975). A manutenção da diversidade, ou seja, a variabilidade de indivíduos em uma população, é importante por impedir que uma proposta de solução, que pode não ser a melhor para o problema, mas é superior a todas as propostas de solução que compõem a população corrente, domine toda a população e impeça que novas soluções sejam geradas. Normalmente, o valor da taxa de mutação é pequeno, seja para evitar a perda de boas soluções já obtidas (devido a grandes saltos descorrelacionados no processo de busca) ou então para refinar uma solução (considerando que o

passo no processo de busca devido ao operador de crossover seria pouco efetivo em buscas locais e refinadas). Sendo assim, os operadores mais tradicionais simplesmente trocam ou perturbam o valor do alelo por um outro valor dentro do intervalo de variação possível (Figura (2.5)).



Fig. 2.5: Operador de mutação (reproduzido de Barra (2007), com permissão do autor)

Os filhos gerados por esse processo reprodutivo com mutação gênica são reinseridos na população, podendo ou não substituir seus pais. Com isso, uma nova geração será formada e o algoritmo reinicia seu ciclo.

Critério de parada

O critério de parada do algoritmo genético pode ser definido pelo valor de *fitness* do melhor indivíduo da população: uma vez superado o nível mínimo desejado, interrompe-se o ciclo e a solução será representada por esse indivíduo. Supõe-se aqui que haja uma tendência de crescimento do *fitness* ao longo das gerações. Pode-se também determinar um número máximo de gerações, e selecionar o indivíduo de maior *fitness* ao final do processo. Existem muitas outras possibilidades para o critério de parada, incluindo aquelas baseadas em grau de diversidade na população ou então na taxa de progresso do *fitness* do melhor indivíduo (Bäck et al., 2000a,b).

Essa seção apresentou conceitos básicos sobre os algoritmos genéticos. Os experimentos descritos no Capítulo 3 e no Capítulo 4 empregam variações de algoritmos genéticos, devidamente explicadas em cada capítulo. A próxima seção aborda os principais conceitos acerca das Redes Neurais Artificiais.

2.2 Redes Neurais Artificiais

Dentre todas as abordagens de computação bio-inspirada, as redes neurais artificiais são as que mais se destacaram. Elas surgiram com o modelo matemático do neurônio biológico proposto por

McCulloch e Pitts em 1943 e passaram por vários estágios de desenvolvimento (Haykin, 1999). Nos últimos anos, basicamente em virtude da incorporação de resultados relevantes derivados da estatística e da teoria de aproximação de funções, as redes neurais vêm recebendo um reconhecimento amplo junto às mais variadas áreas de processamento de informação e análise de dados.

De forma simplificada, as redes neurais podem ser projetadas executando-se três passos básicos: definição de um modelo de neurônio artificial, o qual contém parâmetros ajustáveis denominados pesos de conexão; escolha de uma arquitetura de rede (as mais comuns são as redes sem realimentação e em camadas); e adoção de um algoritmo de treinamento ou aprendizagem para determinar uma configuração apropriada para a rede (ajuste de pesos e, em alguns casos, de arquitetura também).

A interação ou base de troca de informações entre os neurônios artificiais se dá por sinais numéricos, os quais são transmitidos entre neurônios através de conexões ou sinapses. Essas sinapses podem ser excitatórias ou inibitórias, tendo associado a si um peso sináptico ajustável. As sinapses podem ser transmitidas em único sentido, de camada para camada, ou possuir realimentações entre camadas, caracterizando uma rede neural recorrente. Notoriamente, as redes neurais recorrentes são capazes de representar uma grande variedade de comportamentos dinâmicos e memória (Pineda, 1987; Santos & Von Zuben, 2000). Entre os tipos mais comuns de arquiteturas de redes neurais, destacam-se as redes perceptron multi-camadas (Minsky & Papert, 1969), redes de Hopfield (Hopfield, 1982), redes de Kohonen (Kohonen, 1990), e máquinas de vetores-suporte (Vapnik, 1995).

A Figura 2.6 ilustra um modelo de um neurônio de índice k . Sua estrutura é composta por entradas $x_1, \dots, x_m$, ponderadas por pesos $w_{k1}, \dots, w_{km}$, juntamente com uma entrada de polarização b_k , também denominada *bias*. Ao resultado dessa combinação linear, aplica-se uma função de ativação f (aqui representada pela função tangente hiperbólica), que mantém a saída y_k do neurônio limitada. Matematicamente, o bloco pode ser descrito pela Equação 2.1:

$$y_k = \tanh\left[\left(\sum_{j=1}^m w_{kj}x_j\right) + b_k\right] \quad (2.1)$$

Na representação gráfica das redes, formada por um agregado de neurônios, é usual omitir os pesos, o bloco somador e a função de ativação, e deixar explícitas apenas as conexões entre os neurônios, o que irá definir o tipo de arquitetura da rede. A Figura 2.7 ilustra uma rede neural artificial do tipo MLP totalmente conectada.

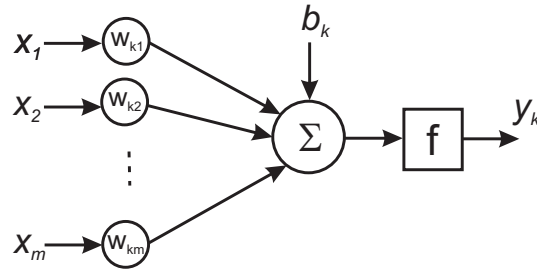


Fig. 2.6: Modelo de um neurônio artificial

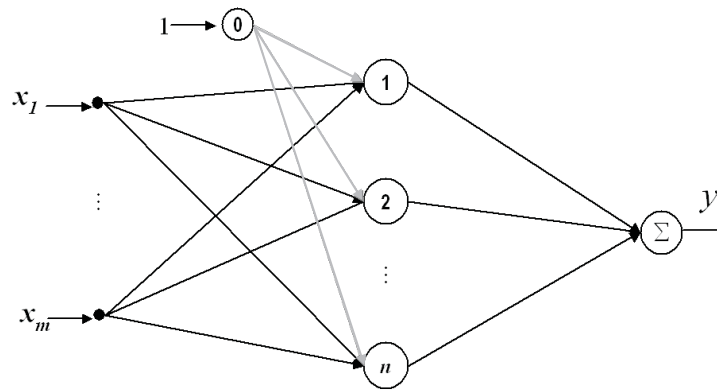


Fig. 2.7: Rede neural artificial do tipo MLP totalmente conectada.

O processo de aprendizagem de uma rede consiste em ajustar os pesos (intensidade de conexão entre neurônios), podendo modificar também o número de neurônios e o padrão de conexão entre neurônios (topologia). As mais utilizadas formas de aprendizagem são (Haykin, 1999):

- supervisionada, quando se tem conhecimento da saída desejada referente a cada conjunto de entradas, e espera-se que o processo de treinamento minimize o erro entre a saída obtida e a desejada. Destaca-se o algoritmo de retropropagação (*backpropagation*, na literatura em inglês).
- não-supervisionada, quando não existe uma saída desejada e a rede neural deve se auto-organizar, de modo que neurônios vizinhos sejam ativados por estímulos de entrada similares.

A plasticidade sináptica, ou seja, a capacidade de modificação das sinapses, é tida como fundamental para a maioria dos modelos artificiais do sistema nervoso, de redes neurais (McClelland et al., 1986; Rumelhart et al., 1986; Haykin, 1999) a outros modelos computacionais com base em neurociência (Churchland & Sejnowski, 1994; O'Reilly & Munakata, 2000; Dayan & Abbot, 2001). Com

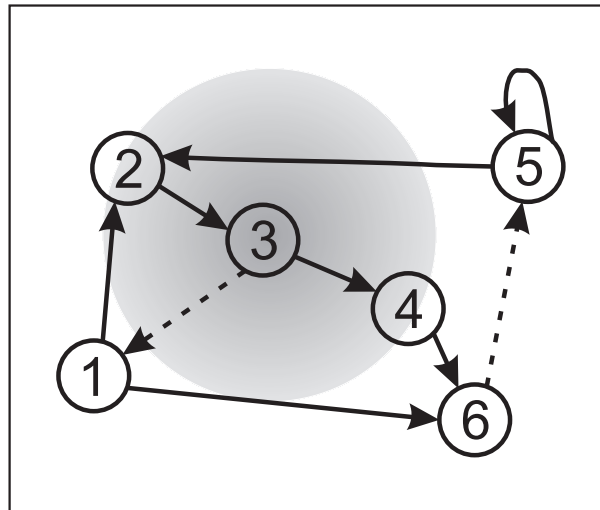


Fig. 2.8: Modelo de uma rede GasNet composta por 6 nós. As conexões sinápticas e gasosas estão ilustradas. Linhas cheias representam conexões elétricas excitatórias e linhas pontilhadas representam conexões elétricas inibitórias. Note que o nó 3 emite uma nuvem de gás e pode modular os nós 2 e 4.

base nesta plasticidade sináptica, modelos tradicionais de redes neurais artificiais foram estendidos. A próxima seção explora as propriedades de uma dessas novas propostas, conhecida como GasNet.

2.3 GasNets

Em sua concepção, a GasNet (Husbands et al., 1998) procura reproduzir a produção e liberação de óxido nítrico (NO) por neurônios reais (Gally et al., 1990b; Holscher, 1997), buscando modular o comportamento de outros neurônios. Esta modulação atua na função de ativação do neurônio, modificando seu comportamento a médio e longo prazo, sendo responsável pelo caráter plástico da arquitetura e tornando-a capaz de adaptação dinâmica (Philippides et al., 2005a,b).

A GasNet é modelada como uma rede neural recorrente com número de nós fixo ou variável, onde cada nó pode produzir estímulos sinápticos, excitatórios ou inibitórios, a outros nós aos quais esteja conectado, e também estímulos químicos, através de gases artificiais, a outros nós que estejam dentro de um determinado raio de ação desses gases. Portanto, a GasNet original trabalha com uma relação espacial entre os neurônios, isto é, a dispersão gasosa atua sobre uma vizinhança, e cada neurônio tem uma posição espacial pré-determinada. Geralmente, as relações espaciais entre neurônios são definidas em duas dimensões. Para a determinação dos parâmetros da rede, empregam-se técnicas

de computação evolutiva, descritas no início deste capítulo e detalhadas no contexto de GasNets ao final desta seção. A Figura 2.8 ilustra uma GasNet com 6 nós e suas respectivas conexões sinápticas e químicas.

2.3.1 Dinâmica e Treinamento da GasNet

Existem dois gases no modelo original da GasNet: gás tipo 1 e gás tipo 2. Um neurônio pode emitir gás tipo 1, tipo 2, ou nenhum tipo de gás. A emissão de gás pode ter início se a ativação elétrica do neurônio for superior a um limiar geneticamente pré-determinado ou se a estimulação gasosa for superior a um limiar geneticamente pré-determinado.

A estimulação gasosa da GasNet segue um padrão determinado pelas Equações 2.2 a 2.4. A concentração de gases a partir de um nó emissor obedece um padrão gaussiano (Equação 2.2), influenciado pela função $T(t)$ (Equação 2.3 e Figura 2.9). Note que a função $T(t)$ influencia linearmente a altura do padrão gaussiano, aumentando-o ou diminuindo-o, dependendo se o nó está ou não emitindo gás. A determinação da emissão de gás por um nó se dá através do parâmetro TE , descrito a seguir na seção 2.3.2, e sofre influência da atividade elétrica e da concentração gasosa no nó.

A concentração de gás total em um neurônio da rede é obtida somando-se as concentrações correspondentes a cada neurônio emissor.

$$C(d, t) = \begin{cases} C_0 \times e^{-(d/r)^2} \times T(t) & , d < r \\ 0 & , \text{alhores} \end{cases} \quad (2.2)$$

$$T(t) = \begin{cases} H\left(\frac{t-t_e}{s}\right) & , \text{nó emitindo gás} \\ H\left(H\left(\frac{t_s-t_e}{s}\right) - H\left(\frac{t-t_s}{s}\right)\right) & , \text{nó não emitindo gás} \end{cases} \quad (2.3)$$

$$H(x) = \begin{cases} 0 & , x \leq 0 \\ x & , 0 < x < 1 \\ 1 & , \text{alhores} \end{cases} \quad (2.4)$$

em que

$C(d, t)$ é a concentração de gás a uma distância d do neurônio no instante t ;

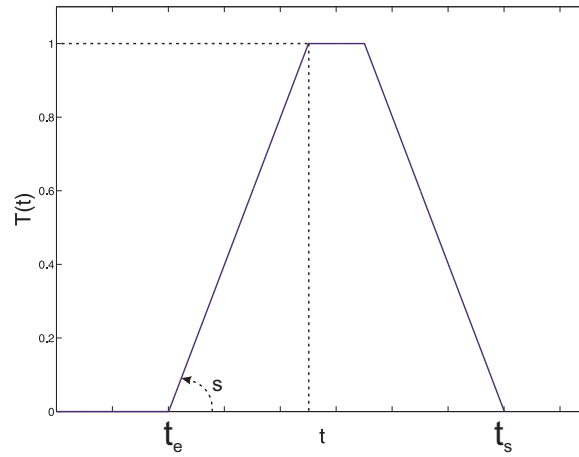
C_0 é uma constante pré-determinada;

r é o raio de influência dos gases do neurônio;

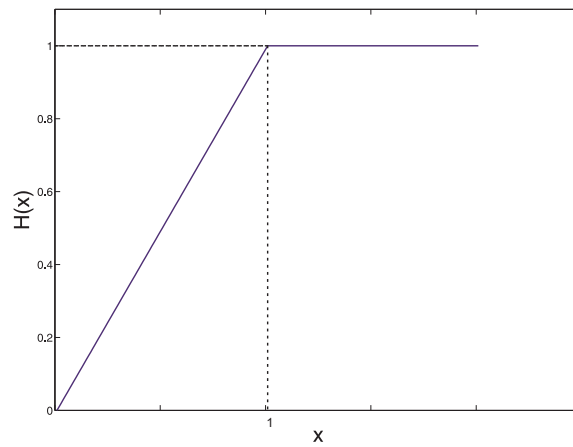
$T(t)$ e $H(x)$ modelam a elevação e decaimento no tempo da concentração de gás emitido a partir de um nó;

t_e e t_s são o último instante que o neurônio começou e interrompeu a emissão de gás, respectivamente;

s é a constante relativa ao crescimento e decaimento da emissão de gás a cada instante t .



(a)



(b)

Fig. 2.9: Função $T(t)$ (a) e função $H(x)$ (b). Note que a função $T(t)$ possui um máximo e satura com valor 1.

A modulação gasosa tem seu efeito representado pelo parâmetro K_i^t , cujo valor é determinado pelo conjunto de Equações 2.5 a 2.8. A cada iteração t , o neurônio referenciado pelo índice i terá sua saída modulada de acordo com a concentração de gases em sua vizinhança. Um aumento da concentração de gás tipo 1 (C_1^t) aumenta o valor de K_i^t , e um aumento da concentração de gás tipo 2 (C_2^t) diminui o seu valor. O valor inicial de K_i^t depende do valor do parâmetro geneticamente determinado D_i^0 . É importante ressaltar que este modelo de difusão e modulação gasosa é bastante simplificado quando comparado à realidade biológica (Santos, 2001), mas esse fato não anula a capacidade do modelo de incorporar propriedades interessantes da modulação gasosa observada em sistemas biológicos reais (Gally et al., 1990a; Changeux, 1993; Wood & Garthwaite, 1994; Lancaster, 1996).

$$K_i^t = P[D_i^t] \quad (2.5)$$

$$P = \{-4, 0; -2, 0; -1, 0; -0, 5; -0, 25; 0, 0; 0, 25; 0, 5; 1, 0; 2, 0; 4, 0\} \quad (2.6)$$

$$D_i^t = f \left(D_i^0 + \frac{C_1^t}{C_0 \times K} (N - D_i^0) - \frac{C_2^t}{C_0 \times K} D_i^0 \right) \quad (2.7)$$

$$f(x) = \begin{cases} 0, & x \leq 0 \\ \lfloor x \rfloor, & 0 < x < N \\ N - 1, & \text{alhores} \end{cases} \quad (2.8)$$

em que

i é o índice que referencia o neurônio em questão;

$P[i]$ é o conjunto de valores que K_i^t pode assumir no vetor de N posições;

D_i^0 é o valor geneticamente determinado de D_i^t no instante $t = 0$;

C_1^t e C_2^t são as concentrações dos gases 1 e 2 no tempo t , respectivamente;

C_0 e K são constantes pré-determinadas.

A saída de um neurônio de uma rede do tipo GasNet é dada pela Equação 2.9. A cada instante t , a saída da rede é função das entradas elétricas e da modulação gasosa determinada pelo parâmetro

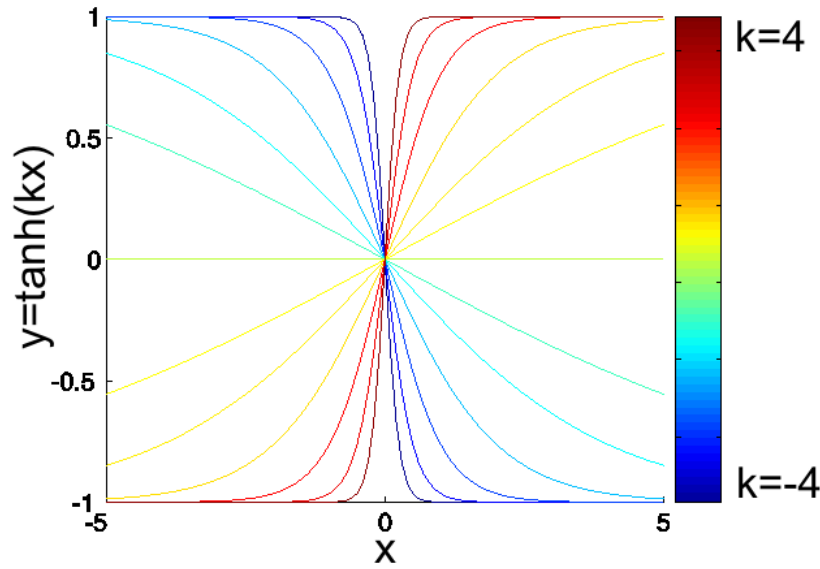


Fig. 2.10: Comportamento das curvas $\tanh(kx)$ para diversos valores de k

K_i^t . A Figura 2.10 ilustra o comportamento das curvas $\tanh(kx)$ para diversos valores de k . Através da análise dessa família de curvas, que retrata o papel que a modulação gasosa pode exercer sobre o comportamento de cada neurônio, pode-se observar que de fato dinâmicas bastante variadas podem ser obtidas.

$$O_i^t = \tanh \left[K_i^t \left(\sum_{j \in C_i} w_{ji} O_j^{t-1} + I_i^t \right) + b_i \right] \quad (2.9)$$

em que

i e j são os índices que referenciam os neurônios da rede;

O_j^{t-1} é a saída do neurônio j no instante $t - 1$;

K_i^t representa a modulação da função de transferência causada pelos gases;

C_i é o conjunto de nós conectados ao nó i ;

w_{ji} é o valor do peso da conexão (no intervalo $[-1, 1]$);

I_i^t é a entrada externa do neurônio i no instante t , caso o nó possua entradas externas;

b_i é o valor de bias do neurônio.

Existem diversas propostas para a determinação das conexões elétricas de determinado nó (Hus-

bands et al., 1998). Pode-se criar o conjunto C_i aleatoriamente no momento de inicialização de um novo indivíduo. Pode-se determinar um arco com centro em um nó, e os demais nós que estejam dentro do setor circular determinado por este arco estarão conectados ao nó. Outra opção seria determinar pontos específicos na grade de neurônios, e a partir destes pontos traçar círculos de raios variados, e nós que estejam dentro destes círculos estariam conectados ao nó de interesse. Por fim, outra opção seria considerar uma rede totalmente conectada. O fundamental é notar que todas as propostas acima implicam em um número de parâmetros e em uma codificação específica no genótipo de cada rede, ficando a critério do projetista qual abordagem utilizar.

A seguir, descreve-se o algoritmo evolutivo empregado no treinamento das redes GasNets.

2.3.2 Algoritmo Evolutivo

Para o treinamento das redes GasNet, propõe-se originalmente o uso de algoritmos genéticos. A codificação pode variar de acordo com o problema, bem como os mecanismos de seleção, recombinação e mutação gênica. Entretanto, uma estrutura básica do genótipo é sempre mantida, e engloba as estruturas de rede presentes em qualquer aplicação. Esta seção destina-se a descrever essa estrutura básica. Detalhes sobre os operadores genéticos empregados nos experimentos desta tese são apresentados no capítulo 4.

O genótipo de cada rede é formado por um conjunto de parâmetros relativo a cada neurônio da rede. Assim, cada linha da estrutura genotípica codifica as características de um neurônio da rede. Cada locus possui um valor inteiro no intervalo $[0, 99]$. Para uma rede GasNet totalmente conectada com j neurônios, teríamos a seguinte estrutura para um neurônio i da rede:

$$\begin{aligned} < genotipo > :: < x > < y > < rec > < TE > < CE > < D_i^0 > \\ & < bias > < s > < r > < w_{1i} > .. < w_{ji} > \end{aligned}$$

em que os parâmetros $x \in \mathbb{N}, 0 \leq x \leq 99$, e $y \in \mathbb{N}, 0 \leq y \leq 99$ determinam a coordenada do neurônio em questão num espaço euclidiano 2D $[100 \times 100]$, $rec \in \{0, 1, 2\}$ determina se a conexão elétrica recorrente é excitatória, inibitória, ou inexistente; $TE \in \{0, 1, 2, 3\}$ determina as circunstâncias sob as quais o neurônio irá emitir gás: se sua atividade elétrica exceder um limiar

predeterminado (adotado originalmente como 0,5), se a concentração de gás tipo 1 exceder um limiar predeterminado (adotado originalmente como 0,5), se a concentração de gás tipo 2 exceder um limiar predeterminado (adotado originalmente como 0,5), ou se o neurônio não emite gás sob circunstância nenhuma; $CE \in \{0,1\}$ especifica qual gás o neurônio emite, gás 1 ou gás 2; $D_i^0 \in \mathbb{N}$, $0 \leq D_i^0 \leq 11$, $bias \in \mathbb{R}$, $-1 \leq bias \leq 1$ e $s \in \mathbb{N}$, $0 \leq s \leq 11$ são relativos às Equações 2.7, 2.9 e 2.3, respectivamente; $r \in \mathbb{R}$, $10 \leq r \leq 60$ é o raio de influência do gás do neurônio; $w_{ji} \in \mathbb{R}$, $-1 \leq w_{ji} \leq 1$ são os pesos associados às conexões elétricas.

A decodificação de genótipo para fenótipo adotada é a mesma utilizada no modelo original (Husbands et al., 1998), e descrita na Equação 2.10:

$$fen = \begin{cases} gen/99, & \text{se } fen \text{ for contínuo} \\ gen \bmod N_{nom}, & \text{se } fen \text{ for nominal} \end{cases} \quad (2.10)$$

em que

fen é o valor do fenótipo;

gen é o valor do genótipo;

N_{nom} é o número de valores fenotípicos possíveis;

mod é o operador que fornece o valor do resto da divisão.

Conforme observado anteriormente, os parâmetros rec , TE , CE , D_i^0 e s assumem valores nominais e os demais parâmetros assumem valores contínuos. É importante lembrar que divide-se o valor de gen por 99 em casos contínuos por causa da escolha do intervalo de valores de cada locus ($[0, 99]$). Os valores contínuos também podem sofrer uma normalização posterior em intervalos distintos, dependendo das escolhas do programador. Por exemplo, o parâmetro r varia entre 10 e 60, assumindo 100 valores igualmente distribuídos nesse intervalo.

Acredita-se que todas essas características forneçam à rede propriedades adaptativas relevantes, uma vez que os valores das sinapses podem ser modificados em tempo de execução pela ação dos gases. Apresenta-se, a seguir, um exemplo ilustrativo de execução da GasNet original, visando uma melhor compreensão da ferramenta.

Exemplo de execução de uma GasNet

Para o exemplo, será implementada uma solução para o problema conhecido como gerador central de padrões (Smith et al., 2003). O objetivo da tarefa é gerar ciclicamente uma sequência numérica pré-estabelecida, nesse caso composta por onze “1” seguidos de sete “0” (e.g. “111111111110000000111111111110000000”).

A GasNet proposta para essa tarefa será formada por quatro neurônios, totalmente conectados eletricamente. Será considerado como saída da rede o sinal do neurônio 1. Se esse valor for maior que 0, a saída será interpretada como 1, se for menor ou igual a 0, a saída será considerada 0.

O genótipo de cada indivíduo da população, que representa uma proposta de configuração de GasNet a ser evoluída, será:

$\langle \text{genotipo} \rangle ::$

$\langle x_1 \rangle \langle y_1 \rangle \langle rec_1 \rangle \langle TE_1 \rangle \langle CE_1 \rangle \langle D_1^0 \rangle \langle bias_1 \rangle \langle r_1 \rangle \langle s_1 \rangle \langle w_{11} \rangle \langle w_{21} \rangle \langle w_{31} \rangle \langle w_{41} \rangle$
 $\langle x_2 \rangle \langle y_2 \rangle \langle rec_2 \rangle \langle TE_2 \rangle \langle CE_2 \rangle \langle D_2^0 \rangle \langle bias_2 \rangle \langle r_2 \rangle \langle s_2 \rangle \langle w_{12} \rangle \langle w_{22} \rangle \langle w_{32} \rangle \langle w_{42} \rangle$
 $\langle x_3 \rangle \langle y_3 \rangle \langle rec_3 \rangle \langle TE_3 \rangle \langle CE_3 \rangle \langle D_3^0 \rangle \langle bias_3 \rangle \langle r_3 \rangle \langle s_3 \rangle \langle w_{13} \rangle \langle w_{23} \rangle \langle w_{33} \rangle \langle w_{43} \rangle$
 $\langle x_4 \rangle \langle y_4 \rangle \langle rec_4 \rangle \langle TE_4 \rangle \langle CE_4 \rangle \langle D_4^0 \rangle \langle bias_4 \rangle \langle r_4 \rangle \langle s_4 \rangle \langle w_{14} \rangle \langle w_{24} \rangle \langle w_{34} \rangle \langle w_{44} \rangle$

O objetivo do exemplo é ilustrar o funcionamento da GasNet original. Portanto, serão omitidos agora detalhes relativos aos operadores genéticos. Entretanto, o modelo de GasNet bem como os detalhes de implementação do algoritmo genético utilizados nos experimentos do capítulo 4, serão minuciosamente explicados em seu decorrer.

Após a evolução, diversos indivíduos obtiveram desempenho ótimo na tarefa. O genótipo de um deles encontra-se descrito abaixo:

$\langle \text{genotipo} \rangle ::$

$\langle 91 \rangle \langle 28 \rangle \langle 46 \rangle \langle 73 \rangle \langle 28 \rangle \langle 18 \rangle \langle 53 \rangle \langle 22 \rangle \langle 74 \rangle \langle 76 \rangle \langle 76 \rangle \langle 84 \rangle \langle 24 \rangle$
 $\langle 62 \rangle \langle 16 \rangle \langle 1 \rangle \langle 7 \rangle \langle 86 \rangle \langle 38 \rangle \langle 81 \rangle \langle 63 \rangle \langle 50 \rangle \langle 87 \rangle \langle 0 \rangle \langle 53 \rangle \langle 55 \rangle$
 $\langle 90 \rangle \langle 42 \rangle \langle 3 \rangle \langle 18 \rangle \langle 46 \rangle \langle 70 \rangle \langle 97 \rangle \langle 72 \rangle \langle 80 \rangle \langle 27 \rangle \langle 36 \rangle \langle 95 \rangle \langle 27 \rangle$
 $\langle 81 \rangle \langle 36 \rangle \langle 30 \rangle \langle 44 \rangle \langle 16 \rangle \langle 35 \rangle \langle 52 \rangle \langle 65 \rangle \langle 1 \rangle \langle 47 \rangle \langle 96 \rangle \langle 87 \rangle \langle 88 \rangle$

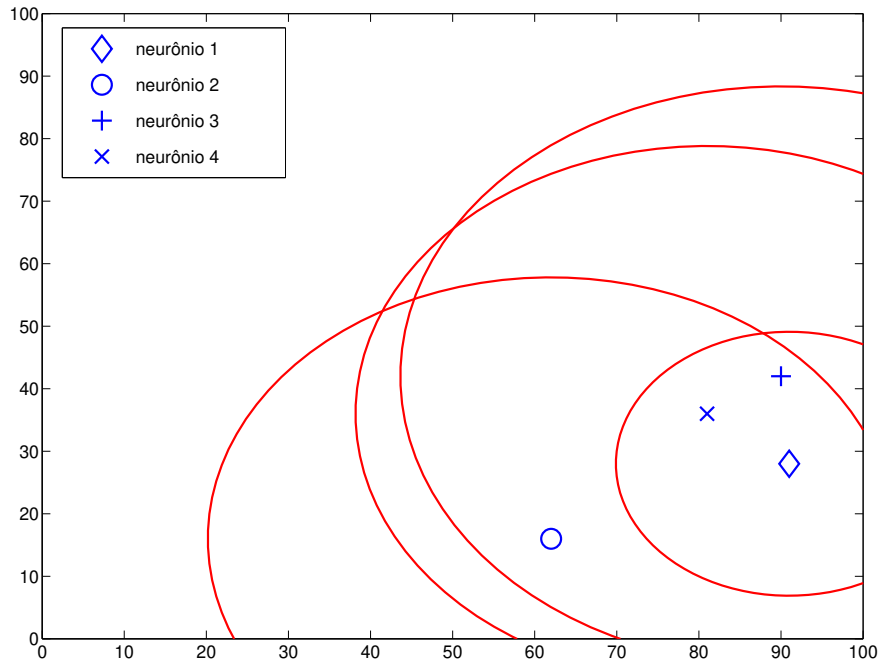


Fig. 2.11: GasNet obtida para resolução do problema de geração de padrões. A rede está limitada a um espaço euclidiano bidimensional de dimensões 100×100 , e só emite gás tipo 1, com o raio máximo de ação do gás de cada neurônio simbolizado pela cor vermelha.

Por consequência, o fenótipo relativo a esse indivíduo, de acordo com as regras listadas anteriormente (Equação 2.10), será:

$\langle fenotipo \rangle ::$

$\langle 91 \rangle \langle 28 \rangle \langle 1 \rangle \langle 1 \rangle \langle 0 \rangle \langle 8 \rangle \langle 0,07 \rangle \langle 21,11 \rangle \langle 9 \rangle \langle 0,53 \rangle \langle 0,53 \rangle \langle 0,69 \rangle \langle -0,51 \rangle$
 $\langle 62 \rangle \langle 16 \rangle \langle 1 \rangle \langle 1 \rangle \langle 0 \rangle \langle 6 \rangle \langle 0,63 \rangle \langle 41,81 \rangle \langle 7 \rangle \langle 0,53 \rangle \langle -1 \rangle \langle 0,07 \rangle \langle 0,11 \rangle$
 $\langle 90 \rangle \langle 42 \rangle \langle 0 \rangle \langle 0 \rangle \langle 0 \rangle \langle 5 \rangle \langle 0,95 \rangle \langle 46,36 \rangle \langle 4 \rangle \langle 0,69 \rangle \langle 0,07 \rangle \langle 0,91 \rangle \langle -0,45 \rangle$
 $\langle 81 \rangle \langle 36 \rangle \langle 0 \rangle \langle 2 \rangle \langle 0 \rangle \langle 3 \rangle \langle 0,05 \rangle \langle 42,82 \rangle \langle 2 \rangle \langle -0,51 \rangle \langle 0,11 \rangle \langle -0,45 \rangle \langle 0,77 \rangle$

A Figura 2.11 ilustra a rede obtida, num espaço euclidiano bidimensional de dimensões 100×100 (como a rede é totalmente conectada, as conexões elétricas foram suprimidas para facilitar a visualização). Note que a rede evoluída só emite gás tipo 1, com o raio máximo de ação do gás de cada neurônio simbolizado pela cor vermelha.

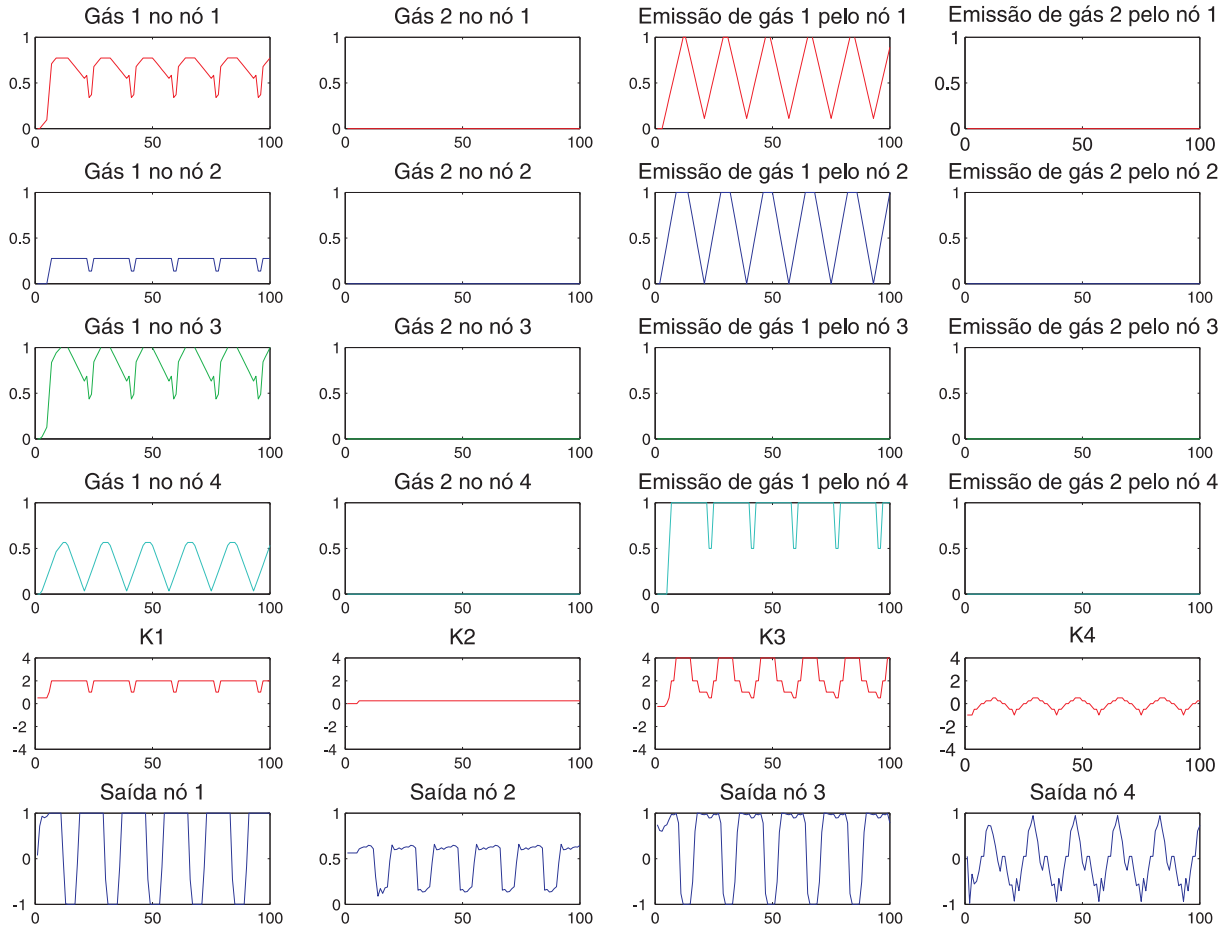


Fig. 2.12: Curvas relativas a algumas das variáveis relevantes para a compreensão da dinâmica da rede GasNet. Estão representados comportamentos obtidos em 100 passos do algoritmo.

A Figura 2.12 traz as curvas relativas a algumas das variáveis relevantes para a compreensão da dinâmica da rede. Observa-se, por exemplo, que o neurônio 3 não emite gás nenhum (o valor de TE_3 é 0, indicando que o nó não emite gás sob circunstância nenhuma), mas sofre a influência de gases emitidos por neurônios vizinhos. Nota-se também a influência da concentração de gases sobre os parâmetros K_1, K_2, K_3, K_4 , que indicam o nível de modulação gasosa que cada nó irá sofrer. Observe a saída do nó 1, destacada na Figura 2.13. Essa saída representa a solução para o problema de gerador de padrões. Recorde que um valor de saída do neurônio acima de 0 indica 1 e abaixo de 0 indica 0. Pode-se observar que a saída do neurônio 1 permanece acima de 0 por 11 iterações e abaixo de 0 por 7 iterações.

Trabalhos recentes têm investigado a relevância da relação espacial entre os neurônios no modelo original da GasNet, sendo que Vargas et al. (2007) propuseram um novo modelo de GasNet, não-

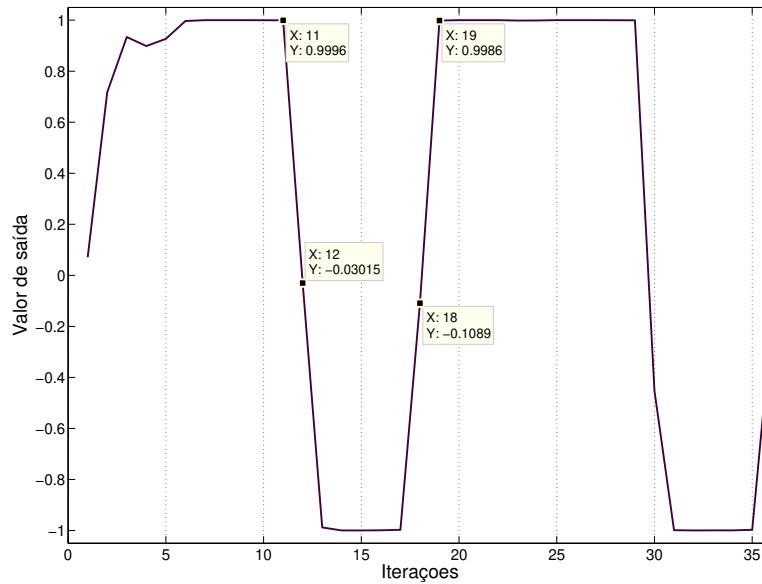


Fig. 2.13: Destaque da saída do nó 1, que representa a solução proposta pela GasNet evoluída para o problema de gerador de padrões.

limitado espacialmente, chamado NSGasNet (do inglês *NonSpatial GasNet*). Este modelo mostrou-se superior em uma série de tarefas envolvendo geração de padrões (Vargas et al., 2007), e será portanto adotado para a concepção de um sistema homeostático artificial a ser abordado no Capítulo 4. Descreve-se, a seguir, o modelo NSGasNet.

2.3.3 NSGasNets

Na abordagem não-espacial (NSGasNet, do inglês *Non-Spatial GasNet*) não existe a noção de espaço físico entre os neurônios. Portanto, um neurônio é capaz de modular, através da emissão de gases, qualquer outro neurônio da rede. A concentração de gás em cada neurônio da rede, devido a um neurônio emissor, depende do grau de estimulação entre os neurônios e da dispersão gasosa, obedecendo à Equação 2.11 (e não mais obedecendo à Equação 2.2). O grau de estimulação entre os neurônios é determinado por um termo geneticamente especificado, chamado *Mbias* (bias modulador), que varia no intervalo $[0, 1]$. O valor “0” significa que o neurônio-alvo não é afetado por esse neurônio emissor. Um valor acima de “0” significa que o neurônio-alvo será afetado por este neurônio emissor, a uma taxa proporcional ao nível de estimulação gasosa.

$$C(t) = Mbias \times T(t) \quad (2.11)$$

Suponha uma rede NSGasNet formada por 3 neurônios, denominados 1, 2 e 3. Suponha que apenas o nó 1 é emissor de gás e que os nós 2 e 3 tenham seus limiares de estimulação ($Mbias$) em relação ao nó 1 geneticamente estabelecidos como 0,1 e 0,8, respectivamente. Isto significa que, se o nó 1 emitir gás ao nível máximo 1, ele pode modular os nós 2 e 3 em um nível máximo de 0,1 e 0,8, respectivamente.

O genótipo de uma rede NSGasNet assemelha-se ao de uma rede GasNet, com exceção dos parâmetros referentes à relação espacial entre os nós e máximo raio de emissão. Para uma rede com j neurônios, totalmente conectada, teríamos o seguinte genótipo para um neurônio i :

$$\begin{aligned} < genotipo > :: < rec > < TE > < CE > < D_i^0 > < bias > < s > \\ & < Mbias_{i1} > .. < Mbias_{ij} > < w_{1i} > .. < w_{ji} > \end{aligned}$$

Igualmente à seção anterior, apresenta-se, a seguir, um exemplo ilustrativo de execução da NSGasNet, visando uma melhor compreensão da ferramenta.

Exemplo de execução de uma NSGasNet

O problema a ser tratado pelo exemplo é o mesmo abordado no exemplo com a GasNet original. Será implementada uma solução para o problema de gerador central de padrões, com o objetivo de gerar ciclicamente uma sequência de onze “1” e sete “0”.

A NSGasNet proposta para essa tarefa será formada por quatro neurônios, totalmente conectados eletricamente. Será considerado como saída da rede o sinal do neurônio 1. Se esse valor for maior que 0, a saída será interpretada como 1, se for menor ou igual a 0, a saída será considerada 0.

O genótipo de cada indivíduo da população (que representa uma proposta de configuração para a NSGasNet) a ser evoluída, será:

$\langle \text{genotipo} \rangle ::$

$\langle \text{rec}_1 \rangle \langle \text{TE}_1 \rangle \langle \text{CE}_1 \rangle \langle D_1^0 \rangle \langle \text{bias}_1 \rangle \langle s_1 \rangle \langle \text{Mbias}_{11} \rangle \langle \text{Mbias}_{12} \rangle \langle \text{Mbias}_{13} \rangle \langle \text{Mbias}_{14} \rangle$
 $\langle w_{11} \rangle \langle w_{21} \rangle \langle w_{31} \rangle \langle w_{41} \rangle$
 $\langle \text{rec}_2 \rangle \langle \text{TE}_2 \rangle \langle \text{CE}_2 \rangle \langle D_2^0 \rangle \langle \text{bias}_2 \rangle \langle s_2 \rangle \langle \text{Mbias}_{21} \rangle \langle \text{Mbias}_{22} \rangle \langle \text{Mbias}_{23} \rangle \langle \text{Mbias}_{24} \rangle$
 $\langle w_{12} \rangle \langle w_{22} \rangle \langle w_{32} \rangle \langle w_{42} \rangle$
 $\langle \text{rec}_3 \rangle \langle \text{TE}_3 \rangle \langle \text{CE}_3 \rangle \langle D_3^0 \rangle \langle \text{bias}_3 \rangle \langle s_3 \rangle \langle \text{Mbias}_{31} \rangle \langle \text{Mbias}_{32} \rangle \langle \text{Mbias}_{33} \rangle \langle \text{Mbias}_{34} \rangle$
 $\langle w_{13} \rangle \langle w_{23} \rangle \langle w_{33} \rangle \langle w_{43} \rangle$
 $\langle \text{rec}_4 \rangle \langle \text{TE}_4 \rangle \langle \text{CE}_4 \rangle \langle D_4^0 \rangle \langle \text{bias}_4 \rangle \langle s_4 \rangle \langle \text{Mbias}_{41} \rangle \langle \text{Mbias}_{42} \rangle \langle \text{Mbias}_{43} \rangle \langle \text{Mbias}_{44} \rangle$
 $\langle w_{14} \rangle \langle w_{24} \rangle \langle w_{34} \rangle \langle w_{44} \rangle$

Após a evolução, diversos indivíduos obtiveram desempenho ótimo na tarefa. O genótipo de um deles encontra-se descrito abaixo:

$\langle \text{genotipo} \rangle ::$

$\langle 87 \rangle \langle 54 \rangle \langle 74 \rangle \langle 9 \rangle \langle 55 \rangle \langle 13 \rangle \langle 52 \rangle \langle 43 \rangle \langle 29 \rangle \langle 72 \rangle$
 $\langle 41 \rangle \langle 16 \rangle \langle 60 \rangle \langle 13 \rangle$
 $\langle 45 \rangle \langle 94 \rangle \langle 7 \rangle \langle 81 \rangle \langle 24 \rangle \langle 25 \rangle \langle 68 \rangle \langle 13 \rangle \langle 88 \rangle \langle 82 \rangle$
 $\langle 24 \rangle \langle 25 \rangle \langle 58 \rangle \langle 69 \rangle$
 $\langle 62 \rangle \langle 85 \rangle \langle 39 \rangle \langle 63 \rangle \langle 40 \rangle \langle 63 \rangle \langle 10 \rangle \langle 43 \rangle \langle 30 \rangle \langle 13 \rangle$
 $\langle 80 \rangle \langle 27 \rangle \langle 71 \rangle \langle 83 \rangle$
 $\langle 86 \rangle \langle 71 \rangle \langle 6 \rangle \langle 61 \rangle \langle 21 \rangle \langle 37 \rangle \langle 11 \rangle \langle 49 \rangle \langle 26 \rangle \langle 9 \rangle$
 $\langle 26 \rangle \langle 39 \rangle \langle 20 \rangle \langle 72 \rangle$

O fenótipo relativo a esse indivíduo, de acordo com as regras listadas anteriormente (Equação 2.10), será:

$\langle \text{fenotipo} \rangle ::$

$\langle 0 \rangle \langle 0 \rangle \langle 0 \rangle \langle 10 \rangle \langle 0,11 \rangle \langle 3 \rangle \langle 0,52 \rangle \langle 0,43 \rangle \langle 0,29 \rangle \langle 0,72 \rangle$
 $\langle -0,17 \rangle \langle -0,67 \rangle \langle 0,21 \rangle \langle -0,73 \rangle$
 $\langle 0 \rangle \langle 1 \rangle \langle 1 \rangle \langle 5 \rangle \langle -0,51 \rangle \langle 4 \rangle \langle 0,68 \rangle \langle 0,13 \rangle \langle 0,88 \rangle \langle 0,82 \rangle$
 $\langle -0,51 \rangle \langle -0,49 \rangle \langle 0,17 \rangle \langle 0,39 \rangle$
 $\langle 2 \rangle \langle 1 \rangle \langle 1 \rangle \langle 9 \rangle \langle -0,19 \rangle \langle 8 \rangle \langle 0,10 \rangle \langle 0,43 \rangle \langle 0,30 \rangle \langle 0,13 \rangle$
 $\langle 0,61 \rangle \langle -0,45 \rangle \langle 0,43 \rangle \langle 0,67 \rangle$
 $\langle 2 \rangle \langle 2 \rangle \langle 0 \rangle \langle 7 \rangle \langle -0,57 \rangle \langle 5 \rangle \langle 0,11 \rangle \langle 0,49 \rangle \langle 0,26 \rangle \langle 0,09 \rangle$
 $\langle -0,47 \rangle \langle -0,21 \rangle \langle -0,59 \rangle \langle 0,45 \rangle$

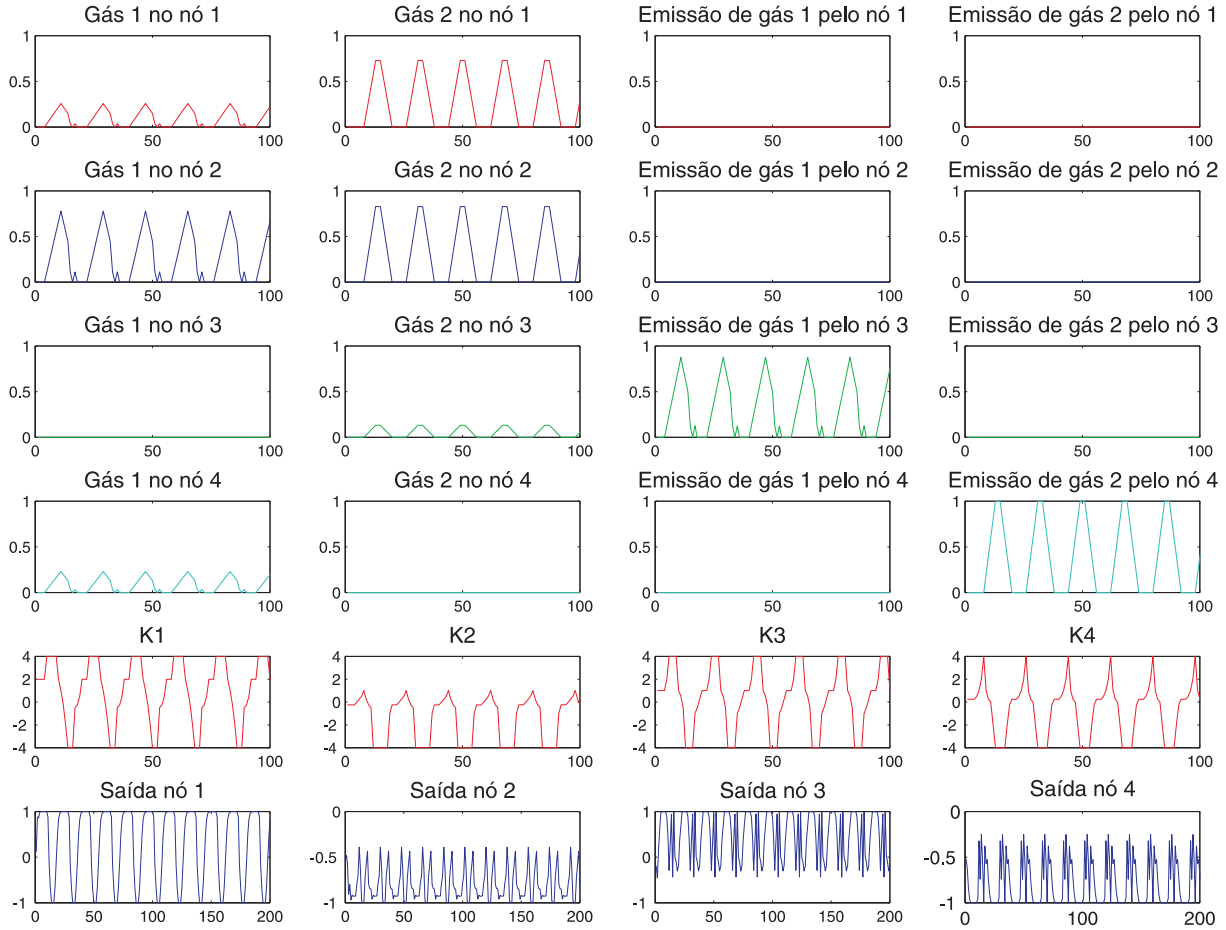


Fig. 2.14: Curvas relativas a algumas das variáveis importantes para a compreensão da dinâmica da rede NS-GasNet. São apresentados comportamentos obtidos em 100 passos do algoritmo.

Como a rede não possui posicionamento espacial, e também porque todos os neurônios estão eletricamente totalmente conectados, a visualização da rede será suprimida. Observe a Figura 2.14, que contém as curvas relativas a algumas das variáveis importantes para a compreensão da dinâmica da rede. Note que o nó 4 é o único a emitir gás tipo 2. A influência de sua emissão nos outros nós será determinada pelo valor do M_{bias} correspondente. Nesse caso, o M_{bias} relativo aos nós 1, 2 e 3 vale 0,72, 0,82 e 0,13, respectivamente. Portanto, a influência do gás 2 deve ser maior no nó 2, seguido pelos nós 1 e 3. Observando-se a Figura 2.14, pode-se confirmar esse fato.

É possível fazer uma análise semelhante para o caso do gás tipo 1, emitido apenas pelo nó 3. Note que o M_{bias} correspondente aos nós 1, 2 e 4 vale 0,29, 0,88 e 0,26, respectivamente. Portanto, o nó 2 deve ser mais influenciado que os demais. Observando-se a Figura 2.14, pode-se confirmar esse fato. Note também a influência das concentrações gasosas sobre os parâmetros relativos à modulação

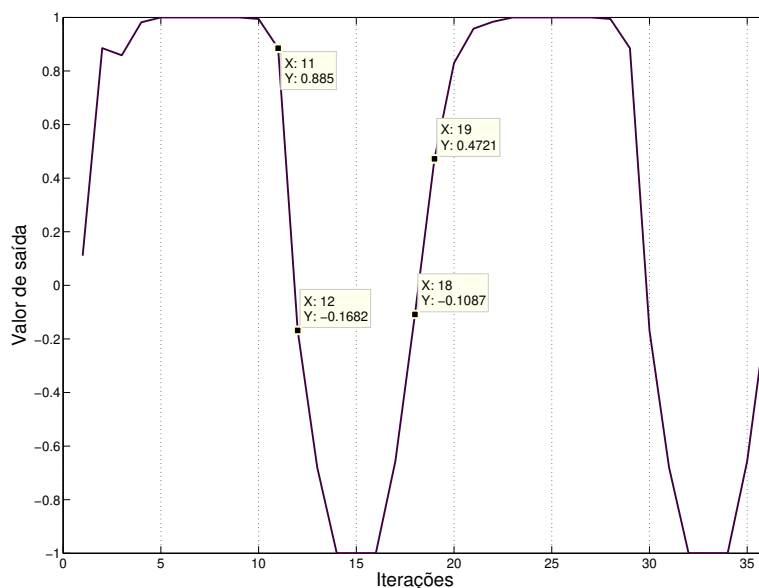


Fig. 2.15: Destaque da saída do nó 1, que representa a solução proposta pela NSGasNet evoluída para o problema de gerador de padrões.

gasosa, K_1, K_2, K_3, K_4 , que apresentam comportamentos diversos durante as iterações.

A Figura 2.13 destaca a saída do nó 1, que representa a solução para o problema de gerador de padrões. Pode-se observar que a saída do neurônio 1 permanece acima de 0 por 11 iterações e abaixo de 0 por 7 iterações, indicando a solução para o gerador de padrões.

Esta seção forneceu os conceitos básicos de redes neurais artificiais, mais especificamente o modelo NSGasNet, a ser explorado no Capítulo 4. O próximo capítulo apresenta um estudo utilizando Sistemas Classificadores com Aprendizado, em tarefas reativas e não reativas desempenhadas por um agente robótico.

Capítulo 3

Investigando Sistemas Classificadores com Aprendizado em tarefas robóticas reativas e não-reativas

"Certainly the notion of a classifier system is appealing, how could roving bands of mating, reproducing, fighting, and dying rules fail to captivate our collective imagination?"

Stewart Wilson e David Goldberg

Resumo - Existem poucas contribuições utilizando Sistemas Classificadores com Aprendizado (SCAs) em navegação autônoma de robôs. O objetivo do capítulo é analisar o comportamento do SCA original baseado em energia (LCS, na literatura em inglês, e aqui denominado SCAO), e do SCA baseado em precisão, XCS, quando aplicados a duas tarefas robóticas distintas. A primeira tarefa é puramente reativa, o que significa que a ação a ser desempenhada pode ser escolhida observando-se apenas o estado atual dos sensores. A segunda tarefa é não-reativa, o que significa que o robô deve utilizar alguma forma de memória para ser capaz de lidar com estados de ambigüidade. Apresenta-se uma análise da evolução de regras, fornecendo exemplos de populações evoluídas e suas peculiaridades para ambos os sistemas. Uma revisão das versões de SCAs em robótica é apresentada em conjunto com uma discussão sobre os principais resultados obtidos.

3.1 Introdução

Um sistema de controle que permite a um robô desempenhar tarefas de navegação da forma que um ser humano, ou mesmo um inseto, desempenha não é trivial. A falta de informação de alta qualidade disponível para o processo de tomada de decisão, os múltiplos objetivos conflitantes e a complexidade espaço-temporal do ambiente apresentam-se como um cenário desafiador para o sistema de controle candidato.

Para abordar esses problemas, técnicas evolutivas e de aprendizado por reforço são cada vez mais aplicadas a problemas de navegação autônoma de robôs. Um exemplo de aplicação de tais técnicas é a Robótica Evolutiva (RE). RE é um campo de pesquisa recente, que pretende aplicar técnicas de computação evolutiva para evoluir a morfologia do robô (*hardware*) e/ou controladores (*software*) para robôs autônomos simulados e reais. Embora seja uma área de pesquisa bem estabelecida, com muitas contribuições relatadas na literatura (Nolfi & Floreano, 2000), ela possui dificuldades intrínsecas, principalmente associadas com o tempo necessário para evoluir um indivíduo. São também sugeridas na literatura técnicas de aprendizado por reforço, que aparentam ser mais adequadas para o

aprendizado em tempo real (Studley & Bull, 2005).

Com o objetivo de combinar evolução e aprendizado de forma integrada, os Sistemas Classificadores com Aprendizado (SCAs), originalmente propostos por Holland (1975), aparecem como uma alternativa. Complementarmente, esses sistemas podem ser considerados um modelo apropriado para a síntese de sistemas complexos adaptativos (Holland, 1995) e processos de inferência em sistemas cognitivos (Holland et al., 1986).

De acordo com Holmes et al. (2000), existem ainda muitos desafios inexplorados e limitações a serem superadas no campo dos SCAs. Dessa forma, propostas alternativas desses sistemas são investigadas com base em suas características adaptativas e também nas peculiaridades do problema. Atualmente, as variantes de SCAs mais estudadas são: (i) Sistema Classificador baseado em precisão, XCS (Wilson, 1995), e suas extensões, XCSF (Wilson, 2002), XCSMH (Lanzi & Wilson, 2000); (ii) Sistema Classificador Antecipatório, ACS (Stolzmann, 1996); (iii) Sistema Classificador Temporal, TCS (Hurst et al., 2002). É oportuno salientar que os SCAs são aplicados em diversas áreas (Bull, 2004), desde problemas de otimização (Vargas et al., 2004) até a análise de séries temporais (Armano, 2004). Entretanto, existem poucas contribuições à navegação autônoma de robôs.

Este capítulo dedica-se a uma análise de uma versão do Sistema Classificador com Aprendizado original, aqui denominado SCAO, e do Sistema Classificador baseado em precisão, XCS. As razões para a escolha de ambos os sistemas são: (i) o XCS é a variante dos SCAs mais investigada atualmente; (ii) o XCS foi proposto numa tentativa de desenvolver um novo procedimento para atribuição de crédito que poderia guiar o sistema classificador para uma melhor performance em situações em que o SCAO não apresenta desempenho satisfatório; (iii) todos os sistemas classificadores propostos até o momento foram inspirados no SCAO e, portanto, acredita-se que uma comparação entre o sistema mais estudado (XCS) e uma versão do SCAO possa trazer benefícios e percepções interessantes para o campo da robótica e também para a comunidade dos SCAs.

Além disso, o capítulo procura contribuir na elucidação dos mecanismos de evolução de regras e suas peculiaridades no domínio de navegação autônoma em robótica, para os dois sistemas, compilando os trabalhos relacionados e, conseqüentemente, fornecendo informações úteis para pesquisas futuras na área. Complementarmente, o capítulo apresenta exemplos de populações evoluídas, ilustrando a dinâmica das regras e o comportamento de navegação resultante.

Acompanhando as propostas, dois experimentos são realizados. O Experimento 1 apresenta uma tarefa puramente reativa, enquanto que o Experimento 2 tem estados ambíguos, o que requer que o robô use algum tipo de memória para resolver a tarefa.

O capítulo está organizado da seguinte forma: a seção 3.2 introduz os fundamentos do SCAO e do XCS. A seção 3.3 apresenta uma revisão dos trabalhos relacionados na literatura. A seção 3.4 descreve os experimentos e apresenta os resultados, juntamente com uma análise da evolução para cada caso. Por fim, apresentam-se as discussões na seção 3.5.

3.2 Fundamentos de Sistemas Classificadores com Aprendizado

Um sistema classificador é composto por diversos classificadores, que representam os indivíduos da população. Cada classificador corresponde a uma regra no formato “se-então”, formada por duas partes: antecedente e conseqüente. O antecedente indica a qual situação ambiental o classificador deve ser associado, ou seja, para cada informação captada do ambiente, deve existir um classificador que se identifique com esse padrão de leitura e que proponha uma ação, expressa pelo conseqüente.

Esta mensagem do ambiente é codificada e comparada com a parte antecedente de cada classificador. De acordo com sua “energia” e similaridade com a situação apresentada, o classificador pode ser ativado e codificar ações específicas a serem tomadas. As ações expressas por todos os classificadores ativados competem para decidir qual irá atuar sobre o ambiente. Dependendo do resultado desta ação sobre o ambiente, o classificador que originou esta ação pode ser penalizado ou recompensado (ganhando ou perdendo energia). Periodicamente, o elenco de classificadores é submetido a um processo evolutivo, com base em sua energia, realizado por meio de algoritmos genéticos (Goldberg, 1989). A Figura 3.1 ilustra a interação do sistema classificador com o ambiente.

As mensagens do ambiente são codificadas como *strings* formadas por um alfabeto binário $\{0,1\}$. Cada classificador é representado por uma *string* formada a partir do alfabeto $\{1,0,\#\}$. O símbolo “#” indica uma situação de “*don’t care*”, ou seja, o valor que o campo assume identifica-se com qualquer outro valor possível no campo equivalente da mensagem. Outra característica importante é a “especificidade” de um classificador. Ela é relacionada à proporção de “*don’t cares*” na mensagem: quanto mais símbolos “#” um classificador tem, a um maior número de mensagens do ambiente ele

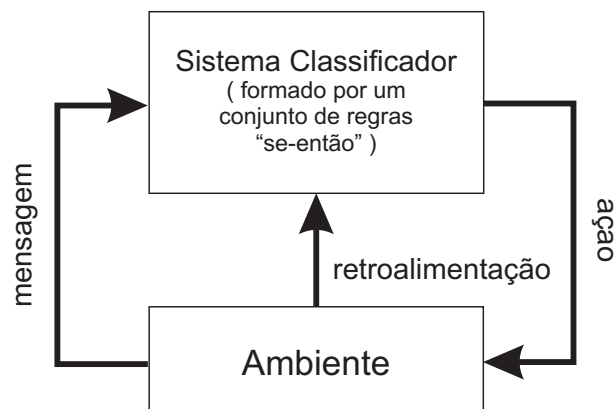


Fig. 3.1: Interação do Sistema Classificador com o ambiente.

pode ser associado, o que implica uma menor especificidade. Esta medida pode contribuir para a redução do tamanho da população de classificadores no final do processo evolutivo, particularmente em situações favoráveis à ocorrência de regras com baixa especificidade. Na seção referente aos experimentos, a codificação será retomada e explicada em maiores detalhes.

Essa estrutura é básica para todos os sistemas classificadores aqui estudados. Nas seções seguintes, os dois tipos de classificadores a serem adotados no trabalho serão detalhados, evidenciando-se suas peculiaridades frente ao modelo genérico descrito acima.

3.2.1 SCAO

O sistema classificador original proposto por Holland (1975), denominado Sistema Classificador com Aprendizado, é uma metodologia para a criação e atualização evolutiva de regras, denominadas classificadores. Em um sistema de tomada de decisões, estes classificadores codificam alternativas de ações específicas para as características de um ambiente em determinado instante (Booker et al., 1989).

Para a implementação do SCAO, tomou-se como base o trabalho de Holland (1975), Vargas (2000) e Cazangi (2004). Um algoritmo simplificado de geração de um sistema classificador é apresentado a seguir (Algoritmo 3.1).

O sistema classificador capta o estado atual do ambiente (Passo 2). A mensagem recebida deve ser codificada para o padrão utilizado na parte antecedente do classificador (Passo 3), de forma a permitir a seleção dos classificadores que mais se identificam com a mensagem. Normalmente se utiliza a

Algoritmo 3.1 Algoritmo simplificado SCA (adaptado de Vargas (2000))

- 1: **INÍCIO DE UMA ÉPOCA DE ITERAÇÕES**
 - 2: Receber mensagem do ambiente
 - 3: Codificar mensagem
 - 4: Selecionar classificadores que se identificam com a mensagem
 - 5: Iniciar “Competição”:
 - 6: Calcular lance de cada competidor
 - 7: Apontar vencedor
 - 8: Cobrar taxas dos participantes e do vencedor
 - 9: Atuar sobre o ambiente
 - 10: Receber mensagem (retroalimentação) do ambiente
 - 11: Codificar mensagem
 - 12: Recompensar ou punir o classificador vencedor
 - 13: Cobrar taxa de vida de todos os indivíduos
 - 14: Caso não seja fim de uma “Época”, retornar ao **Passo 2**
 - 15: **FIM DE UMA ÉPOCA DE ITERAÇÕES**
 - 16: Selecionar indivíduos mais aptos para aplicação do algoritmo genético
 - 17: Aplicar os operadores de crossover e mutação para gerar os filhos
 - 18: Selecionar os indivíduos mais fracos da população
 - 19: Inserir os filhos gerados no **Passo 17** na população, substituindo os indivíduos selecionados no **Passo 18**
 - 20: Caso não seja fim do processo evolutivo, voltar ao **Passo 1**
-

distância euclidiana ou a distância de Hamming (Hamming, 1950) neste processo. Uma vez definido o conjunto de classificadores que se identificam com a mensagem (Passo 4), deve-se decidir qual deles atuará no ambiente, por meio de uma competição. Esta competição é feita na forma de lances (Equação 3.1), onde cada classificador apresenta um valor determinado por sua energia S_t (ponderada pelo parâmetro k_0) e sua especificidade $Spec$ (ponderada pelos parâmetros k_1 , k_2 e $SPow$). Dessa forma, se lances altos devem ser relativos principalmente à regras com alta energia, basta aumentar o valor de k_0 . Se deseja-se que regras generalistas (com pouca especificidade) tenham mais participação na composição do lance, pode-se diminuir o valor do parâmetro k_1 , relativo à parte não específica do classificador, e aumentar o valor dos parâmetros k_2 e $SPow$. Após essas definições, o classificador que obtiver o melhor lance será o escolhido para atuar sobre o ambiente (Passo 9). É importante observar a função do ruído gaussiano N_t . Ele é o responsável pela exploração de classificadores que propõem ações diferentes para uma mesma leitura do ambiente, permitindo que classificadores com baixa energia sejam esporadicamente selecionados.

Após atuar sobre o ambiente, o sistema atualiza a leitura da mensagem proveniente do ambiente. Uma vez que esta mensagem é codificada (Passo 11), ela será utilizada para estabelecer a recompensa ou punição deste classificador (Passo 12), aumentando ou diminuindo a energia do mesmo. Por fim, todos os classificadores são taxados por estarem presentes na população (Passo 13). Esta taxa  o   essencial para que classificadores pouco utilizados possam dar lugar a outros classificadores. A equa  o 3.2 descreve o processo de atualiza  o da energia do classificador a cada itera  o: o primeiro termo desta equa  o refere-se   meia-vida do classificador, fazendo com que classificadores pouco utilizados tendam a ter sua energia reduzida a longo prazo; R_t incorpora o valor da recompensa diretamente na energia do classificador; $T_{vencedor}$ cobra uma taxa do classificador ganhador, por ele ter atuado, e T_{Lance} cobra uma taxa de cada classificador que tenha se identificado com a mensagem do ambiente e competido para atuar. Assim, um classificador que atue sempre e receba recompensas positivas por sua a  o tende a ter a sua energia aumentada.

Os passos 16 a 19 comp  em a parte evolutiva do processo, onde classificadores mais ativos e adaptados ter  o mais chance de se combinar para gerar novos classificadores, contribuindo para a cobertura do espa  o de busca. Esses novos classificadores s  o inseridos no lugar de classificadores pouco utilizados ou de desempenho insatisfat  rio. Os par  metros do algoritmo s  o detalhados na Tabela 3.1.

$$Lance = k_0 * (k_1 + k_2 * Spec^{SPow}) * S_t + \sigma * N_t, \quad (3.1)$$

$$Energia(t + 1) = (1 - \frac{1}{2}) * Energia(t) + R_t - T_{vencedor} - T_{Lance} * Lance, \quad (3.2)$$

3.2.2 XCS

Wilson (1995) prop  s adapta  es junto aos sistemas classificadores originais, criando os sistemas classificadores baseados em precis  o (XCS). Esta proposta difere do SCAO principalmente em dois aspectos:

- a energia de cada classificador   formada por tr  s par  metros, sendo eles predi  o, erro de predi  o e fitness. O fitness de cada classificador   um n  mero calculado por uma fun  o

Tab. 3.1: Descrição dos parâmetros - SCAO

Parâmetros SCAO	
k_0	: constante relativa à energia do classificador ($0 < k_0 < 1$)
k_1	: constante relativa à parte não específica do classificador ($0 < k_1 < 1$)
k_2	: constante relativa à parte específica do classificador ($0 < k_2 < 1$)
Spec	: especificidade de um classificador
SPow	: influência da especificidade de um classificador
S_t	: Energia do classificador no instante t
σ	: desvio-padrão do ruído
N_t	: ruído com distribuição normal
n	: meia vida do classificador
Energia(t)	: energia do classificador no instante t
R_t	: recompensa no instante t
$T_{Vencedor}$	: resultado da equação 3.1 se classificador ganhador; 0 caso contrário
T_{Lance}	: taxa por participar na competição
Lance	: resultado da equação 3.1

inversa do erro médio da predição, ou seja, o fitness baseia-se na medida de quão precisa é uma predição.

- o algoritmo genético é aplicado em nichos definidos pelos conjuntos de classificadores ativados pela situação do ambiente, e não mais sobre toda a população.

A cada execução do Algoritmo 3.2, ilustrado pela Figura 3.2, um conjunto M de classificadores que se identificam com a atual leitura do ambiente é criado (Passo 5). A população inicial no XCS é nula, e o sistema utiliza uma técnica denominada *covering*. Esta técnica cria classificadores sob demanda, ao invés de gerá-los aleatoriamente no início. Essa abordagem evita que classificadores com padrões muito diferentes dos encontrados no ambiente existam, e torna a evolução mais simples. Ao receber uma nova mensagem do ambiente (Passo 3), caso nenhum classificador se identifique, um novo é criado, cujo antecedente é o mesmo da leitura ambiental, porém “*don’t cares*” podem substituir determinado campo, com probabilidade $P_{\#}$.

Após essa criação, um vetor com o valor da predição do pagamento esperado para cada ação existente no conjunto M é formado com base no fitness e no erro de predição de cada classificador

Algoritmo 3.2 Algoritmo simplificado XCS (adaptado de Butz & Wilson (2000))

```

1: Enquanto Critério de parada não alcançado faça
2:   INÍCIO DO EXPERIMENTO
3:   Receber mensagem  $\sigma$  do ambiente
4:   Codificar mensagem
5:   Selecionar classificadores que se identificam com a mensagem e formar o conjunto  $[M]$ 
6:   Gerar vetor de valor da predição de pagamento para cada ação em  $[M]$ 
7:   Selecionar ação com base no vetor de predição PA
8:   Formar o conjunto  $[A]$  com base na ação e no conjunto  $[M]$ 
9:   Atuar sobre o ambiente
10:  Receber recompensa  $\rho$ 
11:  Se Conjunto  $[A]_{-1}$  prévio não estiver vazio Então
12:    Atualizar conjunto  $[A]_{-1}$ 
13:    Aplicar GA em  $[A]_{-1}$ 
14:  Fim se
15:  Se Fim de uma época Então
16:    Atualizar conjunto  $[A]$ 
17:    Aplicar GA em  $[A]$ 
18:    Esvaziar conjunto  $[A]_{-1}$ 
19:  Senão
20:     $[A]_{-1} = [A]$ 
21:     $\rho_{-1} = \rho$ 
22:     $\sigma_{-1} = \sigma$ 
23:  Fim se
24: Fim enquanto

```

(Passo 6). O conjunto de ações $[A]$ é então determinado com base nos valores deste vetor (Passos 7 e 8), pois todos os classificadores em $[M]$ que propuserem a ação escolhida integrarão o conjunto $[A]$. O modo como é feita a seleção da ação a ser desempenhada pode variar, e diferentes propostas são estudadas na literatura (Butz & Wilson, 2002). A forma adotada neste trabalho é conhecida como 50/50 exploração/exploração. Basicamente, as primeiras 50 iterações têm a ação escolhida aleatoriamente, dentre as presentes em M , com 50% de chance, e outros 50% de chance de ser escolhida com base no valor mais alto do vetor de predição. As 50 iterações seguintes têm a ação determinada unicamente pela escolha com base no valor mais alto do vetor de predição. Essa forma de escolha garante que exista exploração do espaço de ações (papel desempenhado pelo ruído gaussiano no SCAO). Em seguida, atua-se sobre o ambiente.

A atribuição de crédito é realizada no conjunto de ações anterior $[A]_{-1}$, caso ele não esteja vazio

Tab. 3.2: Descrição dos parâmetros - XCS

Parâmetros XCS	
N	: tamanho máximo da população
β	: taxa de aprendizado ($0 < \beta < 1$)
α, ϵ_0, ν	: constantes de controle da precisão ($0 < \alpha < 1, \epsilon_0 > 0, \nu > 0$)
θ_{mna}	: número mínimo de ações em $[M]$
γ	: fator de desconto em problemas de múltiplos passos
θ_{GA}	: limiar de aplicação do GA. Quando o tempo médio desde o último GA no conjunto de classificadores superar θ_{GA} , ele é novamente aplicado
χ	: probabilidade de <i>crossover</i>
μ	: probabilidade de mutação
θ_{del}	: limiar de experiência para remoção de um classificador
δ	: fração do <i>fitness</i> médio da população
θ_{sub}	: limiar de substituição. Um classificador precisa ter experiência maior que θ_{sub} para substituir outro da população
$P_{\#}$	: probabilidade de <i>don't care</i>
p_I, ϵ_I, F_I	: parâmetros de inicialização de um novo classificador

$$\epsilon \leftarrow \epsilon + \beta(|P - p| - \epsilon), \quad (3.4)$$

$$P \leftarrow \rho_{-1} + \gamma * \max(PA), \quad (3.5)$$

$$\kappa = \begin{cases} 1 & \text{se } \epsilon < \epsilon_0 \\ \alpha(\epsilon_0/\epsilon)^\nu & \text{caso contrário} \end{cases} \quad (3.6)$$

$$\kappa' = \frac{\kappa}{\sum_{cl \in [A]} cl.\kappa}, \quad (3.7)$$

$$F = F + \beta(\kappa' - F), \quad (3.8)$$

O algoritmo genético (GA, na literatura em inglês) é aplicado apenas no conjunto de ação (Pas-

13 e 17). Ele é iniciado quando os classificadores existentes no conjunto de ação ultrapassarem um valor θ_{GA} , que simboliza o tempo médio desde a última participação de cada classificador no GA. Apenas se aplica o GA em tentativas em que a forma de definição da ação é por exploração. A cada aplicação do GA, os novos classificadores gerados são comparados com toda a população. Caso o novo classificador já esteja presente na população, sua numerosidade é aumentada. Numerosidade refere-se ao número de microclassificadores (a numerosidade de um microclassificador é igual a um) que este macroclassificador representa. Esta abordagem permite a compactação da população e redução do tempo de convergência (Wilson, 1995).

Pode-se perceber que a estrutura básica do SCAO é mantida. No entanto, o sistema de escolha de classificadores e atribuição de crédito foi alterado. Desta forma, os classificadores não são punidos ou recompensados por sua ação imediata, mas sim pelo seu desempenho a longo prazo no cumprimento da tarefa. O *fitness* não é mais determinado pela energia, mas pela precisão com que uma predição é feita. Isso permite ao XCS obter regras generalistas e uma população compacta em alguns casos em que o SCAO não se mostra competente (Wilson, 1995), principalmente em problemas em que existem múltiplas possibilidades de solução e a qualidade da solução mais parcimoniosa não se destaca tanto das demais.

A próxima seção faz uma revisão dos principais trabalhos publicados envolvendo aplicações de SCAO e XCS em navegação autônoma de robôs.

3.3 Trabalhos Relacionados

Esta seção dedica-se a descrever os trabalhos mais relevantes utilizando SCAs e suas derivações em robótica autônoma móvel.

Cliff & Ross (1995) adicionaram mecanismos de memória ao sistema ZCS (Wilson, 1994) e empregaram-no em uma série de tarefas do tipo *woods*. Esta tarefa consiste de um ambiente bidimensional formado por células, onde cada célula pode conter um obstáculo T (árvore), um objetivo F (comida), objetivos secundários (mas que devem ser considerados) como nível de energia (E) e manutenção (M), e ainda espaços em branco (vazio). O número e papel dos objetivos secundários ficam a cargo do projetista, dependendo da complexidade e do comportamento a ser analisado. O

robô é capaz de detectar as oito células mais próximas a ele, e a partir desta leitura executar uma ação. Normalmente, a tarefa do robô neste cenário é aprender o caminho mais curto até o objetivo, evitando colisões com os obstáculos, e eventualmente atender a objetivos secundários, como recarregar sua energia. Uma série de tentativas é feita, posicionando o robô aleatoriamente em uma célula vazia. Caso o robô atinja o objetivo, ou um número de iterações seja alcançado (evitando assim *loops* infinitos), uma nova tentativa é iniciada. Desta forma, um gráfico (número de passos até atingir o objetivo) versus (número de tentativas) pode ser traçado, e uma vez conhecido o valor ótimo de performance (por inspeção do ambiente), uma análise de desempenho se torna rápida e precisa. Bons resultados foram obtidos pelos autores, embora performances ótimas fossem raramente conquistadas. Eles concluíram que o sistema tem dificuldades de estabilidade quando lida com longas cadeias de ação, devido principalmente a problemas na forma de criação de classificadores e a classificadores muito generalistas (que possuem muitos “*don’t cares*”) e conflitantes. Menciona-se que o XCS e seu *fitness* baseado em precisão poderiam ser utilizados para amenizar esses problemas. Os autores também fizeram uma tentativa de aplicação em robôs reais, porém, novamente, longas cadeias de ação foram fatores determinantes para a redução da performance.

Stolzmann (1996) propôs um modelo alternativo, chamado Sistema Classificador Antecipatório (ACS, na literatura em inglês), em que cada classificador tem uma previsão da consequência de sua ação no ambiente. Este sistema tem como base o mecanismo cognitivo de aprendizado “Controle Antecipatório de Comportamento”, introduzido na psicologia cognitiva por Hoffmann (1992). Esta nova arquitetura permite ao sistema criar um modelo interno do ambiente em tempo de execução. A técnica foi aplicada em tarefas de aprendizado robótico e mostrou-se que o robô seria capaz de apresentar aprendizado latente. Em um trabalho futuro, Stolzmann & Butz (1999) combinaram planejamento de ações e aprendizado latente em duas tarefas robóticas, alcançando performance quase ótima em poucas iterações do algoritmo.

Dorigo & Colombetti (1997) aplicaram o SCAO, explorando a abordagem baseada em comportamento, para controlar um robô em ambientes simulados e reais. O uso de sistemas classificadores foi motivado pelo fato dos SCAs apresentarem três notáveis características: paralelismo, possibilitando rápida velocidade de computação aliada à flexibilidade de projeto do sistema; arquitetura distribuída, que permite a divisão de uma tarefa em sub-tarefas, portanto mais simples de serem resolvidas; e

evolução de comportamento, responsável por adaptação contínua em ambientes dinâmicos.

Katagami & Yamada (2000) descrevem um SCA adaptativo, baseado no XCS, que pode estimular a velocidade de aprendizado em um robô móvel, e aplicaram-no em uma tarefa de acompanhamento de paredes e em um jogo de futebol. O sistema apresenta interação homem-robô, em que o agente humano é capaz de interagir com as regras e diretamente ensinar o robô, melhorando a configuração da população inicial e adicionando regras a qualquer momento.

Bonarini (2000) adaptou o SCAO para incorporar modelos de regras *fuzzy* (Pedrycz & Gomide, 1998), apresentando uma investigação introdutória sobre o acoplamento da abordagem *fuzzy* com o paradigma dos SCAs. Em Bonarini (1996), o autor descreve o *ELF*, um controlador *fuzzy* capaz de evoluir uma população de regras *fuzzy*. O sistema é relativamente complexo, com muitas similaridades a alguns algoritmos de SCAs, tais como: a existência de um conceito de energia (relacionado ao desempenho da regra), influenciado por um programa de reforço; a partição de regras em subpopulações, de acordo com sua adequação com a situação atual do ambiente; e competição para desempenhar uma ação. O autor aplica o sistema em um conjunto de experimentos utilizando um agente simulado e um robô real, produzindo bons resultados. Em um trabalho recente, Bonarini & Matteucci (2007) apresentam o FXCS (implementação *fuzzy* do XCS), uma extensão ao SCA *fuzzy* anterior. O autor reporta um maior tempo de aprendizado, compensado pelo aumento de robustez.

Gerard & Sigaud (2000) apresentaram o YACS (“*Yet Another Classifier System*”, na literatura em inglês), um sistema baseado no ACS, com a diferença que no YACS a ênfase é em classificadores que antecipam bem ao invés de classificadores que propõe uma ação ótima. O sistema também combina algoritmos de programação dinâmica e aprendizado latente. Os autores reportam bons resultados em tarefas em ambientes do tipo “*maze*”, mas enfatizam a necessidade de mais investigações relativas à generalização das regras. Ambientes “*maze*” caracterizam-se por possuir múltiplos corredores, podendo ser de mesma dimensão e tamanho, similares a um labirinto. O robô deve então ser capaz de navegar por esses corredores, sem colisão, e de atender a objetivos definidos pelo programador (captura de alvos, determinação de trajetórias, etc.).

Hurst et al. (2002) aplicaram o sistema ZCS (Wilson, 1994) e introduziram o sistema TCS para controlar um robô real em uma tarefa simples de desvio de obstáculos e também em uma tarefa de fototaxia (uma tarefa na qual o robô deve encontrar e ir em direção à fonte luminosa, que representa o

alvo). Studley & Bull (2005) acoplaram as abordagens TCS e XCS para criar o X-TCS, que apresentou bons resultados em uma tarefa de fototaxia. Hurst & Bull (2006) exploram o uso de mecanismos construtivos em um sistema classificador neural (NCS, *Neural Classifier System*, na literatura em inglês). No NCS, cada regra é representada por uma rede neural artificial e os parâmetros são auto-adaptáveis. Cenários do tipo *maze* foram adotados para analisar a performance e experimentos com um robô real são também executados. A abordagem TCS foi combinada com o NCS em uma tentativa de superar as dificuldades em lidar com robôs reais, como o fator tempo e ambigüidade. Os autores reportaram desempenho razoável e mostraram a viabilidade de se empregar sistemas classificadores com aprendizado em ambientes reais.

Webb et al. (2003) implementaram um sistema XCS para controlar um robô Khepera simulado em um ambiente não-markoviano (i.e. um ambiente em que o estado atual não é suficiente para determinar a melhor ação a ser expressa, de modo que estados anteriores devem ser considerados) usando um mecanismo de memória interna descrito em Lanzi & Wilson (2000).

Cazangi et al. (2003) usaram um SCA para controlar um robô simulado e um robô real em tarefas em que o robô deveria aprender a capturar alguns alvos enquanto evitava a colisão com obstáculos. Os resultados apresentados foram muito promissores, evidenciando a robustez e adaptabilidade do sistema.

Vargas et al. (2003) apresentaram um sistema híbrido, chamado Clarinet (*“Classifier Immune Network”*, na literatura em inglês), que consiste de uma rede imuno-genética de classificadores. A nova proposta híbrida teve boa performance e robustez em um problema simulado de navegação autônoma de robôs.

Com base em características de ambientes do tipo *maze*, dependentes e independentes do agente, Zatuchna (2005) realizou uma análise extensa desses ambientes introduzindo novas métricas para medir complexidade. Inspirado por princípios psicológicos, o autor também propõe uma nova abordagem de aprendizado, chamada Aprendizado por Percepção Associativa (*AgentP*, na literatura em inglês). Este novo modelo emprega mecanismos de memória e possui uma estrutura de regra “estado-ação-estado” similar ao ACS. O modelo também possui distribuição de recompensa com base na distância, que reflete o número mínimo de passos até o estado final a partir do estado atual do agente. Adicionalmente, o modelo não pretende aprender generalizações de estados. Mostra-se que essa nova

Tab. 3.3: Situações Possíveis

Antecedente de Distância	Antecedente de Luz	Interpretação
000	000	espaço aberto, sem luz
001	100	obstáculo à direita, luz à esquerda
011	010	obstáculo à frente e à direita, luz à frente
010	000	obstáculo à frente, sem luz
100	001	obstáculo à esquerda, luz à direita
101	000	obstáculo à direita e à esquerda, sem luz

abordagem é capaz de resolver com sucesso a maioria dos problemas *maze* propostos, incluindo situações com estados de ambigüidade. Outras referências sobre o uso do ACS em robótica podem ser encontradas em Kovacs (2002).

Considerando todos os trabalhos envolvendo SCAs publicados na literatura, desde a primeira aparição dos SCAs (Holland, 1975), pode-se perceber que existem relativamente poucas contribuições para o campo da robótica. Assim, o presente trabalho dedica-se a aprofundar o conhecimento de dois modelos muito estudados (SCAO e XCS), procurando manter-se fiel às suas propostas originais. Com isso, acredita-se que o texto possa contribuir para um melhor entendimento dos mecanismos e particularidades de cada sistema, sem se comprometer, num primeiro momento, a obter a melhor performance dentre todas as propostas citadas.

A próxima seção trata de dois experimentos que visam obter uma compreensão complementar dos mecanismos de evolução das regras, e também dos desafios encontrados ao se empregar SCAs em tarefas robóticas reativas e não-reativas.

3.4 Experimentos

Esta seção apresenta experimentos relativos a duas tarefas robóticas distintas. Ambas consistem de propostas experimentais consolidadas, previamente descritas na literatura.

Como os sistemas SCA e XCS foram concebidos para codificação binária, e também por motivos de simplificação do algoritmo, optou-se pela codificação binária do antecedente e conseqüente do classificador em todos os sistemas. Apesar da discretização ocasionar uma perda de sensibilidade (efeitos de quantização na representação), verificou-se em Hurst et al. (2002) e Webb et al. (2003) que é possível a realização de experimentos simulados e reais utilizando esta técnica.

Os experimentos têm como base o robô Khepera II, descrito no Apêndice A. Para a codificação da parte antecedente, considerou-se o sensor lateral esquerdo, um sensor frontal e o sensor lateral direito do robô, resultando em uma *string* de 3 posições. Cada campo é formado da seguinte forma: caso a leitura sensorial seja superior a 50 (variando de 0, ausência de obstáculos, a 1023, obstáculo muito perto), o campo assume valor 1, e valor 0 caso contrário. Da mesma forma, o antecedente referente ao sensor de luminosidade foi codificado considerando-se valor 1 caso a leitura fosse inferior a 100 (variando de 0 a 500) e 0 caso contrário. O símbolo “#” também é utilizado. As ações do robô foram codificadas utilizando-se 2 bits: “00” e “11” indicam movimento para a frente, “01” virar 90° à direita e “10” virar 90° à esquerda. Exemplos de possíveis situações que o robô pode presenciar e suas ações estão na Tabela 3.3.

Para evitar uma sobrecarga de ações e iterações sem mudança efetiva do ambiente, adotou-se uma estratégia proposta em Hurst et al. (2002), onde o robô executa uma ação até que a leitura do ambiente se altere. Com isso, o número de mensagens irrelevantes ao sistema cai muito, e a convergência torna-se mais rápida.

Em todos os experimentos realizados, tanto para o SCA como para o XCS, a população inicial de regras é nula, e emprega-se a técnica de *covering*, descrita na seção 3.2.2.

Os parâmetros da simulação foram escolhidos com base em trabalhos anteriores na literatura. Para o SCA: Holland (1975), Goldberg (1989) e Vargas (2000); para o XCS: Butz & Wilson (2000), Webb et al. (2003) e Kovacs (2004).

3.4.1 Experimento 1

O primeiro experimento é realizado em uma variação do ambiente *woods*, previamente descrito. Este ambiente é um dos mais utilizados para o estudo de aprendizado em robôs (Wilson, 1995; Cliff & Ross, 1995; Lanzi, 1998; Lanzi & Wilson, 2000). Neste experimento, o ambiente não possui discretização, ou células, e, portanto, o robô possui um espaço contínuo de movimento. Observe a Figura 3.3(a). A luz representa o objetivo a ser atingido, e o robô deve aprender a alcançá-la no menor número de iterações possível, sem colidir com paredes ou obstáculos.

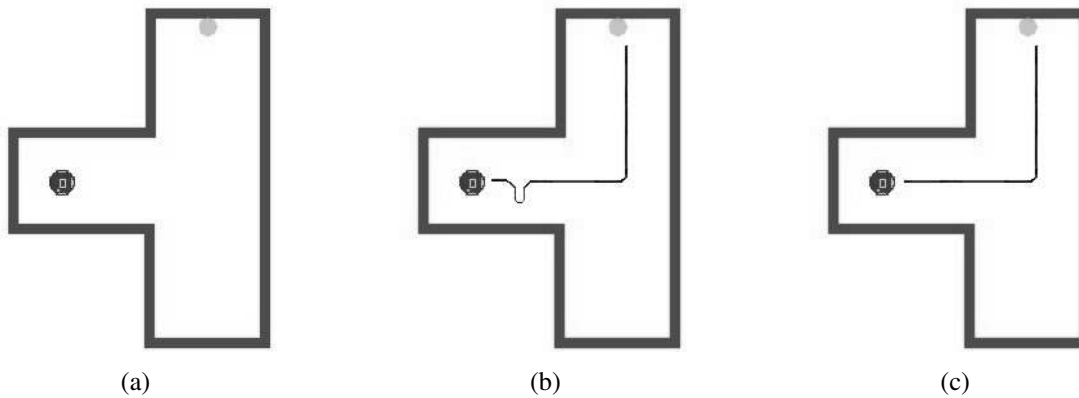


Fig. 3.3: Resultados dos experimentos simulados, incluindo a arena (a), a trajetória do SCA (b), e a trajetória do XCS (c).

Tab. 3.4: Resultados do SCA para o Experimento 1. A tabela apresenta as 5 regras com mais energia dentre as 45 regras presentes na população.

Antecedente de distância	Antecedente de luz	Consequente	Energia
0##	010	11	245,87
01#	000	10	109,98
001	000	11	81,29
100	000	11	64,64
001	000	01	62,55

Resultados do SCA

O SCA foi capaz de resolver o problema com sucesso. Diferentemente do XCS, o SCA atualiza a energia do classificador a cada passo (seção 3.2.1). Assim, a função de recompensa utilizada neste experimento foi a seguinte: 2 unidades, caso o robô se mova para frente sem colisão, 10 unidades para cada classificador ativado, caso chegue ao alvo, e 0, caso contrário. O processo evolutivo ocorre a cada 40 iterações, a cada sucesso de chegada ao alvo, ou caso ocorra colisão. De acordo com a Equação 3.1, a regra com maior lance é selecionada para agir.

A Tabela 3.4 ilustra alguns dos classificadores com maior energia na população evoluída, e a Tabela 3.5 contém os parâmetros da simulação. A Figura 3.3(b) ilustra a trajetória do robô.

Note, observando a Tabela 3.4 e a Figura 3.3(b), que a sequência de movimento do robô é predominantemente definida pelos classificadores com maior energia. Um ponto interessante a se notar é a diferença entre as energias dos classificadores. Como o sistema não possui um mecanismo de

Tab. 3.5: Parâmetros do Experimento 1 referentes ao SCA

Parâmetro	Valor
k_0	0,1
k_1	0,1
k_2	0,01
SPow	3
σ	0,01
n	200
Energia(0)	10
T_{Lance}	0,003
Taxa de mutação	0,05
Taxa de crossover	0,8
Probabilidade de <i>dont't care</i>	0,1

Tab. 3.6: Resultados do XCS para o Experimento 1

Antecedente de distância	Antecedente de luz	Ação	Predição	Erro de Predição	Fitness	Numerosidade	Experiência
0#0	000	10	265,00	6,50	1,00	2	95
#0#	#00	00	410,00	239,81	0,62	5	272
000	010	11	1000,00	0,00	1,000	24	81
#00	#00	00	442,00	278,07	0,62	1	168

distribuição de recompensa a longo prazo nem um mecanismo eficaz para a compactação de regras, como o que existe no XCS, diversas regras e seqüências de comportamento levam o robô a atingir o alvo, e não necessariamente a menor população e seqüência são evoluídas (note o pequeno desvio na trajetória, apresentado na Figura 3.3(b)). Assim, alguns classificadores acabam atuando em circunstâncias mais favoráveis, por exemplo, o primeiro classificador da Tabela 3.4, que indica luz à frente. Sempre que este classificador atuar, o robô chegará ao alvo e a recompensa maior será dada.

A convergência levou 12 gerações, com 45 classificadores gerados, ocorrendo 3 colisões (valores médios).

Resultados do XCS

O sistema classificador XCS foi capaz de resolver o problema com sucesso. Ao chegar até a luz, uma recompensa de 1000 era fornecida, e 0 caso contrário. Uma etapa do experimento consiste de um máximo de 40 iterações, terminando com o robô atingindo o estado alvo ou caso ocorra alguma

Tab. 3.7: Parâmetros do XCS para o Experimento 1

Parâmetro	Valor
θ_{sub}	20
θ_{GA}	25
θ_{del}	25
θ_{mna}	3
δ	0,1
N	400
β	0,2
α	0,1
μ	0,05
χ	0,8
$P_{\#}$	0,1
ϵ_0	10

colisão. As primeiras 10 iterações de uma etapa são feitas por exploração, e as 10 iterações seguintes por exploração. O algoritmo genético é aplicado apenas em iterações de exploração, ou ao final da etapa.

A Tabela 3.6 contém alguns dos classificadores evoluídos e a Tabela 3.7 contém os parâmetros da simulação. A Figura 3.3(c) ilustra a trajetória do robô.

Como o fitness de cada regra é determinado pela precisão com que faz uma predição, e não diretamente através da recompensa do ambiente, como ocorre com o SCA, os classificadores que possuem *fitness* próximo de 1 não são necessariamente os classificadores que prevêm a maior recompensa. Note que a recompensa é distribuída pela população (quarta coluna da Tabela 3.6). Isto é inerente ao algoritmo de atualização de fitness do XCS, o qual faz com que o último classificador a atuar antes do objetivo alcançado possua uma predição de recompensa muito maior do que os classificadores anteriores que levaram o sistema àquele estado. Como a escolha do classificador a atuar depende do vetor de predição de recompensa (PA) (Seção 3.2.2), o sistema evolui para o caminho mínimo até o alvo, evitando situações como a presente no SCA, em que a convergência para o menor caminho possível não é garantida. Ambientes mais complexos causam ao XCS o mesmo problema de não obtenção do caminho mínimo, observado no SCA, conforme descrito em Webb et al. (2003). É importante notar também que os classificadores mais adaptados possuem uma maior numerosidade, ou seja, o número de classificadores que este classificador representa e substitui na população aumenta.

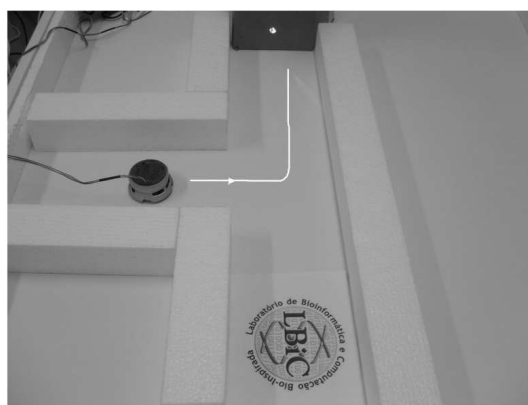
O sistema XCS convergiu para o comportamento esperado após 17 gerações, com 38 classificadores gerados, e ocorrendo 9 colisões (valores médios).

Experimento em ambiente real

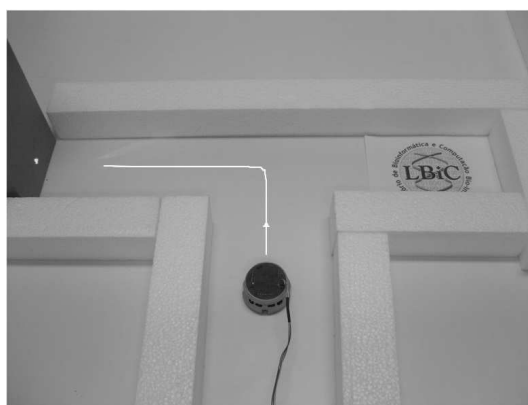
Finalizando a série de experimentos, validou-se os controladores simulados citados anteriormente em um ambiente real. Esta é uma etapa importante do trabalho, dado que a implementação em robôs reais é considerada fundamental, por alguns pesquisadores, para determinar-se o sucesso de uma técnica de controle.

Conforme será explicitado no Apêndice A, o robô real apresenta algumas diferenças em relação ao robô simulado. Os sensores possuem uma leitura em ambiente aberto de aproximadamente 80 (variando de 0, ausência de obstáculos, a 1023, obstáculo muito perto), enquanto que o robô simulado apresenta leitura de aproximadamente 5. Por isso, uma adaptação deve ser realizada, tornando as leituras entre o simulador e o robô compatíveis. Os encoders que indicam o movimento das rodas também possuem precisão diferente. Os sensores de luminosidade se comportam de maneira mais fiel. Porém, devido aos problemas mencionados no Apêndice A com relação ao seu uso conjunto com os sensores de distância, torna-se necessário utilizá-los com mais cuidado. Por isso, os experimentos são realizados em um ambiente escuro, procurando utilizar uma fonte de luz direcionada, com pouco espalhamento, visando não comprometer o movimento do robô nos demais pontos do ambiente. A Figura 3.4 ilustra o ambiente real e as trajetórias do robô quando controlado por cada sistema, SCA e XCS.

O robô apresentou boa performance no experimento real quando comparado ao experimento simulado, isto é, o comportamento do robô atendeu aos requisitos da tarefa de maneira similar ao modo como foi realizado no simulador. Deve-se enfatizar que as regras continuam em evolução no ambiente real, seguindo o mesmo algoritmo que foi implementado em simulação. Não se verificou aqui, no caso do SCA, a pequena “dobra” na trajetória ocorrida na Figura 3.3(b), possivelmente em virtude das diferentes temporizações de leitura sensorial e atuação.



(a) SCA



(b) XCS

Fig. 3.4: Resultados do Experimento Real.

3.4.2 Experimento 2

Em navegação autônoma de robôs, há uma busca pela síntese de controladores que sejam capazes de expressar comportamento não-reativo (Jakobi, 1998; Husbands, 1998; Nolfi & Floreano, 2000), ou seja, o comportamento do robô não depende apenas das leituras instantâneas dos sensores, diferentemente do que ocorre quando o comportamento é puramente reativo.

O experimento abordado nesta seção explora a capacidade de memória do agente, um dos requisitos para se expressar comportamentos complexos, conforme verificado em Webb et al. (2003). O experimento proposto é relatado em Jakobi (1998), valendo-se da evolução de redes neurais artificiais para o controle do robô. Entretanto, neste trabalho serão utilizados sistemas SCA e XCS. A tarefa está ilustrada na Figura 3.5.

Trata-se de um ambiente T-Maze, onde o robô inicia sua navegação pelo primeiro corredor, passa

por uma fonte luminosa e deve decidir-se por um dos lados para virar ao final deste primeiro corredor. Caso vire para o mesmo lado da luz e atinja o final deste segundo corredor (local objetivo), a máxima recompensa é fornecida. Ou seja, o sistema de controle deve se lembrar da posição da fonte luminosa. Este comportamento envolve a capacidade de se movimentar sem colisões pelo ambiente bem como de lembrar-se para que lado virar ao final do primeiro corredor.

Como a tarefa agora é mais complexa, a causalidade de todo o sistema é crucial. Em um cenário estático, entende-se por causalidade o fato de que a repetição de uma mesma ação, em uma mesma posição do cenário, leva sempre às mesmas condições externas futuras.

Por causa das ações contínuas e devido ao ruído inerente e imprecisões no robô e cenário, o robô tem que lidar com uma tarefa multi-objetivo (desvio de obstáculos e encontro do local objetivo) concomitantemente com esse problema da causalidade. Quando a tarefa é relativamente simples, o sistema é capaz de lidar com esses aspectos. Porém, ao enfrentar tarefas mais complexas, esses tipos de incertezas podem levar ao mal funcionamento do sistema e também a uma interpretação errada dos resultados. Com o intuito de amenizar estas incertezas, o ambiente foi dividido em células. Desta forma, pode-se fazer uma análise em cada ponto do cenário, e entender a sequência de ações que conduziram o robô ao comportamento desejado. Ao atuar, o robô pode ir para frente ou girar 90°, para a direita ou esquerda. Este mecanismo é especialmente útil no sistema XCS, por causa de sua dificuldade em lidar com longas sequências de ações.

Resultados do SCA

A estrutura do antecedente do classificador é similar à descrita no Experimento 1, e pode ser interpretada com o auxílio da Tabela 3.3. O esquema de recompensa e os parâmetros são os mesmos. Uma etapa consiste de um máximo de 20 iterações, finalizando caso o robô alcance o estado alvo, ou se uma colisão ocorrer.

O sistema classificador implementado não usa o sistema de transmissão de recompensa a longo prazo chamado *bucket brigade*, descrito em Holland (1975). Alternativamente, o bônus final é igualmente distribuído entre os classificadores que foram ativados do início ao fim de uma etapa de simulação. Riolo (1989) investigou esse mecanismo e concluiu que ele age de forma similar ao *bucket brigade*.

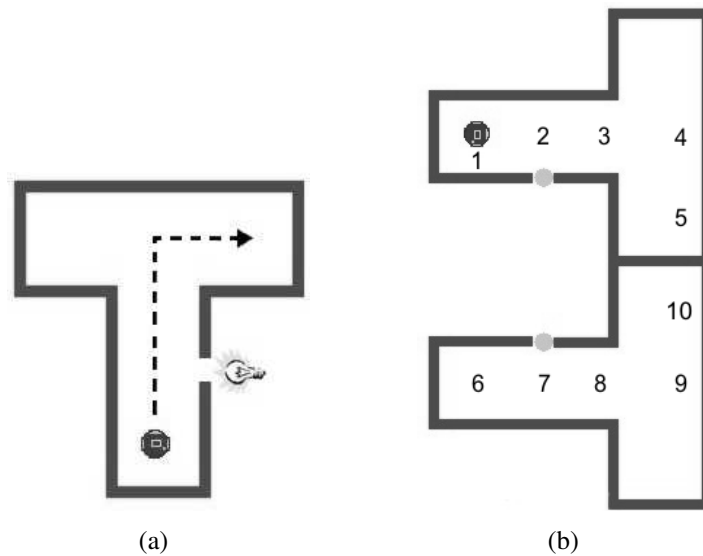


Fig. 3.5: Ambiente de simulação para o Experimento 2 (adaptado de Jakobi (1998)) (a) e implementação do experimento utilizando o simulador WSU (b). A cada rodada de simulação, o robô é posicionado aleatoriamente na posição 1 ou 6 de um dos T-Maze, um com lâmpada à direita e outro com lâmpada à esquerda.

Os primeiros resultados não foram animadores. O sistema era incapaz de se adaptar às diferentes posições da fonte luminosa. Os controladores convergiam para um comportamento correto apenas para uma das opções de fonte luminosa, ou seja, algumas vezes o controlador virava sempre para a esquerda, outras apenas para a direita. Isso é explicado pelo fato de haver uma ambigüidade ao final do corredor. Após percorrer o corredor inicial e chegar ao fim, de frente para a parede, o robô não tem como se lembrar de qual lado estava a luz, a mensagem que vem do ambiente é simplesmente *“parede a frente, ausência de luz”*.

Tornou-se necessário utilizar uma forma de memória interna. Alguns trabalhos utilizando sistemas classificadores com memória interna foram estudados (Riolo, 1989; Lanzi, 1998; Webb et al., 2003), e decidiu-se utilizar um bit extra no antecedente, correspondente à memória interna do robô, e uma ação extra, também interna, que pode modificar o valor deste bit. Assim, o classificador é escolhido com base na mensagem do ambiente e com base no estado interno do robô. O robô pode agir atuando sobre o ambiente e alterando sua memória interna.

Com estas modificações, o sistema foi capaz de encontrar uma solução. A Tabela 3.8 ilustra a sequência de classificadores evoluídos no problema. Perceba que a memória interna é amplamente utilizada, aparentemente de forma pouco coerente, porém de maneira complementar, observando-se

Tab. 3.8: Resultados do SCA para o Experimento 2

Lâmpada à esquerda						
Posição no Cenário	Antecedente de distância	Antecedente de luz	Memória Interna	Conseqüente	Conseqüente de Memória	Energia
6	1##	0##	0	11	0	974,67
7	101	100	0	11	1	1092,74
8	101	000	1	11	1	1093,43
9	0#0	0#0	1	10	0	1094,12
9	001	000	0	11	1	1094,81
10	001	000	1	11	0	1095,15
Lâmpada à direita						
Posição no Cenário	Antecedente de distância	Antecedente de luz	Memória Interna	Conseqüente	Conseqüente de Memória	Energia
1	1##	0##	0	11	0	974,67
2	1##	0##	0	11	0	975,16
3	1##	0##	0	11	0	976,05
4	010	000	0	01	1	998,61
4	100	000	1	11	0	999,30
5	1##	0##	0	11	0	976,23

os dois cenários simultaneamente (lâmpada à esquerda e lâmpada à direita). Os classificadores na Tabela 3.8 estão em ordem de utilização, permitindo perceber a forma como o sistema resolveu e representou a solução para o problema.

Este experimento permitiu evidenciar que o sistema classificador tradicional (SCA) é capaz de resolver tarefas com recompensa a longo prazo, o que não é tão comum na literatura.

Resultados do XCS

Para lidar com estados de ambiguidade, o sistema XCS foi modificado com um mecanismo de memória interna, descrito em Lanzi & Wilson (2000). Ele consiste da adição de um bit extra na parte antecedente, o qual identifica-se com um registrador de memória interno. Adiciona-se também um bit de ação interna, que é capaz de modificar o estado do registrador de memória. A seleção da ação interna é sempre por exploração. Uma etapa consiste de um máximo de 20 iterações, finalizando com o robô alcançando o estado alvo, ou com a ocorrência de uma colisão. Ao início de cada etapa, o sistema determina aleatoriamente se a etapa será exploratória ou explotatória. O algoritmo genético é aplicado apenas em fases exploratórias, ou ao final de uma etapa.

O XCS foi capaz de resolver a tarefa com sucesso, mas algumas considerações devem ser feitas. Normalmente, o XCS recebe a máxima recompensa (1000) quando atinge o estado alvo, e 0 caso contrário. Seguindo essas prerrogativas, que são adotadas em quase todos os trabalhos com XCS, o sistema foi muito instável, no sentido de que boas soluções foram difíceis de se encontrar. Isto pode ser explicado pela longa sequência de ações a serem desempenhadas nesse problema, associado ao uso de memória. O robô deve chegar ao estado alvo para receber a recompensa, e então esta recompensa é distribuída para a população através da interação das regras. O robô chega ao estado alvo pela primeira vez apenas por acaso, não existindo nenhuma recompensa até então. Portanto, a seleção de ações via exploração é inútil. Quando o robô finalmente atinge o estado final, ele recebe a recompensa, mas apenas a última regra a ativar recebe esta recompensa, todas as outras regras na população têm sua predição de recompensa igual a zero. Apenas quando o robô aproxima-se do estado alvo novamente a recompensa pode ser passada para a regra anterior à última regra a ser ativada. Este processo continua até que o mapa de recompensa para essa tarefa esteja completamente formado. Para ilustrar a complexidade desta tarefa, considere o trabalho desenvolvido por Lanzi (1998), que evoluiu um agente usando modificações no XCS similares às feitas nesse trabalho. O ambiente e as ações a serem expressas são discretas. A tarefa não é a mesma estudada aqui, porém, em essência, elas são similares. Nessa tarefa, o desempenho médio ótimo do agente foi 2,9 passos até alcançar o estado alvo e o número médio de passos aleatórios até o estado alvo era 110. Neste trabalho, o número mínimo de passos até o estado alvo é 6. Portanto, o número médio de passos aleatórios até o estado alvo torna-se muito maior.

A solução adotada foi recompensar o robô em 1000 ao atingir o estado alvo, e em 20 quando ele move-se para a frente, pois mover-se para a frente seria a ação mais presente em um sistema evoluído, ajudando a reduzir o tempo de exploração (para esta tarefa, lembre-se que o robô apenas necessita virar ao final do primeiro corredor). Com esta modificação, o robô foi capaz de resolver eficientemente a tarefa. No entanto, o sistema opera agora com recompensas a cada passo, diferentemente da proposta original do sistema XCS.

A Tabela 3.9 mostra um exemplo de algumas regras de uma população evoluída com sucesso. O primeiro ponto a destacar é o uso do registrador de memória. É possível observar que o lado da luz é determinante para o chaveamento do estado interno. O *fitness* é alto e o erro de predição é baixo para

quase todas as regras, exceto para as regras que são ativadas em mais de uma localização no cenário: o número de passos até o estado alvo é diferente, entretanto a leitura sensorial é a mesma, o que leva a predições de recompensa diferentes, que é responsável pelo erro de predição. Note também que, embora o objetivo de virar corretamente ao final do corredor foi alcançado, uma população compacta de regras generalistas ao máximo não foi obtida. Isso pode ocorrer devido ao período de simulação, pois regras mais gerais não dispuseram de tempo suficiente para obter alta precisão e valor de *fitness*.

Tab. 3.9: Resultados do XCS para o Experimento 2

Lâmpada à esquerda										
Posição no Cenário	Antecedente de distância	Antecedente de luz	Memória interna	Ação	Ação interna	Predição	Erro de predição	Fitness	Numerosidade	Experiência
6	101	000	0	00	1	251,00	91,90	0,37	15,00	2151,00
7	101	1#0	1	00	0	180,00	4,18	1,00	17,00	524,00
8	101	000	0	00	1	230,00	94,16	0,32	15,00	2152,00
8	#10	0#0	1	10	0	506,00	4,67	1,00	20,00	465,00
9	0##	000	0	11	1	723,00	4,16	1,00	6,00	818,00
10	001	##0	1	11	0	996,00	4,00	1,00	11,00	485,00
Lâmpada à direita										
Posição no Cenário	Antecedente de distância	Antecedente de luz	Memória interna	Ação	Ação interna	Predição	Erro de predição	Fitness	Numerosidade	Experiência
1	101	000	0	00	1	259,00	105,18	0,38	15,00	2153,00
2	1#1	001	1	11	1	282,00	4,25	1,00	38,00	430,00
3	101	0#0	1	00	0	375,00	4,97	1,00	8,00	462,00
4	#1#	000	0	01	1	507,00	4,91	1,00	12,00	407,00
4	1#0	##0	1	11	0	721,00	4,56	1,00	5,00	293,00
5	100	#00	0	00	1	996,000	4,00	1,00	19,00	644,00

3.5 Discussão

O objetivo deste capítulo foi estabelecer uma análise da performance do sistema classificador baseado em energia, SCA, e do sistema classificador baseado em precisão, XCS, quando aplicados a duas tarefas de navegação distintas. A primeira tarefa é puramente reativa, a segunda é não-reativa. O texto apresentou uma análise da evolução de regras, provendo exemplos de populações evoluídas e suas peculiaridades para cada um dos dois tipos de SCAs estudados.

Ambos os sistemas foram capazes de resolver as tarefas com sucesso. No entanto, o SCA com base em energia encontrou soluções nas quais a menor sequência não era sempre presente. De fato, as regras apresentaram sobre-especialização e sobre-generalização. Isso acontece principalmente porque as regras são recompensadas pelas situações mais frequentes. Portanto, essas regras continuam a atuar em contextos nos quais as mesmas não são adequadas. No domínio da robótica, essas características são prejudiciais à performance, à velocidade de evolução e à robustez do sistema. Entretanto, como a recompensa ocorre imediatamente após uma ação, em alguns casos é mais fácil encontrar uma boa solução usando-se o SCA em comparação com o XCS. Além do mais, o SCA, mesmo apresentando algumas limitações quando comparado a propostas mais recentes, pode levar a idéias inovadoras em trabalhos futuros, pois ele é o modelo que inspirou a criação de novos tipos de sistemas classificadores.

O controle autônomo do robô móvel deve lidar com ambientes dinâmicos e ruidosos. O sistema XCS, por características construtivas, propaga a recompensa através das regras mais recentemente utilizadas. Para se obter a evolução de determinado comportamento, um conjunto de regras deve ser exposto à mesma situação muitas vezes. Com o objetivo de diminuir o custo computacional, a discretização do ambiente e das ações pode ser adotado. Webb et al. (2003) utiliza uma discretização em nível sensorial. Isso causa uma redução no espaço de busca, ajudando a evolução do sistema. Entretanto, essa discretização ou qualquer outro método que incorpore um conhecimento *a priori* reduz o nível de autonomia do sistema, pois este conhecimento deveria ter sido produzido apenas pelo robô e seu sistema de aprendizado. Uma alternativa pode ser observada no trabalho de Studley & Bull (2005), que incorpora o tempo à função de recompensa, criando o X-TCS, aplicando-o a ambientes contínuos, com ações contínuas.

Outro tópico importante associado aos experimentos é a determinação da função de recompensa.

Kovacs (2004) mostra que definir recompensas diferentes para cada comportamento pode comprometer a capacidade de generalização do sistema. Assim, uma questão imediata é: como definir uma recompensa para cada ação? Ainda, como definir punições para ações incorretas em comportamentos críticos (no caso, evitar colisões com obstáculos)? A alternativa proposta por Kovacs (2004) é mudar a frequência com que cada uma dessas situações é encontrada. Neste sentido, situações mais relevantes são presenciadas mais frequentemente. Outra opção seria separar o conjunto de regras em subconjuntos, deixando conjuntos que representam situações mais relevantes com um número maior de regras. No entanto, definir o que é importante ou não é extremamente desafiador, pois a emergência de comportamentos complexos pode surgir de uma sequência de processos simples, que seriam suprimidos por essas abordagens. Infelizmente, esse problema ainda não foi bem explorado na literatura.

Igualmente relevante é a análise dos métodos de seleção de ação. Alguns trabalhos na literatura têm atribuído alta relevância ao modo com que a seleção de ações é definida, devido a variações significantes no desempenho final do sistema (Butz et al., 2003; Kovacs, 2004; Studley & Bull, 2005). As técnicas mais analisadas são roleta, torneio e uma combinação de exploração e exploração. Novamente, a literatura carece de contribuições nessa área no contexto de navegação autônoma de robôs.

O XCS é, atualmente, o SCA mais estudado e utilizado (Butz et al., 2003; Studley & Bull, 2005). Ele possui evolução e mecanismos de aprendizado que previnem a ocorrência de regras sobre-generalistas com alto *fitness*. Outro ponto que justifica o uso do XCS é sua habilidade em distribuir recompensas em tarefas sequenciais e não considerar o desempenho imediato, como ocorre no SCA. Isso é implementado principalmente pelo mecanismo de atribuição de crédito inspirado na técnica *Q-Learning*. A conclusão é que o XCS é capaz de mapear, de maneira mais balanceada, o espaço de ações do problema. Esse aspecto tende a promover uma melhor performance do sistema.

Como observação final, é pertinente salientar que o foco do capítulo não é comparar as performances de dois SCAs em navegação autônoma de robôs, mas analisar qualitativamente o comportamento de cada um em função de suas peculiaridades. Essa análise foi feita apresentando-se revisões e aspectos positivos de cada um em cada tarefa, enfatizando os mecanismos de evolução de regras (forneendo exemplos de populações evoluídas). Espera-se que esta análise contribua para uma melhor compreensão geral do papel que os SCAs podem representar em tarefas cognitivas, mais especifica-

mente em navegação autônoma de robôs.

Capítulo 4

Sistema Homeostático Artificial Evolutivo

Resumo - Este capítulo aborda os conceitos relacionados a homeostase e suas implicações na evolução e adaptação em sistemas biológicos. Descreve-se uma proposta de sistema homeostático artificial sob controle evolutivo, composto por redes neurais artificiais do tipo NSGasNet, e um sistema endócrino artificial. Primeiramente, apresentam-se os detalhes e justificativas da proposta, bem como as técnicas e configurações experimentais adotadas. Em seguida, descrevem-se os experimentos realizados que visam identificar as propriedades do sistema. Por fim, analisa-se o conjunto de resultados e suas repercussões.

4.1 Homeostase

O termo homeostase tem sua origem no trabalho do fisiologista francês Claude Bernard (1813-1878) que introduziu o conceito de ambiente interno. Mais tarde, o termo foi expandido por Cannon, e denominado homeostase (Cannon, 1929). Contudo, para Pfeifer & Scheier (1999), o termo homeostase somente foi completamente definido pelo psiquiatra inglês William Ross Ashby (Ashby, 1952). Para Ashby, a habilidade de adaptar-se a uma mudança contínua em ambientes imprevisíveis, chamada adaptividade (“*adaptivity*”, na literatura em inglês), tem uma relação direta com a inteligência. Durante o processo adaptativo, algumas variáveis internas precisam ser mantidas dentro de

limites predeterminados, seja por mudanças evolutivas, reações fisiológicas e ajustes sensoriais, ou simplesmente por aprendizado de novos comportamentos. Portanto, devido a essa propriedade regulatória, ou homeostase, o organismo ou o agente artificial pode operar e permanecer vivo em uma zona de viabilidade. Além disso, Dyke e Harvey (Dyke & Harvey, 2005, 2006) destacam que, para entender a vida artificial ou real, é primordial entender primeiro os mecanismos básicos relacionados à homeostase. No corpo humano, alguns receptores sensoriais acionam respostas específicas nos sistemas nervoso, imunológico e endócrino, que são os principais sistemas diretamente relacionados ao processo de homeostase (Besendovsky & Del Rey, 1996).

A teoria apresentada por Ashby tem motivado aplicações do conceito de homeostase na síntese de sistemas autônomos em robótica móvel (Di Paolo, 2000; Neal & Timmis, 2003; Harvey, 2004; Hoinville & Hénaff, 2004; Vargas et al., 2005; Moiola et al., 2008). Explorando propriedades particulares de processos homeostáticos, Di Paolo (2000) desenvolveu um controlador neural adaptativo, no qual as conexões entre os neurônios estavam sujeitas a algumas regras plásticas, isto é, o valor dos pesos das conexões entre os neurônios era alterado sempre que a atividade do nó ultrapassava determinados limites. Como o poder computacional de um neurônio está relacionado a seu nível de saturação, esta abordagem homeostática busca evitar saturações, fornecendo alternativas ao sistema para manter seu estado interno quando submetido a perturbações. O caso escolhido para análise foi uma perturbação relacionada à inversão do campo visual, um problema neuro-psicológico investigado em humanos e animais (Welch, 1974; Spillman & Wooten, 1984). Mostrou-se que o controlador foi capaz de se adaptar a perturbações sensório-motoras, e os robôs mantiveram um certo grau de estabilidade. Entretanto, como enfatizado pelo autor, serão necessárias mais investigações para melhor compreensão dos fenômenos envolvidos.

Seguindo algumas idéias do trabalho de Di Paolo (2000), Hoinville & Hénaff (2004) apresentaram um estudo preliminar das vantagens de dois mecanismos homeostáticos bio-inspirados em controladores neurais aplicados a robôs móveis com múltiplos membros. Como resultado, mostrou-se que o potencial evolutivo, a estabilidade e a habilidade de rejeitar perturbações por parte dos controladores neurais plásticos melhoram quando mecanismos homeostáticos são incorporados a eles.

Harvey (2004) investigou a adaptação homeostática em um modelo simplificado, chamado modelo “*Daisyworld*”, usado para explicar a adaptação de margaridas (“*daisies*”, na literatura em in-

glês) a diferentes condições climáticas em um planeta imaginário. Neste modelo, existem dois tipos de margaridas, o tipo preto (que absorve mais o calor) e o tipo branco (que reflete mais o calor). O crescimento e sobrevivência das margaridas depende apenas da temperatura, e é expresso por uma função denominada função “*Hat*” (uma função com formato similar a um chapéu). Por consequência, a temperatura do planeta em questão é determinada pelas taxas de absorção e reflexão de luz por cada um dos tipos de margaridas. A dinâmica de crescimento, absorção e reflexão de luz das margaridas, associada a uma realimentação do ambiente, reflete o chamado “controle por rédeas”(rein control), uma alusão à dinâmica de estímulo e supressão das margaridas em busca de um equilíbrio (zona de viabilidade). Mostra-se, então, que processos homeostáticos inspirados nesse modelo podem ser obtidos, e a temperatura do planeta possui certa robustez diante de perturbações externas (por exemplo, um aumento da luminosidade solar).

As idéias apresentadas por Di Paolo (2000) e Hoinville & Hénaff (2004) englobam homeostase dentro de uma estrutura única, isto é, uma única rede neural artificial capaz de alterar dinamicamente as regras de suas conexões. Alternativamente, o presente trabalho propõe um modelo de sistema homeostático artificial (SHA) inspirado no trabalho de Vargas et al. (2005), formado por redes neurais artificiais e um sistema endócrino artificial. Neste modelo, utilizou-se primeiramente um agente robótico simulado e posteriormente um robô real, os quais deviam aprender inicialmente e de forma independente duas tarefas distintas: seguir paredes evitando colisões e detectar uma fonte de luz (associada a uma fonte de energia). O algoritmo de aprendizagem por retropropagação (Haykin, 1999) foi empregado para treinar separadamente duas redes neurais artificiais do tipo perceptron multi-camadas a partir de dados amostrados. Esses dados foram coletados via amostras de posições predeterminadas do robô na arena, as quais representavam situações de navegação relevantes. O sistema endócrino artificial, então, coordena as duas redes neurais artificiais modulares através da atuação do nível de concentração hormonal diretamente na saída das redes.

As diferenças em relação ao trabalho anterior de Vargas et al. (2005) podem ser expressas em dois pontos principais: (1) emprego de evolução no treinamento das redes neurais artificiais e na determinação dos parâmetros de controle do sistema endócrino artificial; e (2) substituição das redes neurais tipo MLP por redes do tipo NSGasNet (Vargas et al., 2007). A utilização de evolução no projeto de um SHA justifica-se pelas vantagens da evolução de um sistema em comparação a um

treinamento prévio, tais como (Nolfi & Floreano, 2000; Gigliotta & Nolfi, 2008):

- capacidade de incorporação de auto-organização;
- menor necessidade de supervisão, dado que o sistema é avaliado como um todo e por seu desempenho global, diminuindo o grau de conhecimento inicial inserido pelo projetista;
- por ser avaliado pelo seu desempenho global e por incorporar auto-organização, um sistema evolutivo sofre menos com a necessidade de definir recompensas para cada ação desempenhada, uma tarefa muito difícil já que a solução de uma tarefa pode ser resultado de uma série de ações sem coerência evidente;
- a liberdade na procura por soluções, possibilitada pela evolução, pode ajudar na compreensão de fenômenos e comportamentos naturais.

Já a utilização das redes do tipo NSGasNet justifica-se pelo fato de que esse tipo de arquitetura já expressa, por si só, algum tipo de homeostase e emprega evolução, conforme descrito no Capítulo 2. Além disso, o emprego de modelos de redes neurais artificiais em robótica evolutiva é amplamente motivado na literatura (Nolfi & Floreano, 2000), por apresentarem, dentre outras características:

- espaço de busca relativamente suave (mudanças graduais nos parâmetros da rede produzem mudanças graduais em seu comportamento);
- diferentes níveis de adaptação (evolução, desenvolvimento, plasticidade);
- suporte a mapeamentos entrada-saída contínuos;
- robustez a perturbações e ruídos.

De acordo com Arkin (1998), sistemas puramente reativos e sistemas puramente deliberativos possuem limitações quando analisados isoladamente. Ainda de acordo com o autor, existem evidências de que versões híbridas, que incorporam ambos os sistemas, são observadas na natureza, apresentando integração hierárquica e planejamento de ações e reações. Complementarmente, Pfeifer & Scheier (1999) consideram que as consequências das ações, e portanto do comportamento, devem ser

capazes de modular o valor dos pesos relativos às sinapses para que haja adaptação. Assim, a arquitetura do sistema homeostático proposta incorpora esses dois conceitos, tidos como fundamentais no projeto de sistemas inteligentes dentro do contexto da ciência cognitiva incorporada.

A seção 4.2 descreve modelos de sistemas endócrinos artificiais, incluindo o proposto por Vargas et al. (2005), o qual será utilizado como base para o desenvolvimento de um sistema homeostático evolutivo, a ser apresentado na seção 4.3.

4.2 Sistema Endócrino Artificial - SEA

O sistema endócrino na espécie humana caracteriza-se por contribuir para a manutenção da homeostase, regular o crescimento e a diferenciação celular, atuar no metabolismo, no desenvolvimento e na reprodução, e ajudar o organismo a trabalhar em diferentes estágios de operação ou estados fisiológicos (Guyton & Hall, 1996). Todas essas tarefas são realizadas por hormônios, os quais são substâncias químicas produzidas, armazenadas e secretadas por células glandulares (ou glândulas), tidas como principais componentes do sistema endócrino (Purves et al., 2001). Alguns hormônios são secretados na corrente sanguínea, sendo assim capazes de alcançar virtualmente todas as células do corpo. Dessa forma, os hormônios podem provocar respostas específicas em células do corpo que estão em locais distantes da fonte de secreção. A liberação hormonal também pode afetar o sistema nervoso que, por sua vez, pode transmitir impulsos nervosos, afetando com isso a produção e secreção hormonal. Estabelece-se, deste modo, um mecanismo de retroalimentação.

Hormônios são chamados também de ligantes, por se conectarem a receptores. Receptores são moléculas de proteínas localizadas tanto na membrana plasmática como no citoplasma de uma célula. Essa conexão é normalmente responsável por uma resposta celular. Nesse sentido, cada célula possui seus próprios receptores, que irão influenciar a sua resposta a hormônios específicos. A presença de receptores é vital para controlar essas respostas celulares frente à sinalização realizada por diversos tipos de hormônios.

Timmis & Neal (2004) realizaram uma tentativa de modelagem matemática para um sistema endócrino artificial aplicado à navegação autônoma de robôs. De acordo com os autores, a idéia era criar um sistema que proporcionasse a capacidade de manutenção do equilíbrio interno de um agente

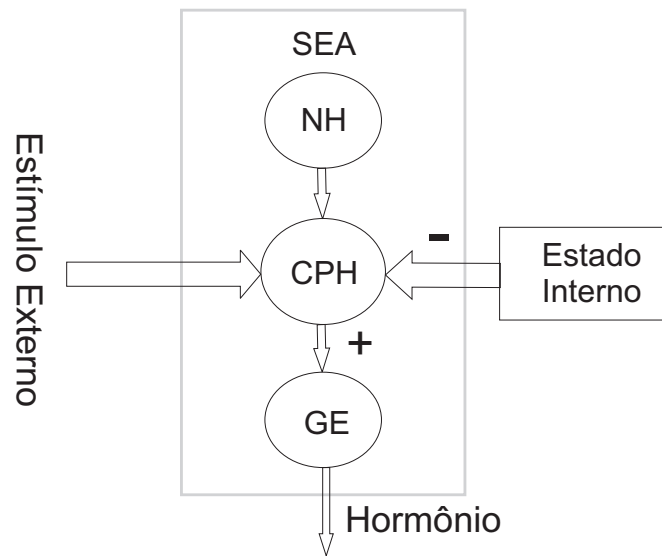


Fig. 4.1: Arquitetura do Sistema Endócrino Artificial (SEA). Aqui estão representados seus três módulos principais: módulo de nível hormonal (NH), módulo controlador de produção hormonal (CPH), e módulo glândula endócrina (GE).

(*e.g.*, um robô) enquanto o mesmo interage com um ambiente externo (Neal & Timmis, 2003). Nesta proposta, o sistema endócrino artificial é composto por equações que representam um agregado de células glandulares que secretam hormônios, os quais são anteriormente armazenados em um tipo de reservatório, em resposta a estímulos externos.

Em busca do desenvolvimento de um sistema mais plausível biologicamente, Vargas et al. (2005) propuseram uma extensão ao modelo de Timmis & Neal (2004). Os hormônios agora são produzidos e liberados sob demanda, através de glândulas artificiais. Em linhas gerais, a partir do início da liberação hormonal, essas glândulas devem controlar sua secreção por meio do nível de concentração dos hormônios (estado interno) e das substâncias vinculadas ao processo (estímulo externo). Todo o processo de realimentação é representado pelo acoplamento de equações a diferenças. Este sistema endócrino artificial foi parte componente de uma proposta de sistema homeostático artificial utilizado como sistema de controle em navegação autônoma de robôs. A Figura 4.1 ilustra o sistema proposto.

Essa proposta de sistema endócrino artificial consiste de três módulos principais: módulo de nível hormonal (NH), módulo controlador de produção hormonal (CPH), e módulo glândula endócrina (GE). O módulo NH tem um registro do nível hormonal no organismo; o CPH é responsável por controlar a produção hormonal em resposta às variações no estado interno do organismo e estímulos

externos; e a GE recebe sinais do CPH, sendo responsável pela produção e secreção hormonal, quando necessário.

O módulo CPH emite sinais excitatórios, que agem como retroalimentação positiva para a glândula GE que, por sua vez, inicia a produção e secreção hormonal (Equação 4.1), aumentando o nível hormonal presente no sistema. O nível hormonal irá alterar o estado interno (Equação 4.2), induzindo respostas diferentes por parte do agente (ações no ambiente). Ao detectar sinais inibitórios, indicados pelo estado interno e que promovem retroalimentação negativa, o módulo CPH cessa gradativamente a produção de sinais excitatórios até que novamente sejam detectadas mudanças específicas no estado interno.

$$HP(t+1) = \begin{cases} 0 & , EI < \theta \\ (100 - \%EE) \times \alpha(Max(NH) - NH(t)) & , \text{caso contrário} \end{cases} \quad (4.1)$$

em que

θ é o limiar do estado interno EI;

HP refere-se à produção hormonal;

EE é o estímulo externo;

α é o fator de escala;

NH é o nível hormonal;

t é o índice de tempo.

Se o estado interno EI é maior ou igual ao limiar, então produz-se hormônio a uma taxa que depende do nível de estímulo externo recebido e do nível hormonal já presente no organismo artificial. Por outro lado, se o estado interno EI for menor que o limiar θ , então interrompe-se a produção hormonal. NH e EI possuem um valor máximo de 100.

$$EI(t+1) = \begin{cases} 0 & , (EE \geq \lambda) \text{ e } (NH \geq \omega) \\ EI(t) + \beta(Max(EI) - EI(t)) & , \text{caso contrário} \end{cases} \quad (4.2)$$

em que

λ e ω são limiares pré-determinados associados a EE e NH , respectivamente;

β é a taxa de crescimento do estado interno.

O nível hormonal (NH) representa a quantidade hormonal estimulando o sistema. Seu valor é submetido à atualização constante de acordo com o valor de sua meia-vida (parâmetro T) e à quantidade de hormônio produzida (HP) (Equação 4.3):

$$NH(t+1) = NH(t) \times e^{-1/T} + HP(t) \quad (4.3)$$

É importante enfatizar que qualquer variação nos estados internos e estímulos externos pode promover ou suprimir a atividade dos sistemas nervoso (RNAs) e endócrino (SEA) artificiais. Por exemplo, a variação do estado interno do organismo devido à produção hormonal pode agir como um mecanismo de retroalimentação para a própria produção hormonal, resultando na liberação de hormônios inibidores ou na suspensão da produção de hormônios.

A seção 4.3 introduz e destaca os módulos e mecanismos utilizados na evolução de um sistema homeostático artificial proposto neste tese, formado por duas redes neurais do tipo NSGasNet (descritas no capítulo 2) e um sistema endócrino artificial. Detalhes dos experimentos sugeridos para o estudo e validação da proposta são apresentados, observando-se as propriedades do sistema e seu desempenho em situações que possam evidenciar seu comportamento homeostático.

4.3 Sistema Homeostático Artificial Evolutivo

4.3.1 Descrição e justificativa do modelo

Baseado no trabalho de Vargas et al. (2005), a proposta de sistema homeostático artificial a ser apresentada fundamenta-se nas interações neuro-endócrinas observadas no ser humano, as quais foram descritas no início deste capítulo.

O novo sistema homeostático artificial evolutivo (SHAE) é composto por um sistema endócrino artificial e duas redes do tipo NSGasNet (Figura 4.2). Propõe-se não apenas a evolução das redes neurais artificiais, mas também do sistema endócrino artificial, visando o desenvolvimento de um sistema homeostático artificial específico para cada cenário de aplicação. Além disso, uma versão estendida do mesmo sistema é desenvolvida para incluir receptores, em uma tentativa de direcionar a influência dos hormônios liberados pelo SEA. Esse direcionamento adicional mostrou-se capaz de

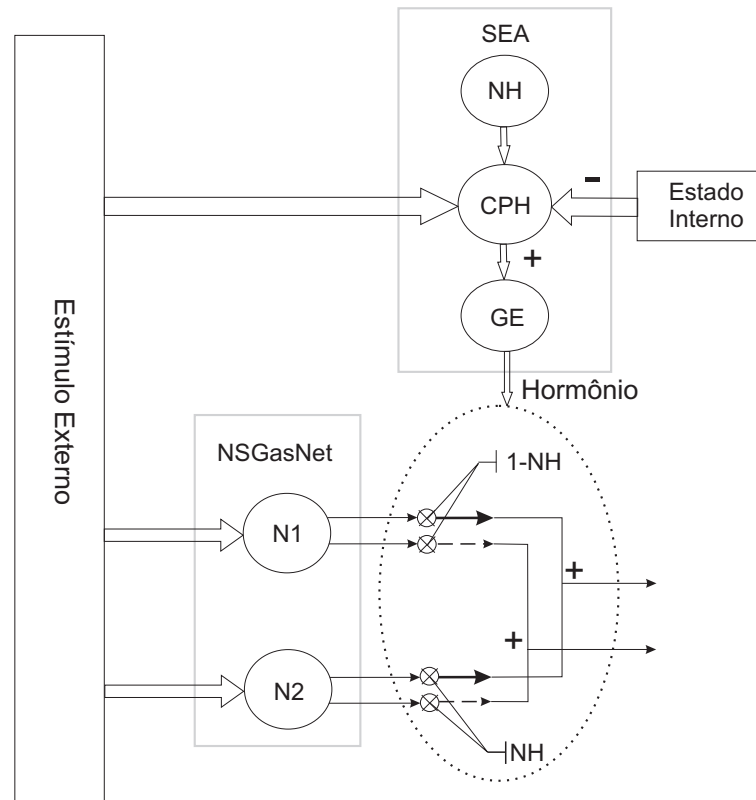


Fig. 4.2: Visão geral do Sistema Homeostático Artificial Evolutivo (inspirado em Vargas et al. (2005))

tornar o sistema mais robusto em ambientes diferentes daqueles empregados durante a evolução de todo o sistema.

O processo evolutivo englobará a evolução dos seguintes parâmetros que compõem as equações 4.1, 4.2 e 4.3: θ , α , ω e T . Os parâmetros β e λ não estão sob controle do processo evolutivo, pois são considerados propriedades particulares do sistema e do ambiente, respectivamente, e, portanto, devem assumir valores pré-determinados. Este fato será abordado novamente na seção 4.3.2, onde todos os parâmetros das equações serão reinterpretados de acordo com o experimento.

A Figura 4.2 provê uma visão geral do sistema e das interações dos módulos componentes do SHAE: SEA e duas NSGasNets (N1 e N2). A figura também ilustra as interações do SHAE com o ambiente (estímulo externo) e com o agente artificial (estado interno). As NSGasNets são previamente evoluídas, e de forma independente, para desempenhar dois comportamentos distintos. A saída das redes é, então, modulada pelo nível hormonal (NH , no intervalo $[0,1]$). O valor de NH multiplica a saída das RNAs, dando origem à coordenação dinâmica dos dois comportamentos. Em particular,

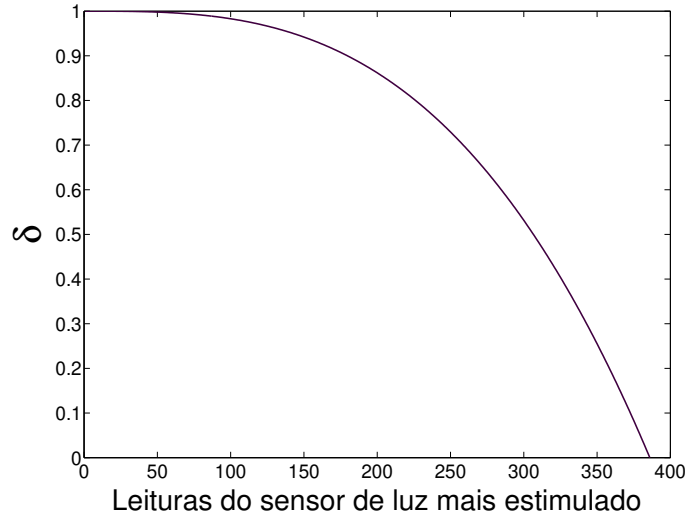


Fig. 4.3: Variação de δ em função das leituras do sensor de luz mais estimulado

para N1 o valor multiplicado é $(1 - NH)$ e para N2 o valor é (NH) . Na versão estendida, que incorpora receptores, o fator de modulação é multiplicado por δ (Equação 4.4). Os componentes do SEA também são apresentados, junto com suas conexões internas. O objetivo geral do SHAE é preservar a integridade do sistema frente a condições de operação variantes no tempo.

$$\delta = 1 - \frac{[\min(\text{leituraSensoresLuz}) - \text{MinR}]^3}{(\text{MaxR} - \text{MinR})^3} \quad (4.4)$$

em que $\min(\text{leituraSensoresLuz})$ corresponde ao sensor de luz mais estimulado (e que tem, devido à estrutura do robô, leituras numericamente baixas). Os parâmetros MinR e MaxR assumem os valores 65 e 450, respectivamente.

Nessa primeira tentativa de incluir receptores, os parâmetros MinR e MaxR da Equação 4.4 foram determinados empiricamente, sendo específicos para a tarefa escolhida. A Figura 4.3 mostra os valores de δ em função do estímulo externo (i.e. a fonte luminosa). É importante destacar que a variável que representa o receptor é bastante sensível à radiação luminosa e portanto atua regulando a resposta do agente à presença do hormônio. Pode-se entender essa sensibilidade à fonte de luz como se um hipotálamo fictício tivesse liberado um neurotransmissor ao perceber a presença de luz. Esse neurotransmissor ligaria-se então ao receptor, mudando a sensibilidade do agente ao nível hormonal, influenciado pela proximidade da fonte de luz.

Estabelecida a arquitetura do SHAE, pode-se questionar o porquê da não evolução das regras que determinam seu comportamento, mais especificamente da estrutura e operações algébricas das equações a diferenças apresentadas nas Eqs. 4.1, 4.2 e 4.3. Di Paolo (2005) argumenta que *“a natureza do acoplamento físico é sempre completamente determinada pelas leis da física. Uma célula não pode simplesmente mudar as leis de reação e difusão. O que é dado ao organismo é o controle paramétrico dessas leis através da influência na intensidade do acoplamento dinâmico”*. Seguindo essas idéias, o presente trabalho faz uma analogia entre essas leis da física, que ditam a natureza do acoplamento físico entre uma célula e seu ambiente externo, e as regras que governam o acoplamento artificial entre o SEA, as NSGasNet, o ambiente e o agente artificial (Equações 4.1, 4.2 e 4.3 e Figura 4.2). Neste sentido, o novo SHAE irá evoluir a composição paramétrica dessas regras, influenciando as relações dinâmicas das equações a diferenças, e não alterando as regras em si.

A seguir, a seção Procedimentos Experimentais descreverá os processos envolvidos na evolução do SHAE e no ambiente de simulação.

4.3.2 Procedimentos Experimentais

Um agente robótico móvel simulado, equipado com um medidor de bateria interna, tem que cumprir duas tarefas relacionadas, porém distintas: explorar o cenário evitando colisões e buscar a fonte de luz quando seu nível de bateria é baixo (a fonte de luz indica a localização do carregador de bateria). Este experimento foi utilizado por Vargas et al. (2005) para analisar a performance de um modelo de sistema homeostático artificial.

A evolução do sistema é dividida em duas etapas. Primeiramente, duas NSGasNets são evoluídas independentemente (uma NSGasNet evoluída para cada tarefa) empregando um algoritmo genético distribuído, descrito na próxima seção. Em seguida, o SEA, que deve atuar como módulo de coordenação e ser responsável pelo chaveamento de comportamentos entre as NSGasNets, é acoplado às redes neurais artificiais e evoluído.

O primeiro experimento verifica a performance do SHAE evoluído. Assim, os parâmetros referentes ao SEA não são alterados. O segundo e terceiro experimentos promovem perturbações nos parâmetros do SEA, objetivando analisar o desempenho do SHAE no processo de regulação homeostática. Espera-se que o sistema seja capaz de lidar dinamicamente com mudanças no ambiente e

perturbações sensoriais e físicas, ajustando a dinâmica de suas variáveis internas. A versão estendida do SHAE, que inclui receptores de hormônio, é comparada com a primeira num quarto experimento, utilizando um robô simulado e real.

Para os experimentos, será utilizado o simulador KIKS (Apêndice B), que reproduz bem o comportamento do mini-robô Khepera II (Apêndice A).

Algoritmo Genético

Cada rede a ser evoluída é formada por um número fixo e pré-determinado de neurônios, no caso deste trabalho, seis. Este número está relacionado com o número de entradas (sensores do robô) e de saídas (atuadores do robô) presentes na rede, e será detalhado na próxima subseção.

O modelo da NSGasNet é composto por seis variáveis associadas a cada nó mais um bias modulador ($Mbias_{ij}$) para cada nó, mais parâmetros dependentes da tarefa. Cada rede possui seis neurônios, portanto as redes N1 e N2 (Figura 4.2) terão seis $Mbias$ para cada nó.

Um neurônio pode se conectar eletricamente a mais dois neurônios. Observando a notação descrita no Capítulo 2, pp. 29, o genótipo de cada neurônio das redes NSGasNet é formado por 16 parâmetros:

$$\begin{aligned} < genotipo > :: < rec > < TE > < CE > < D_i^0 > < bias > < s > \\ < Mbias_{i1} > .. < Mbias_{ij} > < n1 > < n2 > < w1 > < w2 > \end{aligned}$$

em que

$< rec >$ determina se a conexão recorrente é excitatória, inibitória, ou inexistente;
 $< TE >$ determina as circunstâncias sob as quais o neurônio irá emitir gás: se sua atividade elétrica exceder um limiar predeterminado, se a concentração de gás 1 exceder um limiar predeterminado, se a concentração de gás 2 exceder um limiar predeterminado, ou se o neurônio não emite gás sob circunstância nenhuma;

- $\langle CE \rangle$ especifica qual gás o neurônio emite, gás 1 ou gás 2;
- i e j são os índices dos neurônios da rede (neste caso, $1 \leq (i, j) \leq 6$).
- $\langle D_i^0 \rangle$, $\langle bias \rangle$ e $\langle s \rangle$ são relativos às Equações 2.7, 2.9 e 2.3, respectivamente;
- $\langle Mbias_{ij} \rangle$ é o bias modulador do neurônio;
- $\langle n1 \rangle$ e $\langle n2 \rangle$ são os neurônios que possuem conexão com o dado neurônio;
- $\langle w1 \rangle$ e $\langle w2 \rangle$ os pesos associados a essas conexões, respectivamente.

No genótipo, cada parâmetro possui um valor no intervalo $[0, 99]$. A decodificação de genótipo para fenótipo adotada é a mesma utilizada na proposta original da GasNet (Husbands et al., 1998), conforme a Equação 2.10:

$$fen = \begin{cases} gen/99, & \text{se fen for contínuo} \\ gen \bmod N_{nom}, & \text{se fen for nominal} \end{cases} \quad (4.5)$$

em que

fen é o valor do fenótipo;

gen é o valor do genótipo;

N_{nom} é o número de valores fenotípicos possíveis;

mod é o operador que fornece o valor do resto da divisão.

Retomando o Capítulo 2, os parâmetros rec , TE , CE , D_i^0 , s , $n1$ e $n2$ apresentam fenótipo nominal e os demais parâmetros apresentam fenótipo contínuo.

O genótipo do SEA obedece às mesmas regras, porém todos os quatro parâmetros são contínuos: $\omega \in [50, 100]$, $\theta \in [0, 99]$, $\alpha \in [0, 99]$ e $T \in [1, 1000]$. Os parâmetros β e λ são previamente determinados, pois são considerados propriedades particulares do sistema e do ambiente (a seção “Evolução do Sistema Endócrino Artificial”, a seguir, retomará esse fato). Os valores-limite adotados para o fenótipo foram escolhidos de forma a manter a coesão física das equações. Por exemplo, a meia-vida do hormônio não pode ser 0, da mesma forma que $\omega = 0$ anula todos os efeitos hormonais.

A evolução das NSGasNets e do SEA empregam um algoritmo genético distribuído (Hillis, 1990; Collins & Jefferson, 1991), que é o algoritmo evolutivo originalmente adotado no trabalho de Husbands et al. (1998). Observe a Figura 4.4, que resume o processo de aplicação do algoritmo genético.

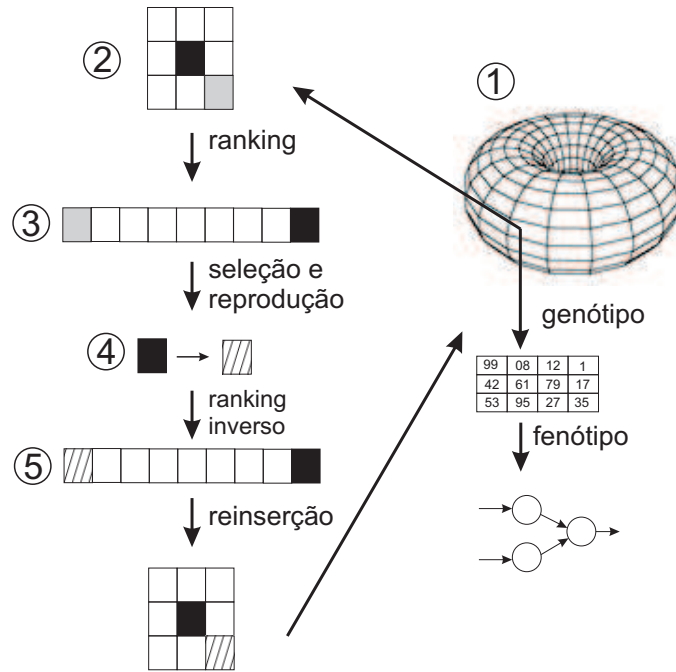


Fig. 4.4: Operação do algoritmo genético distribuído. Em cada aplicação, um conjunto de soluções é formado a partir do conjunto de toda a população. Os indivíduos escolhidos são ordenados, e os de maior *fitness* (cor preta) têm maior chance de reprodução. Por sua vez, os de menor *fitness* (cor cinza) têm maior chance de substituição. O novo indivíduo gerado é representado pelo quadrado hachurado.

A população é inicializada aleatoriamente com genótipos organizados em uma grade toroidal de dimensões 5x5 (Passo 1). Em um evento reprodutivo, um ponto aleatório na grade é escolhido, e um conjunto formado por ele e seus oito vizinhos é estabelecido (Passo 2). O conjunto formado pelos nove genótipos é ordenado de acordo com o valor do *fitness* de cada genótipo (Passo 3). Um único genótipo é, então, escolhido via método da roleta (Equação 4.6) e os operadores de mutação são aplicados em 10% do genótipo (Passo 4). O primeiro operador é válido apenas para variáveis contínuas. Ele perturba o valor de cada locus escolhido por um valor no intervalo $[-10, 10]$. Para o segundo operador de mutação, um locus do genótipo determinado aleatoriamente é alterado para um valor escolhido no intervalo $[0, 99]$.

$$P(i) = \frac{i}{\sum_{j=1}^{j=N} j} = \frac{2i}{N(N+1)} \quad (4.6)$$

em que $P(i)$ é a probabilidade do elemento i do vetor de tamanho N ser escolhido, considerando que este vetor está ordenado de forma crescente (0 a $N - 1$) com relação ao valor de *fitness*.

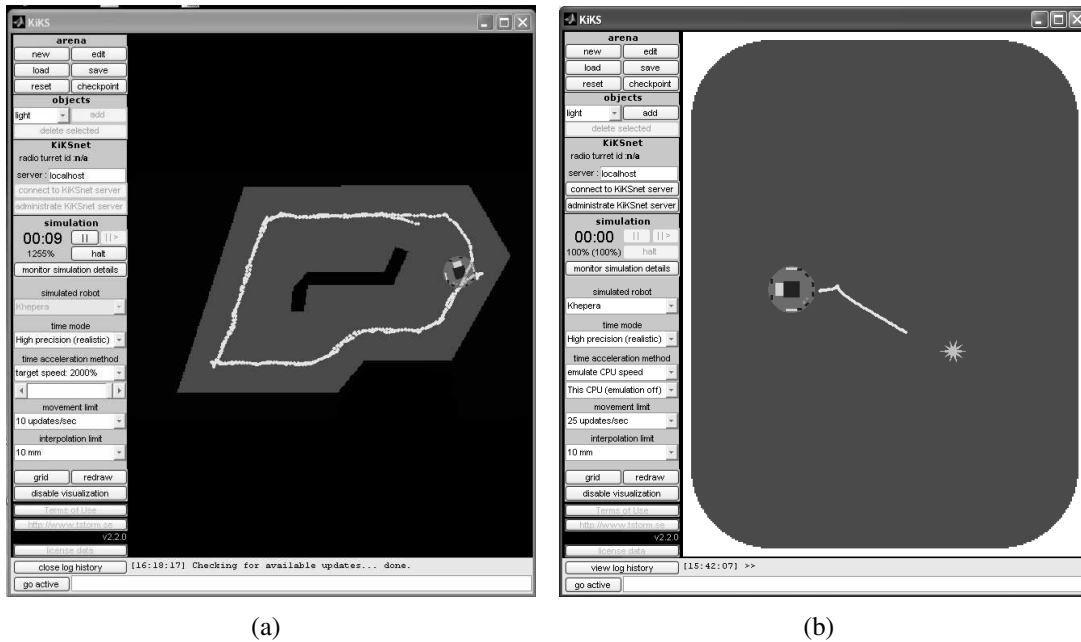


Fig. 4.5: Cenário de treinamento para o comportamento de desvio de obstáculos (a); e para o comportamento de fototaxia (b)

Neste método da roleta (Equação 4.6), o indivíduo de menor *fitness* tem chance zero de ser escolhido como pai. Um novo indivíduo é produzido, através da aplicação de operadores de mutação em uma cópia do pai, e avaliado. A seguir, o novo indivíduo é reinserido no conjunto reprodutivo em uma posição ocupada por um indivíduo com probabilidade de escolha determinada também pela Equação 4.6, desta vez considerando o inverso do *fitness* dos indivíduos (Passo 5). Desta forma, o indivíduo de maior *fitness* tem chance zero de ser substituído pelo filho. Não é utilizado *crossover*. Uma geração é definida como 25 eventos reprodutivos, e o algoritmo evolutivo é aplicado por 50 gerações no máximo. O critério de *fitness* é específico para cada tarefa.

O valor dos parâmetros adotados para o algoritmo genético foram definidos empiricamente. Entretanto, constatou-se que a sensibilidade à variação dos parâmetros é pequena, obtendo-se resultados equivalentes para diferentes tamanhos de grade e taxas de mutação.

Evolução de comportamento: desvio de obstáculos

Para a navegação visando exploração e desvio de obstáculos, a rede N1 (Figura 4.2) possui 4 entradas correspondentes aos sensores de distância do robô. Observe, no Apêndice A, que o robô

possui 8 sensores de distância, dois mais à esquerda, dois frontais, dois mais à direita e dois traseiros. Porém, para colaborar com a redução da complexidade do problema, as quatro entradas da rede são formadas pelo sensor mais estimulado à esquerda, direita, frente e trás. Todas as leituras, que variam de 0 (ausência de obstáculo) a 1023 (obstáculo muito próximo), são normalizadas no intervalo $[-1, +1]$. Dois neurônios diferentes foram considerados como saída, totalizando 6 neurônios na rede. Os neurônios de saída correspondem aos neurônios motores, responsáveis por conduzir o robô. Uma saída de valor 1 corresponde à máxima velocidade de uma roda para frente, e uma saída de valor -1 corresponde à máxima velocidade de uma roda para trás. A diferença nos valores e sentidos de rotação irão determinar a direção de movimento do robô.

A função de *fitness* (Equação 4.7) e o cenário de treinamento (Figura 4.5(a)) foram inspirados no trabalho de Nolfi & Floreano (2000):

$$\phi = V(1 - \sqrt{\Delta v})(1 - i), \quad (4.7)$$

em que V é a soma das velocidades de rotação das rodas (estimulando altas velocidades), normalizada entre 0 e 1, Δv é o valor absoluto da diferença algébrica entre os valores das velocidades das rodas (estimulando movimento em linha reta), normalizado entre 0 e 1, e i é o valor normalizado do sensor de distância mais excitado (estimulando desvio de obstáculos). Considera-se uma etapa como sendo 2.000 iterações do algoritmo de controle. Ao final de cada etapa, o robô é aleatoriamente posicionado no cenário.

Evolução de comportamento: fototaxia

A estrutura da rede N2 (Figura 4.2) para o comportamento de fototaxia é similar à estrutura da rede para desvio de obstáculos, sendo que os sensores de distância foram substituídos pelos sensores de luminosidade. Todas as leituras, que variam de 50 (máxima luminosidade) a 470 (ausência de luz), são normalizadas no intervalo $[-1, +1]$. O cenário de treinamento é composto por uma arena retangular com bordas arredondadas, onde o robô tem sua posição inicial fixa ao início de cada etapa. A luz é aleatoriamente posicionada em um local relativamente distante do robô, embora capaz de estimular seus sensores de qualquer ponto do cenário. Uma etapa consiste do robô explorando o cenário em busca da luz. Sempre que a luz é “capturada”, o robô é posicionado no ponto inicial e a luz

é aleatoriamente reposicionada. Cada etapa corresponde a 2.000 iterações do algoritmo de controle. A função de *fitness* é apresentada na Equação 4.8. O parâmetro i , referente à ativação sensorial, é minimizado quando o robô encontra-se próximo à luz (devido à estrutura sensorial do robô, descrita no Apêndice A). Note que a função é similar à função anterior, entretanto, o componente que estimula movimento frontal é omitido. Isso significa que o robô pode permanecer girando ao redor da luz e ainda assim obter um *fitness* alto. O cenário de treinamento está ilustrado na Figura 4.5(b).

$$\phi = V(1 - i). \quad (4.8)$$

Evolução do Sistema Endócrino Artificial

Recorde, da seção 4.2, que o SEA possui 6 parâmetros de controle a serem definidos: λ , ω , β , θ , α e T (Equações 4.1–4.3). Os parâmetros λ e ω referem-se aos limiares de intensidade de luz e nível hormonal, respectivamente, abaixo dos quais o estado interno (*EI*) cresce a uma taxa influenciada por β . θ refere-se ao limiar do estado interno, acima do qual a produção hormonal inicia-se a uma taxa influenciada por α . T é a meia-vida do hormônio.

No modelo aqui apresentado, o estado interno (*EI*) do agente artificial (Equação 4.2) corresponde ao inverso da leitura do medidor de bateria. Isso implica que quanto menor o nível de bateria, maior o valor de *EI*. Como a bateria está sempre se descarregando (considerando-se que o robô esteja ligado), e essa descarga é uma propriedade construtiva do agente, é justificável que β deva ter um valor predeterminado associado a isso. Se β participasse do processo evolutivo, uma solução fisicamente incompatível poderia ser obtida: o robô poderia funcionar e se locomover sem descarregar sua bateria, ou descarregando-a a taxas muito baixas, comprometendo a análise do comportamento observado. Similarmente, é justificável que a menor intensidade de luz acima da qual o robô pode recarregar-se, λ , seja também predefinida, por ser uma propriedade inerente do robô e de seu sistema de recarga. Do contrário, uma solução fisicamente incompatível poderia ser obtida: o nível de luz acima do qual o robô pode recarregar-se é nulo ou muito baixo, isto é, o robô pode recarregar-se em qualquer ponto do cenário, estando próximo ou não da fonte de luz, inviabilizando a realização do experimento.

Desta forma, o genótipo consiste de 4 parâmetros: ω , θ , α e T (Equações 4.1–4.3). Uma etapa é considerada como sendo 800 iterações sensóri-motoras. O experimento inicia-se com o robô explo-

rando a arena, controlado pela rede de desvio de obstáculos. Se o estado interno crescer acima de 90, numa escala de 0 a 100, o robô é considerado morto. Para obter um bom desempenho, o robô deve ser capaz de chavear eficientemente entre o comportamento de exploração e o comportamento de fototaxia. Esse chaveamento deve ocorrer devido à produção do hormônio relacionado à diminuição do nível de bateria. Após a recarga da bateria (associada com a proximidade da luz), e consequentemente a diminuição do nível hormonal relacionado, o robô deve retornar a seu comportamento original de exploração. A Equação 4.9 mostra a função de *fitness* adotada para essa tarefa:

$$\phi = V(1 - i)t/M, \quad (4.9)$$

em que V é o valor absoluto da soma das velocidades de rotação das rodas (estimulando movimento em linha reta), i é o valor normalizado do sensor de distância de maior ativação (estimulando desvio de obstáculos), t é o número de iterações em que o robô permanece vivo e M é o número máximo de iterações por etapa. Portanto, uma boa performance consiste em ajustar os limiares de produção hormonal e taxa de crescimento objetivando máxima exploração do cenário e levando em conta a necessidade de recarga. Devido às características do ambiente utilizado, para alcançar um bom desempenho o robô não pode permanecer o tempo todo próximo à fonte de luz, dado que essa fonte localiza-se próxima às paredes da arena.

4.3.3 Experimentos

Nesta seção, são apresentados os experimentos que verificam a performance do SHAE evoluído. No primeiro experimento, os parâmetros referentes ao SEA não são alterados. No segundo e terceiro experimentos são feitas perturbações nos parâmetros do SEA, e em sequência analisa-se o processo de regulação homeostática.

Experimento 1 - Análise da performance do SHAE sob condições normais

O primeiro experimento refere-se à evolução do SHAE. A Figura 4.6 ilustra o valor de *fitness* durante o processo evolutivo, para as NSGasNet ((a) e (b)) e para o sistema endócrino artificial que coordena as NSGasNets evoluídas (c). As configurações finais das redes estão ilustradas na Figura

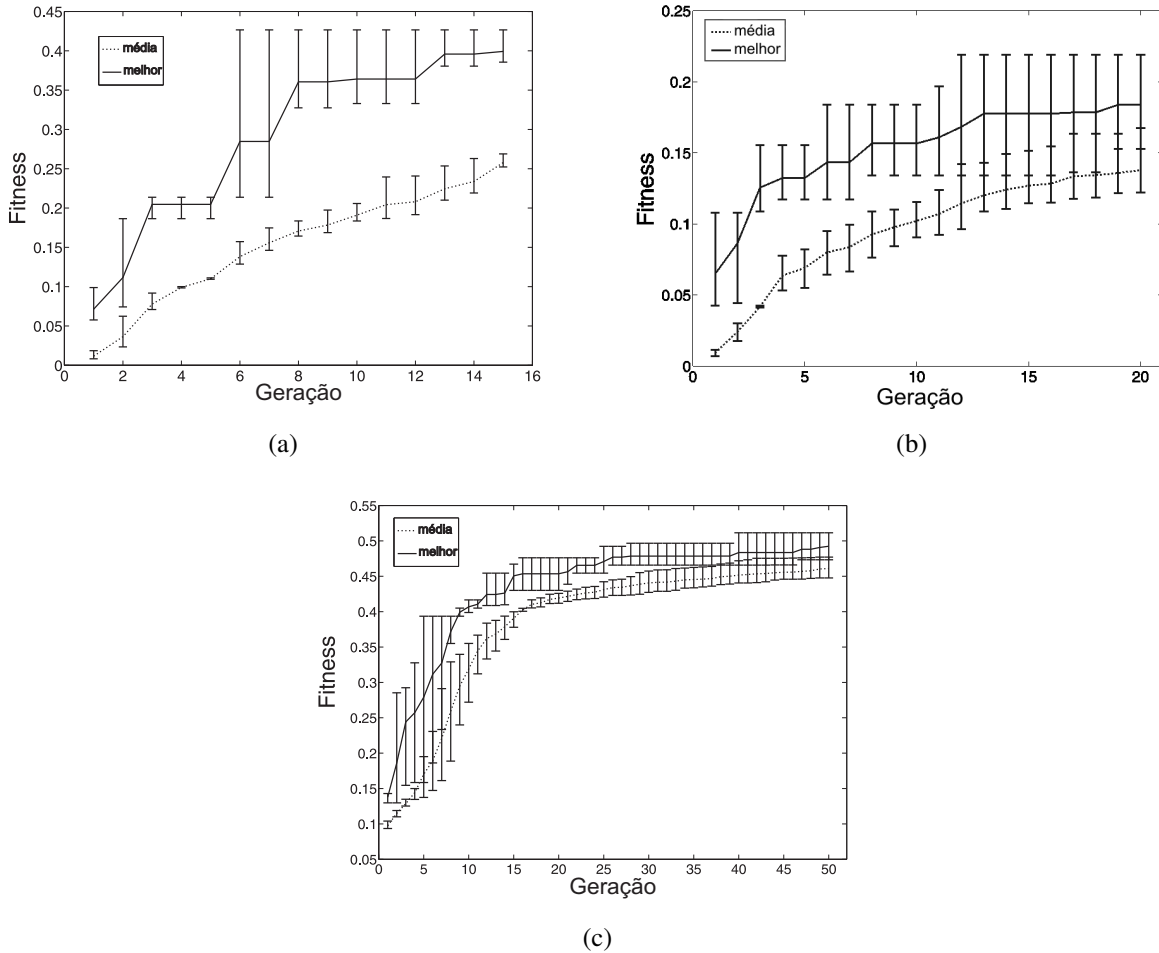


Fig. 4.6: Evolução do *fitness*: desvio de obstáculos (a), fototaxia (b) e SEA (c)

4.7: a rede evoluída para desvio de obstáculos NSGasNet-N1 (a) e a rede evoluída para fototaxia NSGasNet-N2 (b). As linhas tracejadas referem-se a conexões elétricas de supressão, circunferências tracejadas referem-se à emissão do gás tipo 2, e linhas cheias referem-se a conexões elétricas de estímulo. Neste caso específico, não houve neurônios que emitissem gás tipo 1.

As Figuras 4.8(a) e 4.8(b) mostram a trajetória do melhor indivíduo e a dinâmica das variáveis durante uma etapa, respectivamente. O fenótipo do melhor indivíduo do SEA é: $\alpha = 0,0099$; $T = 11,1$; $\omega = 50,5$; e $\theta = 52,0$. Os parâmetros $\beta = 0,01$ e $\lambda = 103$ foram definidos a priori e empiricamente. Perceba que, quando o nível hormonal supera ω , o robô pára de explorar o cenário e inicia a busca pela fonte de luz (fato ilustrado pelos picos inferiores nas leituras de nível de luz na Figura 4.8(b)). Recorde que, devido à estrutura sensorial do robô, picos inferiores nas leituras

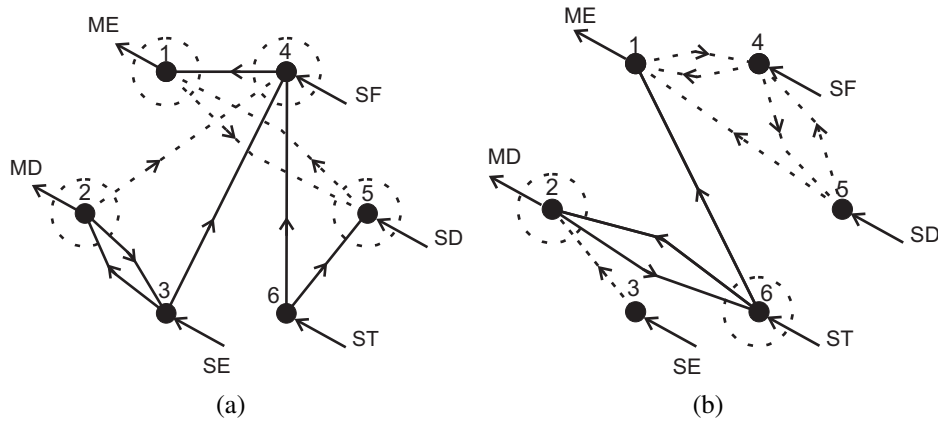


Fig. 4.7: Redes evoluídas: desvio de obstáculos N1 (a) e fototaxia N2 (b). Estão representados os neurônios de saída relativos ao motor esquerdo (ME) e motor direito (MD), e também os neurônios relativos aos sensores de entrada, sensor esquerdo (SE), sensor direito (SD), sensor frontal (SF) e sensor traseiro (ST).

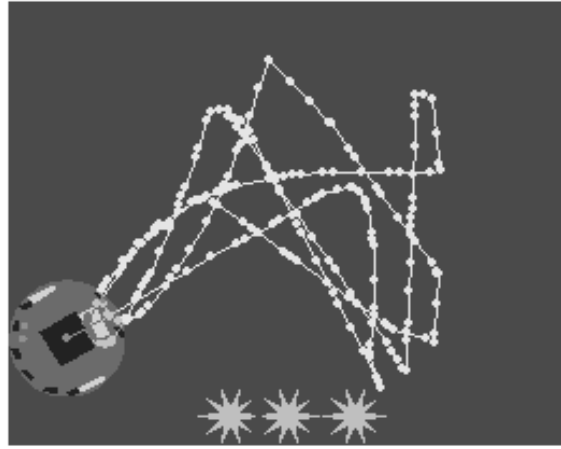
de luminosidade indicam máxima excitação luminosa, maximizando portanto o valor do estímulo externo EE , referenciado na Equação 4.1. Isso confirma a influência do nível hormonal sobre o comportamento autônomo do robô. Também é possível observar que o robô percorre a arena em velocidade máxima (observe a curva de velocidade, normalizada entre 0 e 100), apenas ajustando sua velocidade ao evitar colisões ou no chaveamento para o comportamento de fototaxia.

A Figura 4.9(a) destaca um trecho do experimento. Note que o estado interno (EI , curva vermelha) cresce até o momento em que supera o valor de θ (Ponto A). Inicia-se então a produção hormonal (curva azul), que é responsável por ponderar as saídas das redes neurais, determinando o comportamento do robô. Recorde que a saída da rede de desvio de obstáculo é multiplicada por $(1 - NH)$, enquanto que a rede de fototaxia é multiplicada por NH . Portanto, quando o nível hormonal se aproxima do máximo (Ponto B), o comportamento preponderante do robô é a busca da fonte de luz, indicada pelo decrescimento da leitura sensorial de luminosidade. Quando essa leitura for superior ao parâmetro λ e o nível hormonal for superior a ω , o estado interno valerá 0 (Ponto C)(Equação 4.2).

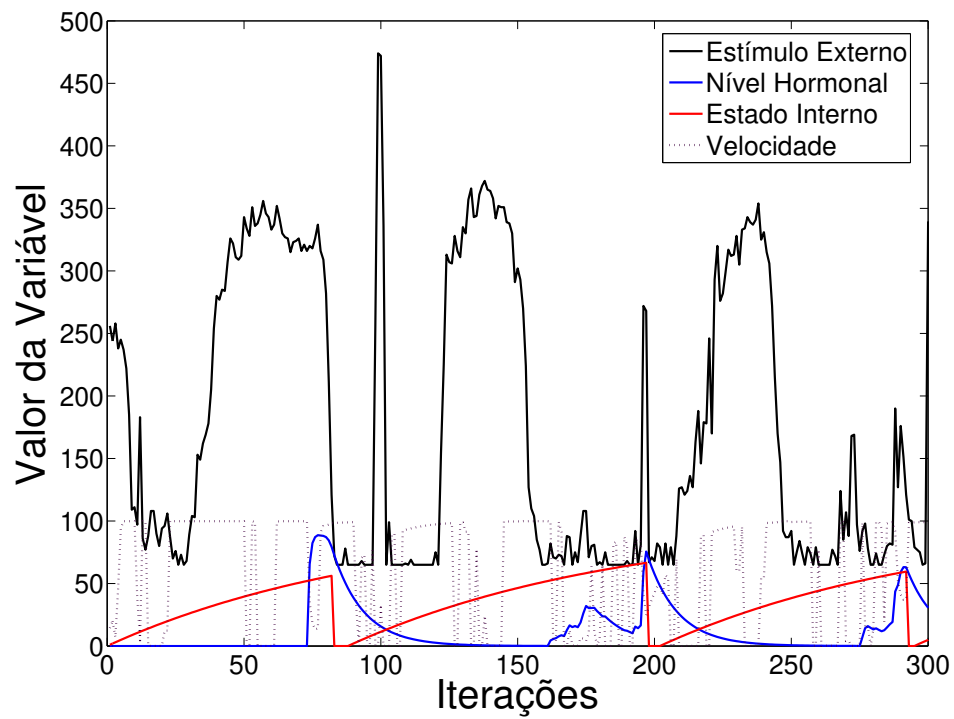
A Figura 4.9(b) evidencia um fato curioso sobre a natureza das equações do sistema endócrino artificial. Lembre-se que o valor do estado interno é o fator determinante para o início da produção hormonal (Equação 4.1). De fato, em torno da iteração 160 o sistema inicia a produção hormonal. Porém, neste caso, o robô encontra-se involuntariamente próximo à fonte luminosa (Ponto D). De acordo com a Equação 4.1, a produção é influenciada pelo estímulo externo e pelo nível hormonal atual. Como o estímulo externo é próximo do máximo, a produção hormonal é bem reduzida. Como

o hormônio é atualizado constantemente de acordo com a Equação 4.3, o decrescimento do nível hormonal em torno da iteração 180 é assim explicado. É relevante notar que o estado interno continua a crescer nessas etapas, pois o nível hormonal não supera ω . Apenas quando o estímulo externo e o nível hormonal superarem determinados níveis é que ocorre o decrescimento do estado interno. Portanto, nota-se que o robô continua a explorar o cenário até que se distancia da fonte de luz (Ponto E), o que, por sua vez, permite a produção hormonal mais intensa, a qual conduz ao chaveamento de comportamentos.

Esse é um ponto importante para futuras extensões. Uma alternativa seria rever as equações do SEA, especialmente a Equação 4.1. Observando esta equação, nota-se que o estado interno influencia apenas na produção ou não do hormônio, mas não em sua quantidade. A produção hormonal poderia, de alguma forma, influenciar mais diretamente a quantidade de hormônio produzida, independentemente do valor do estímulo externo. Como o hormônio é o responsável pela transição e chaveamento dos comportamentos do agente, é importante que sua produção possa ter mais influência do estado interno. Nessa mesma linha, o modo como o valor do estímulo externo influencia a produção de hormônio pode ser revisto, possibilitando a produção hormonal mesmo com estímulo externo máximo.

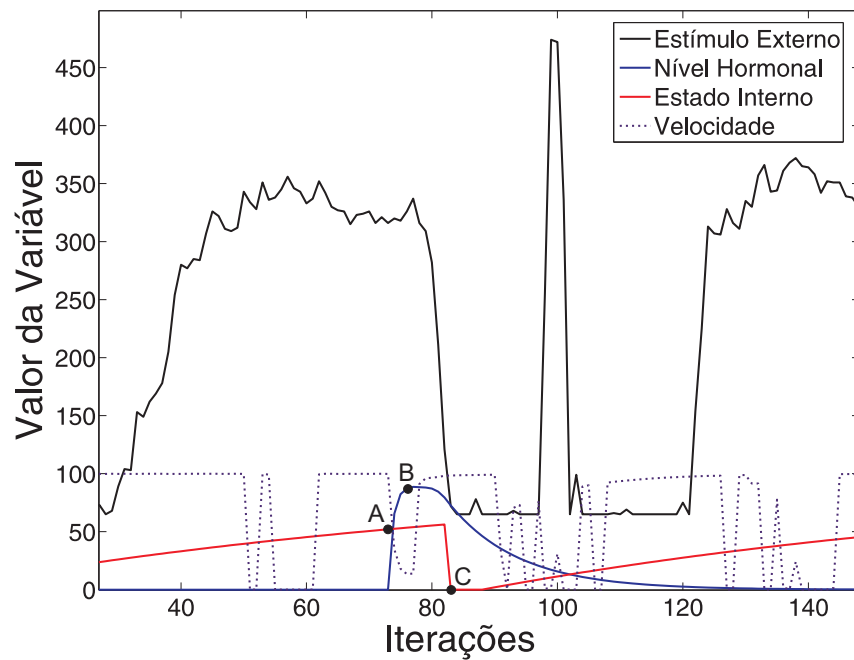


(a)

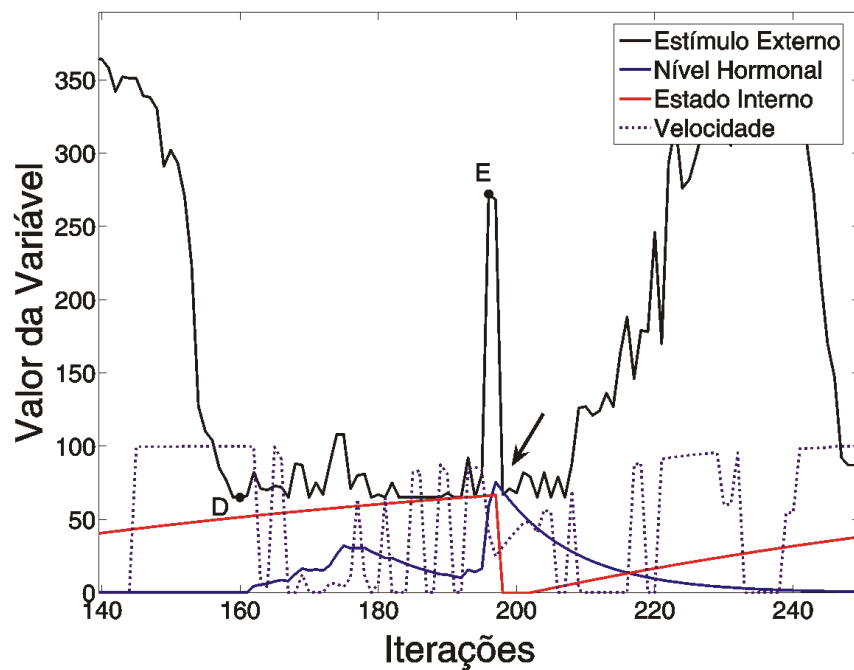


(b)

Fig. 4.8: Evolução do Sistema Homeostático Artificial (SHAE). Trajetória do melhor indivíduo evoluído após 300 iterações (a); e dinâmica das variáveis para $\beta = 0,01$ (b).



(a) Ampliação da Figura 4.8(b) entre as iterações 30 e 150.

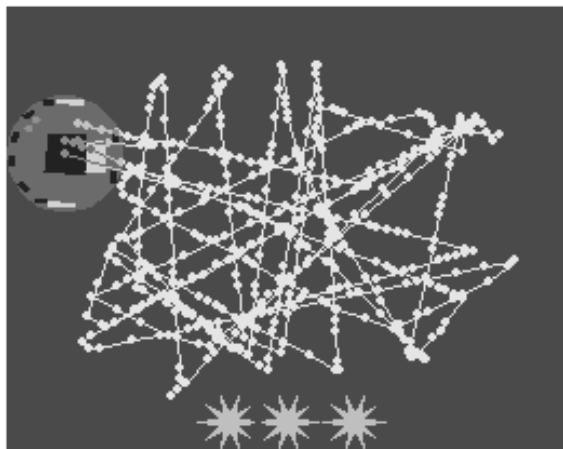
(b) Ampliação da Figura 4.8(b) entre as iterações 140 e 250. A seta indica o momento em que o EE e o NH superam determinados níveis e causam o decrescimento do EI , ou seja, $EE < 103$ e $NH > 50,5$.Fig. 4.9: Destaque da dinâmica das variáveis para $\beta = 0,01$

Experimento 2 - Análise da performance do SHAE sob perturbações persistentes

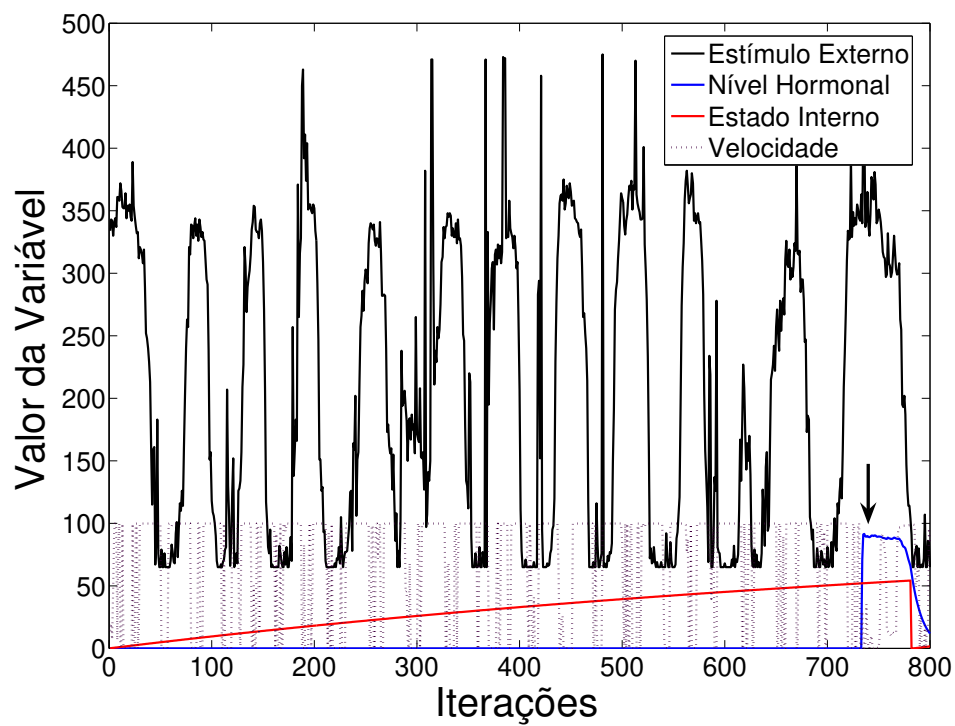
Este experimento visa submeter o SHAE a perturbações para analisar sua performance ao lidar com situações não vivenciadas em fase de evolução. A tarefa consiste em alterar o valor do parâmetro β após o término da etapa evolutiva, simulando uma descarga de bateria mais rápida ou mais lenta. Durante a evolução β tem um valor fixo, mais especificamente $\beta = 0,01$. As Figuras 4.10(a) e 4.10(b) ilustram os resultados para $\beta = 0,001$, isto é, uma descarga de bateria mais lenta, e as Figuras 4.11(a) e 4.11(b) apresentam os resultados para $\beta = 0,02$, simulando uma descarga de bateria mais rápida.

Nota-se que, no primeiro caso, o estado interno do robô cresce lento o suficiente para permitir que o robô explore toda a arena antes de ter necessidade de recarregar (em torno da iteração 800). Entretanto, no segundo caso, quando a taxa de descarga aumenta, o robô não é capaz de explorar a arena ao todo, ele deve chavear seu comportamento e retornar para recarga. O robô acaba desenvolvendo a capacidade de temporizar seu retorno à luz, evitando alcançar lugares distantes do carregador de bateria e, eventualmente, “morrer”. Com um β menor, o robô apenas necessita ajustar sua direção (e conseqüentemente reduzir sua velocidade) ao confrontar uma parede. Com valores maiores de β , o robô necessita periodicamente virar e buscar a luz. Nota-se ainda que a leitura dos sensores de luminosidade indica proximidade da fonte luminosa em grande parte do tempo (picos inferiores da curva de estímulo externo).

A Figura 4.12 ilustra uma etapa do experimento na qual o robô tem seu comportamento determinado pela rede de fototaxia. O nível hormonal permanece alto até o robô localizar e iniciar sua aproximação à fonte de luz. Isto, aliado à meia-vida do hormônio, faz com que sua produção decresça, conduzindo gradualmente à predominância de comportamento para desvio de obstáculos e exploração.

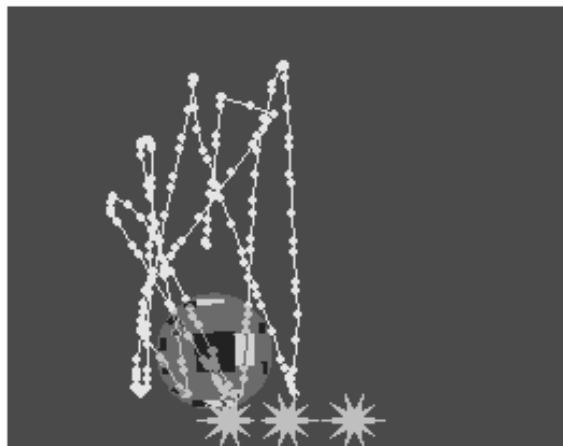


(a)

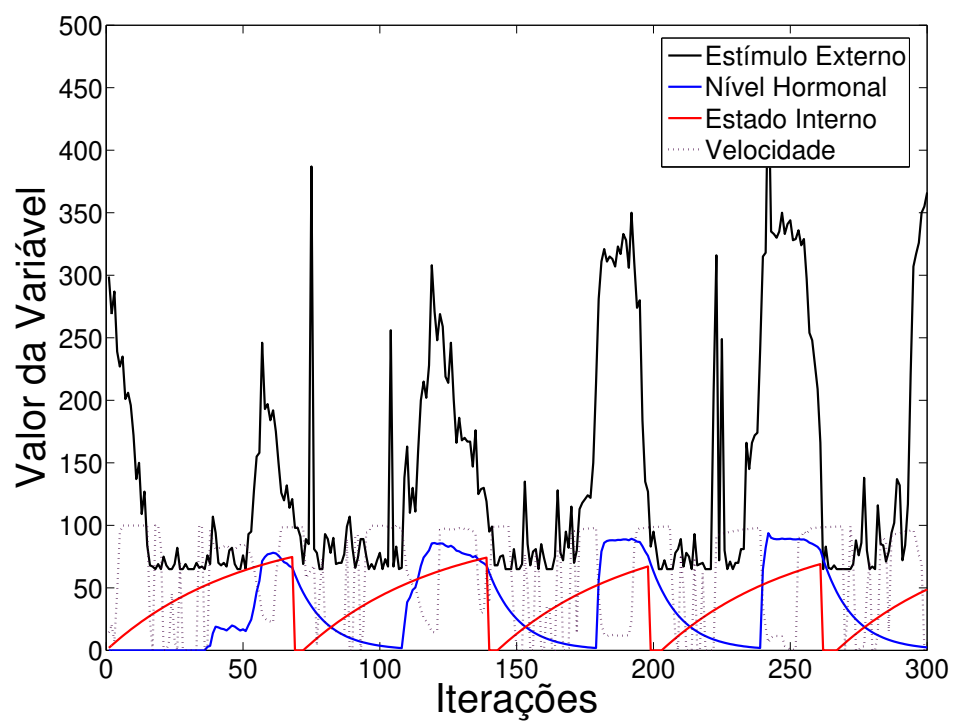


(b)

Fig. 4.10: Trajetória e dinâmica das variáveis para $\beta = 0,001$. A seta indica o momento em que o *EI* supera $\theta = 52,0$, e o robô começa a procurar a fonte luminosa (*EE*). Em muitos outros momentos houve aproximação da fonte luminosa pela simples natureza exploratória da trajetória do robô, sem a intenção de realizar recarga de bateria.



(a)



(b)

Fig. 4.11: Trajetória e dinâmica das variáveis para $\beta = 0,02$. Todo ciclo de alta do nível hormonal (NH) é seguido por aproximação da fonte luminosa (EE).

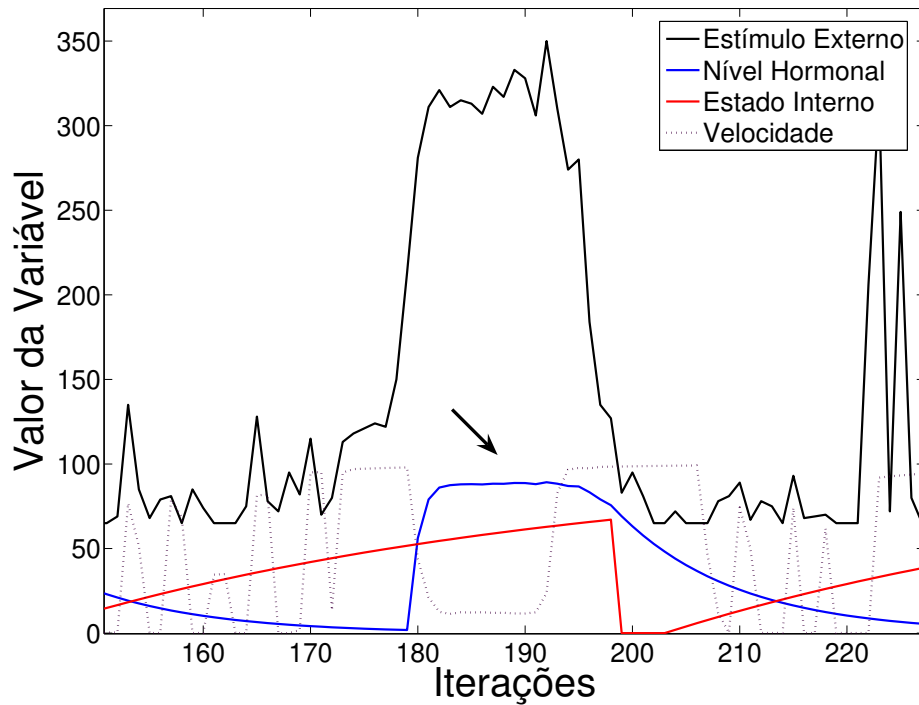
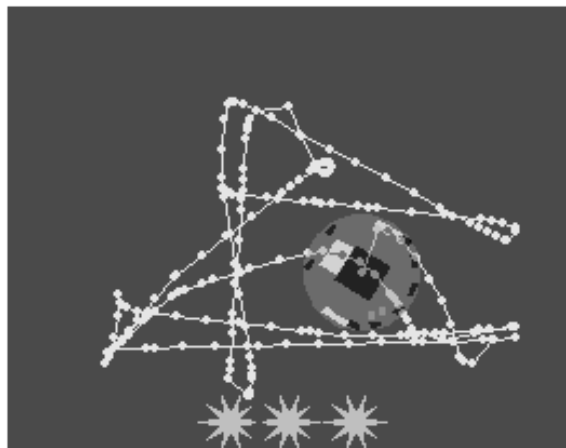


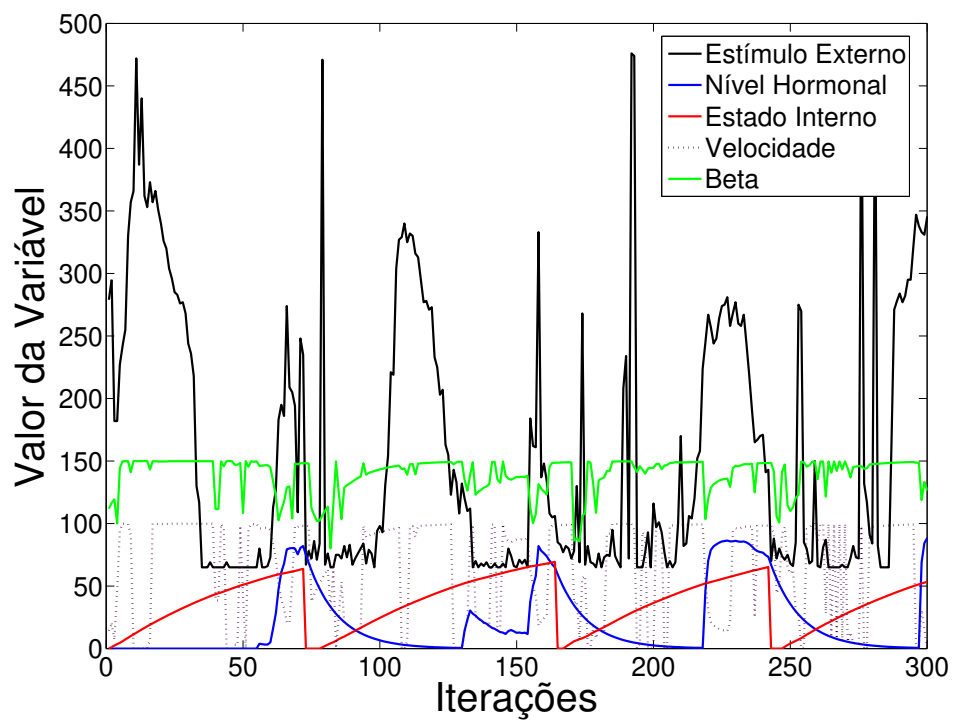
Fig. 4.12: Destaque da dinâmica das variáveis para $\beta = 0,02$. A seta indica o momento em que o sistema é controlado com predominância da rede de fototaxia.

Experimento 3 - Análise da performance do SHAE sob perturbações variáveis

Neste experimento, o parâmetro β varia durante a trajetória do robô. A velocidade de rotação absoluta das rodas do robô é relacionada a um maior consumo de energia. Logo, quanto maior a rotação, maior deve ser o valor de β , que pode variar linearmente entre 0,0015 e 0,015. As Figuras 4.13(a) e 4.13(b) ilustram a trajetória e a dinâmica das variáveis do robô, respectivamente. Observa-se que o robô consegue lidar com essa perturbação variável e é capaz de navegar e explorar o cenário com sucesso, buscando a luz quando a recarga é necessária. Como o comportamento dominante é a exploração e desvio de obstáculos, e a rede neural N1 foi evoluída de forma a maximizar a velocidade do robô, o valor de β (curva verde) é quase sempre alto, indicando altas velocidades. Quando o robô progressivamente altera seu comportamento para fototaxia (indicado pelo alto nível hormonal), o ajuste de direção faz com que a velocidade e, conseqüentemente β , sejam alterados.



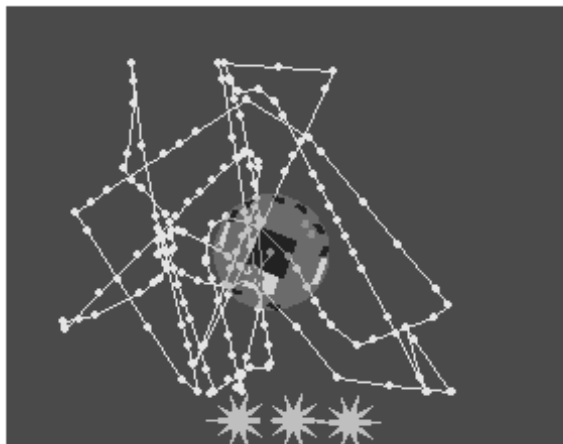
(a)



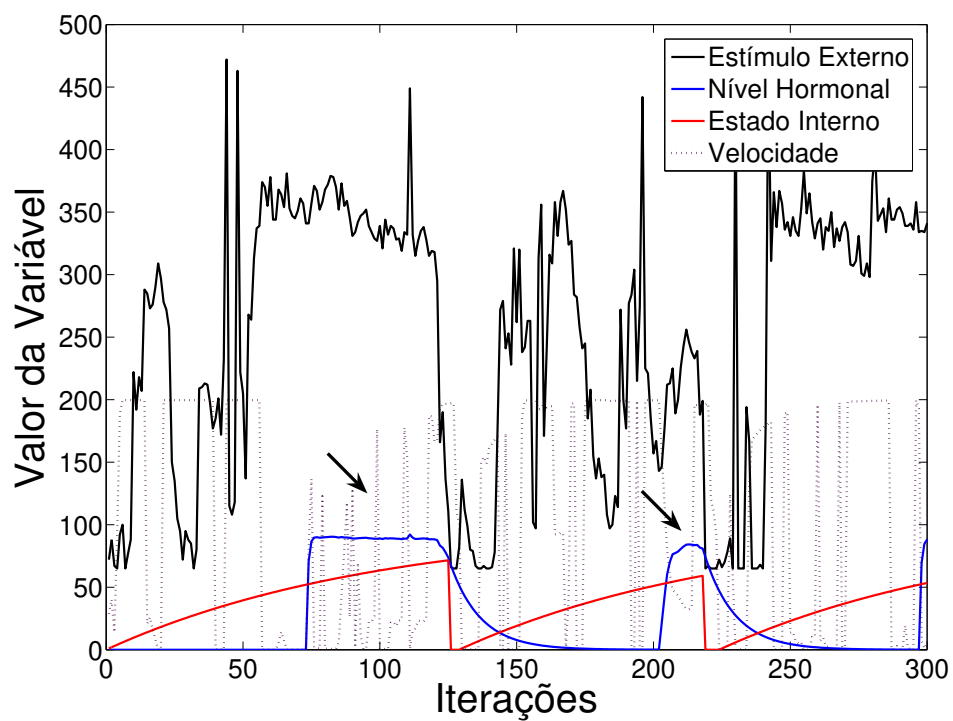
(b)

Fig. 4.13: Trajetória (a) e dinâmica das variáveis (b) para $0,005 \leq \beta \leq 0,015$

O próximo experimento simula uma perturbação na velocidade do robô: os comandos de velocidade são multiplicados por 2 antes de chegarem às rodas, mas essa informação não está diretamente disponível ao robô ou ao sistema de controle. O sistema deve adaptar-se a esta perturbação para obter boa performance, ou seja, deve adaptar-se de forma que o robô possa explorar o cenário sem colisão e direcionar-se à fonte de luz quando necessário. As Figuras 4.14(a) e 4.14(b) ilustram os resultados dos experimentos. Similarmente às perturbações anteriores, o sistema foi capaz de obter boa performance. As setas indicam os momentos em que o comportamento predominante é o de fototaxia. Próximo à iteração 100, o robô encontra-se a uma distância maior da fonte luminosa (fato indicado pela alta leitura do sensor de luminosidade). Já em torno da iteração 200, o robô encontra-se mais próximo à fonte luminosa e, portanto, o tempo em que permanece controlado pela rede de fototaxia é menor (fato indicado pela menor duração do nível hormonal alto).



(a)



(b)

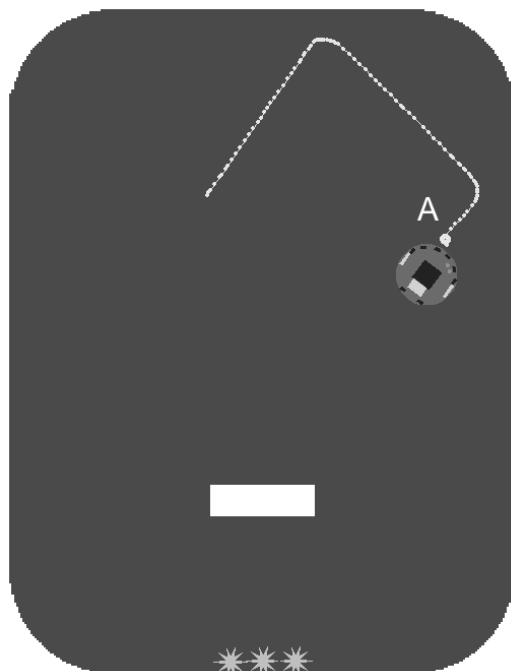
Fig. 4.14: Trajetória (a) e dinâmica das variáveis (b) para perturbação de velocidade (a velocidade dobra). As setas indicam os momentos em que o comportamento predominante é o de fototaxia.

Experimento 4 - Comparando o SHAE com e sem receptores de hormônio

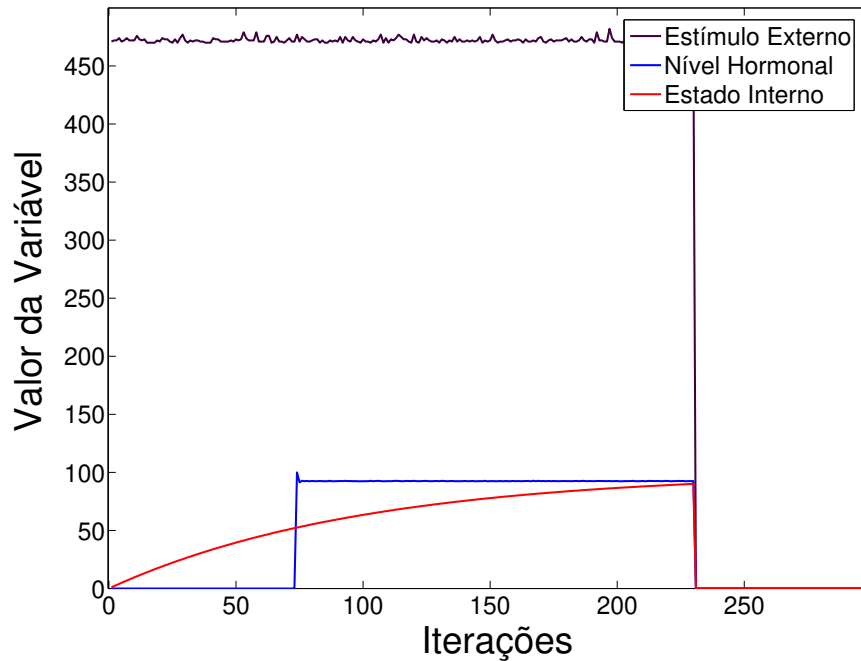
Os experimentos nessa seção procuram fornecer uma melhor compreensão da adição de receptores ao SHAE. Esse novo sistema, com receptor, será chamado SHAE-R. Para comparar a performance dos dois sistemas, um robô (chamado Robô 1) controlado pelo SHAE e outro (chamado Robô 2) controlado pelo SHAE-R são posicionados em uma arena grande, diferente das arenas utilizadas para evoluir todo o sistema. Eles serão analisados separadamente por 400 iterações cada ou até o estado interno do robô superar um limiar, simbolizando sua “morte”.

A Figura 4.15(a) mostra a trajetória do Robô 1 e a Figura 4.15(b) apresenta o comportamento das variáveis do SHAE ao longo das iterações. Note que o Robô 1, controlado apenas pelo SHAE, inicia a exploração da arena, e por volta da iteração de número 70 ocorre a produção de hormônio, indicando a necessidade de busca pela luz. O comportamento predominante é, portanto, fototaxia. Como os sensores do robô não são capazes de detectar a fonte de luz, que simboliza o carregador de bateria, o robô permanece girando sobre o próprio eixo (ponto A na Figura 4.15(a)), devido à característica da rede responsável pelo comportamento de fototaxia. Com isso, seu estado interno cresce acima do limite, e o robô não é capaz de recarregar sua bateria, “morrendo” próximo à iteração 240. É importante notar que o robô permanece um longo tempo sob influência do hormônio, apenas girando, sem conseguir explorar a arena em busca da luz.

Observe que o mesmo não acontece com o robô na Figura 4.16(a). A trajetória mostra que ele é capaz de chavear os comportamentos levando em conta não apenas o nível hormonal, mas também a proximidade da fonte luminosa. Esse fato pode ser verificado na Figura 4.16(b), que exhibe as variáveis do SHAE-R ao longo das iterações. Note que, na primeira vez que o nível hormonal cresce, o robô está próximo à fonte luminosa e portanto o receptor desempenha um papel importante, permitindo o chaveamento entre comportamentos. Entretanto, na segunda vez que o nível hormonal cresce, o robô está longe da fonte luminosa, logo a presença do receptor é vital para que o robô continue explorando a arena em busca da luz, assegurando o controle homeostático.



(a)



(b)

Fig. 4.15: Trajetória (a) e dinâmica das variáveis (b) para SHAE sem receptor. O ponto A indica o momento em que o robô “morre”, após permanecer girando sobre o próprio eixo (comportamento da rede de fototaxia na ausência de luz) por muitas iterações.

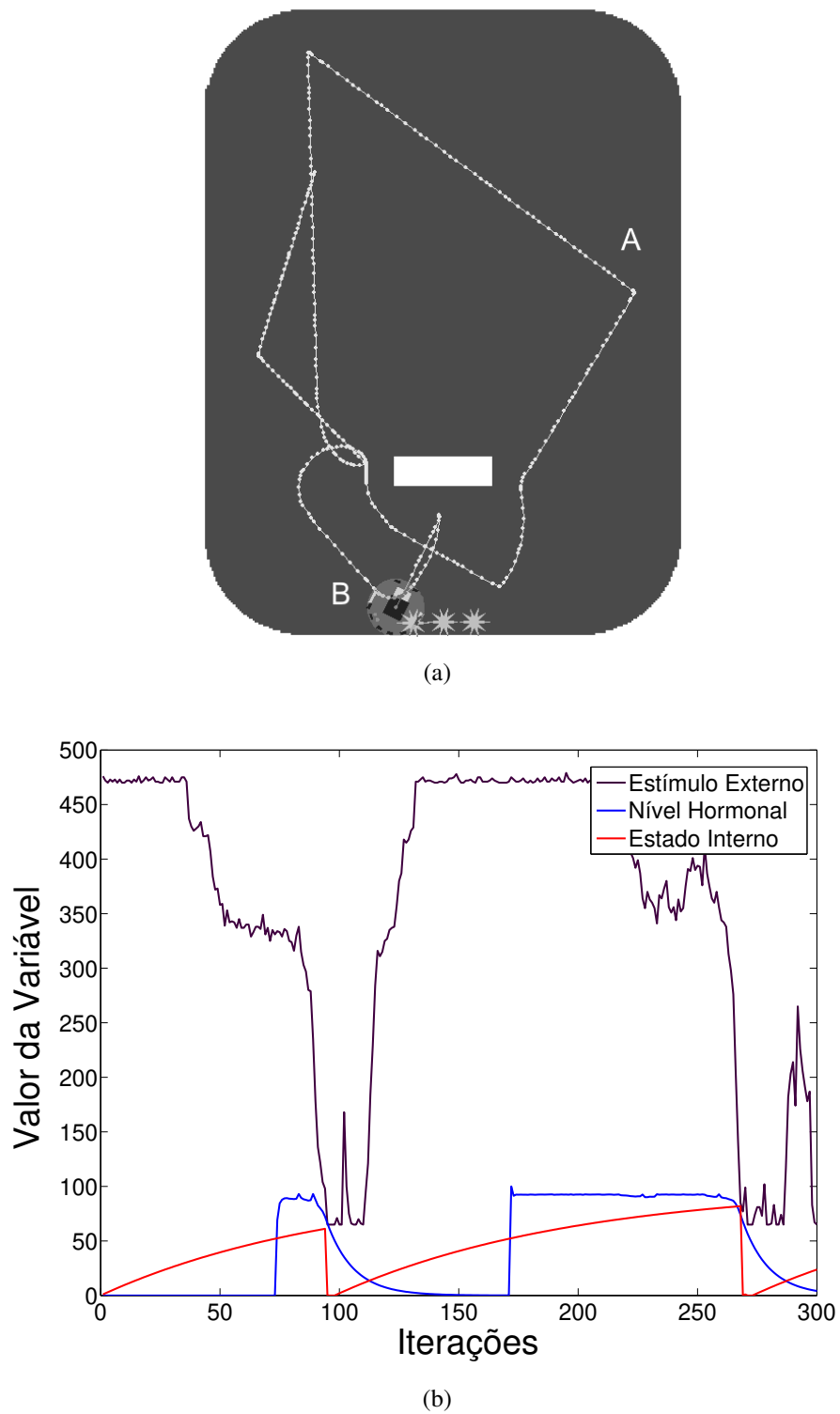
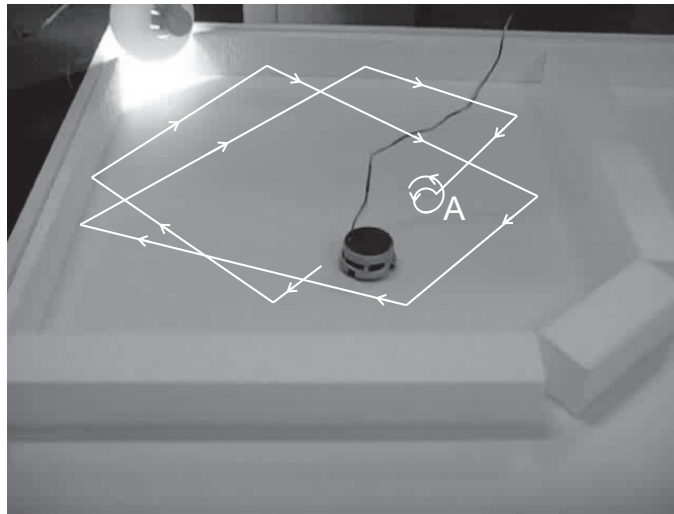
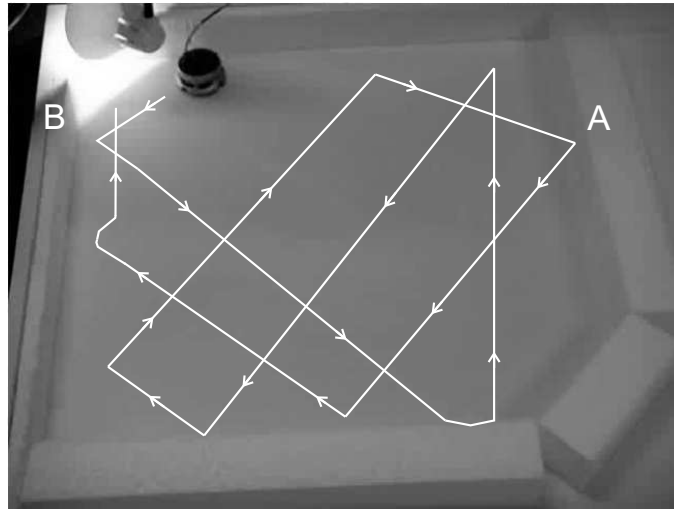


Fig. 4.16: Trajetória (a) e dinâmica das variáveis (b) para SHAE-R. O ponto A indica o momento em que o hormônio começa a ser produzido pela segunda vez, e o ponto B indica o momento em que o robô aproxima-se da fonte luminosa, simbolizando a recarga de suas baterias.

Para verificar se a mesma performance pode ser obtida em outra plataforma de experimentação, uma comparação adicional é feita utilizando um robô real do tipo Khepera-II. As Figuras 4.17(a) e 4.17(b) ilustram as trajetórias para o Robô 1 e o Robô 2, respectivamente. Novamente, o Robô 1 falha ao lidar com a nova arena e “morre” (Ponto A).



(a)



(b)

Fig. 4.17: Trajetória do robô real controlado pelo SHAE (a) e trajetória do robô real controlado pelo SHAE-R (b). Em ambos os casos, o ponto A indica o momento em que o robô chaveia seu comportamento para fototaxia. No segundo caso, o ponto B indica o momento em que o robô encontra a fonte luminosa e consegue manter seu estado interno dentro do limite.

4.3.4 Discussão

Este capítulo descreveu uma proposta de sistema homeostático artificial evolutivo, composto por duas redes neurais artificiais evoluídas, coordenadas por um sistema endócrino artificial. As RNAs foram desenvolvidas segundo o modelo proposto em Vargas et al. (2007), inspirado em modulações neurais causadas por gases no cérebro humano. Investigou-se também a utilização de receptores de hormônio. O objetivo foi criar um sistema mais plausível biologicamente, inspirado nas regulações homeostáticas presentes na natureza, que seriam capazes de atuar no contexto de comportamento adaptativo e coordenação.

Deve-se enfatizar que, embora tenham-se utilizado duas RNAs, a evolução da coordenação de comportamento e a natureza das interações do hormônio na estrutura neural possibilitam a composição de comportamentos, ou seja, é possível que comportamentos não previstos emergjam como uma combinação dos dois comportamentos básicos das RNAs. Uma série de experimentos foram realizados para investigar a performance do sistema, bem como sua robustez a perturbações sensoriais internas.

Com base nos experimentos e na configuração do cenário, seria possível argumentar que apenas o comportamento exploratório seria suficiente para o sistema explorar a arena e manter seu estado interno dentro do limite, não havendo a necessidade da rede de fototaxia. De fato, se a taxa de descarga é pequena (β pequeno), apenas a rede de exploração e desvio de obstáculos, N1, seria suficiente para uma boa performance. Para evidenciar esse comportamento, um outro experimento é proposto, empregando-se dois sistemas: o sistema SHAE para os casos de $\beta = \{0,02, 0,01, 0,001\}$ e β variável; e uma adaptação do sistema SHAE, composto apenas pela rede N1 e adotando-se $\beta = 0,01$.

A Figura 4.18 ilustra uma comparação entre a distância média percorrida pelo robô em cada configuração experimental. O sistema SHAE original, para o caso $\beta = 0,001$, possui um desempenho tão bom quanto o sistema SHAE adaptado (só com a rede N1). Ou seja, o sistema adaptado consegue explorar a arena e manter seu estado interno dentro do limite tão bem quanto o sistema SHAE original. Entretanto, o objetivo do trabalho não é desenvolver um sistema capaz de obter maior performance na tarefa. Pretende-se desenvolver um sistema robusto o bastante para se auto-adaptar a perturbações e manter boa performance em circunstâncias variadas. Note que, quando β aumenta, a necessidade por

uma coordenação mais elaborada também aumenta. Nos experimentos 1, 2 e 3 acima, quando $\beta = 0,02$, o SHAE foi capaz de adaptar-se a perturbações, embora com redução da taxa de exploração. Ao se empregar apenas o comportamento de exploração, se $\beta = 0,02$, o estado interno do robô supera o valor 90, significando “morte”. Esse fato é ilustrado na Figura 4.19 e reforça o comportamento de regulação homeostática do sistema proposto.

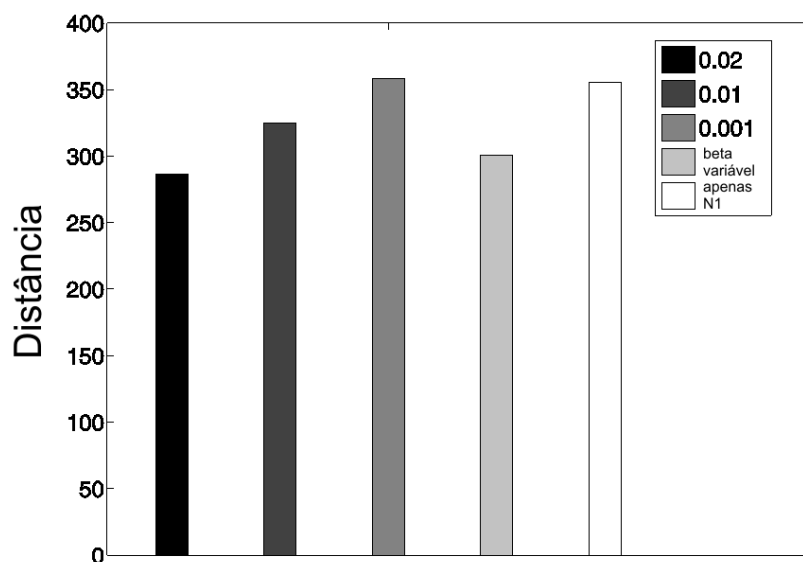


Fig. 4.18: Comparação entre a distância média percorrida pelo robô em cada configuração experimental.

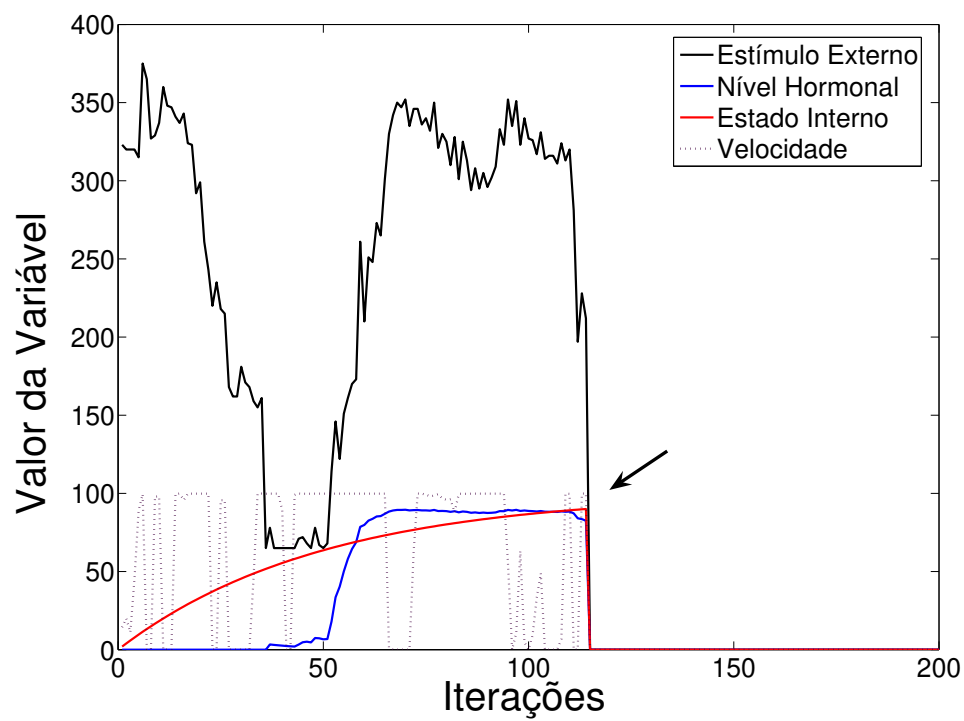


Fig. 4.19: Dinâmica das variáveis considerando apenas a rede N1 e $\beta = 0,02$. A seta indica o momento em que o robô “morre”.

Capítulo 5

Conclusão

Neste capítulo, apresentam-se as principais contribuições deste trabalho bem como extensões e perspectivas futuras relacionadas à área de computação bio-inspirada em robótica e sistemas cognitivos.

5.1 Análise das principais contribuições da tese

A complexidade crescente dos problemas enfrentados pelo ser humano e a necessidade de propor soluções com um grau de inteligência cada vez maior têm motivado os pesquisadores a se inspirarem na natureza para o desenvolvimento de técnicas e sistemas computacionais capazes de obter boas soluções quando técnicas mais tradicionais, que empregam métodos exatos e exaustivos, não são suficientes. Normalmente, esses sistemas computacionais bio-inspirados podem ser vistos como sistemas dinâmicos que reproduzem com algum grau de fidelidade sistemas observados na natureza.

Um sistema dinâmico é capaz de captar informações sobre o ambiente em que está inserido e tomar decisões com base nesses estímulos. As decisões ou ações podem produzir alterações neste ambiente, que por sua vez podem promover alterações sensoriais neste sistema, compondo um ciclo. A ciência cognitiva, ou mais recentemente a ciência cognitiva incorporada (Pfeifer & Scheier, 1999), estuda essa relação sistema-ambiente e suas implicações na manifestação e desenvolvimento de comportamento inteligente. Nesse contexto, a robótica autônoma móvel é considerada um paradigma apropriado para se estudar os princípios da cognição.

Diversas formas de treinamento são propostas na literatura em uma tentativa de estabelecer e preparar o sistema para lidar com diferentes problemas. Destacam-se duas propostas de treinamento: aprendizado por reforço e aprendizado evolutivo.

Os objetivos principais deste trabalho foram estudar, desenvolver e aplicar ferramentas computacionais bio-inspiradas em navegação autônoma de robôs, visando compreender melhor os mecanismos de aprendizado, memória e adaptação. O capítulo 3 concentrou-se em paradigmas de aprendizado por reforço, enquanto que o capítulo 4 abordou o aprendizado evolutivo.

No capítulo 3, apresentou-se um estudo sobre modelos estendidos de sistemas classificadores com aprendizado, capazes de lidar com estados ambíguos e tarefas que demandam algum tipo de memória. Uma extensa revisão bibliográfica foi feita, e os principais trabalhos existentes foram analisados e utilizados como base para a implementação. Experimentos foram realizados, possibilitando a comparação e identificação de pontos críticos em cada sistema, atendendo à proposta inicial de estudo dos sistemas. As principais contribuições deste capítulo podem ser assim enumeradas:

1. análise da evolução de regras e suas numerosas peculiaridades;
2. análise qualitativa da estruturação de uma proposta de solução pelo sistema LCS e pelo sistema XCS em ambientes dinâmicos, sujeitos a ruído, com estados de ambiguidade e recompensa a longo prazo;
3. proposta de um elenco de experimentos pouco explorados na literatura de sistemas classificadores com aprendizado;
4. desenvolvimento de uma interface para auxílio na implementação dos controladores em robôs reais;
5. implementação de propostas de solução em um robô real.

No capítulo 4, detalha-se uma proposta de sistema homeostático artificial evolutivo, composto por duas redes neurais artificiais recorrentes do tipo NSGasNet evoluídas, coordenadas por um sistema endócrino artificial. As principais contribuições deste capítulo podem ser assim enumeradas:

1. emprego e evolução de um novo modelo de rede neural artificial recorrente (NSGasNet) em tarefas robóticas de desvio de obstáculos e fototaxia;

2. evolução do sistema endócrino artificial para coordenação de múltiplos comportamentos;
3. emprego do novo sistema homeostático artificial evolutivo em tarefas robóticas de navegação e busca de alvo;
4. constatação do potencial da ferramenta ao lidar com perturbações diversas.

Não há dúvidas de que as propostas acima podem ser estendidas ou incorporadas a outros problemas. A próxima seção descreve algumas perspectivas para trabalhos futuros.

5.2 Perspectivas futuras e propostas de extensão

Em estudos de sistemas cognitivos, sistemas dinâmicos e conexionistas são comumente empregados, sendo que cada abordagem possui características peculiares para a elaboração e compreensão de sistemas inteligentes. Apresentam-se, a seguir, sugestões para trabalhos futuros englobando as técnicas propostas.

5.2.1 Sistemas classificadores com aprendizado

Os principais desafios encontrados em SCAs foram estabelecer a estrutura de cada regra (codificação da parte antecedente e conseqüente) e como a recompensa deve ser distribuída. Vinculada à recompensa, a forma de competição entre as regras, que determina qual delas deverá atuar, possui papel decisivo e é também de determinação desafiadora.

O grande potencial dos SCAs está na capacidade de representação compacta do conhecimento adquirido e na interação das regras (paralelismo). Considerando esses fatos, algumas extensões são sugeridas:

- implementar codificação real nos sistemas SCA e XCS, utilizando técnica semelhante à descrita em Wilson (2000) e Dam et al. (2005). Com isso, a incorporação de conhecimento a priori do projetista se reduziria;
- incorporar ao processo evolutivo a possibilidade de variação de parâmetros em tempo de execução (com base em dados estatísticos de performance), como taxa de mutação, *crossover*,

tamanho da população. Isso aumentaria a gama de problemas tratáveis sem a necessidade de intervenção externa no sistema;

- estender a análise feita para outros modelos de SCAs, em particular o Sistema Classificador Antecipatório, que já possui alguns trabalhos relacionados na literatura e apresenta-se bastante promissor;
- investigar mecanismos para propiciar a formação de níveis hierárquicos dentre o conjunto de classificadores, conforme sugerido e estudado em Riolo (1989) e Lanzi et al. (2000);
- estudar a utilização de SCAs em conjunto com outros sistemas, como a proposta detalhada em Vargas et al. (2003), que emprega SCAs e sistemas imunológicos artificiais, formando uma arquitetura híbrida, que mostra-se capaz de obter bom desempenho em uma série de tarefas robóticas.

À parte do campo de robótica, cabe destacar a utilização crescente de modelos de SCAs em tarefas de mineração de dados, com resultados muito promissores (Abbass et al., 2004; Bacardit et al., 2008).

5.2.2 Sistema homeostático artificial evolutivo

A arquitetura proposta de SHAE implementa um sistema composto por redes neurais artificiais recorrentes do tipo NSGasNet em conjunto com um sistema endócrino artificial, composto por equações a diferenças acopladas.

A evolução de redes neurais artificiais recorrentes representa uma das propostas mais promissoras para o estudo de sistemas cognitivos (Husbands et al., 1997; Nolfi & Floreano, 2000). Redes neurais artificiais recorrentes podem ser utilizadas em diversos problemas dinâmicos (Funahashi & Nakamura, 1993), e a evolução, por sua vez, consegue explorar formas diferentes de organização desses sistemas, bem como permitir uma avaliação do comportamento global do agente, levando em conta seu desempenho na realização de determinada tarefa.

A incorporação do SEA, formando assim o chamado SHAE, oferece muitas perspectivas de análises futuras, sendo algumas listadas a seguir:

- investigar a capacidade de resolução de problemas e adaptação em função da modificação da função de *fitness*, como sugerido em Tuci et al. (2002), em comparação com a adaptação dinâmica dos parâmetros, conforme implementado em Di Paolo (2000) e Urzelai & Floreano (2001);
- promover uma análise maior das interações neuro-endócrinas, propondo mecanismos para a coordenação de comportamentos mais complexos;
- investigar as interações de mais de um comportamento e/ou a implementação de mais de um hormônio junto aos processos de regulação;
- avaliar o desempenho dos controladores evoluídos em uma ampla gama de tarefas que envolvem robôs reais.

5.3 Considerações finais

É comum na literatura a comparação de paradigmas evolutivos e técnicas de aprendizado por reforço (Moriarty et al., 1999; Schlesinger, 2003; de Croon et al., 2005). Porém, deve-se destacar que não se sabe ao certo qual arquitetura é a mais apropriada para a concepção de um sistema cognitivo, se o ferramental matemático e computacional atual já seria suficiente para atender a demanda atual e futura, nem se a melhor abordagem seria uma versão híbrida das diversas técnicas (Beer, 2000). O que se pode afirmar é que a busca coerente e incremental por novos experimentos e perspectivas de análise configura-se na melhor opção para pesquisas futuras no campo da ciência cognitiva e robótica. Nesse sentido, o presente trabalho investigou e contribuiu com novas análises e perspectivas de ferramentas inéditas e também aquelas amplamente utilizadas na literatura para o estudo de sistemas cognitivos em navegação autônoma de robôs.

Apêndice A

Robô Khepera II

A.1 Introdução

Desenvolvido para ser utilizado em atividades educacionais e de pesquisa, o mini-robô Khepera II foi amplamente aceito e empregado nas mais diversas frentes de pesquisa em robótica móvel. Sua difusão permitiu a grupos distintos a realização e comparação de experimentos, contribuindo para a análise dos resultados. A seção a seguir detalha os aspectos construtivos e funcionalidades desse robô.

A.2 Robô Khepera II

O Khepera II é um mini-robô, desenvolvido pela empresa K-Team (KTEAM, 2007), e é mostrado na Figura A.1. Ele mede 70 mm de diâmetro e 30 mm de altura, e pesa aproximadamente 80 g. Sua alimentação é feita através de um cabo conectado a um adaptador ou então através de baterias internas recarregáveis, que possuem autonomia aproximada de uma hora (em caso de movimento frontal contínuo). Seu tamanho reduzido e a possibilidade de alimentação sem a conexão de cabos permitem a realização de experimentos em ambientes limitados, uma necessidade para muitos grupos de pesquisa.

O robô possui duas rodas conectadas a motores elétricos independentes, que são responsáveis por sua locomoção. Dois pinos de contato deslizantes proporcionam equilíbrio. A direção e a velocidade



Fig. A.1: Mini-robô Khepera II (adaptado de www.k-team.com)

do movimento são determinados pela diferença entre as velocidades das roda e seu sentido de rotação. A velocidade máxima que o robô alcança é de $1m/s$, e a mínima é de $0,08m/s$.

O robô possui 8 sensores, formados por emissores e receptores de luz infra-vermelha, dispostos ao redor de seu corpo. Através da análise da intensidade de luz infra-vermelha e do tempo de retorno de um raio emitido e refletido, é possível associar um valor para a luminosidade do ambiente e para a distância de um objeto até o robô. O alcance dos sensores, em relação a obstáculos, é de no máximo $10cm$. A cada $20ms$ uma medição completa dos valores dos oito sensores de distância e luminosidade é feita. Cada medição de um sensor é um valor analógico convertido para um número de 10 bits. A Figura A.2 traz um esquemático do robô.

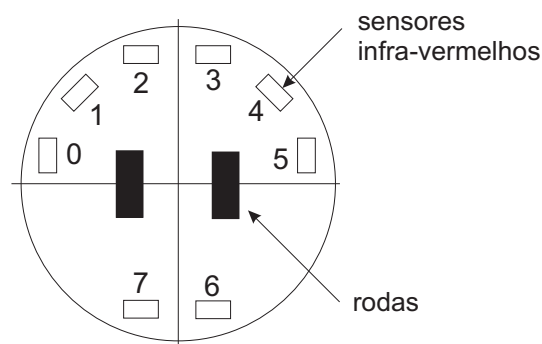


Fig. A.2: Esquemático do robô, com a disposição dos sensores infra-vermelhos e das rodas

Fatores externos, como a presença de lâmpadas incandescentes, podem causar interferência nas medições dos sensores de distância. Isto se deve ao fato de que o mesmo sensor é utilizado para as duas tarefas (distância e luminosidade). Por exemplo, imagine o sensor de distância. Ele emite um raio infravermelho e calcula a distância ao obstáculo com base no tempo que este raio levou para

sair do emissor e retornar ao receptor. Porém, caso exista uma lâmpada incandescente por perto, sua luminosidade afetará os raios recebidos, alterando a leitura sensorial. Entretanto, não se pode utilizar outro tipo de lâmpada em experimentos que envolvam luz, pois o sensor de luminosidade só capta a faixa de infravermelho. Portanto, o cuidado ao se realizar os experimentos deve ser grande, para minimizar os níveis de interferência. Este fato parece ter sido notado pelo fabricante, e a nova série de robôs Khepera, o Khepera III, possui sensores de distância baseados em ultrassom, ficando imunes a interferências de fontes luminosas.

O robô é comandado por um microcontrolador, e possui rotinas básicas para seu controle. O sistema de controle (sequência de instruções) pode ser carregado na memória interna do robô ou então ser transmitido via computador através de um cabo serial. Diversos pacotes de software permitem a elaboração de programas de controle em linguagens de alto-nível. O Apêndice B descreve os programas utilizados nesta tese para a elaboração dos sistemas de controle.

Apêndice B

Simuladores

B.1 Introdução

Graças aos avanços tecnológicos e à capacidade de processamento atual, a modelagem matemática e a simulação computacional de sistemas robóticos vem sendo executada com sucesso. Representar o ambiente e o robô em simulação computacional contribui para uma maior flexibilidade no tratamento de problemas exaustivos e redução do tempo de espera de resultados. Em Nolfi & Floreano (2000), os autores citam um experimento cuja realização completa com robôs reais levou cerca de 66 horas. Em contrapartida, o mesmo experimento seria reduzido a apenas 1 hora via simulação.

As seções a seguir detalham os simuladores utilizados nos experimentos desta tese - Simulador WSU Khepera Robot Simulator, para os experimentos do Capítulo 3, e Simulador KIKS, para os experimentos do Capítulo 4.

B.2 Simulador WSU Khepera Robot Simulator

O simulador WSU Khepera Robot Simulator foi desenvolvido por Perreta & Gallagher (2004), na Wright State University (Figura B.1). Trata-se de um programa que retrata de maneira razoável o comportamento sensorial do robô real, e que implementa uma série de funcionalidades que facilitam a execução de simulações.

A tela inicial do simulador consiste de uma arena, uma reprodução da imagem do robô e de seus

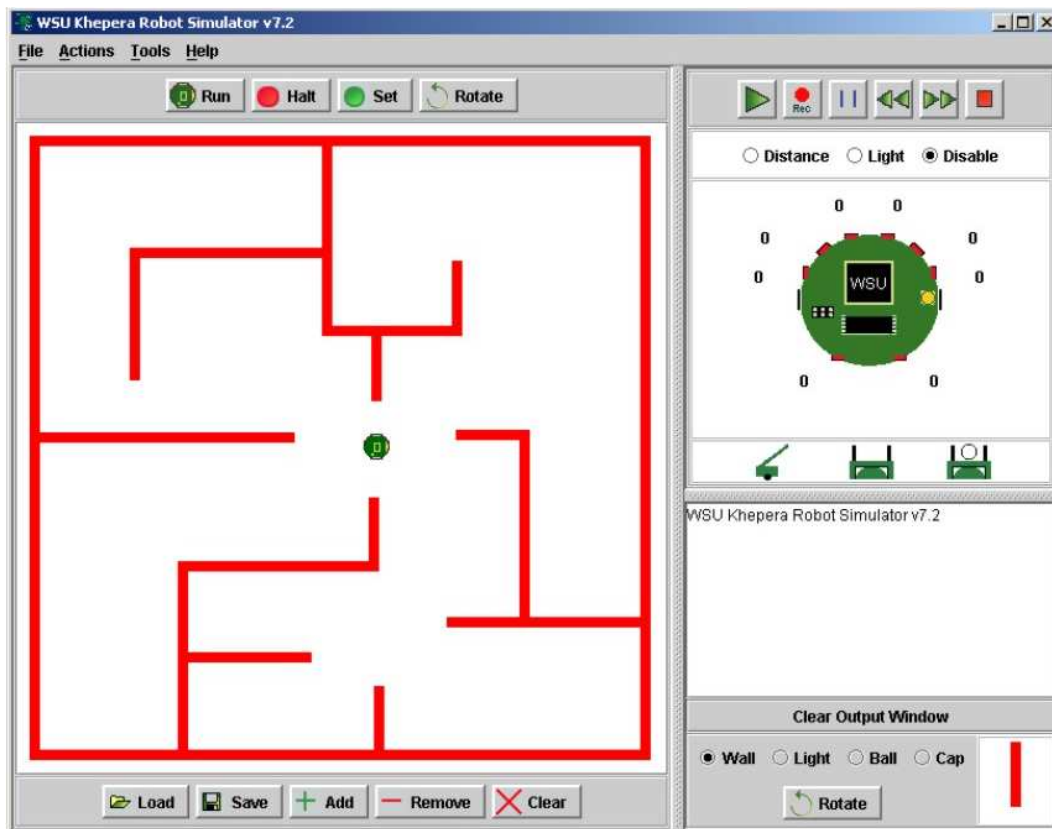


Fig. B.1: Ambiente de simulação do Simulador WSU Khepera Robot Simulator.

sensores, e de botões de controle. O simulador permite a criação de obstáculos no ambiente (em vermelho) e a inserção de lâmpadas (círculo amarelo). O usuário pode observar as leituras sensoriais de distância e luminosidade no canto direito superior, ou pode optar por não fazer nenhuma leitura. Os sensores de distância variam de 0 (ausência de obstáculo) a 1023 (obstáculo muito próximo). Os sensores de luminosidade variam de 50 (máxima luminosidade) a 500 (ausência de luz).

Através do botão *Run*, o programador carrega seu programa, carrega o ambiente desejado através do botão *Load*, e posiciona o robô com o botão *Set*. O simulador faz também a comunicação com o robô real, configurando automaticamente os parâmetros de comunicação serial.

O programa deve ser codificado utilizando a linguagem JAVA™. O simulador fornece uma série de funções pré-definidas (APIs), que permitem ao programador controlar a velocidade de cada roda, obter o valor de cada sensor de distância ou luminosidade, entre outros.

Por fim, é importante ressaltar que este é um dos poucos simuladores de uso gratuito e de código aberto disponíveis.



Fig. B.2: Interface para auxílio em experimentos com robô real e o simulador WSU.

B.2.1 Interfaces

Para auxiliar no acompanhamento e no desenvolvimento dos controladores implementados com o WSU Khepera Robot Simulator, foram desenvolvidas duas interfaces, uma para auxílio na navegação do robô e outra para acompanhamento da população de classificadores (Figuras B.2 e B.3). Estas interfaces foram desenvolvidas em linguagem *JAVATM*, e interagem com o simulador. No primeiro caso, a interface possibilita a configuração dos parâmetros de comunicação do software com o robô real, o acompanhamento da leitura de cada sensor e a movimentação passo a passo do robô através das setas localizadas à direita da interface. Na Figura B.3, cada fração da população de classificadores pode ser observada, desde as partes antecedente e consequente até variáveis como energia, erro de predição, número total de classificadores e número de gerações.

	A	P	E	F	N	Exp
01	000,000	000,000	000,208	001,000	001,000	001,000
10	000,000	000,000	000,493	001,000	001,000	003,000
00	000,000	000,000	000,208	001,000	001,000	001,000
01	000,000	000,000	000,010	000,000	001,000	000,000
10	000,000	000,000	000,010	000,000	001,000	000,000
11	000,000	000,000	000,208	001,000	001,000	001,000
01	000,000	000,000	000,208	001,000	001,000	001,000
10	000,000	000,000	000,208	001,000	001,000	001,000
00	000,000	000,000	000,010	000,000	001,000	000,000
01	000,000	000,000	000,208	001,000	001,000	001,000
10	000,000	000,000	000,010	000,000	001,000	000,000
00	000,000	000,000	000,010	000,000	001,000	000,000
01	000,000	000,000	000,010	000,000	001,000	000,000
10	000,000	000,000	000,010	000,000	001,000	000,000
00	000,000	000,000	000,208	001,000	001,000	001,000
01	000,000	000,000	000,010	000,000	001,000	000,000
10	000,000	000,000	000,010	000,000	001,000	000,000
00	000,000	000,000	000,208	001,000	001,000	001,000
01	000,000	000,000	000,010	000,000	001,000	000,000
10	000,000	000,000	000,010	000,000	001,000	000,000
11	000,000	000,000	000,208	001,000	001,000	001,000
01	000,000	000,000	000,208	001,000	001,000	001,000
10	1000,000	1000,000	000,008	000,000	001,000	001,000
11	000,000	000,000	000,010	000,000	001,000	000,000

Fig. B.3: Interface para auxílio às simulações do Capítulo 3.

B.3 Simulador KIKS

O simulador KiKS (*KiKS is a Khepera Simulator*, em inglês) foi desenvolvido por Storm (2004), e é ilustrado na Figura B.4. Este simulador foi feito para ser utilizado junto ao programa Matlab®, usando um *toolbox* para controle do robô Khepera, desenvolvido pelo K-Team, chamado kMatlab (o *toolbox* pode ser obtido gratuitamente em <http://www.k-team.com/download/khepera.html>).

A tela inicial do simulador é formada por uma arena, um robô, e um painel de controle (à esquerda) que contém botões e *displays* relevantes à simulação. O botão *edit* permite inserir obstáculos de formas variadas no ambiente, bem como aumentar ou diminuir o tamanho da arena. Os botões *load/new/save* permitem carregar ou salvar uma configuração de arena. Pode-se adicionar lâmpadas utilizando o botão *add*. O conjunto de botões seguinte são relativos à configuração da simulação (local ou em rede), o tempo de simulação e a velocidade de simulação (pode-se acelerar a simulação, reduzindo o tempo real do experimento). Ao clicar no botão *monitor simulation details*, o usuário pode observar as leituras sensoriais de distância e luminosidade. Os sensores de distância variam de

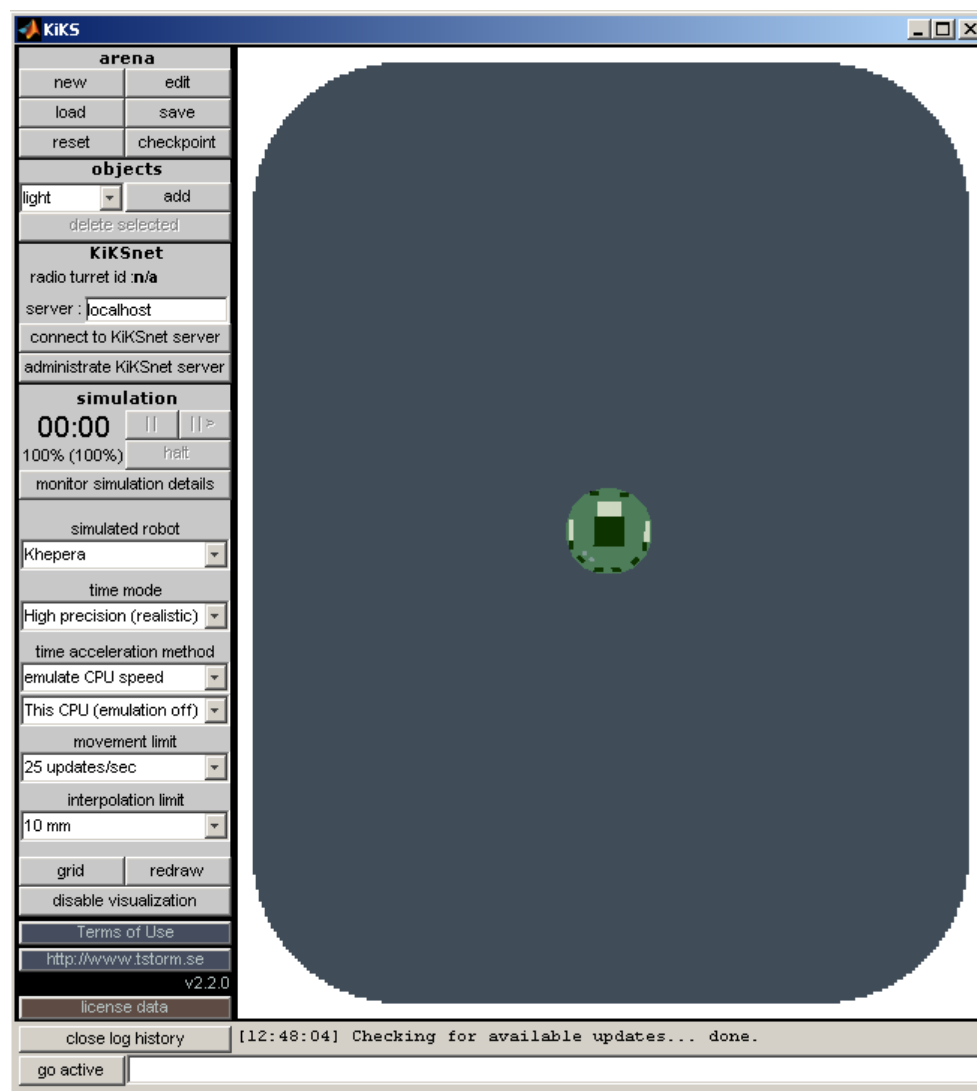


Fig. B.4: Ambiente de simulação do Simulador KIKS.

0 (ausência de obstáculo) a 1023 (obstáculo muito próximo). Os sensores de luminosidade variam de 50 (máxima luminosidade) a 500 (ausência de luz). Para uma descrição completa dos botões e funcionalidades, o leitor deve consultar o manual disponível em Storm (2004).

O programa deve ser codificado utilizando o software Matlab. O simulador fornece uma série de funções pré-definidas, que permitem ao programador controlar a velocidade de cada roda, obter o valor de cada sensor de distância ou luminosidade, entre outros. É possível também realizar simulações via rede, com mais de um robô.

Referências Bibliográficas

- Abbass, H. A., Bacardit, J., Butz, M. V., & Llorca, X. (2004). Online adaption in learning classifier systems: Stream data mining. *IlliGAL Report No. 2004031, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.*
- Andrew, A. M. (1975). Why robotics? *Kybernetes*, vol.4, pp. 2-8.
- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.
- Arkin, R. C. (1998). *Behavior-based Robotics*. MIT Press.
- Armano, G. (2004). NXCS experts for financial time series forecasting. In Bull, L. (ed) *Applications of Learning Classifier Systems*.
- Ashby, W. R. (1952). *Design for a Brain: The Origin of Adaptive Behaviour*. London: Chapman and Hall.
- Bacardit, J., Stout, M., Hirst, J., & Krasnogor, N. (2008). Data mining in proteomics with learning classifier systems. Bull, L., Bernado Mansilla, E. and Holmes, J. (eds), *Learning Classifier Systems in Data Mining*, Springer.
- Bäck, T., Fogel, D. B., & Michalewicz, Z., Eds. (2000a). *Evolutionary Computation 1: Basic Algorithms and Operators*. Bristol, UK: Institute of Physics Publishing.
- Bäck, T., Fogel, D. B., & Michalewicz, Z., Eds. (2000b). *Evolutionary Computation 2: Advanced Algorithms and Operators*. Bristol, UK: Institute of Physics Publishing.

- Barandiaran, X. & Moreno, A. (2006). On what makes certain dynamical systems cognitive: A minimally cognitive organization program. *Adaptive Behavior*, vol. 14. pp.171-185.
- Barra, T. V. (2007). Um ambiente evolutivo para apoio ao projeto de antenas de microfita. *Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, SP, Brasil*.
- Beer, R. D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, Vol.4, No.3.
- Bekey, G. A. (2005). *Autonomous Robots, From Biological Inspiration to Implementation and Control*. MIT Press.
- Besendovsky, H. O. & Del Rey, A. (1996). Immune-neuro-endocrine interactions: Facts and hypotheses. *Endocrine Reviews*, 17, 64–102.
- Boden, M. A. (2006). *Mind as machine: a history of cognitive science*. Oxford University Press. Volumes I and II. Great Britain, UK.
- Bonabeau, E., Dorigo, M., & Theraulaz, T. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press.
- Bonarini, A. (1996). Fuzzy modelling: Paradigms and practice. W. Pedrycz (Ed.), *Fuzzy Modelling: Paradigms and Practice*, Kluwer Academic Press, Norwell, MA.
- Bonarini, A. (2000). An introduction to learning fuzzy classifier systems. *Learning Classifier Systems, From Foundations to Applications*, LNCS Vol. 1813, p.83-106.
- Bonarini, A. & Matteucci, M. (2007). FIXCS: a fuzzy implementation of XCS. *IEEE International Fuzzy Systems Conference, FUZZ-IEEE 2007*.
- Booker, L. B., Goldberg, D. E., & Holland, J. H. (1989). Classifier systems and genetic algorithms. *Artificial Intelligence*, 40, 235–282.
- Bowling, M. & Veloso, M. (2003). Simultaneous adversarial multi-robot learning. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 699–704.

- Bull, L. (2004). *Applications of Learning Classifier Systems*. Springer-Verlag.
- Butz, M. & Wilson, S. W. (2000). An algorithmic description of XCS. *Lecture Notes In Computer Science*, vol. 1996, pp. 267–274.
- Butz, M. & Wilson, S. W. (2002). An algorithmic description of ACS2. *Lecture Notes In Artificial Intelligence*, vol. 2321, pp. 361–390.
- Butz, M. V., Goldberg, D. E., & Tharakunnel, K. (2003). Analysis and improvement of fitness exploitation in XCS: Bounding models, tournament selection, and bilateral accuracy. *Evolutionary Computation*, 11(3), 239–277.
- Cannon, W. B. (1929). Organization for physiological homeostasis. *Physiological Review*, 9, 399–431.
- Cazangi, R. R. (2004). Uma proposta evolutiva para controle inteligente em navegação autônoma de robôs. *Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, SP, Brasil*.
- Cazangi, R. R., Zuben, F. J. V., & Figueiredo, M. (2003). A classifier system in real applications for robot navigation. *The IEEE Congress on Evolutionary Computation, Canberra, Australia*, vol. 1, pp. 574-580.
- Cazangi, R. R., Zuben, F. J. V., & Figueiredo, M. F. (2005). Autonomous navigation system applied to collective robotics with ant-inspired communication. *Genetic and Evolutionary Computation Conference*.
- Changeux, J.-P. (1993). Chemical signaling in the brain. *Scientific American*, 269(5), 58–62.
- Churchland, P. & Sejnowski, T. J. (1994). *The Computational Brain*. MIT Press.
- Cliff, D. & Ross, S. (1995). Adding temporary memory to ZCS, vol. 3, no. 2, 101–150. *Adaptive Behavior*.
- Collins, R. & Jefferson, D. (1991). Selection in massively parallel genetic algorithms. In *Proceedings of the Fourth Int. Conf. on Genetic Algorithms, ICGA-91* (pp. 249–256).: Morgan Kaufmann.

- Copeland, J. & Proudfoot, D. (2004). Alanturing.net: The turing archive for the history of computing. http://www.cs.usfca.edu/www.AlanTuring.net/turing_archive/index.html.
- Crestani, P. (2001). Sistemas inteligentes de navegação autônoma: Uma abordagem modular e hierárquica com novos mecanismos de memória e aprendizagem. *Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, SP, Brasil*.
- Dam, H. H., Abbass, H. A., & Lokan, C. (2005). Be real! XCS with continuous-valued inputs. *GECCO05: Proceedings of the 2005 workshops on Genetic and evolutionary computation, New York, NY, USA*, (pp. 85–87).
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London, UK: John Murray.
- Dayan, P. & Abbot, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*. MIT Press.
- de Castro, L. & Timmis, J. (2002). *Artificial Immune Systems: A New Computational Approach*. London, UK.: Springer-Verlag.
- de Castro, L. & Von Zuben, F. J. E. (2004). *Recent Developments in Biologically Inspired Computing*. Idea Group Publishing.
- de Croon, G., van Dartel, M., & Postma, E. (2005). Evolutionary learning outperforms reinforcement learning on non-markovian tasks. *Learning and Memory Workshop of the European Conference on Artificial Life (ECAL) 2005, Canterbury, UK*.
- Di Paolo, E. A. (2000). Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. In *From Animals to Animals, Proc. of the 6th Int. Conf. on the Simulation of Adaptive Behavior, SAB'2000* (pp. 440–449).: MIT Press.
- Di Paolo, E. A. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences*, 4(4): 429 - 452.

- Dorigo, M. & Colombetti, M. (1997). *Robot Shaping: An Experiment in Behavior Engineering*. MIT Press.
- Dyke, J. & Harvey, I. (2005). Hysteresis and the limits of homeostasis: From daisyworld to phototaxis. In *In Capcarrere, M, Freitas, A A, Bentley, P J, Johnson, C G and Timmis, J, editors, Advances in Artificial Life: 8th European Conf., ECAL 2005, Canterbury, UK, September 5-9*: Springer-Verlag.
- Dyke, J. G. & Harvey, I. R. (2006). Pushing up the daisies. In *In Rocha, L. M., Larry, S. Y., Bedau, M. A., Floreano, D., Goldstone, R. L. and Vespignani, A (Eds.), Alife X Proc. of the 10th Int. Conf. on the Simulation and Synthesis of Living Systems*.
- Figueiredo, M. (1999). Navegação autônoma de robôs. *VII Escola de Informática da SBC - Regional Sul*, (pp. 74–106).
- Fogel, D. B. (1994). An introduction to simulated evolutionary computation. *IEEE Transactions on Neural Networks*, vol. 5, no. 1, (pp. 3–14).
- Fogel, L. J. (1962). Autonomous automata. *Industrial Research*, vol. 4, (pp. 14–19).
- Funahashi, K. & Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6, (pp. 1–64).
- Gally, J. A., Montague, P. R., Reeke, G. N., & Edelman, G. M. (1990a). The no hypothesis: Possible effects of a short-lived, rapidly diffusible signal in the development and function of the nervous system. In *Proceedings of the National Academy of Sciences of the USA*, 87 (pp. 3547–3551).: Springer Verlag.
- Gally, J. A., Montague, P. R., Reeke Jr, G. N., & Edelman, G. M. (1990b). The NO hypothesis: possible effects of a short-lived, rapidly diffusible signal in the development and function of the nervous system. *Proceedings of the National Academy of Science, USA*, vol. 87, (pp. 3547–3551).
- Gerard, P. & Sigaud, O. (2000). YACS: Combining dynamic programming with generalization in classifier systems. In *Stolzmann, W., Lanzi, P. L., and Wilson, S. W., (Eds.), Proceedings of the*

- Third International Workshop on Learning Classifier Systems, LNCS vol. 1996, Springer-Verlag., (pp. 52–69).*
- Gigliotta, O. & Nolfi, S. (2008). On the coupling between agent internal and agent/environmental dynamics: Development of spatial representations in evolving autonomous robots. *Adaptive Behavior, vol. 16. pp.148-165.*
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* New York: Addison-Wesley.
- Guyton, A. C. & Hall, J. E. (1996). *Textbook of Medical Physiology.* W. B. Saunders Company, Philadelphia, 9th edition.
- Hall, M. & Halliday, T. (1998). *Behaviour and Evolution, Biology: Brain and Behaviour Series, The Open University.* Springer-Verlag.
- Hall, M. & Halliday, T. (2004). *Imitation of Life: How Biology is Inspiring Computing.* MIT Press.
- Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal, 26(2):147-160.*
- Harvey, I. (2004). Homeostasis and rein control: From daisyworld to active perception. In *Proc. of the 9th Int. Conf. on the Simulation and Synthesis of Living Systems, ALIFE9:* MIT Press.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 2nd edition.
- Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D, 42, 228–234.*
- Hoffmann (1992). Probleme der begriffsbildungsforschung: Von s-r verbindungen zu sr-k einheiten. *Sprache und Kognition, 11, 223–238.*
- Hoinville, T. & Hénaff, P. (2004). Comparative study of two homeostatic mechanisms in evolved neural controllers for legged locomotion. In *Proccedings of 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2624–2629, vol.3.*

- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
- Holland, J. H. (1995). *Hidden Order*. Addison-Wesley Reading, MA.
- Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. (1986). *Induction: Processes of Inference, Learning, and Discovery*. MIT Press.
- Holmes, J. H., Lanzi, P. L., Stolzmann, W., & Wilson, S. (2000). Learning classifier systems: new models, successful applications.
- Holscher, C. (1997). Nitric oxide, the enigmatic neuronal messenger: its role in synaptic plasticity. *Trends in Neuroscience vol. 20*, (pp. 298–303).
- Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA, vol. 9*.
- Hurst, J. & Bull, L. (2006). A neural learning classifier system with self-adaptive constructivism for mobile robot control. *Artif. Life*, 12(3), 353–380.
- Hurst, J., Bull, L., & Melhuish, C. (2002). TCS learning classifier system controller on a real robot. *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature, LNCS vol. 2439*, pp. 588–597.
- Husbands, P. (1998). Evolving robot behaviours with diffusing gas networks. *Evolutionary Robotics: First European Workshop, Proceedings of Evorob98., Springer Verlag*, pp. 71–86.
- Husbands, P., Harvey, I., Cliff, D., & Miller, G. (1997). Artificial evolution: A new path for artificial intelligence? *Brain and Cognition*, 34, (pp. 130–159).
- Husbands, P., Smith, T., Jakobi, N., & O Shea, M. (1998). Better living through chemistry: Evolving GasNets for robot control. *Connection Science*, 10, 185–210.
- Izquierdo, E. (2004). *Collective Intelligence in Multi-Agent Robotics: Stigmergy, Self-Organization and Evolution*. Technical report, University of Sussex.

- Jakobi, N. (1998). Minimal simulations for evolutionary robotics. *University of Sussex*.
- Jefress, L. e. (1967). *Cerebral Mechanisms of Behavior*. Hafter New York, 1st edition.
- Katagami, D. & Yamada, S. (2000). Interactive classifier system for real robot learning. *Proceedings of the 2000 IEEE International Workshop on Robot and Human Interactive Communication*, pp. 258–263, Osaka, Japan.
- Kennedy, J., Eberhart, R., & Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78:1464-1480.
- Kovacs, T. (2002). A learning classifier systems bibliography, <http://www.cs.bris.ac.uk/~kovacs/lcs/search.html>.
- Kovacs, T. (2004). *Strength or Accuracy: Credit Assignment in Learning Classifier Systems*. Springer.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of computers by Means of Natural Selection*. MIT Press.
- KTEAM, S. A. (2007). <http://www.k-team.com>.
- Kumar, S. & Bentley, P. J. (2003). *An introduction to computational development*. In: S. Kumar & P. J. Bentley (Eds.), *On Growth, Form and Computers*, chapter 1, pages 1-44. Academic Press, New York, NY, USA.
- Lancaster, J. (1996). Diffusion of free nitric oxide. *Methods in Enzymology*, 268, 31–50.
- Lanzi, P. L. (1998). An analysis of the memory mechanism of XCSM. *Proceedings of the Third Annual Conference on Genetic Programming*, pp. 643–651, Morgan Kaufmann.
- Lanzi, P. L., Stolzmann, W., & Wilson, S. W., Eds. (2000). *Learning Classifier Systems, From Foundations to Applications*, London, UK. Springer-Verlag.
- Lanzi, P. L. & Wilson, S. W. (2000). Toward optimal classifier system performance in non-markov environments. *Evolutionary Computation*, 8(4), 393–418.

- McClelland, J. L., Rumelhart, D. E., & Group, T. P. R. (1986). Parallel distributed processing: Explorations in the microstructure of cognition. *Psychological and Biological Models, The MIT Press, volume 2*.
- Michalewicz, Z. (1996). Genetic algorithms + data structures = evolution programs. *Springer*.
- Michalewicz, Z. & Fogel, D. B. (2000). *How to solve It: Modern Heuristics*. Springer.
- Michalewicz, Z., Schmidt, M., Michalewicz, M., & Chiriac, C. (2006). *Adaptive Business Intelligence*. Springer.
- Miller, B. L. & Goldberg, D. E. (1996). Genetic algorithms, tournament selection, and the effect of noise. *Complex Systems, vol. 9, pp. 193–212*.
- Minsky, M. & Papert, S. (1969). *Perceptrons*. MIT Press, Cambridge.
- Moioli, R. C., Vargas, P. A., & Von Zuben, F. J. (2008). Towards the evolution of an artificial homeostatic system. In *2008 IEEE Congress on Evolutionary Computation (CEC 2008)*, pages 4024-4031.
- Montebelli, A., Herrera, C., & Ziemke, T. (2008). On cognition as dynamical coupling: An analysis of behavioral attractor dynamics. *Adaptive Behavior, vol. 16, pp.182-195*.
- Moriarty, D. E., Schultz, A. C., & Grefenstette, J. J. (1999). Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research, 11*, (pp. 241–276).
- Neal, M. & Timmis, J. (2003). Timidity: A useful mechanism for robot control. *Informatica, 7*, 197–203.
- Nolfi, S. & Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Bradford Book.
- O'Reilly, R. C. & Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press.
- Pedrycz, W. & Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, Cambridge, EUA.

- Perreta, S. J. & Gallagher, J. (2004). The Java Khepera Simulator from the Wright State University, Ohio, USA.
- Pfeifer, R. & Scheier, C. (1999). *Understanding Intelligence*. MIT Press.
- Philippides, A., Husbands, P., Smith, T., & O'Shea, M. (2005a). Flexible couplings: Diffusing neuromodulators and adaptive robotics. *Artificial Life*, 11, 139–160.
- Philippides, A., Ott, S. R., Husbands, P., Lovick, T., & O'Shea, M. (2005b). Modeling cooperative volume signaling in a plexus of nitric oxide synthase-expressing neurons. *The Journal of Neuroscience*, 25, 6520–6532.
- Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Phys. Review Letters*, 59, 2229–2232.
- Purves, W. K., Heller, H. C., Orians, G. H., & Sadava, D. (2001). *Life: The Science of Biology*. IE-Macmillan UK., 6th edition.
- Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog.
- Riolo, R. (1989). The emergence of default hierarchies in learning classifier systems. *Proceedings of the Third Congress on Genetic Algorithms*, pp.322–327, Morgan Kaufmann.
- Ross, P., Hart, E., Lawson, A., Webb, A., Prem, E., Poelz, P., & Morgavi, G. (2003). Requirements for getting a robot to grow-up. *Proceedings of the European Conference on Artificial Life, ECAL 2003*.
- Rumelhart, D. E., McClelland, J. L., & Group, T. P. R. (1986). Parallel distributed processing: Explorations in the microstructure of cognition. *Psychological and Biological Models, The MIT Press, volume 1*.
- Russell, S. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition.

- Santos, C. L. R. (2001). Reavaliações da gasnet e sua eficácia: além da proposição inicial. *Tese de Doutorado, Instituto Tecnológico de Aeronáutica, ITA, SP, Brasil.*
- Santos, E. P. & Von Zuben, F. J. (2000). Efficient second-order learning algorithms for discrete-time recurrent neural networks. *L.R. Medsker; L.C. Jain. (Org.). Recurrent Neural Networks: Design and Applications. 1 ed. Washington D. C.: CRC Press, v. 1, p. 47–75.*
- Schalkoff, R. J. (1990). *Artificial intelligence : an engineering approach*. McGraw-Hill, New York :.
- Schank, R. C. (1987). What is AI, anyway? *AI Mag.*, 8(4), 58–65.
- Schlesinger, M. (2003). A lesson from robotics: Modeling infants as autonomous agents. *Adaptive Behavior*, 11:2, (pp. 97–107).
- Schwefel, H. P. (1965). Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. *Tese de Doutorado, Hermann Föttinger Institut für Strömungstechnik, Technische Universität, Berlin.*
- Schwefel, H.-P. (1981). *Numerical Optimization for Computer Models*. John Wiley.
- Sipper, M. (2002). *Machine Nature: The Coming Age of Bio-Inspired Computing*. McGraw-Hill, New York.
- Smith, T., Husbands, P., & Oshea, M. (2003). Temporally adaptive networks: Analysis of GasNet robot controllers. *ICAL 2003: Proceedings of the eighth international conference on Artificial life, Cambridge, MA, USA*, (pp. 274–282).
- Spillman, L. & Wooten, B., Eds. (1984). *Sensory Experience, Adaptation and Perception: A Festschrift for Ivo Kohler*. Lawrence Erlbaum.
- Stolzmann, W. (1996). Learning classifier systems using the cognitive mechanism of anticipatory behavioural control. *Proceedings of the First European Workshop on Cognitive Modelling*, (pp. 82–89).
- Stolzmann, W. & Butz, M. (1999). Latent learning and action planning in robots with anticipatory classifier systems. *Lecture Notes In Computer Science, vol. 1813, pp. 301–320.*

- Storm, T. (2004). KIKS, a Khepera simulator for Matlab 5.3 and 6.0. <http://theodor.zoomin.se/index/2866.html>.
- Studley, M. & Bull, L. (2005). X-TCS: accuracy-based learning classifier system robotics. *The IEEE Congress on Evolutionary Computation*, pp. 2099–2106, vol. 3.
- Sutton, R. & Barto, A. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.
- Timmis, J. & Neal, M. (2004). *Once More Unto the Breach: Towards Artificial Homeostasis*, chapter L. N. de Castro and F. J. Von Zuben, Recent Developments in Biologically Inspired Computing, (pp. 340–366). Idea Group Inc.
- Tuci, E., Quinn, M., & Harvey, I. (2002). An evolutionary ecological approach to the study of learning behavior using a robot-based model. *Adaptive Behavior*, 10(3-4), (pp. 201–221).
- Urzelai, J. & Floreano, D. (2001). Evolution of adaptive synapses: Robots with fast adaptive behavior in new environments. *Evolutionary Computation* 9(4), (pp. 495–524).
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, New York, NY, USA.
- Vargas, P. A. (2000). Sistemas classificadores para redução de perdas em redes de distribuição de energia elétrica. *Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, SP, Brasil*.
- Vargas, P. A., de Castro, L. N., Michelan, R., & Von Zuben, F. J. (2003). An immune learning classifier network for autonomous navigation. *Second International Conference on Artificial Immune Systems (ICARIS'2003), 2003, Edinburgh. Proceedings of the Second International Conference on Artificial Immune Systems*, vol. 1, pp. 69–80.
- Vargas, P. A., Di Paolo, E. A., & Husbands, P. (2007). Preliminary investigations on the evolvability of a non-spatial GasNet model. In *Proc. of the 9th European Conf. on Artificial life ECAL 2007* (pp. 966–975).: Springer-Verlag.

- Vargas, P. A., Filho, C. L., & Zuben, F. J. V. (2004). Application of learning classifier systems to the on line reconfiguration of electric power distribution networks. *Applications of Learning Classifier Systems*, 150, 260–275.
- Vargas, P. A., Moiola, R. C., de Castro, L. N., Timmis, J., Neal, M., & Von Zuben, F. J. (2005). Artificial homeostatic system: a novel approach. In *Capcarrere, M.S.; Freitas, A.A.; Bentley, P.J.; Johnson, C.G.; Timmis, J.. (Org.). Advances in Artificial Life, Lecture Notes in Artificial Intelligence. : Springer, vol. 3630, pp. 754–764.*
- Webb, E., Hart, E., , Ross, P., & Lawson, A. (2003). Controlling a simulated Khepera with an XCS classifier system with memory. *Lecture Notes in Computer Science*, vol. 2801, pp. 885–892.
- Welch, R. B. (1974). Research on adaptation to rearranged vision. *Perception*, 3, 367–392.
- Wilson, S. W. (1994). ZCS: A zeroth level classifier system. *Evolutionary Computation*, 2(1), 1–18.
- Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation*, 3, 149–175.
- Wilson, S. W. (2000). Get real! XCS with continuous-valued inputs. *Lecture Notes in Computer Science*, vol. 1813, pp. 209–219.
- Wilson, S. W. (2002). Classifiers that approximate functions. *Natural Computing: an international journal*, 1(2-3), 211–234.
- Wood, J. & Garthwaite, J. (1994). Models of the diffusional spread of nitric oxide: Implications for neural nitric oxide signaling and its pharmacological properties. *Neuropharmacology*, 33, 1235–1244.
- Zatuchna, Z. V. (2005). Agentp: a learning classifier system with associative perception in maze environments. *Tese de Doutorado, School of Computing Sciences, University of East Anglia.*

Índice Remissivo de Autores

- Abbot, L. F. 17
- Andrew, A. M. 3
- Applegate, D. L. 12
- Arkin, R. C. 6, 68
- Armano, G. 35
- Ashby, W. R. 6, 65
- Barandiaran, X. 3
- Barra, Tiago Venturieri xvii, 12, 14, 15
- Barto, A.G. 5
- Bekey, G. A. 3
- Bentley, P. J. 2
- Besendovsky, H. O. 2, 66
- Bixby, R. E. 12
- Boden, M. A. 3
- Bonabeau, E. 2
- Bonarini, A. 46
- Booker, L. B. 6, 37
- Bowling, M. 4
- Bull, L. 35, 46–49, 61, 62
- Butz, M. 41, 45, 49
- Cazangi, R. R. 4, 37
- Changeux, J.-P. 21
- Chiriac, C. 1
- Churchland, P. 17
- Chvátal, V. 12
- Cliff, D. 44, 49
- Colombetti, M. 45
- Cook, W. J. 12
- Copeland, J. 1
- Crestani, P.R. 4
- Darwin, C. 9
- Dayan, P. 17
- de Castro, L. N. xviii, 2, 6, 7, 66, 67, 69, 70, 72, 73, 75
- de Castro, L.N. 2
- Del Rey, A. 2, 66
- Di Paolo, E. A. 6, 7, 27, 28, 66, 67, 99, 107
- Dorigo, M. 2, 45
- Dyke, J. 6, 66
- Eberhart, R. 2
- Edelman, G. M. 18, 21
- Figueiredo, M. 4

- Figueiredo, M. F. 4
Filho, C. Lyra 35
Floreano, D. 4–7, 34, 54, 68, 80, 106, 113
Fogel, D. B. 13
Fogel, David B. 10
Fogel, L. J. 10
Gally, J. A. 18, 21
Garthwaite, J. 21
Gerard, P. 46
Goldberg, D. E. 6, 37
Goldberg, David E. 10, 13, 36, 49
Gomide, F. 46
Group, The PDP Research 17
Hall, M. 1, 2
Halliday, T. 1, 2
Hamming, R. W. 38
Hart, E. 3, 47–49, 52, 54, 56, 61
Harvey, I. 6, 66
Haykin, S. 2, 16, 17, 67
Heller, H. C. 1, 69
Herrera, C. 3
Hoffmann 45
Holland, J. H. 2, 6, 7, 10, 14, 35, 37, 48, 49, 55
Holmes, J. H. 35
Holscher, C. 18
Holyoak, K. J. 6, 35
Hopfield, J. 16
Hurst, J. 35, 46–49
Husbands, P. 7, 18, 23–25, 27, 28, 67, 77, 99
Izquierdo, E. 4
Jakobi, N. xviii, 18, 23, 24, 54, 56, 77
Jefress, L. (ed.) 1
Katagami, D. 46
Kennedy, J. 2
Kohonen, T. 16
Kovacs, T. 42, 49, 62
Koza, J. R. 10
Kumar, S. 2
Lancaster, J. 21
Lanzi, P. L. 35, 47, 49, 57
Lawson, A. 3
Lovick, T. 18
Matteucci, M. 46
McClelland, J. L. 17
Melhuish, C. 35, 46, 48, 49
Michalevich, M. 1
Michalewicz, Z. 1, 10, 13
Miller, B. L. 13
Minsky, M. 16
Moioli, R. C. xviii, 2, 6, 7, 66, 67, 69, 70, 72, 73, 75
Montague, P. R. 18, 21
Montebelli, A. 3
Moreno, A. 3
Morgavi, G. 3
Munakata, Y. 17

- Neal, M. xviii, 2, 6, 7, 66, 67, 69, 70, 72, 73, 75
- Nisbett, R. E. 6, 35
- Nolfi, S. 4–7, 34, 54, 68, 80, 106, 113
- Norvig, Peter 3
- O Shea, M. 18, 23, 24, 77
- O'Reilly, R. C. 17
- Orians, G. H. 1, 69
- Oshea, M. 25
- Ott, S. R. 18
- Papert, S. 16
- Pedrycz, W. 46
- Pfeifer, R. 3–5, 65, 68, 103
- Philippides, A. 18
- Pineda, F. J. 16
- Poelz, P. 3
- Prem, E. 3
- Proudfoot, D. 1
- Purves, W. K. 1, 69
- Rechenberg, I. 10
- Reeke, G. N. 21
- Reeke Jr, G. N. 18
- Ross, P. 3
- Ross, S. 44, 49
- Rumelhart, D. E. 17
- Russell, Stuart 3
- Sadava, D. 1, 69
- Santos, C. L. R. 21
- Santos, E. P. 16
- Schalkoff, Robert J. 3
- Schank, Roger C. 3
- Scheier, C. 3–5, 65, 68, 103
- Schmidt, M. 1
- Schwefel, H. P. 10
- Schwefel, Hans-Paul 10
- Sejnowski, T. J. 17
- Shi, Y. 2
- Sigaud, O. 46
- Sipper, M. 2
- Smith, T. 18, 23–25, 77
- Stolzmann, W. 35, 45
- Studley, M. 35, 47, 61, 62
- Sutton, R.S. 5
- Thagard, P. 6, 35
- Theraulaz, T. 2
- Timmis, J. xviii, 2, 6, 7, 66, 67, 69, 70, 72, 73, 75
- Vapnik, V. N. 16
- Vargas, P. A. xviii, 2, 6, 7, 27, 28, 35, 37, 38, 49, 66, 67, 69, 70, 72, 73, 75, 99
- Veloso, M. 4
- Von Zuben, F. J. xviii, 2, 6, 7, 16, 66, 67, 69, 70, 72, 73, 75
- Von Zuben, F. J. (Eds.) 2
- Webb, A. 3
- Webb, E. 47–49, 52, 54, 56, 61
- Wilson, S. 35
- Wilson, S. W. 7, 35, 39, 41, 44, 46, 47, 49, 57

Wood, J. 21

Yamada, S. 46

Ziemke, T. 3

Zuben, F. J. Von 4, 35