

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Uma Proposta de Controle Neural
Adaptativo para a Navegação
de Veículos Autônomos**

por

JOELSON COELHO DA SILVA

Dissertação submetida à avaliação como requisito parcial para a obtenção
do grau de Mestre em Ciência da Computação

Prof. Dr. Paulo Martins Engel
Orientador

Porto Alegre, dezembro de 1999.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Silva, Joelson Coelho da

Uma Proposta de Controle Neural Adaptativo para a Navegação de Veículos Autônomos por Joelson Coelho da Silva – Porto Alegre: PPGC da UFRGS, 1999.

101 p.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 1999. Orientador: Engel, Paulo Martins.

1.Redes neurais. 2.Rede neural em modo de atenção. 3.Robótica móvel inteligente. 4.Controle de navegação. 5.Veículos autônomos. 6.Engel, Paulo Martins. II.Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profª Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. Franz Rainer Semmelmann

Diretor do Instituto de Informática: Prof. Dr. Philippe Olivier Alexandre Navaux

Coordenadora do CPGCC: Profa. Dra. Carla Maria Dal Sasso Freitas

Bibliotecária-Chefe do Instituto de Informática: Beatriz Bastos Haro

Agradecimentos

Tarefa difícil de ser realizada esta de colocar no papel a gratidão para aqueles que estiveram presentes nesta jornada. Inevitavelmente, algumas pessoas importantes são esquecidas, contudo estas são as personagens que estiveram presentes no cotidiano deste meu trabalho no Instituto.

Aos meus pais, pela luta, garra e determinação de propiciarem estudo aos filhos. Aos meus irmãos, pelo costumeiro incentivo e apoio.

Ao tio Hilton cujo auxílio foi essencial para o início da vida acadêmica.

À Lucia, pela compreensão e amor.

São inúmeros os amigos e amigas conquistados durante os anos de convivência no mestrado, em especial ao Edson, à Gilda, à Duda, à Renata, à Andréia, à Eunice, ao Felipe, ao Newton, ao Ivonei, ao Silvio, ao Bello, ao Cardon, ao Álvaro, ao Rodrigo e à Carla, ao Simões e à Eliane Padrão (por melhorar minha escrita).

A todos os funcionários do Instituto de Informática, em especial à Eliane (nossa mãe aqui no Instituto), à Margareth, à Cristina (Kika), ao Luiz Otávio, ao pessoal da biblioteca e da secretaria – sempre prestativos e sorridentes.

A todos os professores do Instituto, pelos conhecimentos e pelas lições de vida transmitidas, em especial aos professores Dr. Dante Barone, pelas rotineiras palavras de incentivo, e à inesquecível Profa. Dra. Carla Freitas, pela presteza e auxílio.

Ao Programa de Pós-Graduação em Computação do Instituto de Informática da Universidade Federal do Rio Grande do Sul, pela oportunidade.

Ao Prof. Dr. Paulo Martins Engel, pela amizade, paciência, estímulo e orientação.

Meu eterno muito obrigado a vocês todos.

“Tal é a ironia desses esforços feitos a fim de mudar-se a maneira de ver, para modificar o horizonte daquilo que se conhece e para tentar distanciar-se um pouco.

Levam eles, efetivamente, a pensar diferentemente?

Talvez tenham, no máximo, permitido pensar diferentemente o que já se pensava e perceber o que se faz segundo um ângulo diferente e sob uma luz mais nítida.

Acreditava-se tomar distância e no entanto fica-se na vertical de si mesmo.

A viagem rejuvenesce as coisas e envelhece a relação consigo.”

Michel Foucault

Sumário

Lista de Figuras.....	7
Lista de Tabelas	9
Resumo.....	10
Abstract	11
1 Introdução	12
1.1 Contextualização	12
1.2 Objetivos.....	14
1.3 Motivação	14
1.4 Estrutura da Dissertação	15
2 Sistemas Robóticos Inteligentes.....	17
2.1 Cibernética.....	18
2.2 Inteligência Artificial.....	20
2.3 Robótica Móvel Inteligente	21
2.4 O Controle dos Robôs	24
2.5 Sistemas de Controle Inteligentes.....	26
2.5.1 Sistemas Baseados em Regras	26
2.5.2 Sistemas de Algoritmos Genéticos	26
2.5.3 Sistemas de Lógica Nebulosa (Fuzzy Systems)	28
2.5.4 Modelos Neurocientíficos Abstratos	29
2.5.5 Sistemas Híbridos	30
2.6 O Controle da Navegação.....	30
2.6.1 Planejamento de Caminhos (Path Planning)	31
2.6.1.1 Mapas de Trajetórias (<i>Roadmaps</i>)	32
2.6.1.2 Decomposição Celular	33
2.6.1.3 Campos Potenciais	34
2.6.2 Navegação Baseada em Sensores	35
2.6.3 Percebendo o Ambiente.....	38
2.6.3.1 Contadores Passivos (<i>Dead Reckoning</i>)	39
2.6.3.2 Ultra-som (Sonar)	40
2.6.3.3 Visão Computacional.....	41
2.6.3.4 Exploradores Laser	41
3 Redes Neurais Artificiais (RNAs).....	43
3.1 Embasamento Biológico.....	43
3.2 Histórico das Redes Neurais Artificiais	45
3.3 O Neurônio Artificial.....	47
3.3.1 Função Degrau.....	48
3.3.2 Função Semi-Linear.....	48
3.3.3 Função Sigmoidal	49

3.4	Modelos de Redes Neurais Artificiais	50
3.4.1	Funcionamento das Redes FeedForward.....	51
4	A Rede Neural em Modo de Atenção (AMNN)	53
4.1	Introdução	53
4.2	A Definição do Problema	55
4.3	A Rede AMNN Proposta.....	59
4.4	O Aprendizado na Rede AMNN	61
4.5	A Atualização dos Pesos da Camada de Saída	62
4.5	A Atualização dos Pesos da Camada Oculta	64
4.6	Implementação do Modelo Proposto.....	65
5	Resultados Obtidos	66
5.1	Ambiente Livre de Obstáculos	66
5.1.1	Convergência da Rede AMNN em Ambiente Livre de Obstáculos.....	70
5.1.2	A Capacidade de Repetição da rede AMNN em Ambientes Livres de Obstáculos.....	71
5.1.3	A Estabilidade da Rede AMNN em Ambientes Livres de Obstáculos	72
5.2	Ambiente com Obstáculos Fixos.....	72
5.2.1	Convergência da Rede AMNN em Ambiente Contendo Obstáculos Fixos.....	79
5.2.2	A Capacidade de Repetição da rede AMNN em Ambientes Contendo Obstáculos Fixos	80
5.2.3	A Estabilidade da Rede AMNN em Ambientes Contendo Obstáculos Fixos.....	80
5.3	Ambiente com Obstáculos Móveis.....	81
6	Conclusão e Perspectivas Futuras.....	87
Anexo 1	O Robô Nomad 200.....	88
A1.1	O Sistema Mecânico	88
A1.2	Sistema de Sensoriamento	89
A1.2.1	O Sistema Táctil (Sensus 100).....	89
A1.2.2	O Sistema de Sonar (Sensus 200).....	90
A1.2.3	Sensores de Infravermelho (Sensus 300).....	90
A1.2.4	Sistema Básico de Visão (Sensus 400).....	92
A1.2.5	Sistema de Visão Estruturado (Sensus 500).....	92
A1.2.6	Sistema de Orientação (Bússola – Sensus 600).....	92
A1.3	O Sistema de Processamento	93
A1.4	O Sistema de Energia	93
Bibliografia		94

Lista de Figuras

FIGURA 1 –	Sistemas básicos de um robô.	17
FIGURA 2 –	O circuito da máquina especulativa de Walter.....	19
FIGURA 3 –	A tartaruga de Walter.....	19
FIGURA 4 –	O robô Shakey.....	23
FIGURA 5 –	O robô Hilare	23
FIGURA 6 –	O robô Stanford Cart.....	23
FIGURA 7 –	O robô CMU Rover.	23
FIGURA 8 –	Tipos de controle aplicado aos robôs [ARK 98].....	24
FIGURA 9 –	Os operadores genéticos da reprodução, intercâmbio (<i>crossover</i>) e mutação.	27
FIGURA 10 –	Sistema de controle por lógica fuzzy.	29
FIGURA 11 –	Grafo de visibilidade gerado a partir de um ambiente contendo três obstáculos poligonais.....	32
FIGURA 12 –	O diagrama de Voronoi de um determinado labirinto.	33
FIGURA 13 –	Decomposição celular exata.	33
FIGURA 14 –	Decomposição celular aproximada.	34
FIGURA 15 –	A árvore quaternária gerada pela decomposição celular aproximada de um ambiente exemplo.....	34
FIGURA 16 –	Dois exemplos de percursos usando o campo potencial.	35
FIGURA 17 –	A célula piramidal: um exemplo de neurônio.....	44
FIGURA 18 –	O neurônio artificial.	45
FIGURA 19 –	O perceptron.....	46
FIGURA 20 –	Gráficos das funções de ativação mais utilizadas.	49
FIGURA 21 –	Outros modelos de redes neurais.	50
FIGURA 22 –	A rede neural do tipo <i>backpropagation</i>	51
FIGURA 23 –	Diagrama do Funcionamento da Rede AMNN.....	54
FIGURA 24 –	O Nomad 200.....	54
FIGURA 25 –	A disposição dos sensores no robô virtual.....	55
FIGURA 26 –	Funcionamento do sensor de distância (SD).....	56
FIGURA 27 –	Funcionamento do Sensor de Distância Focalizado (SDF).	57
FIGURA 28 –	A geometria do robô móvel simulado.....	60
FIGURA 29 –	A arquitetura da rede AMNN proposta.....	60
FIGURA 30 –	Número de passos por taxa de aprendizado nas 50 simulações.....	68
FIGURA 31 –	Região gerada pelas 50 simulações sem obstáculos, pesos inicializados entre [-0,25;0,25] e taxa de aprendizado de 0,1.....	68
FIGURA 32 –	Região gerada pelas 50 simulações sem obstáculos, pesos inicializados entre [-1,0;1,0] e taxa de aprendizado de 0,25.....	69
FIGURA 33 –	Região gerada pelas 50 simulações sem obstáculos, pesos inicializados entre [-0,1;0,1] e taxa de aprendizado de 0,05.....	70
FIGURA 34 –	A Convergência da rede AMNN em ambiente livre de obstáculos.....	70
FIGURA 35 –	A repetitividade dos resultados com a rede AMNN em um ambiente livre de obstáculos.	71
FIGURA 36 –	Disposição dos obstáculos no ambiente para as simulações.....	72

FIGURA 37 – Região gerada pelas 50 execuções consecutivas com $\mu=0,15$ e os pesos inicializados com valores pertencendo ao intervalo $[-0,2;0,2]$	75
FIGURA 38 – Região gerada pelas 50 execuções consecutivas com $\mu=0,2$ e os pesos inicializados com valores pertencendo ao intervalo $[-1,0;1,0]$	75
FIGURA 39 – Exemplos da influência do alcance dos sensores na navegação.	76
FIGURA 40 – Simulações com alcance: $SD=4\mu m$ e $SDF=2\mu m$ e partindo de $(0,150)$.	77
FIGURA 41 – Simulações com alcance: $SD=2\mu m$ e $SDF=1\mu m$ e partindo de $(0,150)$.	77
FIGURA 42 – Simulações com alcance: $SD=4\mu m$ e $SDF=2\mu m$ e partindo de $(0,150)$.	77
FIGURA 43 – Simulações com alcance: $SD=2\mu m$ e $SDF=1\mu m$ e partindo de $(0,150)$.	78
FIGURA 44 – Exemplo da não convergência da rede AMNN partindo de $(0,150)$	78
FIGURA 45 – Um dos piores casos encontrados partindo de $(-180,170)$	79
FIGURA 45 – A repetitividade dos resultados com a rede AMNN em um ambiente contendo obstáculos fixos.	81
FIGURA 46 – Exemplo da ocorrência de um problema.	83
FIGURA 47 – Exemplo de uma execução onde o veículo móvel consegue atingir ao alvo.	84
FIGURA 48 – A 50 ^a simulação para os pesos iniciais no intervalo $[-0,2;0,2]$	84
FIGURA 49 – A 50 ^a execução para pesos iniciais no intervalo $[-0,4;0,4]$	85
FIGURA 50 – Exemplo da não convergência da rede com obstáculos móveis.	85
FIGURA A1-1 – As três rodas motrizes e direcionais do Nomad.	88
FIGURA A1-2 – O mapeamento dos sensores de toque.	89
FIGURA A1-3 – Os sensores de toque.	89
FIGURA A1-4 – Os sistema de ultra-som do Nomad.	90
FIGURA A1-5 – O sistema de infravermelho do Nomad.	91
FIGURA A1-6 – O gráfico da leitura dos sensores de infravermelho.	91
FIGURA A1-7 – O sistema de visão por laser e a formação da imagen do Nomad.	92
FIGURA A1-8 – A arquitetura do Nomad.	93

Lista de Tabelas

TABELA 1 – Resultados das simulações para o ambiente sem obstáculos.	67
TABELA 2 – Resultados das simulações para o ambiente com obstáculos fixos.....	74
TABELA 3 – Resultados das simulações para o ambiente com obstáculos móveis. ...	82

Resumo

Os equipamentos robóticos foram inicialmente criados para atuarem em ambientes industriais fechados. Com o passar do tempo, melhorias foram conquistadas. Atualmente, não se limitam mais à realização de tarefas simples e repetitivas em locais especialmente preparados.

Novos equipamentos, capazes de atuarem em ambientes abertos e de realizarem as mais diversas atividades, estão sendo desenvolvidos. Para tanto, é necessário que seus sistemas de controle realizem uma efetiva interação com o mundo onde estão inseridos. Fazem-se necessários, portanto, novos sistemas controladores com capacidade de uma contínua adaptação ao ambiente dinâmico onde operam.

As redes neurais artificiais, devido a sua capacidade de tratamento de problemas não lineares – matematicamente difíceis de serem resolvidos, estão sendo empregadas no controle destes processos.

O gerenciamento da trajetória de um veículo móvel em ambientes abertos ou fechados é um procedimento altamente não-linear, logo, a aplicação das redes neurais artificiais é bastante promissora.

Apesar de sua grande versatilidade, as redes neurais artificiais têm sido utilizadas apenas como sistemas de mapeamento. A grande maioria delas necessita de uma fase de treinamento para que possam armazenar a diversidade de estados possíveis do sistema. Quando atuam, elas simplesmente mapeiam os seus valores de entrada (estado atual) nas soluções previamente armazenadas.

Contudo, esta não é a melhor abordagem para os sistemas abertos, ou seja, para os processos cujas situações e possibilidades não podem ser totalmente enumeradas e que podem ser mutáveis no decorrer do tempo.

Este trabalho apresenta uma metodologia de controle neural adaptativo para guiar um veículo móvel até o seu destino em ambientes contendo obstáculos fixos ou móveis. Diferentemente das abordagens tradicionais, não existe a necessidade de um treinamento prévio da rede. A rede neural artificial escolhida promove uma contínua adaptação do sistema enquanto atua. Neste processo, são utilizados sensores que fornecem subsídios para que a rede possa gerar, adaptativamente, soluções parciais que façam com que o veículo autônomo se aproxime cada vez mais do seu objetivo, até, finalmente, atingi-lo.

Palavras-Chave: Rede Neural em Modo de Atenção – AMNN, redes neurais, controle da navegação em veículos autônomos, sistema de controle adaptativo, sistema de navegação inteligente, robótica móvel.

“Autonomous Vehicle Navigation Control: An Adaptive Neural Networks Proposal”

Abstract

The robotic equipments were created initially to actuate in closed industrial environments. Improvements have been achieved in this area. Nowadays, they are no longer limited to perform simple and repetitive tasks in controlled places.

New equipments, capable of acting in open environments and doing the most several activities, are being developed. For so much, it is necessary that its control systems accomplish an effective interaction with the world where they are inserted. Therefore, new systems controllers with capacity of a continuous adaptation to the dynamic environments are essential.

Artificial neural networks, due to their capacity of dealing with non-linear problems – mathematically difficult to be solved – are being used to control these kind of processes. Guide a mobile vehicle through an open or controlled environments is a highly non-linear procedure; therefore, the use of an artificial neural nets is quite promising.

In spite of its great versatility, they have just been used as mapping systems. Most of them need a training phase so that they can store the diversity of system's possible states. When they actuate, they simply map their input values (current state) to the solutions previously stored.

However, this is not the best approach for open systems, i.e. systems whose situations and possibilities cannot be totally enumerated and that can change in time.

This work presents an adaptive neural control methodology to guide a mobile vehicle to its target in environments with fixed or mobile obstacles. Differently from the traditional approaches, the need of a previous training phase of the neural network doesn't exist. The chosen model of artificial neural net promotes a continuous adaptation of the system while it actuates. Sensors are used to provide informations to the net. This way it generates partial solutions that makes the autonomous vehicle gets closer of its goal, until, finally, reach it.

Keywords: Attentional Mode Neural Network – AMNN, neural networks, adaptive controlling systems, autonomous vehicle navigation control, mobile robots.

1 Introdução

1.1 Contextualização

O Homem, desde o primórdio dos tempos, tem procurado criar ferramentas que o auxiliem na execução das tarefas diárias. Partindo do pressuposto que, segundo a Bíblia, Deus criou o primeiro ser humano a partir da terra [BOT 96], tem-se, desde então, procurado descobrir o segredo da vida e tentado desenvolver criaturas e mecanismos que imitem os seres humanos parcial ou completamente [HIL 88].

Algumas destas criações, datadas do século XVIII, tiveram bastante repercussão pública [WAL 93]; entre elas podemos citar um pato totalmente mecânico, desenvolvido por Jacques Vaucanson, cujas asas eram formadas por cerca de quatrocentas peças e que podia imitar uma enormidade de comportamentos comuns à espécie, como grasnar, nadar e comer.

Três outras “engenhocas” mecânicas realizavam tarefas mais humanas; foram criadas por Pierre e Henri-Louis Jaquet-Droz e batizadas de Escrivão (por ser capaz de escrever cartas), de Desenhista (pois desenhava) e o Músico (que tocava um pequeno órgão).

A maioria dos mecanismos desenvolvidos era usada unicamente para o entretenimento. Os inventores do século XIX não conseguiram ir mais longe do que isto, pois necessitavam de um tipo de máquina que pudesse ser mais “inteligente”, o que só foi possível com o advento dos computadores.

A invenção de equipamentos computacionais capazes de serem programados foi inspirada em uma criação datada também do século XIX. Foram desenvolvidos tendo por base o tear mecânico, inventado por Jacquard [MCK 91], que podia ser configurado para estampar diversas padronagens. Foi o início da programação de mecanismos.

Há algumas décadas, desencadeou-se um processo de robotização nos processos industriais, principalmente na indústria automobilística. O primeiro deles, chamado de Unimate, foi adquirido pela General Motors Corporation para realizar serviços de fundição em moldes. Era capaz de aprender, ou seja, armazenar em sua memória, uma série de até 180 passos ditados por um “professor” humano. Depois deste treinamento, podia repeti-los quantas vezes fossem necessários. Para realizar outra tarefa, efetuava-se um novo “aprendizado” – gravação de passos. Não passava de um braço mecânico capaz de memorizar uma seqüência de ações ditadas por um profissional experiente na execução da tarefa em questão.

Em meados de 1960, cientistas do MIT (Massachusetts Institute of Technology) e do SRI (Stanford Research Institute) começaram a realizar experiências com robôs que incorporavam inteligência artificial e câmeras de TV para guiar seus braços. Depois de muito trabalho conseguiram criar um robô que era capaz de montar a bomba d’água

de um automóvel escolhendo as peças dentre as espalhadas sobre uma bancada.

Os robôs passaram a ter credibilidade na década de 70, mas só a partir de 1980 começaram, realmente, a se estabelecer. A indústria automobilística, pioneira no processo de robotização, descobriu que eles poderiam substituir os humanos em muitas tarefas cansativas e perigosas, liberando-os para atuarem em trabalhos que necessitavam de inteligência. E este foi o mercado por muitos anos.

No mesmo período, outras empresas, como a IBM, a Westinghouse e a General Electric, partiram para a construção de seus próprios robôs industriais. Porém, esta automatização era puramente a substituição de operários por braços mecânicos e outros tipos máquinas que operavam em bases fixas, mais conhecido como automação rígida [TAN 94]. Modificações na organização, distribuição e seqüenciação das máquinas eram processos bastante dispendiosos.

Atualmente, os robôs continuam sendo extensivamente utilizados na fabricação e na montagem de automóveis. Os refinamentos dos projetos e as melhorias nas suas rotinas de inteligência, aliado ao uso de novos sensores mais precisos, estão fazendo com que possam ser empregados em tarefas que requeiram habilidades motoras mais exatas e delicadas, como a montagem de equipamentos eletro-eletrônicos.

A descoberta de que os robôs traziam vantagens perante às operações manuais – como o aumento da produtividade, a melhoria da qualidade e a conseqüente diminuição nos custos – naturalmente corroborava para aumentar os esforços no aperfeiçoamento e nas pesquisas de novos equipamentos. Primeiramente, era necessário flexibilizar os processos de automação e fazer com que os robôs pudessem se deslocar.

E assim foram criados os primeiros equipamentos que podiam se locomover em caminhos fixos, através de trilhos ou esteiras. Conseguiu-se, com isto, aumentar um pouco mais o raio de ação; porém o ambiente devia ser fechado, ou seja, os robôs não conseguiam detectar objetos e pessoas que pudessem estar no seu caminho, o que causou uma enormidade de acidentes.

Era necessário fazer com que pudessem se locomover livremente em qualquer tipo de superfície e que fossem capazes de perceber obstáculos em sua trajetória. Surge então, a robótica móvel, visando pesquisar dispositivos capazes de se deslocar e interagir com o ambiente.

A dinamicidade e o desconhecimento de certos ambientes [LAT 93] [MCK 91] [SEN 92] forçaram o desenvolvimento de novas tecnologias que se utilizam de paradigmas de inteligência (sistemas baseados em regras, autômatos de estados finitos, algoritmos genéticos, sistemas de lógica nebulosa – *fuzzy systems* – e redes neurais artificiais, entre outros).

Desta forma, surge uma nova área de pesquisa que visa tratar e solucionar, de preferência em tempo real, os problemas relativos ao controle, ao tratamento de leituras sensoriais, à adaptabilidade e ao aprendizado de robôs móveis, fazendo com que estes reajam às mudanças do ambiente [MON 94].

Este controle pode ser fornecido por sistemas adaptativos, porém estes necessitam de uma enorme quantidade de parâmetros para funcionarem como se deseja.

Este, aliado a outros fatores – como a adequação dos valores e dos parâmetros, complica consideravelmente tanto o modelamento como a implementação destes sistemas. Por outro lado, as redes neurais artificiais possuem a capacidade de adaptar-se às novas situações, através da modificação dos valores de seus parâmetros a partir do estado do sistema.

Alguns modelos de redes neurais artificiais necessitam de um treinamento prévio, outras não. Tornam-se uma boa opção quando o controle deve ser efetuado em ambientes onde não existe a previsibilidade e o total conhecimento dos acontecimentos que podem se suceder (sistemas abertos).

Este trabalho versa sobre o controle da trajetória de veículos autônomos através da utilização de redes neurais artificiais, mais especificamente da rede neural em modo de atenção – AMNN (*Attentional Mode Neural Network*) [ENG 96].

1.2 Objetivos

O presente trabalho propõe uma nova estratégia para o controle da navegação de robôs autônomos – ou *mobots* – baseado no conceito de adaptabilidade e reatividade através da utilização de uma rede neural artificial.

Esta abordagem permite que o controle da navegação seja feito em tempo real em ambientes sem conhecimento prévio, variáveis no tempo. O sistema se adapta às novas situações encontradas no ambiente através do tratamento das leituras dos sensores.

Nenhum mapeamento do ambiente é realizado durante a navegação. Simplesmente são feitas leituras do ambiente para detectar a presença dos obstáculos e, uma vez localizados, contorná-los objetivando alcançar a posição de destino.

Alguns trabalhos já foram propostos neste sentido. A particularidade deste baseia-se na arquitetura utilizada para o controle: uma única rede neural artificial do tipo AMNN.

1.3 Motivação

A curiosidade e o fato de ser uma área com inúmeras possibilidades de aplicação tem chamado a atenção e atraído novos pesquisadores, principalmente porque as soluções até agora descobertas não são, ainda, definitivas. A miniaturização cada vez maior de componentes e de computadores tem contribuído para a descoberta de novas aplicações para os robôs móveis.

Outro fator que contribui muito para o avanço das pesquisas na área diz respeito aos problemas de segurança para seres humanos em determinadas situações, como por exemplo:

- nas usinas nucleares: o transporte e o manuseio de materiais radioativos é

extremamente perigoso.

- na exploração submarina e espacial [WIL 93], ambientes altamente insalubres e hostis;
- na agricultura e em laboratórios: trabalhos cansativos, tediosos e repetitivos.

Tudo isto incentiva a continuidade do trabalho [MCK 91], principalmente da robótica móvel e em sistemas de controle [CAU 91] [MAE 94] [TIL 93] [RUD 93] [UCK 94].

O refinamento de certas máquinas só foi possível em decorrência do desenvolvimento de outras áreas, como a evolução dos computadores e dos mecanismos sensoriais e motores.

Acima de tudo, tem-se procurado criar mecanismos que facilitem a vida humana no cotidiano, que possam baratear os custos de produção e ajudar o homem a desvendar os segredos ainda não revelados, tanto em nosso planeta como no universo.

Felizmente o progresso tem sido bastante rápido, como pode ser constatado nos avanços obtidos nos robôs que participam anualmente da competição promovida pela Associação Americana para a Inteligência Artificial (AAAI) desde 1992, onde já foram apresentados desde robôs que recolhem latas de refrigerantes até outros que servem de guias turísticos [ARK 98].

O presente trabalho visa contribuir com uma nova abordagem na utilização das redes neurais artificiais para o controle de sistemas em tempo real, mais especificamente no tratamento da robótica móvel inteligente, no Instituto de Informática da Universidade Federal do Rio Grande do Sul. Este estudo poderá viabilizar uma maior aproximação entre a universidade e as empresas interessadas na aplicação desta tecnologia.

1.4 Estrutura da Dissertação

O capítulo 2 apresenta um histórico evolutivo dos sistemas robóticos inteligentes. Também são mostrados os sistemas de controle inteligentes, as metodologias para o controle da navegação e a utilização de sensores.

No capítulo 3 é mostrado um histórico e os modelos das redes neurais artificiais, bem como as funções de ativação mais utilizadas.

A descrição da rede neural em modo de atenção – AMNN se dá no capítulo 4. Nele, também é definido o problema e como a sua solução será implementada.

O capítulo 5 relata e comenta os resultados obtidos nos testes para o controle da trajetória de um veículo móvel em ambientes livres de obstáculos e com obstáculos fixos e móveis.

Para concluir, no capítulo 6, são apresentadas as considerações finais e as

perspectivas futuras.

O apêndice A fornece algumas especificações técnicas do hardware do robô móvel Nomad 200, adquirido recentemente pela Universidade Federal do Rio Grande do Sul.

2 Sistemas Robóticos Inteligentes

Antes de começar, é necessário fazer uma definição atual do que é um robô. Segundo a Associação de Robótica Industrial (RIA), “robô é um manipulador multifuncional e reprogramável projetado para movimentar materiais, peças, ferramentas ou dispositivos especializados através de movimentos variáveis e programados, visando executar uma grande diversidade de tarefas” [JAB 85].

Os robôs fixos já estão consolidados em quase todas as indústrias atuais [WAL 91]. Apesar de existirem ainda alguns problemas a serem solucionados, eles levam consideráveis vantagens quando comparados aos operários humanos. Estas vantagens podem ser identificadas como um aumento da produtividade e a execução ininterrupta das atividades.

Cada vez se investe mais em pesquisas visando descobrir meios que o ajudem e facilitem a vida diária, principalmente na realização de tarefas em ambientes nocivos ao homem.

Basicamente os robôs são formados por três sistemas, como pode ser visto na figura 1, que serão descritos logo a seguir.

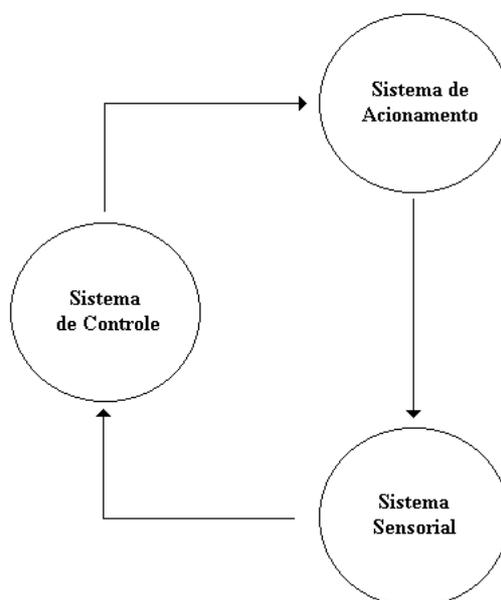


FIGURA 1 – Sistemas básicos de um robô.

O sistema de controle gerencia todo o funcionamento e o comportamento do robô. Pode realizar tarefas de mapeamento, planejamento e controle da atuação do agente robótico [MAI 93]. Pode ser implementado de várias formas, desde os simples limitadores mecânicos (rígidos ou flexíveis), até o uso de avançados sistemas de processamento de dados.

Alguns sistemas de controle podem utilizar técnicas inteligentes na avaliação

dos dados obtidos através dos sensores visando a tomada de decisões. Podem ainda incluir técnicas de aprendizado, buscando incorporar novas informações importantes em sua base de conhecimento [KRO 95] [HOR 93].

Grande parte das pesquisas em robótica móvel utiliza sistemas de controle inteligentes, através de redes neurais artificiais, de algoritmos genéticos [SCH 97a], de sistemas especialistas e de lógica nebulosa, ou uma junção destes, para gerenciar e atuar de forma mais natural no ambiente.

O sistema sensorial, ou de percepção, é o responsável por capturar informações a respeito da situação atual do próprio robô e também do ambiente a sua volta. Sendo este tipo de sistema opcional, quando este não está inserido no projeto, faz com que o agente não tenha um retorno das conseqüências de suas ações [SEN 92]. Vários são os tipos de sensores usados atualmente [MCK 91], entre eles encontram-se os de infravermelho, de ultra-som, os acústicos e as câmaras digitais.

O último integrante de um agente robótico é o sistema de acionamento. São os atuadores, mecanismos que transformam a energia elétrica, pneumática ou hidráulica em movimentos visando desempenhar as tarefas necessárias. Os motores elétricos estão entre os mais utilizados.

O sistema de acionamento visa fornecer a energia cinética necessária para os dispositivos de deslocamento e efetadores (braços, garras, etc.). Entre os mecanismos de deslocamento mais utilizados estão as rodas, as esteiras, as hélices [MIL 93] [MCK 91] e os conjuntos mecânicos que atuam como se fossem pernas [WAL 93].

2.1 Cibernética

A cibernética, cujo princípio é creditado a Norbert Wiener na década de 40, é a junção da teoria do controle, da informática e da biologia para explicar os princípios comuns de controle e de comunicação, tanto em animais quanto em máquinas [WEI 48]

Em 1952, Wiener e Ashby criaram uma máquina que expressava um comportamento natural baseada em uma matemática desenvolvida para a realimentação em sistemas de controle. Em 1953, W. Grey Walter criou um mecanismo precursor da robótica, batizado de Máquina Especulativa (*Machina Speculatrix*), porém este tornou-se mais conhecido como a tartaruga de Grey Walter.

Alguns dos princípios utilizados na sua construção incluem:

- **parcimônia:** preferência pela simplicidade (reflexos simples podem servir de base para o comportamento); segundo ele, “as variações de padrões de comportamento exibidas são complexas e até mesmo impossíveis de prever, mesmo que se economize na estrutura” [WAL 63];
- **exploração ou especulação:** o sistema nunca fica parado, exceto quando está recarregando suas baterias, um comportamento que é considerado adequado e normal dentro de certas circunstâncias. Um exemplo seria evitar que seja apanhado; na exploração de qualquer lugar, inevitavelmente

encontra muitos obstáculos, mas excetuando-se os degraus das escadas e os tapetes, poucas são as situações das quais não pode se livrar por sua própria conta
[WAL 63];

- **atração:** o sistema é motivado a seguir em direção a um determinado objeto ou objetivo (no caso da tartaruga de Walter, procurar uma fonte de luz);
- **aversão:** o sistema procura afastar-se de certos estímulos considerados negativos, como por exemplo os obstáculos;
- **discernimento:** o sistema tem a habilidade de distinguir entre um comportamento produtivo e um improdutivo, adaptando-se conforme a situação.

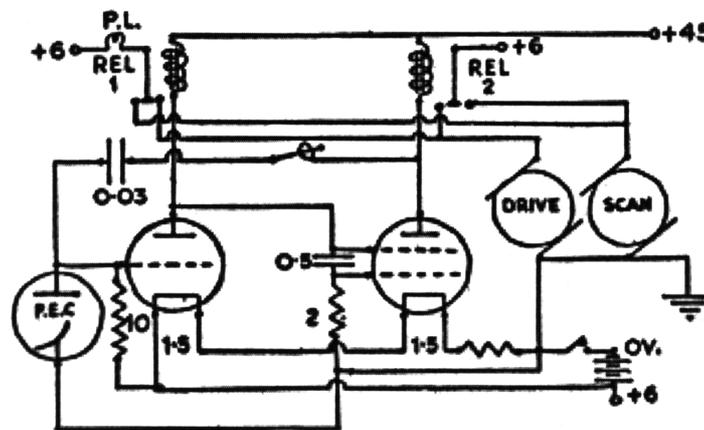


FIGURA 2 – O circuito da máquina especulativa de Walter.

Tal máquina, construída como um dispositivo analógico (ver diagrama na figura 2), era constituída de dois sensores (uma fotocélula e outro de colisão) – para “perceber” o ambiente, dois atuadores (um motor para o deslocamento e outro para dar direção ao movimento) e duas “células nervosas” (duas válvulas). A fotocélula está sempre apontada na mesma direção do movimento da tartaruga, logo podia explorar o ambiente (ver figura 3).

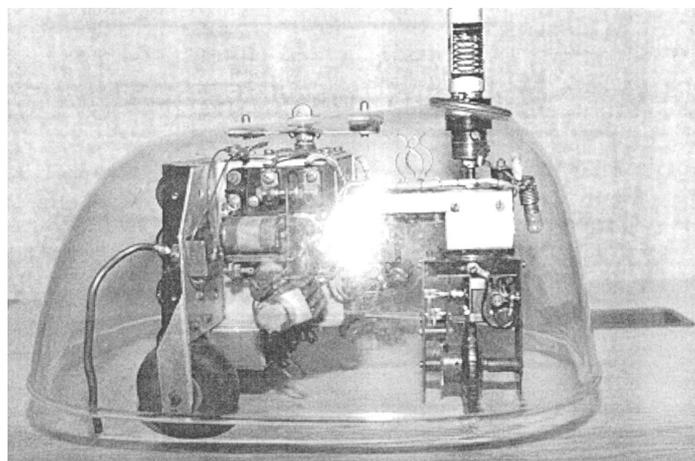


FIGURA 3 – A tartaruga de Walter.

Exibia os seguintes comportamentos:

- procurar a luz: o sensor girava até que uma fonte fraca de luz fosse detectada ao mesmo tempo que se deslocava, explorando, dessa forma, o ambiente;
- ir em direção a uma luz fraca: quando esta era detectada, a tartaruga movia-se em sua direção;
- evitar luzes intensas: um comportamento de aversão a repelia das fontes de grande luminosidade;
- bater e girar: usado para evitar obstáculos (sensor de colisão se sobrepõe ao da luz);
- recarregar a bateria: quando a bateria interna estava com pouca carga, a tartaruga percebia a luz forte (a estação de recarga da bateria tinha uma luz intensa situada sobre ela) como sendo fraca e movia-se em sua direção e estacionava. Após a recarga, o comportamento de aversão retornava, pois a fonte de luz voltava a ser reconhecida como intensa.

Estas mesmas idéias foram recentemente utilizadas como base para a criação de sistemas robóticos adaptativos baseados em comportamento [AGA 97].

2.2 Inteligência Artificial

O nascimento da inteligência artificial como um campo distinto é geralmente associado à Conferência de Verão sobre Pesquisas em Dartmouth - Estados Unidos, em agosto de 1955. Os objetivos desta conferência eram a discussão de temas como uso da linguagem, redes neurais artificiais, teoria da complexidade, auto-aprendizado, abstrações e criatividade, dentre muitos outros.

Na proposta original [MCC 55], Marvin Minsky indicava que *uma máquina inteligente deveria tentar construir ela própria um modelo abstrato do ambiente onde estava inserida e, uma vez deparada com um problema, deveria primeiramente tentar explorar soluções neste ambiente abstrato interno para depois aplicar no ambiente externo.*

Este tipo de abordagem dominou as pesquisas em robótica pelos 30 anos seguintes, período em que a pesquisa em inteligência artificial desenvolveu uma grande dependência da representação do conhecimento e dos métodos de raciocínio para o planejamento robótico. Naquela época, a organização hierárquica para o planejamento de uma tarefa tornou-se bastante popular, pois afirmavam que “um plano era qualquer processo hierárquico no organismo que pode controlar a ordem na qual uma sucessão de operações é executada” [MIL 60].

Alguns dos exemplos mais conhecidos no planejamento tradicional da inteligência artificial incluem os STRIPS [FIK 71], ABSTRIPS [SAC 74], HACKER [SUS 75] e NOAH [SAC 75].

A metodologia clássica da inteligência artificial tem duas características importantes [BOD 95]:

- capacidade de representar estruturas hierárquicas através da abstração;
- uso de um “grande” conhecimento que utiliza asserções simbólicas explícitas para representar o mundo.

Porém, a robótica baseada em comportamento vai contra estes princípios. *Planning is just a way of avoiding figuring out what to do next* foi a definição dada por Brooks [BRO 87]. Apesar da resistência inicial, a noção de sentir e agir dentro do ambiente passou a predominar nas pesquisas relacionadas à robótica inteligente, preterindo-se o enfoque baseado na representação do conhecimento e do planejamento.

Os avanços, tanto na robótica como nos dispositivos sensoriais, têm contribuído para testar as hipóteses geradas pela comunidade que pesquisa a robótica. A computação distribuída também tem cooperado muito para novas descobertas, possibilitando a divisão do processamento através da utilização dos multi-agentes [MIN 86] que, para muitos, é a base para a inteligência¹. Brooks comprovou através da arquitetura por ele proposta, chamada de *subsumption architecture*, que, a partir da interação coordenada e correta de agentes simples, poderia emergir um comportamento inteligente complexo [WAL 93].

2.3 Robótica Móvel Inteligente

Como pôde ser notada, a primeira definição de robô, apresentada no início do capítulo, excluiu os robôs móveis. Uma segunda definição descreve os robôs como sendo a conexão inteligente entre a percepção e a ação [BRA 95]. Esta, por sua vez, parece por demais inclusiva, mas reconhece a importância destes dois ingredientes (perceber e reagir) e a sua relação, indispensável, aos sistemas robóticos.

Uma definição mais atualizada e contextualizada é dada por [ARK 98]: *um robô inteligente é uma máquina capaz de obter informações a respeito do seu ambiente e usar o conhecimento sobre o seu mundo para mover-se com segurança de maneira intencional e propositadamente*. O que nos remete a uma nova área: a robótica móvel.

Os robôs móveis podem ser subdivididos em três categorias:

- **não autônomos**: sem nenhum controle inteligente; o seu comportamento é fixo e previamente programado; necessita do auxílio constante de um “professor” para ensinar-lhe a seqüência correta de movimentos que deverão ser executadas. São conhecidos, também, como automatizados (*Automated Guided Vehicles* – AGVs) [MCK 91];
- **semi-autônomos**: que possuem algum tipo de inteligência em seu mecanismo de controle, mas este não é suficiente para garantir um

¹ Inteligência: “habilidade de aprender ou compreender a partir da experiência; habilidade de adquirir e reter conhecimento”

comportamento adaptativo e autoprogramado;

- **totalmente autônomos:** aqueles que possuem um controle totalmente inteligente, capaz de gerar novos comportamentos em decorrência da percepção de modificações no ambiente, sem qualquer ajuda externa.

Muitas discussões aparecem na literatura sobre a correta definição de agentes autônomos [UCK 94] [BEL 94]. Todos possuem em comum nas suas definições que os agentes autônomos devem ter a capacidade de descobrir soluções para os problemas encontrados no ambiente onde atuam, sem nenhuma supervisão externa [FLO 94].

Na robótica, a palavra “inteligente” é um sinônimo para a capacidade que um certo agente possui de interagir e de se adaptar às condições do ambiente que o cerca (comportamento adaptativo). Para alguns, esta é a característica principal de um comportamento inteligente [BEE 94]. Somente a partir desta interação é que um agente torna-se capaz de desenvolver e aumentar as suas habilidades conforme as suas necessidades. Grande parte do conhecimento a respeito do comportamento adaptativo deve-se à etiologia, que é o estudo e a observação dos animais em seu habitat natural.

Para conduzir experimentos em robótica, robôs são necessários. Ao se trabalhar somente com simulações, muitas vezes ignoram-se questões óbvias, uma vez que é muito difícil gerar todas as possibilidades de acontecimentos e situações. No entanto, a utilização de simuladores traz algumas vantagens. A principal delas é que os erros gerados nas simulações não causam danos materiais. Robôs podem ser complexos para construir e difíceis de manter. Muitos foram os robôs precursores, mas com certeza devemos muito do progresso obtido até agora na robótica móvel a alguns deles, em especial a estes quatro:

- **SHAKY:** construído no início da década de 60 no Stanford Research Institute (SRI) [ARK 98], habitava um mundo artificial, na verdade um escritório com objetos especialmente coloridos e projetados para facilitar o reconhecimento através da visão. Planejava uma ação, como por exemplo: mover um objeto de um lugar para outro, e a executava. Possui dois motores de passos independentes, uma câmera com controles motorizados instalada no topo do robô e sensores de colisão na periferia (figura 4).
- **HILARE:** seu projeto começou em 1977 no Laboratório de Automação e de Análise de Sistemas (Toulouse-França) [ARK 98]. Utilizava três rodas, duas motrizes e outra de apoio (ver figura 5). Pesava cerca de 400kg. Seus sensores eram uma câmera de vídeo, quatorze sensores ultra-sônicos e um medidor de distâncias a laser. O mais interessante sobre ele foi a sua longevidade; estava em uso em experimentações mesmo depois de uma década de sua criação [NOR 89] [NOR 95].
- **Stanford Cart:** era uma plataforma robótica mínima usada por Moravec [MOR 77] simplesmente para testar a visão estérea na navegação [ARK 98] (figura 6). Porém era muito vagaroso, conseguia andar um metro a cada 10-15 minutos; levava cerca de 5 horas para efetivar um movimento completo. O que consumia o maior tempo de processamento era o tratamento dos dados visuais. Mesmo assim, conseguiu navegar por um ambiente um pouco complexo, totalizando 20 metros de percurso, evitando os obstáculos à

medida que iam sendo detectados.

- **CMU Rover:** na década de 80, Moravec foi para a Carnegie Mellon University (CMU), onde desenvolveu esforços para construir um robô menor, cilíndrico, com três rodas motrizes e direcionais, carregando uma câmera de vídeo e também sensores de infravermelho e ultra-som (ver figura 7). Este robô foi o precursor de muitos outros criados naquela entidade.

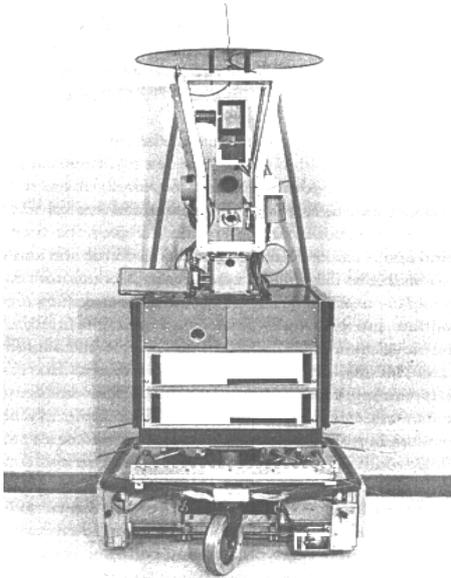


FIGURA 4 – O robô Shakey.

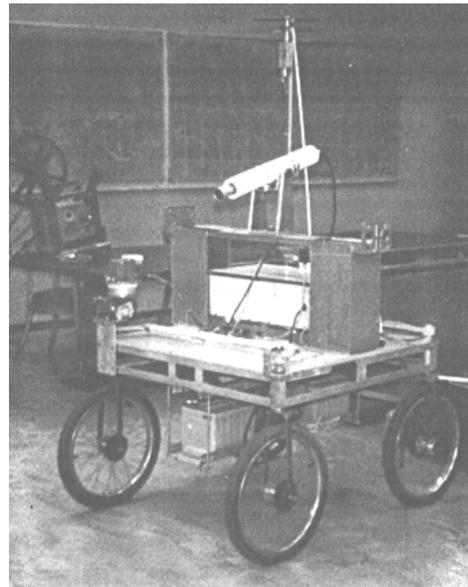


FIGURA 6 – O robô Stanford Cart

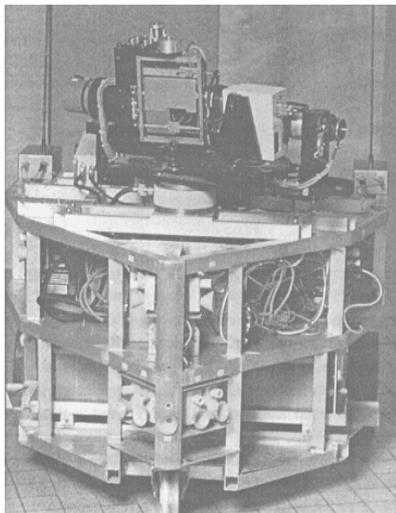


FIGURA 5 – O robô Hilare

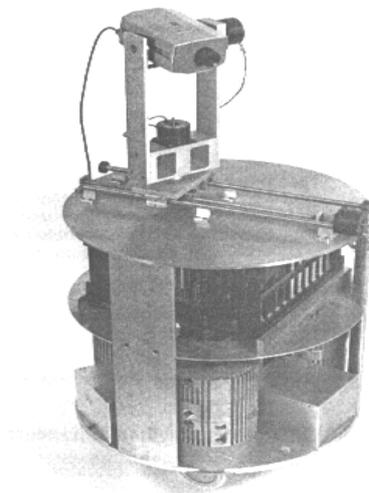


FIGURA 7 – O robô CMU Rover.

2.4 O Controle dos Robôs

Muitas técnicas e modos de controle robótico foram desenvolvidas. A figura 8 apresenta as estratégias de controle que estão sendo mais utilizadas atualmente. O lado esquerdo apresenta os métodos que aplicam conhecimento premeditado e o direito mostra os sistemas de controle reativos.

Um robô que utiliza sistema deliberativo de controle necessita ter um conhecimento relativamente completo sobre o ambiente e usa-o para definir suas futuras ações, permitindo que possa otimizar o seu desempenho. Porém, esta técnica requer que as suposições acerca deste mundo sejam consistentes, confiáveis e corretas. Se a informação utilizada não for confiável, ou se esta sofreu alteração desde que foi obtida, o resultado pode estar seriamente equivocado.

Deliberativo	Reativo
←	→
Puramente simbólico	Reflexivo
→	
Velocidade das respostas	
←	
Capacidade Preditiva	
←	
Dependência da exatidão e da integralidade do modelo do ambiente	
Representação dependente	Representação livre
Respostas mais demoradas	Respostas em tempo real
Alto nível de inteligência (cognitiva)	Inteligência de baixo nível
Latência variável	Operações computacionais simples

FIGURA 8 – Tipos de controle aplicado aos robôs [ARK 98].

Em ambientes abertos ou dinâmicos, onde objetos podem estar se movendo arbitrariamente, é potencialmente perigoso basear-se em informações sobre o passado, que podem não mais ser verdadeiras ou confiáveis para determinar uma ação.

Normalmente os modelos de representação do ambiente se baseiam em conhecimentos prévios e também em dados advindos de sensores. Possuem algumas características entre elas:

- uma estrutura hierarquizada com subdivisões funcionais claramente identificáveis, similar às organizações comerciais ou ao comando militar;
- a comunicação e o controle ocorrem de maneira premeditada e predeterminada, seguindo o sentido vertical da organização, com pouco ou nenhum movimento lateral;

- os altos níveis da hierarquia provêm sub-metas para os níveis inferiores;
- no escopo do planejamento, tanto o espacial quanto o temporal, se alteram durante o trajeto descendente na hierarquia. Nos níveis mais inferiores as necessidades temporais são menores e as espaciais são maiores;
- recaem fortemente nos modelos de representação simbólica do ambiente.

O controle hierárquico pode ser recomendado para ambientes bem estruturados e altamente previsíveis (ambiente fabril). Os sistemas reativos, por outro lado, foram desenvolvidos como uma resposta a algumas das desvantagens aparentemente associadas com o paradigma do projeto hierárquico.

De maneira simples, o controle reativo (mostrado no lado direito da figura 8) é uma técnica capaz de juntar firmemente a percepção à ação, tipicamente no contexto do comportamento motorizado, para produzir respostas robóticas temporizadas em ambientes dinâmicos e desestruturados. Faz-se necessário, então, definir:

- **comportamento individual:** um par estímulo/resposta para uma determinada configuração do ambiente que é modulada pela atenção e determinada pela intenção;
- **atenção:** prioriza tarefas e focaliza os recursos sensoriais; é determinado pelo contexto atual do ambiente;
- **intenção:** determina qual o conjunto de comportamentos que deve estar ativo, baseado nos objetivos e metas do agente robótico;
- **comportamento definido ou emergente:** o comportamento global do robô ou do organismo é uma consequência da interação dos comportamentos ativos individuais;
- **comportamento reflexivo (alternativamente, puramente reativo):** comportamento que é gerado por comportamentos reativos fortemente ligados, através de rígidas conexões entre sensores e atuadores, onde as informações sensoriais não são persistentes e nenhum modelo do ambiente é utilizado.

Alguns aspectos chaves desta metodologia baseada em comportamento incluem [BRO 91]:

- **situacionismo:** o robô é uma entidade situada e cercada pelo mundo real. Ele não opera sobre representações abstratas da realidade, mas exatamente sobre ela;
- **personalismo:** o robô possui uma presença física (um corpo). Esta realidade espacial gera consequências em sua interação dinâmica com o mundo que não pode ser fielmente simulada;
- **evolucionismo:** a inteligência surge a partir das interações do agente robótico com o seu ambiente. Não é uma característica isolada do ambiente

ou do robô, mas um resultado da interação destes.

2.5 Sistemas de Controle Inteligentes

Os comportamentos inteligentes podem ser obtidos utilizando-se sistemas inteligentes artificiais, tanto a inteligência simbólica quanto a conexionista [MYE 92], podem ser utilizadas nas três sub-tarefas constituintes do controle robótico (mapeamento, planejamento e execução). Alguns sistemas utilizam modelos de comportamento gerenciados por mecanismos inteligentes. Dentre muitos, podemos citar os controlados por lógica nebulosa (*fuzzy*) [FUK 94] [ZIM 94] [MON 94], por redes neurais artificiais [BAL 91] [PIM 92] [NAG 90] [WIL 93] [KOH 94] e por algoritmos genéticos [KOZ 94] [BEL 94] [NOL 94] [SCH 97a].

2.5.1 Sistemas Baseados em Regras

Um sistema baseado em regras é, provavelmente, o modo mais fácil de representar conhecimento em um computador. Na maioria das linguagens de programação, a implementação destes sistemas é feita através das instruções SE-ENTÃO, que combinam condições específicas às ações determinadas. As regras são do estilo “SE o robô detectar um obstáculo próximo e na sua frente ENTÃO gire 15 graus para a direita”. Usando várias regras desse tipo, um programa de computador pode representar um enorme número de condutas que imitam o comportamento animal.

É necessário um especialista para projetar o sistema. Ele vai procurar definir todas as situações passíveis de ocorrência. Logo, o desempenho do sistema está intimamente ligado ao número de regras geradas pelo especialista e, quanto maior for o número de regras, mais complexo e demorado tornam-se a depuração e os testes. O grande problema dos sistemas baseados em regras é que não é possível aprender novos comportamentos [RAB 95].

Possuem um desempenho que pode ser considerado bom, quando em manipulações simbólicas [WIN 87], o que não ocorre quando da utilização de dados sensorizados, pois existe a necessidade de transformação destes em símbolos [MON 94].

2.5.2 Sistemas de Algoritmos Genéticos

Os algoritmos genéticos (AG's) [GOL 89] formam uma classe de métodos no qual uma solução de alta qualidade é encontrada aplicando-se operadores biologicamente inspirados sobre pontos individuais pertencentes ao espaço de busca, criando novas gerações de indivíduos melhor adaptados à solução do problema (evolução com o passar do tempo). A aplicação de AG é muito variada, podendo ser utilizado no controle da trajetória de veículos [BAR 96] [SCH 97] [SCH 97a], em previsão de séries temporais [MAR 96] e em problemas de otimização [RAM 94].

A aptidão de cada indivíduo da população (o conjunto de pontos no espaço de procura) é calculada através da aplicação de uma função de avaliação – chamada de função de aptidão – que informa o grau de habilidade do indivíduo para a realização da tarefa. Os melhores elementos da população são recompensados de acordo com as suas aptidões, enquanto que os menos habilitados são penalizados ou então extintos. À medida que as gerações vão sendo criadas, progride também a qualidade da solução do problema. Mesmo assim os AG's não garantem uma solução ótima global; em geral certamente produzem soluções de alta qualidade dentro de um determinado espaço-tempo do problema.

Algoritmos genéticos normalmente requerem representações de conhecimento especializadas (codificações) visando facilitar a aplicação dos operadores. As codificações tipicamente tomam a forma de *strings* binárias, dependentes da posição de cada bit na *string*, onde cada bit representa um gene e toda a *string* é o cromossomo.

A população inicial (um conjunto representável de *strings* binárias) é, de alguma forma, estabelecida. Geralmente através da inicialização aleatória. Os operadores são então aplicados aos cromossomos de cada elemento da população. Os operadores mais utilizados são a reprodução, o intercâmbio de genes (*crossover*) e a mutação (figura 9).

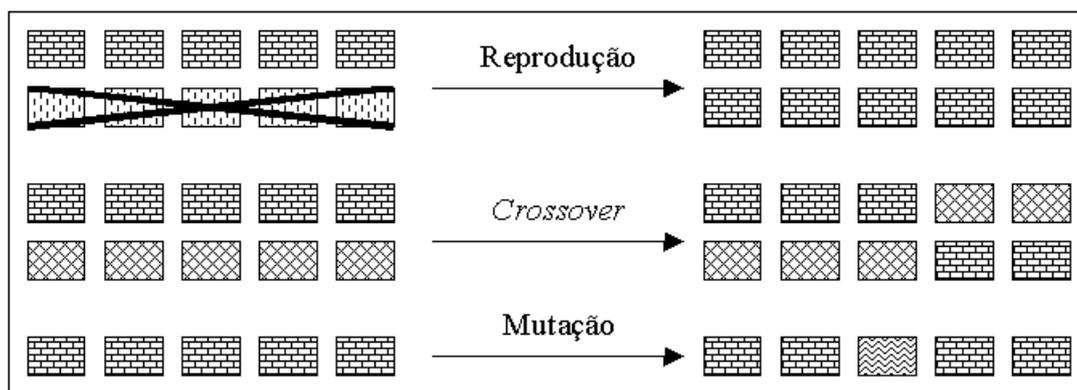


FIGURA 9 – Os operadores genéticos da reprodução, intercâmbio (*crossover*) e mutação.

Antes da aplicação do operador, a habilidade de cada indivíduo é calculada usando a função de aptidão. Utilizando o operador de reprodução, os indivíduos mais aptos são copiados para a próxima geração, substituindo os menos habilitados. Normalmente isto é feito, probabilisticamente, através do uso da técnica da roleta de pesos, aumentando a possibilidade, mas não garantindo a reprodução dos indivíduos melhores adaptados, seguindo o princípio de Darwin da sobrevivência dos mais habilitados.

Crossover envolve dois cromossomos individuais e é a troca de informações genéticas pela transferência de uma determinada quantidade de genes de um elemento para o outro. Este processo cria novos indivíduos que podem, ou não, ser mais aptos para a resolução do problema que os seus pais. Normalmente a seleção dos indivíduos, a quantidade e a especificação de quais genes serão intercambiados são escolhidos aleatoriamente. O efeito gerado é uma maior diversidade dos elementos na população.

Mutação é a simples inversão aleatória de valores genéticos. É aplicado individualmente aos indivíduos e não aumenta o tamanho da população. Este operador

possibilita uma maneira de escapar dos mínimos locais. Da mesma forma que em mutações biológicas, a maioria das mutações produzirá indivíduos inferiores, mas, ocasionalmente, um mais ajustado emergirá. Como a probabilidade associada à mutação é geralmente muito baixa e a quantidade de indivíduos mais aptos na população é maior (devido à reprodução), a aplicação deste operador geralmente gera soluções de alta qualidade, inacessíveis por outros métodos.

A aplicação destes operadores genéticos propicia, com o passar do tempo, o surgimento de uma grande variedade de elementos na população. Alguns indivíduos gerados possuem uma aptidão inferior a de seus pais. Mas a aptidão média da população (e também dos melhores ajustados) aumenta à medida que novas gerações vão sendo criadas.

Se for bem projetado, o sistema de aprendizado conduzirá à obtenção de indivíduos ótimos, com muitos genes comuns entre si. A qualidade da solução final e o tempo dispendido para a sua obtenção, dependem muito da natureza do problema a ser resolvido e também dos parâmetros de configuração do AG.

2.5.3 Sistemas de Lógica Nebulosa (*Fuzzy Systems*)

Os sistemas de controle por lógica *fuzzy* produzem ações usando um conjunto de regras baseado em lógica nebulosa. Esta foi a maneira encontrada, baseada na teoria de conjuntos e na teoria das possibilidades [ZAD 83] [ZAD 92], para possibilitar o tratamento das ambigüidades presentes no mundo. Tradicionalmente, os computadores trabalham com valores lógicos absolutos – verdadeiro (1) ou falso (0), sem nenhum outro valor entre eles. Por isto são chamados de valores *crisp*.

Já a lógica nebulosa fornece uma outra perspectiva, permitindo que as variáveis assumam valores determinados por uma função de pertinência, que avalia o quanto uma variável pertence a este ou àquele conjunto. Em *fuzzy*, estas variáveis são referenciadas como variáveis linguísticas. Estas possuem valores linguísticos predefinidos (exemplo: rápido, lento, grande, pequeno, gordo, magro, longe, perto, etc.).

Nos laboratórios do SRI, foi projetado um controlador reativo baseado em regras *fuzzy* para a navegação do robô Flakey [SAF 93]. O robô Marge, desenvolvido por [GOO 94] na Universidade da Carolina do Norte, foi o campeão em 1995 da competição entre robôs organizada anualmente pela AAAI.

Temos alguns trabalhos produzidos no Brasil que utilizam lógica *fuzzy* para o controle de veículos autônomos. Em [BOT 96], é tratado o problema de guiar um robô móvel até um objetivo em um ambiente desconhecido utilizando sistema de controle por lógica nebulosa. O sistema controla a velocidade e o ângulo direcional ao mesmo tempo em que o robô está se movendo.

Um sistema de controle baseado em lógica *fuzzy* consiste do seguinte (figura 10):

- **“fuzzificador”**: responsável pelo mapeamento dos valores exatos lidos dos

sensores em uma coleção de conjuntos de entrada em lógica nebulosa;

- **base de regras *fuzzy***: base de coleções de regras do tipo SE-ENTÃO;
- **mecanismo de inferência *fuzzy***: responsável pelo mapeamento de conjuntos *fuzzy* em outros conjuntos *fuzzy* de acordo com as regras e os valores de pertinência;
- **“defuzzificador”**: efetua o mapeamento do conjunto de valores *fuzzy* gerados na saída do sistema para um conjunto de valores exatos utilizado para comandar o atuador.

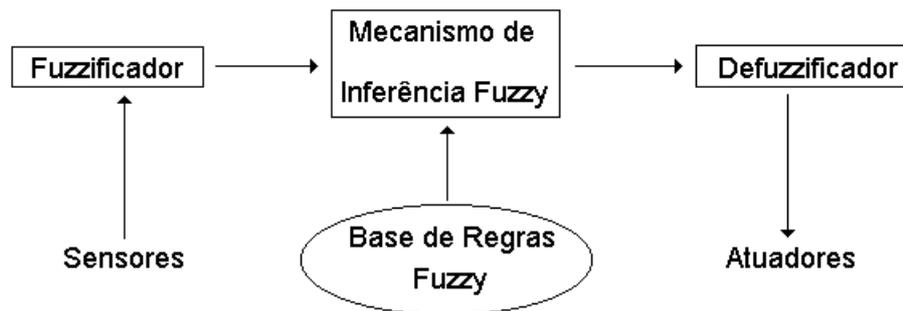


FIGURA 10 – Sistema de controle por lógica fuzzy.

2.5.4 Modelos Neurocientíficos Abstratos

Infelizmente, o conhecimento que temos a respeito do cérebro e do seu funcionamento ainda é extremamente superficial. A neurociência está progredindo em passos rápidos à medida que novas ferramentas para compreensão do funcionamento do cérebro tornam-se disponíveis.

Mesmo assim tem sido dito que, mesmo que tivéssemos um mapa completo da estrutura cerebral (todos os seus neurônios e suas interconexões), nossa compreensão ainda seria inadequada. A atividade cerebral sobre o substrato neural é altamente dinâmica, e informações a respeito do processamento e do controle necessitariam ainda ser elaboradas.

Os modelos computacionais abstratos usados para expressar o comportamento cerebral possuem duas formas básicas: um esquema teórico e as redes neurais artificiais. Estes dois modelos são altamente compatíveis. A teoria de esquemas é uma abstração em alto nível pela qual o comportamento pode ser expresso modularmente. As redes neurais artificiais fornecem uma base para modelar o problema com uma melhor granularidade, onde o processamento paralelo ocorre em um nível mais interno.

Atualmente a teoria de esquemas está mais apta a expressar a função, enquanto que as redes neurais podem refletir mais fielmente a estrutura do cérebro. Os esquemas, uma vez formulados, podem ser traduzidos para modelos de redes neurais, se desejado

ou julgado necessário [ARK 98].

2.5.5 Sistemas Híbridos

Dada a sua capacidade de controlar entradas imprevisíveis, uma rede neural é bem mais adequada para a criação de bancos de conhecimentos a partir de entradas do mundo real. Entretanto, depois que o banco de conhecimentos foi criado, um sistema simbólico, geralmente, será mais adequado para o seu controle. Este relacionamento complementar indica que talvez a melhor solução seja algum tipo de sistema híbrido, que combine o melhor de ambas as abordagens.

Para fazer programas mais inteligentes, combina-se várias técnicas de inteligência artificial. Em [FAB 95] pode ser visto um sistema que utiliza as redes neurais e sistemas *fuzzy* como pré-processadores de regras heurísticas, onde dados numéricos são transformados em conjuntos de dados simbólicos para que possam ser manipulados.

Sistemas que possuam um elevado número de parâmetros de entrada é outro exemplo de aplicação. Isto se deve ao fato de ser muito difícil para um projetista humano desenvolver um conjunto completo de regras para a descrição do sistema. Uma solução seria a aplicação de redes neurais artificiais para auxiliar na geração das regras [FUK 94]. Muitas outras pesquisas [GAT 92] [FAB 95] [BAL 91] estão utilizando sistemas *neuro-fuzzy*, ou seja, usando redes neurais artificiais em conjunto com lógica *fuzzy*.

Também os algoritmos genéticos são utilizados. São úteis na otimização das topologias de redes neurais artificiais, propiciando que a topologia da rede evolua com o passar do tempo [BRO 93] [MON 94]. A capacidade de manipulação de símbolos dos algoritmos genéticos pode fazer com que novas regras sejam produzidas quando utilizados em conjunto com sistemas especialistas [RAM 94a] [KUB 94] [BEL 91] [WOL 91].

2.6 O Controle da Navegação

Navegação é a ciência (ou a arte) de direcionar o curso de um robô móvel durante o seu deslocamento pelo ambiente terrestre, aquático ou aéreo. Inerente à qualquer esquema de navegação está o desejo de alcançar um determinado objetivo sem se perder ou bater em algum obstáculo. Muitos problemas ainda devem ser solucionados até que os robôs consigam se locomover com as habilidades de um humano. A maioria dos algoritmos aborta a execução quando se deparam com situações que dificultam a navegação.

A navegação baseada em sistemas de orientação é, atualmente, comum na indústria; já a navegação utilizando pontos de referência para corrigir possíveis desvios em um caminho pré-estabelecido está começando a ser utilizado. A navegação baseada em um processo de planejamento de rota autônomo continuará a ser pesquisado durante

os próximos anos.

Existem muitas técnicas para o controle do movimento de robôs móveis. Algumas só podem ser usadas em ambientes fechados; outras, ao contrário, podem ser utilizadas em meios totalmente desconhecidos, baseando-se em dados sensoriais [SMI 94] [JAN 95] [BAL 91]. Será abordado, primeiramente, as usadas em ambientes previamente conhecidos [KEY 90] [SEN 92] [TAN 92].

As técnicas para o gerenciamento da movimentação dos robôs incluem três sub-tarefas principais [MCK 91]:

- **mapeamento do ambiente** [ZIM 94b] [NEH 94]: Pode ser feito através de estruturas de dados, a partir de mapas pré-existentes [ZAR 93] ou através dos dados obtidos por meio de seu mecanismo sensorial durante a navegação [KRO 94] [MAT 90];
- **planejamento das tarefas**: Foco central do controle de movimentos. Neste, procuram-se maneiras de implementar um determinado comportamento [MCK 91] como, por exemplo, seguir um alvo, desviar de obstáculos, estacionar em uma plataforma de carga, etc.;
- **execução do planejamento**: Quando os atuadores do robô colocam em prática tudo aquilo que foi previamente planejado. Algumas vezes parece sobreposta à anterior, como em [MON 94] [WIL 93].

Esta subdivisão não é definitiva, tampouco rígida. Podem ocorrer casos em que as tarefas podem estar sobrepostas. A partir de agora, serão apresentadas as técnicas mais comuns para o controle dos movimentos de robôs móveis. Primeiramente, abordam-se aquelas utilizadas em ambientes conhecidos *a priori*. Logo após, aparecem as que utilizam somente os dados sensoriais [FAB 95].

2.6.1 Planejamento de Caminhos (*Path Planning*)

O controle dos movimentos robóticos em locais conhecidos caracteriza-se por ter a sub-tarefa do mapeamento do ambiente restringida a uma estruturação de dados que é baseada em informações previamente conhecidas sobre o ambiente [FIK 72]. Logo após este passo, planejam-se e executam-se os movimentos.

Muitas são as técnicas de controle da navegação para os *robots* em locais conhecidos. A grande maioria baseia-se em análises matemáticas e no cálculo de caminhos no ambiente previamente conhecido [CHO 95] [RIM 92]. Em alguns casos, existe a necessidade de que o espaço seja bidimensional; em outros há uma maior liberdade. Também, quanto aos obstáculos ou objetos inseridos no ambiente, existe uma grande variedade de possibilidades. Alguns possibilitam trabalhar somente com objetos poligonais; outros trabalham com qualquer tipo de formas.

Os três modelos principais para a navegação em ambientes conhecidos, segundo [LAT 93], e desconsiderando-se as particularidades, são a criação de mapas de trajetória

(roadmaps), o campo potencial e a decomposição de células.

2.6.1.1 Mapas de Trajetórias (*Roadmaps*)

Esta técnica procura conectar os espaços livres de obstáculos através de arcos unidimensionais que definem os caminhos por onde o robô poderá deslocar-se pelo ambiente sem colidir com os objetos [LAT 93]. Muitos são os modelos propostos, sendo que os grafos de visibilidade [LAT 93] e os diagramas de Voronoi [CHO 95] são os principais.

- **Grafo de visibilidade:** os nodos são construídos a partir do ponto de partida (origem), do alvo (destino) e dos vértices dos objetos. Consiste na criação de linhas que conectam todos os pares de pontos visíveis entre si, ou seja, quando uma reta pode ser traçada para interligá-los de forma que não interseccione nenhum obstáculo ou objeto. Tendem a gerar caminhos que passam bem próximos dos obstáculos (figura 11). Para escolher o caminho pode ser feito um roteamento em árvore; o escolhido deve ser aquele com o menor percurso.

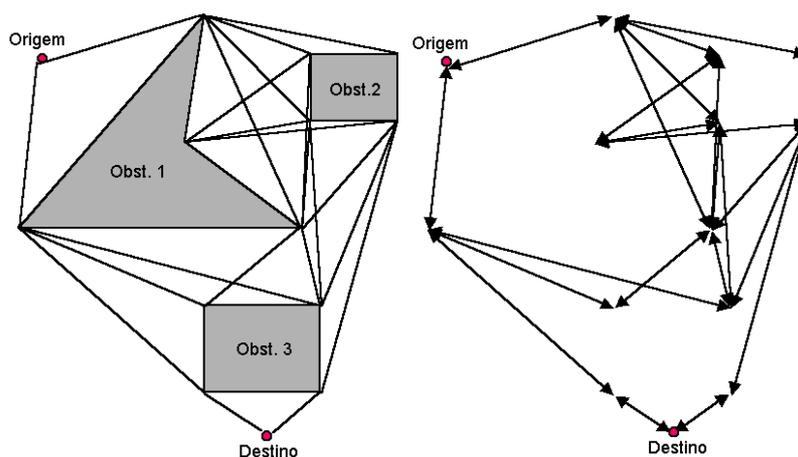


FIGURA 11 – Grafo de visibilidade gerado a partir de um ambiente contendo três obstáculos poligonais.

- **O diagrama de Voronoi:** procura construir um esqueleto [LAT 93] dos espaços livres de obstáculos, particionando o plano de trabalho em regiões poligonais [KOR 98]. Cada uma destas regiões é associada a um ponto gerador [MCK 91]. A delimitação das regiões é determinada pelos pontos equidistantes dos dois pontos geradores da mesma. Este método não garante que o caminho planejado será o mais curto possível; ele procura manter o robô o mais afastado possível dos objetos. Diagramas de Voronoi podem ser adaptados para armazenar as distâncias desde o esqueleto gerado até os obstáculos. Pode ser usado também para determinar a posição do robô. Veja, na figura 12, o exemplo de um diagrama de Voronoi.

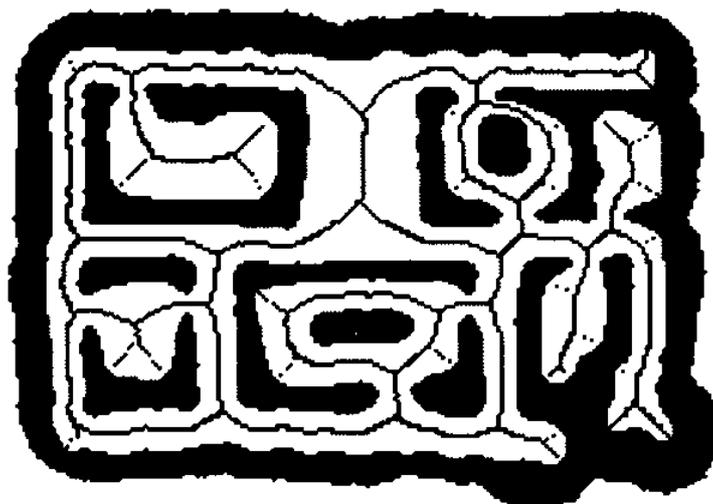


FIGURA 12 – O diagrama de Voronoi de um determinado labirinto.

2.6.1.2 Decomposição Celular

Outra técnica que pode ser utilizada é a decomposição celular, que consiste na divisão do espaço livre em pequenas unidades – as células. Existem duas variedades, a decomposição celular exata e a aproximada.

- **A decomposição celular exata** é mais indicada para modelos de ambientes artificiais, na maioria das vezes formados por polígonos (figura 13). Pode consistir de um intrincado conjunto de polígonos. A união das células modela com exatidão o espaço livre e os objetos. Se um caminho existir, o planejamento da trajetória estará fadado ao sucesso. A delimitação das células, normalmente de formas triangulares ou trapezoidais, geralmente define as características do objeto. As células podem estar cheias ou vazias.

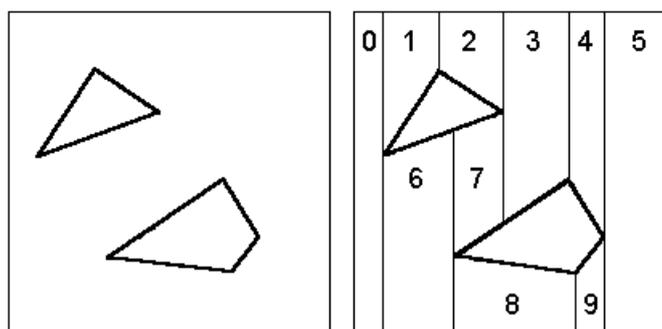


FIGURA 13 – Decomposição celular exata.

- **A decomposição aproximada** (figura 14) é mais fácil de ser utilizada, por ser uma simples aproximação matemática do ambiente, porém faz com que o modelo careça de detalhes. Mesmo assim, superfícies curvas podem ser modeladas. A união de células vazias é um subconjunto do espaço livre. As células têm um formato uniforme. A maneira mais utilizada é a enumeração da ocupação do espaço, porém pode tornar-se um método com um

armazenamento ineficiente. Torna-se um grande problema quando uma alta resolução é necessária. Pode-se efetuar divisões recursivas das regiões até que se chegue a células que estejam completamente cheias ou vazias. Árvores quaternárias são utilizadas para efetuar o mapeamento das regiões, como pode ser visto na figura 15.

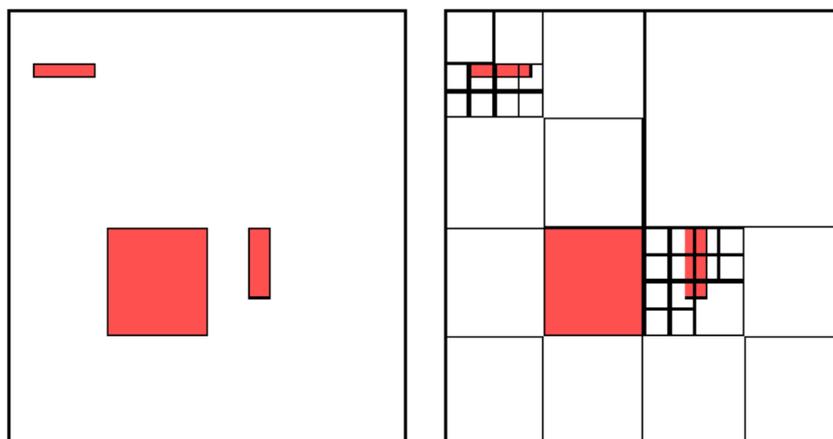


FIGURA 14 – Decomposição celular aproximada.

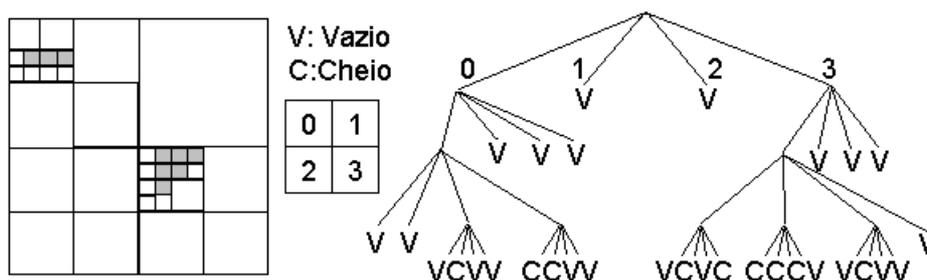


FIGURA 15 – A árvore quaternária gerada pela decomposição celular aproximada de um ambiente exemplo.

2.6.1.3 Campos Potenciais

Outra técnica que tem sido usada por muitos pesquisadores para a simples geração da trajetória em um ambiente com obstáculos é a dos campos potenciais artificiais [KOR 98] [KHA 84] [KRO 84]. Neste esquema (veja a figura 16), o robô é representado por um simples ponto. Vales com potenciais atratores (representando os objetivos) e “montanhas” com potenciais repulsivos (indicando obstáculos) são utilizados para gerenciar o deslocamento do robô pelo espaço.

Na figura 16 aparecem dois modelos de trajetórias geradas por campos potenciais. À esquerda temos uma rota onde o destino (círculo no canto inferior direito em ambas as figuras) é o fator predominante. À direita, um percurso que foi gerado partindo do princípio que a tarefa primordial é evitar os obstáculos (os grandes círculos escuros).

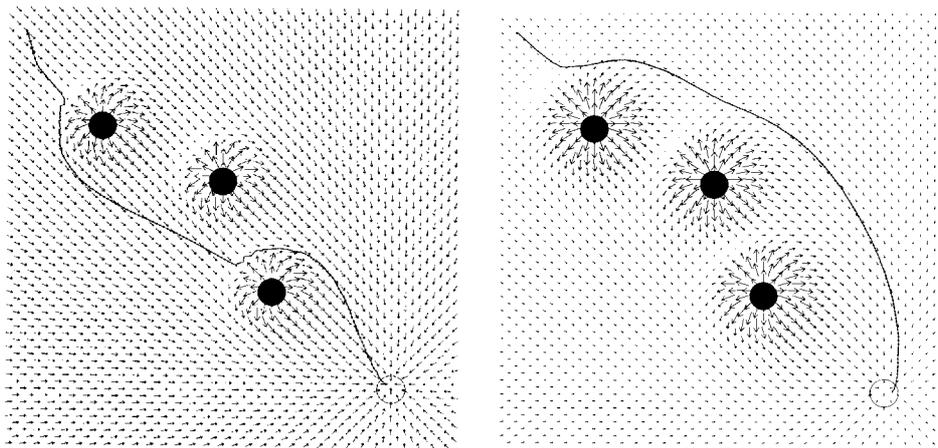


FIGURA 16 – Dois exemplos de percursos usando o campo potencial.

A utilização do campo potencial serve particularmente bem para o caso onde um plano de movimento global foi pré-computado, tal que a necessidade de o robô se desviar de seu caminho intencional acontece somente para evitar obstáculos inesperados que impedem o seu progresso ou ameaçam uma colisão. Os obstáculos em questão podem ser estacionários ou móveis.

Dentro desta abordagem, é possível que o robô venha a cair em mínimos potenciais locais, como por exemplo, em um canto onde tem um obstáculo por perto, que o impeça de progredir diretamente para o seu alvo ou destino.

A metodologia do campo potencial para o planejamento de movimentos é inerentemente um esquema para planejamento de movimentos “locais”. É insuficiente para capturar a informação de trajetória global. Por outro lado, as computações necessárias são grandemente paralelizáveis e podem ser efetuadas com muita rapidez – a aproximação potencial pode ser vista e usada, então, como um complemento importante para um gerador inteligente de uma trajetória global.

2.6.2 Navegação Baseada em Sensores

A grande vantagem da navegação baseada em sensores reside no fato de o robô comportar-se levando em conta a situação do meio onde está atuando e também por considerar as suas condições internas [FAB 95]. O sistema sensorial é o responsável por captar estas informações – utilizando uma grande variedade de sensores – e por repassá-las para o sistema de controle, para que sejam levadas em consideração durante a realização de uma tarefa ou no próprio deslocamento [HAL 94] [MAT 90]. A verificação do nível de energia restante nas baterias internas ou a detecção da entrada de um obstáculo no raio de ação do robô são exemplos simples da utilização dos mecanismos sensoriais.

Outra vantagem advinda da utilização da navegação baseada em sensores é que os robôs podem atuar livremente, ou seja, sem qualquer conexão física com equipamentos de apoio – podem ser completamente autônomos [BAL 91] [WAL 91] [BOS 93]. Isto possibilita verificar um “comportamento inteligente”, uma vez que suas

decisões baseiam-se exclusivamente na “leitura”, periodicamente atualizada, do ambiente e do seu próprio estado, inexistindo influências controladoras externas.

Para se utilizar a navegação baseada em sensores, algumas especificações adicionais se fazem necessárias. Tendo em mente que o controle é realizado em tempo real [LUO 94] [ZIM 94c], mecanismos sensoriais mais complexos e uma informática embarcada mais potente se fazem necessárias, uma vez que as decisões devem ser tomadas em exíguos intervalos de tempo, visando atingir um alto desempenho.

Outra capacidade que precisa ser incorporada é o tratamento de ruídos, pois nenhum sensor possui uma total acurácia. O sistema de controle deve “perceber” e tratar estas variações de forma transparente [SUK 94] [FER 94] [FLO 95]. Algumas condições afetam a atuação dos sensores, como o que acontece, por exemplo, com os sensores de infravermelho que são altamente influenciáveis pela quantidade de luz no ambiente.

Quando se trabalha com a navegação baseada em sensores, os robôs móveis em questão são chamados de agentes autônomos. Podem ser definidos como um agente artificial atuando sobre um ambiente real ou fictício, continuamente mapeando as leituras do sistema sensorial na saída dos atuadores [ROI 97]. O ambiente é tudo o que está externamente posicionado em relação ao agente, o que pode incluir desde outros agentes, obstáculos e até pessoas. O agente opera em ciclos, recebendo os estímulos ambientais através dos sensores e realizando algum processamento [MAT 94]. Este, gera uma saída que vai ser executada pelos atuadores, que possivelmente afetarão o ambiente. Isto é chamado de comportamento [KAE 93].

Em uma extremidade do espectro controlador dos agentes, encontra-se uma estratégia deliberativa muito tradicional – o tratamento baseado em planos (PBA: *Planner Based Approach*) – uma arquitetura de controle do tipo *top-down* [MAT 94]. É utilizado um modelo centralizado do ambiente para verificar as informações advindas dos sensores e planejar as atitudes. Esta forma de atuação permite formular e explicitar as tarefas e os objetivos do sistema e também avaliar o desempenho e a qualidade das atitudes do agente.

Muitos dos modelos utilizados para o planejamento do caminho em ambientes conhecidos estão sendo adaptados para que possam ser utilizados com a navegação sensorial, como o campo potencial e os diagramas de Voronoi [CHO 95] [RIM 92] [ROQ 97]. Um grande problema que pode advir da utilização de sensores é o fato de ser frequentemente necessário o replanejamento, gerando uma necessidade computacional bastante alta quando em sistemas muito complexos. Logo, a aplicação em tempo real é prejudicada [BRO 91b] [BRO 90].

Pesquisas estão sendo desencadeadas visando descobrir metodologias de controle que possam ser utilizadas em tempo real, especialmente as do tipo puramente reativa (pares de ação/reação), com uma arquitetura que seguem o modelo *bottom-up*. É interessante salientar também que estes sistemas não guardam modelos para representar o ambiente, simplesmente gerenciam o sistema para responder corretamente aos dados fornecidos pelos sensores. São propostas que se mostram eficazes para uma grande variedade de problemas, mas que são inflexíveis para utilização em tempo real, pois são incapazes de armazenar dinamicamente as informações [BOT 96].

As arquiteturas híbridas misturam as abordagens puramente reativas com as deliberativas, uma vez que utilizam um sistema reativo para o controle de baixo nível e um planejador para decisões de alto nível. O planejamento ou execução reativa (RPA–*Reactive Actions Packages*), que utilizam primitivas de alto nível para o planejamento, responsáveis por todos os detalhes da execução e o *PRS (Procedural Reasoning System)* – uma arquitetura que invoca regras flexíveis [MAT 94] [ARK 98], são alguns exemplos. Tais sistemas tendem a dividir o controle em módulos independentes, porém estes se comunicam entre si. Geralmente os processos reativos de baixo nível garantem a sobrevivência do robô, enquanto níveis mais altos escolhem e coordenam a execução das ações a serem realizadas.

Outra abordagem que está ganhando muitos adeptos – e que é uma extensão dos sistemas reativos – é a baseada em comportamento [BRO 86] [ARK 98]. É consideravelmente mais avançada do que os sistemas puramente reativos, pois seus estados internos, formados também por pares de ação-reação, não apresentam limitação alguma. O processamento das informações pode ser feito de forma paralela e distribuído [MAT 94] – bem mais do que uma simples pesquisa em tabelas como em outros sistemas – levando em consideração os dados sensoriais e onde podem ser usadas diversas formas de representação interna do ambiente.

Uma de suas subdivisões é o comportamento emergente, que provoca uma sensação quase mística, dada as capacidades dos sistemas baseados em comportamento. Este, implica em uma capacidade holística, onde o valor global é consideravelmente maior do que o somatório das suas partes constituintes. É verdade, também, que alguns acontecimentos que ocorrem em sistemas baseados em comportamento freqüentemente são uma surpresa, inclusive para o seu projetista.

A noção do comportamento emergente como um fenômeno místico precisa ser dispersado, mas o conceito de uma maneira bem definida ainda pode ser útil. Um grande número de pesquisadores discutiu o significado de “emergente” em seus vários aspectos, dentre estes, podemos citar:

- Emergente é "o surgimento de novas propriedades em sistemas já considerados completos" [MOR 88].
- “A funcionalidade global emerge da interação paralela de determinados comportamentos” [STE 90];
- "Inteligência emerge da interação dos componentes do sistema" (na qual a funcionalidade do sistema, isto é, o planejamento, a percepção, a mobilidade, etc., é o resultado dos componentes geradores de comportamento) [BRO 91a];
- “A funcionalidade emergente surge em virtude de interação entre componentes que foram projetados com uma determinada função particular” [MCF 93].

Uma característica comum em todas estas declarações, é que a emergência é uma propriedade de uma coleção de componentes interagindo entre si. Porém, surge uma questão. Se o comportamento individual e a funcionalidade dos componentes são bem definidos, então o que causa o surgimento de um resultado inesperado a partir de

uma coleção de determinados comportamentos?

Funções de coordenação são, na verdade, algoritmos e, conseqüentemente, não contêm nenhuma surpresa, tampouco alguma perspectiva mágica. Em alguns casos, eles são diretos, como escolher o comportamento mais adequado; em outros, podem ser mais complexos e envolver a fusão de múltiplos comportamentos ativos. Todavia, eles geralmente possuem um funcionamento determinístico e, certamente, computável.

O surgimento de novos comportamentos acontece em decorrência da relação do robô com o seu ambiente. Para a maioria das situações nas quais o paradigma baseado em comportamento é aplicado, o próprio ambiente resiste em ser analiticamente modelado. O não determinismo é bastante grande. O mundo real está cheio de incertezas e propriedades dinâmicas.

Além disso, o próprio processo de percepção também é pobremente caracterizado; não existem modelos sensoriais precisos para ambientes abertos. Se um modelo do mundo pudesse ser criado de forma a capturar com precisão todas as suas particularidades, então não existiria a emergência de novos comportamentos: previsões precisas poderiam ser feitas.

Mas é uma característica do mundo resistir a tal tipo de caracterização, conseqüentemente não se pode prever *a priori*, com qualquer grau de confiança, como o ambiente aberto se apresentará, mesmo o mais simples deles. “Os modelos probabilísticos podem indicar caminhos a serem seguidos, mas jamais fornecem graus de certeza para estes direcionamentos” [ARK 98]

2.6.3 Percebendo o Ambiente

Para considerar que algo ou alguém quer dizer alguma coisa, é necessário que exista uma maneira de se entender a informação, ou seja, um protocolo de comunicação. De certa forma, a percepção do ambiente é uma maneira de comunicação – a informação segue o sentido do ambiente para o agente que pretende tratá-la. Claro que, se o agente não sabe como tratá-las, de nada adianta ficar percebendo o ambiente. Nota-se também, que o ambiente está informando alguma coisa, se o agente estiver “prestando atenção”. A forma com que os sensores estão dispostos e direcionados depende muito da motivação e da intenção.

A literatura etiológica está repleta de exemplos a respeito de informações providas de sensores, fornecendo pistas para o despertar do comportamento [ARK 98]. A evolução forneceu mecanismos muito apurados aos agentes biológicos, capazes de captar eficazmente as informações necessárias para realizarem ações úteis. Encontram-se seres capazes de discernir objetos e outros que simplesmente não conseguem.

As percepções necessárias para a sobrevivência e para os procedimentos rotineiros de um organismo são extraídas do ambiente de forma eficaz e a um custo muito baixo, enquanto que as informações irrelevantes não são processadas (na verdade, elas nem mesmo são adquiridas). O que aconteceu, de fato, é que estes organismos evoluíram seus mecanismos, possibilitando uma comunicação eficiente visando a

perceber as informações relevantes (necessárias para a sua sobrevivência) do seu habitat.

Isto significa que os agentes robóticos devem possuir mecanismos capazes de discernir e extrair as informações que são relevantes a sua manutenção. Dependendo de suas condições internas, seu estado de motivação ou de suas metas e da limitação nos seus mecanismos sensoriais, pode-se desenvolver algoritmos que forneçam informações úteis e direcionadas.

A tecnologia dos sensores avançou rapidamente na última década e resultou em sistemas sensoriais de baixo custo que podem ser prontamente aplicados na robótica. Os sensores podem ser categorizados, em termos da sua interação com o ambiente, de passivo ou ativo:

- **Sensores passivos:** usam a energia naturalmente presente no ambiente para obter informação. Visão de computador é talvez a mais típica forma de sensoriamento passivo. A passividade é particularmente importante em aplicações militares, em que os mecanismos devem atingir altos graus de imperceptibilidade.
- **Sensores ativos:** envolvem a emissão de energia, por algum mecanismo, para o ambiente; de alguma forma esta energia retorna, total ou parcialmente, para o robô. Sensores de ultra-som e medidores de distância através de feixe de laser são dois exemplos comuns deste tipo de mecanismo usados em agentes robóticos.

A seguir, faz-se uma breve apresentação do funcionamento dos sistemas de sensoriamento mais importantes e utilizados, inclusive o uso de contadores passivos. Estes, não são sensores ambientais, na verdade, eles só medem as rotações dos motores do robô. Entretanto, eles são amplamente usados para estimar o posicionamento do robô e merecem uma discussão adicional. Informações complementares sobre os sensores podem ser obtidas em [EVE 95].

2.6.3.1 Contadores Passivos (*Dead Reckoning*)

Estes contadores servem para estimar o espaço percorrido pelo robô levando em conta o giro de seus motores, rodas ou passos (odometria). Podem ser de dois tipos: contadores de giro ou sistemas de navegação inercial (INS – *Inertial Navigating Systems*).

Os contadores de giro são os métodos mais freqüentemente utilizados devido ao seu baixo custo. Operam mantendo um contador do número de voltas que a haste do motor direcional ou motriz (ou os eixos das rodas) efetuaram. Convertem, então, estes dados na distância percorrida pelo robô e sua orientação.

Apesar de fornecerem informações posicionais confiáveis através da cinemática direta, quando instalados em braços mecânicos que são fixos com relação ao ambiente, nos sistemas móveis, podem ser extremamente enganadores.

Se os eixos onde estão situados os contadores, como o que acontece por

exemplo em um automóvel com tração dianteira (normalmente em uma das rodas dianteiras está instalado o sensor de deslocamento), se suas rodas motrizes giram em falso (patinam), a informação da distância percorrida (leitura do odômetro) estará completamente errada. Por isso, estes devem ser complementados com outros sensores ambientais, visando fornecer resultados mais confiáveis a respeito da localização do robô no ambiente.

Os sistemas de navegação inercial não se baseiam na contagem das rotações e sim no rastreamento das acelerações sofridas pelo robô, convertendo tal informação em deslocamentos. São mais precisos e também mais caros, o que faz com que sejam menos utilizados.

Outro mecanismo que, apesar de não ser um contador passivo, pode fornecer informações sobre o posicionamento geográfico do robô são os sistemas de posicionamento global (*GPS – Global Positioning System*). Estes, podem fornecer a latitude, a longitude e a altitude, através de cálculos sobre o tempo de deslocamento do sinal dos satélites e também pela triangulação.

Um problema encontrado é que os sistemas de GPS públicos não fornecem uma boa precisão. Na verdade, no melhor dos casos, sua precisão é de aproximadamente 100 metros. Os militares, obviamente, são mais exatos.

Outro problema que afeta estes sistemas é o bloqueio dos sinais dos satélites em algumas construções. Logo, os GPS precisam ser aperfeiçoados para que possam ser efetivamente utilizados.

2.6.3.2 Ultra-som (Sonar)

Este é um mecanismo de sensoriamento ativo. Opera no mesmo princípio pelo qual os morcegos navegam pelo ambiente. É emitido um breve sinal sonoro (*click*) que é refletido por uma superfície próxima e que retorna até o emitente em um tempo que pode ser calculado. Através desta demora, do retorno do sinal, pode ser calculada a distância até o objeto, uma vez que se sabe a velocidade de propagação do sinal.

Um típico sensor de ultra-som (*Polaroid*) emite um feixe com uma cobertura de 30 graus. Estes sensores podem operar rapidamente, com 10 ou mais leituras por segundo. A precisão de muitos sistemas é da ordem de 3 centímetros com um alcance máximo de 10 metros.

Existe uma infinidade de sistemas comercialmente disponíveis, cobrindo um grande espectro de frequências, cada qual com variações na largura do feixe e também no alcance.

Possui vantagens e desvantagens. Uma das vantagens é o seu baixo custo e também a capacidade de capturar informações tridimensionais a respeito do ambiente; também retorna uma quantidade de dados que podem ser interpretados.

Suas desvantagens também são significativas. É muito suscetível a ruídos e às distorções devido às condições ambientais. Frequentemente fornecem leituras erradas, devido às reflexões do sinal, e o feixe pode espalhar-se demasiadamente. São indicados

para detectar e evitar obstáculos nas proximidades (curto alcance). A dificuldade em discernir os diferentes tipos de objetos, por exemplo, entre um obstáculo e um alvo, limita a sua aplicação.

2.6.3.3 Visão Computacional

Disponível há quase meio século, só recentemente a tecnologia das câmeras CCD's (*Charge-Coupled Device*) avançou rapidamente em termos de miniaturização e de redução de custo. Imagens coloridas estão agora disponíveis com um custo bastante variado.

Apesar do surgimento das câmeras digitais, a maioria dos sistemas de visão robóticos utiliza sistemas com saída analógica. Consistem de uma ou duas câmeras em preto e branco, ou coloridas, de um ou mais digitalizadores e um processador de imagem. A necessidade da interpretação dos dados em tempo real força a utilização de sistemas de processamento de imagens de arquiteturas proprietárias.

O volume de informações fornecidas por um sistema de vídeo pode ser problemático. Uma câmera simples gera tipicamente uma imagem com 512x512 *pixels*, onde cada *pixel* possui um *byte* de informação indicando as 256 tonalidades possíveis de cinza (para imagens coloridas, basta multiplicar por 3 – um *byte* para cada cor: vermelho, verde e azul).

O grande problema advém da necessidade de se tratar estas imagens a uma velocidade de 30 quadros por segundo (aproximadamente 24mb de informações por segundo). Somente sistemas de processamentos de imagens especializados, e logicamente dispendiosos, são capazes de tratar este fluxo de dados.

2.6.3.4 Exploradores Laser

São sensores ativos que emitem um raio laser de baixa intensidade que é rastreado sobre uma superfície. Através de técnicas, como a modulação da amplitude em fase, a distância dos pontos individuais pode ser obtida como resultado de uma seqüência de pontos da imagem na qual cada qual tem uma profundidade associada.

A reflexão do sinal também pode ser usada para determinar a natureza da superfície do objeto. O resultado é uma fonte de dados tridimensionais extremamente rica.

Também possuem suas desvantagens. A primeira delas é o alto custo, muitas vezes maior do que os sistemas de ultra-som. Outro problema decorre das instabilidades mecânicas associadas a muitas das implementações destes dispositivos.

Dificuldades substancialmente maiores decorrem da amostragem de dados esparsos encontrados em grandes distâncias, o que pode causar a não detecção de objetos, e problemas com periodicidades no alcance, quando se utilizam sistemas de fase modulada, que força a interpretação de ambigüidades nos dados de entrada relativos à distância da superfície.

Exploradores laser lineares de baixo custo estão disponíveis, porém fornecem menos informações. Todavia, com a melhora na tecnologia de sensores nos próximos anos, espera-se que estes dispositivos fiquem mais úteis e tornem-se mais comuns.

3 Redes Neurais Artificiais (RNAs)

3.1 Embasamento Biológico

O trabalho em redes neurais artificiais, comumente referidas como “redes neurais”, tem sido motivado desde o início pelo reconhecimento de que o cérebro trabalha de maneira totalmente diferente de um computador digital. A batalha para compreender o funcionamento do cérebro deve-se muito ao trabalho pioneiro de Ramón y Cajál em 1911, que introduziu a idéia de que os neurônios eram as estruturas constituintes do cérebro.

Tipicamente os neurônios são de cinco a seis ordens de grandeza mais lentos do que as portas lógicas de silício; os eventos em um *chip* acontecem em nanosegundos (10^{-9}), enquanto que os eventos cerebrais acontecem na faixa dos milisegundos (10^{-3}).

No entanto, o cérebro compensa sua menor velocidade de operação através da grande quantidade de neurônios massivamente interconectados; estima-se que deva existir cerca de 10 bilhões de neurônios no córtex cerebral humano e 60 trilhões de sinapses ou conexões. A rede resultante indica que o cérebro é uma estrutura extremamente eficiente. Um exemplo é o consumo energético. O cérebro consome aproximadamente 10^{-16} joules por operação por segundo, enquanto que os melhores computadores digitais em uso atualmente consomem aproximadamente 10^{-6} joules[ARK 98]

O cérebro é um computador altamente paralelo, complexo e não linear. Tem a capacidade de organizar os neurônios para executar certas tarefas computacionais (o reconhecimento de padrões, a percepção e controle motor) muitas vezes mais rápidos do que o computador digital mais veloz disponível.

Considere, por exemplo, a visão humana, que é uma tarefa típica de processamento de informações [ARK 98]. A função do sistema visual é fornecer uma representação do ambiente a nossa volta e, mais importante, o fornecimento de informações necessárias para que possamos interagir com o meio.

Para ser mais específico, o cérebro executa estas tarefas (o reconhecimento de um rosto familiar imerso em uma cena completamente desconhecida) de maneira rotineira. Necessita de um tempo da ordem de 100 à 200 milisegundos para efetuá-la, enquanto que uma tarefa de muito menor complexidade leva dias em um potente computador [ARK 98].

Um outro exemplo é o sistema de sonar de um morcego; um sistema de eco-localização ativo que provê informações importantes, como a distância do alvo (digamos, um inseto voando), a sua velocidade relativa e o seu tamanho. Inclui também algumas características principais como o azimute e a elevação [ARK 98].

Toda a complexidade computacional necessária para extrair estas informações ocorre em um cérebro menor do que uma ameixa. Isto acontece em altíssima velocidade, permitindo que os morcegos persigam e capturem suas presas durante o voo.

Quando nascemos, o cérebro tem uma estrutura grandiosa e a habilidade de criar suas próprias regras (o que chamamos comumente de experiência). Na verdade, a experiência é construída no decorrer de anos, mas grande parte se dá nos 2 primeiros anos de vida (cerca de 1 milhão de sinapses por segundo são criadas neste primeiro estágio).

Os neurônios possuem diversos formatos e tamanhos de acordo com a região do sistema nervoso onde se localizam. A figura 17 mostra a estrutura de uma célula piramidal, um dos tipos mais comuns de neurônios corticais. Este neurônio pode possuir 10.000 ou mais contatos sinápticos – através dos seus dendritos e projetar a sua saída em milhares de outros neurônios – usando os seus terminais sinápticos.

A partir deste conhecimento, contruíram-se as redes neurais artificiais.

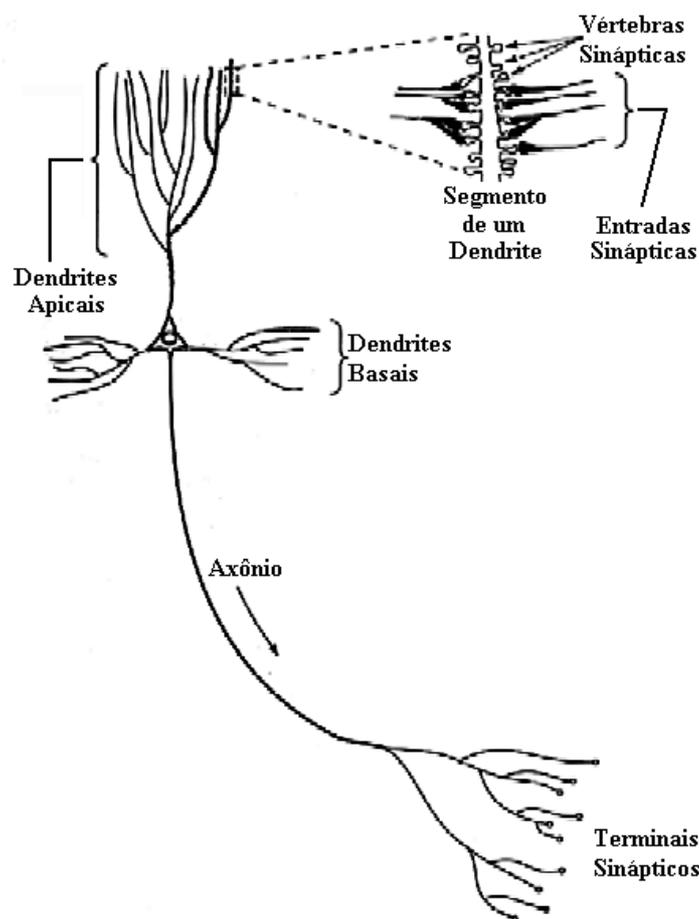


FIGURA 17 – A célula piramidal: um exemplo de neurônio.

3.2 Histórico das Redes Neurais Artificiais

As redes neurais artificiais procuram imitar o cérebro. Os modelos computacionais para as redes neurais artificiais, também referenciados como sistemas conexionistas, possuem uma história bastante rica, desenvolvida paralelamente com a tradicional inteligência artificial simbólica.

Alguns dos trabalhos iniciais nesta área podem ser atribuídos aos neurônios definidos por McCulloch-Pitts [MCC 43].

No trabalho, McCulloch e Pitts descreveram o cálculo lógico de uma rede neural artificial. Eles usaram uma única unidade funcional (figura 18) que possuía um limiar linear e com pesos sinápticos associados a cada conexão existente na entrada. Se este limiar fosse excedido, o neurônio disparava, levando seu resultado para o próximo neurônio.

A teoria de McCulloch-Pitts sobre as redes neurais artificiais tornou-se proeminente na segunda das quatro conferências feitas por von Neumann na Universidade de *Illinois* em 1949.

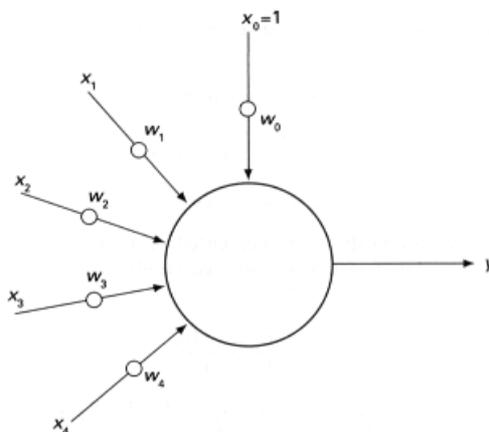


FIGURA 18 – O neurônio artificial.

O próximo grande passo na área deu-se em 1949, com a publicação do livro de Hebb: “*The Organization of Behavior*” [HEB 49], na qual uma declaração explícita da regra de aprendizagem fisiológica para modificação das sinapses foi apresentada pela primeira vez.

Hebb propôs, mais especificamente, que a conectividade do cérebro altera-se à medida que o organismo aprende novas atividades funcionais e que as topologias neurais são criadas por estas modificações.

Seguindo sugestões, introduziu o famoso postulado do aprendizado, no qual informa que a efetividade de uma sinapse qualquer entre dois neurônios é aumentada pela repetida ativação de um neurônio pelo outro através daquela sinapse.

O livro de Hebb tornou-se uma fonte de inspiração para o desenvolvimento de modelos computacionais de aprendizado e sistemas adaptativos. O relatório feito por Rochester, Holland, Habit e Duda em 1956 [ARK 98] é talvez a primeira tentativa de

uma simulação por computador para testar a teoria neural baseada no postulado de Hebb; a simulação mostrou que era necessário incluir a inibição para que a teoria realmente funcionasse.

Passados 15 anos desde a publicação de McCulloch-Pitts, uma nova abordagem para o reconhecimento de padrões foi apresentada por Rosenblatt, em 1958, em seu trabalho sobre o perceptron (veja figura 19). O coroamento do trabalho de Rosenblatt foi o teorema de convergência do perceptron. Em 1960, Widrow e Hoff apresentaram o algoritmo LMS (*Least Mean-Square*, ou Mínimo Médio Quadrado) e utilizaram-no para formular o Adaline. A diferença entre o perceptron e o adaline está no treinamento. Uma das primeiras redes neurais treináveis com múltiplos elementos adaptativos foi o Madaline [WID 60] [WID90].

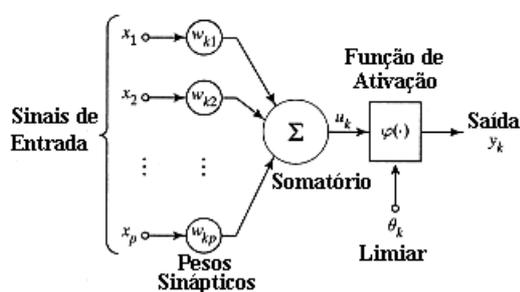


FIGURA 19 – O perceptron.

Nas décadas de 60 e 70, as pesquisas em redes neurais entraram em declínio por uma grande variedade de motivos, incluindo a publicação do livro “Perceptrons” [MIN 69], no qual provaram as limitações das redes de perceptrons de uma única camada (o problema era a função lógica ou-exclusivo). Acrescentaram ainda que as redes neurais utilizavam muitos processos empíricos.

Nos anos 80, entretanto, esta área de pesquisa ressurgiu. Primeiramente pela descoberta de Hopfield – que foi um físico e biólogo. Ele apresentou o seu modelo de redes neurais, baseado em informações coletadas nas inúmeras observações feitas sobre o sistema neurológico de uma lesma de jardim, que contestava as teses matemáticas de Minsky e Papert. O seu modelo, conhecido por rede de Hopfield [FRE 91], consiste em um conjunto de neurônios interconectados que tendem a buscar um estado de energia mínimo.

Outra grande contribuição para o ressurgimento das pesquisas em redes neurais foi o advento da rede neural em multi-camadas e o uso do algoritmo *backpropagation* como técnica de treinamento destes sistemas. Com este modelo de rede resolveu-se o clássico problema da função ou-exclusivo e as redes neurais voltaram a ter um pouco mais de credibilidade.

Muitos outros esforços notáveis dentro do conexionismo, numerosos demais para serem todos aqui citados, na última década renderam resultados altamente significativos. Porém, devido à importância, dois deles devem ser aqui mencionados. O primeiro são os mapas auto-organizáveis (SOM – *Self Organizing Maps*) criados por Teuvo Kohonen [KOH 82]; o segundo é o modelo ART (*Adaptive Resonance Theory*), definido por G. Carpenter e S. Grossberg [CAR 86].

Vale salientar, porém, que a grande maioria das redes neurais artificiais é inspirada no conhecimento biológico dos neurônios e que reproduzem as funções cerebrais com pouca fidelidade.

3.3 O Neurônio Artificial

O modelo de neurônio ainda amplamente utilizado é aquele mesmo proposto por McCulloch e Pitts.

O neurônio é um elemento processador de informação fundamental no funcionamento das redes neurais. Pode-se identificar três elementos básicos em um neurônio artificial ou perceptron são eles:

- um conjunto de sinapses ou elos de ligação, cada qual caracterizado por um peso. Mais especificamente o sinal x_j na entrada sináptica j conectada ao neurônio k é multiplicado pelo peso sináptico w_{kj} . Se o peso associado a esta sinapse for positivo, esta é dita excitatória; se for negativo, é dita inibitória;
- um acumulador para efetuar a soma dos sinais de entrada multiplicados pelos respectivos pesos das sinapses (as operações aqui mostradas constituem o combinador linear);
- uma função de ativação para limitar a amplitude do sinal de saída do neurônio. Também é conhecida como função limitadora por reduzir o intervalo permitido para o sinal de saída para alguns valores finitos. Tipicamente, a saída de um neurônio é normalizada para um intervalo $[0;1]$ ou $[-1;1]$.

O modelo de neurônio apresentado na figura 19 também inclui um limiar externo aplicado θ_k que tem a função de diminuir o valor de entrada para a função de ativação. Por outro lado, este valor pode ser incrementado aplicando-se o termo *bias* ao invés do limiar; o *bias*, portanto, é o contrário do limiar.

Matematicamente falando, podemos descrever o neurônio k pelos pares de equações abaixo:

$$V_k = \sum_{j=1}^p W_{kj} * X_j \quad (3.3a)$$

e

$$Y_k = \varphi(U_k - \theta_k) \quad (3.3b)$$

onde:

- X_1, X_2, \dots, X_p são os sinais de entrada;
- $W_{k1}, W_{k2}, \dots, W_{kp}$ são os pesos sinápticos do neurônio k ;
- V_k é a saída do combinador linear;

- θ_k é o limiar;
- $\varphi(\cdot)$ é a função de ativação; e
- Y_k é a saída do neurônio k .

Várias funções de ativação podem ser usadas, dentre elas podemos citar:

- **degrau**: possui dois estados estáveis (ativo ou inativo);
- **semi-linear**: garante uma continuidade, isto é, além dos dois estados estáveis, possui uma região de transição; e
- **sigmoidal**: sua saída é contínua dentro do intervalo.

3.3.1 Função Degrau

Para este tipo de função de ativação, como pode ser vista no gráfico a) da figura de número 20, temos:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases}$$

Esta é a função que foi utilizada no neurônio definido por [MCC 43], indicando a propriedade do tudo ou nada.

3.3.2 Função Semi-Linear

Para a função semilinear, gráfico b) da figura de número 20, apresenta o seguinte funcionamento:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq +1/2 \\ v, & \text{se } +1/2 > v > -1/2 \\ 0, & \text{se } v \leq -1/2 \end{cases}$$

Assumindo que o fator de amplificação do sinal na região linear é unitário. As duas formas especiais mostradas abaixo podem ser vistas como uma função semi-linear:

- a saída é o próprio valor do combinador linear se a região de operação for mantida sem entrar em saturação;
- a função semilinear pode reduzir a função de ativação se o fator de amplificação na região linear for infinitamente alto.

3.3.3 Função Sigmoidal

A função sigmoidal é, sem sombra de dúvida, a função de ativação mais utilizada na construção de redes neurais. É definida como uma função estritamente crescente e possui uma curva suave e assintótica. Um exemplo de função sigmoidal é a função logística, definida como:

$$\varphi(v) = \frac{1}{(1 + \exp(-av))}$$

Onde a é o parâmetro de inclinação da curva.

Variando-o, obtém-se funções sigmoidais de diferentes declividades (ver figura 20-c).

Enquanto uma função degrau assume valores 0 ou 1, a função sigmoidal fornece um intervalo contínuo entre 0 e 1. Note-se, também, que a função sigmoidal é derivável, enquanto que a função degrau não o é.

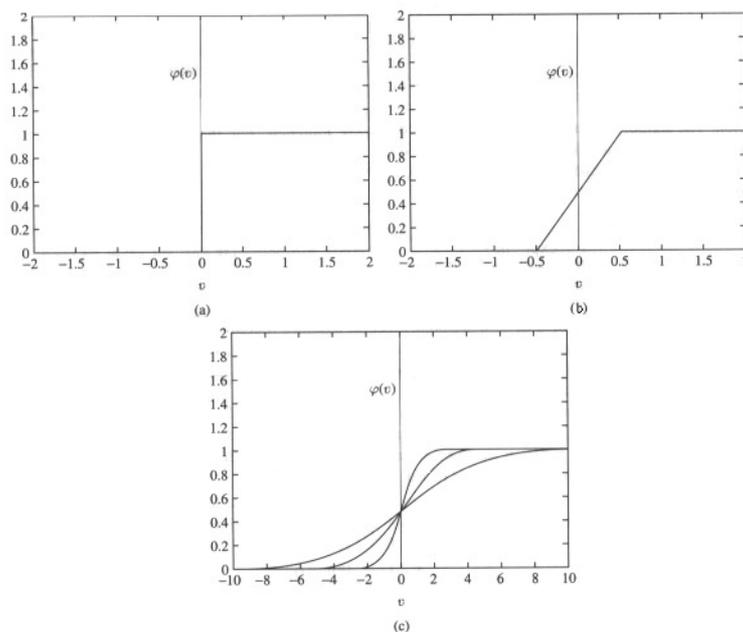


FIGURA 20 – Gráficos das funções de ativação mais utilizadas.

Na robótica, as funções preferidas são aquelas que oferecem uma saída contínua, do tipo sigmoidal. Uma das mais utilizadas é a tangente hiperbólica, que possui uma saída ininterrupta variando entre $[-1;1]$ e assume uma forma anti-simétrica em relação à origem [HAY 94]:

$$\varphi(v) = \tanh(v) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

3.4 Modelos de Redes Neurais Artificiais

Como relatado anteriormente, as redes neurais artificiais tentam modelar o comportamento do cérebro utilizando como unidades básicas os neurônios artificiais. Porém, devido a grande lacuna existente entre o comportamento de um neurônio biológico e o comportamento de um neurônio artificial, ainda não existe um modelo de rede neural artificial que seja tão genérico, a ponto de ser utilizado em qualquer tipo de aplicação.

Apesar das limitações, a cada dia que passa surge um novo tipo de aplicação para as redes neurais artificiais. Dentre elas podemos citar a robótica [BOT 96] [BOR 96], classificação de imagens digitais [LIB 97], reconhecimento de padrões visuais [BOR 97], controle de trajetórias [FER 96][SIL 97a], reconhecimento de voz [ADA 97], na área biomédica e financeira, na previsão do tempo, etc.

As RNAs possuem algumas vantagens frente às outras técnicas passíveis de serem utilizadas, como as baseadas em heurísticas, dentre as quais podemos citar:

- o processamento é distribuído e paralelo;
- ser tolerante a falhas;
- capacidade de generalização e aprendizado;
- podem ser aplicadas aos problemas altamente não-lineares.

Existem dois tipos de redes neurais quanto ao modo com que os neurônios estão interconectados. Podem ser do tipo *feedforward* e *feedback*, indicando o modelo sem e com realimentação do sinal, respectivamente (ver figura 21).

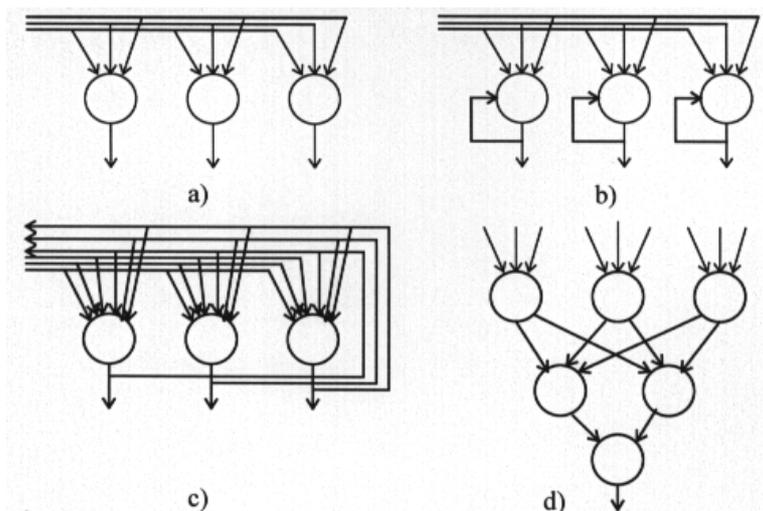


FIGURA 21 – Outros modelos de redes neurais.

O primeiro deles indica que as saídas dos neurônios estão conectadas somente aos neurônios adjacentes da camada superior; logo não poderá ser influenciado por sua própria saída. Um exemplo é o *backpropagation* (BPN), como mostrado na figura 22.

O segundo modelo é caracterizado pela realimentação dos sinais, ou seja, a saída de um neurônio serve de entrada para ele próprio ou para outros situados em camadas inferiores da rede, fazendo com que a sua saída possa ter influência direta ou indireta no seu comportamento. A rede Hopfield é um exemplo bem claro deste modelo.

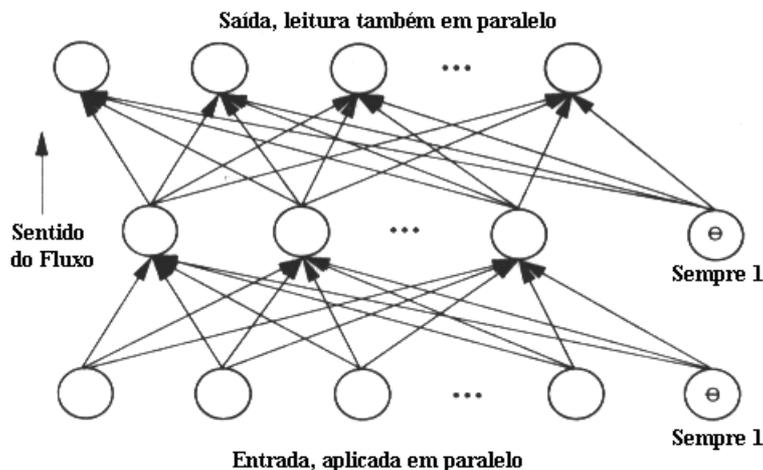


FIGURA 22 – A rede neural do tipo *backpropagation*.

Existem dois modos utilizados para fazer o mapeamento do conjunto de dados de entrada no conjunto de dados da saída: o supervisionado e o não supervisionado [FRE 91]. O treinamento supervisionado é feito com o auxílio de pares de exemplos de entrada com sua saída respectiva. À medida que vão sendo apresentados os valores de entrada e a saída está em desacordo com a desejada, é aplicada uma técnica de atualização dos pesos visando minimizar o erro gerado (a mais conhecida é a do gradiente descendente, usada nas redes BPN e BAM – *Bidirectional Associative Memory*).

Já o aprendizado não supervisionado, ou auto-supervisionado, não necessita de pares entradas-saídas. Ela vai se auto-organizando enquanto vão sendo fornecidas as entradas, segundo um critério próprio. Pode ser utilizado um processo de cooperação ou competição entre os neurônios. Um exemplo deste tipo de rede é a de Kohonen [FRE 91].

As redes podem, ainda, serem classificadas quanto ao número de camadas existentes. Assim, têm-se dois modelos: as em monocamada, que possuem somente uma camada de neurônios, e as em multicamada, que possuem mais de uma camada. Tipicamente as redes neurais possuem três camadas: a de entrada, a oculta e a de saída (figura 21 - d).

3.4.1 Funcionamento das Redes *FeedForward*

O funcionamento deste tipo de rede é bastante simples, possui apenas duas fases: a propagação e a adaptação. Os dados são apresentados para a camada de entrada da rede e propagados pela(s) camada(s) oculta(s) e de saída. Isto gera uma saída, que é comparada com a saída desejada. É calculado um erro, que representa a diferença entre

a saída da rede e aquela desejada. Este é então retropropagado, ajustando as conexões da rede. A utilização desta técnica direciona o sistema para um erro médio quadrado mínimo.

A atualização dos pesos é feita através da seguinte fórmula:

$$W_{ij}(t+1) = W_{ij}(t) + \eta \nabla E \quad (3.3c)$$

Onde:

η : indica o coeficiente de aprendizado (intensidade na atualização dos pesos para que atinja a configuração ideal);

∇E : é o gradiente do erro (representa o vetor direção para a configuração de pesos ideal).

Um problema que acompanha a técnica do gradiente descendente na atualização dos pesos é a ocorrência de mínimos locais, fazendo com que o sistema muitas vezes não atinja o mínimo global. Serão apresentados, no decorrer deste trabalho, mais detalhes a respeito da atualização dos pesos, pois este varia segundo o modelo de rede utilizado.

A seguir é detalhado o modelo de rede utilizado no trabalho, a rede AMNN.

4 A Rede Neural em Modo de Atenção (AMNN)

4.1 Introdução

Este modelo de rede neural proposto por Paulo M. Engel [ENG 96] caracteriza-se pelo seu inovado conceito de adaptação. Esta nova metodologia, onde o aprendizado é feito enquanto a rede atua sobre o ambiente, dispensando a fase de treinamento, privilegia-a para a solução de problemas não-lineares em tempo real.

Seus elementos básicos também são os perceptrons, dispostos em multicamadas. A solução que utiliza os multi-níveis de perceptrons para resolver os problemas de classificação através de um treinamento supervisionado, já é bastante conhecida na rede BPN.

Para o funcionamento deste clássico algoritmo, a rede deve sofrer um treinamento prévio. Nesta fase é apresentado um conjunto de exemplos, compostos de valores de entradas e suas respectivas saídas para a solução do problema. A cada iteração, a rede vai calculando o erro gerado na saída, retropropagando-o para a camada oculta. Normalmente uma rede BPN possui três camadas: uma de entrada, uma oculta e outra de saída.

Devido à necessidade da fase de treinamento, sua aplicação só é bem sucedida em problemas os quais um conjunto de exemplos para treinamento pode ser bem definido; tal tipo de treinamento é chamado de supervisionado. Uma série de estudos que vem sendo realizados evidencia esta característica. Em alguns problemas, é quase impossível criar um conjunto de treinamento de forma a abranger todo o espaço das soluções, tanto pelo volume de amostras como pelo tempo necessário para o treinamento, tornando-se necessário estabelecer uma estratégia para a sua definição.

A semelhança com a rede BPN, na qual foi baseada, limita-se à arquitetura e à atualização dos pesos. Devido à inexistência da fase de treinamento, à medida que a rede vai atuando no ambiente, seu erro vai sendo calculado e retropropagado para a camada oculta, em tempo de reconhecimento, visando à atualização dos pesos (figura 23). A contribuição para o erro em cada unidade de processamento é calculada e usada para ajustar o seu respectivo peso. Isto a qualifica para a resolução de problemas não-lineares altamente complexos.

As características mais marcantes da rede AMNN são

- estrutura em camadas;
- a fase de treinamento é dispensável;
- adaptação constante;

- cálculo do erro é baseado no estado final e o momentâneo do sistema;
- a correção dos pesos da conexão baseia-se no gradiente descendente.

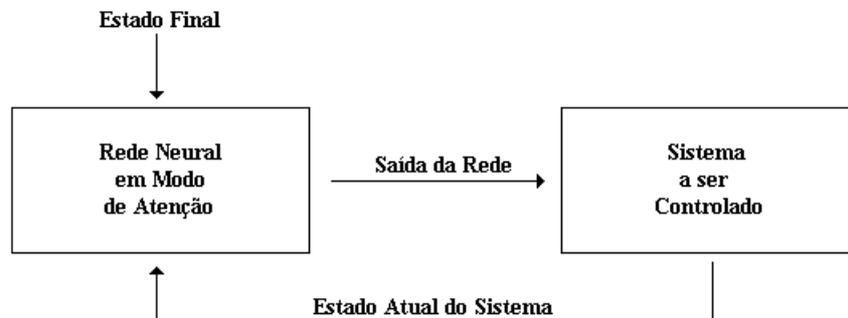


FIGURA 23 – Diagrama do Funcionamento da Rede AMNN.

Um problema clássico é o estacionamento de um veículo em uma plataforma de carga partindo de qualquer posição, como proposto por em [NGU 90] [ENG 96] e [COE 96]. Outros exemplos de aplicação são o posicionamento de um braço mecânico, abordado com sucesso em [FER 96] [SIL 97a] e mecanismos de controle de trajetória para robôs móveis apresentado por [BOT 96].

O presente trabalho visa aplicar a rede AMNN na robótica móvel, mais precisamente no controle da trajetória de um robô móvel em um ambiente não conhecido. Servirá como base para aplicação futura no robô móvel Nomad 200, recentemente adquirido pelo projeto interdisciplinar Robotec.

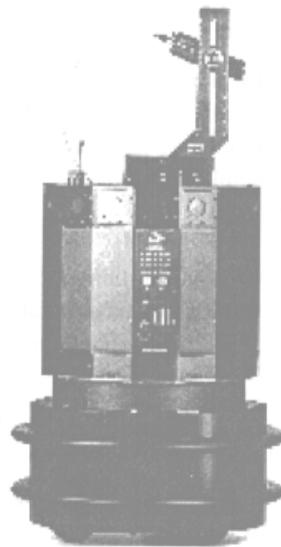


FIGURA 24 – O Nomad 200.

4.2 A Definição do Problema

Uma vez que o problema relativo à navegação em ambientes artificiais e sem obstáculo já tinha sido tratado usando a rede neural AMNN [ENG 96] [COE 96] com sucesso, passou-se a um segundo estágio.

Neste novo desafio, a rede deve ser capaz de controlar a navegação de um veículo em um ambiente artificial desconhecido e com obstáculos. Para tanto, foi necessária a incorporação de sensores para que o veículo fosse capaz de detectá-los.

Criou-se, então, dois deles (figura 25), seguindo os modelos mais utilizados atualmente na robótica móvel – apresentados anteriormente. O primeiro (SD – Sensor de Distância), situado no centro do veículo, é um sensor de distância capaz de detectar obstáculos (os numerados de O1 à O6 na figura 25) em qualquer direção; retorna uma leitura indicando o objeto, entre todos os existentes no ambiente, mais próximo do robô (identificada pela reta d_2 na figura).

O segundo (SDF: Sensor de Distância Focalizado), funciona de forma análoga, porém sua faixa de atuação está restrita à região frontal do veículo. Situado na frente do robô, possui um foco de leitura direcionado (região em cinza claro na figura 25). Também retornará a distância até o obstáculo – na figura abaixo, d_1 .

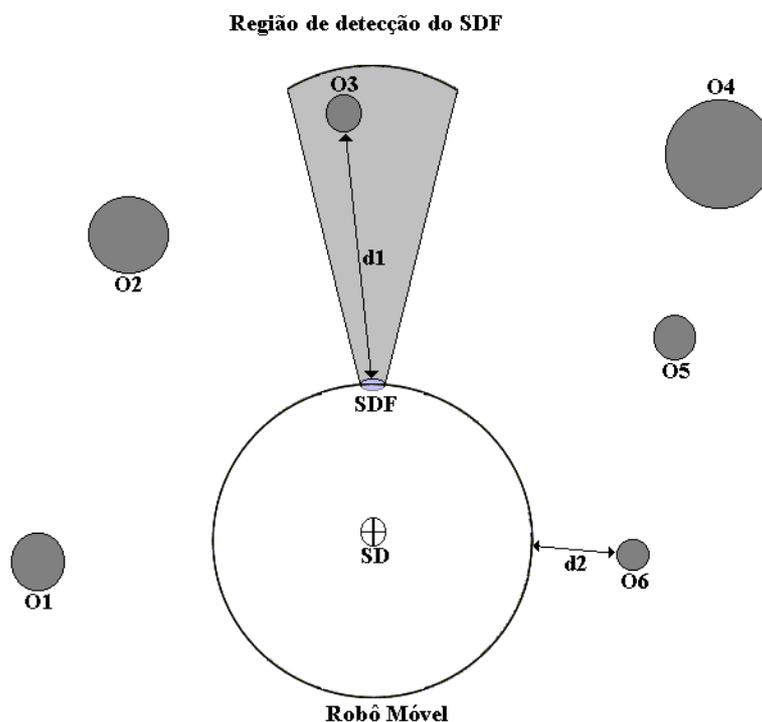


FIGURA 25 – A disposição dos sensores no robô virtual.

Os sensores possuem, individualmente, uma limitação para o alcance, ou seja, é determinado um valor para indicar um raio de ação. Os objetos ou obstáculos, para serem detectados, devem estar no raio de ação dos sensores. Neste trabalho, os obstáculos, todos circulares, são aleatoriamente distribuídos no ambiente ao iniciar-se a execução. Da mesma forma, o raio dos objetos varia em um intervalo previamente

estabelecido. A utilização de objetos circulares visa facilitar a simulação dos sensores e o cálculo das distâncias. Toda a simulação dos sensores é feita utilizando-se técnicas advindas da geometria analítica, como as equações de reta que passam por dois pontos, distância entre dois pontos, distância de um ponto a uma reta, etc., e também pela trigonometria.

No caso do sensor SD, o cálculo é simples: verifica-se a distância entre os dois centros em questão, ou seja, o afastamento entre o centro do robô e o do obstáculo. Tendo esta sido calculada, basta subtrair os raios do robô e do obstáculo e tem-se a real separação dos mesmos. O procedimento é efetuado para todos os obstáculos presentes no ambiente (de O1 até O6). Se a distância calculada for maior do que o alcance do sensor, o obstáculo é tido como invisível. O valor retornado pelo sensor é a menor distância calculada dentro do limite de “visibilidade” do sensor. Na figura 26, seria a distância d_2 , supondo-se que o objeto estivesse no alcance do sensor.

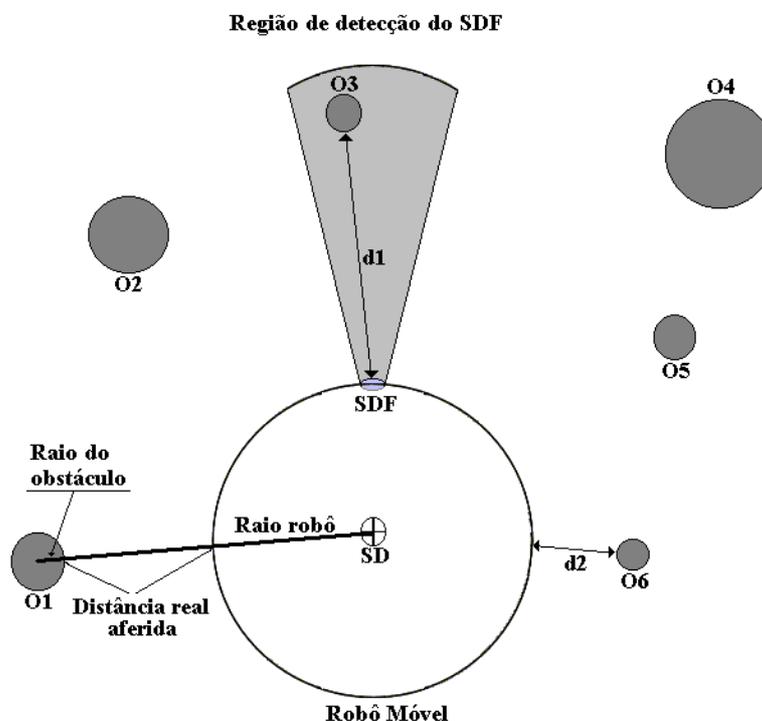


FIGURA 26 – Funcionamento do sensor de distância (SD).

Para o sensor SDF, o cálculo é um pouco diferente (figura 27). Veja os passos necessários:

- primeiramente monta-se a equação da reta que passa pelo centro do robô e pelo centro do sensor, a qual chamaremos de *reta de suporte*;
- o segundo passo é calcular a distância entre o centro do sensor e o centro do obstáculo;
- o terceiro passo, é calcular a distância entre o centro do obstáculo até a *reta de suporte*;
- agora, utilizando as relações trigonométricas, calcula-se o ângulo entre o

centro do obstáculo e o sensor. É feito dividindo-se o valor obtido no segundo passo (distância entre o centro do sensor e o centro do obstáculo) pelo obtido no terceiro passo (distância entre o centro do obstáculo até a *reta de suporte*). Com isso tem-se o seno do ângulo em questão. Basta usar, então, a função arco-seno e obtém-se o ângulo. Se este for menor ou igual ao raio de ação do sensor e a distância entre o obstáculo e o sensor estiver dentro do limite de alcance (d_1), então o obstáculo é dito “visível” (somente o objeto três – O3 – na figura 27); caso contrário, não será “notado” pelo sensor como ocorre com os demais objetos (O1, O2 e O4) na figura 27.

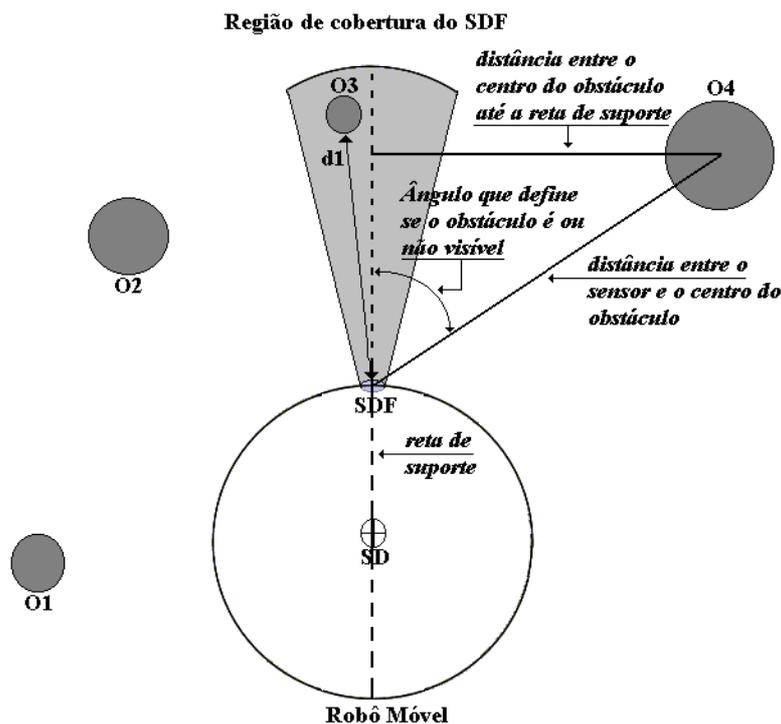


FIGURA 27 – Funcionamento do Sensor de Distância Focalizado (SDF).

Cabe salientar que, da mesma forma que no SD, aqui os raios do objeto e do robô são subtraídos da distância aferida, possibilitando a descoberta do real afastamento entre eles.

Se mais de um obstáculo estiver na região de cobertura do sensor, então este retornará a distância do que estiver menos afastado do robô.

O sensor SDF possui um alcance ligeiramente inferior ao do SD. Porém, sua influência no comportamento do robô tem prioridade ao SD, visto que detecta os obstáculos que estão à frente e também mais próximos do robô.

A maioria dos robôs móveis reais não possui somente dois sensores. No caso do Nomad, onde prevê-se a aplicação desta metodologia, existe uma série deles (veja mais detalhes no Anexo 1). Os sensores de ultra-som e também de infravermelho estão dispostos na periferia do robô, provendo uma cobertura de 360 graus ao redor do veículo. No entanto, podemos ativar somente aqueles que são de interesse, desligando os demais.

O objetivo, então, é fazer com que a rede AMNN encontre, incrementalmente,

ou seja, durante a sua atuação, um caminho livre de obstáculos que deverá levar o robô móvel até o seu objetivo, uma posição predeterminada dentro do espaço de trabalho, independente de sua posição de partida ou inicial. Esta posição é definida por três informações, quais sejam:

- a posição inicial do robô, dada pelas coordenadas (x_i e y_i);
- o ângulo formado pelo robô móvel com o plano horizontal (ϕ).

A posição atual do veículo móvel é indicada pelas coordenadas do seu centro. O controlador recebe estas entradas, juntamente com as leituras normalizadas dos sensores, e calcula um ângulo direcional que deverá ser aplicado às rodas do veículo para fazer com que este se aproxime do destino. Deverá, também, encontrar a função correta que mapeie os pontos tridimensionais dados pela entrada em uma variável de saída. Isto deve ser executado, ininterruptamente, desde o ponto inicial até o ponto final e para todos os possíveis pontos iniciais.

Neste tipo de problema é evidente a dificuldade na geração de um conjunto de exemplos de treinamento para uma rede neural, de forma a transformá-la em um controlador. Logo, as redes que necessitam de um treinamento prévio são completamente descartadas.

Outras formas de controle para esta tarefa já foram testadas, como sistemas *fuzzy*, algoritmos genéticos e redes neurais. Mesmo sendo usadas redes neurais neste trabalho, seu ineditismo advém da quantidade de redes. Normalmente se utiliza uma rede para desviar dos obstáculos e outra para a convergência até o objetivo, como em [SIL 97a]. Neste trabalho procurou-se realizar a mesma tarefa, porém com uma só rede neural.

Em um problema mais simples, o controle da navegação de um veículo móvel em um ambiente sem obstáculos, Kosko apresentou uma solução híbrida para o problema [KOS 92]. Inicialmente ele construiu um controlador baseado em regras *fuzzy* (FKBC - *Fuzzy Knowledge-Based Controller*) que foi capaz de solucionar o problema utilizando 35 regras. Logo após, ele utilizou estas trajetórias curvilíneas geradas pelo controlador *fuzzy* como sendo as trajetórias ideais para treinar uma rede *backpropagation*. Ele utilizou 35 vetores como exemplos de entrada. A rede necessitou mais de 100.000 iterações para ser treinada.

Vê-se nesta aplicação o quão difícil é a criação dos exemplos para treinamento. Kosko utilizou-se de uma outra técnica para solucionar o problema. Logo após, usufruindo as saídas geradas pelo controlador *fuzzy*, treinou a rede, que aprendeu somente a reproduzir os padrões de entrada, porém mantendo a capacidade de generalização. Entretanto a rede foi incapaz de distinguir entre as boas e as más soluções.

Uma outra solução foi proposta por Nguyen e Widrow [NGU 90] que apresentaram uma técnica de acúmulo de experiência chamada de *self learning* (ou auto-aprendizado). Nesta técnica a rede era inicializada randomicamente, o que também acontecia com a posição inicial. O veículo deslocava-se de ré durante um período determinado e parava. O erro era então computado e retropropagado para atualizar os pesos da rede. O processo era então repetido, colocando-se o carrinho em uma outra

posição inicial e deslocando-o novamente.

No entanto, estudos realizados por Kosko determinaram um alto custo computacional. Milhares de passos são necessários até que a rede aprenda. A rede *backpropagation* não convergiu para todos os exemplos de treinamento, tornando-se, portanto, inviável. Necessita-se, assim, de um outro tipo de rede. Uma opção é a rede AMNN.

4.3 A Rede AMNN Proposta

A rede AMNN não necessita da fase de treinamento, ou seja, não necessitamos definir um conjunto de exemplos. O aprendizado é feito enquanto a própria rede atua. O caminho que irá levar o carrinho até o seu destino é descoberto em apenas uma execução, pois a rede é mantida sempre no modo adaptativo. Os pesos são continuamente corrigidos baseados no erro momentâneo de cada ciclo de movimento. Assim, o sistema vai evoluindo, ajustando os seus pesos de acordo com o erro gerado.

Esta é a principal diferença entre as redes AMNN e *backpropagation*. Na rede AMNN ela é mantida em constante fase de aprendizado durante a evolução do sistema. Em cada iteração, o vetor que define a situação atual é comparado com o desejado. A partir disto, são calculados os valores para corrigi-los e, de forma similar à rede *backpropagation*, os pesos são atualizados. No entanto, esse cálculo leva em conta a contribuição individual de cada neurônio na composição do valor de saída. Logo após, são derivados os valores que irão corrigir os pesos.

Na verdade, o algoritmo consiste em uma seqüência de iterações, onde é executada uma adaptação para cada propagação realizada. A rotina de propagação, simplesmente recebe o vetor que indica a posição atual e calcula o ângulo direcional (ϕ), baseando-se nas expressões fundamentais dos perceptrons em multicamada.

Dada a geometria mostrada na figura 28, a posição atual, indicada por (x, y, ϕ) , é derivada a partir da posição anterior, denotada por $(x_{ant}, y_{ant}, \phi_{ant})$, utilizando-se equações cinemáticas simples em uma iteração a partir das fórmulas mostradas a seguir:

$$\phi = \phi_{ant} + \theta \quad (1a)$$

$$x = x_{ant} + (desloc * \cos(\phi)) \quad (1b)$$

$$y = y_{ant} + (desloc * \sin(\phi)) \quad (1c)$$

Onde *desloc* indica o quanto o robô móvel se deslocará em cada iteração da rede. Neste trabalho, *desloc* pôde variar entre 1um (um – unidade de medida) e 2um, dependendo da existência ou não de obstáculo(s) nas proximidades do sensor; *desloc* recebe o valor 1, quando for detectado um obstáculo nas proximidades do robô. Quando não mais existirem objetos próximos, sua velocidade retorna para 2.

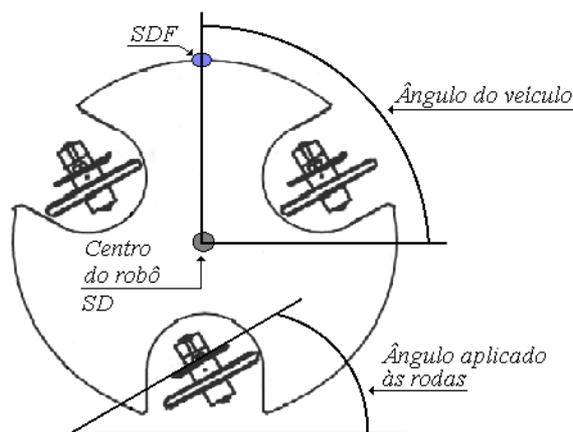


FIGURA 28 – A geometria do robô móvel simulado.

Para solucionar o problema foi utilizada uma rede multinível, composta de três camadas, conforme pode ser visto na figura 29:

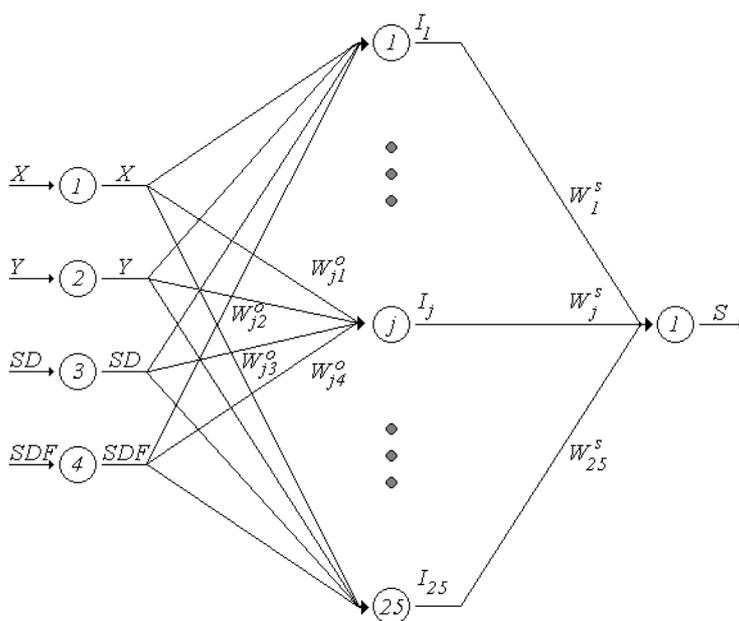


FIGURA 29 – A arquitetura da rede AMNN proposta.

- a camada de entrada possui 4 perceptrons (um elemento para cada entrada, posições x e y atuais e a leitura dos sensores SD e SDF);
- a oculta, com 25 perceptrons; e
- a de saída, com somente uma unidade (irá indicar o ângulo em que deve ser aplicado ao veículo para este se aproximar, até alcançar, o objetivo).

O vetor (x, y, sd, sdf) é fornecido para a camada de entrada da rede, que o distribui para as unidades da camada oculta. O valor net calculado para o elemento de índice j da camada oculta, net_j^o , é obtido a partir do somatório do produto de cada

entrada pelo seu respectivo peso, como mostra a fórmula:

$$net_j^o = \sum_{i=1}^N w_{ji} * x_i \quad (2a)$$

onde a letra ‘o’ sobrescrita indica a camada oculta; w_{ji} , representa o peso da conexão e x_i o valor de entrada do i -ésimo neurônio. A saída é calculada pela função f , a sigmoidal; a função de ativação de cada neurônio é a tangente hiperbólica ($f = \tanh(net)$), ou seja:

$$i_j^o = \tanh(net_j^o) \quad (2b)$$

Para a camada de saída, as equações são as seguintes:

$$net_k^s = \sum_{j=1}^N w_{kj} * i_j \quad (2c)$$

$$s_k = \tanh(net_k^s) \quad (2d)$$

O valor de todas as variáveis de entrada, incluindo os pesos (que são inicializados randomicamente), estão normalizados para o intervalo $[-1;+1]$. A saída da rede, com valores no mesmo intervalo anterior, representa o ângulo θ que indica o ajuste na direção a ser seguida pelo veículo. Logo após, este valor é normalizado para o intervalo $[-60^\circ;+60^\circ]$. O robô móvel Nomad não possui esta limitação de 60° . Suas rodas motrizes e direcionais podem girar livremente para qualquer lado.

4.4 O Aprendizado na Rede AMNN

O algoritmo de adaptação da rede AMNN é baseado na regra da descida mais íngreme chamada de erro médio quadrado, clássico do algoritmo *Backpropagation*. A rede leva em consideração o estado final e atual do sistema. A seguir é mostrado o erro médio quadrado.

$$E = \frac{1}{2} * \sum_{k=1}^n \delta_k^2 \quad (3a)$$

O valor assumido por $\delta_k = (y_k - s_k)$, onde a saída desejada é y_k e a saída atual é s_k .

A rede efetua seu aprendizado através da retropropagação dos erros, visando à atualização dos pesos. Esta atualização fará com que a rede forneça saídas que levem o robô em direção ao alvo, ou que o afaste dos obstáculos, quando estes forem detectados pelos seus sensores.

O cálculo do erro depende da situação em que se encontra o veículo naquele momento, ou seja, se existem obstáculos ou não, sendo detectados. Eis as situações possíveis:

- **Situação 1:**
nenhum objeto detectado pelos sensores: utiliza-se, então, o erro de posicionamento, procurando levar o robô até o destino. Os elementos δ_k , que são utilizados para o cálculo do erro nesta situação são:

$$\delta_1 = (x_f - x_{atual}) \quad (4a)$$

$$\delta_2 = (y_f - y_{atual}) \quad (4b)$$

onde (x_f, y_f) indicam a posição final desejada pelo sistema (alvo).

- **Situação 2:**
objeto detectado pelo sensor SDF, ou seja, o robô está detectando algum obstáculo em seu “campo de visão”. Neste caso, teremos somente um erro parcial, definido pelo inverso da leitura do sensor:

$$\delta_1 = \left(\frac{1}{sdf} \right) \quad (4c)$$

- **Situação 3:**
objeto detectado pelo sensor SD, ou seja, o robô está se aproximando de algum obstáculo. Neste caso utiliza-se uma função de penalização. Assim, teremos novamente um cálculo de erro parcial único, indicado pelo inverso do valor retornado pelo sensor:

$$\delta_1 = \left(\frac{1}{sd} \right) \quad (4d)$$

4.5 A Atualização dos Pesos da Camada de Saída

A adaptação dos pesos da camada de saída é dada pela seguinte equação:

$$w_{kj}^s(t+1) = w_{kj}^s(t) + \mu \nabla E \quad (5a)$$

onde:

- w_{kj}^s representa o peso que liga o neurônio k da camada de saída ao neurônio j da camada oculta;
- μ indica o coeficiente de aprendizado;
- ∇E : representa o gradiente descendente do erro gerado pela rede.

A equação (5a) representa o ajuste a ser aplicado nos pesos em relação aos erros gerados pela rede.

O gradiente descendente do erro ∇E é definido por:

$$\nabla E = -\frac{\partial E}{\partial w_{kj}^s} \quad (6a)$$

detalhando um pouco mais têm-se:

$$\nabla E = -\delta^s * \frac{\partial f_k^s}{\partial net_k^s} * \frac{\partial net_k^s}{\partial w_{kj}^s} \quad (6b)$$

onde δ^s irá variar com as situações 1, 2 e 3 apresentadas anteriormente. Desta forma, δ^s assumirá os seguintes valores:

- **Situação 1:**

$$\delta^s = \left((x_f - x_a) \frac{\partial x_a}{\partial \phi} + (y_f - y_a) \frac{\partial y_a}{\partial \phi} \right) \quad (6c)$$

- **Situação 2:**

$$\delta^s = -\delta_1 \left(\frac{\partial \delta_1}{\partial sdf} \right) \quad (6d)$$

- **Situação 3:**

$$\delta^s = -\delta_1 \left(\frac{\partial \delta_1}{\partial sd} \right) \quad (6e)$$

Detalhando a equação (6b), tem-se:

$$\frac{\partial f_k^s}{\partial (net_k^s)} = f_k^{s'} \text{ (derivada da função de ativação)} \quad (7a)$$

$$\frac{\partial (net_k^s)}{\partial w_{kj}^s} = i_j \text{ (saída do neurônio } j \text{ da camada oculta)} \quad (7b)$$

Combinando as equações apresentadas acima, temos o gradiente descendente como:

$$-\frac{\partial E}{\partial w_{kj}^s} = \delta^s * f_k^{s'}(net_k^s) * i_j \quad (8)$$

A seguir é apresentada a regra de atualização para as regras de atualização dos pesos da camada oculta.

4.5 A Atualização dos Pesos da Camada Oculta

A atualização dos pesos da camada oculta é feita de forma semelhante à da camada de saída. É efetuado pela seguinte equação:

$$w_{ji}^o(t+1) = w_{ji}^o(t) + \mu \nabla E \quad (9a)$$

onde:

- w_{ji}^o : representa o peso que liga o neurônio i da camada oculta ao neurônio j da camada de entrada;
- μ indica o coeficiente de aprendizado;
- ∇E : representa o gradiente descendente do erro gerado pela rede.

O gradiente descendente do erro ∇E é definido por:

$$\nabla E = - \frac{\partial E}{\partial w_{ji}^o} \quad (9b)$$

detalhando um pouco mais têm-se:

$$\nabla E = -\delta^s * \frac{\partial f_k^s}{\partial net_k^s} * \frac{\partial net_k^s}{\partial i_j^o} * \frac{\partial i_j^o}{\partial net_j^o} * \frac{\partial net_j^o}{\partial w_{ji}^o} \quad (9c)$$

Detalhando a equação (9c) tem-se:

$$\frac{\partial (f_k^s)}{\partial (net_k^s)} = f_k^{s'} \quad (\text{derivada da função de ativação na saída do neurônio } k) \quad (10a)$$

$$\frac{\partial (net_k^s)}{\partial (i_j^o)} = w_{kj}^s \quad (\text{saída do neurônio } j \text{ da camada oculta}) \quad (10b)$$

$$\frac{\partial (i_j^o)}{\partial (net_j^o)} = f_j^{o'} \quad (\text{derivada da função de ativação na saída do neurônio } j) \quad (10c)$$

$$\frac{\partial (net_j^o)}{\partial (w_{ji}^o)} = x_i \quad (\text{entrada da rede no neurônio } i) \quad (10d)$$

Das equações acima obtêm-se:

$$\nabla E = -\delta^s * f_k^{s'} * w_{kj}^s * f_j^{o'} * x_i \quad (11)$$

A seguir é apresentada a implementação da rede AMNN proposta.

4.6 Implementação do Modelo Proposto

A solução foi implementada utilizando a rede neural de modo de atenção, para resolver o problema proposto: conduzir o robô móvel até o seu destino.

Foi construído um ambiente de simulação para validar a proposição. O programa foi feito em linguagem C. Escolheu-se o ambiente PC devido à facilidade de utilização e também pela maior disponibilidade. O código do programa já está sendo portado para os ambientes Linux (ambiente de simulação e controle do Nomad) e também para o Windows 9X (em fase de conclusão).

Os obstáculos são criados aleatoriamente. Porém, para fazer com que os testes utilizassem sempre a mesma disposição dos obstáculos, estes foram gerados uma única vez e armazenados em arquivo para as futuras utilizações. Outro fator que agrava o processo de aleatoriedade da criação e posicionamento dos obstáculos é o fato da possibilidade do usuário do programa posicionar o robô próximo demais ou até mesmo “dentro” de um obstáculo, o que faz com que o sistema não atue corretamente.

Todos os pesos das interconexões da rede são inicializados aleatoriamente em um intervalo que pode ser definido em tempo de execução.

O sistema solicita que o usuário forneça a posição inicial do robô, o ângulo que este forma com o plano horizontal e também o coeficiente de aprendizado para a rede. Logo após, a rede começa a efetuar os cálculos visando levar o veículo móvel até o alvo situado em $(0,0)$, não importando com que ângulo este chegue ao objetivo.

Todo o resultado do cálculo é mostrado graficamente, com as sucessivas posições do robô durante a execução.

5 Resultados Obtidos

Neste capítulo são apresentados os resultados obtidos no controle neural adaptativo da trajetória do veículo móvel até o seu destino utilizando-se a rede neural AMNN. Estes testes visam à validação do sistema de acordo com critérios de convergência, estabilidade e repetitividade.

Inúmeros testes foram realizados para se poder atestar a funcionalidade e a eficiência da rede AMNN. Variou-se o intervalo para a inicialização dos pesos e também as taxas de aprendizado. A definição básica da rede para os testes foi esta:

- Número de neurônios na camada de entrada: 4
- Número de neurônios na camada oculta: 25
- Número de neurônios na camada de saída: 1
- Intervalo do ângulo fornecido pela rede para atualizar a posição do veículo: $[-60^{\circ}; +60^{\circ}]$.

Foram efetuados três tipos de testes com a rede:

- ambiente livre de obstáculos;
- com obstáculos fixos no ambiente;
- com obstáculos deslocando-se pelo ambiente.

Primeiramente será apresentado o estudo dos resultados da utilização da rede AMNN no controle da trajetória de um veículo móvel em um ambiente sem obstáculos.

5.1 Ambiente Livre de Obstáculos

Primeiramente buscou-se validar a rede para uma situação na qual o ambiente não apresentava nenhum obstáculo. A posição inicial selecionada foi a (150;150), pois, na situação com obstáculos fixos, partindo desta posição inicial, a rede terá de desviar de diversos objetos para fazer com que o veículo atinja o alvo. Assim poderemos comparar as trajetórias selecionadas pela rede.

Foram feitas 50 simulações para cada situação. Em cada uma delas, os pesos eram inicializados a cada nova execução. Os pesos variaram nos intervalos de $[-1,0;1,0]$, $[-0,75;0,75]$, $[-0,5;0,5]$, $[-0,25;0,25]$ e $[-0,1;0,1]$.

Para cada faixa de inicialização dos pesos, foram utilizados cinco valores possíveis para a taxa de aprendizado da rede (0,05; 0,10; 0,15; 0,20 e 0,25). Assim,

pôde-se constatar uma relação entre os valores iniciais para os pesos e a taxa de aprendizado, cujos resultados são apresentados na tabela 1.

TABELA 1 – Resultados das simulações para o ambiente sem obstáculos.

Situação	Taxa de Aprendizado (μ)				
	0,05	0,10	0,15	0,20	0,25
Pesos entre [-0,1;0,1]					
Menor número de passos	185,00	169,00	160,00	161,00	159,00
Média de passos	206,50	181,00	172,50	168,00	166,00
Maior número de passos	2610,00	217,00	1121,00	203,00	180,00
Nº de simulações que não atingiu o alvo	6,00	0,00	0,00	0,00	0,00
Percentual de acerto (encontrou o alvo)	88,00%	100,00%	100,00%	100,00%	100,00%
Pesos entre [-0.25;0.25]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	158,00	156,00	158,00	158,00	160,00
Média de passos	161,50	159,00	162,00	167,00	172,00
Maior número de passos	173,00	162,00	181,00	185,00	207,00
Nº de simulações que não atingiu o alvo	0,00	0,00	0,00	0,00	0,00
Percentual de acerto (encontrou o alvo)	100,00%	100,00%	100,00%	100,00%	100,00%
Pesos entre [-0.5;0.5]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	155,00	156,00	157,00	165,00	167,00
Média de passos	160,00	185,00	194,00	220,50	284,00
Maior número de passos	184,00	334,00	3980,00	1124,00	4251,00
Nº de simulações que não atingiu o alvo	0,00	0,00	0,00	0,00	1,00
Percentual de acerto (encontrou o alvo)	100,00%	100,00%	100,00%	100,00%	98,00%
Pesos entre [-0.75;0.75]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	155,00	168,00	167,00	177,00	176,00
Média de passos	197,04	400,59	677,74	1016,39	723,81
Maior Número de passos	596,00	2028,00	4938,00	4965,00	3521,00
Nº de simulações que não atingiu o alvo	0,00	1,00	4,00	12,00	14,00
Percentual de acerto (encontrou o alvo)	100,00%	98,00%	92,00%	76,00%	72,00%
Pesos entre [-1,0;1,0]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	155,00	161,00	192,00	179,00	185,00
Média de passos	218,00	338,50	396,50	277,00	1603,00
Maior número de passos	2985,00	3897,00	3304,00	2976,00	4730,00
Nº de simulações que não atingiu o alvo	0,00	6,00	12,00	17,00	29,00
Percentual de acerto (encontrou o alvo)	100,00%	88,00%	76,00%	66,00%	42,00%

Cabe salientar que as médias das execuções que aparecem na tabela não consideram as execuções em que a rede não conseguiu atingir o alvo. A execução era encerrada em duas situações:

- quando a rede atingia o objetivo;
- quando a rede excedia o limite de 5000 iterações, independente da posição

do veículo móvel.

O gráfico da figura 30 serve para ilustrar a distribuição do número de passos em cada uma das 50 simulações efetuadas para as diferentes taxas de aprendizado. Nele, percebe-se claramente que a taxa de aprendizado igual a 0,1 obtém melhores resultados.

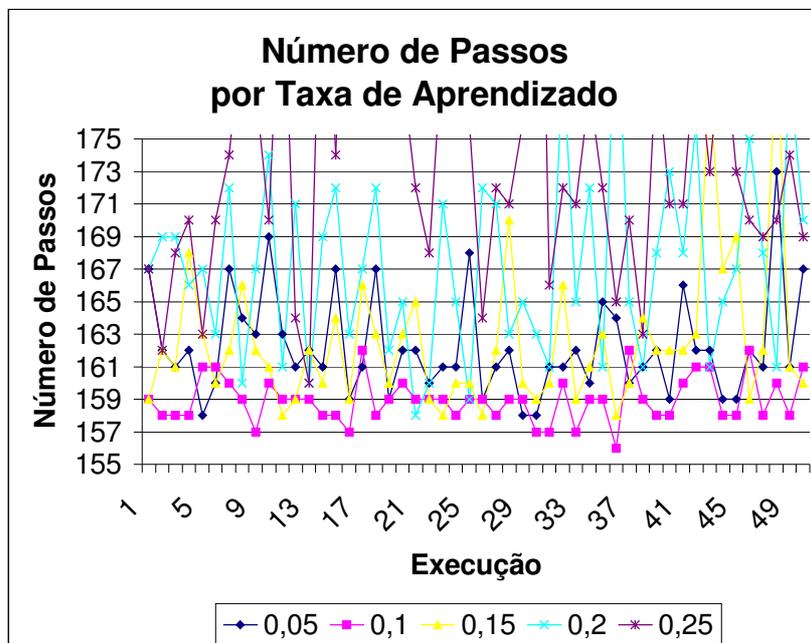


FIGURA 30 – Número de passos por taxa de aprendizado nas 50 simulações.

A figura 31 mostra as rotas criadas pela rede durante a execução das 50 simulações realizadas partindo da posição (150;150), com os pesos variando entre $[-0,25;0,25]$ e o coeficiente de aprendizado valendo 0,1. Mesmo com a inicialização aleatória dos pesos, a rede convergiu rapidamente e mostrou uma grande repetitividade.

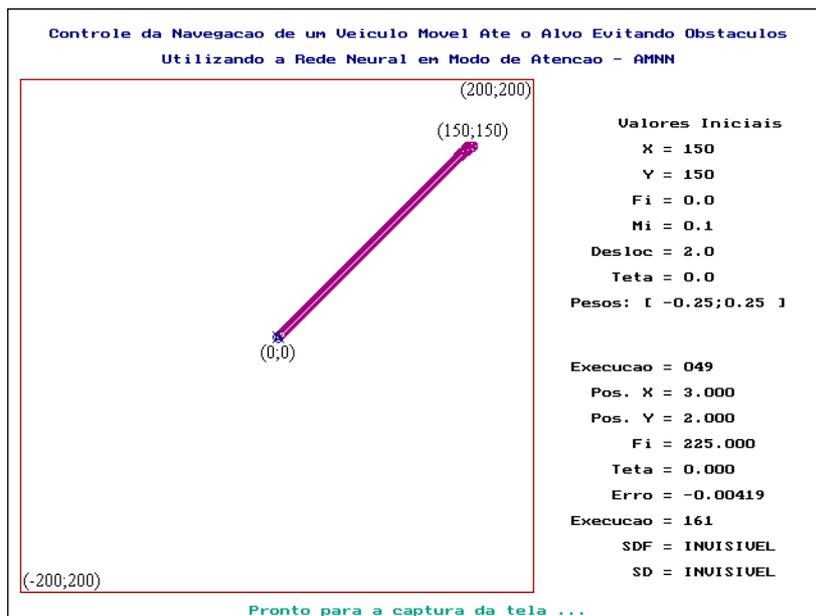


FIGURA 31 – Região gerada pelas 50 simulações sem obstáculos, pesos inicializados entre $[-0,25;0,25]$ e taxa de aprendizado de 0,1.

Ao se utilizar pesos mais elevados, próximos de $[-1;1]$, e coeficientes de aprendizado mais altos, a rede tornava-se muito instável e reduzia a sua eficiência, conforme pode ser visto na figura 32.

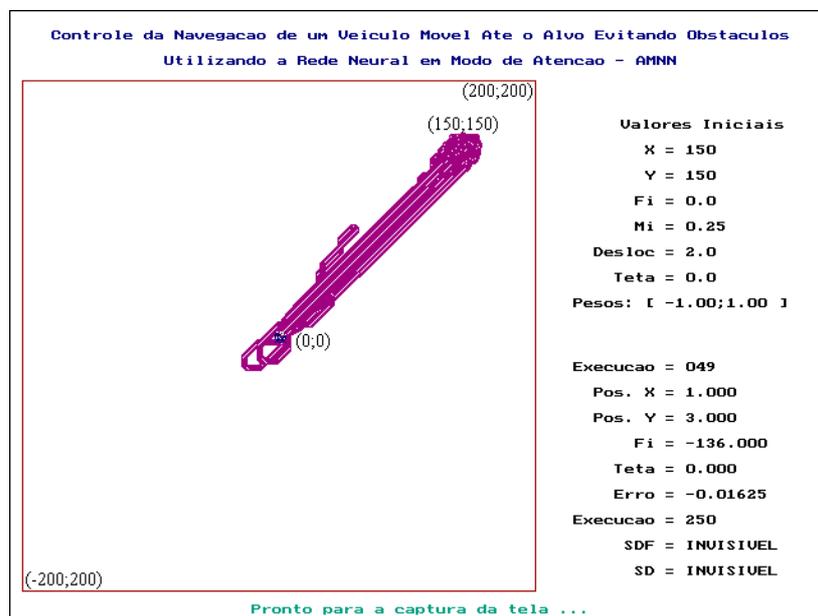


FIGURA 32 – Região gerada pelas 50 simulações sem obstáculos, pesos inicializados entre $[-1,0;1,0]$ e taxa de aprendizado de 0,25.

Os melhores resultados foram com os pesos sendo inicializados no intervalo $[-0,25;0,25]$. Com esta faixa de pesos iniciais, todas as taxas de aprendizado obtiveram sucesso na solução do problema, e com índices bastante parecidos. Porém, o que utilizou uma taxa de aprendizado de 0,1 mostrou ser mais eficiente, pois obteve os menores valores nos quesitos de menor número de passos, menor média e maior número de passos necessários para alcançar o alvo.

Quando se utiliza um intervalo pequeno para a inicialização dos pesos, os melhores resultados são obtidos com coeficientes de aprendizado mais altos. Com taxas de aprendizado pequenas, a rede consegue convergir, mas esta é obtida de forma mais lenta, como pode ser visto na figura 33.

Em muitos casos, a rede estava convergindo em direção ao alvo quando a execução era abortada, em decorrência do limite máximo de execuções (5000) ter sido atingido.

Com base nos testes realizados e apresentados anteriormente, escolheu-se a configuração que utilizava o intervalo de inicialização dos pesos entre $[-0,25;0,25]$ e um coeficiente de aprendizado igual a 0,1. Feito isto, passou-se aos testes de convergência e repetitividade da configuração escolhida.

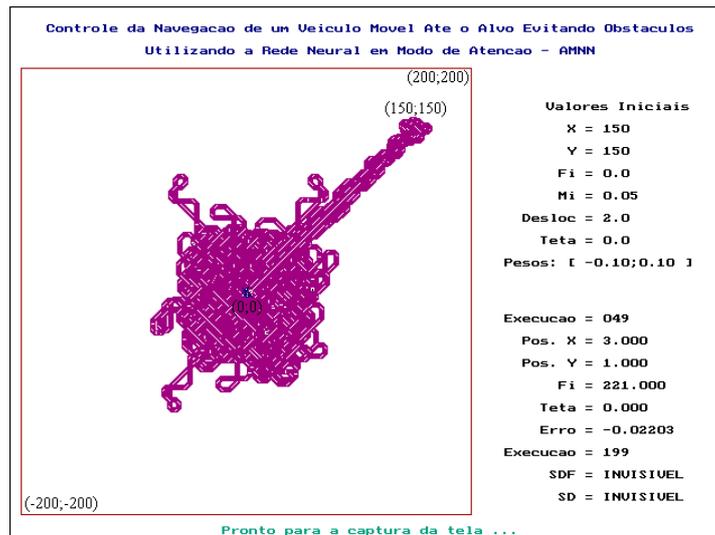


FIGURA 33 – Região gerada pelas 50 simulações sem obstáculos, pesos inicializados entre $[-0,1;0,1]$ e taxa de aprendizado de 0,05.

5.1.1 Convergência da Rede AMNN em Ambiente Livre de Obstáculos

Para demonstrar a convergência da rede AMNN em ambientes sem obstáculos foram realizadas 50 simulações. A posição inicial de cada simulação foi definida aleatoriamente, respeitando-se o intervalo $[-200;200]$.

Os valores da taxa de aprendizado e o intervalo para a inicialização dos pesos mantiveram-se constantes e iguais a 0,1 e $[-0,25;0,25]$ respectivamente.

A figura 34 apresenta as trajetórias seguidas pelo veículo e serve para concluir que a rede proposta atende plenamente ao objetivo de guiar o veículo móvel até o alvo partindo de qualquer posição inicial do espaço de atuação sem obstáculos.

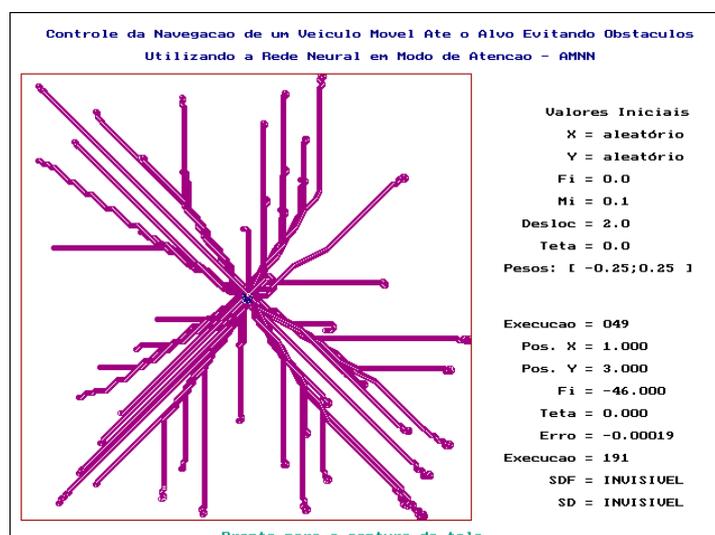


FIGURA 34 – A Convergência da rede AMNN em ambiente livre de obstáculos.

5.1.2 A Capacidade de Repetição da rede AMNN em Ambientes Livres de Obstáculos

Para demonstrar a repetitividade dos resultados obtidos na utilização da rede AMNN, foram executadas várias execuções, cada qual contendo 50 simulações. Em cada uma delas, a posição inicial foi definida aleatoriamente no ambiente de trabalho $[-200;200]$.

Também os pesos foram aleatoriamente inicializados com valores pertencendo ao intervalo $[-0,25;0,25]$. Uma vez gerados os pesos iniciais, estes foram armazenados em arquivo para que todas as 50 execuções realizadas utilizassem os mesmos valores.

A taxa de aprendizado manteve-se constante em 0,1.

Escolheu-se, dentre as posições iniciais geradas, somente 4 delas. Cada uma das imagens escolhidas ilustra a posição inicial pertencendo a um quadrante diferente.

A figura 35 (a, b, c e d) comprova a repetitividade dos resultados da rede AMNN. Nos testes, os números gerados pela rede em cada uma das 50 execuções foram rigorosamente iguais.

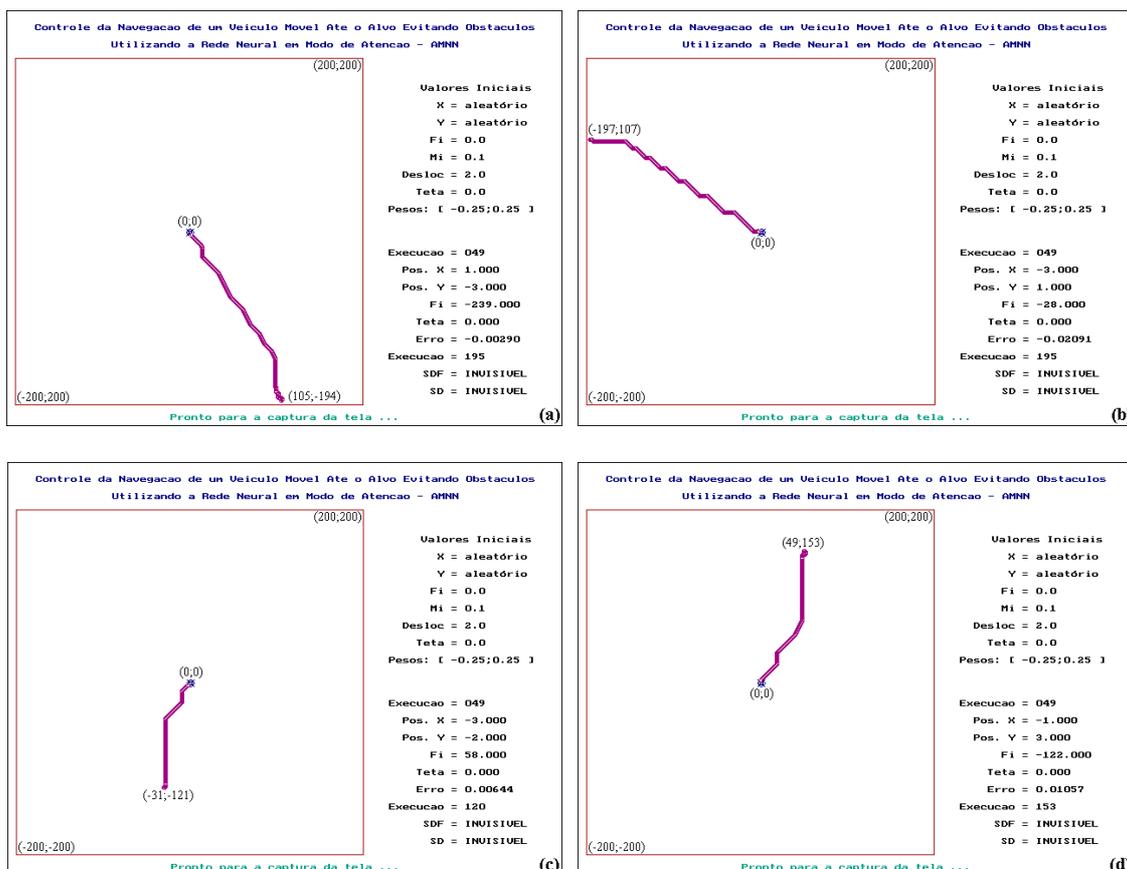


FIGURA 35 – A repetitividade dos resultados com a rede AMNN em um ambiente livre de obstáculos.

5.1.3 A Estabilidade da Rede AMNN em Ambientes Livres de Obstáculos

Todos os testes realizados com a rede AMNN proposta, com o intervalo para os valores da inicialização dos pesos e para a taxa de aprendizado escolhido, obtiveram sucesso em gerar uma trajetória que levasse o veículo móvel até o alvo, não importando a posição de partida.

Algumas vezes, a rede oscilava um pouco ao iniciar a execução, porém rapidamente se estabilizava e convergia para o objetivo.

Os resultados obtidos nos testes serviram para confirmar que a rede AMNN consegue obter pleno êxito no controle da trajetória de um veículo móvel em ambientes livres de obstáculos. Passou-se então aos testes para validar a utilização da mesma rede para resolver o problema do controle da trajetória do veículo móvel quando em ambientes com obstáculos.

5.2 Ambiente com Obstáculos Fixos

Os testes realizados para validar a rede AMNN proposta para ambientes com obstáculos fixos foram efetuados seguindo as mesmas premissas das execuções efetuadas anteriormente.

Os obstáculos presentes no ambiente são todos circulares. As coordenadas do centro dos círculos, e também o raio de cada um deles, foram gerados aleatoriamente uma única vez e armazenados em arquivo para que o ambiente fosse o mesmo em todos os testes. O resultado pode ser visto na figura 36.

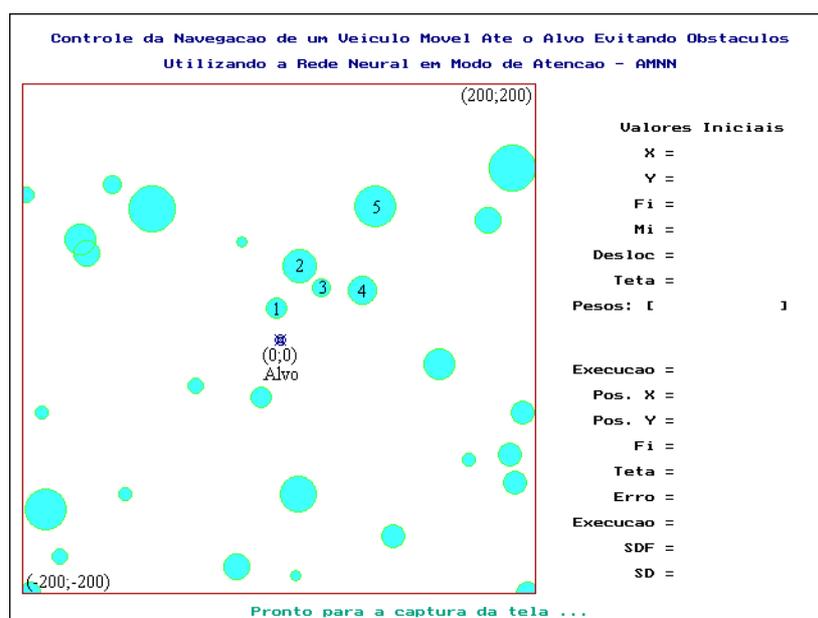


FIGURA 36 – Disposição dos obstáculos no ambiente para as simulações.

Foram feitas 50 simulações para cada situação. A posição inicial escolhida foi a coordenada (150;150) devido a presença de vários obstáculos no caminho até o alvo (no mínimo os que aparecem numerados na figura 36). Em cada uma delas, os pesos eram inicializados a cada nova execução. Para cada grupo de simulações foram utilizados intervalos delimitadores para os pesos. São eles: $[-1,0;1,0]$, $[-0,75;0,75]$, $[-0,5;0,5]$, $[-0,25;0,25]$, $[-0,2;0,2]$ e $[-0,1;0,1]$.

Para cada faixa de inicialização dos pesos, foram utilizados cinco valores possíveis para a taxa de aprendizado da rede (0,05, 0,10, 0,15, 0,20 e 0,25). Assim, pôde-se constatar uma relação entre os valores iniciais para os pesos e a taxa de aprendizado, cujos resultados são apresentados na tabela 2.

De posse dos valores obtidos, foi possível escolher a melhor configuração para a rede neural artificial visando o tratamento de obstáculos fixos no ambiente enquanto guia o veículo móvel até o alvo. Os melhores resultados surgiram com os pesos sendo inicializados entre $[-0,2;0,2]$ e com uma taxa de aprendizado de 0,15.

A simulação era encerrada em duas situações: quando o veículo móvel atingia o destino (alvo – posição cartesiana (0;0)), ou ao ser atingido o limite de 5000 passos.

Na tabela 2, o cálculo da média de passos leva em consideração somente as execuções nas quais o alvo foi atingido, ou seja, excluíram-se do cálculo as simulações nas quais a rede não alcançou o seu objetivo.

Similarmente à simulação em ambientes sem obstáculos, algumas vezes a rede estava convergindo em direção ao alvo, porém o limite de 5000 passos era atingido e a execução encerrada.

Partindo da coordenada (150;150) a rota mais rápida até o alvo seria uma semi-reta com uma inclinação de 45 graus.

A figura 37 apresenta a região de rotas geradas em uma das muitas simulações efetuadas para gerar os resultados visando construir a tabela 2.

A figura 38, pelo contrário, mostra a região gerada pela pior configuração da tabela 2.

Todos os obstáculos e o robô simulado são circulares, posto que o robô móvel real onde se prevê a aplicação desta proposta é circular e também para facilitar o tratamento dos sensores. Como já foi relatado anteriormente, o raio dos obstáculos é definido aleatoriamente (mínimo 2 e máximo 10um).

Foram simulados dois sensores, SD e SDF, para auxiliar a navegação, como explicados no capítulo anterior. Torna-se relevante afirmar que os raios de ação dos sensores influenciam o comportamento da rede, visto que fazem parte do cálculo do erro. Eles definem uma região de “atenção” que o veículo deve manter para conseguir evitar os obstáculos. No entanto, existe uma ligação entre a sensibilidade dos sensores e o raio do veículo móvel.

Da mesma forma que se determina o alcance dos sensores, pode-se definir a distância mínima permitida entre o veículo e os obstáculos. Se a reduzirmos, o veículo móvel contornará os objetos presentes no ambiente com uma maior proximidade. O

inverso acontece quando se aumenta o valor para a distância mínima.

TABELA 2 – Resultados das simulações para o ambiente com obstáculos fixos.

Situação	Taxa de Aprendizado (μ)				
	0,05	0,10	0,15	0,20	0,25
Pesos entre [-0,1;0,1]					
Menor número de passos	514,00	219,00	218,00	215,00	218,00
Média de passos	950,00	530,50	451,00	329,00	290,00
Maior número de passos	4783,00	3351,00	2913,00	1769,00	1128,00
Nº de simulações que não atingiu o alvo	3,00	2,00	0,00	0,00	0,00
Percentual de acerto (encontrou o alvo)	94,00%	96,00%	100,00%	100,00%	100,00%
Pesos entre [-0,2;0,2]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	218,00	218,00	203,00	210,00	213,00
Média de passos	301,50	251,00	239,50	247,50	247,00
Maior número de passos	916,00	342,00	334,00	692,00	1460,00
Nº de simulações que não atingiu o alvo	0,00	0,00	0,00	0,00	0,00
Percentual de acerto (encontrou o alvo)	100,00%	100,00%	100,00%	100,00%	100,00%
Pesos entre [-0,25;0,25]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	228,00	209,00	206,00	213,00	212,00
Média de passos	256,00	240,00	237,50	252,00	265,00
Maior número de passos	430,00	826,00	620,00	2265,00	1123,00
Nº de simulações que não atingiu o alvo	0,00	0,00	0,00	0,00	0,00
Percentual de acerto (encontrou o alvo)	100,00%	100,00%	100,00%	100,00%	100,00%
Pesos entre [-0,5;0,5]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	214,00	224,00	214,00	249,00	250,00
Média de passos	264,00	329,00	307,50	413,00	500,00
Maior número de passos	1971,00	1129,00	1242,00	1841,00	3942,00
Nº de simulações que não atingiu o alvo	0,00	0,00	0,00	0,00	0,00
Percentual de acerto (encontrou o alvo)	100,00%	100,00%	100,00%	100,00%	100,00%
Pesos entre [-0,75;0,75]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	225,00	233,00	255,00	297,00	247,00
Média de passos	438,94	505,11	766,69	886,91	1303,91
Maior número de passos	4400,00	1315,00	2920,00	4956,00	3804,00
Nº de simulações que não atingiu o alvo	0,00	5,00	2,00	7,00	17,00
Percentual de acerto (encontrou o alvo)	100,00%	90,00%	96,00%	86,00%	66,00%
Pesos entre [-1,0;1,0]	0,05	0,10	0,15	0,20	0,25
Menor número de passos	235,00	247,00	339,00	262,00	334,00
Média de passos	372,00	587,00	734,00	838,00	831,00
Maior número de passos	1272,00	3675,00	3797,00	4652,00	3175,00
Nº de simulações que não atingiu o alvo	3,00	14,00	14,00	27,00	27,00
Percentual de acerto (encontrou o alvo)	94,00%	72,00%	72,00%	46,00%	46,00%

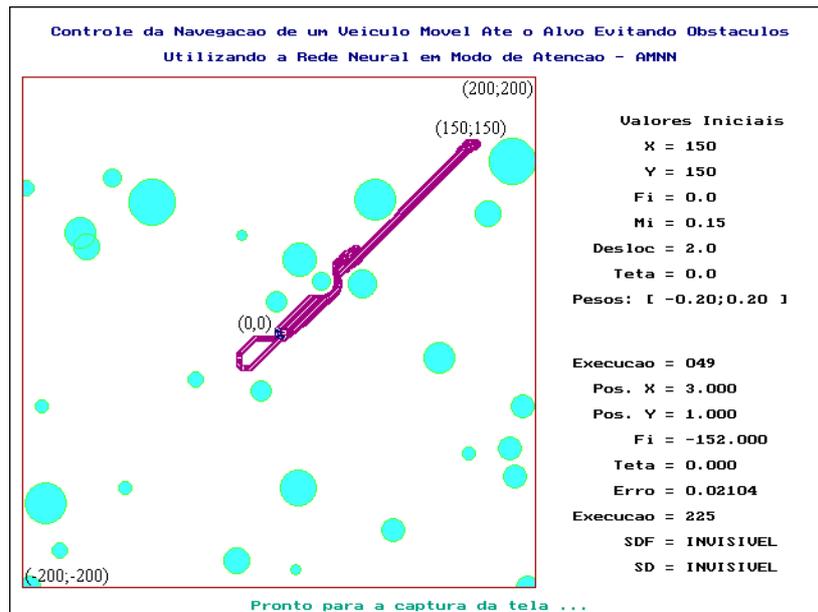


FIGURA 37 – Região gerada pelas 50 execuções consecutivas com $\mu=0,15$ e os pesos inicializados com valores pertencendo ao intervalo $[-0,2;0,2]$.

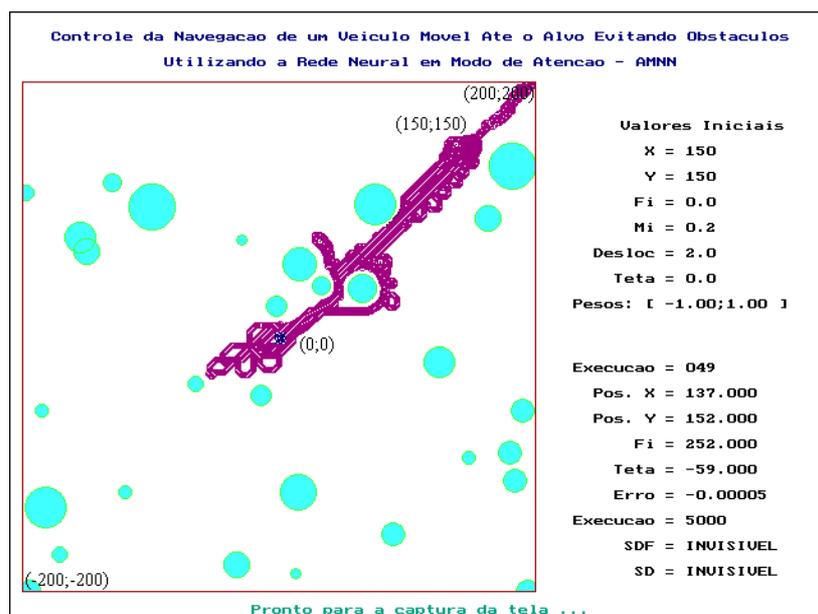


FIGURA 38 – Região gerada pelas 50 execuções consecutivas com $\mu=0,2$ e os pesos inicializados com valores pertencendo ao intervalo $[-1,0;1,0]$.

Nos testes efetuados buscou-se definir uma relação entre o raio do veículo móvel e a distância mínima. Para o veículo com raio igual a 2um, as melhores configurações para os sensores foram:

- alcance máximo: 3um para o SD e 1,5um para o SDF
- distância mínima aceitável (região de “atenção”): $\frac{\text{raio_do_rob\^o_m\^ovel}}{3}$

Se a “visibilidade” do sensor for alta, o veículo não chegará tão próximo dos

obstáculos. Porém, não conseguirá passar por corredores, como o formado pelos objetos 3 e 4 na figura 39(a) e (b). Por outro lado, reduzindo-se o poder de detecção dos sensores, corre-se o risco do veículo móvel tocar os obstáculos.

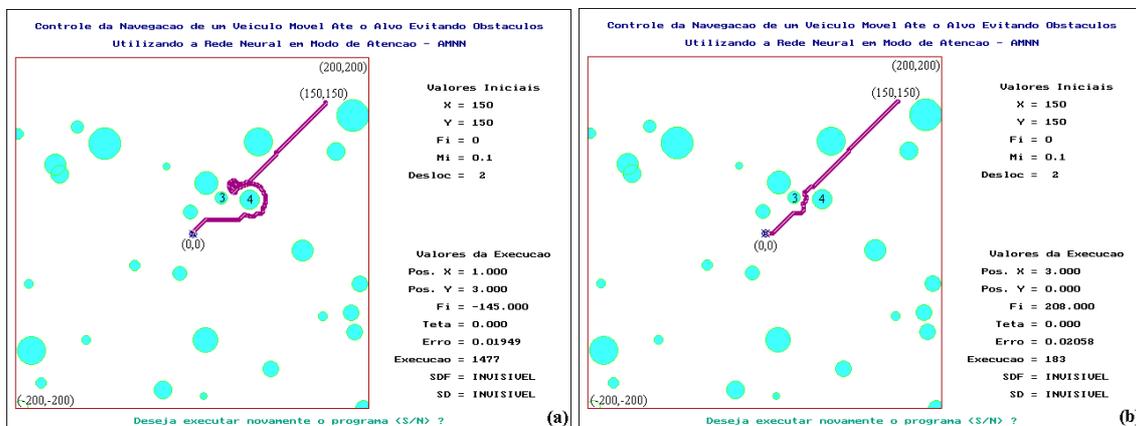


FIGURA 39 – Exemplos da influência do alcance dos sensores na navegação.

A figura 39(a) mostra a simulação na qual o alcance dos sensores SD e SDF foi ajustado para 4um e 2um, respectivamente. Nota-se que a rede demorou um pouco para encontrar uma rota alternativa para alcançar o alvo, ou seja, contornar o obstáculo de número 4.

Já na figura 39(b), na qual o alcance dos sensores SD e SDF foi ajustado para 2um e 1um, respectivamente, os obstáculos foram percebidos um pouco mais tarde e também devido a redução na área de “atenção”, a rede conseguiu guiar o veículo móvel através do corredor formado pelos obstáculos 3 e 4, reduzindo o trajeto.

Na versão atual do programa, a configuração dos sensores é feita em tempo de compilação. O sistema está sendo portado para o ambiente Windows e, nesta nova versão, a configuração do ambiente poderá ser feita em tempo de execução, uma vez que a construção e a manipulação da interface com o usuário naquele ambiente é um processo mais simples do que no modo gráfico do DOS. A implementação atual mostrou-se suficiente para validar a proposta.

A seguir são apresentadas as figuras 40 e 41 ilustrando resultados de outras execuções, onde a posição inicial foi mantida constante em (0;150) e os valores de alcance dos sensores foram alterados, como mostram as legendas das figuras.

A modificação da coordenada inicial implicou a alteração dos obstáculos presentes na rota a ser percorrida até atingir o alvo (os de número 1, 2, 3 e 4).

Os exemplos de execução mostrados na figura 42 e figura 43 utilizam os mesmos valores de alcance para os sensores SD e SDF, porém a rede descobriu um caminho alternativo para alcançar o alvo, decorrente da diferente inicialização dos pesos. Nota-se também a presença dos 4 obstáculos que aparecem na trajetória do veículo (numerados de 1 a 4 nas figuras).

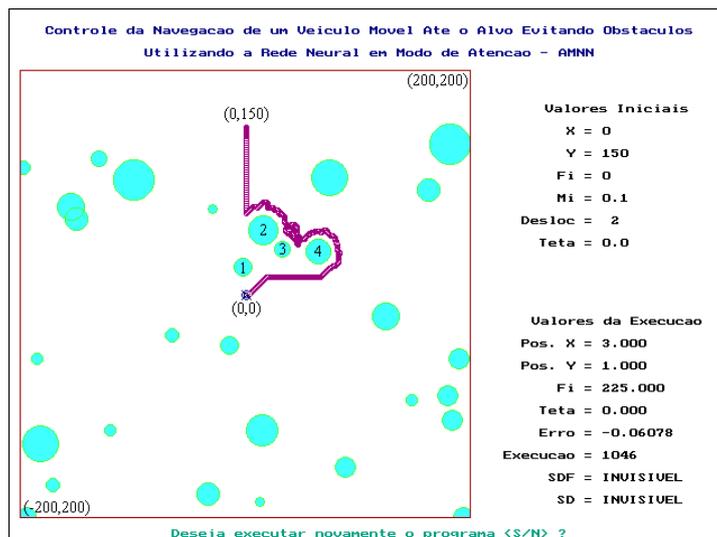


FIGURA 40 – Simulações com alcance: $SD=4\mu m$ e $SDF=2\mu m$ e partindo de $(0,150)$.

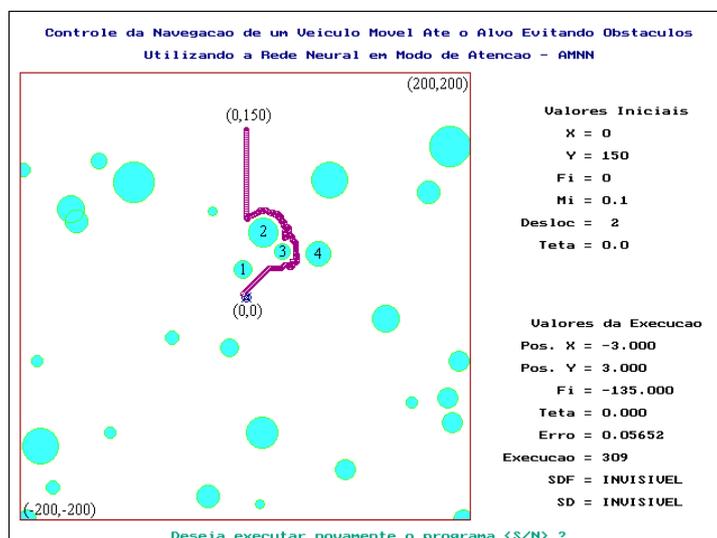


FIGURA 41 – Simulações com alcance: $SD=2\mu m$ e $SDF=1\mu m$ e partindo de $(0,150)$.

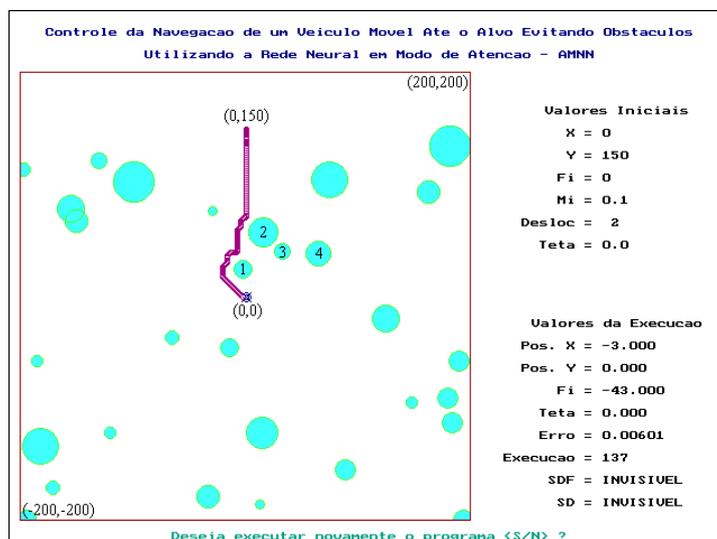


FIGURA 42 – Simulações com alcance: $SD=4\mu m$ e $SDF=2\mu m$ e partindo de $(0,150)$.

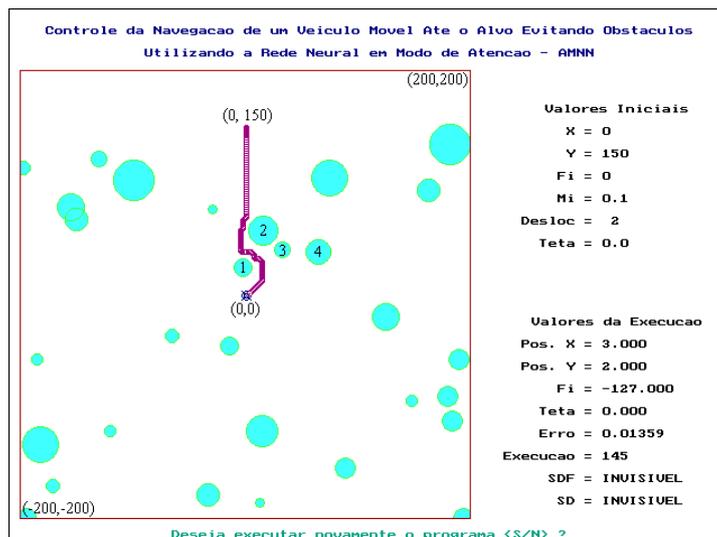


FIGURA 43 – Simulações com alcance: $SD=2um$ e $SDF=1um$ e partindo de $(0,150)$.

A rede consegue contornar obstáculos e guiar o veículo até o alvo na grande maioria das vezes. Em algumas delas, um percentual bastante reduzido, a rede não converge, como pode ser visto na figura 44.

Constatou-se que isto ocorre devido aos valores de inicialização dos pesos das interconexões dos neurônios na rede, que é feita de forma aleatória. Foram efetuados testes com os pesos variando em diversos intervalos. Os valores máximos para os pesos devem estar entre $[-1;1]$. Porém, a rede comportou-se melhor quando o intervalo foi restringido para $[-0,20;0,20]$, atingindo o alvo em 100% das execuções.

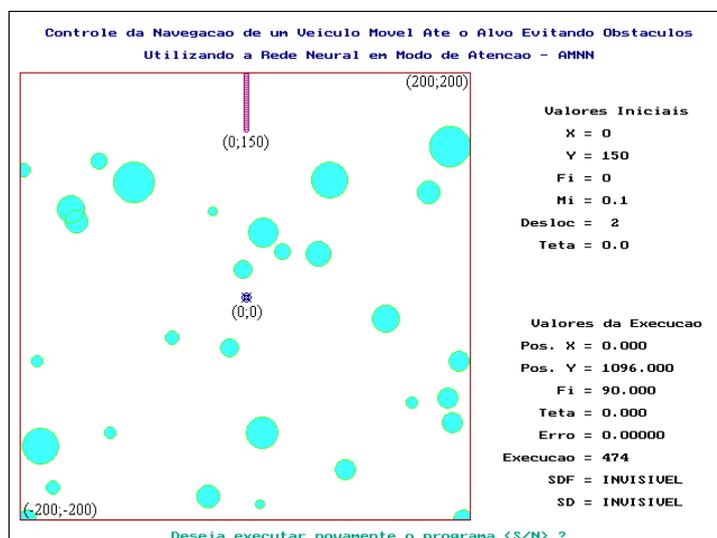


FIGURA 44 – Exemplo da não convergência da rede AMNN partindo de $(0,150)$.

Às vezes a rede demora em convergir, mesmo assim consegue atingir o alvo, como pode ser visto na figura 45.

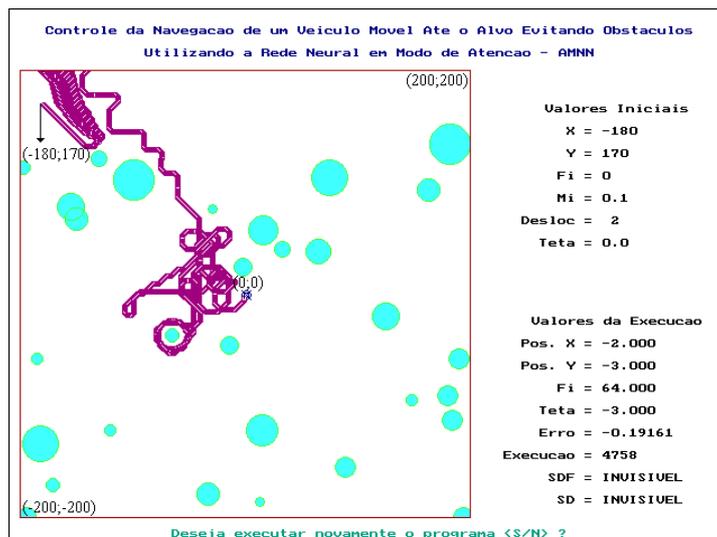


FIGURA 45 – Um dos piores casos encontrados partindo de (-180,170).

A quantidade de neurônios na camada oculta pode ser considerada pequena para a complexidade do problema em questão. Outras quantidades foram consideradas nas simulações (20, 30, 40, 50, 60, 80); porém, quanto mais neurônios se punham na camada oculta, mais oscilações eram obtidas na saída da rede, considerando o mesmo valor para o coeficiente de aprendizado.

Constatou-se que existe uma relação inversa entre a quantidade de neurônios na rede e o coeficiente de aprendizado. Se o número de neurônios for incrementado, o coeficiente de aprendizado deve ser decrementado.

Os melhores resultados em termo de estabilidade dos ângulos para a correção da rota do veículo foram obtidos com quantidades pequenas, entretanto nunca menores do que 20.

A sensibilidade dos sensores também é importante. Os melhores resultados foram aqueles que possibilitavam que o veículo pudesse passar por locais mais estreitos, ou seja, quando se diminuía o alcance dos sensores. Porém deve-se ter um cuidado para não reduzi-los demasiadamente, caso contrário a rede não terá condições de se recuperar.

5.2.1 Convergência da Rede AMNN em Ambiente Contendo Obstáculos Fixos

Como se sabe, as redes neurais artificiais são muito suscetíveis aos seus parâmetros de configuração. Analisando-se a tabela 2, verifica-se que, respeitando alguns parâmetros (intervalo para inicialização dos pesos e também o coeficiente de aprendizado), a rede obtém valores próximos de 100% de convergência para o alvo. As raras exceções, como a apresentada anteriormente na figura 45, ficam por conta dos valores iniciais gerados para os pesos.

Algumas vezes a rede encontra regiões “atratoras” no ambiente, de onde demora

algum tempo para sair. Com isto, a convergência acontece, mas de forma muito lenta. Nos testes efetuados, limitou-se o número máximo de passos por simulação em 5000 simplesmente devido ao tempo de execução e ao número de testes a serem realizados.

Porém, em outras simulações sem limites de passos, a rede neural artificial ficava executando por horas, até que “descobria” uma maneira de se desvencilhar deste mínimo local, conduzindo o veículo móvel até o seu destino.

Para executar 100 passos com tratamento dos sensores a rede dispndia aproximadamente 2s em uma máquina com processador AMD K6-2 de 266MHz. Já sem obstáculos, demorava algo em torno de 1s para executar o mesmo número de repetições.

5.2.2 A Capacidade de Repetição da rede AMNN em Ambientes Contendo Obstáculos Fixos

Para atestar a capacidade de repetição dos resultados da rede neural artificial proposta, foram executadas diversas simulações partindo de uma mesma posição e com o mesmo valor inicial para os pesos.

Para tanto, gerou-se, na primeira execução, um conjunto de pesos iniciais para a rede que foram armazenados em um arquivo, tal qual foi feito nas simulações em um ambiente sem obstáculos.

Realizaram-se vários conjuntos de 50 execuções para uma mesma posição inicial. Cada execução armazenava resultados em arquivos para posterior comparação.

Algumas telas foram capturadas para demonstrar a repetitividade da rede neural artificial proposta, como podem ser vistas na figura 46(a-d).

5.2.3 A Estabilidade da Rede AMNN em Ambientes Contendo Obstáculos Fixos

Todos os testes realizados com a rede AMNN proposta, com os parâmetros para os valores iniciais dos pesos e para a taxa de aprendizado escolhidos, obtiveram êxito em conduzir o veículo móvel até o alvo, independente da posição inicial.

Algumas vezes, a rede oscilava um pouco ao iniciar a execução, porém rapidamente se estabilizava e convergia para o objetivo.

Muitos testes foram necessários para atingir uma boa estabilidade da rede, pois as redes neurais artificiais são bastante suscetíveis aos valores utilizados em sua configuração. A boa escolha dos parâmetros garante a eficiência da rede.

Os resultados obtidos nos testes serviram para ratificar o sucesso das

configurações da rede AMNN proposta no controle da trajetória de um veículo móvel em ambientes com ou sem obstáculos. Passou-se então aos testes para validar a utilização da mesma rede para resolver o problema do controle da trajetória do veículo móvel, quando em ambientes com obstáculos que podem se deslocar pelo ambiente.

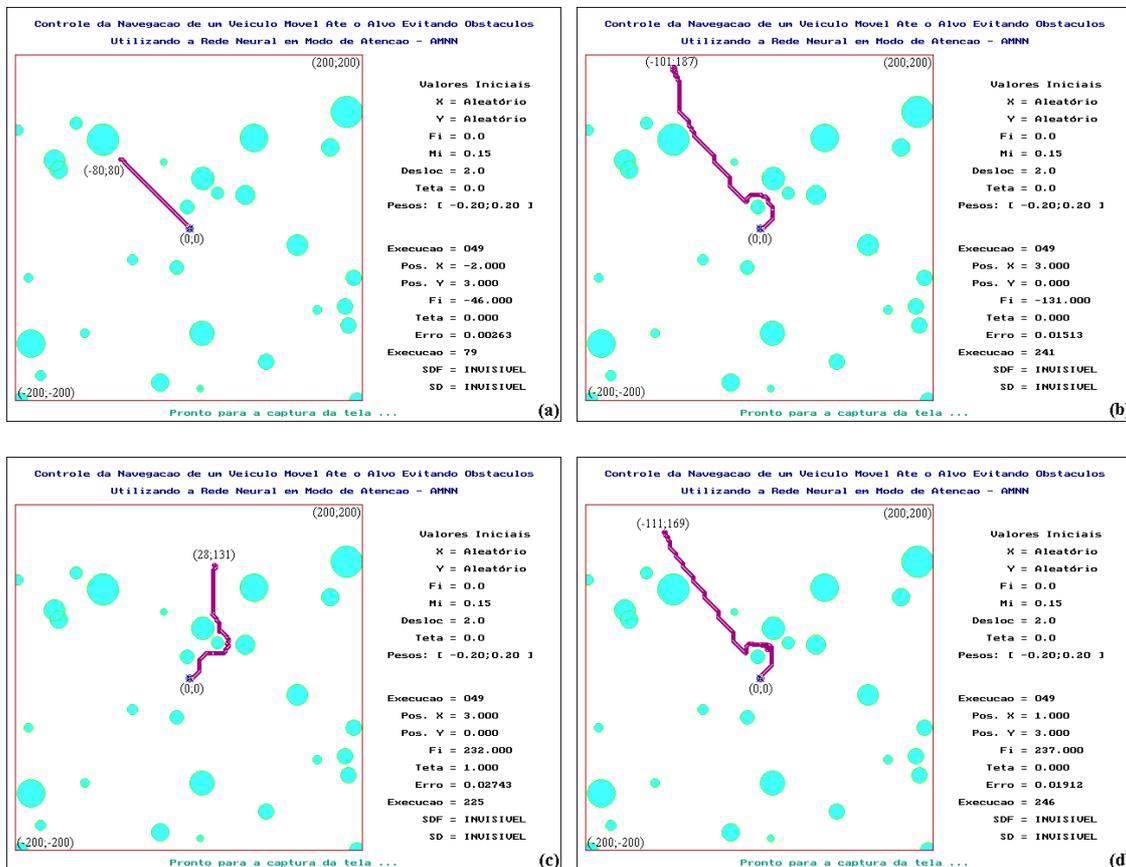


FIGURA 46 – A repetitividade dos resultados com a rede AMNN em um ambiente contendo obstáculos fixos.

5.3 Ambiente com Obstáculos Móveis

Os testes realizados para validar a rede AMNN proposta para ambientes com obstáculos móveis foram realizados dentro dos mesmos moldes dos anteriores, com exceção de que neste, os objetos também se movimentam aleatoriamente pelo ambiente. Aproveitaram-se os mesmos obstáculos gerados para as simulações efetuadas quando estes eram fixos.

Como verificado nas simulações anteriores, os pesos inicializados em intervalos próximos a $[-0,2;0,2]$ obtiveram melhores resultados. Sendo assim, para validar a utilização da rede neural proposta no tratamento de obstáculos móveis, restringiram-se os intervalos pesquisados. Foram escolhidas as faixas $[-0,2;0,2]$, $[-0,25;0,25]$, $[-0,3;0,3]$ e $[-0,4;0,4]$ para a inicialização dos pesos da rede.

Os coeficientes de aprendizado testados para objetos móveis foram valores iguais a 0,05, 0,1, 0,15 e 0,2, pois foram as taxas que obtiveram um melhor desempenho

nas outras simulações.

As simulações sempre partiram da posição cartesiana (150;150) devido a existência de muitos obstáculos nas proximidades. Os resultados obtidos estão apresentados na tabela 3.

TABELA 3 – Resultados das simulações para o ambiente com obstáculos móveis.

Situação	Taxa de Aprendizado (μ)			
	0,05	0,10	0,15	0,20
Pesos entre [-0,2;0,2]				
Menor número de passos	173,00	189,00	173,00	182,00
Média de passos	572,50	593,50	455,00	528,00
Maior número de passos	4741,00	1818,00	2021,00	4450,00
Nº de simulações que não atingiu o alvo	6,00	2,00	2,00	3,00
Percentual de acerto (encontrou o alvo)	88,00%	96,00%	96,00%	94,00%
Pesos entre [-0,25;0,25]	0,05	0,10	0,15	0,20
Menor número de passos	257,00	167,00	171,00	193,00
Média de passos	539,50	530,00	444,00	642,00
Maior número de passos	4592,00	1865,00	2837,00	4799,00
Nº de simulações que não atingiu o alvo	2,00	2,00	6,00	6,00
Percentual de acerto (encontrou o alvo)	96,00%	96,00%	88,00%	88,00%
Pesos entre [-0,3;0,3]	0,05	0,10	0,15	0,20
Menor número de passos	177,00	198,00	213,00	185,00
Média de passos	404,50	496,00	480,00	630,00
Maior número de passos	2876,00	3870,00	3577,00	2301,00
Nº de simulações que não atingiu o alvo	0,00	5,00	3,00	7,00
Percentual de acerto (encontrou o alvo)	100,00%	90,00%	94,00%	86,00%
Pesos entre [-0,4;0,4]	0,05	0,10	0,15	0,20
Menor número de passos	179,00	290,00	327,00	323,00
Média de passos	441,50	657,50	1291,50	2380,00
Maior número de passos	1152,00	2848,00	5000,00	5000,00
Nº de simulações que não atingiu o alvo	4,00	2,00	6,00	15,00
Percentual de acerto (encontrou o alvo)	92,00%	96,00%	88,00%	70,00%

A simulação era encerrada quando a rede conseguia levar o veículo até o objetivo ou quando o limite de 5000 passos era atingido.

A movimentação dos obstáculos pelo ambiente é feita a cada interação da rede e em grupos aleatórios de, no máximo, 15% da quantidade de obstáculos presentes no ambiente (30 no total).

Uma vez escolhido um obstáculo a ser movido, são gerados dois números randômicos (entre -1 e 1), que serão utilizados para atualizar a sua posição atual. Antes de movê-lo; porém, é realizado um teste para verificar se o objeto atual, ao ser movido, não vai colidir, ou até mesmo invadir um outro obstáculo. Caso isto se verifique, ele não poderá ser movimentado. Com dois obstáculos mais à esquerda no ambiente, verifica-se

exatamente este problema, mas isto ocorre na geração dos obstáculos e, estes, não são movimentados, pois recaem neste problema.

Mesmo com os cuidados tomados neste sentido, algumas vezes o veículo móvel acabava chegando muito próximo dos obstáculos e encontrava um mínimo local, de onde não conseguia se recuperar. Este exemplo está ilustrado pela figura 47.

Da mesma forma que o veículo móvel deixa seu rastro durante a navegação, os obstáculos que se deslocam pelo ambiente também o fazem. Servem para acompanhar a sua movimentação com o passar do tempo.

Em algumas execuções, um obstáculo se sobrepunha ao alvo (como acontece na figura 46), fazendo com que o veículo fique “rondando-o”, sem poder atingi-lo.

Uma outra alteração que se fez necessária foi o aumento da região de sensibilidade do sensor SDF, dado ao fato de os obstáculos também se moverem pelo ambiente e pela necessidade do veículo móvel precisar detectar se um obstáculo move-se em sua direção. Caso contrário, a rede não terá tempo de se recuperar. O novo valor para o alcance do sensor SDF é de 5um.

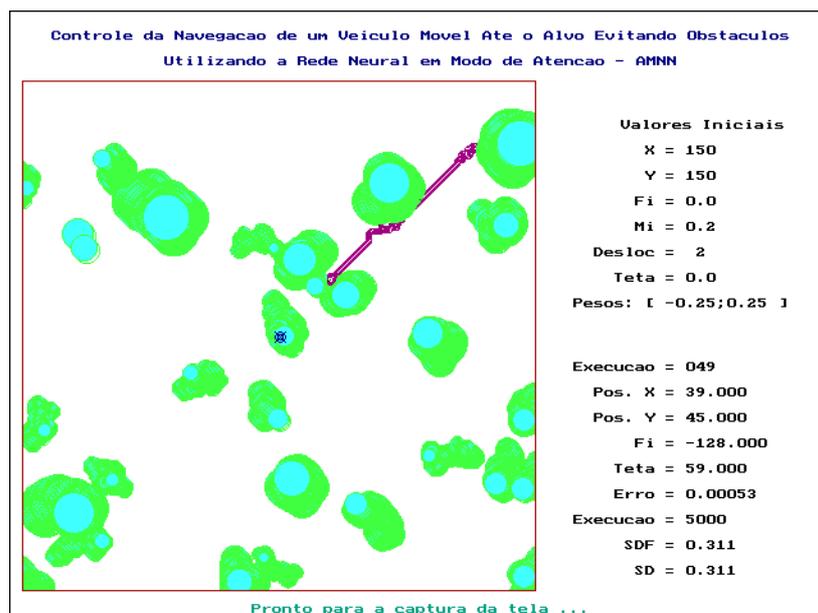


FIGURA 47 – Exemplo da ocorrência de um problema.

A rede conseguiu atingir o alvo em 100% das execuções somente na configuração que utilizou um coeficiente de aprendizado de 0,05 e com os pesos inicializados no intervalo [-0,3;0,3]. A imagem da última execução, da série de 50, para esta configuração é apresentada na figura 48.

Às vezes o rastro do veículo móvel fica encoberto pelas marcas deixadas pela movimentação dos obstáculos, sugerindo que o veículo tenha atingido ou tangenciado algum deles, como o que acontece na figura 48.

A mesma figura 48 mostra que o veículo móvel ficou preso entre alguns obstáculos e “aguardou” até que estes ficassem um pouco mais distantes. Ao detectar o surgimento de um espaço por onde poderia passar, procurou alcançar o seu objetivo.

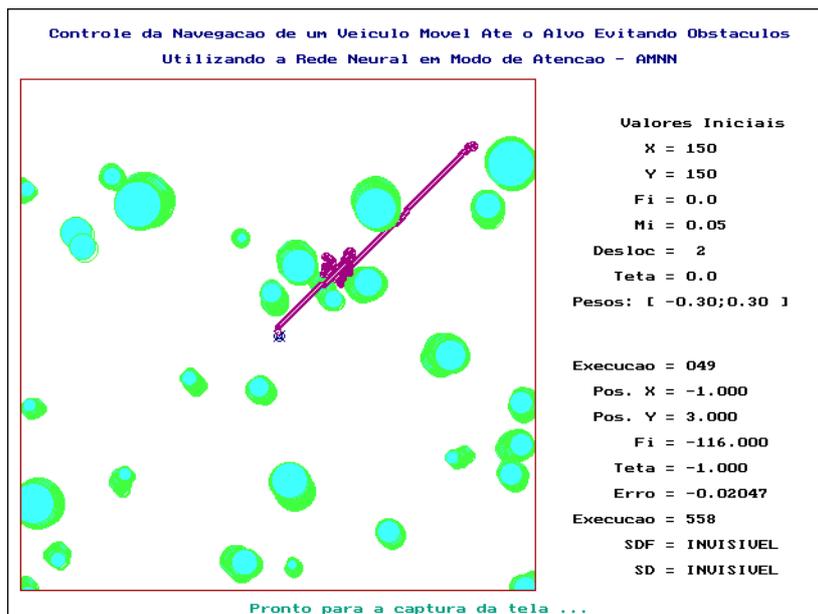


FIGURA 48 – Exemplo de uma execução na qual o veículo móvel consegue atingir o alvo.

As figuras 48 e 49 ilustram outros exemplos de trajetos criados pela rede para chegar até a posição cartesiana (0;0), o alvo do veículo móvel.

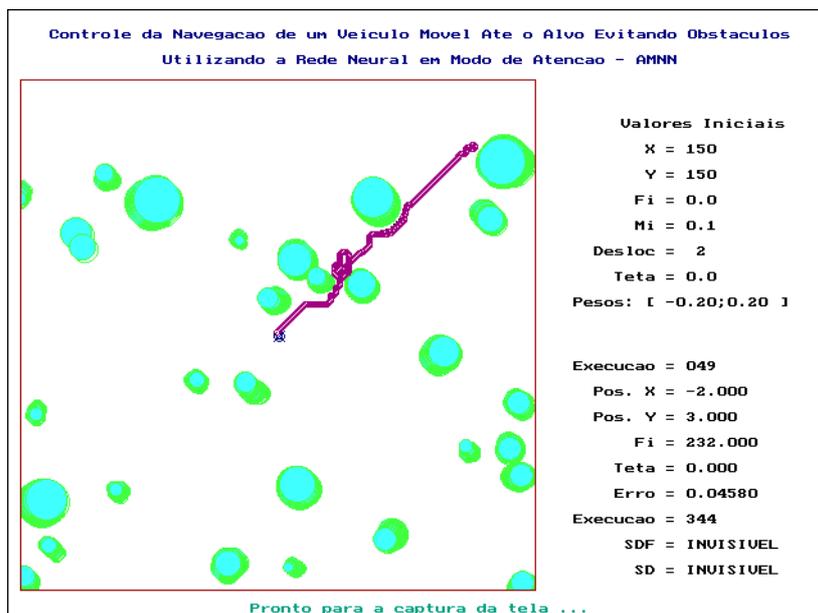


FIGURA 49 – A 50ª simulação para os pesos iniciais no intervalo $[-0,2;0,2]$.

Da mesma forma que nas simulações efetuadas com obstáculos fixos, algumas vezes a rede não converge, por força dos valores iniciais dos pesos, como acontece na figura 51.

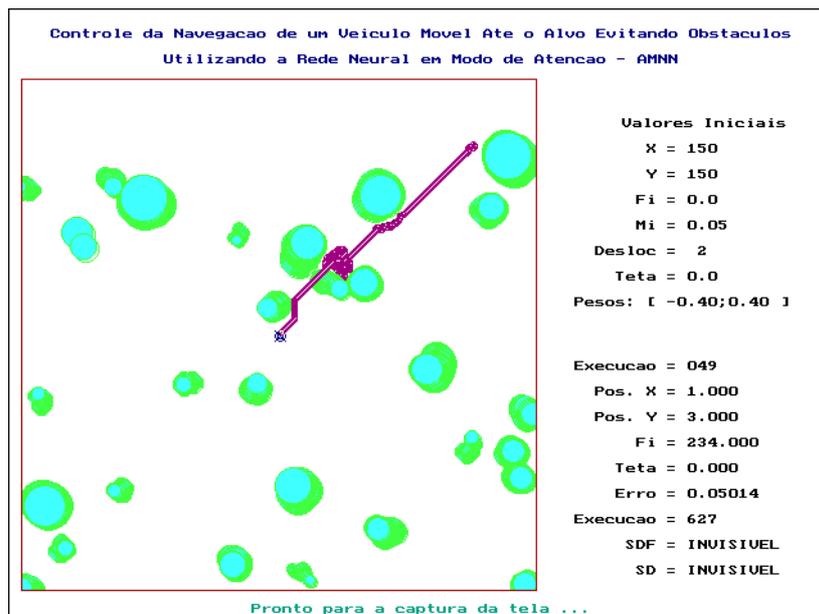


FIGURA 50 – A 50ª execução para pesos iniciais no intervalo $[-0,4;0,4]$.

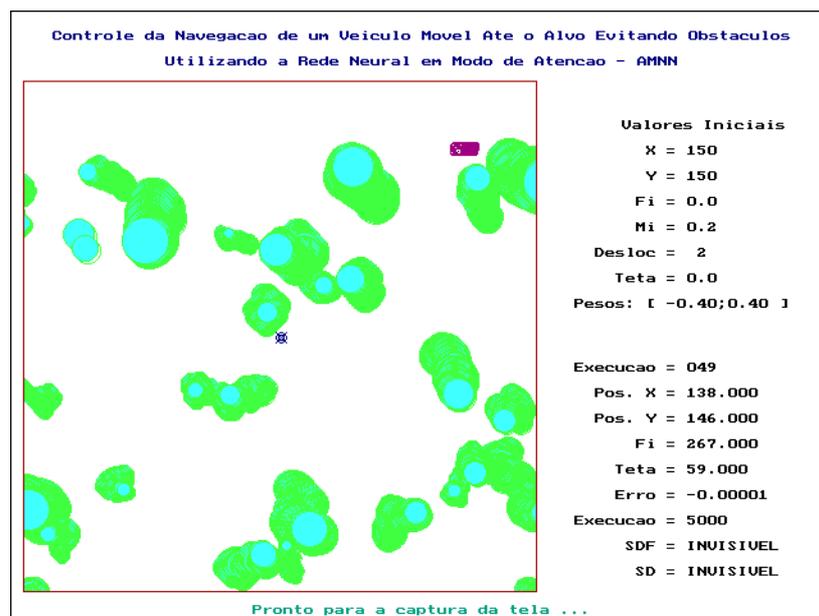


FIGURA 51 – Exemplo da não convergência da rede com obstáculos móveis.

Mesmo com algumas restrições, a rede proposta mostra que é capaz de guiar o veículo até o seu objetivo. No entanto, é necessário fazer um estudo aprofundado para poder escolher os melhores parâmetros para cada modelo de aplicação.

A grande vantagem advém da pouca complexidade computacional associada à resolução do problema. Com alguns refinamentos, os resultados obtidos poderão ser ainda melhores.

A repetitividade e a convergência da rede já foram comprovadas com as simulações anteriores. É um pouco difícil repetir toda a infinidade de números aleatórios que são gerados em uma simulação com 5000 repetições. Mesmo porque são randômicos.

Haverá a necessidade de uma reestruturação do modelo da rede para que possa ser utilizado no Nomad, tendo em vista que ele pode seguir em qualquer direção, ou seja, suas rodas podem girar 180° . Outra mudança deve-se ao fato do número de sensores passíveis de utilização, que fornecem uma cobertura de 360° ao redor do veículo.

Ao ser aplicada ao Nomad, esta proposta vai poder ser expandida, dada a facilidade de trabalho, e também com a quantidade de sensores disponíveis. Isto poderá facilitar em “sugerir” rotas alternativas para a rede, pois a região coberta por sensores é bem maior.

6 Conclusão e Perspectivas Futuras

Este trabalho teve por objetivo o estudo da aplicabilidade da rede neural AMNN no controle da navegação de um robô móvel em ambientes com obstáculos. Para tanto, foi proposto, implementado e avaliado um sistema de controle da trajetória de veículos móveis utilizando uma única rede neural. A proposta foi plenamente atingida apesar da rede não ter conseguido convergir em algumas situações, mas isto decorre do fato dos pesos iniciais serem inicializados aleatoriamente. Raramente valores aleatórios muito ruins, que causam problemas, são gerados.

Mais uma vez esta rede ratifica sua capacidade de tratamento de problemas relativos a sistemas abertos. Serve, inclusive, para mostrar que as redes neurais têm uma grande possibilidade de aproveitamento na área da robótica, principalmente a rede AMNN.

A rede AMNN também demonstrou o seu grau de adaptabilidade, pois, uma vez que se baseia no gradiente descendente, a possibilidade de cair em mínimos locais é bastante alta. Conseguiu, sem outros artifícios, “aprender” a contornar os obstáculos sem perder, entretanto, sua meta principal: alcançar o alvo. Tudo isto com um gasto computacional relativamente baixo, se comparada com outros métodos.

A próxima meta é testar e aplicar esta proposta no robô móvel Nomad. Com um robô e ambientes reais mutáveis, podem ser validadas as potencialidades, já conhecidas através de trabalhos de simulação, da rede AMNN aplicada à robótica móvel.

Anexo 1 O Robô Nomad 200

O Nomad 200¹ é um sistema robótico móvel integrado com 6 módulos sensoriais, a saber: táteis, infravermelho, ultra-som, visão básica, visão estruturada da luz e sistema de localização. Possui computadores dedicados ao controle dos sensores, dos motores (atuadores) e também para comunicação com o computador central, através de uma ligação através de microondas. O sistema móvel mantém o rastreamento da sua posição e orientação (integradas com o passar do tempo). Complementa o pacote um conjunto de softwares para o sistema central com uma interface gráfica e o simulador do robô.

É uma plataforma bastante interessante para ensinar e pesquisar a robótica e a inteligência artificial.

A1.1 O Sistema Mecânico

A base móvel do Nomad 200 possui três motores, três rodas sincronicamente guiadas em um sistema não-holonômico (ver figura A1-1) com um raio de giro nulo, ou seja, pode girar em torno do seu centro. As três rodas são direcionadas simultaneamente por um único motor e giram sob o controle de um segundo motor. Um terceiro motor controla a posição angular da torre. O robô só pode navegar para frente e para trás na direção de alinhamento indicado pelas três rodas (o que caracteriza os sistemas não-holonômicos, como acontece com um carro). Sua velocidade máxima é de 60,96cm (24 polegadas) por segundo e uma velocidade de rotação máxima de 60 graus por segundo.

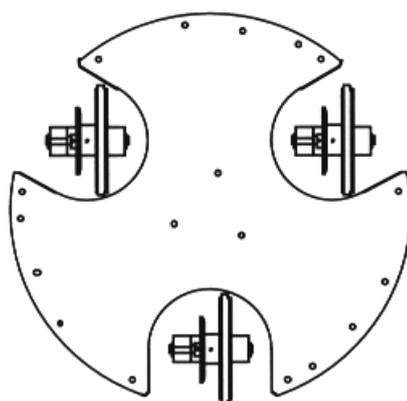


FIGURA A1-1 – As três rodas motrizes e direcionais do Nomad.

¹ Todas as informações aqui apresentadas foram obtidas através dos manuais que acompanham o robô Nomad 200, em especial o *Nomad 200 Hardware Manual*, fornecidos pela Nomadic Tehnologies, Inc., Mountain View, California, 1997.

A1.2 Sistema de Sensoriamento

O Nomad 200 possui seis sistemas de sensoriamento, são eles:

- sistema táctil;
- sistema de infravermelho;
- sistema de ultra-som;
- sistema básico de visão;
- sistema de visão estruturado por raios laser; e
- sistema de localização.

A1.2.1 O Sistema Táctil (Sensus 100)

É um sistema táctil com 20 canais que consiste em 20 sensores de pressão (interruptores); utilizados para detectar o contato com um objeto. Estão dispostos em dois anéis (ver figura A1-2), cada um com 10 interruptores, intercalados de forma a cobrir os 360 graus do perímetro, com uma resolução de 18 graus (ver figura A1-3). Cada interruptor tem uma sensibilidade aproximada de 226,8 gramas (8 ounce). É ideal para executar tarefas como detectar e empurrar objetos. O sistema rudimentar de absorção de impactos foi projetado para proteger o Nomad contra eventuais colisões acidentais.

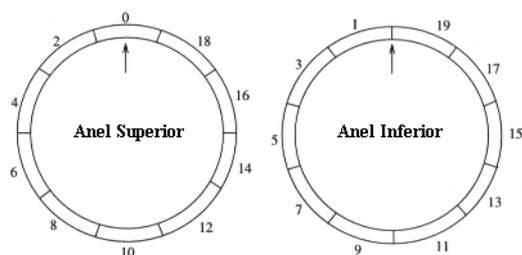


FIGURA A1-2 – O mapeamento dos sensores de toque.

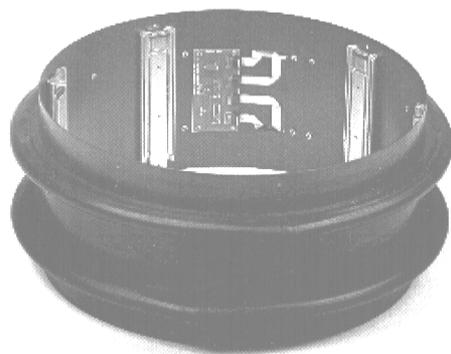


FIGURA A1-3 – Os sensores de toque.

A1.2.2 O Sistema de Sonar (Sensus 200)

É um sistema de sonar composto por 16 canais, fornecendo informações de proximidade que variam de 43,18cm (17 polegadas) até 6,477m (255 polegadas) com uma precisão de 1% sobre a distância total. Está baseado no tempo de deslocamento do sinal acústico desde o seu envio até o retorno.

Cada transdutor possui um raio de ação de 25 graus. A circunferência do robô pode ser coberta pelos 16 sensores e estão dispostos conforme a figura A1-4. Ao iniciar um ciclo de leitura o transdutor emite uma série de 16 pulsos em uma frequência de 49,4KHz. Para eliminar a confusão do som do transdutor com o sinal de retorno, o circuito detector do retorno é desabilitado ou apagado, o que permite visualizar uma distância mínima de 15,24cm (6 polegadas).

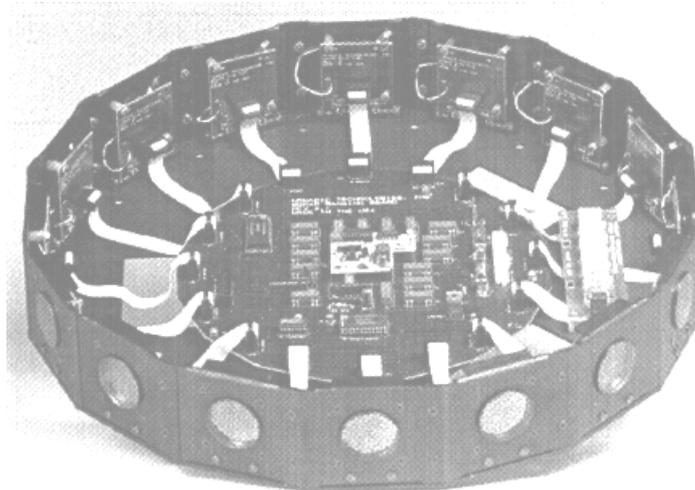


FIGURA A1-4 – O sistema de ultra-som do Nomad.

A1.2.3 Sensores de Infravermelho (Sensus 300)

É um sistema rastreador de intensidade refletiva do infravermelho composto por 16 canais, conseguindo um alcance de até 60,96cm (14 polegadas), dependendo das condições do ambiente. Cada um dos 16 sensores é composto por dois diodos emissores de luz (LED) e um fotodiodo detector. Pode ser configurado para efetuar uma cobertura total do perímetro do robô (360 graus) utilizando todos os 16 sensores, como mostra a figura A1-5.

A distância entre o robô e o objeto é determinado pela intensidade da luz emitida que retorna ao refletir no objeto. A intensidade da luz é medida digitalmente por um VCO (Osciloscópico Controlado pela Voltagem), um dispositivo que emite uma onda quadrada com uma frequência proporcional à voltagem de entrada. Quanto mais alta for a voltagem na entrada, mais rápido oscilará (figura A1-6).

No início de um ciclo de leitura, o VCO oscila em uma frequência proporcional à leitura da voltagem na entrada, que, por sua vez, é proporcional à quantidade de luz no

ambiente que incide sobre o detector. Simultaneamente, um contador de alta velocidade inicia a contagem do tempo decorrido desde o início do ciclo de leitura. Multiplexadores são utilizados, tanto nos emissores como nos receptores, para selecionar individualmente os sensores.

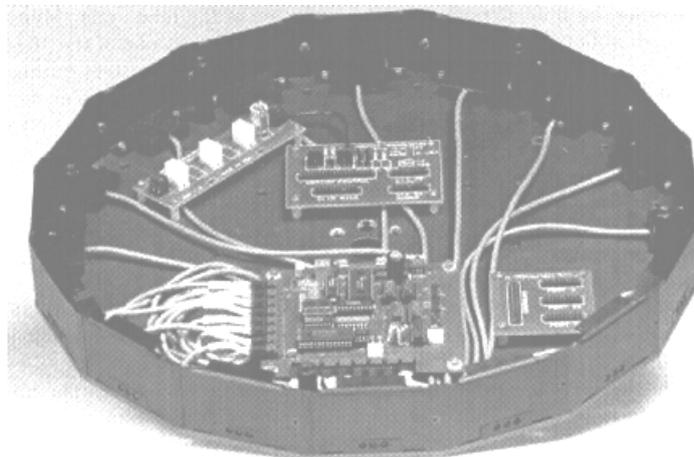


FIGURA A1-5 – O sistema de infravermelho do Nomad.

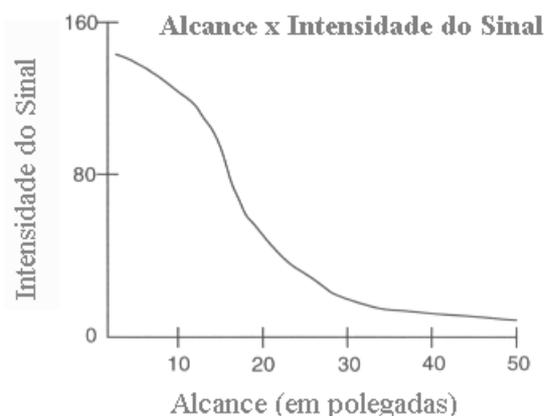


FIGURA A1-6 – O gráfico da leitura dos sensores de infravermelho.

Uma linha de calibragem é utilizada para controlar o estado dos emissores durante a leitura (ligado ou desligado). Isto é muito útil para determinar o nível de ruído no ambiente. A diferença da leitura quando os emissores estão ligados e quando estão desligados informa um valor dependente unicamente da luz proveniente dos emissores.

O número de pulsos emitidos pelo VCO em um ciclo de leitura determina a resolução. Quanto maior forem os pulsos emitidos, maior será a resolução na conversão. Uma maior resolução indica valores mais precisos, porém requer um tempo de conversão igualmente maior. Em alta resolução, o tempo de conversão leva aproximadamente 0,0025 segundo.

O nível de reflexão da superfície e de luz no ambiente afetará a leitura, devendo ser calibrado no ambiente de uso.

A1.2.4 Sistema Básico de Visão (Sensus 400)

O sistema básico de visão consiste em uma câmera, uma placa coletora de imagens e uma biblioteca de funções para interfaceamento. A câmera padrão é uma CCD com uma resolução de 490x512 pixels em preto e branco.

A placa capturadora de imagens pode trabalhar com uma imagem em alta resolução (486x512 pixels) ou com até 4 imagens em baixa resolução (243x256 pixels). A escala de tons de cinza é determinada em todos os casos por um byte.

A biblioteca possui funções que servem para controlar a placa, capturar imagens e alguns processamentos básicos de imagens, como aqueles de convolução, detecção de bordas, operações binárias, extração de linhas, etc.

A1.2.5 Sistema de Visão Estruturado (Sensus 500)

É um sistema que utiliza a técnica da triangularização para indicar a distância. Baseia-se em um diodo laser como fonte de luz e uma câmera matricial CCD para a geração da imagem (ver figura A1-7). Seu alcance vai desde 45,72cm (18 polegadas) até 3,04m (120 polegadas). A resolução aumenta a não-linearidade à medida que se aproxima do objeto.

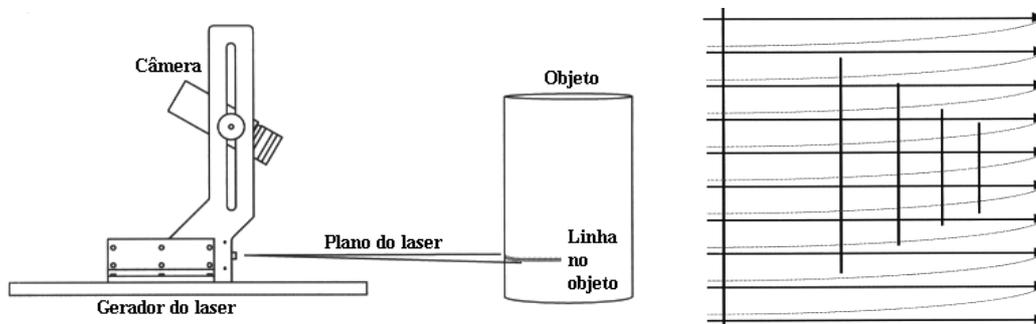


FIGURA A1-7 – O sistema de visão por laser e a formação da imagem do Nomad.

São 480 linhas por frame e 30 frames por segundo, totalizando 57,6 bytes de dados por segundo. Os dados são armazenados diretamente em uma memória RAM do tipo porta dupla (Dual Port RAM – DPR)

A1.2.6 Sistema de Orientação (Bússola – Sensus 600)

Utiliza um dos mais modernos sistemas de orientação por microprocessador controlado por porta de fluxo. Através do uso da capacidade única de compensação da bússola, podem ser obtidas precisões de até 1 grau. É colocado na parte superior do robô.

A1.3 O Sistema de Processamento

O sistema utiliza um sistema multiprocessado com memória compartilhada; toda a comunicação entre os processadores é feita através do barramento ISA (padrão dos PC's). O controle central é desempenhado por um processador compatível com a família X86 da Intel (veja figura A1-8).

Possui ainda 2 portas seriais e 1 paralela, uma interface padrão "IDE", uma unidade de disco flexível de 3,5 polegadas, um disco rígido e uma porta para joystick.

Toda esta plataforma executa o sistema operacional Linux.

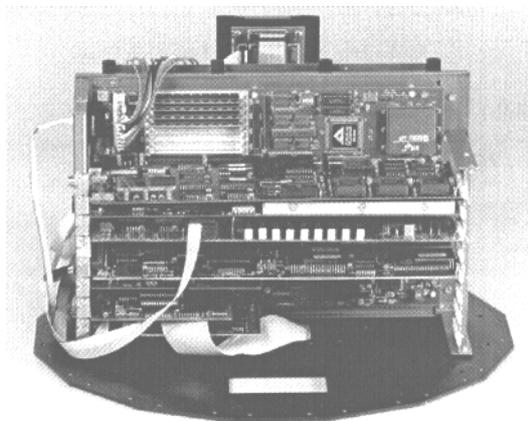


FIGURA A1-8 – A arquitetura do Nomad.

A1.4 O Sistema de Energia

A alimentação do sistema é feita através de dois conjuntos de baterias recarregáveis que juntos fornecem 840 watt/hora, sendo distribuídos em dois sistemas:

- Energia para o sistema de computação:
 - trabalha em 12V e é composto por 3 baterias;
 - 552 Watts por hora;
 - alimenta os sensores e os computadores;
 - controlada pelo sistema de gerenciamento de energia.
- Energia para o sistema motor:
 - Trabalha em 24V e é composto por 2 baterias;
 - 288 Watts por hora;
 - fornece energia somente para os motores.

Os dois sistemas de energia são completamente isolados entre si.

Bibliografia

- [ADA 97] ADAMI, A. G. **Sistema de Reconhecimento de Locutor Utilizando Redes Neurais Artificiais**. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de Mestrado.
- [AGA 97] AGAH, A.; BEKEY, G. Philogenetic and Ontogenetic Learning in a Colony of Interacting Robots. **Autonomous Robots**, USA, v.1, n.4, 1997.
- [ARK 98] ARKIN, Ronald. **Behavior-Based Robotics**, USA, MIT Press, USA, 1998.
- [BAL 91] BALOCH, A.; WAXMAN, A. Visual Learning, Adaptative Expectations, and Behavioral Conditioning of the Mobile Robot MAVIN. **Neural Networks**, New York, v.4, p. 271-302, 1991.
- [BAR 96] BARONE, D.A.C. et al. Algoritmo Genético no Controle da Trajetória de um Veículo. In: SIMPÓSIO BRASILEIRO DE REDES NEURAI 3., SBRN 3., 1996, Florianópolis, **Anais ...**
- [BEE 94] BEER, R. A Dynamical Systems Perspective on Agent-Environment Interaction. **Artificial Intelligence Journal**, New York. June 1994.
- [BEL 91] BELEW, R. et al. Evolving Networks: using the genetic algothm with connectionist learning. In: WORKSHOP ON ARTIFICIAL LIFE. USA, **Proceedings...**Addison Wesley, 1991, p.9-14.
- [BEL 94] BELEW, R. Evolution, Learning, and Culture: Computational Metaphors for Adaptative Algorithms. **Complex Systems**, New York, USA, n.4, p. 11-49, 1994.
- [BOD 95] BODEM, M. AI's Half Century. **AI Magazine**, USA, v.16, n.2 p.96-99, USA, 1995.
- [BOR 96] BORGES, N. C. K. Neocognitron e Robótica. In: SEMANA ACADÊMICA DO CPGCC DA UFRGS, 1996, p. 247-250 **Anais...** Porto Alegre: Editora da Universidade.
- [BOR 97] BORGES, N. C. K. **Reconhecimento de Padrões Visuais pela Rede Neocognitron**. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de Mestrado.
- [BOS 93a] BOSCO, J. et al. Ficção, Realidade e Expectativa de Robôs Inteligentes baseados em Comportamento. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE 1., 1993. **Anais...** Rio Claro, Brasil. v.1, p.145-154.
- [BOT 96] BOTELHO, S.S.C. **Desenvolvimento de Sistemas Inteligentes para Controle de Robôs Móveis**. Porto Alegre: CPGCC da UFRGS, 1996. Dissertação de Mestrado.
- [BRA 95] BRADY, M. Artificial Intelligence and Robotics. **Artificial intelligence and Robotics**, USA, v.26, pp.79-121, 1985.
- [BRO 86] BROOKS, R; CONNELL, J. Asynchronous Distributed Control System for a Mobile Robot. In: **SPIE**. Massachusetts: MIT Press, 1986.

- [BRO 87] BROOKS, R. **Planning is Just a Way of Avoiding Figuring Out What to do Next**. Relatório de Pesquisa n.303. MIT AI Laboratory, USA, 1987.
- [BRO 90] BROOKS, R. Elephants Don't Play Chess.. In: **Designing Autonomous Agents**. Massachusetts: MIT Press, 1990. p.3-15.
- [BRO 91] BROOKS, R. **New Approaches to Robotics**. *Science*, v.253, pp.1227-1232. USA, 1991.
- [BRO 91a] BROOKS, R. **Intelligence Without Reason**, A.I. Memo no. 1293, MIT AI Laboratory, USA, 1991.
- [BRO 91b] BROOKS, R.. Artificial Life and Real Robot. In: **Conference on Artificial Life**. Massachusetts: MIT Press, 1991. p.3-10.
- [BRO 93] BROOKS, R. et al. **Building Brains for Bodies**. In: AI-memo 1439, Artificial Intelligence Laboratory, MIT, Massachusetts, USA, 1993.
- [CAR 86] CARPENTER, A.; GROSSBERG, S. Associative Learning, Adaptive Pattern Recognition and Cooperative-Competitive Decision Making by Neural Networks. In: H. Szu, editor, Hybrid and Optical Computing. **SPIE**, 1986.
- [CAU 91] CAUDILL, M. Driving Solo. **AI Expert**. New York, USA, v.5, n.9.
- [CHO 95] CHOSET, H; BURDICK, J. Sensor based planning part II: Incremental construction of the Generalized Voronoi Graph. In: IEEE International Conference on Robotics and Automation, 1995, Nagoya, Japan. **Proceedings...** 1995.
- [COE 96] COELHO, J.S. A aplicação da Rede Neural em Modo de Atenção na Robótica Móvel, **Trabalho Individual**, CPGCC/UFRGS, 1996.
- [ENG 96] ENGEL, P.M. Attentional Mode Neural Network: A New Approach for Real-Time Self Learning. In: IEEE International Symposium on Circuits and Systems. **Proceedings ...** Atlanta, USA, 1996.
- [EVE 95] EVERETT, B. **Sensors for Mobile Robots**. A.K. Peters, USA, 1995.
- [FAB 95] FABRO, J.; GOMIDE, F. Self Organizing NeuroFuzzy Control of Autonomous Vehicles. In: II Simpósio Brasileiro de redes Neurais, 1995, São Carlos, SP. **Anais...** São Carlos: SBC, 1995. 238p, p. 63-79.
- [FER 94] FERRELL, C. Failure Recognition and Fault Tolerance of an Autonomous Robot. **Adaptative Behavior**, Massachusetts,USA, v.2, n.4, p.375-398, 1994.
- [FER 96] FERREIRA, A. P. **Uma Proposta de Controle Adaptativo para o Posicionamento de um Braço Mecânico**. Dissertação de Mestrado. CPGCC/UFRGS, 1996.
- [FIK 71] FIKES, R., NILSSON, N. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. **Artificial Intelligence**, v.2, pp189-208, USA, 1971.
- [FIK 72] FIKES, R. et al. Learning and executing generalized robot plans. In Artificial Intelligence 3 - AI3, Cambridge, USA. **Proceedings...** Cambridge: MIT Press-Bradford Books, 1972. p.251-288.
- [FLO 94] FLOREANO, D.; MONDADA, F. Autonomous and Self-Sufficient: Emergent Homing Behaviours in a Mobile Robot. In: **LAMI Technical Report R94.14I**, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1994.

- [FRE 91] FREEMAN, J.A., SKAPURA, D.M. **Neural Networks – Algorithms, Application and Programming Techniques**. USA: Addison-Wesley, 1991.
- [FUK 94] FUKUDA, T. Fuzzy-Neuro-GA Based Intelligent Robotics. In: Computational Intelligence Imitating Life. New York: IEEE Press, 1994. p. 352-364.
- [GAT 92] GAT, E. Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots. In: AAAI92. **Proceedings...** 1992. p. 11.1.1-11.1.5.
- [GOL 89] GOLDBERG, D. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison-Wesley, Reading, USA, 1989.
- [GOO 94] GOODRIDGE, S. e LUO, R. Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot: MARGE. IEEE International Conference on Robotics and Automation, **Proceedings ...** 1994.
- [HAL 94] HALLAM, B. et al. An Ethological Model for Implementation in Mobile Robots. **Adaptive Behavior**, Massachusetts, USA, v.3, n.1, p.51-79, 1994.
- [HAY 94] HAYKIN, S. **Neural Networks – A Comprehensive Foundation**. Macmillan College Publishing Company, Inc., New Jersey, 1994.
- [HEB 49] HEBB, D. **The Organization of Behavior**. Willey, New York, 1949.
- [HIL 88] HILLIER, M. **Automata and Mechanical Toys**. Bloomsbury, London, 1988.
- [HOR 93] HORSWILL, I. Analysis of Adaptation and Environment. **Artificial Intelligence Journal - Special Issue on Computational Theories of Interaction**, New York. Dec. 1993.
- [JAB 85] JABLONSKY, J.; POSEY, J. Handbook of Industrial Robotics. J. Willey, New York, 1985.
- [JAN 95] JANSEN, A. et al. Nested Networks for Robot Control. In: **Applications of Neural Networks**. Amsterdam: Kluwer Academic Publishers, 1995. p.221-239.
- [KAE 93] KAELBLING, L.P. **Learning in Embedded Systems**. EUA, MIT Press, 1993.
- [KEY 90] KEYMEULEN, D.; DECUYPER, J. **A Flexible Path Generator for a Mobile Robot: a Mathematical Approach**. In: AI-memo 90-5, VUB AI-Lab, Vrije Universiteit Brussel, Belgium, 1990.
- [KHA 85] KHATLIB, O. Real Time Obstacle Avoidance for Manipulators and Mobile Robots. IEEE International Conference on Robotics and Automation, **Proceedings ...** St. Louis 1985, MO, pp. 500-05.
- [KOH 82] KOHONEN, T. Analysis of a Simple Self Organizing Process. **Biological Cybernetics**. V.44, pp.135-140. USA, 1982., USA, 1982.
- [KOH 94] KOH, K. et al. A Neural Network-Based Navigation System for Mobile Robots. In: THE 1994 IEEE International Conference on neural Networks. Orlando, USA. **Proceedings...** Piscataway: IEEE press, 1994. v.4, p.2709-2713.
- [KOR 98] KORTENKAMP, D.; BONASSO, R. Peter; MURPHY, Robin. **Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems**. USA, MIT Press/AAAI Press, 1998.

- [KOS 92] KOSKO, B. **Neural Networks and Fuzzy Systems**. Prentice Hall, USA, 1992.
- [KOZ 94] KOZA, J. Evolution of Subsumption Using Genetic Programming. In: IWK'94. Ilmenau, Germany. **Proceedings...**1994. v.1, p.102-108.
- [KRO 84] Krogh, B. A Generalized Potential Field Approach to Obstacle Avoidance Control, SME-RI **Technical Paper** MS84-484, Society of Manufacturing Engineers, Dearborn, Michigan, 1984.
- [KRO 94] KRÖSE, B.; EECEN, Marc. A self-organizing representation of sensor space for mobile robot navigation. In: Conference on Intelligent Robots and Systems. München, Germany. **Proceedings...** 1994. p.9-14.
- [KRO 95] KRÖSE, B.; DAM, Joris. **Learning to Avoid Collisions: a Reinforcement Learning Paradigm for Mobile Robot Navigation**. Technical Report 93. Faculty of Mathematics and Computer Science, University of Amsterdam, Amsterdam, Netherlands, 1995.
- [KUB 94] KUBE, R. et al. Collective Robotics: From Social Insects to Robotics. **Adaptive Behavior**, Massachusetts, USA, v.2, n.2, p.189-218, 1994.
- [LAT 93] LATOMBE, J. **Robot Motion Planning**. New York: Kluwer Academic Publishers, 1993.
- [LIB 97] LIBERMAN, F. **Classificação de Imagens Digitais por Textura Usando Redes Neurais**. Dissertação de Mestrado, CPGCC/UFRGS, 1997.
- [LUO 94] LUO, R. Landmark Recognition using Projection Learning for Mobile Robot Navigation. In: THE 1994 IEEE International Conference on Neural Networks. Orlando, USA. **Proceedings...** Piscataway: IEEE press, 1994. v.4, p.2703-2708.
- [MAE 94] MAES, P. Learning Behaviour Networks from Experience. In: **IWK'94**. Ilmenau, Germany. **Proceedings...**1994. v.1, p.102-108.
- [MAR 96] MARQUES, I.S. **Algoritmos Genéticos em Previsão de Séries Temporais**. Trabalho Individual, CPGCC/UFRGS, 1996.
- [MAI 93] MAIO, D.; RIZZI, S. Map Learning and Clustering in Autonomous Systems. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, New York, v.15, n.12, p. 76-80. Dec. 1993.
- [MAT 90] MATARIC, M. Learning a Distributed Map Representation Based on Navigation Behaviors. In: USA-JAPAN Symposium on Flexible Automatica, 1990, Japan. **Proceedings...** 1990.
- [MAT 94] MATARIC, M. Interaction and Intelligent Behavior. In: **Thesis of Doctor of Philosophy** at the MIT. Massachusetts, USA: MIT, 1994. 160p.
- [MCC 43] MCCULLOCH, W.; PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. **Bulletin of mathematical biophysics**, v.5, pp.115-153, USA, 1943.
- [MCC 55] MCCARTHY, J.; MINSKY, M.; ROCHESTER, N.; SHANNON, C. A Proposal for the **Dartmouth Summer Research Project on Artificial Intelligence**. USA, 1955.
- [MCF 93] MCFARLAND, D. BOSSER, U. **Intelligent Behavior in Animal and Robots**. MIT Press, Cambridge, USA, 1993.

- [MCK 91] MCKERROW, P. **Introduction to Robotics**. New York: Addison-Wesley Publishing Company, 1991.
- [MEL 98] MEL, B. W. Vision-Based Robot Motion Planning. **Neural Networks for Control**, pg.229-253. AAIL Press/MIT Press 1998.
- [MIL 60] MILLER, G., GALANTER, E.; PRIBRAM, K. **Plans and the Structures of Behavior**. Holt, Rinehart and Winston, New York, 1960.
- [MIL 93] MILLER, T. Real-Time Control of a Biped Walking Robot. In: World Congress on neural Networks. Oregon , USA. **Proceedings...** New Jersey: Lawrence Erlbaun Associates Inc. Publisher, 1993. v.3, p.153-156.
- [MIN 69] MINSKY, M. and PAPERT, S. **Perceptrons**. MIT Press, USA, 1969.
- [MIN 86] MINSKY, M. **The Society of the Mind**. Simon and Schuster, New York, 1986.
- [MON 94] MONDADA, F; FRANZI, E. **Biologically Inspired Mobile Robot Control**. Technical Report PNR 23. Swiss National Research Foundation – Lausanne, Swiss, 1994.
- [MOR 77] MORAVEC, H. Towards Automatic Visual Obstacle Avoidance. Fifth International Joint Conference on Artificial Intelligence. **Proceedings ...** pp.584. Cambridge, USA, 1977.
- [MOR 88] MORAVEC, H. **Mind Children: The Futuer of Robot and Human Intelligence**, Harvard University Press, Cambridge, USA, 1988.
- [NAG 90] NAGATA, S. et al. Mobile Robot Control by a Structured Hierarchical Neural Network. **IEEE Control and Systems Magazine**, New York, 1991, p. 69-76, Apr. 1990.
- [NEH 94] NEHMZOW, U.; SMITHERS, Tim. Using Motors Actions for Location Recognition. In: IWK'94. Ilmenau,Germany. **Proceedings...**1994. v.1, p.102-108.
- [NGU 90] NGUYEN, D., WIDROW, B. **The Truck Backer-Upper: Na Example of Self-Learning in Neural Networks**. *N: Neural Networks for Control*, editado por: W. Thomas Miller III, richard S. Sutton e Paul J. Werbos, MIT Press, EUA, 1990.
- [NOL 94] NOLFI, S. et al. How to evolve autonomous robots: different approaches in evolutionary robotics. In: Fourth Workshop on Artificial Life, 1994, Boston, USA. **Proceedings...** 1994. p.658-687.
- [NOR 89] NOREILS, F.; CHATILA, R. Control of Mobile Robot Actions. IEEE International Conference on Robotics and Automation. **Proceedings...** pp.701-707. EUA 1989.
- [NOR 95] NOREILS, F.; CHATILA, R. Plan Execution Monitoring and Control Architecture for Mobile Robots. **IEEE Transactions on Robotics and Automation**, v.11, n.2, pp.255-266. EUA 1995.
- [PIM 92] PIMENTEL, J. et al. On-line Performance Enhancement of a Behavioral Neural Network Controller. In: Workshop on Artificial life, 10., 1991, USA. **Proceedings...** USA: Addison Wesley Publishing Co.,1991, p. 694-702.
- [RAB 95] RABUSKE, R. **Inteligência Artificial**. Florianópolis: Editora DAUFSC, 1995. 240p.
- [RAM 94] RAMOS, A. **Aplicação de Algoritmos Genéticos em Problemas de Otimização**. Trabalho Individual – TI403. CPGCC/UFRGS, 1994.

- [RAM 94a] RAM, A. Using Genetic Algorithms to Learn Reactive Control Parameters for Autonomous Robotic Navigation. **Adaptive Behavior**, Massachusetts, USA, v.2, n.3, p.277-305, 1994.
- [RIM 92] RIMON, E.; KODISCHEK, D. Exact Robot Navigation Using Artificial Potential Functions. **IEEE Transactions on Robotics and Automation**, New York, v.8, n.5, p. 501-518. Oct. 1992.
- [ROI 97] ROISEMBERG, M., BARRETO, J.M., AZEVEDO, F.M. **Modeling Behaviors With Artificial Neural Networks**. Workshop de Robótica Inteligente, Brasília, Agosto de 1997.
- [ROQ 97] ROQUE, W.L., HECKLER, C.A. On a Path Planning System for Autonomous Mobile Robots with Voronoi Diagrams and Visibility Pathways. **Workshop de Robótica Inteligente**, Brasília, Agosto de 1997.
- [RUD 93] RUDOLPH, F. A Trajectory Planner with Multiple Interconnected CMACs: Supervised Learning in a Redundant System Using Direct Inverse Modeling. In: World Congress on neural Networks. Oregon, USA. **Proceedings...** New Jersey: Lawrence Erlbaum Associates Inc. Publisher, 1993. v.3, p.198-1202.
- [SAC 74] SACERDOTI, E. Planning in a Hierarchy of Abstraction Spaces. *Artificial Intelligence*, v.5, n.2, pp.115-135.
- [SAC 75] SACERDOTI, E. A Structure for Plans and Behavior. Dissertação de Doutorado. SRI International, USA, 1975.
- [SAF 93] SAFFIOTTI, A., RUPINI, E. e KONOLIGE, K. Blending Reactivity and Goal-Directedness in a Fuzzy Controller, Second IEEE International Conference on Fuzzy Systems. **Proceedings ...** San Francisco, California, Março de 1993.
- [SCH 97] SCHNEIDER, A.M., BARONE, D.A.C., SILVA, J.C., BORGES, N.C.K. Speed Evaluation of a Genetic Algorithm for Mobile Robots Trajectory Control. **Workshop de Robótica Inteligente**, Brasília, Agosto de 1997.
- [SCH 97a] SCHNEIDER, A.M. Proposta de Técnicas Adaptativas Inteligentes para o Acompanhamento da Trajetória de Agentes Móveis Autônomos. Semana Acadêmica do CPGCC/UFRGS. **Anais**. CPGCC/UFRGS, Porto Alegre, 1997.
- [SEN 92] SENEVIRATNE, L. et al. Implementation of Robot Collision-free path planning in a maze using the triangulation algorithm. In: II Conference on Automation, Robotic and Computer Vision, 1992. **Proceedings ...** 1994.
- [SIL 97] SILVA, I.N., AMARAL, W.C., GOMIDE, F.A.C. A Fuzzy Approach to Mobile Robots Navigation. **Workshop de Robótica Inteligente**, Brasília, Agosto de 1997.
- [SIL 97a] SILVA JUNIOR, E.P. Tratamento de Obstáculos no Posicionamento de um Braço Mecânico Utilizando Redes Neurais Artificiais. **Trabalho Individual**, CPGCC/UFRGS, 1997.
- [SMI 94] SMITHERS, Tim. Taking Eliminative Materialism Seriously: A Methodology for Autonomous Systems Research. In: IWK'94. Ilmenau, Germany. **Proceedings...** 1994. v.1, p.102-108.
- [STE 90] STEELS, L. Exploiting Analogical Representations. **Designing Autonomous Agents**. Ed. P. Maes, MIT Press, Cambridge, pp.71-88.

- [SUK 94] SUKHAN, L. Sensor-Knowledge-Command Fusion Network for Intelligent Robotics. In: THE 1994 IEEE International Conference on Neural networks. Orlando, USA. **Proceedings...**Piscataway: IEEE press, 1994. v.4, p.2696-2702.
- [SUS 75] SUSSMAN, G. A **Computer Model of Skill Acquisition**. American Elsevier, New York, 1975.
- [TAN 92] TANG, K.; JARVIS, R. Collision-free path finding amongst polygonal obstacles using efficient free space triangulation. In: II Conference on Automation Robotic and Computer Vision, 1992. **Proceedings...** 1994. p. 11.1.1-11.1.5.
- [TAN 94] TANCHOCO, J. **Material Flow Systems in Manufacturing**. London: Chapman & Hall, 1994.
- [TIL 93] TILBURY, D. et al. Control Strategies for Mobile Robots with Trailers. **Technical Report**, EUA, 1993.
- [UCK 94] UCKUN, S.; ROTH, B.. A Control Architecture for Intelligent Mobile Robots. **Technical Report 93**. Knowledge Systems Laboratory, Stanford University, Palo Alto, 1994.
- [WAL 63] WALTER, W.G. **The Living Brain**. Norton, New York, 1963.
- [WAL 91] WALLICH, P. Silicon Babies. **Scientific American**, New York, v. 1, n. 12, p. 125-134, Dec. 1991.
- [WAL 93] WALNUM, C. **Adventures in Artificial Life**. USA: Que Corporation, 1993.
- [WEI 48] WEINER, N. **Cybernetics, or Control and Communication in Animals and Machines**. Willey, New York, 1948.
- [WID 60] WIDDROW, B., HOFF, M. E. Adaptive Switching Circuits. **IRE WESCON Convention Record**, v.4, pp.96-104.
- [WID 90] WIDROW, B.; LEHR, M. A. 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. **Proceedings of the IEEE**, v.78, n.9, USA, 1990.
- [WIL 93] WILSON, E.; ROCK, S. Experiments in Control of a Free-Flying Space Robot Using Fully-Connected Neural Networks. Oregon, USA. **Proceedings...** New Jersey: Lawrence Erlbaum Associates Inc. Publisher, 1993, v.3. UCB/ERL M93/23. College of Engineering, University of California, Berkeley, 1993.
- [WIN 87] WINSTON, P. **Inteligência Artificial**. Rio de Janeiro: Livros Técnicos e Científicos Editora Ltda, 1987. 500p.
- [WOL 91] WOLPERT, L. Pattern Formation. In: **The triumph of the Embryo**. Oxford: Oxford University Press, 1991. p. 31-73.
- [ZAD 83] ZADEH, L. Commonsense Knowledge Representation Based on Fuzzy Logic. **Computer**, p. 61-65. Oct. 1983.
- [ZAD 92] ZADEH, L.; KACPRZYK, J. **Fuzzy Logic for the Management of Uncertainty**. New York: John Wiley & Sons, Inc., 1992.
- [ZAR 93] ZAREMBA, M; PORADA, E. Neural Processing of Geometrical Features for Local Control of Mobile Robots. In: World Congress on Neural Networks. Oregon, USA. **Proceedings...** New Jersey:

- [ZIM 94] ZIMMER, Uwe; PUTTKAMER, Ewald. Comparing World-Modelling Strategies for Autonomous Mobile Robots. In: IWK'94. Ilmenau, Germany. **Proceedings...**1994. v.1, p.102-108.
- [ZIM 94b] ZIMMER, U. et al. Navigation on Topologic Feature-Maps. In: IIZUKA'94, Aug. 1994, Japan. **Pocceedings...** 1994. 250p. p.172-174.
- [ZIM 94c] ZIMMER, U. Connectionist Decision Systems for a Visual Search Problem. In: IIZUKA'94, Aug. 1994, Japan. **Pocceedings...** 1994. 250p. p.112-114.