
Estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs

Gustavo Pessin

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs

Gustavo Pessin

***Orientador:* Prof. Dr. Fernando S. Osório**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA.*

USP – São Carlos
Junho de 2013

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

P475e Pessin, Gustavo
Estratégias inteligentes aplicadas em robôs móveis
autônomos e em coordenação de grupos de robôs /
Gustavo Pessin; orientador Fernando S. Osório. --
São Carlos, 2013.
178 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2013.

1. Robôs Móveis Autônomos. 2. Aprendizado de
Máquina. 3. Localização. 4. Navegação. 5. Grupos de
Robôs. I. Osório, Fernando S. , orient. II. Título.

Agradecimentos

Como descrito na folha de rosto, essa tese, apresentada ao Instituto de Ciências Matemáticas e de Computação, é *requisito parcial* para obtenção do título de Doutor em Ciências de Computação e Matemática Computacional. Dizer que é *requisito parcial* minimiza a aflição de tentar espelhar quatro anos da vida em um texto de ≈ 150 páginas, afinal, muito mais foi realizado e nem tudo se encontra nesta tese. Esse período teve como maiores objetivos (i) o aprendizado, (ii) a tentativa de estender o estado da arte na área da robótica móvel e (iii) o desenvolvimento de *coisas* úteis e práticas para o mundo real. Tentei estender o estado da arte. Tentei desenvolver coisas úteis. Não posso dizer se consegui – não posso julgar essa parte. Entretanto, sei que o aprendizado obtido foi imenso. Muito maior que o imaginado.

Se cheguei até aqui, foi por não estar sozinho. Fernando S. Osório me ajudou a construir e remar este barco. Seu apoio, seu modelo de condução e sua paciência foram fundamentais no desenvolvimento desta tese. Muito obrigado Osório, não teria chegado aqui sem você!

Considerando válida a metáfora entre tese e barco, agradeço ao Prof. Denis F. Wolf por tentar me mostrar que os ventos mudam de direção e que a geografia submersa também impacta na navegação. Muito obrigado Denis, se cheguei perto de um trabalho útil, muito disso devo também a você!

O Prof. Jó Ueyama me mostrou, com muito bom humor, que existem diversos outros barcos no oceano e que existem outros oceanos a serem navegados. Muito obrigado Jó, se aprendi mais do que eu imaginava ser possível, devo isso também a você!

Devo um agradecimento especial a Jefferson R. Souza, meu calouro. Aprendi muito com nossas pesquisas, tanto no que se refere a métodos

científicos como no que se refere a ser um melhor ser humano. Muito obrigado Jefferson!

Agradeço ao Prof. Torsten Braun por ter me recebido com uma cordialidade impressionante e me auxiliado durante a visita técnica na Universität Bern, além das diversas inspirações e conselhos para o desenvolvimento de melhores trabalhos.

Agradeço também a Prof. Patrícia A. Vargas por ter me recebido e me auxiliado incondicionalmente durante meu ano na Heriot-Watt University. Neste período, os amigos Neil Mackenzie, Barry Hewitt e Ross Donald, do South Morningside Explorer Scout Unit (SMESU), foram mais que especiais.

Ainda, não posso deixar de lembrar, neste agradecimento, de todos os bons professores que fizeram parte da minha vida. Desde o ensino fundamental até as aulas na pós-graduação. Agradeço aos professores que mantiveram a semente da curiosidade viva dentro de mim. Além disso, agradeço aos diversos amigos e colegas dos laboratórios nos quais convivi e que foram responsáveis por excelentes discussões científicas e por auxílio na resolução de diversas dúvidas.

Agradeço também minha família e meus amigos: vocês foram fundamentais no desenvolvimento deste trabalho. O apoio da mãe, da Juli, do Vini, da Mano, da tia Eda, do Kuaki – e as cobranças do pai ;-) – sempre me ajudaram a manter o foco e ter a força necessária para realizar este trabalho.

Gostaria também de agradecer a CAPES pelo apoio financeiro e a secretaria do ICMC pelo auxílio com burocracias. Sem vocês tudo seria muito mais difícil.

Para finalizar, dizem que quando entramos numa nova igreja temos direito a fazer um pedido. Em geral, meus pedidos foram pra que Deus se preocupasse com pessoas mais necessitadas. Eu daria um jeito, sem incomodá-lo. Não sei se os pedidos foram atendidos, mas sei que quando precisei Ele estava de olho em mim! Muito obrigado!

Resumo

O contínuo aumento da complexidade no controle de sistemas robóticos, bem como a aplicação de grupos de robôs auxiliando ou substituindo seres humanos em atividades críticas tem gerado uma importante demanda por soluções mais robustas, flexíveis, e eficientes. O desenvolvimento convencional de algoritmos especializados, constituídos de sistemas baseados em regras e de autômatos usados para coordenar estes conjuntos físicos em um ambiente dinâmico é um desafio extremamente complexo. Diversos modelos de desenvolvimento existem, entretanto, muitos desafios da área da robótica móvel autônoma continuam em aberto. Esta tese se insere no contexto da busca por soluções inteligentes a serem aplicadas em robôs móveis autônomos com o objetivo de permitir a operação destes em ambientes dinâmicos. Buscamos, com a investigação e aplicação de estratégias inteligentes por meio de aprendizado de máquina no funcionamento dos robôs, a proposta de soluções originais que permitam uma nova visão sobre a operação de robôs móveis em três dos desafios da área da robótica móvel autônoma, que são: localização, navegação e operações com grupos de robôs.

As pesquisas sobre localização e coordenação de grupos apresentam investigação e propostas originais, buscando estender o estado da arte, onde apresentam resultados inovadores. A parte sobre navegação tem como objetivo principal ser um elo entre os conceitos de localização e coordenação de grupos, sendo o foco o desenvolvimento de um veículo autônomo com maior implicação em avanços técnicos.

Relacionado com a coordenação de grupos de robôs, fizemos a escolha de trabalhar sobre uma aplicação modelada como o problema de combate a incêndios florestais. Buscamos desenvolver um ambiente de simulação realístico, onde foram avaliadas quatro técnicas para busca de

estratégias de formação do grupo: Algoritmos Genéticos, Otimização por Enxame de Partículas, *Hill Climbing* e (iv) *Simulated Annealing*. Com base nas diversas avaliações realizadas pudemos mostrar quais das técnicas e conjuntos de parâmetros permitem a obtenção de resultados mais acurados que os demais. Além disso, mostramos como uma heurística baseada em populações anteriores pode auxiliar na tolerância a falhas da operação.

Relacionado com a tarefa de navegação, apresentamos o desenvolvimento de um veículo autônomo de grande porte funcional para ambientes externos. Buscamos aperfeiçoar uma arquitetura para navegação autônoma, baseada em visão monocular e com capacidade de seguir pontos esparsos de GPS. Mostramos como a simulação e os usos de robôs de pequeno porte auxiliaram no desenvolvimento do veículo de grande porte e apresentamos como as redes neurais podem ser aplicadas nos modelos de navegação autônoma.

Na investigação sobre localização, mostramos um método utilizando informação obtida de redes sem fio para prover informação de localização para robôs móveis. As informações obtidas da rede sem fio são utilizadas para aprendizado da posição de um robô móvel por meio de uma rede neural. Diversas avaliações foram realizadas buscando entender o comportamento do sistema com diferentes números de pontos de acesso, com uso de filtros, com diferentes topologias. Os resultados mostram que o modelo usando redes sem fio pode ser um possível método prático e barato para localização de robôs móveis.

Esta tese aborda temas relevantes e propostas originais relacionadas com os objetivos propostos, apresentando métodos que provenham autonomia na coordenação de grupos e nas atividades individuais dos mesmos. A busca por altos graus de eficiência na resolução de tarefas em ambientes dinâmicos ainda é um campo que carece de soluções e de um aprofundamento nas pesquisas. Sendo assim, esta pesquisa buscou agregar diversos avanços científicos na área de pesquisa de robôs móveis autônomos e coordenação de grupos, por meio da aplicação de estratégias inteligentes.

Abstract

The constant increasing of the complexity in the control of robotic systems, as well as the application of groups of robots assisting or replacing human beings in critical activities has generated a significant demand for more robust, flexible and efficient solutions. The conventional development of specialized algorithms consisted of rule-based systems and automatas, used to coordinate these physical sets in a dynamic environment is an extremely complex challenge. Although several models of development of robotic issues are currently in use, many challenges in the area remain open. This thesis is related to the search for intelligent strategies to be applied in autonomous mobile robots in order to allow practical operations in dynamic environments. We seek, with the investigation of intelligent strategies by means of the use of machine learning in the robots, to propose original solutions to allow contributions in three challenges of the robotic research area: localization, navigation and coordination of groups of robots.

The investigations about localization and groups of robots show novel and original proposals, where we sought to extend the state of the art. The navigation part has as its major objective to be a link between the subjects of localization and navigation, being its aim to help the deployment of an autonomous vehicle implying in greater technical advances.

Related to the robotic group coordination, we have made the choice to work on an application modeled as a wildfire combat operation. We have developed a simulation environment in which we have evaluated four techniques to obtain strategies for the group formation: genetic algorithms, particle swarm optimization, hill climbing and simulated annealing. The

results showed that we can have very different accuracy with different techniques and sets of parameters. Furthermore, we show how a heuristic based on the use of past populations can assist in fault tolerant operation.

Related to the autonomous navigation task, we present the development of a large autonomous vehicle capable of operating in outdoor environments. We sought to optimize an architecture for autonomous navigation based on monocular vision and with the ability to follow scattered points of GPS. We show how the use of simulation and small robots could assist in the development of large vehicle. Furthermore, we show how neural networks can be applied as a controller to autonomous navigation systems.

In the investigation about localization, we presented a method using wireless networks to provide information about localization to mobile robots. The information gathered by the wireless network is used as input in an artificial neural network which learns the position of the robot. Several evaluations were carried out in order to understand the behavior of the proposed system, as using different topologies, different numbers of access points and the use of filters. Results showed that the proposed system, using wireless networks and neural networks, may be a useful and easy to use solution for localization of mobile robots.

This thesis has addressed original and relevant topics related to the proposed objectives, showing methods to allow degrees of autonomy in robotic operations. The search for higher degrees of efficiency in tasks solving in dynamic environments is still a field that lacks solutions. Therefore, this study sought to add several scientific contributions in the autonomous mobile robots research area and coordination of groups, by means of the application of intelligent strategies.

Sumário

| | |
|---|-------------|
| Agradecimentos | i |
| Resumo | iii |
| Abstract | v |
| Lista de Figuras | xi |
| Lista de Tabelas | xvii |
| Lista de Siglas | xix |
| 1 Introdução | 1 |
| 1.1 Contextualização | 1 |
| 1.2 Definição do Problema e Abordagens | 4 |
| 1.3 Objetivos | 7 |
| 1.4 Organização do Texto | 10 |
| 2 Robótica Móvel | 13 |
| 2.1 Robôs Móveis Autônomos | 13 |
| 2.1.1 Abordagens para sistemas com múltiplos robôs | 20 |
| 2.1.2 Características básicas de sistemas com múltiplos robôs | 22 |
| 2.2 Simulação e Modelagem | 24 |
| 2.3 Enxame Robótico | 26 |
| 2.4 Robótica Evolucionária | 29 |
| 2.5 Considerações Finais | 32 |

| | | |
|----------|---|------------|
| 3 | Aprendizado de Máquina | 35 |
| 3.1 | Aprendizado Evolutivo | 36 |
| 3.1.1 | Otimização por Enxame de Partículas | 36 |
| 3.1.2 | Algoritmos Genéticos | 39 |
| 3.1.3 | Arrefecimento Simulado | 42 |
| 3.2 | Aprendizado Conexionista | 43 |
| 3.2.1 | Redes Neurais Artificiais | 43 |
| 3.3 | Considerações Finais | 46 |
| 4 | Localização | 49 |
| 4.1 | Motivação | 51 |
| 4.2 | Metodologia | 54 |
| 4.3 | Resultados | 57 |
| 4.3.1 | Investigação sobre o efeito do uso de filtros | 57 |
| 4.3.2 | Investigação sobre o número de pontos de acesso | 61 |
| 4.3.3 | Efeito da mudança da topologia | 63 |
| 4.3.4 | Evolução de topologias de RNA para Localização | 65 |
| 4.3.5 | Outras experiências com evolução | 73 |
| 4.3.6 | Tolerância a falhas | 74 |
| 4.3.7 | Uma abordagem para maior autonomia | 75 |
| 4.4 | Considerações Finais | 76 |
| 4.5 | Trabalhos Futuros | 78 |
| 5 | Navegação Autônoma | 81 |
| 5.1 | Da Simulação aos Robôs Reais | 84 |
| 5.2 | Navegação Autônoma em Ambientes Externos | 88 |
| 5.2.1 | Identificação da pista | 90 |
| 5.2.2 | Identificação de geometrias | 92 |
| 5.2.3 | Navegação baseada em aprendizado | 92 |
| 5.2.4 | Resultados | 93 |
| 5.3 | Navegação com Mapeamento | 98 |
| 5.4 | Considerações Finais | 100 |
| 5.5 | Trabalhos Futuros | 102 |
| 6 | Grupos de Robôs | 105 |
| 6.1 | Descrição Geral | 107 |
| 6.2 | Evolução de Estratégias | 109 |
| 6.2.1 | Escolha do <i>fitness</i> | 111 |
| 6.2.2 | Simulador de incêndio | 112 |
| 6.3 | Investigações em Formação de Grupos | 114 |
| 6.3.1 | Tolerância a falhas | 116 |
| 6.3.2 | Buscando melhorias na eficiência do AG | 119 |
| 6.3.3 | Otimização por enxame de partículas | 123 |
| 6.3.4 | Outras técnicas de otimização | 127 |
| 6.4 | Abordagens Baseadas em Etologia e Engenharia | 130 |

| | | |
|----------|---|------------|
| 6.5 | Considerações Finais | 131 |
| 6.6 | Trabalhos Futuros | 133 |
| 7 | Considerações Finais | 135 |
| 7.1 | Revisão do Tema | 135 |
| 7.2 | Resultados e Contribuições | 138 |
| 7.2.1 | Localização | 138 |
| 7.2.2 | Navegação Autônoma | 140 |
| 7.2.3 | Grupos de Robôs | 142 |
| 7.3 | Trabalhos Futuros | 144 |
| | Referências Bibliográficas | 147 |
| A | Artigos Publicados | 165 |
| A.1 | Publicados em periódicos | 165 |
| A.2 | Publicados em conferências | 166 |
| B | Incêndios em Ambientes Naturais | 169 |
| B.1 | Combustíveis Florestais | 170 |
| B.2 | Técnicas Reais de Operação | 172 |
| B.2.1 | Operações de combate | 172 |
| B.2.2 | Equipamentos de combate | 173 |
| B.2.3 | Métodos e estratégias de combate | 174 |
| B.2.4 | Sistemas de detecção | 175 |
| B.3 | Considerações sobre Incêndios em Ambientes Naturais | 176 |

Lista de Figuras

| | | |
|-----|--|----|
| 1.1 | Exemplos de formações para equipes de quatro agentes. Da esquerda para a direita: linha, coluna, circular e ferradura (Balch e Arkin, 1998). | 6 |
| 2.1 | Robôs aéreos de combate: (a) <i>X-45A</i> (Boeing, 2013); (b) <i>nEUROn</i> (Dassault, 2013). Robôs aquáticos de exploração: (c) <i>Depthx</i> (Wettergreen, 2013); (d) <i>Isis</i> (NOC, 2013). Robôs terrestres: (e) <i>R-Gator</i> (Deere, 2013), um veículo que permite teleoperação, navegação automática entre pontos tipo “ensinar-repetir”, desvio de obstáculos e operação manual; (f) <i>Urbie</i> (Matthies, 2013), robô para fins militares de reconhecimento de terrenos urbanos. | 16 |
| 2.2 | Robôs de campo: (a) <i>Mars Exploration Rover</i> (JPL/NASA, 2013) para exploração espacial; (b) <i>Dante II</i> (Bares e Whittaker, 2013) para exploração de cavernas e vulcões. Robôs de serviço: (c) Aspirador de pó <i>Roomba</i> (iRobot, 2013); (d) Colheitadora autônoma (NREC/CMU, 2013). | 17 |
| 2.3 | Formigas trabalhando cooperativamente na criação de uma ponte, permitindo a passagem de outras formigas carregando comida. Crédito: (Metcalf, 2010). . . | 27 |
| 2.4 | Exemplos de trabalho cooperativo. (a) Transporte de comida. As formigas sobre a folha estão em postura defensiva contra moscas parasitárias. (b) Ataque a uma formiga inimiga. Créditos: (Wild, 2010). | 28 |
| 3.1 | Inspirações do PSO. (a) Bando de pássaros. Crédito: (Prestri, 2010). (b) Cardume de peixes. Crédito: (Hall, 2010). | 36 |
| 3.2 | Fluxograma de um PSO padrão. | 37 |
| 3.3 | Cruzamento e mutação, adaptado de Lacerda e Carvalho (1999). | 41 |
| 3.4 | Neurônio de McCulloch e Pitts (MCP), adaptado de (Braga <i>et al.</i> , 2007). | 44 |
| 3.5 | Descida do gradiente de uma superfície de erro. | 45 |
| 3.6 | Curvas de erro em aprendizado e validação. | 45 |

| | | |
|------|--|----|
| 4.1 | (a) Representação gráfica da área de trabalho considerando uma área de 180 cm x 180 cm. (b) Foto parcial do ambiente com o robô, similar a representação gráfica da Figura 4.1(a). Cada pequena cruz está posicionada a 60 cm uma da outra. Em (a), a posição dos pontos de acesso é aproximada. | 55 |
| 4.2 | (a) Exemplo de comportamento do filtro de mediana móvel. A linha azul apresenta o valor bruto da leitura de um ponto de acesso. A linha preta apresenta como o filtro de mediana móvel remove parte dos ruídos (janela de 8 leituras). (b) Exemplo de topologia de rede neural utilizada. | 56 |
| 4.3 | Resultados usando o filtro de mediana móvel para diferentes tamanhos de janelas ($n = 30$). Podemos ver que quanto maior a janela de tempo, menor o erro no aprendizado. | 58 |
| 4.4 | Histogramas dos resultado do uso da melhor RNA de cada janela de tempo ($n \approx 900$). | 59 |
| 4.5 | Secção do plano (Figura 4.1) com valores esperados e obtidos usando diferentes janelas de tempo na mediana. (a) Valor bruto. (b) Usando filtro de mediana de 10 leituras. (c) Usando filtro de mediana de 60 leituras. Os eixos x e y estão em centímetros. | 60 |
| 4.6 | Amplitude do sinal lido de cada antena (AP1 até AP8). | 62 |
| 4.7 | Erro quadrático médio resultante de 30 execuções de cada rede neural com diferentes quantidades de entradas. | 62 |
| 4.8 | Histogramas dos resultados do uso da melhor rede neural considerando diferentes entradas ($n \approx 900$). | 63 |
| 4.9 | Erro quadrático médio resultante das avaliações das diferentes topologias de redes neurais ($n = 30$). | 64 |
| 4.10 | Fluxograma do AG responsável pela evolução da topologia da RNA. | 66 |
| 4.11 | Topologias obtidas pela evolução. (a) Gráfico de pontos do número de neurônios. (b) Gráfico de pontos do número de conexões. Podemos ver que em ambos os valores, a evolução baseada em apenas uma execução da rede neural apresentou a maior dispersão nos resultados. | 68 |
| 4.12 | Erro quadrático médio resultante das avaliações com as RNAs obtida pela evolução – redes apresentadas na Tabela 4.6. | 69 |
| 4.13 | Resultados de 30 execuções com variação no número de ciclos de treino e validação sobre a rede E_{10b} . O erro está mostrado como MSE. As referências no eixo x representam número de ciclos de treino e validação, por exemplo, 2k20 representa 2.000 ciclos de treino e validação a cada 20 ciclos. | 70 |
| 4.14 | Erro médio quadrático resultante da comparação entre a rede obtida em (Pessin <i>et al.</i> , 2011a) e a rede evoluída neste trabalho. Resultados de 30 execuções. A hipótese de igualdade não é rejeitada pelo teste estatístico. | 72 |
| 4.15 | Resultados para a rede com evolução de pesos (E_2) e para a rede treinada (E_1). Além do treino ser mais rápido do que a evolução, a rede treinada apresenta resultados $\approx 20\%$ melhores do que a rede com evolução de pesos. | 74 |

| | | |
|------|--|----|
| 5.1 | (a) CaRINA I – primeiro veículo autônomo de grande porte do LRM/ICMC/USP. A implementação do controle de atuação permitiu o começo de sua utilização no início de 2010. (b) CaRINA II – segundo veículo autônomo de grande porte do LRM/ICMC/USP. A implementação do controle de atuação permitiu o começo de sua utilização no fim de 2012. | 82 |
| 5.2 | Ambientes de simulação para grupos de robôs. (a), (b) e (c) Ambiente de simulação utilizado em (Pessin, 2008; Pessin <i>et al.</i> , 2010a) – conta com terrenos irregulares e suporte a características ambientais (<i>e.g.</i> fogo, vento). (d) Ambiente de simulação simplificado, com foco em tarefa de navegação autônoma e desvio de obstáculos utilizado em (Souza <i>et al.</i> , 2010). | 85 |
| 5.3 | Robôs de pequeno porte. (a) <i>Setup</i> do experimento usando o robô Pioneer P3-AT. (b) Robô de pequeno porte SRV-1Q (Surveyor, 2010). | 86 |
| 5.4 | Sistema de esterçamento do robô CaRINA I (Fernandes <i>et al.</i> , 2012). Em detalhes: (1) motor, (2) sistema de polia de acoplamento, (3) componente do sistema de direção, (4) encoder digital, (5) interruptor de acoplamento, (6) alavanca de acoplamento, (7) controlador de motor RoboteQ, (8) dispositivos limitadoras. | 87 |
| 5.5 | Dispositivo controlador de aceleração (Fernandes <i>et al.</i> , 2012). Na direita, vista do interior com destaque para o controlador Arduino (Arduino, 2013) e o acoplamento mecânico entre servo motor e potenciômetro. | 87 |
| 5.6 | (a) Trajeto percorrido pelo veículo considerando o primeiro ambiente (≈ 200 m). P0 até P6 especificam os pontos das coordenadas de GPS utilizados. A linha amarela apresenta o trajeto percorrido aproximado. (b) Trajeto percorrido pelo veículo considerando o segundo ambiente ($\approx 1,08$ km). P0 até P6 especificam os pontos das coordenadas de GPS utilizados. A linha amarela apresenta o trajeto percorrido aproximado. As linhas vermelhas apresentam a orientação calculada pelo veículo para ir de um ponto a outro. | 89 |
| 5.7 | Modelo completo do controlador de navegação. | 90 |
| 5.8 | Imagem com os blocos usados na classificação (Shinzato e Wolf, 2011). | 91 |
| 5.9 | (a) Imagem obtida da câmera. (b) Comitê de redes neurais para identificação de áreas navegáveis ou não. (c) Mapa de navegabilidade: tons mais claros representam zonas mais seguras, tons mais escuros representam zonas inseguras (não navegáveis). | 91 |
| 5.10 | Cinco possíveis geometrias definindo as direções que o veículo pode seguir. Da esquerda para a direita: curva acentuada para a esquerda, curva leve a esquerda, em frente, curva leve a direita, curva acentuada a direita. | 92 |
| 5.11 | Estrutura da rede neural utilizada para a aceleração e esterçamento do veículo (Souza <i>et al.</i> , 2013). PO significa percentual de ocupação. Diferença de orientação é a diferença entre a orientação do veículo e a orientação para o destino (coordenada GPS). A orientação para o destino também é chamada de azimute. | 93 |
| 5.12 | Histograma do erro das saídas (valor obtido - valor esperado) considerando as melhores redes neurais das topologias 4 (execução 3, HT) e 5 (execução 4, LS). | 95 |

| | | |
|------|---|-----|
| 5.13 | Gráfico de valores esperados e obtidos pela RNA, considerando o primeiro trajeto (mostrado na Figura 5.6(a)) – valores da base de validação limitados na primeira centena para fins de melhor visualização. (a) Esterçamento (em radianos). (b) Velocidade (em m/s). | 97 |
| 5.14 | Histograma do erro das saídas considerando novos treinos usando as topologias 4 e 5 para o segundo trajeto. (a) Topologia 4. (b) Topologia 5. | 98 |
| 5.15 | Gráfico de valores esperados e obtidos pela RNA, considerando o segundo trajeto (mostrado na Figura 5.6(b)) – valores da base de validação limitados na primeira centena para fins de melhor visualização. (a) Esterçamento (em radianos). (b) Velocidade (em m/s). | 99 |
| 5.16 | Dados de mapeamento (Fernandes <i>et al.</i> , 2012; Hata <i>et al.</i> , 2009a). Detecção de áreas navegáveis (branco) e não navegáveis (vermelho). Os pontos verdes representam a pose do veículo durante a navegação e captura dos dados – fusão dos dados de <i>laser</i> , IMU e GPS. | 100 |
| 6.1 | Diagrama da operação multi-agente. | 108 |
| 6.2 | Fases do sistema proposto. | 108 |
| 6.3 | Motoniveladora (Caterpillar, 2013). | 110 |
| 6.4 | (a) Fragmento de carta topográfica de UTM (0578000, 6764000) até (0584000, 6770000) representando um quadrante de 6km x 6km. (b) Mapa de modelos de combustíveis (MA, 2006). | 113 |
| 6.5 | (a) e (b) Resultados insatisfatórios, os aceiros são excessivamente grandes e mal distribuídos entre os robôs. (c) e (d) Resultados satisfatórios. (e) Visão detalhada da navegação dos robôs móveis. (f) Resultado satisfatório aplicado no protótipo 3D. | 117 |
| 6.6 | (a) e (b) Reorganização com envio de mais dois robôs: (a) Melhor indivíduo após primeira geração sem heurística e (b) Melhor indivíduo após primeira geração com heurística. (c) e (d) Reorganização para 3 robôs: (c) Melhor indivíduo após primeira geração sem heurística e (d) Melhor indivíduo após primeira geração com heurística. | 119 |
| 6.7 | Evolução do <i>fitness</i> de acordo com o número de gerações (reorganização para 3 robôs de combate, com e sem heurística). O eixo <i>x</i> está limitado em 150 gerações. | 120 |
| 6.8 | Resultados das avaliações apresentadas na Tabela 6.4. O eixo <i>x</i> apresenta a descrição do conjunto de parâmetros como (método de escala; percentual de sobreposição). | 122 |
| 6.9 | (a) Resultados das avaliações considerando os quatro melhores conjuntos de parâmetros da Figura 6.8 usando diferentes tamanhos de população. O eixo <i>x</i> apresenta a descrição do conjunto de parâmetros como (método de escala; percentual de sobreposição; número de indivíduos). (b) Gráfico de evolução do <i>fitness</i> de acordo com o número de gerações. | 122 |
| 6.10 | (a) Resultados das avaliações apresentadas na Tabela 6.7. O eixo <i>x</i> apresenta o conjunto de parâmetros do experimento como (inércia; tamanho do enxame). (b) Curvas de queda de <i>fitness</i> de acordo com o número de gerações, para diferentes tamanhos de enxame. Inércia de 0,5. | 126 |

| | | |
|------|---|-----|
| 6.11 | (a) Resultados das avaliações apresentadas na Tabela 6.8. O eixo x apresenta a descrição do conjunto de parâmetros como (método; número de vizinhos; mutação). (b) Curvas de queda de <i>fitness</i> dos melhores HC e SA obtidos nesta avaliação – HC e SA com 200 vizinhos e 10% de mutação. Curva de queda de <i>fitness</i> do AG (Pessin e Osório, 2009b) – AG com 150 indivíduos, 10% de mutação, cruzamento de dois pontos. Curva de queda de <i>fitness</i> do PSO (Pessin <i>et al.</i> , 2010c) – PSO com 200 partículas, inércia de 0,5 e modelo completo. . . . | 128 |
| B.1 | Avião-tanque <i>Canadair CL-415</i> (Bombardier, 2007), transporta 6 t de água, abastece sem pousar em uma superfície de água com 1,2 km de comprimento e 2 m de profundidade. | 174 |
| B.2 | (a) Foco de incêndio próximo ao Parque Nacional dos Aparados da Serra, imagem da Divisão de Processamento de Imagens do INPE, satélite <i>LandSat 2000</i> (DPI/INPE, 2007). (b) Tabela de informações disponibilizada junto a imagem do foco de incêndio (DPI/INPE, 2007). | 177 |

Lista de Tabelas

| | | |
|-----|---|----|
| 4.1 | Sumário dos resultados do uso da melhor RNA de cada tamanho de janela ($n \approx 900$). O erro é apresentado em centímetros e representa a distância vetorial entre a posição esperada e a posição obtida.) | 61 |
| 4.2 | Valores-p resultantes de teste de comparação de distribuições. Com exceção das distribuições Q3 e Q4, as demais apresentam rejeição na hipótese de igualdade. | 62 |
| 4.3 | Sumário dos resultados do uso da melhor RNA de cada conjunto de entradas ($n \approx 900$). Erro em centímetros apresentado como a distância vetorial entre a posição esperada e a posição obtida. | 63 |
| 4.4 | Erro quadrático médio resultante das avaliações das diferentes topologias de redes neurais ($n = 30$). | 64 |
| 4.5 | Valores-p resultantes do teste de Wilcoxon para as topologias avaliadas. Em negrito, conjuntos com valor-p maior que 0,05 (igualdade não rejeitada considerando 95% de confiança). | 65 |
| 4.6 | Topologias obtidas pela evolução. Resultado de 5 rodadas com cada conjunto de configurações. | 68 |
| 4.7 | Sumário numérico das 30 execuções de cada RNA obtida pela evolução – redes apresentadas na Tabela 4.6. O erro está mostrado como MSE. | 70 |
| 4.8 | Resultados da comparação estatística entre 100k20 (100.000 ciclos de treino com validação a cada 20 ciclos) e os demais. | 71 |
| 4.9 | Sumário numérico dos resultado da comparação entre a rede obtida em (Pessin <i>et al.</i> , 2011a) (C) e a rede obtida por evolução (T) neste trabalho. Resultado de 30 execuções. Erro apresentado como MSE. A hipótese de igualdade não é rejeitada pelo teste estatístico. | 72 |

| | | |
|-----|--|-----|
| 5.1 | Atributos de entrada das diferentes RNAs usadas como classificadores de áreas navegáveis ou não, de (Shinzato e Wolf, 2011). Componentes R, G, B são respectivamente vermelho (<i>red</i>), verde (<i>green</i>) e azul (<i>blue</i>). Componentes H, S, V são respectivamente brilho (<i>hue</i>), saturação (<i>saturation</i>), e valor (<i>value</i>). Y, U, V = Luminância-Crominância. | 91 |
| 5.2 | Resultado das avaliações considerando diferentes topologias e diferentes funções de transferências (<i>Logistic Sigmoid</i> – LS e <i>Hyperbolic Tanget</i> – HT). O erro é apresentado como MSE (10^{-1}]. Também é apresentado o resultado do Ponto Ótimo de Generalização (POG) de cada execução das RNAs. Asterisco apresenta os melhores (menores) valores. | 94 |
| 5.3 | Valores-p do resultado de teste de normalidade de <i>Shapiro-Wilk</i> . Asterisco apresenta valor onde hipótese de normalidade não é rejeitada (valor-p maior que 0,05). 95 | 95 |
| 5.4 | Valores-p do resultado do teste de Wilcoxon-Mann-Whitney para verificação de similaridade de conjuntos. Asterisco apresenta valor onde existe rejeição da hipótese de igualdade (valor-p inferior a 0,05) | 96 |
| 5.5 | Sumário dos resultados dos novos treinos das Topologias 4 e 5 para o segundo trajeto. | 96 |
| 6.1 | Estrutura proposta para solução (grupo com quatro robôs – ângulos e raios relativos ao ponto inicial do incêndio). | 110 |
| 6.2 | Lista das variações de parâmetros avaliados no Algoritmo Genético para definição das posições de atuação. | 115 |
| 6.3 | Segunda lista das variações de parâmetros avaliados no Algoritmo Genético para definição das posições de atuação. Utilizamos como base os cinco melhores conjuntos da Tabela 6.2, com aumento no número de indivíduos (de 100 para 150) e no número de gerações (de 500 para 700). Experiência <i>I'</i> permitiu obter 100% dos resultados finais com <i>fitness</i> abaixo de 0,14. | 116 |
| 6.4 | Lista de variações de parâmetros no AG considerando uso de populações sobrepostas e de escalas de <i>fitness</i> | 120 |
| 6.5 | Conjunto de avaliações iniciais utilizando PSO. | 124 |
| 6.6 | Novo conjunto de avaliações considerando PSO com modelo social. | 125 |
| 6.7 | Novo conjunto de avaliações considerando PSO com modelo completo. | 125 |
| 6.8 | Conjunto de parâmetros avaliados no HC e SA. | 127 |

Lista de Siglas

| | |
|----------|---|
| AG | Algoritmo Genético (<i>Genetic Algorithms</i>) |
| AP | <i>Access Point</i> (Ponto de Acesso) |
| CaRINA | Carro Robótico Inteligente para Navegação Autônoma |
| ER | <i>Evolutionary Robotics</i> (Robótica Evolucionária) |
| GPS | <i>Global Positioning System</i> |
| HC | <i>Hill Climbing</i> (Subida de Encosta) |
| HT | <i>Hyperbolic Tangent</i> |
| INCT-SEC | Instituto Nacional de Ciência e Tecnologia - Sistemas Embarcados Críticos |
| LRM | Laboratório de Robótica Móvel |
| LS | <i>Logistic Sigmoid</i> |
| ML | <i>Machine Learning</i> (Aprendizado de Máquina) |
| MSE | <i>Mean squared error</i> (Erro Quadrático Médio) |
| PSO | <i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas) |
| RMA | Robôs Móveis Autônomos |
| RNA | Rede Neural Artificial |
| RPROP | <i>Resilient Backpropagation</i> |
| RSSI | <i>Received Signal Strength Indication</i> |
| SA | <i>Simulated Annealing</i> (Arrefecimento Simulado) |
| SNNS | <i>Stuttgart Neural Network Simulator</i> |
| SRRS | <i>Stochastic Remainder Sampling Selector</i> |

Introdução

Este capítulo inicia com a apresentação do contexto no qual este trabalho está inserido, bem como suas principais motivações. Na sequência, são descritos os objetivos gerais e específicos que nortearam o desenvolvimento das investigações. Finalizando, é apresentada a estrutura e a organização dos capítulos subsequentes desta tese.

1.1 Contextualização

Dotar robôs de capacidade de autonomia, inteligência e de interação com o meio em que estão inseridos é uma área de pesquisa que tem atraído a atenção de um grande número de pesquisadores (Dudek e Jenkin, 2000). A evolução proporcionada pelas pesquisas em robótica móvel tem desenvolvido robôs cada vez mais eficientes, onde estudos sobre formas de controle das complexidades motoras tem sido realizados em diversos centros de pesquisa ao redor do mundo, englobando o desenvolvimento de sensores e atuadores bem como diversas pesquisas relacionadas ao *hardware* e *software* robótico, como apresentam Siegwart *et al.* (2011), Bekey (2005) e Buehler *et al.* (2009). O desenvolvimento de algoritmos especializados, constituídos de sistemas baseados em regras e de autômatos usados para coordenar estes conjuntos físicos em um ambiente dinâmico é um desafio extremamente complexo (Go *et al.*, 2004; Nolfi e Floreano, 2000).

Um robô móvel autônomo (RMA) pode ser definido como um agente inteligente (Garcia e Sichman, 2003). *Móvel* para ter capacidade de locomoção em um ambiente; *autônomo* para ter o mínimo necessário de intervenção e controle por parte de seres humanos; e, *inteligente* para, a partir de dados de sensores, poder atuar e se adaptar de maneira coerente ao ambiente onde está inserido. A robótica móvel é uma área de pesquisa que lida com o controle de veículos autônomos ou semi-autônomos (Bekey, 2005; Dudek e Jenkin, 2000).

O que diferencia a robótica móvel de outras áreas de pesquisa em robótica, como a robótica de manipuladores, é sua ênfase nos problemas relacionados com a operação (movimentação) em ambientes que se modificam dinamicamente, compostos de obstáculos estáticos e móveis. Para operar neste tipo de ambiente o robô deve ser capaz de adquirir e utilizar conhecimento sobre o ambiente, estimar uma posição, possuir a habilidade de reconhecer obstáculos e responder em tempo real as diferentes situações que possam ocorrer; todas estas funcionalidades devem operar em conjunto. As tarefas de *perceber*, *se localizar*, e *se mover* pelo ambiente evitando colisões e realizando tarefas são *problemas fundamentais* no estudo dos RMAs (Bekey, 2005; Dudek e Jenkin, 2000).

Como ressaltado por Dudek e Jenkin (2000) e Bekey (2005), um dos problemas fundamentais da robótica é a navegação: processo de mover-se em um ambiente de trabalho, geralmente povoado de obstáculos, de uma pose¹ inicial para uma pose destino (Siegwart *et al.*, 2011). Ir de um lugar para outro, de forma minimamente planejada (não *wandering*), depende de alguns aspectos fundamentais: pose inicial (localização mais orientação), posição ou pose destino² e controle motor. Para conhecer sua pose inicial, um RMA deve possuir sensores adequados (*e.g.* GPS³, bússola). Para conhecer sua pose destino, é necessário ou comunicação (por exemplo, para receber por mensagem uma coordenada destino, utilizada para cálculo do azimute) ou outros esquemas (*e.g.* mapeamento) que forneçam ao robô a coordenada por meio de planejamento ou pré-conhecimento. Em se tratando de navegação em modo *wandering*, é comum que o conhecimento de poses iniciais e finais não sejam relevantes. O modo *wandering* é comumente utilizado para testes básicos de capacidade de navegação com desvio de obstáculos. Para o controle motor, um RMA deve possuir um número adequado de motores (um veículo autônomo, por exemplo, usualmente possui um motor angular, para o controle da direção das rodas e um motor para o deslocamento linear, controlando a tração das rodas).

¹Em robótica, o termo *pose* usualmente se refere a localização como (x, y) ou (x, y, z) mais orientação (θ) .

²Podemos considerar que o destino do robô pode ser dado por meio do conhecimento de uma pose (x, y, θ) ou de uma posição (x, y) . A diferença básica entre os modelos é que no primeiro, o robô poderia ser capaz de terminar sua operação em uma orientação específica.

³GPS pode ser pensado como um sistema, afinal, é a sigla de *Global Positioning System*, todavia, na área de robótica, é comum pensar em GPS como sendo um sensor que provê localização.

O atual sistema de GPS é bastante útil para obtenção de posição (localização) em ambientes externos. Entretanto, é comum que apresente imprecisão de alguns metros. Algumas exceções que prometem maior acuracidade existem, todavia são extremamente caras. Nos modelos mais comuns, correções por *software* têm sido aplicadas, bem como a integração com outros sensores (como unidades inerciais). Mesmo assim, em alguns ambientes externos, o sistema de GPS continua a apresentar maiores graus de imprecisão ou mesmo deficiência de funcionamento, como em áreas com grande cobertura vegetal, proximidade a grandes blocos de rochas – problemas de ponto cego ou refração. Informação sobre localização é base para o desenvolvimento de métodos inteligentes de navegação.

A construção de um robô fisicamente simulado é uma opção que permite reduzir os custos da implementação física e a realização de experimentos garantindo uma maior segurança, porém é quase tão árdua quanto a construção de um robô real, salvo algumas restrições de projeto. Pieri (2002) discute algumas questões relacionadas com desafios básicos da robótica móvel, seja na implementação de um sistema real ou simulado. Estes desafios são: (i) *Reconhecimento*: Como fazer um dispositivo robótico capaz de reconhecer o ambiente no qual está inserido? Existem sérias restrições computacionais com relação ao processamento de dados sensoriais. (ii) *Navegação*: Como criar formas que permitam a representação de um ambiente? Até que ponto o conhecimento e a representação de um ambiente precisa ser refinado e livre de erros? Até que ponto podemos utilizar a navegação reativa e ainda assim maximizar o desempenho do robô? (iii) *Aprendizado*: Adicionar sensores e atuadores a um robô adiciona complexidade ao trabalho de programação, dificultando a tarefa de calibração dos sensores e o controle de ações. Técnicas de Aprendizado de Máquina devem permitir que um robô desempenhe a sua tarefa de forma mais autônoma e adaptável. (iv) *Cooperação*: Um grupo de robôs pequenos e baratos pode realizar qualquer tipo de tarefa complexa apenas usando cooperação? (v) *Cognição*: Até que ponto podemos criar formas para armazenar conhecimento e utilizá-lo para a tomada de decisão dotando o robô de capacidades cognitivas de alto nível que o permitam ter total autonomia com relação ao meio?

Outra maneira de pensar em desafios da robótica foi sugerida por Russel e Norvig (2002) e se deve as seguintes constatações: (i) *O mundo é inacessível*: Sensores não são perfeitos em perceber o mundo real. (ii) *O mundo é não-determinístico*: Devido as incertezas envolvidas, o resultado final não pode ser completamente conhecido. (iii) *O mundo é não-episódico*: Os efeitos de uma ação mudam com o passar do tempo, problemas relacionados a decisão e aprendizado devem ser tratados. (iv) *O mundo é dinâmico*: É preciso saber quando planejar e quando agir. (v) *O mundo é contínuo*: A evolução dos estados internos da programação do robô evoluem de forma discreta enquanto que o mundo ao redor é contínuo, dificultando a programação e a estimativa de ações e comportamentos.

Estes desafios tem sido foco constante de pesquisas. Novas propostas têm surgido solucionando alguns aspectos pontuais destes desafios. Entretanto, a maioria das soluções não apresenta características de aceitação universal. Grande parte dos desenvolvimentos são bastante focados em aplicações específicas para dispositivos físicos e ambientes pré-determinados. Soluções mais flexíveis, tolerantes a falha e escaláveis ainda são desejáveis. Considerando as constatações e os desafios descritos anteriormente, na seção subsequente iremos apresentar os problemas a serem abordados nesta tese. Por meio da aplicação de estratégias inteligentes junto a estes problemas, visamos trazer novas contribuições junto à área de pesquisas em robótica móvel autônoma.

1.2 Definição do Problema e Abordagens

Como resultado da evolução das pesquisas em robótica móvel, tem surgido interesse crescente na utilização de grupos de robôs para resolução de diversas tarefas. As mais comuns justificativas para o uso de sistemas com múltiplos indivíduos, de acordo com Cao *et al.* (1997), Arai *et al.* (2002), Farinelli *et al.* (2004) e Parker (2008) são: (i) A construção e uso de pequenos robôs ao contrário do uso de um grande robô pode tornar a solução mais barata, flexível, fácil, e tolerante a falhas. (ii) Possível solução de tarefas impossíveis de serem realizadas individualmente. (iii) A abordagem multiagente pode levar em consideração *insights* de problemas de ciências sociais (teorias econômicas, organizacionais) de qualquer ser vivo, não necessariamente humano. Todavia, de acordo com Crandall *et al.* (2003), a maioria das soluções atualmente propostas ainda não se apresentam mais fáceis, mais baratas, ou mesmo mais flexíveis. Os paradigmas e as abordagens utilizadas para controlar grupos de robôs serão apresentados nos Capítulos 2 e 3.

O controle individual de robôs é difícil de ser implementado com conjuntos de regras (como por exemplo, por meio de sistemas especialistas e pré-programados). A coordenação de sistemas coletivos também sofre desta característica. Algumas soluções usando conjuntos de regras para a formação de grupos apresentam resultados satisfatórios considerando focos específicos de operação, entretanto, é comum que careçam de capacidades de flexibilidade, escalabilidade e tolerância a falhas, assim como também carecem de uma melhor adequação para atuar em ambientes muito dinâmicos e pouco estruturados. Ainda, trabalhos que buscam completa autonomia na definição dos comportamentos, como as abordagens com base em comportamentos de insetos sociais (Seção 2.3) e a robótica evolucionária (Seção 2.4) supostamente suportam maiores graus de flexibilidade, escalabilidade e tolerância a falhas, contudo, apresentam pouco grau de otimização na resolução de atividades mais complexas. Os conceitos descritos neste parágrafo serão detalhados no Capítulo 2.

Normalmente sensores e atuadores são sujeitos a erros e interferências, assim o controle das ações de um robô deve sempre levar em conta a imprecisão dos sensores e motores envolvidos. Um sistema robusto deve permitir que, mesmo com sensores e atuadores imprecisos, o agente cumpra adequadamente com o seu objetivo. Técnicas de aprendizado de máquina (como Redes Neurais Artificiais) podem ser utilizadas para auxiliar no desenvolvimento de sistemas mais robustos. A técnica de Redes Neurais é muito utilizada no controle de navegação de sistemas reativos, como as apresentadas nos trabalhos encontrados na literatura (Dezfoulian *et al.*, 2013; Engedy e Horvath, 2009; Zou *et al.*, 2006) e em trabalhos desenvolvidos em nosso grupo (Pessin *et al.*, 2010a; Sales *et al.*, 2010; Souza *et al.*, 2012a). Conceitos e aplicações de Redes Neurais serão descritos no Capítulo 3.

Com base na bibliografia consultada, percebe-se que ainda existe um grande abismo entre a busca por soluções completamente autônomas e a utilização prática destas soluções, principalmente quando as atividades dos robôs considerarem ambientes altamente dinâmicos. Com o intuito de buscar uma solução para este problema, foi proposto em (Pessin, 2008) um sistema multirrobótico completamente autônomo para atuação em um desastre ambiental do tipo incêndio florestal. Tal atividade foi realizada com grupos de agentes para identificar e combater o incêndio. O controle de navegação dos robôs, incluindo o deslocamento em direção a uma posição alvo e o desvio de obstáculos, foi desenvolvido em um ambiente de simulação e conta com uma Rede Neural Artificial para o controle; a obtenção das posições de atuação (estratégia de grupo) foi realizada por um Algoritmo Genético. Este sistema será detalhado no Capítulo 6.

O uso de grupos de robôs depende que os mesmos apresentem características individuais funcionais. Considerando que algumas tarefas básicas individuais ainda não apresentam soluções globalmente consensuais, foram também foco de pesquisa nesta tese as características de navegação e localização. A investigação sobre navegação autônoma foi desenvolvida em trabalhos colaborativos com demais colegas do Laboratório de Robótica Móvel do ICMC/USP; será detalhada no Capítulo 5. A atividade de estudo, proposta e desenvolvimento de técnicas de localização robótica foi desenvolvida integrando interesses do LRM/USP (2013), do Laboratório de Sistemas Distribuídos e Programação Concorrente do ICMC/USP, do *Communication and Distributed Systems Group* (CDS) do *Institute of Computer Science and Applied Mathematics*, Universidade de Berna – Suíça e do *Robotics Lab*, da *School of Maths and Computer Science* (MACS), Universidade Heriot-Watt – Edimburgo, Reino Unido. A investigação sobre localização será detalhada no Capítulo 4

A pesquisa realizada em navegação e localização tem como foco o desenvolvimento de partes individuais em robôs reais que poderiam ser usados como parte da implementação real da pesquisa sobre grupos de robôs. Por restrições de tempo, a pesquisa com grupos de robôs

é apresentada em simulação, enquanto as pesquisas com navegação e localização fazem uso de robôs reais.

No que tange a pesquisa sobre coordenação de grupos de robôs, fizemos a escolha de trabalhar sobre uma aplicação modelada como o problema de combate a incêndios florestais. Incêndios são problemas graves e recorrentes em nossa sociedade, com soluções ainda bastante ineficientes até mesmo do ponto de vista da resolução por humanos. Os incêndios florestais causam diversos tipos de danos humanos, materiais e ambientais. Danos ambientais na fauna e flora afetam desde o solo até o aspecto de sobrevivência de áreas não incendiadas. Além das perdas humanas – como exemplos de prejuízos humanos e ambientais, na Grécia, um incêndio em 2007 matou 64 pessoas; na Austrália, incêndios entre 2008 e 2009 mataram mais de 170 pessoas. No Brasil, em 2008, um incêndio na reserva ecológica do Taim, queimou mais de 4500 hectares de vegetação preservada, e em 2010, um incêndio que atingiu o Parque Nacional das Emas, queimou cerca de 60% dos 131.800 hectares do parque (LIF/UFPR, 2013). Constantemente têm sido observados incêndios de grandes proporções na Rússia, Portugal, Grécia, Califórnia (EUA), e mesmo no Brasil, que foram amplamente divulgados nos meios jornalísticos.

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

Na tarefa de controle de incêndio, uma das questões mais importantes tem relação com a estratégia de posicionamento dos agentes de combate. De acordo com a capacidade de atuação de cada agente, as condições climáticas (vento, chuva), a topografia e a vegetação, diferen-

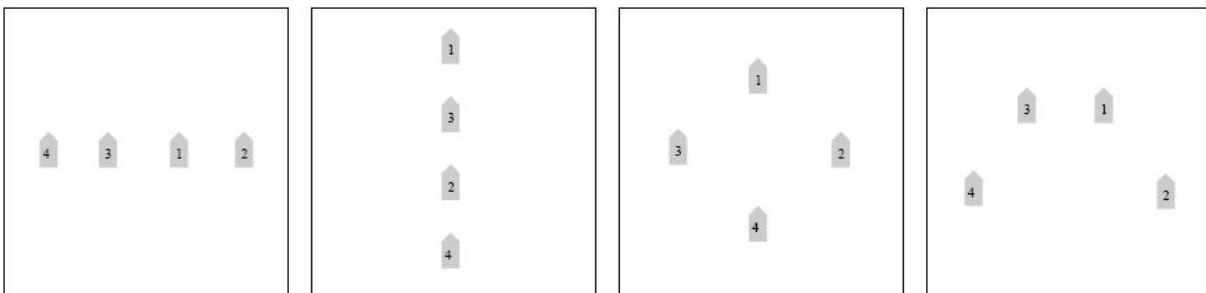


Figura 1.1: Exemplos de formações para equipes de quatro agentes. Da esquerda para a direita: linha, coluna, circular e ferradura (Balch e Arkin, 1998).

tes formações podem ser sugeridas. A Figura 1.1 apresenta quatro formações estruturadas de posicionamento. No caso de incêndios florestais, formações não estruturadas podem ser mais eficientes. Estas estratégias, quando sugeridas por especialistas, podem não levar em conta o grande número de variáveis envolvidas, assim, a definição do posicionamento poderia fazer uso de técnicas de otimização e Aprendizado de Máquina (*Machine Learning* - ML). Uma técnica de ML indicada para estes casos é a de Algoritmos Genéticos (*Genetic Algorithms* - GA) (Holland, 1975; Mitchell, 1998). Principalmente pelo fato de serem técnicas não supervisionadas, onde soluções podem surgir sem a necessidade da criação de uma base de dados para treinamento supervisionado.

O uso de simulação no desenvolvimento de sistemas robóticos é comumente incentivado por tornar o desenvolvimento e a validação mais ágil e segura (Osório *et al.*, 2010). Assim, consideramos para uma implementação física real que o sistema proposto seja projetado, desenvolvido e testado anteriormente em ambientes de simulação realísticos. A simulação de sistemas robóticos é especialmente interessante para robôs caros, grandes, ou frágeis (Go *et al.*, 2004). É uma ferramenta extremamente poderosa para agilizar o ciclo de desenvolvimento de sistemas robóticos eliminando desperdício de recursos, tanto financeiros como computacionais. Para que uma simulação seja útil, entretanto, ele deve capturar características importantes do mundo físico. Detalhes de aspectos de simulação e modelagem serão descritos na Seção 2.2. A adoção de ambientes com representação de terrenos 3D (com relevo, vegetação e obstáculos), bem como a simulação de robôs sujeitos à forças físicas (*e.g.* aceleração, atrito, colisões, gravidade), torna o sistema ainda mais realístico, tornando mais aceitável que as soluções obtidas em simulação sejam aplicadas em atividades reais.

Considerando os desafios e as características da área da robótica móvel autônoma descritos nos parágrafos anteriores, buscamos nesta Tese apresentar investigações que conduzissem a formas de resolução, utilizando Aprendizado de Máquina, para as questões de localização, navegação e coordenação de grupos. Os objetivos gerais e específicos de cada uma das questões estão detalhados na seção subsequente.

1.3 Objetivos

O objetivo geral deste trabalho tange a busca por sistemas de robótica móvel autônomos e inteligentes, para aplicação em ambientes dinâmicos. Como ponto inicial, a investigação sobre a atuação com grupos de robôs em atividades críticas foi o principal elemento motivador para o desenvolvimento de técnicas bio-inspiradas em simulação. Buscando o desenvolvimento de um sistema mais completo e que pudesse ter aplicação real, tornaram-se necessários o desenvolvi-

mento e a investigação em duas áreas base da robótica, necessárias para que o grupo de robôs tivesse atuação eficiente: localização e navegação. Sendo assim, o trabalho se desenvolve em três grandes áreas: localização, navegação e coordenação de grupos.

As pesquisas sobre localização e coordenação de grupos apresentam investigação e propostas originais, buscando estender o estado da arte e apresentam resultados inovadores. A parte sobre navegação tem como objetivo principal ser um elo entre os conceitos de localização e coordenação de grupos, sendo o foco o desenvolvimento de um veículo autônomo com maior implicação em avanços técnicas.

Na **investigação sobre formação de grupos** (Capítulo 6), é objetivo geral a busca de soluções de alto grau de autonomia para atuação de grupos de robôs em missões críticas, considerando ambientes dinâmicos. Focando em aspectos de autonomia e eficiência na resolução das tarefas e buscando avaliar e propor aspectos que suportem altos graus de escalabilidade, flexibilidade e tolerância a falhas. São objetivos específicos desta parte da Tese:

- Desenvolvimento e avaliação de uma arquitetura multirroboótica para combate de incêndios florestais por meio de métodos de otimização, considerando autonomia e eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis.
- Proposta de um simulador de incêndio florestal.
- Desenvolvimento e avaliação de um método de tolerância a falhas na operação multirroboótica.
- Investigações buscando otimizar o método de otimização por meio de avaliação de diversas características do algoritmo proposto.
- Avaliações com outros métodos de otimização, como Otimização por Enxame de Partículas, *Simulated Annealing* e *Hill Climbing*.
- Proposta e discussão de possíveis alternativas às abordagens utilizadas.

Na **investigação sobre navegação** (Capítulo 5), é objetivo geral o desenvolvimento técnico de um veículo autônomo de grande porte, funcional para ambientes dinâmicos. São objetivos específicos desta parte da Tese:

- Aperfeiçoamento de uma arquitetura para navegação autônoma, baseada em visão monocular e com capacidade de seguir pontos esparsos de GPS.

- Apresentação do estudo do *gap* entre simulação e realidade entre o modelo proposto por Pessin *et al.* (2010a) e os trabalhos com robôs reais do LRM/USP (2013).
- Avaliação de diferentes topologias de redes neurais buscando verificar as topologias que permitam o aprendizado com maior acuracidade.
- Discussão sobre as capacidades de aprendizado.
- Apresentação de algumas contribuições técnicas no desenvolvimento de controladores de sensores para o robô CaRINA I⁴.

Na **investigação sobre localização** (Capítulo 4), é objetivo geral a busca por um método de localização alternativo ao GPS (passível de ser usado em ambientes internos) que seja fácil, prático e barato de implementar. Os objetivos específicos desta parte da Tese são:

- Desenvolvimento e avaliação de uma arquitetura para localização de robôs móveis usando sinal de potência de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina (Redes Neurais Artificiais).
- Avaliação de diferentes topologias de redes neurais buscando verificar as topologias que permitam o aprendizado com maior acuracidade.
- Avaliação de métodos de evolução de redes neurais, considerando evolução de pesos e de topologias.
- Proposta de diferentes tipos de filtros, a serem aplicados sobre o sinal lido dos pontos de acesso, a fim de avaliar o comportamento geral do sistema bem como do aprendizado da RNA.
- Estudo e avaliação da rede neural considerando possíveis diferentes quantidades de entradas de sinal da rede sem fio, a fim de avaliar como o aprendizado da rede se comporta com mais ou menos pontos de acesso disponíveis.
- Estudo e proposta de métodos para evolução de pesos e topologias de RNAs, seguido da comparação e discussão dos resultados entre os métodos de evolução com métodos clássicos.
- Proposta de uma possível alternativa com o objetivo de tornar o sistema proposto mais autônomo e tolerante a falhas.

⁴O desenvolvimento do robô CaRINA I, que é um dos projetos vinculados ao Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos (INCT-SEC), será detalhado na Seção 5.1

As contribuições, tanto técnicas como científicas, serão descritas no início dos capítulos que apresentam as investigações realizadas.

1.4 Organização do Texto

Este documento está organizado em 7 capítulos, que abordam a motivação e os objetivos, a base conceitual, a metodologia, os trabalhos relacionados, os experimentos e os resultados, bem como as considerações finais. A estrutura e os conteúdos abordados neste documento, em cada um de seus capítulos, estão resumidos a seguir:

- O Capítulo 2 tem como objetivo descrever um panorama geral da robótica móvel, apresentando conceitos teóricos e aplicações de sistemas robóticos reais individuais e coletivos. Além disso, são descritos aspectos relacionados à simulação e modelagem de sistemas robóticos, com foco na busca de simulações cada vez mais realísticas. Por fim, descrevem-se duas abordagens relativamente novas que buscam aprimorar a autonomia em sistemas robóticos individuais e coletivos.
- O Capítulo 3 trata de Aprendizado de Máquina (*Machine Learning* - ML). Detalhes de funcionamento das técnicas relacionadas com este trabalho serão abordados. O foco do capítulo é nos paradigmas conexionista e evolutivo, que são os paradigmas mais fortemente relacionados com Robótica Autônoma e Robótica Evolucionária. Visando apresentar de forma mais objetiva e sintética este capítulo, deixamos para os capítulos de investigações (4, 5 e 6) alguns dos detalhes mais pontuais avaliados sobre cada técnica. Estes detalhes pontuais serão descritos nas seções estritamente relacionadas com sua importância, a fim de apresentar um texto com melhor fluxo para leitura.
- O Capítulo 4 está inserido no contexto da investigação do uso de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina para auxiliar na tarefa de localização interna de robôs móveis. O objetivo do capítulo é descrever as propostas e as avaliações de métodos de localização de agentes móveis usando a potência do sinal obtido de Pontos de Acesso (*Access Point* – AP). Inicialmente é discutida a motivação do método proposto, inserida dentro do estado da arte. Em seguida, apresentamos a metodologia proposta, a bateria de avaliações realizadas e discussão sobre as avaliações. Ainda, descrevemos uma possível alternativa com o objetivo de tornar o sistema proposto completamente autônomo e tolerante a falhas.
- O Capítulo 5 apresenta resultados de pesquisas em navegação autônoma, sendo basicamente construído a partir de trabalhos em colaboração. O capítulo se divide nas seguintes

seções: a primeira discute sucintamente as etapas iniciais em simulação, robôs de pequeno porte, e aspectos de desenvolvimento da autonomia do robô de grande porte CARINA I. A segunda parte apresenta a descrição e os resultados do modelo proposto de navegação autônoma. Em seguida, são apresentados sucintamente modelos de mapeamento e sua relação com o desenvolvimento de métodos mais seguros e de alta acuracidade de navegação autônoma.

- O Capítulo 6 está inserido no contexto da investigação do uso de técnicas de otimização para formação de estruturas para grupos de robôs atuarem em atividades críticas. O objetivo do capítulo é descrever a proposta, o desenvolvimento, a utilização e a avaliação de modelos que permitam graus de autonomia e eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis. O capítulo é dividido nas seguintes seções: inicialmente, é apresentada a descrição geral do sistema, detalhando a arquitetura proposta e os detalhes do simulador. Em seguida, apresentamos como as técnicas de otimização de (i) Algoritmos Genéticos, (ii) Otimização por Enxame de Partículas, (iii) *Hill Climbing* e (iv) *Simulated Annealing* podem ser aplicadas no problema. Ainda, apresentamos como uma heurística baseada em conceitos de algoritmos genéticos dinâmicos pode auxiliar na capacidade de tolerância a falhas do sistema. Finalizando, apresentamos possíveis alternativas às abordagens utilizadas.
- O Capítulo 7 encerra o documento, onde serão revisados os objetivos e conceitos que motivaram o desenvolvimento deste trabalho. Serão detalhadas as contribuições resultantes, finalizando com a descrição dos trabalhos futuros vislumbrados durante o desenvolvimento desta pesquisa.

Pontualmente, neste documento serão feitas as referências aos trabalhos publicados pelo autor na área específica. As principais referências relacionadas com esta pesquisa do autor encontram-se disponíveis para acesso na Internet⁵ e estão discriminadas no Anexo A. São elas relacionadas a:

- Ambientes de simulação e plataformas robóticas: (Ailinca *et al.*, 2012; Fernandes *et al.*, 2012; Osório *et al.*, 2010; Pessin *et al.*, 2007b).
- Formação de grupos de robôs: (Pessin e Osório, 2009a,b,c, 2010; Pessin *et al.*, 2010b,c,d, 2009; Vargas *et al.*, 2012).
- Atuação robótica em fenômenos ambientais: (Costa *et al.*, 2011, 2012; Pessin *et al.*, 2010a, 2007a).

⁵<http://sites.google.com/site/pessin>

- Navegação autônoma e mapeamento: (Hata *et al.*, 2009a,b; Pessin *et al.*, 2010a; Sales *et al.*, 2010; Souza *et al.*, 2012a, 2010, 2011a, 2012b, 2011b, 2013).
- Localização em ambientes internos usando redes sem fio: (Marinello *et al.*, 2012; Pessin *et al.*, 2012, 2011a,b).

Robótica Móvel

Este capítulo tem como objetivo descrever um panorama geral da robótica móvel, apresentando conceitos teóricos e aplicações de sistemas robóticos reais individuais e coletivos, assim como as abordagens e as características básicas de sistemas com múltiplos robôs. Além disso, são descritos aspectos relacionados a simulação e modelagem de sistemas robóticos, com foco na busca de simulações cada vez mais realísticas. Por fim, descrevem-se duas abordagens que buscam aprimorar a autonomia em sistemas robóticos individuais e coletivos.

2.1 Robôs Móveis Autônomos

De acordo com Bekey (2005), um robô pode ser considerado como uma máquina equipada com sensores, com habilidade de processamento que emula algum aspecto cognitivo e com atuadores. Segundo Dudek e Jenkin (2000), a robótica móvel estende a linha da robótica clássica por introduzir questões relacionadas ao controle de veículos autônomos ou semi-autônomos. A robótica móvel dá ênfase aos problemas relacionados a operação em ambientes complexos. O trabalho conjunto de especialistas das áreas de engenharia mecânica, de engenharia elétrica e de ciência da computação tem provido dispositivos de hardware e software capazes de permitir o controle dos robôs móveis de forma cada vez mais eficiente (Dudek e Jenkin, 2000).

Para operar de forma eficiente, os robôs devem possuir dispositivos sensores capazes de obter informação do ambiente, tornando possível obter ou estimar uma pose (posição e orientação), reconhecer obstáculos, e obter outras diversas características do ambiente (*e.g.* iluminação, som, temperatura). Além disso, também devem possuir dispositivos que permitam a atuação no ambiente, como motores lineares ou angulares, bem como pistões ou garras. Normalmente sensores e atuadores são sujeitos a erros e interferências, assim o controle das ações de um robô deve sempre levar em conta tais imprecisões. A capacidade de interpretação sensorial para geração de comandos para os atuadores deve sempre buscar capacidade de execução em tempo real.

Usualmente, a interação de um robô com o ambiente é realizada por meio de ciclos de percepção-ação, que consistem de três passos (Bekey, 2005; Siegwart *et al.*, 2011): (i) Obtenção de informação através de sensores; (ii) Processamento das informações para seleção da ação; (iii) Execução da ação através do acionamento dos atuadores. Este conjunto de operações, em uma análise superficial, pode parecer simples, porém o controle robusto de sistemas robóticos tem complicações físicas e mecânicas (*e.g.* cinemática e dinâmica), eletrônicas (*e.g.* falta de precisão de sensores) e computacionais que tornam a criação de um conjunto de regras de controle uma tarefa árdua e sujeita a erros. Um trabalho prévio do autor deste documento (Pessin, 2008) envolveu o levantamento bibliográfico e a revisão destes conteúdos. Entre outros assuntos, descreveu-se sobre sistemas individuais de controle robótico, sensores e atuadores, além de aspectos de morfologia e de simulação de sistemas multirrobóticos fisicamente realísticos, permitindo conhecer diversas classificações e características de sistemas robóticos. É de fundamental importância este conhecimento, a fim de propor *hardwares* que sejam apropriados ao tratamento eficiente dos problemas propostos. O trabalho com grupos de robôs depende que individualmente os robôs tenham capacidades bem desenvolvidas, podendo contar inclusive com robôs com capacidades distintas (sistema heterogêneo).

Alguns trabalhos que demonstram aplicações de relativo sucesso com o uso de robótica móvel (sistema individual) em operações hostis em ambientes dinâmicos e situações críticas podem ser citados, como o *Raposa* (IdMind, 2010) (Robô semiautônomo para operações de salvamento) que é projetado com a finalidade de auxílio em resgate. É um projeto português que envolve o *Regimento de Bombeiros* de Lisboa e o *Perceptual Robotics Lab* da Universidade do Sul da Flórida (EUA) na criação de um robô teleoperado para atuar em ambientes hostis à presença humana, tais como escombros resultantes de um terremoto ou atentado. Em termos sensoriais, o *Raposa* possui 17 sonares e um GPS. Ainda, o *SACI* (Sistema de Apoio ao Combate de Incidentes) (Macedo *et al.*, 2007) é um robô teleoperado projetado para atuar em combate de incêndios. Sua capacidade extintora envolve a possibilidade de gerar névoa ou de liberar um jato compacto de água ou espuma. O *SACI* não tem capacidade de carregar junto

de si a fonte de água ou da espuma, assim, mangueiras devem ser acopladas a sua estrutura. A última versão do robô conta com controle remoto sem fio e câmeras de vídeo.

Outras aplicações também relacionadas com aplicações em missões críticas podem ser citadas, como os protótipos militares *Boeing X-45A* (Boeing, 2013) (Figura 2.1(a)) que, sob supervisão humana em terra (sem nenhum piloto embarcado), tem sido testado para missões de combate e o *nEUROn* (Dassault, 2013) (Figura 2.1(b)), protótipo europeu com a mesma finalidade. Também podem ser citados robôs aquáticos como o *Depthx* (Wettergreen, 2013) (Figura 2.1(c)) que é fruto de um projeto para criação de um robô autônomo de mapeamento de solo subaquático. O *Isis* (Figura 2.1(d)) (NOC, 2013) é um veículo remotamente operado, fruto de um projeto inglês de exploração marítima.

Ainda na linha de robôs com possíveis aplicações em missões críticas, podemos incluir modelos terrestres que usam pernas, rodas (Figura 2.1(e)) ou esteiras (Figura 2.1(f)). Robôs com rodas provem a mais alta eficiência de mobilidade em terrenos planos (Siegwart *et al.*, 2011). Robôs com pernas possuem um número muito maior de graus de liberdade, o que tem impacto na maior complexidade de *hardware* para esse controle e conseqüentemente maior consumo de energia. Robôs com esteira, por possuírem uma maior área de contato com o chão, podem ser utilizados para atuar em terrenos sinuosos e superar obstáculos não viáveis para robôs com rodas (Dudek e Jenkin, 2000; Siegwart *et al.*, 2011). Entretanto, devido a mudança de orientação depender de deslizamento com uma grande área de contato, é difícil prever exatamente o centro de massa da rotação, visto a atuação em terreno irregular, assim, a confiabilidade da estimação odométrica é muito mais baixa que em modelos com rodas. Segundo Siegwart *et al.* (2011) a única forma de se obter localização e orientação com graus de confiança aceitáveis em um robô com esteira é usando um dispositivo externo para tal função.

Outros sistemas robóticos com aplicações em atividades cotidianas merecem ser citados, como a colheitadeira autônoma mostrada na Figura 2.2(d) e os robôs de campo que trabalham em ambientes completamente não estruturados, desconhecidos e geralmente perigosos. As principais atividades destes robôs são a exploração de ambientes hostis (e.g. espacial, cavernas, vulcões). A Figura 2.2(a) apresenta o robô *Mars Exploration Rover* (JPL/NASA, 2013), de exploração espacial e a Figura 2.2(b) apresenta o robô *Dante II*, de exploração de cavernas e vulcões. Além destes modelos, existem também modelos mais simples, como o robô de serviço doméstico (aspirador de pó autônomo) *Roomba* (iRobot, 2013) (Figura 2.2(c)). Os robôs citados podem ser considerados como exemplos promissores de operação de robôs individuais, servindo como inspiração para modelagem de *hardware* para outras aplicações.

Originalmente, controles reativos para navegação robótica baseados em regras foram apresentados ainda na década de 50. Walter (1950) construiu um robô móvel capaz de executar tarefas tais como desviar de obstáculos e seguir fontes luminosas. O robô móvel *SHAKY* de-

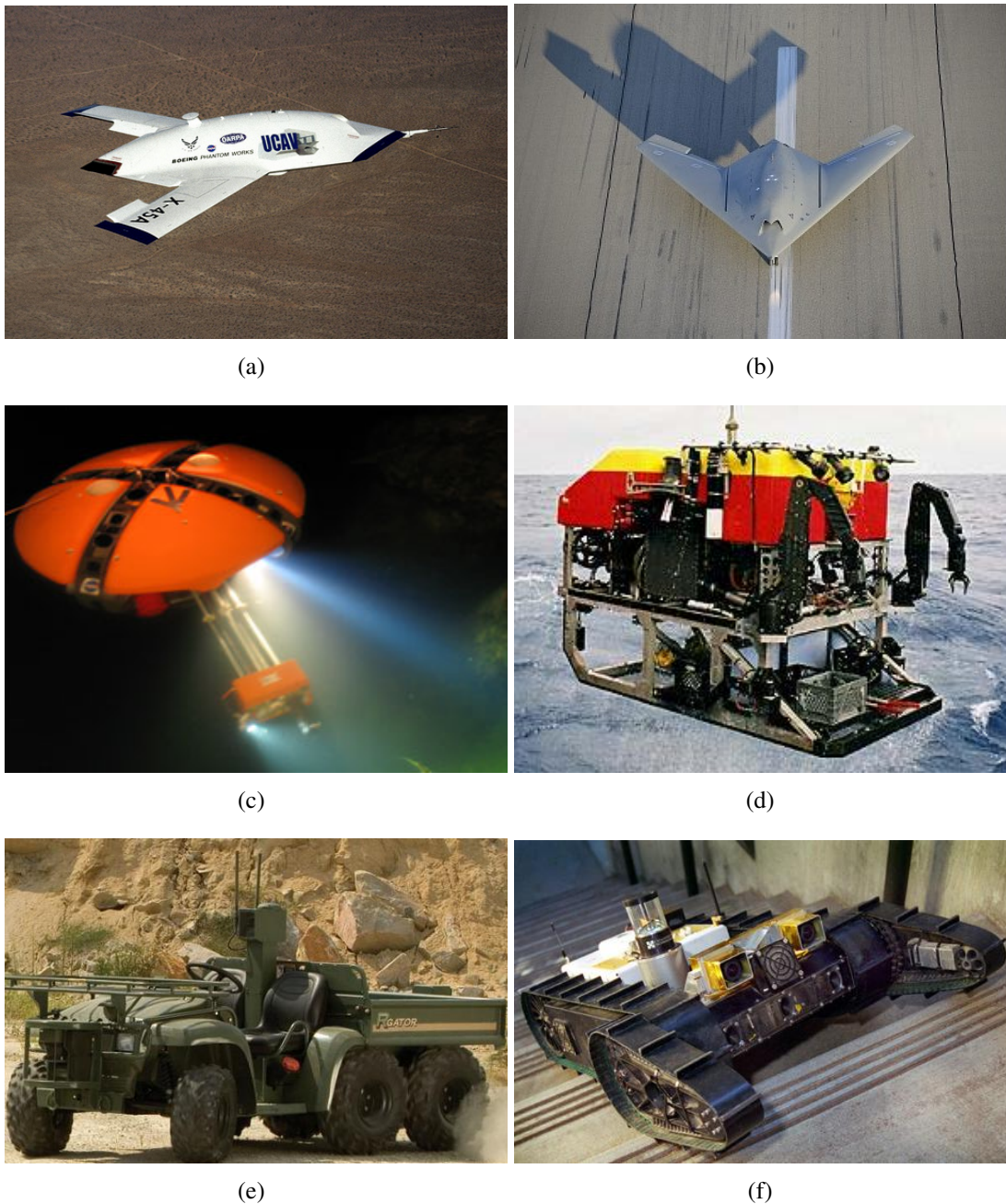


Figura 2.1: Robôs aéreos de combate: (a) *X-45A* (Boeing, 2013); (b) *nEUROn* (Dassault, 2013). Robôs aquáticos de exploração: (c) *Depthx* (Wettergreen, 2013); (d) *Isis* (NOC, 2013). Robôs terrestres: (e) *R-Gator* (Deere, 2013), um veículo que permite teleoperação, navegação automática entre pontos tipo “ensinar-repetir”, desvio de obstáculos e operação manual; (f) *Urbie* (Matthies, 2013), robô para fins militares de reconhecimento de terrenos urbanos.



Figura 2.2: Robôs de campo: (a) *Mars Exploration Rover* (JPL/NASA, 2013) para exploração espacial; (b) *Dante II* (Bares e Whittaker, 2013) para exploração de cavernas e vulcões. Robôs de serviço: (c) Aspirador de pó *Roomba* (iRobot, 2013); (d) Colheitadora autônoma (NREC/CMU, 2013).

envolvido por Nilsson (1969) em Stanford, utilizava dois motores de passo, usando cinemática diferencial para se locomover; era equipado com sensores de distância, câmeras de vídeo e sensores táteis. Conectado a dois computadores, utilizava programas para percepção, modelagem e atuação no ambiente. As tarefas desempenhadas pelo robô incluíam desviar de obstáculos e a movimentação de blocos coloridos. Este robô tinha grandes dificuldades em processar e interpretar as informações sensoriais obtidas do ambiente, sendo que nunca foi capaz de completar uma sequência completa de ações em um ambiente real.

Na metade da década de 1990 foi proposto o *ALVINN* (*Autonomous land vehicle in a neural network*) (Baluja, 1996; Jochem *et al.*, 1993, 1995; Pomerleau, 1994), um programa baseado em Redes Neurais Artificiais que, a partir de um prévio treinamento supervisionado, era capaz de

guiar um veículo em uma pista, usando o sistema de visão computacional do *NavLab* (NL/CMU, 2013).

Nos anos de 2004 e 2005, o *DARPA Grand Challenge* (Buehler *et al.*, 2007), financiado pelo *Defense Advanced Research Projects Agency*, a principal organização de pesquisa do departamento de defesa dos Estados Unidos, organizou uma competição com o objetivo de construir veículos autônomos para navegar grandes distâncias em estradas. O objetivo principal foi a pesquisa para o desenvolvimento de tecnologias para a criação do primeiro veículo completamente autônomo capaz de completar um longo trecho em estradas de terra em um tempo limitado. Em 2007 foi alterado o foco da competição, renomeada para *DARPA Urban Challenge* (Buehler *et al.*, 2009), tendo como novos objetivos a criação de um veículo capaz de se autodirigir no tráfego urbano, e de realizar tarefas como ultrapassagem, estacionamento e negociação em interseções. Percebem-se as tendências a estudos de cooperação e de interação múltipla. Os trabalhos (Buehler *et al.*, 2007) e (Buehler *et al.*, 2009) contém coletâneas descrevendo os times e diversos aspectos da construção dos veículos participantes do evento.

No ano de 2009, teve início o INCT-SEC, sendo o ICMC/USP um dos institutos participantes, e onde um dos projetos associados era o desenvolvimento de um veículo autônomo de grande porte. Este veículo será apresentado no Capítulo 5 e é uma importante iniciativa nacional no que se refere a veículos para navegação autônoma, bem como para auxílio a motoristas.

O *Fire-Fighting Home Robot Contest* (Trinity, 2010) e o *Concurso Robô Bombeiro* (Guarda, 2010) são competições robóticas que põem à prova pequenos robôs móveis autônomos que têm como missão encontrar e apagar uma vela, como um incêndio simulado, num modelo de aproximadamente 2 m² representando um casa formada por corredores e quartos. Esta tarefa simula a operação real de um robô autônomo realizando proteção contra incêndios em residências. Em ambas as competições, os robôs devem ser construídos pelos competidores; vence o robô que apagar a vela no menor espaço de tempo. Os trabalhos (Miller *et al.*, 2003; Pack *et al.*, 2004) apresentam o desenvolvimento do hardware de robôs designados a participar de competições como as descritas acima. O *Mini Grand Challenge* (PennState, 2010) apresenta uma competição similar a proposta no *DARPA Grand Challenge* (Buehler *et al.*, 2007), porém com o uso de robôs de pequeno e médio porte.

Como dito anteriormente, um grupo de robôs deve ser formado por robôs capazes de atuar com certos graus de eficiência em tarefas, assim, é de vital importância o conhecimento em controle robótico, e seu contínuo aperfeiçoamento. Alguns trabalhos que apresentam fundamentos interessantes sobre robótica individual podem ser vistos em (Heinen *et al.*, 2006; Hoffmann *et al.*, 2004; Pessin *et al.*, 2010a; Souza *et al.*, 2013). Em (Heinen *et al.*, 2006) é apresentada uma Rede Neural Artificial para fazer controle de estacionamento de um robô não holonômico. Em (Pessin *et al.*, 2010a) é utilizada uma Rede Neural para realizar navegação. Ambos os trabalhos

foram realizados em simuladores com propriedades físicas. Já, os trabalhos (Souza *et al.*, 2013) e (Hoffmann *et al.*, 2004) apresentam modelos de aprendizado de máquina relacionados com visão computacional para realizar navegação em robôs reais.

Constata-se, a partir dos trabalhos citados, que o controle de navegação em robôs móveis autônomos não é uma tarefa simples, onde muitas pesquisas foram direcionadas no sentido de adotar técnicas de Aprendizado de Máquina, notadamente através do uso de Redes Neurais Artificiais para o controle de robôs móveis autônomos. Estes trabalhos demonstram que a aplicação de robótica móvel é um tópico ativo e importante de pesquisa e desenvolvimento. As diversas competições nesta área, com resultados nem sempre satisfatórios, também demonstram que ainda não existe uma solução definitiva ou mais adequada ao problema de controle robótico individual, e que este é um tema de pesquisa em aberto.

Assim como o controle individual de robôs é difícil de ser implementado com conjuntos de regras, a coordenação de sistemas coletivos também sofre desta característica. Entretanto, algumas soluções interessantes têm sido apresentadas. Osagie (2006) apresenta e compara dois tipos de algoritmos de controle para manter formações em grupos de robôs caiaques. Um dos controles é baseado em comportamentos (*behavior-based control*) (Balch e Arkin, 1998; Brooks, 1986), onde os robôs possuem conjuntos de comportamentos independentes (como manter formação, explorar o ambiente, construir mapa – isoladamente, estas características podem ser construídas com técnicas de aprendizado de máquina) e um sistema coordenador num nível mais alto hierarquicamente que escolhe qual comportamento ativar em cada instante de tempo. O outro controle usa um conjunto definido de regras (*rule-based approach*) para cálculo de posições e decisões de navegação, neste modelo, todas as ações dos robôs são ativadas por meio de regras. Osagie (2006) ressalta que ambos os modelos são muito dependentes da qualidade do desenvolvedor na definição das regras e dos comportamentos. Os resultados, inicialmente avaliados em simulação e após avaliados em robôs reais apresentaram que o modelo baseado em comportamentos resultou em um sistema relativamente instável, porém, com melhor resposta a mudanças dinâmicas no ambiente e com capacidade de escalabilidade bem superior ao modelo usando o conjunto de regras.

Em um trabalho de referência em termos de mapeamento e exploração, Fox *et al.* (2006) apresenta uma abordagem para exploração e mapeamento usando grupos de robôs. O sistema permite que times de robôs explorem o ambiente a partir de posições diferentes e desconhecidas. O trabalho trata diretamente de coordenação, integração de informação e aspectos de comunicação limitada. A garantia da consistência na criação compartilhada dos mapas é obtida por meio de encontros físicos entre os robôs. Encontros físicos que não forem completados fazem com que as informações dos robôs sejam descartadas e o processo de mapeamento reiniciado naqueles robôs. O mapa compartilhado é usado para definir estratégias de exploração.

Tais mapas são criados com informações de laser e de odometria. Um robô líder é o responsável pela coordenação das estratégias de exploração das áreas ainda não mapeadas. A perda do robô líder faz com que o um próximo robô assuma o seu papel. Este controle de quem é o líder é feito por meio do ID dos robôs. A perda da comunicação entre os robôs faz com que se ative o modo de exploração individual, onde o próprio robô, com o seu mapa, escolhe a estratégia de exploração. O mapeamento usa a técnica de mapeamento e localização simultânea (*Simultaneous localization and mapping* - SLAM) (Leonard e Durrant-Whyte, 1991). Nos mapas compartilhados, a localização dos robôs é estimada usando filtro de partículas. Experiências com robôs reais foram realizadas, e, em inúmeras rodadas os mapas construídos ficaram muito similares. A proposta de usar encontros físicos garante uma confiabilidade excelente para o sistema.

2.1.1 Abordagens para sistemas com múltiplos robôs

Em uma definição de grupos de tarefas, bastante abrangente, feita por Dudek *et al.* (1996), quatro grandes grupos foram identificados: (i) Tarefas tradicionalmente multiagente (por exemplo, o sistema de transporte público); (ii) Tarefas que obrigatoriamente requerem múltiplos agentes (por exemplo, tarefas que envolvam sincronismo de tempo em espaços físicos diferentes); (iii) Tarefas que podem se beneficiar do uso de múltiplos agentes, como mapeamento ou exploração; (iv) tarefas inerentemente individuais.

Algumas das tarefas podem claramente se beneficiar muito do uso de sistemas com múltiplos indivíduos. Outras tarefas só podem ser realizadas com o uso de sistemas com múltiplos indivíduos. Ainda, algumas tarefas não necessitam, e, de certa forma, podem ter seu desempenho diminuído se realizadas por mais de um agente.

Dentre as tarefas que podem se beneficiar, as mais comuns justificativas para o uso de sistemas multiagentes, de acordo com Cao *et al.* (1997), Arai *et al.* (2002) e Farinelli *et al.* (2004) são: (i) a construção e uso de pequenos robôs ao contrário do uso de um grande robô pode tornar a solução mais barata, flexível, fácil, e tolerante a falhas; (ii) possível solução de tarefas impossíveis de serem realizadas individualmente; (iii) a abordagem multiagente pode levar em consideração *insights* de problemas de ciências sociais (teorias econômicas, organizacionais) de qualquer ser vivo, não necessariamente humano.

Diversos tipos de tarefas, de acordo com Cao *et al.* (1997), Arai *et al.* (2002) e Bayindir e Sahin (2007) podem ser consideradas eixos de pesquisas em sistemas com múltiplos agentes, como: (i) Controle de tráfego, que envolve diretamente solução de conflitos (regras de trânsito, evitar colisão, prioridades, decisões de trajetórias); (ii) Transporte e manipulação de objetos; (iii) Construção de objetos, como construção civil; (iv) Coordenação de movimentos (manutenção de pose, agregação, formação); (v) Localização, mapeamento e exploração; (vi)

Reconfiguração de robôs (o que não trata exatamente de resolução de uma tarefa em si, mas sim da criação/alteração da morfologia).

Assim como existem diversos grupos de tarefas, diversos paradigmas computacionais foram desenvolvidos para propor soluções usando diferentes abordagens para os diversos tipos de tarefas. Cada paradigma abstrai o problema de uma forma diferente, o que permite a modelagem do sistema de acordo com estratégias próprias. Entretanto, nem sempre os paradigmas são propícios para todos os tipos de interação (Parker, 2008).

Nolfi e Floreano (2000), dividem em três grupos as abordagens para desenvolvimento de trabalhos em grupos de robôs. Estas abordagens têm perspectivas na engenharia, na etologia¹ e na biologia. Nesta linha de raciocínio, Nolfi e Floreano (2000) descrevem que a perspectiva na engenharia tem como princípio fundamental a divisão do sistema em partes tratáveis com conjuntos de regras. A perspectiva etológica tem como princípio fundamental a tentativa de modelar sistemas robóticos com base em comportamento social de animais simples. A perspectiva biológica, por sua vez, é relacionada com a teoria da evolução natural, e tem como princípio fundamental a tentativa de modelar sistemas que se aperfeiçoem com base na evolução.

O trabalho de Parker (2008) também trata das abordagens para desenvolvimento de trabalhos em grupos de robôs. Sendo a divisão, de certa forma, uma reestruturação das ideias apresentadas por Nolfi e Floreano (2000). De acordo com Parker (2008), as abordagens comumente utilizadas são baseadas em (i) paradigmas sociais e organizacionais, (ii) paradigmas baseados em conhecimento e ontologia e (iii) paradigmas bioinspirados.

Para Parker (2008), paradigmas sociais e organizacionais são tipicamente baseados em teorias organizacionais derivadas de sistemas humanos. O conhecimento dos campos de sociologia, economia ou psicologia tem apresentado formas eficientes para deduzir e modelar comportamentos em sistemas multirrobóticos. Nessas abordagens, a interação entre agentes/robôs é feita pelo desenvolvimento das dinâmicas individuais e do grupo como parte de uma organização; a redução dos requerimentos de comunicação se dá pelo uso de modelos já utilizados nestes campos. O uso deste paradigma aplicado em sistemas multirrobóticos se dá com a aplicação de papéis. Duas técnicas utilizadas neste paradigma são o *Market Based Approach* (Dias *et al.*, 2006) e o *Teamwork Models* (Pynadath e Tambe, 2002). Os papéis são seguidamente definidos com a finalidade de dividir a aplicação em porções gerenciáveis que podem ser atribuídas a diferentes robôs de uma equipe. O paradigma baseado em conhecimento e ontologia é citado por Parker (2008) como dominante na área de sistemas multiagentes, mas pouco explorado em sistemas multirrobóticos. Existe uma sobreposição bastante grande neste paradigma com o paradigma descrito anteriormente. O paradigma bioinspirado descrito por Parker (2008)

¹Parte da ecologia que trata dos hábitos dos animais e da acomodação dos seres vivos às condições do ambiente (Michaelis – Moderno Dicionário).

é o conjunto das abordagens etológicas e biológicas citadas por Nolfi e Floreano (2000). Estas abordagens usam técnicas como Algoritmos Genéticos, Algoritmos de Otimização por Enxame de Partículas e Redes Neurais Artificiais, detalhadas no Capítulo 3. Além destas, também são utilizadas as técnicas de Enxames Robóticos e de Robótica Evolucionária, detalhadas, respectivamente, nas Seções 2.3 e 2.4.

O que podemos ver com estas descrições de abordagens é que existe falta de um consenso global nestas caracterizações. É muito comum que autores de determinadas áreas ressaltem e detalhem mais as áreas relacionadas com sua abordagem de preferência. Definições complementares podem ser vistas em (Arai *et al.*, 2002; Dudek *et al.*, 1996; Farinelli *et al.*, 2006; Huhns e Stephens, 2000; Iocchi *et al.*, 2003; Wooldridge, 2002).

2.1.2 Características básicas de sistemas com múltiplos robôs

Cada um dos paradigmas ou abordagens descritas anteriormente são formas de pensar soluções, porém, com maiores ou menores graus, todos precisam levar em conta uma série de características básicas de sistemas multiagentes, independente do tipo de aplicação para a qual os sistemas múltiagentes são construídos. Em geral, são estas características que permitem classificar os sistemas de uma forma mais nítida.

Uma caracterização muito aceita na área foi descrita por Dudek *et al.* (1996), onde define como primárias as seguintes características de sistemas multirrobóticos:

- Tamanho do grupo: pode ser individual, em pares, limitado a um n fixo ou considerado infinito.
- Topologia de comunicação: pode ser por *broadcast*, por endereço (ponto a ponto), por hierarquia ou por grafo.
- Alcance da comunicação: pode ser sem comunicação, com comunicação de curto alcance ou com comunicação de longo alcance.
- Banda de comunicação: tem relação com o custo da comunicação, pode ter custo inexpressivo (como o envio de mensagens usando rede sem fio) ou ter custo elevado, relacionado com a movimentação do indivíduo – similar a comunicação realizada pelas abelhas, por meio de movimentos corporais.
- Tipos de reconfiguração: pode ter arranjo estático, rearranjo coordenado ou rearranjo dinâmico.

- Habilidade de processamento: pode ser baseado em máquina de estado ou equivalente a máquina de Turing.
- Composição do grupo: pode ser homogêneo ou heterogêneo.

Os trabalhos de Smith e Davis (1981), Huhns e Stephens (2000) e Farinelli *et al.* (2004) ressaltam que diversas características inerentes aos sistemas podem ser separadas quanto a capacidade de auto-gerência ou não. Os sistemas podem apresentar comportamento explícito (pré-programado) ou emergente nas seguintes questões:

- Qual a origem da cooperação?
- Como são tratadas as resoluções de conflito?
- Como é feita a decomposição das tarefas?
- Como é o aprendizado dos parâmetros de controle?

Cada abordagem citada, ainda, tem definições com maiores ou menores graus de tendência nas questões que envolvem conceitos de cooperação, colaboração, coordenação e suas sub-questões, como negociação e competição. Estas definições podem ser vistas em (Farinelli *et al.*, 2006; Garcia e Sichman, 2003; Huhns e Stephens, 2000; Parker, 2008; Wooldridge, 2002). Para cada uma das características e questões citadas nos parágrafos anteriores, existem diversas abordagens de tratamento. Algumas com finalidades em deixar os sistemas mais usáveis do ponto de vista da realidade prática. Outros com finalidades mais científicas, onde o foco não é a utilidade do sistema em si, mas sim a exploração de conceitos de como as questões citadas podem ser resolvidas sem a estipulação de regras pré-programadas, buscando a completa autonomia na operação do sistema robótico.

Pode-se perceber que a aplicação de técnicas de Inteligência Artificial para explorar a formação autônoma de grupos robóticos ainda é uma área carente de soluções realmente eficientes. Apesar de existirem algumas técnicas sugeridas para esta tarefa, parece não existir técnicas sugeridas como *ideais* para o trabalho com grupos robóticos. A pesquisa e experimentação de diversas técnicas, foco deste trabalho, propõem soluções práticas e permite que sejam criados e testados novos métodos para formação e atuação autônoma de grupos para atuar em diversas tarefas extremamente importantes para nossa sociedade, evitando a atuação de seres humanos em tarefas perigosas ou onde a atuação humana é ineficiente. Além disso, a aplicação de robótica em tarefas importantes do ponto de vista social pode permitir que se vislumbre-se também o desenvolvimento de novos tipos de dispositivos robóticos individuais.

2.2 Simulação e Modelagem

É praticamente de consenso geral que o desenvolvimento e a validação inicial em simuladores torna a pesquisa em robótica muito mais ágil e segura. A segurança, neste aspecto, tem relação tanto com a integridade física do robô como a dos pesquisadores. A agilidade tem relação com aspectos que podem ser simplificados na simulação, como a reposição das posições iniciais dos robôs e dos objetos, de forma extremamente simples, a não preocupação com alimentação (carga de bateria) e a possibilidade de trabalhar com variáveis de tempo de forma relativa.

Sistemas robóticos possuem diversas aplicações, diversas formas de controle e diversas formas de interação com o ambiente. Dada esta diversidade, de acordo com Jakobi *et al.* (1995) e Go *et al.* (2004), os simuladores que dão suporte a essas operações devem levar em conta apenas as características que impactam na estreita relação entre os mundos real e simulado, o que chamam de *conjunto de características básicas*. A consideração apenas destes aspectos importantes na relação mundo físico *versus* mundo simulado ajuda a evitar ainda mais desperdícios de tempo e de recursos computacionais.

A popularização de ambientes de simulação robótica, com ênfase em propriedades físicas, tem permitido o desenvolvimento de aplicações simuladas que se comportam relativamente bem nos robôs reais. Porém, é comum que existam pequenas adaptações na implementação de um sistema real anteriormente realizado em simulação. Por exemplo, o trabalho apresentado em (Pessin, 2008) possuía, entre outras coisas, um conjunto de regras para navegação seguindo pontos de GPS (localização) e utilizando uma bússola magnética (orientação). Este conjunto de regras pôde ser implementado em um robô real com pequenas adaptações, principalmente relacionadas com as sutis diferenças entre os modelos de atuadores e sensores desenvolvidos na simulação em relação ao modelo real. Considerando que a morfologia dos robôs não era exatamente a mesma, o maior impacto foi percebido nos valores aplicados nos atuadores (motores linear e angular). Entretanto, o núcleo lógico do programa si (as regras de comportamento) não foi alterado, e desta forma, demonstrou a validade da implementação em simulação como uma forma adequada para o projeto, teste e validação prévia de sua implementação em um sistema real, onde o referido experimento real será abordado no Capítulo 5.

Pessin (2008) descreve diversos ambientes de simulação bem como bibliotecas para desenvolvimento de ambientes. Dentre as bibliotecas citadas, encontram-se a biblioteca de simulação física *Open Dynamics Engine* – ODE (Smith, 2006)), a biblioteca de computação gráfica (visualização) *Open Scene Graph* – OSG (OSG, 2010) e o *Demeter* (Fowler, 2010)) para renderização de terrenos 3D. Entre os motivos da escolha por bibliotecas, ressalta-se a liberdade de implementar características ambientais não usuais, como a propagação de incêndio – tal desenvol-

vimento necessita de variáveis para diversas características ambientais, como terreno, vento e vegetação. Considerações semelhantes, sobre características ambientais, também motivaram o uso ou a criação de simuladores específicos nos trabalhos realizados por Osagie (2006), Werfel *et al.* (2007) e Fritsch *et al.* (2007).

Como citado anteriormente, algumas aplicações podem ser construídas em simuladores que apresentam menor acuracidade. Porém, algumas áreas da robótica precisam de simuladores extremamente precisos e confiáveis do ponto de vista do comportamento real dos sensores e atuadores dos robôs. Uma das áreas que mais sofre com essa necessidade é a área da robótica evolucionária (detalhada na Seção 2.4). O princípio básico da robótica evolucionária é a evolução de pesos de redes neurais usando algoritmos evolutivos. Sutis diferenças entre os modelos de simulação e os modelos reais têm efeitos que possivelmente inviabilizam a utilidade prática dos resultados das simulações. Este fato motivou a proposta de métodos para desenvolvimento de simuladores mais precisos, a fim de tornar a robótica evolucionária viável tanto do ponto de vista de simulação como do ponto de vista da robótica real, como apresentam os trabalhos (Brooks, 1992; Husbands e Harvey, 1992; Jakobi, 1993; Miglino *et al.*, 1996; Nolfi e Floreano, 2000).

De acordo com os trabalhos supracitados, o desenvolvimento de modelos confiáveis deve levar em conta as seguintes questões:

- Sensores e atuadores, aparentemente idênticos, podem ter comportamentos sutilmente diferentes. Desta forma, sugere-se a amostragem individual de cada sensor e atuador a fim de mapear suas características particulares. Miglino *et al.* (1996) apresenta a amostragem de oito sensores de distância do tipo infra-vermelho de um robô real. Tal amostragem permitiu a construção de modelos específicos para cada um dos oito sensores, apresentando que as diferenças são sutis porém impactam diretamente nos algoritmos de aprendizado de máquina utilizados. Dadas as diferenças entre os sensores, Miglino *et al.* (1996) sugere que os parâmetros obtidos pela técnica de aprendizado de máquina não seriam transferíveis para outros robôs não fosse o pré mapeamento dos dados.
- Sensores não retornam valores extremamente precisos, mas sim aproximações nebulosas, e retornam valores diferentes (incertezas ou ruídos) nas mesmas leituras, assim como atuadores têm efeitos incertos. Por exemplo, um mesmo motor pode ter variações mínimas de tempo para os mesmos comandos, que impactam na movimentação do robô. De acordo com Jakobi (1993), a inclusão de ruído completamente aleatório deve ser evitada, o correto deve ser a realização do mapeamento de cada dispositivo em busca de seus percentuais de ruído e incerteza, e então a aplicação destes ruídos em proporção similar.

- A morfologia e as características do ambiente devem ser as mais similares possíveis. Como ressaltam Brooks (1992) e Jakobi (1993), mundos tratados como *grids* bidimensionais são considerados inexpressivos para o desenvolvimento de simulações que possam ter um paralelo com execuções em sistemas reais.

O tratamento destes aspectos é usualmente muito trabalhoso. Em um *survey* sobre ambientes de simulação robótico apresentado por Kramer e Scheutz (2007), percebe-se a falta de facilidades no tratamento dos itens citados anteriormente. Ferramentas e métodos para prover recursos para os dois primeiros aspectos são bastante superficiais. O item três é o que mais tem tido acréscimo de recursos.

Por meio dos desenvolvimentos realizados nesta tese, pode-se perceber que a simulação ajudou em diversas partes iniciais, como apresentados nos Capítulos 5 e 6. O *reality gap* se mostrou bastante pequeno no desenvolvimento apresentado no Capítulo 5, provavelmente pelo fato do mesmo ter sido utilizado em simulação para atividades mais simples. Entretanto, propostas avaliadas em simulação para o sistema de localização (Capítulo 4) não se mostraram adequadas para serem implementadas em um sistema real devido a diferença entre a simulação do ruído e a realidade do ruído dos pontos de acesso utilizados. Contudo, a simulação teve um papel extremamente importante no desenvolvimento tanto de um modelo de propagação de incêndio quanto do comportamento de robôs para atuar no combate. Como veremos no Capítulo 6, a implementação real desta etapa de combate de incêndios com múltiplos robôs reais é um dos trabalhos futuros.

2.3 Enxame Robótico

Enxame Robótico (*Swarm Robotics*) pode ser definido como o estudo das formas pelas quais o comportamento inteligente coletivo pode emergir apenas de interações locais de um grande número de robôs relativamente simples (Bayindir e Sahin, 2007; Dorigo *et al.*, 2004). A maior parte das abordagens de enxames de robôs tem como inspiração o comportamento de insetos sociais, como formigas, cupins, abelhas e vespas. O comportamento social dos animais citados apresentam exemplos considerados fascinantes de como sistemas coletivamente inteligentes podem ser gerados a partir de um grande número de indivíduos simples (Dorigo e Sahin, 2004; Sahin e Winfield, 2008). Apesar da aparente falta de um modelo global de comunicação, os insetos sociais demonstram capacidade de coordenar atividades para a execução de tarefas além da simples capacidade individual (Bonabeau *et al.*, 1999, 2000). Cupins constroem grandes e complexos cupinzeiros e formigas apresentam impressionantes comportamentos de exploração, ataque e transporte de carga, como apresentam as Figuras 2.3 e 2.4.

A compreensão dos mecanismos de funcionamento dos sistemas de insetos sociais, como a robustez a perda de indivíduos (tolerância a falhas), a execução de tarefas com números diferentes de indivíduos (escalabilidade), e a capacidade de adaptação para resolução de diferentes tarefas em diversos ambientes (flexibilidade) têm permitido intuir sobre como construir sistemas robóticos com as mesmas características. A área de enxames de robôs é comumente trabalhada de forma conjunta entre biólogos, cientistas da computação e especialistas em robótica (Bona-beau *et al.*, 1999; Dorigo e Sahin, 2004).



Figura 2.3: Formigas trabalhando cooperativamente na criação de uma ponte, permitindo a passagem de outras formigas carregando comida. Crédito: (Metcalf, 2010).

As pesquisas utilizando enxames de robôs têm, em geral, quatro princípios básicos que a destacam das outras áreas de trabalho com grupos de robôs. Estes princípios são: (i) Estudo da coordenação de grandes grupos de robôs, pelo menos maior que dezenas, tendendo a centenas ou milhares. Tornando inatas as características de escalabilidade e robustez; (ii) O sistema robótico deve consistir de poucos grupos relativamente homogêneos, sendo que o número de robôs em cada grupo deve ser grande – estudos considerando grupos muito heterogêneos não se encaixam na forma canônica de enxame; (iii) Os robôs devem apresentar capacidades de *hardware* e *software* limitadas, ou ao menos demonstrar dificuldade na realização da tarefa de forma individual; (iv) Os robôs utilizados no estudo devem possuir apenas informação local, com limitações sensoriais e de comunicação – o uso de um sistema de coordenação global



Figura 2.4: Exemplos de trabalho cooperativo. (a) Transporte de comida. As formigas sobre a folha estão em postura defensiva contra moscas parasitárias. (b) Ataque a uma formiga inimiga. Créditos: (Wild, 2010).

possivelmente resulta em problemas de escalabilidade, dada a consideração feita sobre a grande quantidade de robôs.

Bonabeau e Meyer (2001) ressaltam que, possivelmente, o maior *insight* obtido pelo estudo dos animais sociais é que o comportamento coletivo complexo pode emergir de indivíduos seguindo regras simples. Os próximos parágrafos descrevem duas aplicações interessantes de trabalhos canônicos em enxames de robôs.

Fritsch *et al.* (2007) apresenta um *swarm* homogêneo com a finalidade de atuar em derramamento marinho de óleo. Os robôs apresentam quantidades limitadas de energia e de capacidade de recolher óleo, assim, quando qualquer um destes itens encontra-se em nível crítico, o robô retorna a um navio base para recarga de energia ou para esvaziar o reservatório de óleo. A atuação do *swarm* é simulada em um ambiente bidimensional. O trabalho é parte do projeto EU-MOP (Elimination Units for Marine Oil Pollutions). Os autores buscaram avaliar como três diferentes formas de interação/capacidade sensorial entre os robôs impactam na eficiência do sistema. As formas de interação/capacidade sensorial avaliadas foram: (i) Interação por meio de contato físico e utilizando sensores de colisão - os robôs se movem de forma aleatória em uma área georreferenciada; (ii) Interação por meio de contato físico, utilizando sensores de colisão e utilizando sensores de densidade de óleo para decisões de direcionamento - os robôs se movem dentro da área georreferenciada de acordo com a densidade de óleo na água; (iii) Interação por meio de contato físico e comunicação explícita - os robôs em áreas de baixa densidade de óleo trocam informações com os outros robôs sobre a densidade do óleo, movimentando-se para as áreas mais densas. Diferentes características de *hardware* foram consideradas para cada caso, bem como um conjunto específico de regras. Na simulação, fica claro o ganho de eficiência (diminuição no tempo de recolhimento do óleo) com o uso de uma maior quantidade de robôs.

Foram rodadas simulações considerando 1, 2, 5 e 20 unidades. Nas simulações realizadas, os níveis de interação (ii) e (iii) apresentaram resultados bastante superiores ao nível (i), porém, devido aos conjuntos de regras desenvolvidas, os tipos de interação (ii) e (iii) não apresentam grande ganhos entre si. Sendo, inclusive, o tipo (ii) superior para grupos menores de robôs.

Werfel *et al.* (2007) apresenta um *swarm* com robôs simulados em um ambiente bidimensional que aborda a tarefa de construção civil. O *swarm* tem capacidade para construir estruturas com as seguintes características: (i) Respeitando exatamente formatos pré-especificados; (ii) Adaptando o formato pré-especificado devido a obstáculos no ambiente; (iii) Formando uma barreira de espessura especificada em torno de um item do ambiente. Para cada tipo de construção, existe um conjunto específico de regras. As regras tem como base a ocupação de células vizinhas, pelas quais os robôs determinam onde anexar os blocos. Os autores abstraem o conceito de blocos construtores como se fossem peças com encaixes. Não existe comunicação explícita, sendo a única forma de comunicação entre os robôs realizada por meio do ambiente ou pelo contato físico entre os robôs. A comunicação pelo ambiente ocorre por meio dos blocos construtores que possuem tags RFID que especificam a sua posição. Experiências buscando avaliar a escalabilidade do sistema mostraram que, para até 60 robôs, o decaimento no tempo de construção é praticamente linear, porém, de 80 até 140, o decaimento ocorre com certa perda na linearidade.

Ambos os trabalhos citados, e diversos outros na literatura, apresentam resultados funcionais para as operações propostas com enxames, porém, tais pesquisas ainda provem soluções com níveis de otimização aparentemente baixos. Na Seção 6.4 serão apresentadas discussões relacionadas com o uso de sistemas com base em insetos sociais relacionados diretamente com os resultados obtidos no Capítulo 6.

2.4 Robótica Evolucionária

A área da Robótica Evolucionária (*Evolutionary Robotics* - ER) pode ser definida com a área de pesquisa que explora o aprendizado de robôs usando modelos análogos aos da evolução natural, aplicados em robôs reais ou simulados (Floreano e Keller, 2010; Harvey *et al.*, 2005; Nolfi e Floreano, 2000).

Duas são as abordagens mais comuns utilizadas na evolução dos comportamentos: (i) o uso de Redes Neurais Artificiais responsáveis pelo comportamento, sendo os valores de pesos sinápticos e outras características das redes obtidos por meio da evolução usando Algoritmos Genéticos e (ii) o uso de Algoritmos Genéticos onde os próprios genes codificam comporta-

mentos de forma direta. Esta segunda abordagem tem se apresentado mais simplista e restritiva, dando lugar quase que exclusivamente a primeira (Floreano e Keller, 2010).

Nolfi e Floreano (2000) citam que a ER compartilha diversos conceitos com o aprendizado robótico de uma forma geral, visto que no aprendizado robótico são utilizadas técnicas como Redes Neurais e Aprendizado por Reforço, porém, ressaltam que o nível de supervisão aplicado na ER é inferior aos aplicados nas técnicas usuais de aprendizado robótico. Além disso, na ER, a princípio, não são adicionadas restrições aos modelos, e, de fato, a evolução do comportamento pode levar em conta aspectos particulares da morfologia de cada robô (Cliff e Miller, 1996). Por este motivo, é de certa forma comum a não intercambialidade do código entre os robôs. Para tornar possível tal intercambialidade, o mapeamento individual dos sensores e atuadores deve ser finamente realizado, levando em consideração os aspectos ressaltados na Seção 2.2.

Brooks (1992) ressalta que uma vantagem da ER é que, ao contrário da necessidade de um desenvolvedor para realizar a modularização de aspectos como percepção, modelagem do ambiente, planejamento e execução, a ER constrói sistemas de controle inteligente onde estes aspectos são gerados autonomamente, independente da qualidade do desenvolvedor. Tal aspecto é considerado por Harvey *et al.* (2005) como *modelo mínimo de cognição*. Onde existe a capacidade de gerar sistemas de integração sensório-motor com mínima informação do ambiente.

Ainda, de acordo com Harvey *et al.* (2005), na ER a cognição tem relação com a organização de comportamentos por meio da interação do robô com seu ambiente. Neste aspecto, a cognição é considerada como a capacidade de um agente interagir com seu ambiente mantendo atividades viáveis. Não sendo assim uma propriedade interna do agente, mas uma propriedade relacional que envolve tanto o agente, o ambiente, e as restrições propostas².

Diversas aplicações têm sido desenvolvidas buscando avaliar estes conceitos e suas capacidades. Aplicações em ER podem ser vistas em áreas como navegação (com desvio de obstáculos, seguindo pontos, retornando a base), em evolução de comportamento de modelos presa-predador, em transporte colaborativo, em evolução de comunicação, entre outros. Relacionado a navegação, Floreano e Mondada (1994) utilizam redes neurais (com evolução de pesos) onde as entradas são as leituras sensoriais e as saídas são os valores aplicados nos motores; pôde-se perceber que a velocidade do robô teve forte relação com a taxa de leitura sensorial. Desta forma, a velocidade não era a máxima possível para o robô, mas sim a velocidade que permitia identificação de possíveis colisões de forma segura. Outros trabalhos relacionados a navegação podem ser vistos em (Harter, 2005; Moshaiov e Ashram, 2009). Outras experiências com ER têm relação com a capacidade de retornar para a base (por exemplo, para recarga

²Restrições, neste caso, são as considerações relacionadas a sobrevivência do indivíduo, como busca por alimento, evitar de se tornar alimento, gerar prole, etc.

de bateria), o que exige a evolução de características de certa representação interna do ambiente (Burgess *et al.*, 1997; Schmajuk e Blair, 1993). Floreano e Mondada (1996) descrevem o desenvolvimento de uma experiência neste estilo, onde a base é representada por uma cor diferente do ambiente, assim, a RNA tem mais duas entradas, uma relacionada com a cor do chão do ambiente e outra relacionada com a carga da bateria. Ainda, os trabalhos (Floreano e Nolfi, 1997; Floreano *et al.*, 1998; Nolfi e Floreano, 1998) apresentam a evolução de modelos presa-predador. Os robôs evoluem em duplas, sendo o *fitness* do predador relacionado a encostar na presa e o *fitness* da presa relacionado a escapar do predador. Existem câmeras e marcadores específicos nos robôs com a finalidade de possibilitar tais identificações. Diferentes velocidades foram consideradas nas avaliações, assim como diversas distâncias de capacidades de percepção tanto da presa como do predador. Os resultados obtidos podem ser considerados como evoluções satisfatórias de modelos reativos.

Um trabalho relacionado com a avaliação de formas de cooperação é apresentado em Waibel *et al.* (2009), onde são dispostas em uma arena caixas possíveis de serem movimentadas por um único robô e caixas que necessitam de mais de um robô para serem movimentadas. Neste trabalho, o *fitness* é aumentado em dois casos: (i) aumento individual em uma unidade quando o robô movimenta uma caixa e (ii) aumento global de uma unidade quando qualquer robô participa da movimentação das caixas maiores. A identificação das caixas é feita por meio de diferentes cores. Os resultados das simulações sugerem que populações com genomas mais similares carregavam de forma colaborativa as caixas, porém, em experiências com menor pressão seletiva (onde existe maior diversidade na população) alguns robôs se tornam especializados em apenas mover as caixas pequenas.

Ainda relacionado a modelos de cooperação, o trabalho desenvolvido por Mitri *et al.* (2009) apresenta um modelo de evolução de comunicação em um grupo de robôs com capacidade de emissão de luz, inicialmente de forma aleatória. Quando o *fitness* considerado é aumentado para todo o grupo quando qualquer robô obtém comida, os robôs aprendem a usar os sinais de luz para sinalizar que encontraram fontes de alimento. Quando o *fitness* passa a considerar apenas a alimentação do próprio robô, tal aspecto de comunicação não evolui.

Diversos destes trabalhos têm buscado dar explicações de seus resultados com base em comportamentos de animais sociais, porém, como o desenvolvimento dos comportamentos tem como base um *fitness* programado por um humano, os comportamentos evoluídos são simples evoluções de uma otimização para tal *fitness*. Embora apresentem similaridades com algumas espécies observadas na natureza, traçar um paralelo entre estes modelos de evolução e modelos sociais atualmente é, de certa forma, especulativo. Contudo, por ser uma área de pesquisa mais nova que as demais, e visto que vai de encontro com a ideia de aprendizado automático pelos robôs – que ainda apresenta muitas dificuldades em termos de pesquisas e de sua utilização em

robôs inteligentes – a robótica evolucionária merece atenção. No desenvolvimento desta tese, o Capítulo 6 usa princípios obtidos desta área de pesquisa, porém, onde a evolução não é de robôs individuais, mas sim de todo um grupo.

2.5 Considerações Finais

Este capítulo teve como objetivo descrever um panorama geral da robótica móvel, apresentando conceitos teóricos e aplicações de sistemas robóticos reais individuais e coletivos, assim como as abordagens e as características básicas de sistemas com múltiplos robôs. Além disso, foram descritos aspectos relacionados com a simulação e modelagem de sistemas robóticos, com foco na busca de simulações cada vez mais realísticas. Por fim, descrevemos duas abordagens relativamente novas (Seções 2.3 e 2.4) que buscam aprimorar a autonomia em sistemas robóticos individuais e coletivos.

Como pudemos ver, diversos trabalhos com robótica individual tem apresentado soluções interessantes e funcionais, entretanto, é cada vez mais comum a busca por maiores graus de autonomia e inteligência, tanto em robôs individuais como em operações com grupos de robôs. Para o trabalho com grupos de robôs, diversas abordagens têm sido propostas, contudo, ainda não existe consenso no que tange aos dilemas de eficiência, robustez e flexibilidade. Alguns trabalhos se apresentam bastante funcionais, entretanto, com muito pouca flexibilidade. Outros apresentam maiores graus de eficiência, entretanto, ainda carecem de robustez. Podemos ver com isso que a área da robótica é uma área com diversas soluções pontuais mas que ainda pode ter grandes contribuições de novas pesquisas.

Os trabalhos citados sobre exames robóticos mostram resultados funcionais para algumas operações propostas, porém, tais pesquisas ainda provem soluções com níveis de otimização aparentemente baixos, principalmente pelo fato de serem desenvolvidas em simulação, onde o desenvolvimento real parece bastante difícil de ser transportado da simulação. Embora apresentem capacidade de escalabilidade inerente, mais estudos focando eficiência e investigações sobre o desenvolvimento real dos sistemas são necessários.

Os trabalhos descritos sobre robótica evolucionária têm como ideia canônica que o aprendizado do robô seja completamente autônomo e individualizado, sendo, em geral, bastante ressaltado o fato de que o aprendizado de um robô não seja intercambiável para outro robô com o mesmo hardware devido as pequenas diferenças nos sensores e atuadores – mesmo que os sensores e atuadores sejam “os mesmos”. A área da robótica evolucionária (que é, de certa forma, também bio-inspirada) parece promissora no desenvolvimento de sistemas robóticos mais autônomos, embora, os resultados atuais ainda se apresentem bons para sistemas de pequena escala,

é possível que essa área tenha muito o que agregar no desenvolvimento de robôs individuais de larga escala, ou grupos de robôs. Algumas pesquisas têm buscado apresentar similaridades entre os robôs evoluídos com algumas espécies observadas na natureza, entretanto, este não é nosso objetivo. Por ser uma área de pesquisa mais nova que as demais, e visto que vai de encontro com a ideia de aprendizado automático pelos robôs – que ainda é uma linha de pesquisa com muitas questões em aberto referente ao desenvolvimento e uso de robôs inteligentes – a robótica evolucionária merece atenção. No desenvolvimento desta tese, o Capítulo 6 usa princípios obtidos desta área de pesquisa, porém, onde a evolução não é de robôs individuais, mas sim do comportamento de todo um grupo.

As iniciativas nacionais, como o Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos³ (INCT-SEC) e o Centro de Robótica de São Carlos⁴ (CRob), das quais o LRM faz parte, possuem, entre outros, projetos relacionados com o desenvolvimento de veículos autônomos de grande porte (para navegação autônoma), bem como o desenvolvimento de veículos autônomos de pequeno porte para patrulhamento e vigilância. Esta tese se insere também neste contexto, onde buscamos soluções inteligentes que para o desenvolvimento de robôs móveis autônomos. Este capítulo de revisão mostrou alguns trabalhos que podem servir como inspiração para desenvolvimento de novos métodos. As iniciativas do INCT-SEC e do CRob são importantes iniciativas nacionais que devem colocar o Brasil junto ao estado da arte da pesquisa em robótica.

³<http://www.inct-sec.org/br>

⁴<http://www.eesc.usp.br/crob>

Aprendizado de Máquina

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

Das diversas técnicas de ML existentes, pode-se dizer que a grande maioria tem focos em grupos relativamente específicos de aplicações. É difícil imaginar uma técnica que seja capaz de resolver todas as gamas de problemas que envolvam aprendizado automático, seja ele supervisionado, não supervisionado ou semi-supervisionado (Mitchell, 1997).

Segundo Monard e Baranauskas (2003), podemos classificar paradigmas de aprendizado em cinco grandes grupos: (i) Simbólico, como Árvores de Decisão e Redes Semânticas; (ii) Estatístico, como Aprendizado Bayesiano; (iii) Baseado em exemplos, como sistemas de Raciocínio Baseado em Casos; (iv) Conexionista, como Redes Neurais Artificiais; (v) Evolutivo, como Algoritmos Genéticos.

A pesquisa desenvolvida e descrita no presente documento tem relação estreita com os paradigmas conexionista e evolutivo, assim, no decorrer deste capítulo, serão descritos fundamentos e exemplos de aplicações das técnicas utilizadas. Neste contexto, o conceito de aprendizado pode ser considerado como a obtenção de regras na abordagem simbólica, a obtenção de pesos de conexões na abordagem conexionistas ou ainda, a otimização de funções na abordagem

evolutiva (Braga *et al.*, 2007; Mitchell, 1997). A escolha pelos paradigmas conexionista e evolutivo se deu devido a estes dois serem os paradigmas que tem apresentado resultados mais promissores em se tratando de desafios da área de robótica móvel. Tais resultados promissores tem relação tanto com a eficiência obtida pelos robôs como pela diminuição da dificuldade no desenvolvimento dos comportamentos.

Como descrito na Introdução, tendo como objetivo apresentar um capítulo mais sucinto, deixamos para os capítulos de investigações (Capítulos 4, 5 e 6) alguns dos detalhes mais pontuais avaliados sobre cada técnica. Estes detalhes pontuais serão descritos nas seções estreitamente relacionados com sua importância relacionada ao item específico de pesquisa avaliado.

3.1 Aprendizado Evolutivo

3.1.1 Otimização por Enxame de Partículas

A Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) (Eberhart e Kennedy, 1995; Kennedy e Eberhart, 1995) é uma técnica estocástica de otimização, inspirada em comportamentos sociais como os de bandos de pássaros e cardumes de peixes (Eberhart *et al.*, 2001; Engelbrecht, 2006) (Figura 3.1). A otimização no PSO ocorre por meio da cooperação (aprendizado do grupo) e da competição (aprendizado individual) entre indivíduos de um bando. A cooperação e a competição são as principais características que guiam o algoritmo do PSO na exploração do espaço de busca (Engelbrecht, 2006).

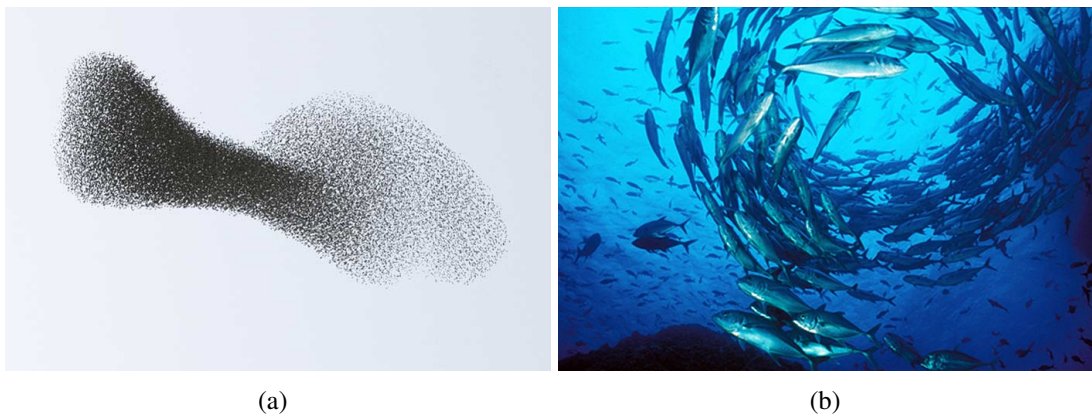


Figura 3.1: Inspirações do PSO. (a) Bando de pássaros. Crédito: (Prestri, 2010). (b) Cardume de peixes. Crédito: (Hall, 2010).

A otimização no PSO é dada pela evolução de um enxame de partículas (análogo a uma população de indivíduos em um Algoritmo Genético (AG)), inicialmente aleatórias. A cada

iteração, cada partícula é atualizada, seguindo dois melhores valores: o melhor *fitness* da população e o melhor *fitness* encontrado pela partícula (considerando suas gerações passadas). Cada partícula possui uma posição x (ou um vetor de posições) e uma velocidade v (ou um vetor de velocidades). A *posição* é a representação da solução do problema, e a *velocidade* é o peso do direcionamento para o deslocamento das partículas. A Figura 3.2 apresenta o fluxograma de um PSO padrão. Uma diferença conceitual importante entre AG e PSO é que a força motriz no primeiro é a sobrevivência dos melhores indivíduos, os quais competem para sobreviver e gerar uma prole. No segundo, a força motriz é a interação social e a troca de conhecimento sobre o espaço de busca (Engelbrecht, 2006).

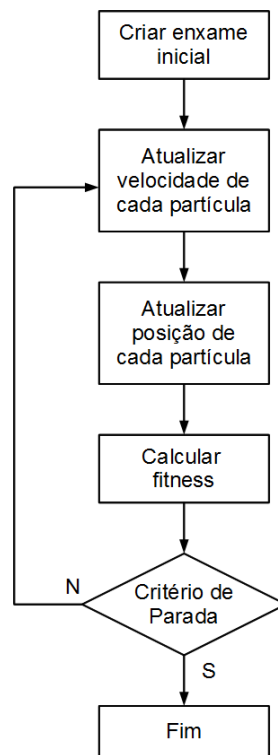


Figura 3.2: Fluxograma de um PSO padrão.

As Equações 3.1 e 3.2 regem o comportamento da otimização. O cálculo da nova posição da partícula é dado pela Equação 3.1. Nesta, x_k^i é a posição da partícula i no instante k , e v_k^i é a velocidade da partícula i no instante k . A velocidade da partícula é atualizada de acordo com a Equação 3.2, onde v_k^i é a velocidade da partícula i no instante k , w é um parâmetro que representa a inércia da partícula; $pbest$ é a melhor posição dentre todas já encontradas pela partícula (*particle best*); $gbest$ é a melhor posição dentre todas já encontradas pelo grupo (*group best*); c_1 e c_2 são parâmetros de confiança; r_1 e r_2 são números aleatórios compreendidos entre 0 e 1. Os parâmetros (w , c_1 , c_2 , r_1 e r_2) são detalhados no decorrer desta seção.

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (3.1)$$

$$v_{k+1}^i = w \cdot v_k^i + c_1 \cdot r_1(pbest - x_k^i) + c_2 \cdot r_2(gbest - x_k^i) \quad (3.2)$$

A velocidade é o parâmetro responsável pela modificação dos valores da partícula (atuando no processo de otimização) (Engelbrecht, 2006) e reflete tanto o conhecimento individual da partícula como o conhecimento do grupo. O conhecimento individual é identificado como *Componente Cognitivo*, enquanto o conhecimento do grupo é conhecido como *Componente Social*. A velocidade consiste de uma soma de três termos: (i) Velocidade Prévia, que serve como uma memória da direção de deslocamento e pode ser vista como um parâmetro que evita a mudança drástica de direção; (ii) Componente Cognitivo, que direciona o indivíduo para a melhor posição encontrada, até então, pela partícula; assemelha-se à memória individual da posição que era melhor para a partícula; (iii) Componente Social, que direciona o indivíduo para a melhor posição encontrada, até então, pelo grupo. Considerando que a partícula é atraída pelo *gbest* (que pode ser o melhor global ou o melhor vizinho, de acordo com a topologia de vizinhança utilizada) e pelo *pbest*, essa atração é o que faz o PSO tender o movimento das partículas para as melhores soluções encontradas até o momento.

Os parâmetros c_1 e c_2 , também chamados de confiança, são utilizados para definir a importância da tendência individual ou social. O PSO padrão trabalha com valores de confiança iguais ($c_1 = c_2$), atribuindo igual importância à experiência do grupo e à experiência individual (conhecido como *Modelo Completo*). Quando o parâmetro c_1 possui valor zero, e o valor c_2 tem valor maior que zero, o PSO usa apenas informação do grupo, não considerando a experiência individual. Este modelo é conhecido como *Modelo Social*. Quando o parâmetro c_2 possui valor zero, e o valor c_1 tem valor maior que zero, o PSO usa apenas informação da própria partícula, não considerando a experiência do grupo. Este modelo é conhecido como *Modelo Cognitivo*.

O uso dos valores randômicos (r_1 e r_2) no ajuste da velocidade tem como finalidade permitir ao PSO explorar de forma mais ampla o espaço de busca. É com o uso destes parâmetros que evita-se uma convergência rápida e unânime, explorando de forma mais ampla o espaço de busca (Eberhart *et al.*, 2001; Engelbrecht, 2006).

O parâmetro de inércia (w) tem como finalidade permitir o ajuste no que se refere à busca local ou global (Shi e Eberhart, 1998; Silveira *et al.*, 2009). Quanto mais perto de 1,0 for este valor, mais global é a busca, enquanto tamanhos pequenos possibilitam uma busca local. Este valor, em geral, varia entre 0,4 e 0,9. Alguns trabalhos exploram avaliações como decaimento linear da inércia ou o uso de funções como senóides no controle da inércia. Entretanto, como

ressaltam Kennedy e Eberhart (1995) e Engelbrecht (2006) a grande maioria dos parâmetros é dependente do problema.

As estruturas sociais (ou topologias) definem a forma de se obter o *gbest*, que pode ser o melhor global ou o melhor vizinho (Eberhart *et al.*, 2001; Engelbrecht, 2006). O trabalho feito por Kennedy (1999) mostra que a vizinhança do tipo estrela (em que a partícula tem conhecimento da melhor posição encontrada entre todas as partículas) é a forma para convergência mais rápida. Contudo, de acordo com Saxena e Vora (2008), a topologia estrela explora de forma menos ampla o espaço de busca.

As aplicações de PSO são, em geral, parecidas com as da Computação Evolucionária. Um trabalho de referência que descreve uma comparação entre AG e PSO pode ser visto em (Eberhart e Shi, 1998). Em (Pugh e Martinoli, 2006), é descrito o desenvolvimento de um PSO para realizar o controle de navegação de robôs, em que valores de sensores de distância são usados como informação de entrada em um sistema que comanda os motores dos robôs para realizar a navegação, desviando de obstáculos. O trabalho de (Rong *et al.*, 2009) descreve o desenvolvimento e a avaliação de um PSO para realizar o caminhar robótico com robôs de quatro pernas, no qual se busca otimizar o tempo de aplicação de forças e valor das forças a serem aplicadas em diversas articulações. Uma abordagem semelhante é apresentada em (Niehaus *et al.*, 2007), porém, neste caso, o estudo de caso ocorre com um robô bípede.

Os trabalhos (Pessin e Osório, 2009c; Pessin *et al.*, 2010c) apresentam avaliações de diversos parâmetros que influenciam o comportamento da otimização de um PSO aplicado à formação de grupos de robôs. O PSO é responsável por encontrar posições de atuação para os robôs do grupo, sendo a finalidade dos robôs a atuação no combate de um incêndio florestal por meio da criação de aceiros. Para as condições propostas pelos autores, os melhores resultados foram obtidos utilizando o Modelo Completo, ao contrário do Modelo Social ou Modelo Cognitivo. O valor de inércia se mostrou suscetível ao tipo de modelo, sendo os melhores resultados obtidos com Modelo Completo utilizando inércia de 0,5 e os melhores resultados obtidos com Modelo Social utilizando inércia de 0,8. Estes trabalhos demonstram que o uso de algoritmos de Aprendizado de Máquina, neste caso, usando a otimização por exame de partículas (PSO), podem ser efetivos em aplicações de robótica móvel autônoma que buscam definir estratégias de ação de grupos robóticos. Estes trabalhos, usando PSO, serão detalhados na Seção 6.3.3

3.1.2 Algoritmos Genéticos

Algoritmos Genéticos (AG) (Holland, 1975; Mitchell, 1998) são técnicas de otimização que tem como inspiração a teoria da evolução pela seleção natural. O primeiro passo em um Algoritmo Genético típico é a geração de uma população inicial, usualmente formada por um conjunto

aleatório de valores que representam possíveis soluções do problema. Durante o processo evolutivo, cada indivíduo da população é avaliado por meio de uma função de aptidão (*fitness*) que, em geral, deve permitir manter os indivíduos mais aptos e descartar os menos aptos. Nos indivíduos selecionados, é feito cruzamento e mutação, gerando descendentes para a próxima geração (Carvalho *et al.*, 2003; Goldberg, 1989; Nolfi e Floreano, 2000). Este processo é repetido até que a condição de parada seja satisfeita, como apresenta o Algoritmo 3.1.

AGs são estruturados de forma que as informações referentes a um determinado sistema possam ser codificadas de maneira análoga a cromossomos biológicos, como uma sequência de valores, onde usualmente cada fragmento da sequência representa uma variável.

Algoritmo 3.1 Algoritmo Genético Típico

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

$t \leftarrow 0$

inicializar $S(t)$

avaliar $S(t)$

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

$t \leftarrow t+1$

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

aplicar *crossover* sobre $S(t)$

aplicar mutação sobre $S(t)$

avaliar $S(t)$

fim enquanto

O cruzamento e a mutação são as rotinas que transformam uma população através de sucessivas gerações. Esses operadores possibilitam que o AG percorra pontos do espaço de busca até chegar a um resultado satisfatório (Goldberg, 1989). Estas operações ocorrem após a seleção dos indivíduos, que tem como princípio de escolha o valor do *fitness* do indivíduo. A aplicação do cruzamento faz com que, para cada dupla de indivíduos pais selecionados, sejam feitos um ou mais pontos de corte na sequência, e, para cada ponto destes, seja feita a troca das sequências, gerando cromossomos filhos. A mutação faz com que um número na sequência genética do cromossomo seja modificado, isso melhora a diversidade da população, porém, se a taxa de mutação for muito elevada, a população pode nunca convergir, ficando essencialmente aleatória (Carvalho *et al.*, 2003). Um exemplo de cruzamento e mutação pode ser visto na Figura 3.3.

O princípio básico do funcionamento dos AGs é que um critério de seleção permita que, a partir de uma população inicial, obtenhamos uma população final mais apta a resolver o problema em questão. Para que isto ocorra, é necessário que consigamos medir, de alguma forma, o quanto cada indivíduo está apto a resolver o problema. Esta medida é feita através de uma

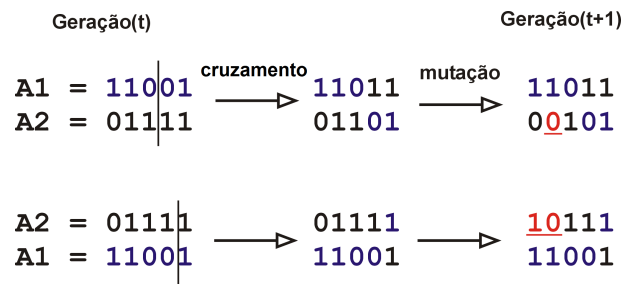


Figura 3.3: Cruzamento e mutação, adaptado de Lacerda e Carvalho (1999).

função de aptidão (*fitness*) que, usualmente, é uma fórmula matemática. Esta fórmula matemática, em geral, pode estar relacionada ao modo como as percepções de um indivíduo geram reações afetando o próprio indivíduo e o ambiente.

Uma característica importante a ser considerada na implementação de um AG é o elitismo, proposto por (Jong, 2002). O princípio do elitismo é a transferência do melhor indivíduo da geração atual para a próxima, evitando a perda do melhor indivíduo devido ao cruzamento ou a mutação. Outra característica que pode ser comparada ao elitismo é o uso de populações sobrepostas, onde grupos da população original são mantidos, não apenas o melhor indivíduo. Trabalhos explorando o uso de populações sobrepostas podem ser vistos em (Pessin *et al.*, 2010d; Souza *et al.*, 2012b). Na Seção 6.3.2 é descrito como o uso de populações sobrepostas é implementado junto ao AG, bem como é analisada a influência da mesma na investigação realizada.

Outros parâmetros que têm grande influência no processo de busca de um AG são o tamanho da população, as taxas de mutação e cruzamento, os tipos de cruzamento e de seleção e o critério de parada. O trabalho (Pessin e Osório, 2009b, 2010) apresenta a avaliação destes diversos parâmetros em um AG para encontrar posições de atuação para grupos de robôs, aplicados em combate de incêndio florestal. Mais detalhes sobre esta avaliação serão descritos no Capítulo 6.

Os trabalhos realizados com AG pelo autor deste documento fizeram uso da biblioteca GAlib (Wall, 1996). A GAlib é uma biblioteca de software livre desenvolvida por Matthew Wall em sua tese no *Massachusetts Institute of Technology* (MIT). Esta biblioteca contém um conjunto bastante amplo de funções relacionadas a programação de Algoritmos Genéticos; permitindo a implementação de diversas funções de *fitness*, diversas representações genéticas, diversos operadores genéticos, e diversos métodos de seleção.

Algumas aplicações com AGs podem ser vistas no seguintes trabalhos: (Ren *et al.*, 2008) e (Orlando *et al.*, 2010) apresentam o desenvolvimento de AGs para realizar planejamento de trajetória de dispositivos como braços robóticos com diversas juntas. Os trabalhos (Jun e Qingbao, 2010) e (Jianguo *et al.*, 2010) apresentam o desenvolvimento de AGs para realizar navegação

robótica. Ainda, em (Heinen e Osório, 2009) é apresentado um AG para evoluir a morfologia e os parâmetros de controle de robôs com quatro pernas, buscando capacidade de caminhar da forma rápida e estável. Diversos trabalhos com AGs apresentam resultados satisfatórios para ambientes estáticos. Uma proposta com o uso de AGs em ambientes dinâmicos é apresentada em (Pessin e Osório, 2009b, 2010), onde o sistema possui duas fases: (i) fase de evolução, onde são obtidos os valores para atuação; (ii) fase de atuação, onde o sistema atua conjuntamente com um modo monitor. Qualquer modificação no ambiente ativa uma nova evolução, que tem como base não o uso de uma população aleatórias, mas a reutilização da população previamente obtida pela evolução – de certa forma, esta forma de utilização de população inicial é um tipo de heurística, e permite obter resultados satisfatórios com um número menor de gerações. Este trabalho representa uma importante proposta de abordagem e será detalhado na Seção 6.3.1.

3.1.3 Arrefecimento Simulado

Arrefecimento Simulado (*Simulated Annealing* - SA) (Kirkpatrick *et al.*, 1983) é uma técnica com origem em mecânica estatística, baseada em uma metáfora de um processo térmico de recozimento, utilizado em metalurgia para obtenção de estados de baixa energia num sólido. O processo de recozimento consiste de (i) a temperatura do sólido é aumentada para um valor máximo no qual ele se funde e (ii) o resfriamento deve ser realizado lentamente até que o material se solidifique. O resfriamento lento faz com que os átomos que compõem o material sejam organizados em uma estrutura uniforme, como resultado, é esperado uma redução dos defeitos do material.

A idéia fundamental do SA é que nele sejam permitidas modificações no grupos de soluções, apresentando algumas alternativas piores que a solução corrente, o que permite uma exploração do espaço de busca a fim de tentar escapar de mínimos locais (Becceneri, 2009). A otimização no SA ocorre através da avaliação de soluções vizinhas de uma única solução inicial aleatória. Esta avaliação é feita através de uma função de avaliação (*fitness*) (Michalewicz e Fogel, 2004). O SA padrão usa apenas um valor inicial, e toda a otimização tem como base esse único valor. A partir deste valor inicial são geradas soluções candidatas (também chamadas de soluções vizinhas). O algoritmo executa por um número de passos definido (análogo a quantidade de gerações em um AG). O algoritmo de SA substitui a solução atual por uma solução próxima escolhida de acordo com o *fitness* e com uma variável de temperatura t . Quanto maior for t , maior a aleatoriedade incluída na próxima solução escolhida. Com a execução do algoritmo, o valor de t é decrementado, fazendo o algoritmo estabilizar e convergir para uma solução ótima local.

Outros parâmetros que influenciam diretamente a otimização são (i) quantidade de vizinhos e (ii) forma de obtenção de vizinhos. A quantidade de vizinhos é o parâmetro que define a quantidade de soluções avaliadas em cada passo da otimização; é, de certa forma, análoga ao conceito de população dos AGs. A forma de obtenção de vizinhos é análoga ao conceito de mutação.

Aplicações de SA tem encontrado sucesso em diversas áreas de problemas não lineares, como problemas de distribuição (logística de transporte de carga) e planejamento de produção. Exemplos de aplicações podem ser vistos em (Koulamas *et al.*, 1994) e (Suman e Kumar, 2006). A investigação apresentada por Pessin *et al.* (2010b), utilizando SA para encontrar posições de atuação para um grupo de robôs, será detalhada na Seção 6.3.4.

3.2 Aprendizado Conexionista

3.2.1 Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) é uma coleção de unidades (neurônios) conectadas por ligações com pesos (sinapses). Unidades de entrada e saída recebem e transferem os sinais do ambiente e para o ambiente. Unidades internas são chamadas de ocultas, dado que elas não tem contato com o ambiente externo da RNA (Nolfi e Floreano, 2000). RNAs são sistemas essencialmente paralelos (Braga *et al.*, 2007; Mitchell, 1997), embora facilmente implementados em computadores sequenciais. Os atributos básicos de uma RNA podem ser divididos em arquitetura e neurodinâmica. A arquitetura determina a estrutura da rede, ou seja, o número de neurônios e sua interconectividade e a neurodinâmica, por sua vez, define as propriedades funcionais da rede, ou seja, como ela aprende, recupera, associa e compara novas informações com o conhecimento já armazenado (Kartalopoulos, 1995). Matematicamente, RNAs são aproximadores universais, que realizam mapeamentos em espaços de funções multivariáveis (Hornik *et al.*, 1989). A capacidade de aprender e generalizar¹ das RNAs é um dos seus maiores atrativos, o que lhe proporciona um poder muito além do simples mapeamento direto de entradas e saídas.

O processamento da informação em uma RNA é feito por meio de estruturas neurais artificiais (Braga *et al.*, 2007), sendo que esta estrutura, bem como o próprio neurônio artificial são uma analogia biológica ao funcionamento do cérebro. O neurônio artificial proposto por (McCulloch e Pitts, 1943) (Figura 3.4) pode ser descrito como um modelo com n terminais de entrada $x_1, x_2, x_3, \dots, x_n$ que, para cada entrada, possui um peso w_i correspondente. A soma das

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

entradas x_i ponderadas pelos pesos correspondentes w_i produzem a chamada *saída linear* u . A saída y do neurônio é obtida pela aplicação de uma função $f(\cdot)$ à saída linear u . A função $f(\cdot)$ é chamada *função de ativação* e pode assumir diferentes formas (e.g. linear, limiar, hiperbólica, sigmóide).

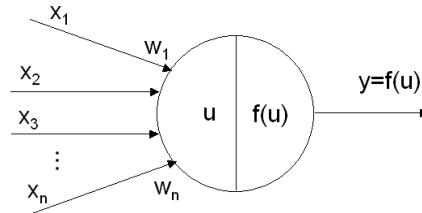


Figura 3.4: Neurônio de McCulloch e Pitts (MCP), adaptado de (Braga *et al.*, 2007).

A partir do trabalho de McCulloch e Pitts (1943) que enfocou a modelagem de um neurônio, Rosenblatt (1958) propôs o modelo *Perceptron*, introduzindo o conceito de aprendizado em RNA. O modelo *Perceptron* era composto por uma estrutura de rede tendo como unidades básicas nodos MCP e de uma regra de aprendizado. De acordo com Rosenblatt (1962), o aprendizado no modelo *Perceptron* sempre converge caso o problema em questão seja linearmente separável. Entretanto, para a solução de problemas não linearmente separáveis é necessária mais de uma camada de neurônios, como ressalta Braga *et al.* (2007). A solução de aprendizado para redes com *Perceptrons* combinados em diversas camadas, formando a chamada rede neural do tipo *Multi Layer Perceptron* (MLP) foi apresentada em Rumelhart e McClelland (1987) com o algoritmo de aprendizado *Backpropagation*.

O algoritmo *Backpropagation* é um algoritmo supervisionado. A base de treinamento é um conjunto de dados que deve apresentar, para cada entrada, a saída prevista do sistema. Este tipo de aprendizado ocorre em várias épocas; cada época representa a apresentação do conjunto inteiro de dados à rede neural para o ajuste dos pesos. O treinamento no *Backpropagation* ocorre em duas fases: a fase *forward* é utilizada para definir a saída da rede para um dado padrão de entrada; a fase *backward* utiliza a saída desejada e a saída fornecida pela rede para atualizar o peso das conexões. O *Backpropagation* é baseado na Regra Delta Generalizada (Widrow e Hoff, 1960) e os ajustes dos pesos são realizados pelo método da Descida do Gradiente (Braga *et al.*, 2007; Mitchell, 1997).

Por utilizar o método da Descida do Gradiente, não há garantia de que o aprendizado não tenha ficado preso em mínimos locais (Figura 3.5), assim, o treinamento de uma rede neural deve envolver várias rodadas de simulação, inicializando os pesos de forma aleatória. Outra questão importante é o grau de generalização. Quando uma RNA é treinada, os pesos sinápticos vão se ajustando para que sejam dadas respostas satisfatórias à base de treinamento apresentada. Se o aprendizado for feito por um número muito grande de épocas pode ocorrer o chamado

overfitting, onde a Rede Neural começa a decorar a saída, perdendo sua capacidade de generalização. A solução deste problema usualmente envolve o uso de uma base de validação usada em paralelo a base de treinamento, como mostra a Figura 3.6.

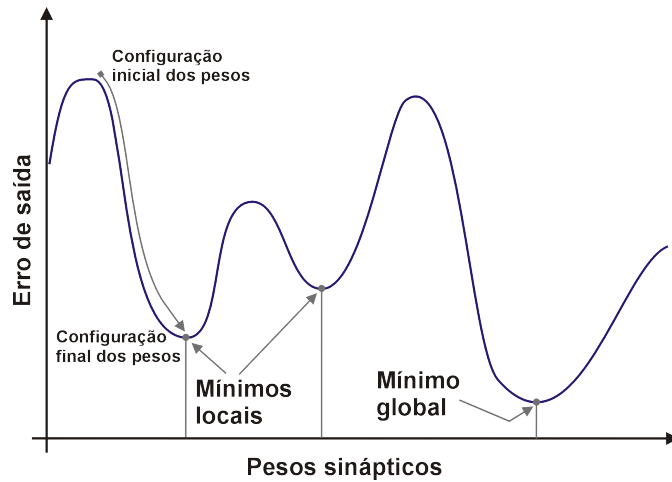


Figura 3.5: Descida do gradiente de uma superfície de erro.

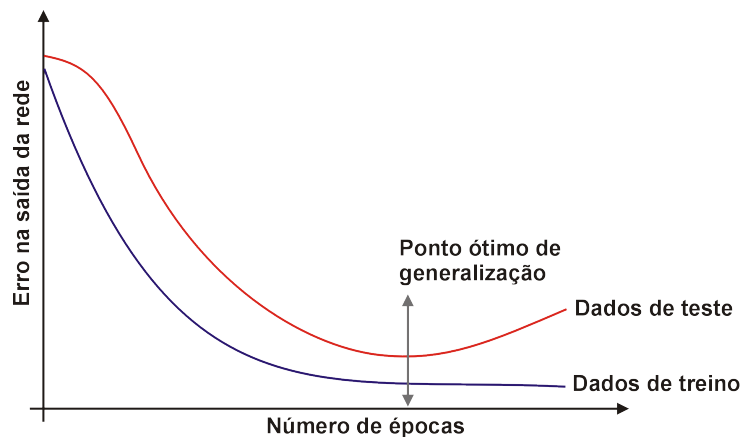


Figura 3.6: Curvas de erro em aprendizado e validação.

O algoritmo *Backpropagation* padrão é muito lento, sendo que seu desempenho piora sensivelmente para problemas maiores e mais complexos. Mesmo para problemas relativamente simples, geralmente requer que todos os padrões de treinamento sejam apresentados centenas ou milhares de vezes, o que limita sua utilização prática (Braga *et al.*, 2007). Os algoritmos normalmente empregados no treinamento das RNAs são variações do *Backpropagation*, tendo em geral, o foco de aumentar a velocidade do treinamento ou melhorar a classificação de padrões. As variações mais conhecidas são: *Backpropagation* com *Momentum* (Rumelhart e McClelland, 1987); *Quickprop* (Fahlman, 1988); e *Resilient Backpropagation* – RProp (Riedmiller e Braun, 1994).

É comum encontrar aplicações de Redes Neurais Artificiais em problemas que necessitem aproximação, predição ou classificação. O trabalho (Pessin *et al.*, 2007d) apresenta uma RNA capaz de realizar a navegação de um robô simulado em um ambiente bidimensional, a RNA é capaz de direcionar o robô para áreas com menor densidade de ocupação por vegetação, as entradas da RNA são as densidades de vegetação observadas e a saída é o ângulo para o qual o agente deve se deslocar. Em (Pessin *et al.*, 2010a) são apresentadas melhorias das propostas anteriores, onde é descrita uma RNA capaz de realizar a navegação de um robô simulado em um ambiente tridimensional, as entradas da RNA são informações coletadas de sensores (localização, orientação, distância de obstáculos) e as saídas são a velocidade e o ângulo a serem aplicados nos motores linear e angular, respectivamente. Ainda, o trabalho (Sales *et al.*, 2010) apresenta uma RNA para classificar regiões navegáveis e não navegáveis de imagens, as entradas da RNA são atributos de cor, como média dos canais de cor e entropia.

Diversas outras aplicações com redes neurais podem ser vistas em (Braga *et al.*, 2007; Mitchell, 1997; Silva *et al.*, 2010). A principal característica das redes neurais que vai de encontro com nossas necessidades é a possibilidade quase direta em substituir partes de sistemas que tenham como base conjuntos de regras. Ou seja, no lugar de um conjunto de regras, sujeitas a qualidade do desenvolvedor, podemos colocar uma rede neural que aprende de forma automática utilizando uma base de dados disponibilizada pelo desenvolvedor. As redes neurais ainda apresentam inerentes capacidades de generalização e não-linearidade. As investigações com redes neurais serão apresentadas nos Capítulos 4 e 5.

Os trabalhos apresentados em (Pessin *et al.*, 2007d) e (Pessin *et al.*, 2010a) utilizaram, no desenvolvimento da RNA, o *Stuttgart Neural Network Simulator* (SNNS) (Zell *et al.*, 1995). O SNNS é um ambiente de desenvolvimento e treino RNAs, possuindo um grande número de algoritmos de aprendizado, como *Backpropagation*, *Quickprop*, *RProp*, *Backpercolation*, *Counterpropagation*, entre outros. O núcleo do sistema é desenvolvido em linguagem C e sua utilização pode ser completamente feita por meio de linha de comando, porém também possui uma interface desenvolvida em JAVA (JavaNNS). Um aplicativo do pacote SNNS, o SNNS2C, permite a conversão de uma RNA em código C, que pode então ser facilmente inserido em uma outra aplicação.

3.3 Considerações Finais

Este capítulo teve como objetivo descrever fundamentos teóricos e exemplos de aplicações das técnicas de Aprendizado de Máquina diretamente relacionadas a esta pesquisa. A partir das características descritas, pudemos perceber quais as abordagens utilizadas para diferentes classes

de problemas. Usualmente, o aprendizado evolutivo tem maior relação com trabalhos onde o foco é o aprendizado não supervisionado – onde não são utilizadas (ou não são possíveis de serem obtidas) bases de dados para o aprendizado. O aprendizado supervisionado, notadamente por meio de redes neurais, é geralmente utilizado quando bases de dados de treinamento são disponíveis.

Ainda, é importante ressaltar, que o capítulo teve como objetivo descrever de forma relativamente sucinta as características dos métodos. Detalhes particulares de alguns parâmetros serão descritos nos Capítulos 4, 5 e 6, além de também poderem ser encontrados nas bibliografias sugeridas.

Existem ainda diversas outras técnicas de aprendizado de máquina que não foram utilizadas nesta tese, nem mesmo apresentadas neste capítulo. Algumas delas foram avaliadas em artigos relacionados, como as árvores de decisão (Quinlan, 1986) investigadas no problema de navegação em (Souza *et al.*, 2010) e as máquinas de suporte vetorial (Boser *et al.*, 1992) investigadas no problema de localização em (Marinello *et al.*, 2012), porém não são detalhadas na tese pela questão de não terem apresentado resultados significativamente melhores que seus pares. Uma técnica que não fez parte da tese, mas que está sendo considerada para trabalhos futuros por ter se mostrado eficiente em diversos casos na literatura, é a de processos Gaussianos (Rasmussen, 2004).

Como fruto da utilização das diversas técnicas, é vislumbrado como trabalho futuro o estudo de diversos modelos híbridos entre as técnicas, com objetivo de avaliar possíveis melhorias nos métodos para classes específicas de problemas.

Localização

Este capítulo trata da pesquisa e desenvolvimento de métodos alternativos para auto-localização de robôs móveis autônomos, sendo uma alternativa ao uso do GPS, bem como, podendo ser usado em ambiente internos (indoor). Este trabalho está inserido no contexto da investigação do uso de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina para auxiliar na tarefa de localização interna de robôs móveis. O objetivo do capítulo é descrever as propostas e as avaliações de métodos de localização de agentes móveis usando a potência do sinal obtido de Pontos de Acesso (*Access Point – AP*). Inicialmente é discutida a motivação do método proposto, inserida dentro do estado da arte. Em seguida, apresentamos a metodologia proposta, a bateria de avaliações realizadas e discussão sobre as avaliações. Ainda, descrevemos uma possível alternativa com o objetivo de tornar o sistema proposto completamente autônomo e tolerante a falhas.

Os trabalhos de localização apresentados neste capítulo são trabalhos integrando interesses do Laboratório de Robótica Móvel do ICMC/USP¹, do Laboratório de Sistemas Distribuídos e Programação Concorrente do ICMC/USP, do *Communication and Distributed Systems Group (CDS)* do *Institute of Computer Science and Applied Mathematics*, Universidade de Berna² – Suíça e do *Robotics Lab*³, da *School of Maths and Computer Science (MACS)*, Universidade

¹<http://www.lrm.icmc.usp.br/>

²Universidade onde foi realizada visita técnica em outubro de 2012.

³<http://www.macs.hw.ac.uk/RoboticsLab>

Heriot-Watt⁴ – Edimburgo, Reino Unido. Diversos trabalhos aqui apresentados são resultantes deste projeto de cooperação internacional, sendo que as seguintes publicações advêm destes trabalhos em forma de artigos: (Ailincă *et al.*, 2012; Marinello *et al.*, 2012; Pessin *et al.*, 2012, 2011a,b).

As principais contribuições deste capítulo são:

- Proposta e avaliação de uma arquitetura para localização de robôs móveis usando o valor da potência de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina (por meio de redes neurais).
- Investigação sobre filtros aplicados no valor da potência lido dos pontos de acesso. O valor da potência é bastante instável, sendo assim objetivo da investigação avaliar as mudanças no comportamento geral do sistema, bem como da acuracidade no aprendizado, quando usando diferentes filtros. Esta investigação está detalhada na Seção 4.3.1.
- Investigação sobre a relação entre a quantidade de redes *wireless* disponíveis (quantidade de entradas no sistema) e a acuracidade no aprendizado da posição pela da rede neural. Detalhada na Seção 4.3.2.
- Investigação sobre topologias de redes neurais, buscando avaliar a acuracidade de diferentes modelos. Detalhada na Seção 4.3.3.
- Proposta e avaliação de métodos para evolução de redes neurais. São realizadas avaliações tanto com evolução dos pesos como com evolução de topologias. São apresentados resultados comparativos entre o uso de métodos evolucionários e métodos clássicos de treinamento. As técnicas de evolução investigadas vão de encontro com pesquisas da área de Robótica Evolucionária, de acordo com a Seção 2.4. Estas investigações estão detalhadas nas Seções 4.3.4 e 4.3.5.
- Descrição de possíveis alternativas para tornar o modelo proposto mais autônomo e tolerante a falhas. Detalhadas nas Seções 4.3.6 e 4.3.7.

O restante do capítulo é dividido em quatro principais seções. A primeira descreve a motivação da pesquisa. A segunda seção apresenta a metodologia. Em seguida são descritos os resultados e as discussões, detalhando as investigações realizadas. Por fim, são apresentadas as considerações finais e os trabalhos futuros.

⁴Universidade onde foi realizado o estágio de doutorado no exterior (sanduíche), de novembro de 2011 até setembro de 2012.

4.1 Motivação

Localização é uma informação fundamental para a realização de diversas tarefas nas áreas de robótica móvel (Siegwart *et al.*, 2011) e de redes de sensores sem fio (Dargie e Poellabauer, 2010). Localização pode ser informação obtida por sensores presentes no agente (*e.g.* usando GPS, odometria). Ainda, localização pode ser uma informação obtida por sensores externos ao agente, ou seja, obtida por um sistema externo (*e.g.* como uma câmera de teto). Tal informação de localização por fonte externa pode ser tanto usada para informar o agente de sua posição por meio de comunicação ou então apenas para rastreamento do mesmo (*e.g.* rastreamento da movimentação de um veículo dentro de um estacionamento).

Graus mais ou menos acurados na obtenção da posição dependem da natureza da tarefa. Por exemplo, um sistema de rastreamento de pessoas ou veículos poderia ser útil com erro de poucos metros. Talvez um sistema de monitoração de incêndio florestal possa ser útil com erro na casa de algumas dezenas de metros. Entretanto, algumas tarefas robóticas podem necessitar de localização com erro inferior a poucos centímetros (*e.g.* tarefas domésticas).

Como descrito na introdução, o atual sistema de GPS é bastante útil para obtenção de posição (localização) em ambientes externos. Entretanto, é comum que apresente imprecisão de alguns metros – principalmente quando adotados dispositivos padrão de GPS, de custo relativamente baixo. Algumas exceções que prometem maior acuracidade existem, como as que utilizam correção diferencial, baseada no uso de mais de um receptor, mas que, para se ter acesso a uma precisão na ordem de poucos centímetros os custos destes equipamentos podem se tornar proibitivos para muitas aplicações. Nos modelos mais comuns, correções por *software* têm sido aplicadas, bem como a integração com outros sensores (como unidades inerciais). Mesmo assim, em alguns ambientes externos, o sistema de GPS continua a apresentar maiores graus de imprecisão ou mesmo deficiência de funcionamento, como em áreas com grande cobertura vegetal, proximidade a grandes blocos de rochas – problemas de ponto cego ou refração. Informação sobre localização é base para o desenvolvimento de métodos inteligentes de navegação.

Alguns estudos tem buscado uma alternativa para localização em ambientes internos (neste caso, áreas com grande cobertura vegetal podem ser consideradas similares a áreas internas). Em ambientes internos, sensores como câmeras e *lasers* são comumente utilizados para estimação de posição (Lang *et al.*, 2008; Napier *et al.*, 2010). Tais sensores usualmente necessitam de mapas ou marcos no ambiente e necessitam de algoritmos que dependem de grande poder computacional. Além disso, é comum que tais sensores tenham campo de visão restrito, o que dificulta a tarefa de localização. Câmeras também apresentam problemas relacionados com ilu-

minação do ambiente e *lasers* apresentam problemas de alto custo e até mesmo de reflexão em certos tipos de superfícies (*e.g.* superfícies muito escuras, vidro, espelhos).

Uma outra abordagem para localização pode ser por meio do uso de odometria. Embora odometria usando *encoders* apresente erro incremental elevado (Chong e Kleeman, 1997; Martinelli, 2002). Uma alternativa que tem recebido grande atenção nos últimos anos é a odometria visual, que tem provido algoritmos cada vez mais eficientes, embora apresentem custo computacional altíssimo, dependam de câmeras de alta resolução e também sofram com os problemas de iluminação.

O estudo de localização interna, usando redes sem fio, tem sido uma área de pesquisa com crescente atenção. O uso de diferentes tecnologias (*e.g.* *Wireless LAN*, *Bluetooth*, *ZigBee*) e diferentes métricas (*e.g.* tempo, potência) tem provido soluções cada vez mais interessantes.

É mais comum o estudo de localização em grandes áreas, como prédios, onde o objetivo é saber em que sala um dispositivo se encontra. Tal abordagem, em geral, utiliza o sinal da rede sem fio utilizado como uma metáfora do modelo de GPS. Além disso, redes sem fio convencionais são dispositivos baratos e que poderiam ser uma fonte eficiente de localização. Por fazer uso de algoritmos relativamente simples, o custo computacional esperado é baixo.

Um dos problemas na utilização de sinais da rede sem fio é sua grande instabilidade (ruído) e problemas devidos a atenuação e reflexão do sinal. Desta forma, o uso de mais pontos de acesso, por aumentar a possibilidade de atenuação ou reflexão, poderia não ter relação direta com a melhoria da acuracidade na obtenção de posição? O mesmo acontece com o ruído, a identificação e remoção de pontos de acesso com mais ruído poderia também auxiliar na acuracidade? Estas perguntas, na forma de hipóteses, foram avaliadas e são respondidas neste capítulo.

O uso de funções matemáticas ou regras de estimação para determinar o posicionamento baseado em um comportamento padrão/estático das leituras geralmente provê resultados não satisfatórios devido a atenuação e reflexão do sinal. Uma técnica que poderia auxiliar na redução da instabilidade e considerar o sinal mais especificamente relacionado com o ambiente seria alguma técnica de aprendizado de máquina.

Elnahrawy *et al.* (2004) apresenta que a obtenção de boas performances em localização usando redes sem fio depende de configurações específicas do ambiente. Isso significa que diferentes abordagens devem ser utilizadas em diferentes ambientes, como usando diferentes tipos de sinais ou filtros. Localização em grandes prédios podem envolver diferentes requisitos, mais do que no caso da localização em pequenas áreas. Essas dificuldades podem ser relacionadas com a atenuação e reflexão do sinal em paredes e objetos móveis bem como outras fontes de interferência. Trabalhos usando localização dentro de prédios podem ser vistos em (Espinace

et al., 2008; Ladd *et al.*, 2004). Outra abordagem para localização é com o uso de dezenas ou centenas de nós provendo sinal, numa configuração conhecida como *fingerprint* (Robles *et al.*, 2010). Uma desvantagem do modelo canônico de *fingerprint* é que, devido ao uso de centenas de sensores, o mesmo pode ser mais caro tanto em termos financeiros como computacionais.

Usualmente, o sinal de uma rede sem fio propaga em uma forma radial. O sinal pode ser direcional dependendo do tipo de antena que está sendo usado. A potência do sinal decresce de acordo com a distância da fonte. Por meio de trilateração e usando pelo menos três fontes, pode ser utilizada uma metáfora do modelo de GPS para obter a posição de um nó e, da mesma forma, estimar outras informações como orientação e velocidade caso o objeto esteja em movimento. Entretanto, ao contrário do GPS, o sinal de rede sem fio apresenta grande instabilidade e sofre mais com interferências (Gast, 2002; Jacobsson *et al.*, 2010; Ross, 2008).

Maisano *et al.* (2008) introduz e descreve aspectos técnicos de um método chamado de *indoor GPS* (iGPS). A abordagem proposta apresenta como leituras com base em luz estruturada podem ser utilizadas para rastrear elementos em uma área delimitada. O sistema tem como princípio básico de funcionamento o cálculo da posição por meio do ângulo relativo (ângulo horizontal – azimute e ângulo vertical – elevação) dos emissores aos receptores. Baseado no conhecimento da posição inicial dos emissores, obtidas na fase de inicialização/calibragem do sistema o sistema pode calcular a posição dos sensores (fixado junto ao objeto a ser localizado). Alguns pontos críticos do sistema são a necessidade de contato visual entre emissores e receptores e uma fase inicial de calibragem relativamente trabalhosa.

Franceschini *et al.* (2009) apresenta um sistema de localização/rastreamento chamado de *Mobile Spatial Coordinate Measuring System* (MScMS) proposto para localização de objetos de larga escala (dimensões na ordem de dezenas de metros). Apresenta uma natureza distribuída, com uma coleção de sensores *Bluetooth*, e faz uso de *Time Difference of Arrival* (TDoA) (Gustafsson e Gunnarsson, 2003) para calcular a posição por meio de uma série de equações. Um dos problemas identificados é a sensibilidade a fatores externos, como temperatura, umidade e presença de obstáculos, que afetam consideravelmente o TDoA. O sistema apresenta uma fase de inicialização onde é necessária a existência de um objeto de medida de referência (*standard reference artifact*) para calibrar as posições. Tal objeto pode ser usado durante a fase operacional do sistema a fim de regular possíveis alterações do TDoA devido a mudanças ambientais. O uso do TDoA depende de dois sinais, um de radiofrequência (RF) e outro de ultrassom (US). É pela diferença no tempo entre os sinais de RF e US que o TDoA é calculado. Este valor não é padrão em redes sem fio convencionais e dependem de um hardware específico.

Nossa abordagem utilizando Wi-Fi usa um ambiente relativamente similar aos propostos por Franceschini *et al.* (2009); Maisano *et al.* (2008); Welch *et al.* (2001) entretanto, nossa abordagem faz uso de robôs móveis e pretende ser uma opção de maior acuracidade e também

de calibragem automática pelo robô móvel. Buscamos avaliar o uso de Wi-Fi pelo fato do mesmo ser mais barato e prático de implementar.

Perkins *et al.* (2011) apresenta uma comparação entre resoluções obtidas com uso de TDoA e o *Received Signal Strength Indication* (RSSI). Utilizando TDoA a resolução obtida ficou em 1,1 cm e utilizando RSSI a resolução obtida ficou em 2,4 cm. Entretanto, Hara e Anzai (2008) apresenta outra comparação entre TDoA e RSSI para localização, onde a conclusão aponta que o RSSI é menos sensível a mudanças do ambiente, sendo mais indicado para ambientes dinâmicos.

Nossas avaliações fazem uso do RSSI pelos seguintes motivos: (i) é um valor disponível em praticamente todas as placas de redes atuais, (ii) não depende de hardware adicional, (iii) apresentou melhores resultados em ambientes dinâmicos em comparação com o TDoA (Hara e Anzai, 2008), (iv) apresentou melhores resultados em relação a outras métricas simples de tempo (Marinello *et al.*, 2012).

Liu *et al.* (2007) apresenta uma coletânea de técnicas e sistemas de posicionamento indoor por meio de redes sem fio; nele, é destacado o fato de que tanto a pesquisa na área como os produtos que estão indo ao mercado ainda precisam de sérias melhorias. Nosso objetivo é, como previamente mencionado, avaliar e propor um método para uso em robôs móveis autônomos sem necessidade de hardwares especiais e que seja prático, de baixo custo e fácil de operar, objetivando a completa autonomia do mesmo.

4.2 Metodologia

O modelo proposto para localização tem como objetivo o aprendizado da posição (x, y) de um agente por meio de uma RNA em que as entradas são o sinal da potência de pontos de acesso. A Figura 4.1 apresenta a configuração proposta. Embora utilizando neste experimento o robô NAO (Aldebaran, 2013), esta metodologia pode ser aplicada em qualquer outro tipo de dispositivo que suporte rede *wireless*; ainda, o código-fonte desenvolvido (disponível em <http://goo.gl/sjgnd>) pode ser utilizado em qualquer sistema GNU/Linux compatível. A posição dos pontos de acesso (Figura 4.1(a)) é aproximada, sendo que tais posições poderiam ser diferentes. Além disso, não é necessário possuir conhecimento das posições *a priori*. É importante ressaltar que nossa metodologia envolve a investigação em uma área pequena (delimitada em uma sala), com todos os pontos de acesso na mesma sala e com distância de no máximo 20 metros entre o robô e os pontos de acesso. Isso tem nos permitido obter um valor final bastante acurado.

Nestas avaliações, o sinal obtido da rede sem fio é o *Received Signal Strength Indication* (RSSI). Este valor é obtido por meio do comando do GNU/Linux *iwlist* (usado como *iwlist*

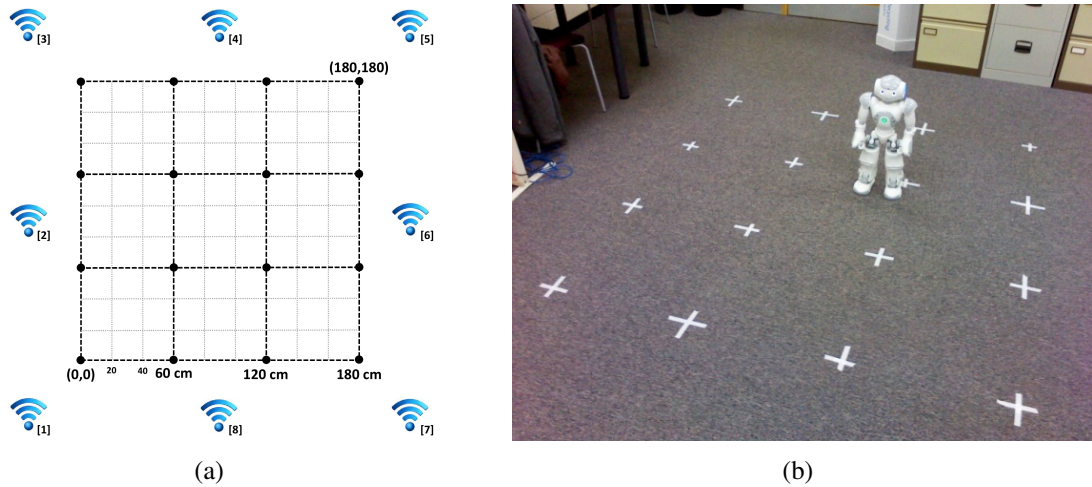


Figura 4.1: (a) Representação gráfica da área de trabalho considerando uma área de 180 cm x 180 cm. (b) Foto parcial do ambiente com o robô, similar a representação gráfica da Figura 4.1(a). Cada pequena cruz está posicionada a 60 cm uma da outra. Em (a), a posição dos pontos de acesso é aproximada.

<interface> scanning). Como usamos o comando *iwlist*, não existe a necessidade de conexão ou mesmo qualquer tipo de *login* nos diferentes pontos de acesso. A leitura do sinal das redes com independência de conexão provê informação suficiente para o sistema e torna o sistema mais fácil de usar, leve e flexível. RSSI é uma métrica de potência presente em sinais recebidos de fontes de rádio. Praticamente qualquer dispositivo de rádio implementa em seu circuito a possibilidade de entrega deste valor sem necessidade de desenvolvimento ou adaptação adicional (Gast, 2002; Ross, 2008). Devido às ferramentas escolhidas, o tempo de obtenção de cada leitura das redes (e conseqüente obtenção do RSSI) é de cerca de um segundo. Assim, podemos pensar no tamanho da janela aproximadamente como sendo o tempo necessário para obtenção da devida janela.

Salvo algumas exceções a serem explicadas posteriormente, as redes neurais são treinadas com o SNNS (Zell *et al.*, 1995) usando dados coletados pelo robô. O treino e teste se dá utilizando *k-fold cross-validation* com $k = 3$. Um *script* para encontrar ponto ótimo de generalização foi utilizado e está disponível em <http://goo.gl/sjgnd>. A coleta de dados se dá utilizando o robô dentro do plano apresentado (Figura 4.1(b)) e coletando 3 minutos de leituras (i.e. ≈ 180 leituras) em cada ponto marcado. Com um deslocamento de 60 cm, mapeando um plano de 180 cm x 180 cm, isso significa 16 pontos para ler resultando num total de ≈ 2880 leituras.

Nossa abordagem se baseia na capacidade de aprendizado e generalização de Redes Neurais Artificiais a fim de reduzir o efeito de instabilidade nas entradas. Entretanto, os valores obtidos dos pontos de acesso são muito ruidosos, o que, em avaliações iniciais, apresentou que o apren-

dizado da rede obteve baixa acuracidade. Isso motivou o estudo e avaliação de filtros. Uma avaliação inicial de filtros de média móvel e mediana móvel realizada por Pessin *et al.* (2011a) apresentou que o filtro de mediana, com mesmo tamanho de janela do filtro de média, permitia um aprendizado um pouco mais eficiente da rede neural. Acredita-se que isso seja devido a capacidade da mediana de ignorar *outliers*. O comportamento do filtro pode ser entendido com a Figura 4.2(a), onde duas linhas representam (em azul) a leitura do valor bruto de um ponto de acesso e (em preto) o resultado do filtro de mediana móvel com janela de 8 leituras.

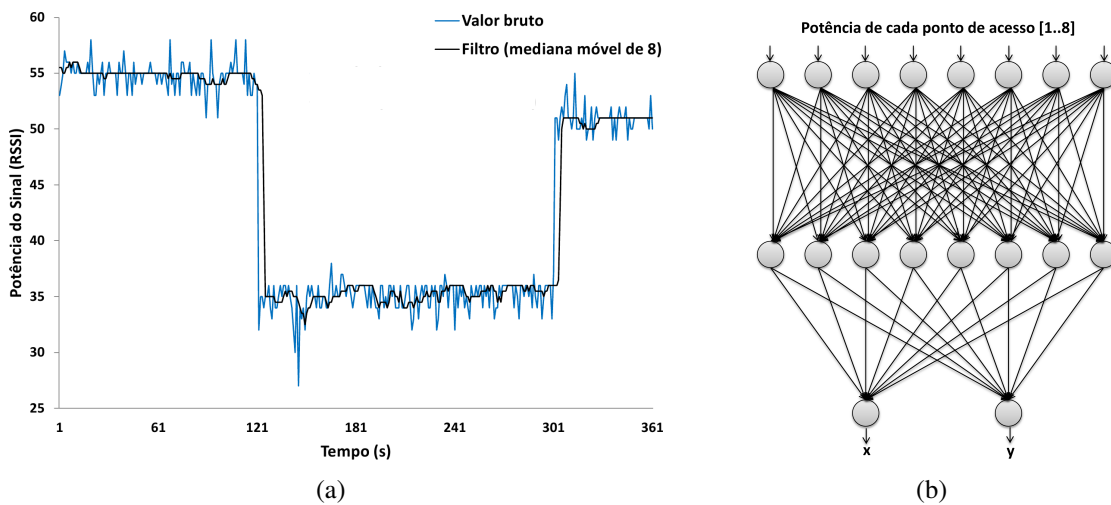


Figura 4.2: (a) Exemplo de comportamento do filtro de mediana móvel. A linha azul apresenta o valor bruto da leitura de um ponto de acesso. A linha preta apresenta como o filtro de mediana móvel remove parte dos ruídos (janela de 8 leituras). (b) Exemplo de topologia de rede neural utilizada.

O número de neurônios na camada de entrada é equivalente ao número de pontos de acesso disponíveis. Assim, como utilizamos 8 pontos de acesso, existem 8 neurônios de entrada na rede neural, como mostra a Figura 4.2(b). No nosso modelo, a ordem de entrada é importante, assim, o ponto de acesso um é ligado ao neurônio um, o ponto de acesso dois ligado ao neurônio dois e assim por diante. Esse controle é feito por meio do endereço físico (MAC) dos pontos de acesso. Tal informação provém do escaneamento dos pontos de acesso.

As saídas da rede são dois valores, as coordenadas (x, y) da posição do robô. O erro é medido por meio da distância euclidiana. O valor d é o erro (distância, em centímetros), (x_1, y_1) são os valores esperados da base de validação da rede neural e (x_2, y_2) são os valores obtidos enquanto usando a rede neural. Um exemplo de topologia pode ser visto na Figura 4.2(b).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4.1)$$

Aspectos pontuais de metodologia serão descritos no início das seções de investigação, detalhadas na sequência.

4.3 Resultados

Essa seção descreve os resultados da investigação do uso de redes sem fio e aprendizado de máquina para auxiliar na tarefa de localização interna de robôs móveis. Investigações iniciais foram apresentadas em Pessin *et al.* (2011a), Pessin *et al.* (2011b) e Pessin *et al.* (2012). Esta seção faz uso de alguns conceitos obtidos nos trabalhos supracitados porém estende os mesmos a fim de apresentar uma investigação mais aprofundada.

A metodologia de avaliação dos resultados faz uso de métodos estatísticos por meio do valor-p^{5,6}. Embora não sendo de consenso geral, o uso de tal método auxilia na interpretação dos resultados, ao menos de forma parcial. É de consenso geral que, mais importante que a *significância estatística* é a *quantidade* da diferença entre os resultados. Dessa forma, a maioria das análises estatísticas tem como objetivo mostrar quais conjuntos são similares ou não entre si. O que não deve refletir diretamente na qualidade do resultado.

Usamos o método de *Shapiro-Wilk* para verificar a normalidade das distribuições. Caso as distribuições tenham a hipótese de normalidade rejeitada, são usados métodos não paramétricos de avaliação, como os métodos de *Wilcoxon-Mann-Whitney* ou de *Kruskal-Wallis*. Caso as distribuições não tenham a hipótese de normalidade rejeitada, são usados testes paramétricos como teste-t ou ANOVA.

Esta seção está dividida em 7 subseções. São 5 subseções descrevendo investigações, com avaliações e resultados e duas subseções propondo mudanças na arquitetura proposta. Na parte de investigações, são descritas: (i) avaliações sobre o efeito do uso de filtros; (ii) avaliações sobre o efeito do número de pontos de acesso; (iii) avaliações sobre efeito da mudança da topologia; (iv) avaliações sobre evolução das topologias. Na parte de mudança de arquitetura, são descritas propostas para tornar o sistema (i) tolerante a falhas e (ii) completamente autônomo.

4.3.1 Investigação sobre o efeito do uso de filtros

A investigação inicial realizada por Pessin *et al.* (2011a) apresentou que o uso de mediana é ligeiramente melhor que o filtro de média para os dados analisados, por permitir que a rede

⁵Na estatística clássica, o valor-p é uma estatística utilizada para sintetizar o resultado de um teste de hipóteses. Formalmente, o valor-p é definido como a probabilidade de se obter uma estatística de teste igual ou mais extrema quanto aquela observada em uma amostra, assumindo verdadeira a hipótese nula (Wikipédia:Valor-p, 2013).

⁶É comum, na descrição do valor-p, que o mesmo seja truncado na terceira casa decimal. Usamos esta padronização neste documento.

neural apresentasse maior acuracidade no aprendizado. Ainda, das diversas topologias de redes neurais avaliadas (camadas ocultas com 4, 4x4, 8, 8x8, 12, 16, 20, 24, 28 e 32 neurônios), todas as topologias apresentaram erro médio, bem como desvio padrão relativamente similares.

Dessa forma, iniciamos novas investigações considerando conceitos e valores obtidos previamente para uma nova avaliação do filtro de mediana de forma mais detalhada. Avaliamos as seguintes janelas no filtro de mediana móvel: {1, 2, 4, 6, 8, 10, 30, 60}. Neste caso, o que chamamos de mediana de 1 é na verdade o valor bruto (*raw*), e, o que chamamos de mediana de 2 pode ser pensado como média. A quantidade de neurônios utilizados na camada intermediária é doze; valor arbitrário baseado no trabalho de Pessin *et al.* (2011a).

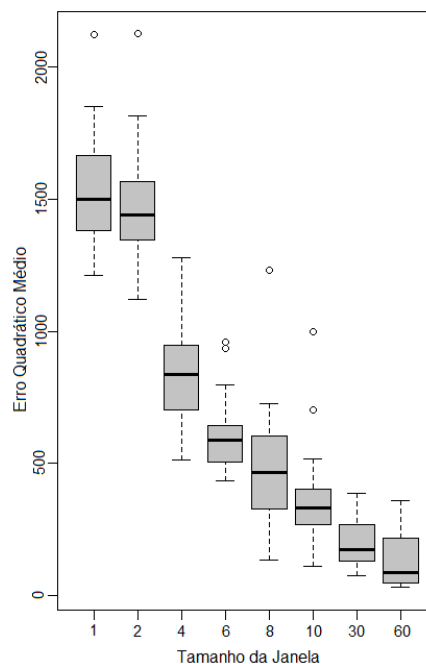


Figura 4.3: Resultados usando o filtro de mediana móvel para diferentes tamanhos de janelas ($n = 30$). Podemos ver que quanto maior a janela de tempo, menor o erro no aprendizado.

A Figura 4.3 apresenta resultados usando o filtro de mediana móvel para diferentes tamanhos de janelas. São apresentados resultados de 30 execuções de cada rede neural. Podemos ver que quanto maior a janela, menor o erro no aprendizado. A desvantagem no uso do filtro se dá pelo motivo do mesmo ser dependente da janela de tempo, que, na prática aplica algo como um atraso na obtenção do valor da posição.

A fim de realizar testes estatísticos (comparação e correlação) sobre o conjunto dos resultados da Figura 4.3, avaliamos inicialmente a adequação dos conjuntos a distribuições normais. Para tanto, utilizamos o teste de *Shapiro-Wilk*. Os valores-p obtidos para M1, M2, M4, M6, M8, M10, M30 e M60 foram, respectivamente, {0,242, 0,178, 0,490, 0,012, 0,009, 0,000,

0,152, 0,002}. Assim, utilizando 95% de confiança, metade dos conjuntos tem a hipótese de normalidade rejeitada e metade não (usamos valor limite de 0,05).

Utilizando os testes adequados para comparação entre conjuntos, pudemos ver que, para todos os conjuntos (com exceção de M1 e M2) a hipótese de rejeição de igualdade é aceita. Ainda, sobre os resultados apresentados na Figura 4.3, o valor do teste de correlação (*Spearman's rank correlation test*⁷) apresenta ρ de -0,929. Ou seja, uma correlação negativa forte.

A Figura 4.4 e a Tabela 4.1 apresentam resultados do uso da melhor RNA de cada janela de tempo. Em ambos, o erro apresentado está em centímetros e o tamanho da amostra é de ≈ 900 . Podemos ver na Figura 4.4 que, quanto maior a janela de tempo, menor o erro. O histograma apresenta cada vez mais os dados concentrados em zero (menor cauda).

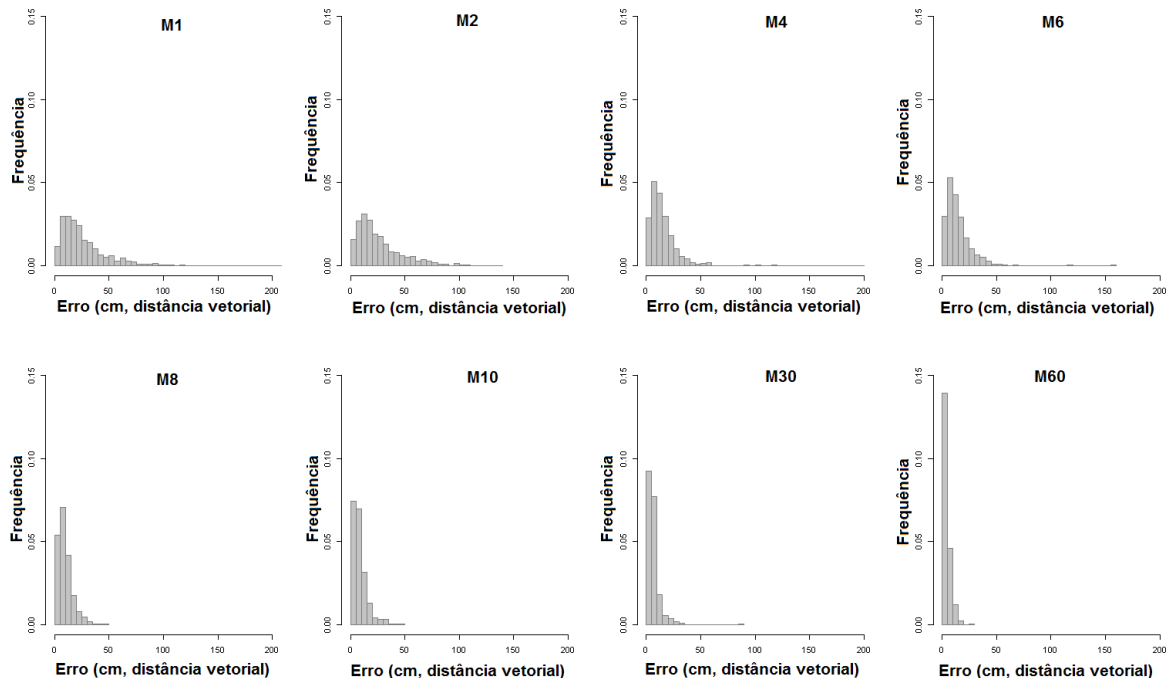


Figura 4.4: Histogramas dos resultado do uso da melhor RNA de cada janela de tempo ($n \approx 900$).

A Figura 4.5 apresenta uma secção do plano (Figura 4.1) com valores esperados e obtidos usando diferentes janelas de tempo no filtro de mediana móvel. São apresentados os valores usando o valor bruto, usando filtro de mediana de 10 leituras e usando filtro de mediana de 60 leituras. Podemos ver o resultado dos erros no plano de trabalho proposto. Assim como apresentados na Figura 4.4 e na Tabela 4.1, podemos ver que o filtro permite que a rede neural tenha um aprendizado cada vez mais acurado.

⁷Teste de correlação para conjuntos não paramétricos.

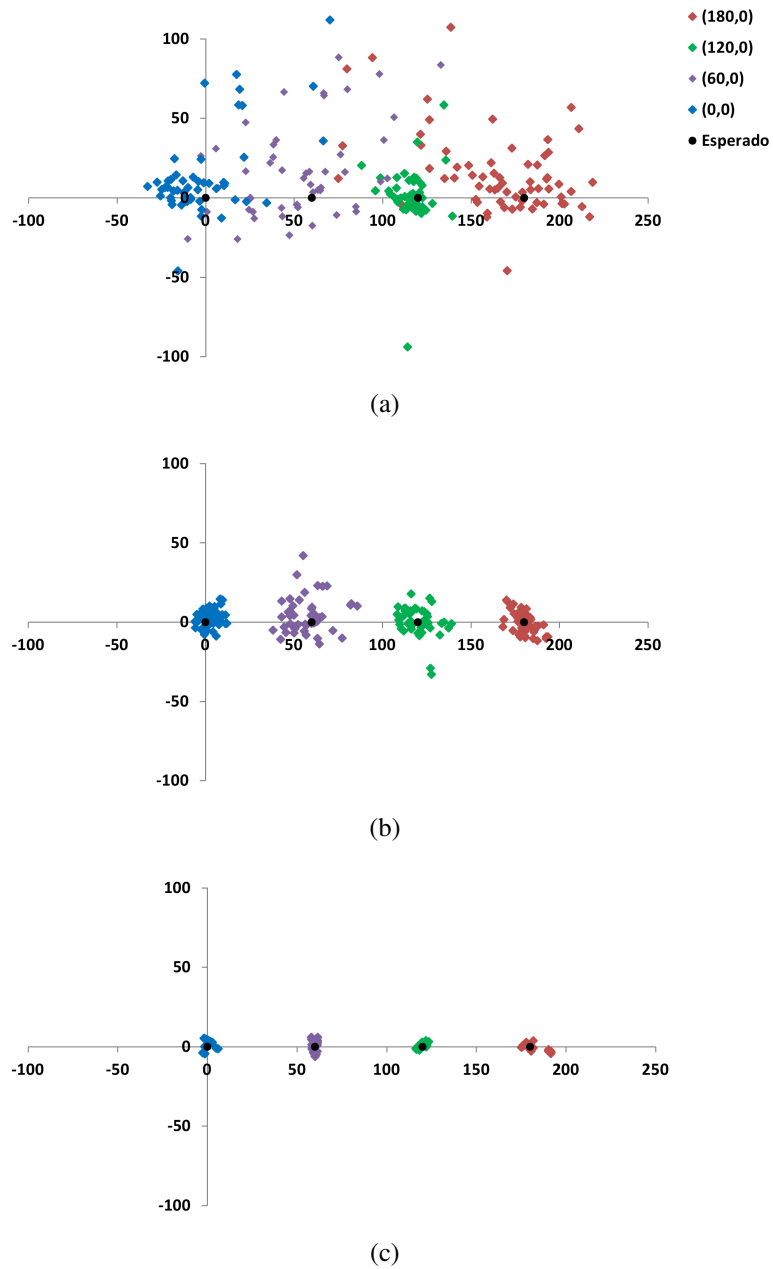


Figura 4.5: Secção do plano (Figura 4.1) com valores esperados e obtidos usando diferentes janelas de tempo na mediana. (a) Valor bruto. (b) Usando filtro de mediana de 10 leituras. (c) Usando filtro de mediana de 60 leituras. Os eixos x e y estão em centímetros.

Tabela 4.1: Sumário dos resultados do uso da melhor RNA de cada tamanho de janela ($n \approx 900$). O erro é apresentado em centímetros e representa a distância vetorial entre a posição esperada e a posição obtida.).

| Tamanho da janela | Média | Desvio | Mediana | Mínimo | Máximo | Menor que 10 cm |
|-------------------|-------|--------|---------|--------|--------|-----------------|
| 1 | 26,47 | 22,64 | 20,40 | 0,74 | 215,23 | 20,7% |
| 2 | 26,10 | 20,99 | 19,45 | 0,58 | 137,78 | 21,4% |
| 4 | 15,76 | 16,25 | 12,23 | 0,83 | 198,76 | 39,8% |
| 6 | 15,07 | 14,34 | 11,87 | 0,39 | 158,36 | 41,3% |
| 8 | 9,55 | 6,67 | 7,89 | 0,17 | 45,27 | 62,4% |
| 10 | 8,27 | 6,49 | 6,44 | 0,15 | 47,07 | 72,0% |
| 30 | 6,52 | 5,76 | 5,23 | 0,28 | 87,35 | 85,0% |
| 60 | 4,49 | 3,23 | 3,70 | 0,37 | 28,17 | 93,0% |

A contribuição desta seção foi apresentar como o erro no aprendizado da rede neural se comporta com o uso do filtro de mediada móvel para diferentes tamanhos de janelas. Pudemos ver que não existe rejeição da hipótese de igualdade entre o uso do valor bruto ou da mediana (média) de duas leituras. Entretanto, para as demais janelas de tempo, o aumento na acuracidade do aprendizado da rede neural é bastante promissor. As figuras mostram que, a partir de uma janela de 8 leituras, o erro mínimo não diminui muito, entretanto as medidas dos erros continuam caindo.

Uma janela de tempo apresenta um dilema na questão acuracidade x tempo de obtenção da posição. Assim, consideramos a janela de 8 leituras como base para os próximos experimentos. Esta escolha se dá de forma arbitrária, entretanto, é a que aparenta ser a mais promissora em termos de aplicação prática.

4.3.2 Investigação sobre o número de pontos de acesso

A segunda etapa de avaliação envolve a investigação do aprendizado com diferentes quantidades de pontos de acesso. Utilizamos 3, 4, 6 e 8 pontos de acesso. A escolha da remoção dos pontos se deu pela amplitude de erro nas leituras. A Figura 4.6 apresenta a amplitude do RSSI de cada ponto de acesso utilizado. Interpreta-se a figura como sendo: maior a amplitude, maior o ruído do dispositivo.

Buscamos avaliar o treinamento das redes considerando as 8 entradas originais e também com menos entradas, i.e. removendo as entradas mais ruidosas. Podemos ver na Figura 4.6 que, a ordem de amplitude é, do menos estável para o mais estável: {AP6, AP5, AP4, AP3, AP8, AP1, AP2, AP7}. Dessa forma, para a avaliação com seis entradas, removemos os pontos AP6 e AP5 (os dois com maior amplitude). A mesma ideia para 4 e 3 entradas.

A Figura 4.7 apresenta resultados de 30 execuções de cada rede neural considerando diferentes quantidades de entradas (Q_i). Sobre os dados apresentados na Figura 4.7, o teste de

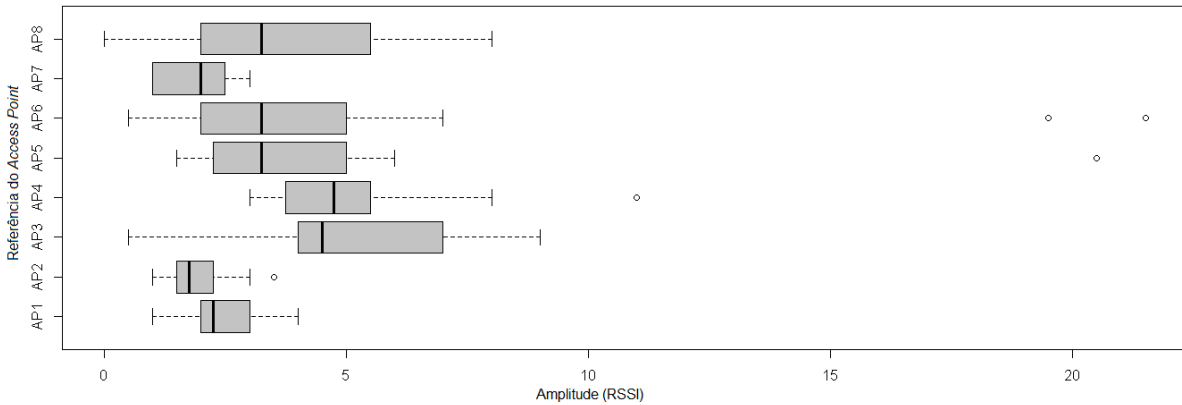


Figura 4.6: Amplitude do sinal lido de cada antena (AP1 até AP8).

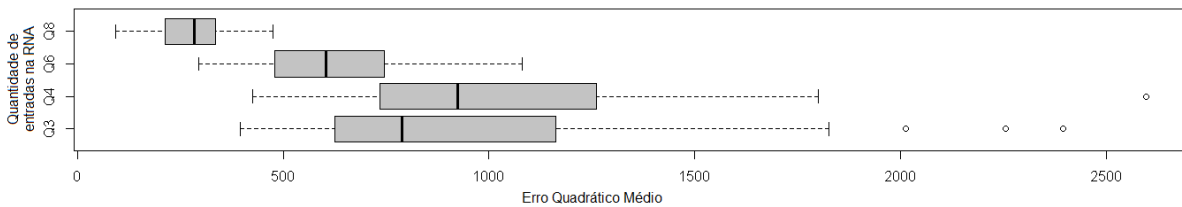


Figura 4.7: Erro quadrático médio resultante de 30 execuções de cada rede neural com diferentes quantidades de entradas.

adequação a normalidade (Shapiro-Wilk normality test) apresentou os seguintes valores-p para Q3, Q4, Q6 e Q8: {0,000, 0,002, 0,830, 0,987}. Desta forma, usamos testes paramétricos entre Q6 e Q8 e não paramétricos entre os outros conjuntos. A Tabela 4.2 apresenta o resultado dos testes de comparação. Podemos ver que, com 95% de confiança, com exceção das distribuições Q3 e Q4, as demais apresentam rejeição na hipótese de igualdade.

Tabela 4.2: Valores-p resultantes de teste de comparação de distribuições. Com exceção das distribuições Q3 e Q4, as demais apresentam rejeição na hipótese de igualdade.

| | Q3 | Q4 | Q6 |
|----|--------------|-------|-------|
| Q4 | 0,260 | | |
| Q6 | 0,000 | 0,000 | |
| Q8 | 0,000 | 0,000 | 0,000 |

A Figura 4.8 apresenta os histogramas dos erros no aprendizado das redes removendo APs com maior amplitude. Podemos ver que o histograma de Q8 é o que apresenta a menor cauda, tendo os resultados mais concentrados próximos de zero. Em relação a Q4, a distribuição Q6 também apresenta uma menor cauda.

A contribuição desta seção foi mostrar que, mesmo com pontos de acesso com amplitude mais alta, a remoção destes não auxiliou no aprendizado. É importante ressaltar que os pontos

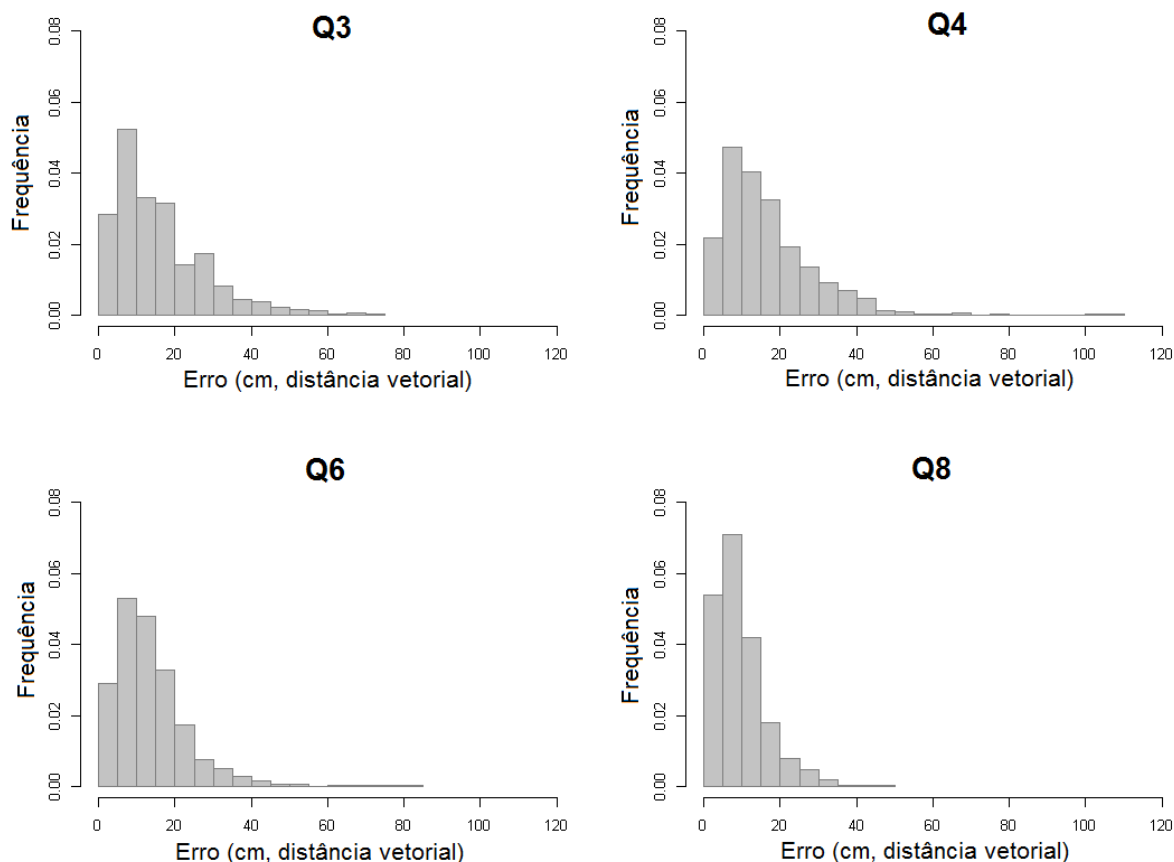


Figura 4.8: Histogramas dos resultados do uso da melhor rede neural considerando diferentes entradas ($n \approx 900$).

Tabela 4.3: Sumário dos resultados do uso da melhor RNA de cada conjunto de entradas ($n \approx 900$). Erro em centímetros apresentado como a distância vetorial entre a posição esperada e a posição obtida.

| | Média | Desvio | Mediana | Mínimo | Máximo | Menor que 10 cm |
|----|-------|--------|---------|--------|--------|-----------------|
| Q3 | 15,81 | 12,08 | 12,52 | 0,71 | 71,03 | 40,4% |
| Q4 | 16,60 | 12,30 | 14,06 | 0,72 | 109,12 | 34,6% |
| Q6 | 13,83 | 10,18 | 11,63 | 0,01 | 83,95 | 40,9% |
| Q8 | 9,55 | 6,67 | 7,89 | 0,17 | 45,27 | 62,4% |

de acesso utilizados não apresentam amplitudes muito grandes. As amplitudes na faixa de 10 decibéis são normais para o problema tratado (e o filtro de mediana remove os *outliers*).

4.3.3 Efeito da mudança da topologia

O número de neurônios na camada oculta de uma rede neural tem relação com a sua capacidade de aprendizado e generalização. Poucos neurônios podem conduzir ao chamado *underfit-*

ting (deficiência no aprendizado) enquanto o excesso de neurônios pode conduzir ao *overfitting* (aprendizado excessivo com perda de capacidade de generalização – a.k.a. decorar os dados) além de desperdício de recursos computacionais (Huang, 2003; Teoh *et al.*, 2006). Assim, após obter os resultados sobre efeito de filtro e efeito do número de entradas, voltamos a avaliar topologias na camada intermediária das redes neurais.

Iniciamos as avaliações considerando 10 diferentes configurações de camadas intermediárias. A Tabela 4.4 e a Figura 4.9 apresentam os resultados de 30 execuções de cada configuração. O erro apresentado é o MSE. Ainda, é apresentado o valor-p do teste Shapiro-Wilk a fim de verificar a adequação a normalidade dos conjuntos. Podemos ver que, com exceção da rede de 12 neurônios na camada oculta, todas as outras apresentam valor-p inferior a 0,05, ou seja, a hipótese de normalidade é rejeitada utilizando 95% de confiança.

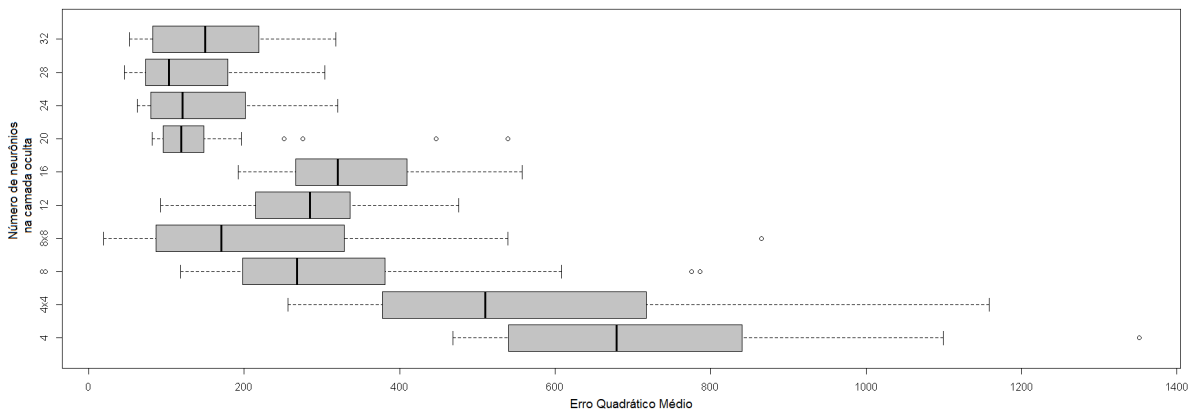


Figura 4.9: Erro quadrático médio resultante das avaliações das diferentes topologias de redes neurais ($n = 30$).

Tabela 4.4: Erro quadrático médio resultante das avaliações das diferentes topologias de redes neurais ($n = 30$).

| Neurônios na camada oculta | Média | Desvio | Mediana | Mínimo | Máximo | Valor-p (Shapiro) |
|----------------------------|--------|--------|---------|--------|---------|-------------------|
| 4 | 718,30 | 207,21 | 679,50 | 468,70 | 1351,80 | 0,017 |
| 4x4 | 586,20 | 264,15 | 509,90 | 256,60 | 1158,70 | 0,002 |
| 8 | 314,10 | 173,76 | 267,60 | 118,40 | 786,80 | 0,000 |
| 8x8 | 222,27 | 185,16 | 170,17 | 19,30 | 865,54 | 0,001 |
| 12 | 276,04 | 92,18 | 284,90 | 92,68 | 476,27 | 0,987 |
| 16 | 341,30 | 100,99 | 320,10 | 192,60 | 557,60 | 0,026 |
| 20 | 151,79 | 103,83 | 118,87 | 81,32 | 539,12 | 0,000 |
| 24 | 147,81 | 84,31 | 120,66 | 62,12 | 320,35 | 0,000 |
| 28 | 127,73 | 65,51 | 102,73 | 45,72 | 304,13 | 0,012 |
| 32 | 156,34 | 80,42 | 149,63 | 52,48 | 317,48 | 0,008 |

A comparação entre os conjuntos pode ser vista na Tabela 4.5. A tabela apresenta os valores-p resultantes do teste de Wilcoxon para as topologias avaliadas. Em negrito, são apre-

sentados os conjuntos com valor-p maior que 0,05 (igualdade não rejeitada considerando 95% de confiança). Podemos ver que, para os dados analisados, as redes com 8x8, 20, 24, 28 e 32 neurônios na camada oculta não apresentam rejeição na hipótese de igualdade entre si.

Tabela 4.5: Valores-p resultantes do teste de Wilcoxon para as topologias avaliadas. Em negrito, conjuntos com valor-p maior que 0,05 (igualdade não rejeitada considerando 95% de confiança).

| | 4 | 4x4 | 8 | 8x8 | 12 | 16 | 20 | 24 | 28 |
|-----|-------|-------|--------------|--------------|-------|-------|--------------|--------------|--------------|
| 4x4 | 0,012 | | | | | | | | |
| 8 | 0,000 | 0,000 | | | | | | | |
| 8x8 | 0,000 | 0,000 | 0,016 | | | | | | |
| 12 | 0,000 | 0,000 | 0,947 | 0,026 | | | | | |
| 16 | 0,000 | 0,000 | 0,072 | 0,001 | 0,028 | | | | |
| 20 | 0,000 | 0,000 | 0,000 | 0,242 | 0,000 | 0,000 | | | |
| 24 | 0,000 | 0,000 | 0,000 | 0,168 | 0,000 | 0,000 | 0,592 | | |
| 28 | 0,000 | 0,000 | 0,000 | 0,065 | 0,000 | 0,000 | 0,248 | 0,415 | |
| 32 | 0,000 | 0,000 | 0,000 | 0,366 | 0,000 | 0,000 | 0,994 | 0,730 | 0,173 |

As redes com 8x8, 20, 24, 28 e 32 neurônios na camada oculta apresentaram os menores erros no aprendizado (como apresenta a Tabela 4.5). Além disso, essas redes apresentam similaridade entre si quando considerando o teste de Wilcoxon e 95% de confiança. Entretanto, tais avaliações de topologias, de forma manual, tem se apresentado um trabalho árduo e com falta de garantias de qualidade no resultado. Tal falta de garantia, vinculada com a grande demanda de trabalho manual para a avaliação das redes, motivou o desenvolvimento e a avaliação de um método evolutivo para descoberta de topologias de forma automática. Tal método será explicado na próxima seção.

4.3.4 Evolução de topologias de RNA para Localização

Diversos trabalhos realizados no Laboratório de Robótica Móvel (LRM/USP, 2013) apresentam o uso e a avaliação de Redes Neurais Artificiais aplicadas nas diversas áreas da robótica móvel. Entre estes trabalhos podemos citar trabalhos sobre mapeamento (Hata *et al.*, 2009a), (Hata *et al.*, 2009b), trabalhos sobre localização (Pessin *et al.*, 2011a), (Pessin *et al.*, 2011b), e trabalhos sobre navegação (Souza *et al.*, 2012a), (Souza *et al.*, 2013). A avaliação de diferentes redes neurais se faz importante pelo fato de que o aprendizado é relacionado à topologia da rede. Nem toda rede neural apresenta capacidade de aprendizado e nem toda rede neural grande é melhor que uma rede menor. Entretanto, escolher e avaliar topologias de forma manual tem se mostrado um trabalho relativamente árduo e que consome tempo considerável.

Isso motivou o desenvolvimento e a investigação de um algoritmo genético para busca (evolução) de topologias. O objetivo principal é investigar o comportamento do AG na evolução de topologias e compará-lo com as topologias previamente obtidas pelo processo manual. Embora

sendo um processo estocástico, o AG desenvolvido deveria ser capaz de evoluir redes neurais com topologias estatisticamente similares.

A ideia básica deste AG pode ser entendida com a Figura 4.10. Neste AG evoluímos topologia da RNA, ou seja, o número de neurônios e o número de camadas ocultas. Os parâmetros padrão para evolução são de 1 a 3 camadas de neurônios e de 0 a 50 neurônios em cada camada. Desta forma, o cromossomo é representado por um vetor de três posições. Tal cromossomo simples tem como objetivo tornar o espaço de busca menor. Porém, uma inicialização randômica nesta população dificilmente geraria redes neurais com menos de 3 camadas (dada que a probabilidade de um número ser sorteada é de $n/50$). Para ter uma melhor dispersão na inicialização da população, a mesma é realizada usando a seguinte heurística: (i) um terço da população é inicializado com uma camada, ou seja, um número de 1 a 50 na primeira camada e zero nas outras; (ii) outro um terço da população é inicializado com duas camadas, ou seja, números entre 1 e 50 nas duas primeira camadas e zero na terceira; (iii) o um terço restante é inicializado com 3 camadas.

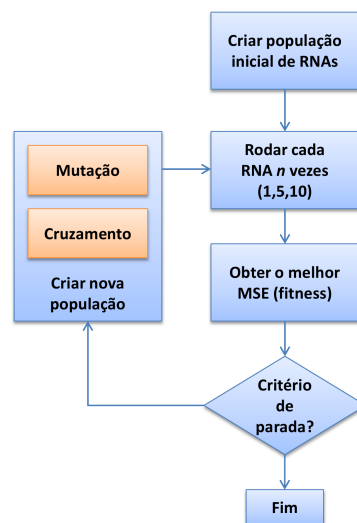


Figura 4.10: Fluxograma do AG responsável pela evolução da topologia da RNA.

Quanto as configurações do AG, utilizamos 15 indivíduos e 20 gerações. Para uma melhor dispersão na inicialização, além da heurística de 1/3 para cada camada, também fizemos a inicialização da rede com múltiplos de 5. Quanto ao cromossomo, um vetor com {10, 5, 25} produzira uma rede neural com 3 camadas ocultas na respectiva ordem do vetor. E, um vetor com {10, 0, 5} produzira uma rede neural com 2 camadas, desconsiderando o zero.

O AG é responsável pela topologia da RNA sendo o treino da rede (atualização dos pesos) realizado pelo SNNS (Zell *et al.*, 1995). Assim, para cada indivíduo – que é a representação de uma topologia – o processo de obtenção do *fitness* depende de chamadas ao SNNS. O *fitness*

que o AG usa como parâmetro de evolução é o MSE da base de validação das redes treinadas. Inicialmente, utilizamos no SNNS o algoritmo de aprendizado *rprop* por 20.000 ciclos e fazemos validação a cada 200 ciclos. Percorremos os valores de validação procurando pelo ponto ótimo de generalização.

O critério de parada do AG é o número de gerações. O esquema de mutação desenvolvido é o modelo gaussiano, onde uma mudança no número de neurônios ocorre de acordo com uma distribuição normal com média no valor do gene (quantidade de neurônios) e um desvio padrão baixo. O que faz com que novos valores caiam na vizinhança do valor, evitando mudanças muito bruscas. A probabilidade de mutação é definida em 20 por cento.

O esquema de cruzamento (*crossover*) é feito em três diferentes formas: com probabilidade randômica, um indivíduo pode trocar genes com o (i) melhor de todos, (ii) com o segundo melhor ou (iii) pode fazer a média entre seus neurônios e os neurônios do melhor indivíduo. O cruzamento ocorre em todos os indivíduos, com exceção dos dois melhores de cada geração, onde usamos o conceito de elitismo – mantendo os dois melhores indivíduos da população inalterados.

Considerando a existência de 3 camadas ocultas com até 50 neurônios em cada camada, poderíamos ter 50^3 (125.000) diferentes topologias. Usando o AG podemos reduzir o espaço de busca, neste caso, 15 indivíduos e 20 gerações resulta em 15×20 comparações, um total de 300 avaliações.

A questão que fica em aberto aqui é: quantas vezes deve ser necessário executar a rede neural a fim de minimizar a influência dos números aleatórios da inicialização dos pesos? Para minimizar a influência da inicialização, avaliamos três tipos de evolução: (i) AG evoluindo baseado em apenas uma execução de cada RNA; (ii) AG evoluindo baseado em 5 execuções de cada RNA – o *fitness* então não é mais o MSE da rede, mas sim a média das 5 execuções; (iii) AG evoluindo baseado em 10 execuções de cada rede – o *fitness* então passa a ser a média de 10 execuções.

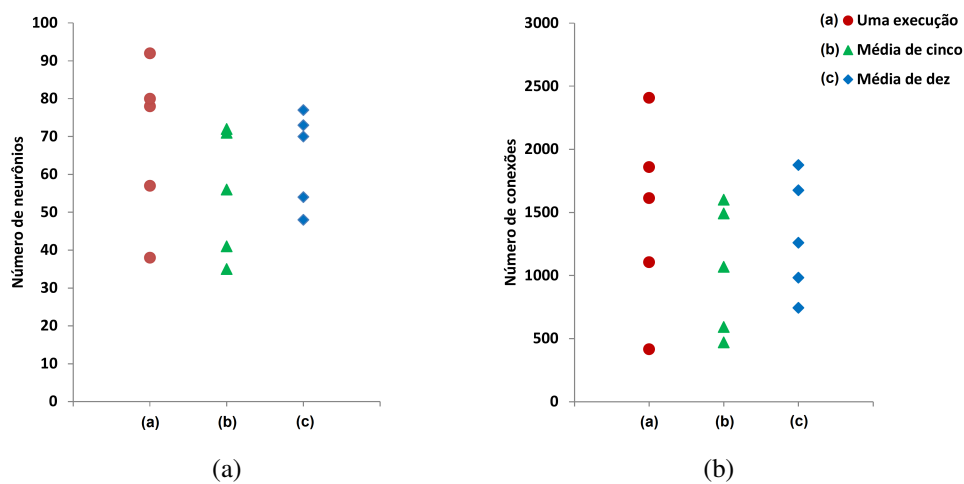
O uso de média de cinco ou dez execuções da rede dá mais importância para redes com menor dispersão nos resultados, diminuindo a influência da aleatoriedade nos resultados. Enquanto o uso de apenas uma execução dá mais importância a uma única rede boa, mas que pode apresentar alta dispersão nos resultados quando treinada novamente com outra semente de inicialização de pesos. Buscamos avaliar dois aspectos-chaves nessa avaliação: (i) quantas vezes devemos rodar cada ANN a fim de ter resultados mais homogêneos; e (ii) quantos ciclos de treino e teste são necessários para obter resultados satisfatórios.

As topologias obtidas pela evolução podem ser vistas na Tabela 4.6. Podemos ver que os resultados de uma única execução da rede teve resultados com redes de 2 e 3 camadas. En-

Tabela 4.6: Topologias obtidas pela evolução. Resultado de 5 rodadas com cada conjunto de configurações.

| | | Topologia obtida |
|---|------------------|------------------|
| AG usando <i>fitness</i> como uma única execução da RNA | E _{1a} | 32 x 25 |
| | E _{1b} | 50 x 35 x 07 |
| | E _{1c} | 13 x 30 x 35 |
| | E _{1d} | 30 x 40 x 10 |
| | E _{1e} | 10 x 28 |
| AG usando <i>fitness</i> como média de 5 execuções | E _{5a} | 47 x 25 |
| | E _{5b} | 50 x 21 |
| | E _{5c} | 25 x 10 |
| | E _{5d} | 30 x 11 |
| | E _{5e} | 33 x 23 |
| AG usando <i>fitness</i> como média de 10 execuções | E _{10a} | 18 x 30 |
| | E _{10b} | 35 x 19 |
| | E _{10c} | 41 x 36 |
| | E _{10d} | 20 x 50 |
| | E _{10e} | 45 x 28 |

quanto o resultado de média de cinco e dez apresentou resultados apenas com duas camadas. A Figura 4.11 apresenta o gráfico de pontos do número de neurônios e de conexões resultantes da evolução. Podemos ver que em ambos os valores, a evolução baseada em apenas uma execução da rede neural apresentou a maior dispersão nos resultados.

**Figura 4.11:** Topologias obtidas pela evolução. (a) Gráfico de pontos do número de neurônios. (b) Gráfico de pontos do número de conexões. Podemos ver que em ambos os valores, a evolução baseada em apenas uma execução da rede neural apresentou a maior dispersão nos resultados.

Buscando uma comparação estatisticamente válida sobre os resultados da evolução, executamos cada rede neural 30 vezes. O resultado pode ser visto na Figura 4.12. Podemos ver que

os resultados das redes neurais que foram evoluídas com base em uma única execução são os que apresentam maior dispersão. A Tabela 4.7 apresenta o sumário numérico da Figura 4.12.

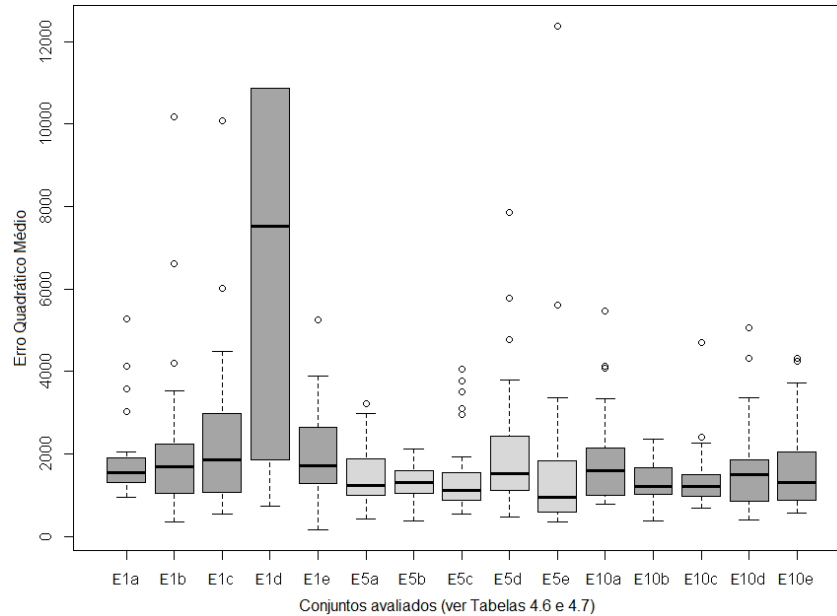


Figura 4.12: Erro quadrático médio resultante das avaliações com as RNAs obtida pela evolução – redes apresentadas na Tabela 4.6.

A análise estatística (*Kruskal-Wallis rank sum test*) sobre o conjunto de dados de 5 e 10 (Figura 4.12) apresenta p-valor de 0,075 ($>0,05$). Ou seja, os conjuntos 5 e 10 não apresentam diferenças significativas usando 95% de confiança. Mas, considerando o conjunto dos resultados com uma execução, o valor-p resultante do teste (*Kruskal-Wallis rank sum test*) é de $1,838e^{-09}$ ($<0,05$). Ou seja, existem diferenças significativas entre os conjuntos de 1, 5 e 10 execuções, mas não entre 5 e 10 execuções.

Os resultados usando média de 5 e 10 execuções que apresentaram as redes neurais com menor dispersão e menores médias e medianas são E_{5b} e E_{10b} . Essas redes têm respectivamente 50×21 e 35×19 neurônios e 1492 e 983 conexões cada (considerando oito entradas e duas saídas). Como descrito anteriormente, tais redes tiveram suas topologias evoluídas usando *rprop* por 20.000 ciclos e sendo validada a cada 200 ciclos. A questão que gostaríamos de responder agora é: faz diferença aumentar ou diminuir estes números? Para isso, alteramos o treino da rede para os seguintes ciclos: {2k, 5k, 10k, 20k e 100k}. E o número de ciclos de validação para {20, 50, 200, 500}. Selecionamos então a rede E_{10b} para o próximo passo de avaliação devido ao fato de possuir menos neurônios e conexões que a rede E_{5b} .

Executamos 30 vezes a RNA com a nova combinação de ciclos de treino/validação. Os resultados podem ser vistos na Figura 4.13. Podemos ver que o resultado utilizando 2.000

Tabela 4.7: Sumário numérico das 30 execuções de cada RNA obtida pela evolução – redes apresentadas na Tabela 4.6. O erro está mostrado como MSE.

| | | Média | Desv. Pad. | Mediana | Mínimo | Máximo |
|---|------------------|----------|------------|----------|--------|-----------|
| AG usando <i>fitness</i> como uma única execução da RNA | E _{1a} | 1.833,13 | 958,16 | 1.553,86 | 943,87 | 5.280,21 |
| | E _{1b} | 2.119,00 | 1.991,59 | 1.686,12 | 352,36 | 10.168,70 |
| | E _{1c} | 2.400,45 | 1.948,91 | 1.857,94 | 529,43 | 10.074,10 |
| | E _{1d} | 6.509,84 | 4.075,46 | 7.517,76 | 730,61 | 10.877,00 |
| | E _{1e} | 1.993,44 | 1.165,56 | 1.714,01 | 154,14 | 5.258,25 |
| AG usando <i>fitness</i> como média de 5 execuções | E _{5a} | 1.436,53 | 638,98 | 1.235,46 | 430,25 | 3.224,80 |
| | E _{5b} | 1.279,25 | 392,14 | 1.307,34 | 363,34 | 2.115,46 |
| | E _{5c} | 1.502,86 | 965,58 | 1.107,57 | 545,21 | 4.058,94 |
| | E _{5d} | 2.121,41 | 1.653,40 | 1.521,87 | 468,97 | 7.850,45 |
| | E _{5e} | 1.709,32 | 2.311,57 | 937,73 | 360,33 | 12.385,10 |
| AG usando <i>fitness</i> como média de 10 execuções | E _{10a} | 1.907,57 | 1.116,57 | 1.590,71 | 783,77 | 5.466,26 |
| | E _{10b} | 1.319,18 | 471,92 | 1.197,38 | 373,37 | 2.350,92 |
| | E _{10c} | 1.369,88 | 754,49 | 1.213,12 | 684,27 | 4.711,67 |
| | E _{10d} | 1.616,52 | 1.065,26 | 1.499,18 | 398,74 | 5.056,44 |
| | E _{10e} | 1.655,47 | 1.085,52 | 1.297,95 | 566,40 | 4.311,60 |

ciclos (2k) de treino apresentou os piores resultados. Isso leva a crer que esse número é muito pequeno para a rede apresentar aprendizado.

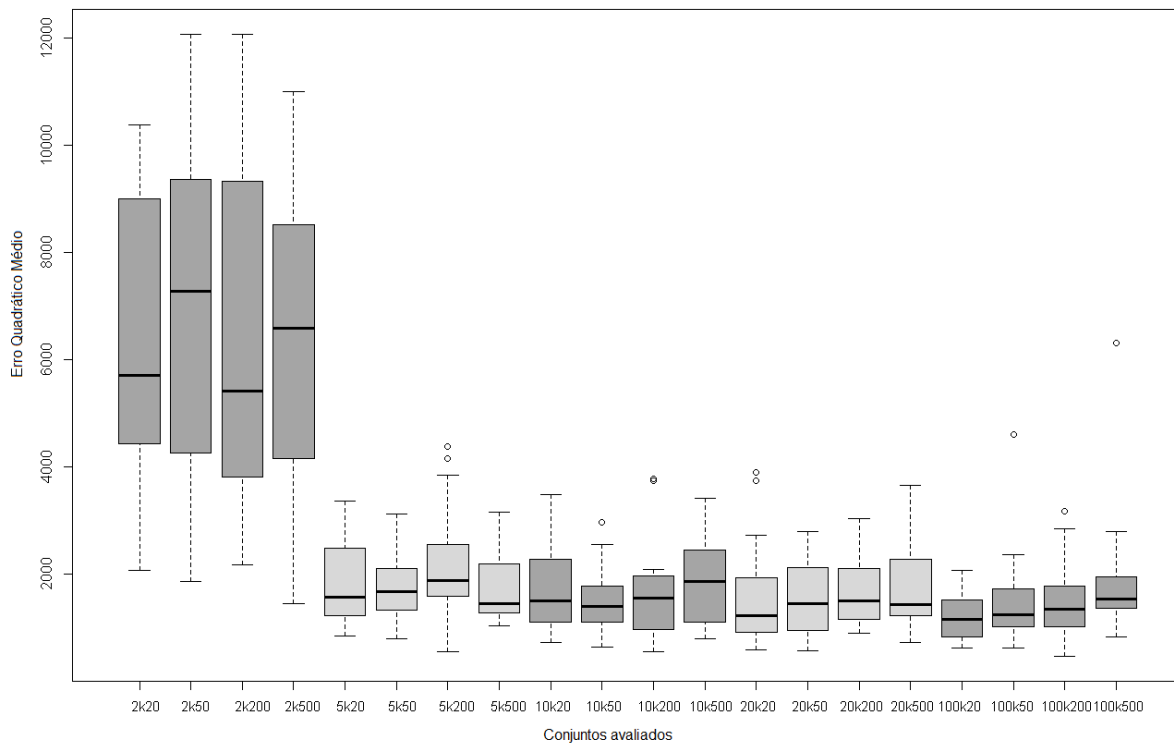


Figura 4.13: Resultados de 30 execuções com variação no número de ciclos de treino e validação sobre a rede E_{10b}. O erro está mostrado como MSE. As referências no eixo *x* representam número de ciclos de treino e validação, por exemplo, 2k20 representa 2.000 ciclos de treino e validação a cada 20 ciclos.

A análise estatística (*Kruskal-Wallis rank sum test*) sobre o conjunto de dados (desconsiderando os conjuntos com 2.000 ciclos) apresenta valor-p de 0,00027 ($<0,05$). Ou seja, os conjuntos apresentam diferenças significativas usando 95% de confiança. Dessa forma, buscamos avaliar quais membros da população seriam diferentes.

Ainda, podemos ver na Figura 4.13 que o conjunto que apresentou menor mediana e também menor dispersão é o conjunto que utilizou 100.000 ciclos (100k) de treino com validação a cada 20 ciclos. Dessa forma, usamos esse valor como melhor valor e comparamos os demais valores com esse conjunto. A Tabela 4.8 apresenta o resultado do teste estatístico entre o aparente melhor conjunto e todos os demais.

Podemos ver que existe rejeição da hipótese de igualdade entre os resultados de 100k20 com todas as redes treinadas por 5.000 ciclos (5k). Ainda, o mesmo ocorre com as redes treinadas por 10.000 ciclos (10k) com exceção da rede validada a cada 50 ciclos. Entretanto, tal valor-p está muito próximo do limite de corte (0,051). Assim, consideramos como aceitável a rede treinada por 20k ciclos, que é cerca de 5 vezes mais rápida que a rede treinada por 100k ciclos. Algo como de 30 minutos ao contrário de 150 minutos. Assim, a não diferença entre os resultados de 100k20 e 20k20 nos faz crer que, para estes dados, o treino de 20k ciclos, validado a cada 20 ciclos é suficiente.

Tabela 4.8: Resultados da comparação estatística entre 100k20 (100.000 ciclos de treino com validação a cada 20 ciclos) e os demais.

| Número de ciclos usados em treino e validação | Valor-p do teste <i>Wilcoxon rank sum test</i> | Rejeita igualdade com 95% de confiança |
|---|--|--|
| 100k50 | 0,173 | Não |
| 100k200 | 0,077 | Não |
| 100k500 | 0,000 | Sim |
| 20k20 | 0,236 | Não |
| 20k50 | 0,033 | Sim |
| 20k200 | 0,003 | Sim |
| 20k500 | 0,004 | Sim |
| 10k20 | 0,005 | Sim |
| 10k50 | 0,051 | Não |
| 10k200 | 0,034 | Sim |
| 10k500 | 0,001 | Sim |
| 5k20 | 0,001 | Sim |
| 5k50 | 0,000 | Sim |
| 5k200 | 0,000 | Sim |
| 5k500 | 0,000 | Sim |

No passo final da avaliação sobre evolução de topologias buscamos comparar a rede evoluída neste processo com a rede obtida por processo manual em (Pessin *et al.*, 2011a); para tanto, treinamos a topologia proveniente de (Pessin *et al.*, 2011a) (referenciada como **C**) com o *rprop* com 20k ciclos sendo a validação feita a cada 20 ciclos (de acordo com o obtido nos passos anteriores) para comparar com a rede de topologia evoluída (referenciada como **T**).

A Figura 4.14 e a Tabela 4.9 apresentam o resultado da comparação. São apresentados resultados de 30 execuções de cada rede neural, sendo o erro apresentado como MSE. O teste de adequação de normalidade (*Shapiro-Wilk normality test*) apresentou os seguintes valores-p para T e C: 0,334 e 0,085. Sendo assim, ambos os conjuntos não tem a hipótese de normalidade rejeitada. Desta forma, consideradas como distribuições normais, requerem um teste de comparação paramétrico, como o teste-t (*Welch Two Sample t-test*). O teste-t apresenta valor-p de 0,104. Ou seja, usando 95% de confiança a hipótese de igualdade entre as amostras não é rejeitado.

Podemos ver que a rede de topologia evoluída (T) apresenta valores mínimos menores que a rede do trabalho (Pessin *et al.*, 2011a) (C). No histograma podemos ver que apenas a rede de topologia evoluída (T) apresenta valores inferiores a 1.000 (MSE). Entretanto, a rede de topologia evoluída (T) é a que apresenta a maior dispersão nos resultados.

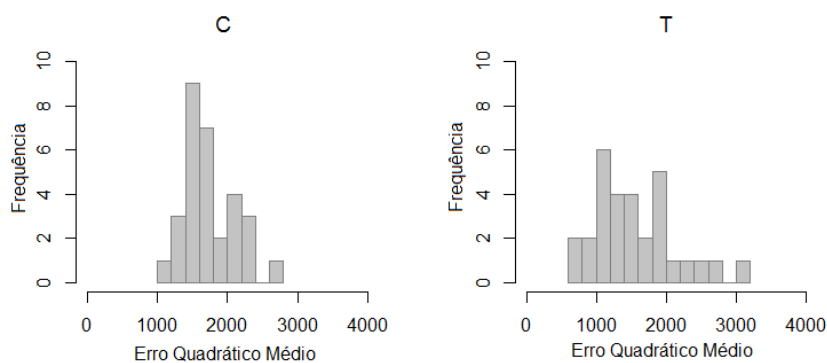


Figura 4.14: Erro médio quadrático resultante da comparação entre a rede obtida em (Pessin *et al.*, 2011a) e a rede evoluída neste trabalho. Resultados de 30 execuções. A hipótese de igualdade não é rejeitada pelo teste estatístico.

Tabela 4.9: Sumário numérico dos resultado da comparação entre a rede obtida em (Pessin *et al.*, 2011a) (C) e a rede obtida por evolução (T) neste trabalho. Resultado de 30 execuções. Erro apresentado como MSE. A hipótese de igualdade não é rejeitada pelo teste estatístico.

| | Média | Desvio | Mediana | Mínimo | Máximo |
|---|----------|--------|----------|----------|----------|
| C | 1.743,55 | 390,81 | 1.614,19 | 1.025,88 | 2.732,00 |
| T | 1.533,66 | 575,94 | 1.425,86 | 614,71 | 3.015,82 |

A maior contribuição desta seção é apresentar que um método de otimização (algoritmo genético) pôde ser utilizado para buscar (evoluir) topologias de redes neurais estatisticamente similares com as redes obtidas por processo manual. Pudemos ver que o uso de apenas uma execução de cada rede apresentou resultados sensivelmente piores que utilizando média de 5

ou 10 execuções. Estatisticamente, o conjunto resultante de uma única execução não é aceito como similar as execuções utilizando média de 5 ou 10. Execuções considerando médias de 5 ou 10 execuções apresentaram resultados mais homogêneos e com maior acuracidade.

Ainda, pudemos ver que não foi rejeitada a hipótese de igualdade entre a execução do treino da rede por 20.000 ciclos ou 100.000 ciclos. Esta descoberta é bastante importante visto que o treino de uma rede neural por um número menor de ciclos permite uso mais rápido do sistema e, por ser uma tarefa que usa bastante poder computacional, recursos como bateria, memória e CPU são poupados.

4.3.5 Outras experiências com evolução

Os trabalhos (Pessin *et al.*, 2012) e (Souza *et al.*, 2012b) fazem uma extensa busca por parâmetros em métodos de evolução a fim de apresentar resultados de evolução de redes neurais no que se refere ao peso de conexões (respectivamente, fazendo uso de otimização por enxame de partículas e algoritmos genéticos). Diferente do apresentado na seção anterior, a evolução se dá sobre os pesos das conexões e não sobre a estrutura (topologia).

Os dados utilizados na evolução dos pesos em (Souza *et al.*, 2012b) são os mesmos dados utilizados no treinamento da rede utilizando *rprop* em Souza *et al.* (2012a). Pôde-se perceber, em uma análise superficial, que a rede tendo os pesos evoluídos apresentou resultados piores que a rede treinada com o algoritmo *resilient backpropagation* (*rprop*), tanto em termos de tempo como de acuracidade. Para a rede evoluída em (Pessin *et al.*, 2012) apresentamos nessa seção a comparação com o treino usando o algoritmo *rprop*.

A Figura 4.15 apresenta o resultado de 30 execuções da rede neural treinada neste trabalho (E_1) e a rede com pesos evoluídos, de (Pessin *et al.*, 2012) (E_2).

O teste de comparação (*Welch Two Sample t-test*) apresenta valor-p inferior a 0,000, ou seja, usando 95% de confiança, os conjuntos são diferentes. Os valores-p do teste de adequação (*Shapiro-Wilk normality test*) para E_2 e E_1 são respectivamente 0,925 e 0,915.

Além de apresentar resultados melhores, o uso do *rprop* é consideravelmente mais rápido. Foram necessárias 10.000 gerações e 1.000 indivíduos para obter os resultados de E_2 (i.e. 10.000.000 de avaliações), enquanto que para o *rprop* foi utilizado 100.000 ciclos. Assim, podemos concluir com este resultado que é melhor usar um bom algoritmo de aprendizado, como o *rprop* ao contrário de um método evolutivo para a correção dos pesos das conexões.

Podemos ver, ao fim destas seções, que este trabalho mostrou que é possível encontrar a topologia da rede de modo automático e sem perda de qualidade da rede obtida. Foi possível também notar que podemos diminuir o número de épocas no treino da rede, obtendo bons resultados com economia de tempo computacional. Além disto, conforme descrito anteriormente,

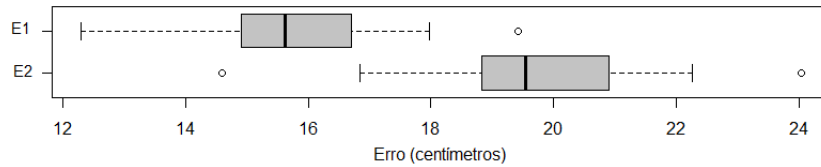


Figura 4.15: Resultados para a rede com evolução de pesos (E_2) e para a rede treinada (E_1). Além do treino ser mais rápido do que a evolução, a rede treinada apresenta resultados $\approx 20\%$ melhores do que a rede com evolução de pesos.

podemos ver que para os dados analisados, a rede neural apresentou capacidade de estimar a localização com um erro médio de $9,6 \text{ cm} \pm 6,7 \text{ cm}$ (usando uma janela de oito leituras no filtro de mediana móvel). A remoção de pontos de acesso, considerando o *hardware* utilizado, não melhora significativamente a estimativa da localização. As próximas duas subseções apresentam propostas, baseadas na experiência obtida nos experimentos anteriores, que devem vir a ser implementadas e investigadas como trabalho futuro. Após, serão descritas as considerações finais do capítulo, discutindo de forma sintética os resultados obtidos.

4.3.6 Tolerância a falhas

Em (Pessin *et al.*, 2011a) – investigação sobre localização *indoor* utilizando o robô SRV ao contrário do robô NAO – foi simulado erro nos APs (i.e. erro nas entradas da rede neural). O erro simulado foi realizado por meio da injeção de valores ruidosos em algumas das entradas da RNA, representando falhas nos APs. Com a injeção de erro (e sem o uso de filtros), o sistema passou a apresentar resultados sensivelmente piores, indo de $\approx 90 \text{ cm}$ de erro médio para $\approx 286 \text{ cm}$ de erro médio.

Dessa forma, descrevemos uma proposta, baseada na experiência obtida nos experimentos anteriores, que deve vir a ser implementada e investigada como trabalho futuro. O procedimento proposto tem como objetivo tornar o sistema capaz de manter certo grau de acuracidade mesmo com mudanças nos valores de entrada. Duas são as propostas, a primeira envolve apenas o agente por si mesmo (*stand alone*), a segunda envolve uma abordagem multi-agente.

Na abordagem *stand alone*, sugere-se o uso de um limite (*threshold*) de RSSI para cada ponto de acesso disponível na fase de coleta de dados. Se, durante a fase de treino, os valores ficam dentro dos limites, na fase de uso do sistema os valores fora do limite deveriam ser desconsiderados. Os valores poderiam ser desconsiderados de duas formas: (i) ignorar a leitura do ponto de acesso no tempo atual, atrasando as leituras; o que pode deixar a obtenção da posição mais lenta, ou (ii) ignorar a leitura do ponto de acesso e utilizar redes neurais específicas para o número de entradas disponíveis; o que faz com que necessitemos de uma coletânea de

redes neurais com todas as possíveis combinações de entradas. Para evitar o treino de todas as possíveis configurações no *startup* do sistema, é possível fazer o treinamento *on the fly*. Ou seja, o sistema inicia treinando apenas a rede de 8 entradas (a rede completa) e então, ao identificar alguns poucos valores fora do limite poderia usar a abordagem (i), e, ao identificar um maior número de valores fora da faixa passaria para a abordagem (ii) treinando as redes necessárias.

Na abordagem multiagente, o agente (robô) a ser localizado deveria ter capacidade de comunicação com um dos pontos de acesso, sendo que o ponto de acesso deveria ter capacidade de obter os RSSIs dos demais pontos de acesso (chamaremos este ponto de acesso de âncora). A ideia proposta então é a seguinte: na fase de treinamento, o robô se desloca no plano como atualmente proposto. O ponto de acesso âncora, fixo em uma posição, deve então realizar a leitura dos demais pontos de acesso. Na fase de uso do sistema pelo robô (usando sua rede neural treinada) ele deve se comunicar com o ponto âncora a fim de solicitar informação sobre a qualidade do sinal. Por estar em uma posição fixa, o ponto âncora tem como saber quais dos pontos de acesso estão apresentando mudanças na potência. Esta abordagem suporta erros sem se preocupar se os valores estarão dentro de limites. Quaisquer discrepâncias nos valores, ou mesmo descida ou subida de padrões de potência, poderiam ser identificadas e corrigidas.

Estas duas abordagens são propostas baseadas na experiência obtida nos experimentos anteriores, visando uma possível manutenção da metodologia de localização mesmo com erros nos APs. Ambas as propostas devem vir a ser implementadas e investigadas como trabalhos futuros.

4.3.7 Uma abordagem para maior autonomia

A tarefa envolvida neste capítulo tem basicamente 3 fases: (i) coleta dos dados; (ii) treino da rede neural e (iii) uso da rede neural. As tarefas (ii) e (iii) podem ser realizadas autonomamente, como apresentado previamente neste capítulo. Entretanto, a fase de coleta dos dados foi realizada manualmente. Tal coleta de forma manual não é desejada e nem mesmo é possível de ser realizada em diversos casos.

Nas avaliações apresentadas, um humano foi responsável pela marcação dos pontos no plano e pela movimentação/colocação do robô sobre os pontos para a coleta das informações dos pontos de acesso e de posição (Figura 4.1). Sugere-se então, para uma abordagem mais autônoma, que o robô seja autossuficiente na coleta dos dados. Tal autossuficiência pode ser obtida com a substituição do humano por um sistema que permita ao robô medir seu deslocamento. Neste caso, o uso de odometria visual poderia ser realizado para obter a posição do robô e mensurar o deslocamento. Sugere-se o uso de odometria visual apenas para a coleta dos dados que servirão

de treinamento visto que a técnica é muito cara em termos computacionais e apresenta algumas imprecisões quando utilizada por longos períodos.

Esta abordagem é também uma proposta baseada na experiência obtida nos experimentos anteriores, visando a possibilidade de dotar o sistema de um maior grau de autonomia. Esta proposta deve vir a ser implementada e investigada como trabalho futuro.

4.4 Considerações Finais

O capítulo se inseriu no contexto da investigação do uso de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina para auxiliar na tarefa de localização de robôs móveis em ambientes internos. Descrevemos propostas e avaliações de métodos de localização de agentes móveis usando RSSI obtido de APs. Embora extensamente avaliado, o trabalho sobre localização usando redes sem fio ainda apresenta temas abertos de investigação, a serem descritos na seção de trabalhos futuros (ver Seção 4.5 a seguir). Entretanto, os atuais resultados utilizando o RSSI se apresentaram encorajadores e motivam a continuidade da pesquisa.

A investigação realizada mostrou uma metodologia para localização interna. Tal metodologia tem como objetivo prover informação de localização de forma prática, barata e acurada tanto para robôs móveis como para outros tipos de dispositivos (por exemplo, uma rede de sensores sem fio). A metodologia inicial, embora funcional, mostrou algumas limitações na questão de completa autonomia e correção contínua de erro. Entretanto, tais limitações podem vir a ser corrigidas com as propostas descritas nas Seções 4.3.6 (Tolerância a falhas) e 4.3.7 (Uma abordagem para maior autonomia).

Pelo fato da potência do sinal de rede ter grandes oscilações, foi proposto e avaliado o uso de filtro de mediana móvel. Pudemos ver que não existe diferença estatística entre o uso do valor bruto ou da mediana (média) de duas leituras. Entretanto, para as demais janelas de tempo, o aumento na acuracidade do aprendizado da rede neural foi bastante promissor. Mostramos inclusive que existe correlação forte entre o tamanho da janela e a acuracidade no aprendizado. Pudemos ver que, a partir de uma janela de oito leituras, o erro mínimo, aparentemente, não diminui muito, entretanto as medidas dos erros continuam caindo. Usamos o filtro de mediana ao contrário do filtro de médias pela característica das medianas em remover inerentemente os *outliers*. É importante perceber que o tamanho da janela significa, sob um ponto de vista mais crítico, um atraso na obtenção da posição. Assim, um usuário deste tipo de sistema deveria enfrentar o dilema de escolher entre acuracidade e velocidade na obtenção da informação.

Alguns pontos de acesso apresentam valores com mais oscilações que outros. Foi um dos objetivos avaliar o aprendizado relacionando a quantidade de redes *wireless* disponíveis (quan-

tidade de entradas no sistema) e a acuracidade no aprendizado da posição pela rede neural. Mostramos que, mesmo com pontos de acesso com amplitude mais alta, a remoção destes não auxiliou no aprendizado. Ou seja, mesmo mantendo os pontos com nível mais alto de oscilação, para o conjunto avaliado, não obtivemos melhorias. Isso possivelmente se deve ao fato de que a amplitude nos pontos de acesso utilizados não é muito elevada e também pela capacidade inerente das redes neurais em suavizar as entradas.

Avaliações de topologias de redes neurais por meio de processo manual se mostraram eficientes na obtenção de boas redes, entretanto, se apresentaram como um trabalho bastante árduo e extenuante. Uma falta de maiores garantias na acuracidade do aprendizado também decorre deste processo manual, caso seja considerado um conjunto muito pequeno de avaliações. Tal problema motivou o desenvolvimento de um método de evolução de topologias a fim de tornar o processo de descoberta de boas redes um processo automatizado. Mostramos como um método de otimização (algoritmo genético) pôde ser utilizado para buscar (evoluir) topologias de redes neurais com resultados semelhantes aos das redes obtidas por processo manual. Pudemos ver que o uso de apenas uma execução de cada rede apresentou resultados sensivelmente piores que utilizando média de 5 ou 10 execuções para obtenção do *fitness*. Execuções considerando médias de 5 ou 10 execuções apresentaram resultados mais homogêneos e com maior acuracidade. Ainda, pudemos ver que não é rejeitada a hipótese de igualdade nos resultados quando executando o treino das redes por 20.000 ciclos ou 100.000 ciclos. Esta descoberta é bastante importante visto que o treino de uma rede neural por um número menor de ciclos permite uso mais rápido do sistema e, por ser uma tarefa que usa bastante poder computacional, recursos como bateria, memória e CPU são poupados.

É importante perceber a importância do AG na busca por topologias (adotando uma abordagem híbrida com o SNNS para treinar os pesos das conexões com o algoritmo *rprop*). Pudemos encontrar topologias com resultados semelhantes as obtidas por processo manual sem o gasto de tempo do pesquisador. O processo evolucionário é demorado (dependendo da base de dados e de outros parâmetros, pode consumir um tempo elevado de processamento), entretanto, o pesquisador não fica preso ao processo. Otimizações no algoritmo ou execução em paralelo poderiam ser realizadas a fim de tornar o processo mais rápido.

Também avaliamos um algoritmo evolucionário no que se refere à evolução de pesos de conexões. A evolução de pesos é importante e necessária em sistemas onde as bases de dados não podem ser pré-obtidas (que não é exatamente nosso caso). Entretanto, a dúvida entre evolução e treino existia. Pudemos ver que o método de otimização, eficiente para a busca de topologias, não se mostrou interessante para evolução dos pesos neste trabalho. Os resultados usando *rprop*

no treino da rede foram melhores que usando o PSO⁸ tanto em termos de acuracidade como em tempo de execução. Era esperada, segundo alguns autores, que a evolução dos pesos usando PSO (ou outro método qualquer de otimização/evolução) proporcionasse resultados finais com uma menor dispersão, dada a capacidade de fazer os pesos das conexões escaparem de mínimos locais. Entretanto, tal característica não foi observada para o tipo de problema tratado.

Para finalizar, é bom lembrar que a abordagem desenvolvida para localização em ambientes internos teve como principal motivador a possibilidade de uso de robôs de pequeno porte em ambientes indoor. O uso de robôs, em atividades práticas úteis, usualmente depende de capacidades de localização e navegação. Sendo que a navegação (a ser investigada no Capítulo 5), em geral, também depende da capacidade de localização. Ainda, o uso de robôs de pequeno porte em ambiente indoor é um dos requisitos do desenvolvimento real, em pequena escala, da investigação descrita no Capítulo 6. Além disso, os próximos pesquisadores que realizarem investigações com os robôs SRV-1Q (a exemplo dos trabalhos apresentados nos artigos (Pessin *et al.*, 2011a,b; Sales *et al.*, 2010)), ou Pioneer AT (a exemplo dos trabalhos apresentados nos artigos (Hata *et al.*, 2009a; Souza *et al.*, 2011a)), disponíveis no LRM, ou com o robô NAO, disponível no *Robotics Lab*, podem se beneficiar desta metodologia de localização desenvolvida.

4.5 Trabalhos Futuros

Foram vislumbrados, no decorrer desta investigação, uma série de trabalhos futuros relacionados com possíveis otimizações, tanto em termos de acuracidade no aprendizado como na praticidade do uso do sistema proposto. São eles:

- O estudo sobre a influência do tamanho da base de treinamento. Os resultados apresentados utilizaram leituras de 3 minutos em cada posição. Seria isso um tamanho além do suficiente? Como o aprendizado no sistema de localização se comporta com coletas de 10, 30, 90 ou 180 segundos?
- O uso de redes neurais é realmente melhor que o uso de outras técnicas de aprendizado de máquina, como máquinas de suporte vetorial ou processos gaussianos?
- Para o ambiente utilizado, o quão diferente em termos de arquitetura e acuracidade pode ser o modelo proposto se considerada a implementação sem aprendizado de máquina, utilizando funções matemáticas de descrição de sinal?

⁸É importante salientar que o PSO utilizado na evolução (para uma comparação mais justa) usa um conjunto de parâmetros bastante investigado, obtido de Pessin *et al.* (2012), onde foram realizadas 3 passos de avaliações considerando 9, 9 e 18 diferentes configurações de parâmetros no PSO para a obtenção do melhor conjunto.

- Como o modelo baseado em rede sem fio pode ser adaptado para obtenção de orientação e de altitude, a fim de prover a pose completa do robô não apenas a posição?
- Como o sistema proposto, e suas possíveis variantes, se comportam em ambientes altamente dinâmicos?
- Não foram feitos estudos sobre o uso a longo prazo do sistema de localização usando redes sem fio. Seria esse tipo de sistema passível de deteriorização com o passar do tempo? Ou ainda, seria o sistema sensível a mudanças ambientais, como temperatura e umidade? Poderiam essas questões se tornar itens fora de preocupação quando considerado o uso do sistema com a proposta de tolerância a falhas?
- Implementação e validação das hipóteses apresentadas nas Seções 4.3.6 (Tolerância a falhas) e 4.3.7 (Uma abordagem para maior autonomia).

As questões acima demonstram que este tema é ainda um tema de pesquisa aberto e que permite uma série de questionamentos e experimentos futuros visando aperfeiçoar as abordagens propostas. Por outro lado, os resultados obtidos já permitem sua aplicação prática em diversos experimentos práticos de sistemas multirroboóticos que requeiram uma determinação da posição de robôs em um ambiente fechado, constituindo-se assim de uma interessante contribuição para as pesquisas na área e inclusive para nosso grupo no LRM.

Navegação Autônoma

Este capítulo apresenta o desenvolvimento e os resultados de pesquisas em navegação autônoma. Nos Capítulos 4 e 6, o autor desta tese foi o proponente e autor principal da maior parte dos trabalhos. No entanto, os trabalhos apresentados neste capítulo foram fortemente construídos a partir de trabalhos em colaboração, sendo a atuação do autor deste documento em co-autoria em trabalhos de colegas do LRM/USP (2013), principalmente com Jefferson R. Souza, Alberto Y. Hata, Patrick Y. Shinzato, Daniel O. Sales, Leandro C. Fernandes, e, sob orientação dos professores Fernando S. Osório e Denis F. Wolf.

O capítulo é descrito com base na sequência lógica usada para o desenvolvimento de um veículo real para navegação autônoma. Inicialmente abordamos sucintamente aspectos de simulação (investigados em (Pessin, 2008) e (Pessin *et al.*, 2010a)) que serviram como base para a proposta e aperfeiçoamento de uma arquitetura para navegação autônoma usando robôs reais de pequeno porte, seguido do desenvolvimento e investigação com um robô de grande porte. No final do capítulo descrevemos sucintamente modelos de mapeamento e sua relação com o desenvolvimento de abordagens mais reativas ou deliberativas e seu impacto no desenvolvimento de um método seguro e de alta acuracidade de navegação autônoma.

Como descrito na introdução, este capítulo tem como objetivo principal ser um elo entre os conceitos de localização e coordenação de grupos, sendo o foco do capítulo a descrição de aspectos que tangem um desenvolvimento mais técnico de um veículo autônomo, onde destacam-se que as principais contribuições científicas originais desta tese encontram-se des-

critas nos Capítulos 4 (Localização) e 6 (Grupos de Robôs). O capítulo descreve contribuições relacionadas com o desenvolvimento de controladores para sensores e atuadores dos robôs do LRM/USP (2013), bem como sobre o desenvolvimento geral do veículo. Não é objetivo entender muito esta discussão, visto que este é um tema de pesquisa compartilhado com outros colegas do Laboratório de Robótica Móvel (ICMC/USP), e será produto de outras teses e dissertações. O que será apresentado aqui será contextualizado em relação a presente tese e aos trabalhos que tiveram a efetiva participação do autor desta.

O desenvolvimento de métodos de navegação autônoma é um tópico muito ativo de pesquisa, tanto do ponto de vista das pesquisas internacionais em universidades e mais recentemente na indústria automotiva, bem como no próprio LRM/ICMC/USP. Os resultados apresentados neste capítulo tem relação principal com o início do desenvolvimento e início das investigações com o primeiro veículo completamente autônomo de grande porte do LRM/USP (2013) (Carina I – Figura 5.1(a)). Tal veículo tem sido utilizado para tarefas de mapeamento (Hata *et al.*, 2009a,b, 2010), estacionamento (Prado, 2013), navegação com visão estéreo (Mendes *et al.*, 2012; Mendes e Wolf, 2011) entre outros. Um novo veículo de grande porte (CaRINA II – Figura 5.1(b)) está atualmente sendo desenvolvido e utilizado paralelamente ao CaRINA I.



(a)



(b)

Figura 5.1: (a) CaRINA I – primeiro veículo autônomo de grande porte do LRM/ICMC/USP. A implementação do controle de atuação permitiu o começo de sua utilização no início de 2010. (b) CaRINA II – segundo veículo autônomo de grande porte do LRM/ICMC/USP. A implementação do controle de atuação permitiu o começo de sua utilização no fim de 2012.

Mais especificamente, este capítulo é construído com base nos trabalhos (Fernandes *et al.*, 2012; Hata *et al.*, 2009a,b; Pessin *et al.*, 2010a; Souza *et al.*, 2012a, 2010, 2011a,b, 2013). Os trabalhos (Pessin *et al.*, 2010a; Souza *et al.*, 2010) são peças iniciais de investigação, explorando

aspectos de navegação em ambientes simulados. Foram utilizados robôs fisicamente simulados construídos com a biblioteca de simulação física *Open Dynamics Engine* (Smith, 2006). Respectivamente, os trabalhos fazem uso de Redes Neurais Artificiais e Árvores de Decisão no controle de navegação e desvio de obstáculos dos veículos desenvolvidos. Os trabalhos (Souza *et al.*, 2012a, 2011a,b) são peças intermediárias de investigação que conduziram ao trabalho (Souza *et al.*, 2013). Este último apresenta de forma completa as maiores contribuições e o aprendizado no desenvolvimento de um veículo autônomo para navegação em ambientes externos. Fora do eixo de navegação, o trabalho (Fernandes *et al.*, 2012) apresenta uma coletânea dos trabalhos do LRM/USP (2013) e também apresenta de forma detalhada a arquitetura do veículo desenvolvido, os trabalhos (Hata *et al.*, 2009a,b) são investigações sobre mapeamento.

A fim de tornar um capítulo mais sucinto, não serão descritos de forma muito detalhada os experimentos intermediários que conduziram ao maior experimento e suas contribuições. Todos os experimentos intermediários (em simulação e em robôs de pequeno porte) tiveram como objetivo a condução da investigação como um processo gradual. Desta forma, o último trabalho é o mais relevante e será o mais detalhado neste capítulo.

As principais contribuições deste capítulo são:

- Aperfeiçoamento de uma arquitetura para navegação autônoma com capacidade de seguir pontos esparsos de GPS. Tal aperfeiçoamento, com uso de robôs reais, é resultado de novas investigações sobre os trabalhos apresentados em (Pessin, 2008) e Pessin *et al.* (2010a), onde uma arquitetura de navegação autônoma com desvio de obstáculos foi avaliada em simulação.
- Avaliação de diferentes topologias de redes neurais buscando verificar as topologias que permitam o aprendizado com maior acuracidade.
- Discussão sobre as capacidades de aprendizado.
- Apresentação de algumas contribuições técnicas no desenvolvimento de controladores de sensores para o robô CaRINA I.

O restante do capítulo é dividido nas seguintes seções: a primeira discute sucintamente as etapas iniciais em simulação, robôs de pequeno porte, e aspectos de desenvolvimento da autonomia do robô de grande porte CaRINA I. A segunda parte apresenta a descrição e os resultados do modelo proposto de navegação autônoma. Em seguida, são apresentados sucintamente modelos de mapeamento e sua relação com o desenvolvimento de métodos mais seguros e de alta acuracidade de navegação autônoma. Finalizamos com as considerações finais e os trabalhos futuros.

5.1 Da Simulação aos Robôs Reais

Os primeiros passos no desenvolvimento de métodos de navegação autônoma foram realizados em simulação. A biblioteca de simulação física *Open Dynamics Engine* (Smith, 2006) foi utilizada para o desenvolvimento dos ambientes de simulação, apresentados na Figura 5.2.

Pessin *et al.* (2010a) (Figuras 5.2(a), 5.2(b) e 5.2(c)) apresenta robôs simulados equipados com os seguintes sensores: bússola (orientação), GPS (localização), sete sensores de distância (como *lasers* simulados). Os atuadores são um motor para giro da barra de direção (esterçamento) e um motor para deslocamento linear (velocidade). Um sistema baseado em regras foi desenvolvido para a navegação, sendo utilizado para geração de logs e subsequente treinamento de uma rede neural. Souza *et al.* (2010) (Figura 5.2(d)) apresenta robôs com a mesma carga sensorial do robô usando em (Pessin *et al.*, 2010a), entretanto, foi avaliado o aprendizado da navegação com uso de árvores de decisão (AD). A capacidade de navegação, tanto usando redes neurais ou árvores de decisão se mostrou bastante eficiente. O maior problema com os modelos é que ambos tiveram bases de dados para treino do método extraídas de um sistema de regras. Tal sistema de regras foi criado por um especialista, assim, a base de dados de treino é extraída de um comportamento definido previamente e não apenas do uso normal do sistema por um motorista humano.

Após os modelos em simulação terem alcançado um comportamento adequado, passos foram realizados no desenvolvimento em robôs de pequeno porte, reforçando conceitos referentes ao uso de dispositivos físicos, permitindo, a partir de experiências práticas, a percepção das sutis diferenças entre os “mundos” simulado e não simulado. Com este intuito, o trabalho previamente desenvolvido em ambiente de simulação apresentado em (Pessin *et al.*, 2010a) foi adaptado para atuação em ambiente real utilizando o robô Pioneer 3-AT (MobileRobots, 2013). A Figura 5.3(a) apresenta a cena do *setup* do experimento¹. O *software* de controle é instalado no computador (*notebook*) posicionado sobre o robô, possuindo comunicação serial com o *hardware* do mesmo, a fim de enviar os comandos para os motores. Os sensores presentes neste experimento são responsáveis pela obtenção de localização (GPS) e orientação (bússola), conectados diretamente no *notebook*. O objetivo deste experimento foi validar, em termos práticos, um modelo de navegação por meio de sequência de coordenadas UTM. A evolução deste modelo foi desenvolvida para suportar comunicação por meio de rede sem fio entre dois robôs, sendo um protótipo de sistema de comboio².

¹ Vídeio do experimento disponível em <http://www.youtube.com/watch?v=xRURzHdfbIQ>

² Arquivo contendo a apresentação do sistema em formato de *slides* disponível em <http://goo.gl/O9ha8>

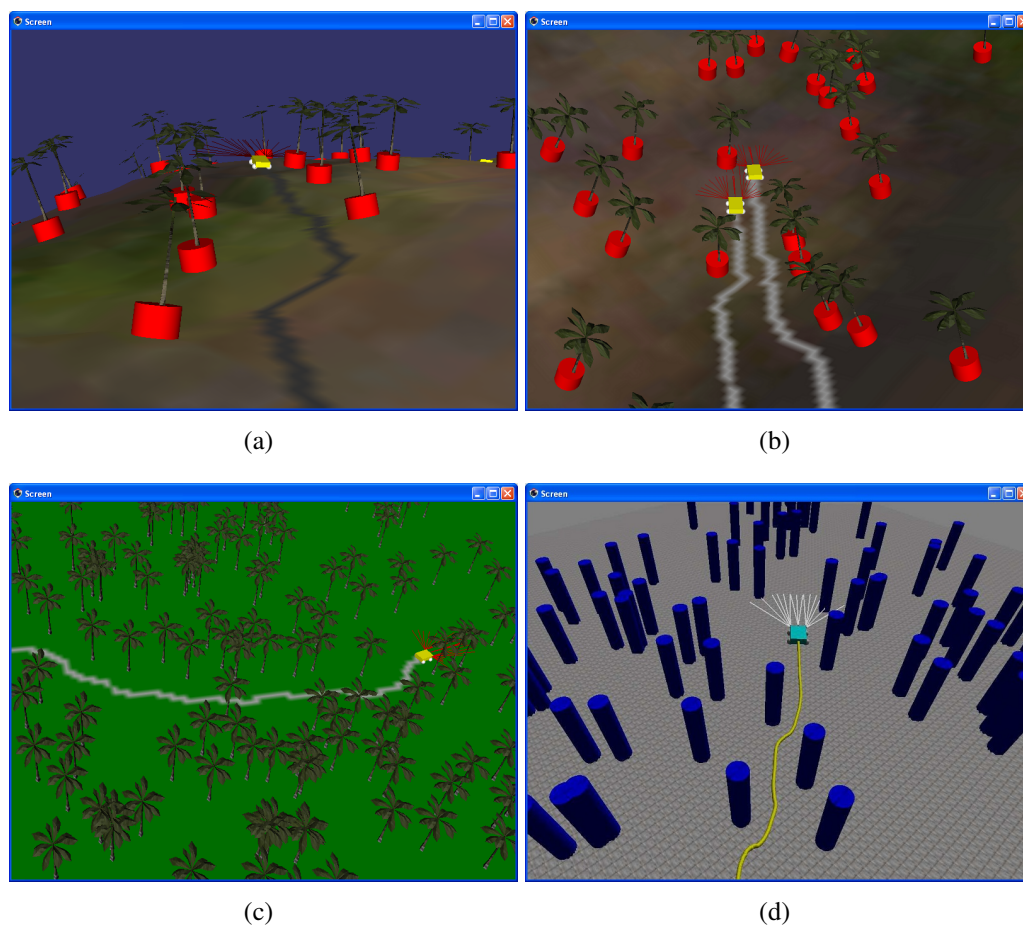


Figura 5.2: Ambientes de simulação para grupos de robôs. (a), (b) e (c) Ambiente de simulação utilizado em (Pessin, 2008; Pessin *et al.*, 2010a) – conta com terrenos irregulares e suporte a características ambientais (*e.g.* fogo, vento). (d) Ambiente de simulação simplificado, com foco em tarefa de navegação autônoma e desvio de obstáculos utilizado em (Souza *et al.*, 2010).

Para tal sistema, foram desenvolvidos controladores (*drivers*) para o *GPS Garmin 18x 5Hz*³ e para a bússola *True North Revolution GS*⁴. Ambos os dispositivos possuem comunicação serial em formato ASCII, de acordo com o padrão NMEA 0183⁵. Os sistemas desenvolvidos (navegação individual e modo comboio) se apresentaram completamente funcionais, porém, pôde-se perceber que a imprecisão e as interferências externas nos sensores e atuadores deixam o sistema relativamente frágil.

³Código-fonte disponível em <http://goo.gl/OTgWr>

⁴Código-fonte disponível em <http://goo.gl/PAZPM>

⁵NMEA 0183 é um padrão para comunicação de dados de dispositivos definido e controlado pela *National Marine Electronics Association*.



(a)

(b)

Figura 5.3: Robôs de pequeno porte. (a) *Setup* do experimento usando o robô Pioneer P3-AT. (b) Robô de pequeno porte SRV-1Q (Surveyor, 2010).

Outro trabalho na linha de robôs reais, foi o desenvolvimento de uma Interface de Programação de Aplicativos (*Application Programming Interface* – API) para o robô de pequeno porte SRV-1Q⁶ (Surveyor, 2010) (Figura 5.3(b)). O robô basicamente pode ser tratado como um dispositivo de rede, onde é feita a conexão com o mesmo por meio de uma rede sem fio. Os comandos de movimentação e as leituras sensoriais são realizadas por meio de envio e recebimento de *sockets*. Tendo como objetivo o desenvolvimento de código-fonte independente de plataforma, na API desenvolvida não foram utilizadas as bibliotecas padrão dos sistemas operacionais *GNU/Linux* ou *MS-Windows*. Foi utilizada a biblioteca *Simple Directmedia Layer* (SDL, 2013). Esta biblioteca promete suportar diversos sistemas operacionais. Um trabalho utilizando o robô SRV-1Q, no controle de navegação em ambientes internos, baseado em visão e usando redes neurais para classificação de vias navegáveis e autômatos finitos para decisões de trajetória, feito em colaboração com colegas do Laboratório de Robótica Móvel (ICMC/USP) pode ser visto em (Sales *et al.*, 2010).

Como citado anteriormente, estes trabalhos tiveram o objetivo de proporcionar ao aluno um maior contato com sistemas robóticos reais, dando uma base prática bem definida na área da robótica móvel real, além de uma maior base em diversas características e métodos de aprendizado de máquina para controle e navegação de robôs móveis atuando de forma individual. Além disto, um estudo aprofundado e importante sobre o funcionamento e uso de sistemas de orientação por GPS e bússola foram também desenvolvidos nesta etapa do trabalho. Fatos estes que tornam possível o avanço da pesquisa proposta utilizando robôs reais em ambientes dinâmicos e com colaboração técnica importante ao LRM/USP (2013).

⁶Dicas de utilização e código-fonte disponível em <http://lrm.icmc.usp.br/wiki/index.php/SRV-1Q>

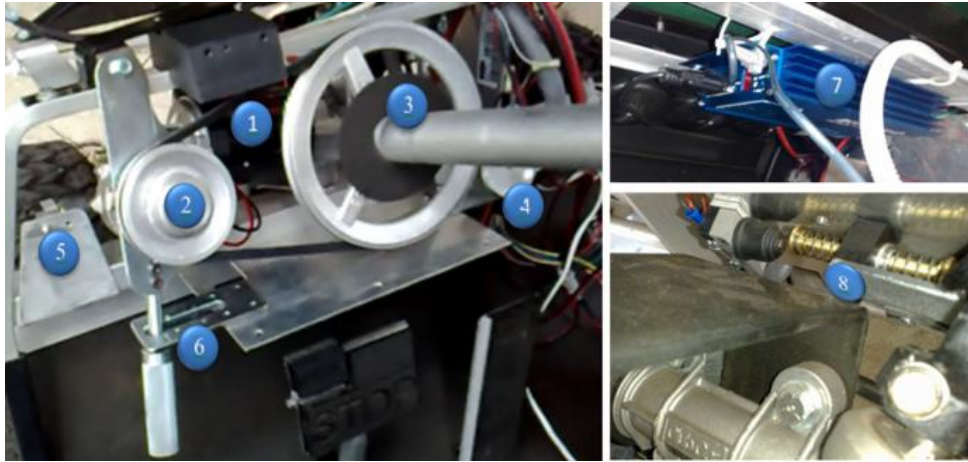


Figura 5.4: Sistema de esterçamento do robô CaRINA I (Fernandes *et al.*, 2012). Em detalhes: (1) motor, (2) sistema de polia de acoplamento, (3) componente do sistema de direção, (4) encoder digital, (5) interruptor de acoplamento, (6) alavanca de acoplamento, (7) controlador de motor RoboteQ, (8) dispositivos limitadoras.



Figura 5.5: Dispositivo controlador de aceleração (Fernandes *et al.*, 2012). Na direita, vista do interior com destaque para o controlador Arduino (Arduino, 2013) e o acoplamento mecânico entre servo motor e potenciômetro.

O robô de grande porte, utilizado nos próximos experimentos, é o Carina I (Figura 5.1(a)). Na investigação realizada, em termos sensoriais, o veículo é composto por um GPS, uma bússola, e uma câmera monocular. Em termos de atuação, o veículo é composto por atuadores de barra de direção (Figura 5.4) e de aceleração (Figura 5.5). Nestes experimentos, os drivers de bússola e GPS utilizados foram os mesmos utilizados no robô de pequeno porte Pioneer AT (MobileRobots, 2013).

5.2 Navegação Autônoma em Ambientes Externos

Esta seção descreve os trabalhos realizados com foco em navegação autônoma em ambientes externos. O objetivo desta investigação é a proposta e a avaliação de métodos para navegação autônoma baseado em aprendizado. O desenvolvimento teve por objetivo fazer com que o veículo aprendesse e repetisse uma rota baseada em pontos esparsos de GPS (por exemplo, foram utilizado sete coordenadas em um percurso de $\approx 1,08$ km). Os ambientes utilizados para avaliação podem ser vistos na Figura 5.6.

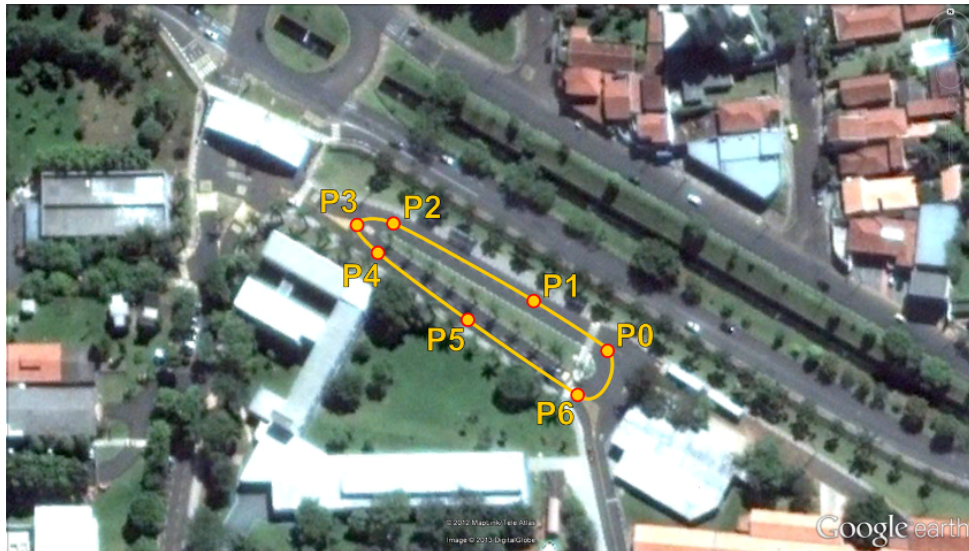
Apresentamos um sistema de controle que permite a navegação autônoma de um veículo, sendo a abordagem baseada em processamento de imagens e utilizando uma câmera monocular. Tal parte é responsável por reconhecer áreas navegáveis ou não. A definição de áreas navegáveis serve como base para o conjunto de regras definir os comandos de atuação. O veículo segue uma lista de coordenadas obtidas por GPS. São providos esterçamento (giro da barra de direção) e velocidade. Entretanto, a velocidade só deve ser considerada pensando que a mesma foi obtida por um conjunto de regras, principalmente pelo fato de que o veículo, na época da realização dos experimentos, não apresentava capacidade de frenagem controlada, assim, foi bastante comum utilizar no experimento um humano responsável por pisar no acelerador/freio, aumentando a segurança do experimento do ponto de vista da preservação do veículo.

Em termos de execução e uso do sistema, o mesmo é dado pelas seguintes etapas:

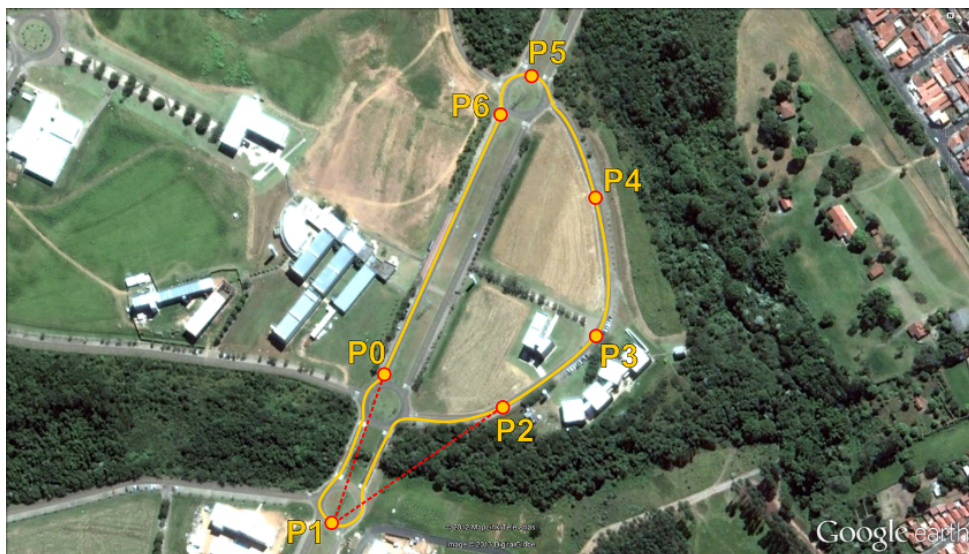
Passo 1: Coleta de pontos esparsos de GPS, como apresenta a Figura 5.6.

Passo 2: Coleta de imagens (≈ 30 s) para (i) treinamento do algoritmo de visão de Shinzato e Wolf (2011) (item (a) na Figura 5.7) e (ii) definição de geometrias sobre as imagens (item (b) na Figura 5.7). O treino do algoritmo de visão identifica áreas navegáveis e não navegáveis com base em imagens obtidas a partir de uma câmera monocular. O resultado é um mapa de navegação visual, a ser detalhado na Seção 5.2.1. Este mapa de navegação visual é utilizado para definição de geometrias (*template matching*), ou seja, é utilizado a fim de identificar possíveis geometrias na frente do veículo, como linha reta, curva leve ou acentuada para a esquerda e curva leve ou acentuada para a direita. Sobre estas geometrias é calculado o percentual de ocupação de cada área, o que implica em dizer que tal geometria está livre (área navegável) ou não. Esta etapa será detalhada na Seção 5.2.2.

Passo 3: Tarefa de navegação propriamente dita (por um humano ou por um conjunto de regras); o veículo navega seguindo as coordenadas GPS. É obtido um arquivo de *log* com os percentuais de ocupação dos *templates* (que tem como base o método de visão do passo 2). O arquivo de *log* contém então as entradas e as saídas a serem utilizadas no método de aprendizado de máquina.



(a)



(b)

Figura 5.6: (a) Trajeto percorrido pelo veículo considerando o primeiro ambiente (≈ 200 m). P0 até P6 especificam os pontos das coordenadas de GPS utilizados. A linha amarela apresenta o trajeto percorrido aproximado. (b) Trajeto percorrido pelo veículo considerando o segundo ambiente ($\approx 1,08$ km). P0 até P6 especificam os pontos das coordenadas de GPS utilizados. A linha amarela apresenta o trajeto percorrido aproximado. As linhas vermelhas apresentam a orientação calculada pelo veículo para ir de um ponto a outro.

Passo 4: Treinamento da rede neural usada como controlador de navegação, com os dados obtidos no passo 3. A ser detalhado na Seção 5.2.3. Item (c) na Figura 5.7.

Passo 5: Uso do veículo com a rede neural como controlador de navegação. A avaliação do treinamento e do uso do veículo com a rede neural como controlador será detalhada na Seção 5.2.4; é usado o controlador completo, mostrado na Figura 5.7.

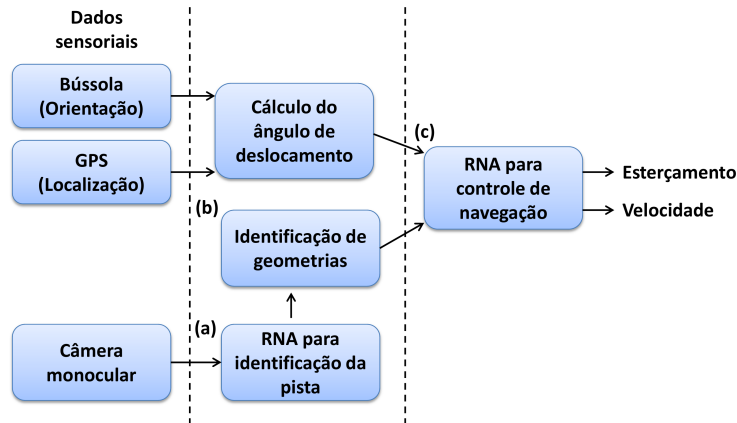


Figura 5.7: Modelo completo do controlador de navegação.

5.2.1 Identificação da pista

A identificação da pista se dá por meio da identificação de áreas navegáveis utilizando o método proposto por Shinzato e Wolf (2011). Em tal método, é criado um comitê de redes neurais (comitê de classificação) que classifica regiões de cada imagem. A imagem obtida pela câmera é dividida em blocos de n por n píxeis, como mostra a Figura 5.8.

A classificação em área navegável (segura, *e.g.* asfalto, rua) e área não navegável (inadequada para o veículo, *e.g.* calçada e vegetação) de cada bloco é dada pelo valor médio obtido por um comitê de RNAs de classificação. Para cada bloco são obtidas características relacionadas com canais de cor, usadas pela RNA para determinar se o bloco pertence ao grupo de regiões navegáveis ou não, sendo que a rede é treinada previamente podendo ser adaptada para classificar diferentes tipos de vias: pista de asfalto, estrada de terra, etc. As características e estatísticas utilizadas são, de acordo com o especificado em (Shinzato e Wolf, 2011): média, entropia, variância e energia. Estas medidas são associadas com canais de cor RGB, HSV, YUV e RGB normalizado. A Tabela 5.1 apresenta os atributos de forma detalhada.

A Figura 5.9 apresenta um exemplo das etapas de obtenção de imagem, passagem pelo comitê de classificação e saída do comitê de RNAs como um mapa de navegabilidade. As RNAs utilizadas nesta etapa são *multi-layer perceptron* (MLP) treinadas com o algoritmo *resilient*

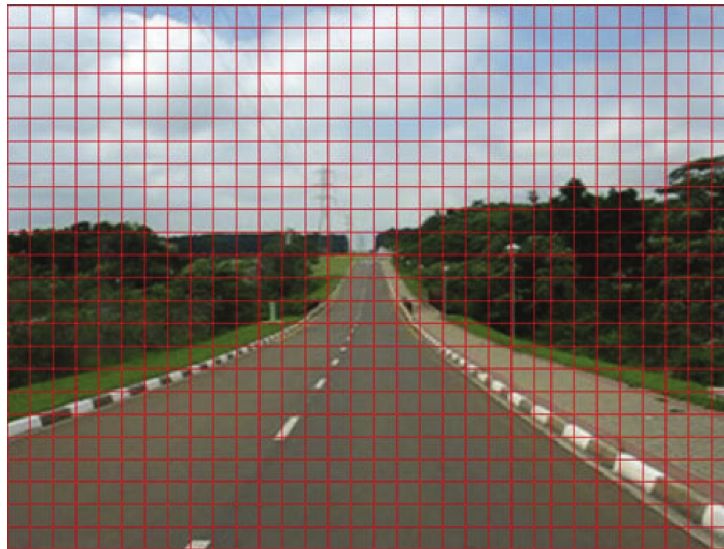


Figura 5.8: Imagem com os blocos usados na classificação (Shinzato e Wolf, 2011).

Tabela 5.1: Atributos de entrada das diferentes RNAs usadas como classificadores de áreas navegáveis ou não, de (Shinzato e Wolf, 2011). Componentes R, G, B são respectivamente vermelho (*red*), verde (*green*) e azul (*blue*). Componentes H, S, V são respectivamente brilho (*hue*), saturação (*saturation*), e valor (*value*). Y, U, V = Luminância-Crominância.

| RNAs | Atributos de entrada |
|------------------|---|
| RNA ₁ | Médias U, V e H, Média normalizada B, Energia normalizada G, Entropia H |
| RNA ₂ | Médias U, V, H, R e G, Médias normalizadas B e G, Energia normalizada G, Entropia H e Y |
| RNA ₃ | Médias U, V, H e S, Médias normalizadas B e G, Variância B, Entropia G normalizada |
| RNA ₄ | Médias U, V e H, Média normalizada B, Entropia H, Entropia normalizada G |
| RNA ₅ | Médias U, V, H, R e G, Médias normalizadas B e G, Energia G normalizada, Entropia H e Y |
| RNA ₆ | Médias U, V, H e S, Médias normalizadas B e G, Variância B, Entropia normalizada G |

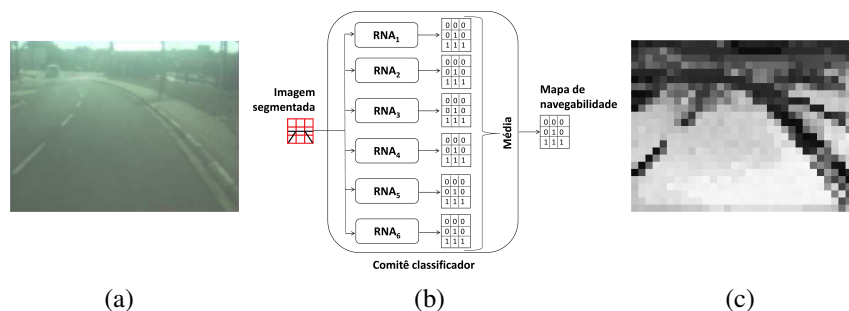


Figura 5.9: (a) Imagem obtida da câmera. (b) Comitê de redes neurais para identificação de áreas navegáveis ou não. (c) Mapa de navegabilidade: tons mais claros representam zonas mais seguras, tons mais escuros representam zonas inseguras (não navegáveis).

backpropagation (rprop). As topologias consistem de três camadas, uma entrada, uma camada oculta com 5 neurônios, e a camada de saída. A rede é treinada para classificar padrões em 0 ou 1. A escolha por uma coletânea de pequenas redes com entradas diferentes, ao contrário de uma rede grande com todas as entradas é justificada por Shinzato e Wolf (2011) por manter um número menor de neurônios na camada oculta, tornando o aprendizado mais rápido. Além disto, o comitê de redes neurais apresenta um comportamento mais robusto, pois cada rede pode “ajudar” as demais na classificação, suportando melhor as variações de luminosidade e sombras.

5.2.2 Identificação de geometrias

Após obter o mapa de navegabilidade, é realizada a identificação de cinco possíveis geometrias (*templates*) na pista, considerando a área frontal do veículo – área de visualização da câmera. A Figura 5.10 apresenta as geometrias utilizadas. Para cada geometria, é calculado o percentual de ocupação (*score*) – quantidade de quadros navegáveis no *template* dividido pelo total de quadros no *template*; o que resulta num valor real, entre 0 e 1.



Figura 5.10: Cinco possíveis geometrias definindo as direções que o veículo pode seguir. Da esquerda para a direita: curva acentuada para a esquerda, curva leve a esquerda, em frente, curva leve a direita, curva acentuada a direita.

Uma primeira tentativa de treinar a rede neural com o percentual de ocupação (valor real) não apresentou capacidade de aprendizado satisfatório. Assim, uma abordagem considerando um limiar (*threshold*) sobre o percentual de ocupação foi realizado, transformando o valor real entre 0 ou 1 em um valor lógico com base em um limiar. Assim, usamos 0 para significar área não navegável e 1 para significar área navegável.

5.2.3 Navegação baseada em aprendizado

O controle de navegação é realizado por uma rede neural que pode ser vista na Figura 5.11. Esta rede está esquematizada como item (c) na Figura 5.7. A rede tem como entradas a diferença entre a orientação do veículo e a orientação para o destino e os valores de cada *template* obtidos na etapa de verificação de geometrias – áreas obstruídas ou não. As saídas são (i) ângulo de esterçamento e (ii) velocidade.

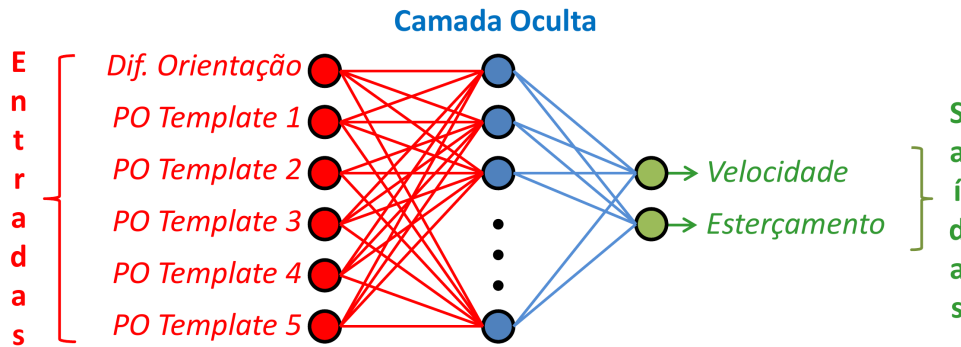


Figura 5.11: Estrutura da rede neural utilizada para a aceleração e esterçamento do veículo (Souza *et al.*, 2013). PO significa percentual de ocupação. Diferença de orientação é a diferença entre a orientação do veículo e a orientação para o destino (coordenada GPS). A orientação para o destino também é chamada de azimute.

Avaliamos a rede com diferentes quantidades de neurônios e diferentes funções de transferência (também chamadas de função de ativação). O uso de funções de ativação do tipo *logistic sigmoid* (LS) e *hyperbolic tangent* (HT) dão capacidade de saída não linear aos neurônios. O algoritmo de aprendizado utilizado é o *rprop*. Os resultados são detalhados na próxima seção.

5.2.4 Resultados

Os experimentos foram realizados utilizando o *hardware* disponível no LRM/USP (2013). No período dos experimentos, o *hardware* disponível era o veículo CaRINA I (Figura 5.1(a)) equipado com uma câmera *VIDERE DSG*, um controlador de motor *ROBOTEQ AX2580* para o esterçamento, uma bússola *TNT Revolution GS* para orientação do veículo, um GPS *GARMIN 18X-5Hz* para localização e um circuito baseado em Arduino para aceleração. A aquisição das imagens é realizada com a resolução de 320 por 240 píxeis. Fernandes *et al.* (2012) apresenta uma coletânea de trabalhos utilizando o robô CaRINA I, onde é detalhado o desenvolvimento dos *drivers* controladores.

Esta seção tem por objetivo descrever as avaliações realizadas com o treinamento da rede neural utilizada como controlador de navegação (Seção 5.2.3), além disso, descrevemos aspectos gerais de funcionamento relacionado ao modelo completo do controlador de navegação (Figura 5.7).

Dois ambientes diferentes foram escolhidos para a avaliação da navegação autônoma, mostrados na Figura 5.6. O primeiro trajeto tem ≈ 200 m e o segundo $\approx 1,08$ km. Foram utilizadas sete coordenadas GPS em cada trajeto. O número sete, nestes experimentos, é apenas uma coincidência, visto que o sistema trabalha com uma lista de coordenadas em formato texto. Na Figura 5.6(b) as linhas vermelhas entre P0 e P1 e entre P1 e P2 mostram o azimute, i.e. a

orientação que o veículo “realmente desejaria” realizar. Entretanto, por conter regiões não navegáveis, o veículo percorre o trajeto como apresenta a linha amarela (devido a sua capacidade de desvio de obstáculos, pensando em uma área não navegável como obstáculo). A trajetória do veículo, representada pelas linhas amarelas nas Figuras 5.6(a) e 5.6(b), é uma ilustração deste trajeto final, porém não representa um plot de coordenadas exatas do experimento.

Tabela 5.2: Resultado das avaliações considerando diferentes topologias e diferentes funções de transferências (*Logistic Sigmoid* – LS e *Hyperbolic Tangent* – HT). O erro é apresentado como MSE (10^{-1}). Também é apresentado o resultado do Ponto Ótimo de Generalização (POG) de cada execução das RNAs. Asterisco apresenta os melhores (menores) valores.

| | | | Exec 1 | Exec 2 | Exec 3 | Exec 4 | Exec 5 | Média | Desv. Pad. |
|-------------|----|------|--------|--------|--------|--------|--------|--------|------------|
| Topologia 1 | HT | Erro | 0,412 | 0,342 | 0,374 | 0,339 | 0,331 | 0,359 | 0,03 |
| | HT | POG | 1400 | 2300 | 1000 | 2000 | 2600 | | |
| 6x3x2 | LS | Erro | 0,527 | 0,392 | 0,375 | 0,541 | 0,486 | 0,464 | 0,08 |
| | LS | POG | 600 | 3000 | 1400 | 700 | 1000 | | |
| Topologia 2 | HT | Erro | 0,327 | 0,341 | 0,215 | 0,540 | 0,322 | 0,349 | 0,12 |
| | HT | POG | 2200 | 700 | 1900 | 9700 | 26400 | | |
| 6x6x2 | LS | Erro | 0,249 | 0,326 | 0,325 | 0,367 | 0,341 | 0,321 | 0,04 |
| | LS | POG | 7600 | 1200 | 8300 | 1700 | 5600 | | |
| Topologia 3 | HT | Erro | 0,316 | 0,332 | 0,365 | 0,235 | 0,429 | 0,335 | 0,07 |
| | HT | POG | 1100 | 1100 | 800 | 800 | 400 | | |
| 6x9x2 | LS | Erro | 0,377 | 0,501 | 0,458 | 0,505 | 0,319 | 0,432 | 0,08 |
| | LS | POG | 800 | 600 | 500 | 900 | 2300 | | |
| Topologia 4 | HT | Erro | 0,404 | 0,390 | 0,296 | 0,360 | 0,390 | 0,368 | 0,04 |
| | HT | POG | 400 | 1400 | 800 | 300 | 7400 | | |
| 6x12x2 | LS | Erro | 0,284 | 0,380 | 0,375 | 0,205 | 0,565 | 0,361 | 0,13 |
| | LS | POG | 91800 | 15200 | 1000 | 23600 | 1400 | | |
| Topologia 5 | HT | Erro | 0,323 | 0,200 | *0,188 | 0,351 | 0,299 | *0,272 | 0,07 |
| | HT | POG | 2600 | 76100 | 10600 | 2500 | 1000 | | |
| 6x15x2 | LS | Erro | 0,423 | 0,460 | 0,400 | 0,532 | 0,319 | 0,426 | 0,08 |
| | LS | POG | 400 | 600 | 1300 | 400 | 700 | | |

Utilizando o primeiro ambiente (Figura 5.6(a)), uma primeira bateria de avaliações foi realizada, considerando cinco diferentes topologias sendo cada topologia também avaliada com a mudança da função de transferência na camada oculta. A diferença básica entre as funções de transferência HT e LS é que a primeira mapeia o espaço de saídas entre -1 e 1 enquanto a segunda mapeia o espaço de saída entre 0 e 1 .

Os resultados podem ser vistos na Tabela 5.2. Podemos ver que a topologia que apresentou o resultado com menor erro (média de 5 execuções) é a Topologia 5 (6x15x2) utilizando função de transferência HT. A Figura 5.12 apresenta os histogramas dos erros de saída (valor obtido - valor esperado) considerando as duas melhores redes neurais – Topologia 4 (execução 3, HT) e Topologia 5 (execução 4, LS).

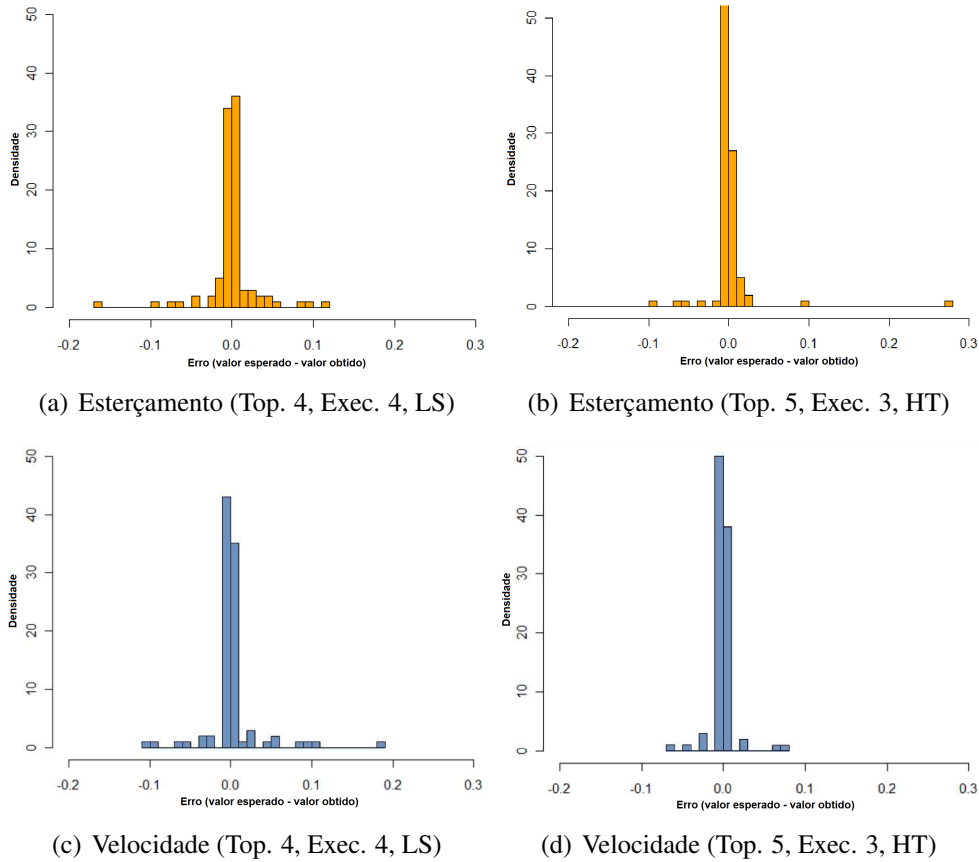


Figura 5.12: Histograma do erro das saídas (valor obtido - valor esperado) considerando as melhores redes neurais das topologias 4 (execução 3, HT) e 5 (execução 4, LS).

Uma inferência estatística foi realizada sobre os dados de saída das RNAs a fim de verificar a similaridade entre os conjuntos, iniciamos verificando a normalidade das distribuições. Tais resultados podem ser vistos na Tabela 5.3. Nesta avaliação, usamos a melhor execução de cada uma das topologias (1 a 5) apresentadas na Tabela 5.2. Podemos ver que, com exceção do esterçamento para a Topologia 1 (valor-p superior a 0,05), todos os demais tem a hipótese de normalidade rejeitada (valor-p inferior a 0,05).

Tabela 5.3: Valores-p do resultado de teste de normalidade de *Shapiro-Wilk*. Asterisco apresenta valor onde hipótese de normalidade não é rejeitada (valor-p maior que 0,05).

| | Esterçamento | Velocidade |
|-------------|--------------|------------|
| Topologia 1 | *0,246 | 0,000 |
| Topologia 2 | 0,000 | 0,000 |
| Topologia 3 | 0,000 | 0,000 |
| Topologia 4 | 0,000 | 0,000 |
| Topologia 5 | 0,000 | 0,000 |

A rejeição na hipótese de normalidade das distribuições (Tabela 5.3) implica no uso de um teste não-paramétrico para avaliação estatística da similaridade dos conjuntos – o teste de *Wilcoxon-Mann-Whitney*. A Tabela 5.4 apresenta os valores-p do resultado do teste de *Wilcoxon-Mann-Whitney* para comparação dos conjuntos. Foi feita a comparação entre a Topologia 5 (suposta melhor) com as demais topologias. Podemos ver que a hipótese de semelhança entre as distribuições não é rejeitada (valor-p superior a 0,05). O único conjunto que teve a hipótese de semelhança rejeitada foi o esterçamento da Topologia 1, ou seja, a Topologia 1 tem rejeição na hipótese de igualdade com a topologia 5, as demais não tem rejeição na hipótese de igualdade. A Figura 5.13 apresenta um gráfico de linha com valores esperados e obtidos pela RNA da Topologia 5, considerando o primeiro trajeto (mostrado na Figura 5.6(a)) – valores da base de validação limitados na primeira centena para fins de melhor visualização.

Tabela 5.4: Valores-p do resultado do teste de Wilcoxon-Mann-Whitney para verificação de similaridade de conjuntos. Asterisco apresenta valor onde existe rejeição da hipótese de igualdade (valor-p inferior a 0,05)

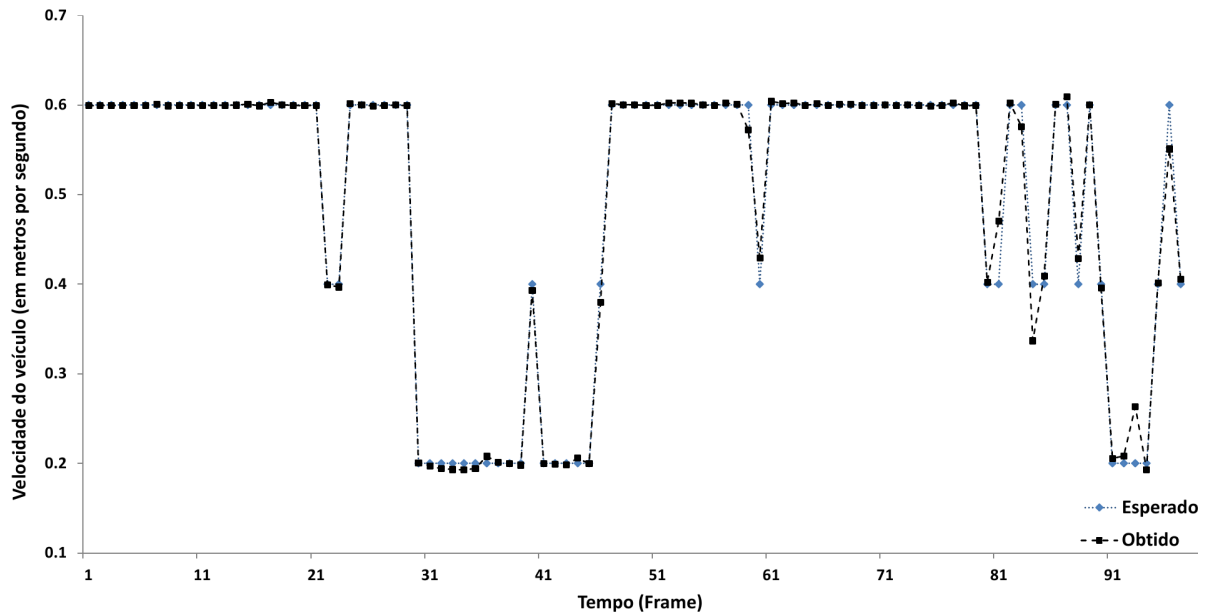
| | Topologia 1 | Topologia 2 | Topologia 3 | Topologia 4 |
|----------------------------|-------------|-------------|-------------|-------------|
| Topologia 5 (Esterçamento) | *0,031 | 0,115 | 0,397 | 0,114 |
| Topologia 5 (Velocidade) | 0,464 | 0,403 | 0,715 | 0,585 |

Utilizando o segundo ambiente (Figura 5.6(b)), realizamos duas avaliações com novos treinos considerando as topologias 4 e 5. A Figura 5.14 apresenta os histogramas do erro (valor esperado menos valor obtido). O objetivo destas imagens é a visualização da dispersão. O teste de *Wilcoxon-Mann-Whitney* também foi realizado e mostrou que a hipótese de igualdade não é rejeitada considerando 95% de confiança. A Tabela 5.5 apresenta o sumário dos resultados dos novos treinos das topologias 4 e 5 para o segundo trajeto.

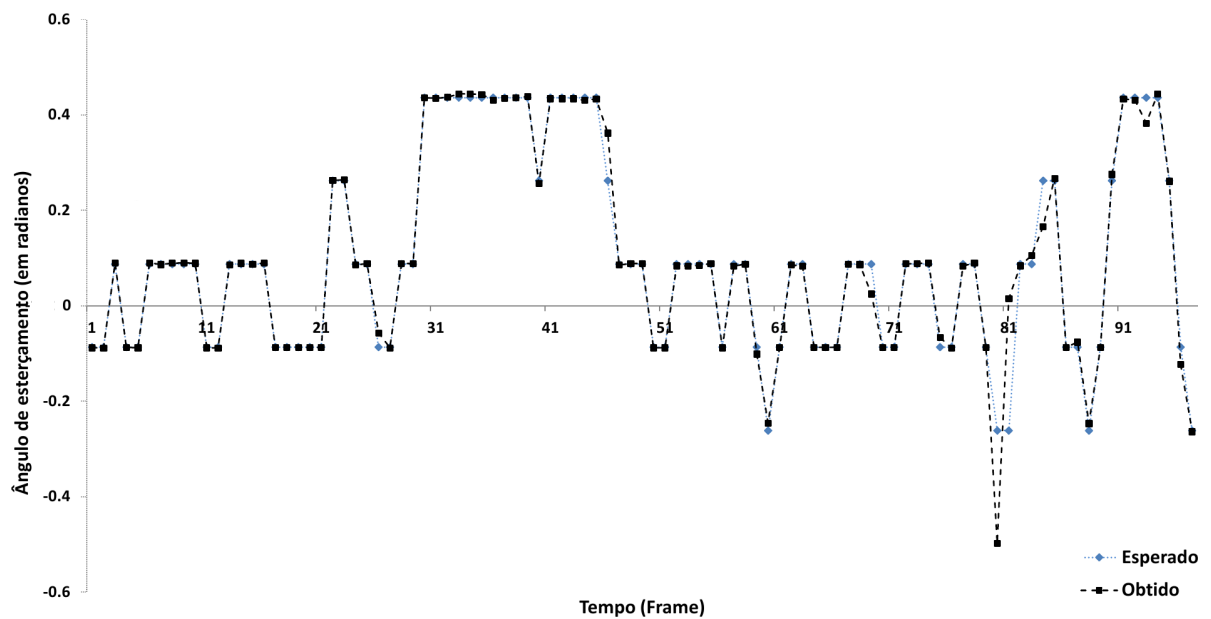
Tabela 5.5: Sumário dos resultados dos novos treinos das Topologias 4 e 5 para o segundo trajeto.

| | Velocidade | | Esterçamento | |
|-------------|------------|-----------|--------------|-----------|
| | Média | Desv. Pad | Média | Desv. Pad |
| Topologia 4 | 0,073 | 0,038 | 0,014 | 0,002 |
| Topologia 5 | 0,055 | 0,035 | 0,007 | 0,000 |

A Figura 5.15 apresenta um gráfico de linha com valores esperados e obtidos pela RNA das Topologia 4 e 5, considerando o segundo trajeto (mostrado na Figura 5.6(b)) – valores da base de validação limitados na primeira centena para fins de melhor visualização. O robô CaRINA I foi capaz de navegar entre as coordenadas GPS utilizando a câmera monocular para se manter na zona navegável. Entretanto, não foi possível até o momento fazer com que o veículo autônomo aprendesse a conduzir de modo adequado a partir dos dados coletados da condução deste por



(a)



(b)

Figura 5.13: Gráfico de valores esperados e obtidos pela RNA, considerando o primeiro trajeto (mostrado na Figura 5.6(a)) – valores da base de validação limitados na primeira centena para fins de melhor visualização. (a) Esterçamento (em radianos). (b) Velocidade (em m/s).

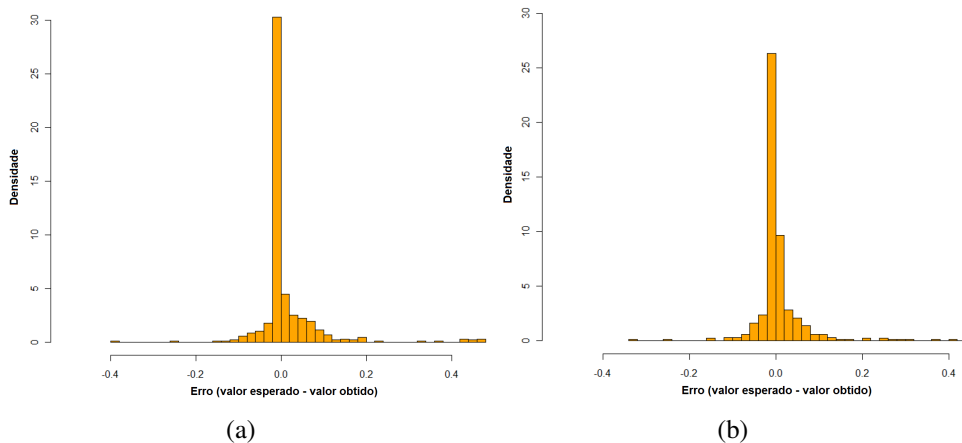


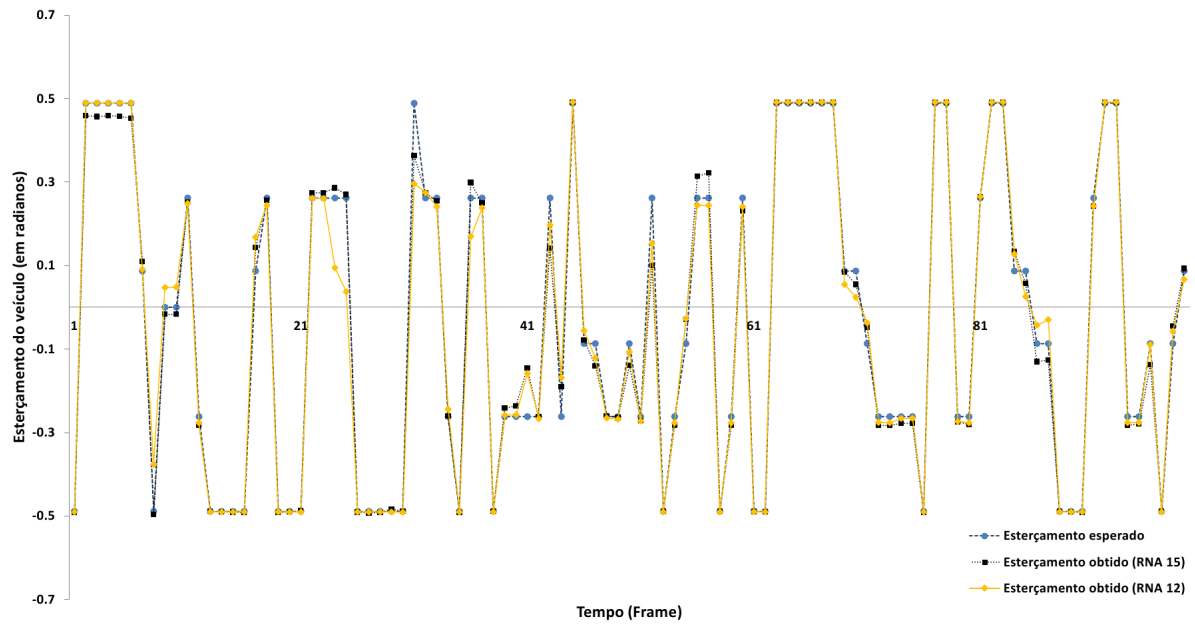
Figura 5.14: Histograma do erro das saídas considerando novos treinos usando as topologias 4 e 5 para o segundo trajeto. (a) Topologia 4. (b) Topologia 5.

um motorista humano. A rede neural aprendeu adequadamente a conduzir o veículo, porém a partir de dados provenientes de um sistema baseado em regras. O sistema baseado em regras foi criado apenas para validar a metodologia usando redes neurais, entretanto, dada a grande quantidade de ruídos nos dados obtidos enquanto conduzido por um humano, serão necessários diversos novos experimentos utilizando filtros sobre os dados lidos. Um sistema usando filtros já foi apresentado no Capítulo 4 e pode servir de inspiração para este desenvolvimento.

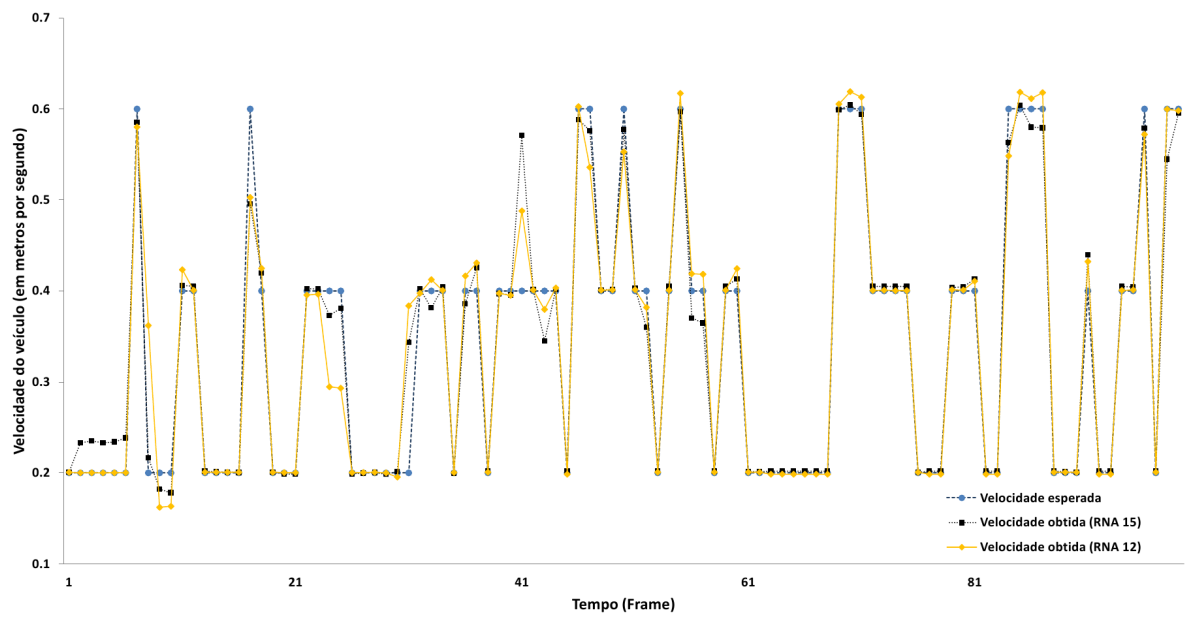
Outras dificuldades encontradas foram o erro alto do GPS utilizado e diversos problemas com mudanças na posição da câmera e mudanças na iluminação. Tais problemas estão sendo investigados no LRM. Entretanto, a pesquisa apresentada neste capítulo serve como uma etapa importante no desenvolvimento de um veículo autônomo de grande porte. Este trabalho, cuja detalhes estão descritos em (Souza *et al.*, 2013), desenvolvido em parceria com os demais colegas do LRM, teve uma importante contribuição e impacto técnico-científico, sendo inclusive considerado como o primeiro resultado mais relevante da condução de um veículo autônomo em vias urbanas em termos de Brasil.

5.3 Navegação com Mapeamento

O trabalho apresentado anteriormente teve relação com métodos de navegação com pouca informação do ambiente para fins de deliberação, pois não dependia de um mapa prévio detalhado do ambiente. Embora tenha sido utilizada uma lista de coordenadas de GPS que pode ser considerada uma espécie de mapa, tarefas mais complexas de navegação podem ser realizadas quando utilizando mapas mais completos, de acordo com Wolf *et al.* (2005) e Thrun *et al.* (2000).



(a)



(b)

Figura 5.15: Gráfico de valores esperados e obtidos pela RNA, considerando o segundo trajeto (mostrado na Figura 5.6(b)) – valores da base de validação limitados na primeira centena para fins de melhor visualização. (a) Esterçamento (em radianos). (b) Velocidade (em m/s).

O processo de mapeamento consiste na geração de modelos computacionais de cenários com base em informações de sensores (Hata *et al.*, 2009b). Os trabalhos (Hata *et al.*, 2009a,b, 2010) apresentam um esquema de mapeamento em desenvolvimento, baseado em sensor *laser* e com integração de outros sensores como unidade inercial (*inertial measurement unit* – IMU) e GPS. Um exemplo deste tipo de mapa pode ser observado na Figura 5.16.

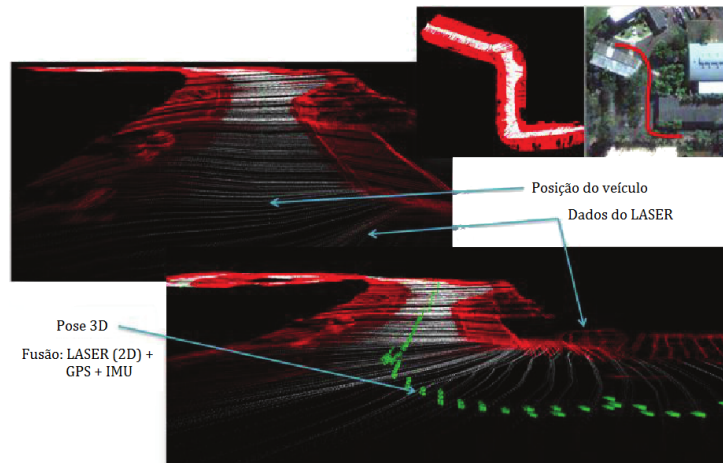


Figura 5.16: Dados de mapeamento (Fernandes *et al.*, 2012; Hata *et al.*, 2009a). Detecção de áreas navegáveis (branco) e não navegáveis (vermelho). Os pontos verdes representam a pose do veículo durante a navegação e captura dos dados – fusão dos dados de *laser*, IMU e GPS.

O desenvolvimento de métodos de mapeamento é tópico ativo de pesquisas também no LRM e tem como objetivo permitir maiores capacidades de planejamento de trajetória. É possível que um método de planejamento com base em mapas complexos seja utilizado integrado ao método de navegação proposto nas seções anteriores. Pelo planejamento de trajetórias, seria obtida uma lista de coordenadas de GPS, usadas como rota a ser percorrida pelo veículo; a qual seria seguida respeitando as regiões não navegáveis (obstáculos em geral) de uma forma reativa.

5.4 Considerações Finais

O capítulo apresentou o desenvolvimento e os resultados de pesquisas em navegação autônoma. Descrevemos uma investigação com objetivo de desenvolver um veículo de grande porte, abordando sucintamente aspectos de simulação que foram base para o desenvolvimento de sistemas de navegação em robôs reais de pequeno porte seguido do desenvolvimento e investigação com um robô de grande porte.

A investigação realizada apresentou uma arquitetura para navegação com capacidade de localização (pose atual e destino) e percepção local (área navegável e não navegável), baseado na

integração de uma câmera monocular e GPS. Tal arquitetura possui capacidade reativa de desvio de obstáculos (áreas não-navegáveis) e tem certo grau de deliberação, visto que é “atraída” por pontos de GPS. Como vimos anteriormente, é possível que nossa arquitetura seja estendida para que seja utilizada em conjunto com outros métodos que utilizem mapas mais complexos, desde que estes forneçam uma lista de coordenadas a serem seguidas. Navegação autônoma é um dos fundamentos necessários para o desenvolvimento das investigações apresentadas no próximo capítulo, sobre atividades com grupos de robôs.

A arquitetura se mostrou capaz de percorrer os trajetos propostos, evitando obstáculos (áreas não navegáveis). Entretanto, algumas restrições foram encontradas no modelo. A utilização de geometrias e a identificação de pista proposta por Shinzato e Wolf (2011) apresentou falta de capacidade de fazer o veículo se manter no centro da pista. Em certas situações, como em uma pista dupla ou muito larga, o veículo usa “toda” a área navegável livremente, inclusive mudando de posição e de faixa, sem que isto seja realmente desejado. Melhorias neste aspecto estão sendo consideradas como trabalhos futuros.

O desenvolvimento dos trabalhos (Pessin, 2008; Pessin *et al.*, 2010a) mostrou uma rede neural com capacidade para realizar a navegação com desvio de obstáculos com um robô simulado. Este capítulo estendeu a pesquisa em navegação e mostrou as adaptações necessárias com relação ao uso de um robô real e para o uso de uma câmera monocular. Em comparação ao modelo simulado, que utilizava 5 sensores *laser* simulados, o novo modelo com o robô de grande porte utilizou 5 entradas de distância na rede neural (5 geometrias), enquanto em (Pessin, 2008; Pessin *et al.*, 2010a) foi utilizada também uma rede com 5 neurônios para as entradas, relacionadas não com geometrias livres ou ocupadas, mas com a distância percebida pelos sensores. Ou seja, mantivemos o mesmo número de entradas relacionadas com a percepção de distância/obstáculos. Com relação à escolha do movimento (navegação ou desvio de obstáculos) o mesmo modo se manteve – o robô é “atraído” por um ponto de GPS e tem capacidade reativa para desviar de obstáculos no trajeto. O recebimento das coordenadas é similar também, dado por um arquivo texto contendo coordenadas de latitude e longitude. Uma diferença entre os modelos é a entrada, na rede neural, relacionada com o destino (em termos de orientação). Em (Pessin, 2008; Pessin *et al.*, 2010a), a rede neural usa como entrada a orientação do robô (0,0 a 360,0) e a orientação para o alvo (azimute, também de 0,0 a 360,0). Neste novo modelo, fazemos com que a entrada seja a diferença entre as orientações e também fazemos com que a diferença seja representada por -180,0 a 180,0. Tal mudança passa o modelo de um sistema de orientação absoluto para um modelo relativo – o valor 0,0 que antes representava o norte passa a representar a frente do robô). A rede neural, neste novo modelo, apresenta treinamento satisfatório com um número menor de épocas.

Diversas topologias foram avaliadas e não apresentaram rejeição na hipótese de igualdade. Embora não tenha sido possível até o momento fazer com que o robô aprenda com um motorista humano – o que é uma investigação corrente no LRM – a rede neural aprende com os dados provenientes de um sistema baseado em regras. É importante o desenvolvimento de métodos ou filtros que permitam o aprendizado da rede neural com dados provenientes da condução do veículo por pessoas; o atual modelo, com base em regras, apresenta a dificuldade de que, para cada diferente conjunto de sensores, conjuntos específicos de regras devem ser construídos. Esta é uma tarefa muito árdua e sujeita a erros. O uso de um método de aprendizado de máquina que aprenda com motorista humano deve diminuir esta dificuldade e tornar o sistema livre de possíveis erros em regras.

Outras dificuldades encontradas foram o erro alto do GPS utilizado e diversos problemas com mudanças na posição da câmera e mudanças na iluminação. Tais problemas estão sendo investigados no LRM/USP (2013) e devem ser resolvidos com a integração de mais sensores ao sistema. Apesar disso, a investigação descrita neste capítulo apresenta uma importante etapa do desenvolvimento de um veículo autônomo de grande porte.

5.5 Trabalhos Futuros

Foram vislumbrados, no decorrer desta investigação, uma série de trabalhos futuros relacionados com possíveis otimizações e novas funcionalidades, tanto em termos de acuracidade no aprendizado como na praticidade do uso do sistema proposto. São eles:

- Uma análise detalhada sobre filtros, a serem aplicados nos dados obtidos pelo veículo quando o mesmo for conduzido por um humano, a fim de permitir o aprendizado com base nestes dados.
- A integração (ou substituição) com sensores como câmera estéreo, *laser*, câmera térmica, radares, entre outros, a fim de proporcionar uma melhor percepção do ambiente e consequente melhoria no comportamento da navegação autônoma.
- A adaptação do modelo de navegação completamente autônoma para um modo de auxílio a motoristas, onde o sistema poderia ser utilizado para dar alertas ao motorista ou para atuação em casos de desatenção do condutor (como risco eminente de colisão).
- A investigação do modelo proposto em outros ambientes, como estradas de terra ou ainda fora de estrada, como em desertos, campos e alguns tipos de lavouras.

- O desenvolvimento de atividades específicas (estacionamento, ultrapassagem) e sua integração ao atual modelo proposto.
- A investigação e proposta de melhorias para suportar mudanças ambientais, como chuvas, neblina e mudanças repentinas (ou não) de iluminação.
- A fim de preservar a segurança, tanto do veículo, como dos pesquisadores, talvez um passo de grande importância seja a investigação e proposta de métodos para identificação de falhas e tratamento das mesmas de forma adequada.

Grupos de Robôs

Este capítulo está inserido no contexto da investigação do uso de técnicas de otimização para organização e coordenação de grupos de robôs para atuação em atividades críticas. O objetivo do capítulo é descrever a proposta, o desenvolvimento, a utilização e a avaliação de modelos que permitam graus de autonomia e eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis.

O grupo de robôs tem sua atuação em uma atividade crítica, simulada como um incêndio florestal. Uma equipe de robôs autônomos deve trabalhar cooperativamente a fim de realizar com sucesso a identificação e o combate ao incêndio, assim, se fez necessário o desenvolvimento de um ambiente de simulação capaz de suportar características fundamentais para a operação. Buscamos, dentro do possível, tornar o simulador realístico, com terrenos irregulares, diferentes tipos de vegetação, diferentes velocidades e orientações de vento e restrições físicas (com simulação de falhas) na criação e no uso dos robôs móveis. Neste ambiente de simulação é realizada a operação multirrobótica propriamente dita, que depende essencialmente de duas etapas: *planejamento e ação*. Detalhes do simulador e das etapas serão descritos na Seção 6.1.

O *planejamento da formação*¹ da estrutura do grupo de robôs se dá por meio de técnicas de otimização em uma unidade monitora. A *divisão de tarefas* se dá de forma implícita, pela função de aptidão desenvolvida e adotada no algoritmo evolutivo. A *coordenação* se dá devido

¹Formação, neste sentido, tem relação com a disposição espacial (posicionamento) dos robôs tanto no início da operação de combate como no fim da mesma.

a comunicação entre as unidades de combate e a unidade monitora e o possível replanejamento das operações.

Os trabalhos de formação de grupo apresentados neste capítulo são trabalhos que estendem e aperfeiçoam avaliações iniciais com grupos de robôs apresentadas detalhadamente na dissertação de mestrado de Pessin (2008). As publicações que advém destes trabalhos mais recentes, realizados durante o período do doutorado, podem ser vistas em (Pessin *et al.*, 2010a; Pessin e Osório, 2009a,b,c, 2010; Pessin *et al.*, 2010b,c, 2009).

As principais contribuições deste capítulo são:

- Descrição e avaliação de uma arquitetura multirrobótica para combate de incêndios florestais por meio de métodos de otimização, considerando autonomia e buscando eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis.
- Descrição e proposta de um simulador de incêndio florestal.
- Descrição e avaliação de um método de tolerância a falhas na operação multirrobótica.
- Avaliações buscando otimizar o método proposto por meio da investigação de diversas características dos algoritmos utilizados.
- Avaliações e descrição das características com quatro métodos de otimização: Algoritmos Genéticos, Otimização por Enxame de Partículas, *Hill Climbing* e *Simulated Annealing*.
- Proposta e discussão de possíveis alternativas às abordagens utilizadas.

O restante do capítulo é dividido nas seguintes seções: inicialmente, são apresentadas a descrição geral do sistema, detalhando a arquitetura proposta e os detalhes do simulador. Em seguida, apresentamos como as técnicas de otimização de (i) Algoritmos Genéticos, (ii) Otimização por Enxame de Partículas, (iii) *Hill Climbing* e (iv) *Simulated Annealing* podem ser aplicadas no problema. Ainda, apresentamos como uma heurística baseada em conceitos de AGs dinâmicos pode auxiliar na capacidade de tolerância a falhas do sistema. Finalizando, apresentamos possíveis alternativas às abordagens utilizadas, discutindo teoricamente como abordagens baseadas em etologia² e baseada em engenharia¹ poderiam ser aplicadas no problema.

²Baseada na nomenclatura de Nolfi e Floreano (2000).

6.1 Descrição Geral

Em uma área de floresta, como um parque ou uma reserva nacional de proteção ao meio ambiente ou ainda em plantações (*e.g.* algodão, milho, soja), incêndios relativamente aleatórios podem ter início. Incêndios podem ter início de forma relativamente aleatória devido a causas naturais (*e.g.* raios) e/ou artificiais (*e.g.* causados por humanos). Uma operação de combate a incêndio³ pode ser mais eficiente quando temos um método rápido de identificação do foco inicial. Sendo assim é aconselhável a existência de uma ou mais unidades de monitoração de focos de incêndio. Propomos o uso de uma unidade de monitoração que, ao identificar um foco de incêndio, tenha capacidade de *planejar* a formação de atuação (estratégias de combate) para a quantidade de unidades de combate a incêndios disponíveis.

Indo de encontro com a sugestão de Batista (2004), a unidade monitora proposta inicialmente faz uso de informações como as captadas por satélites (como as informações mostradas nas Figuras B.2(a) e B.2(b)). Ao identificar um foco, a unidade dispara o módulo de planejamento (evolução de estratégias). Ao finalizar a evolução, envia uma mensagem, passando para cada agente de combate: “*agir*(x_i, y_i)(x_f, y_f)”, então cada agente de combate deve se deslocar de sua posição atual até a posição (x_i, y_i) e começar a criar um aceiro⁴ até a posição (x_f, y_f). Os agentes de combate inicialmente ficam em uma base de operação. O método de atuação de combate é, como sugerido por LIF/UFPR (2013) o de criação de aceiros.

A Figura 6.1 apresenta o diagrama da operação; a seta entre o ambiente e o agente monitor representa a capacidade do agente monitor perceber o ambiente. As setas entre o ambiente e os agentes de combate representam a capacidade destes agentes perceberem por meio de sensores e atuarem no ambiente. A seta entre o agente monitor e os agentes de combate representam a capacidade de comunicação da ordem de execução. Desta forma, o sistema proposto possui duas fases principais: a primeira é o planejamento da estratégia de ação usando técnicas de otimização; a segunda é a execução da ordem do planejamento por robôs de combate, como apresenta a Figura 6.2.

Cada fase do sistema proposto, cujos modelos são detalhados nas próximas seções, foram implementadas em protótipos. O primeiro protótipo – que possui o mapa em 2D – é referente a Fase 1: realiza a evolução das estratégias usando um método de otimização, não necessita obrigatoriamente de visualização, porém está implementado com a possibilidade de uma saída

³Uma compilação de trabalhos sobre incêndios, combustíveis florestais, técnicas e equipamentos reais de operação em combate a incêndios pode ser vista no Anexo B. As informações apresentadas neste anexo foram utilizadas como base do desenvolvimento do simulador de incêndio e da proposta de combate de incêndio apresentada neste capítulo.

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

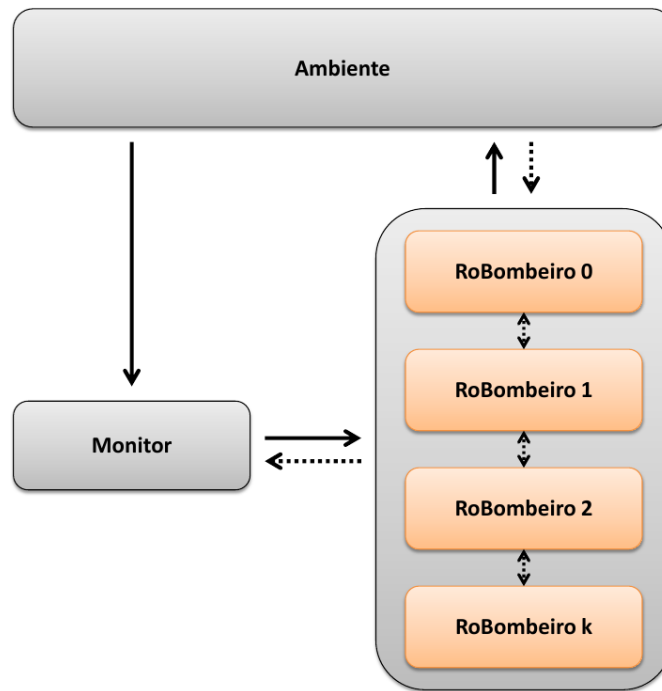


Figura 6.1: Diagrama da operação multi-agente.

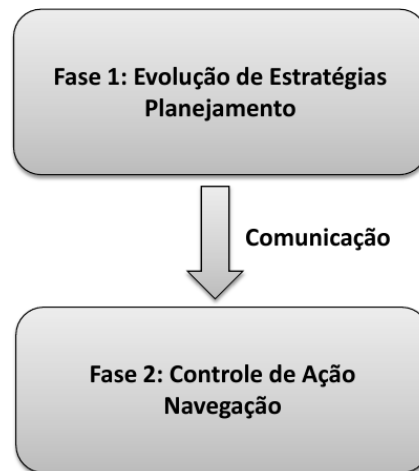


Figura 6.2: Fases do sistema proposto.

gráfica 2D desenvolvida com SDL (SDL, 2013). Uma imagem deste protótipo pode ser vista na Figura 6.5(a).

O segundo protótipo – que possui o mapa em 3D – é referente a Fase 2: suporta a ação dos robôs bombeiros, é simulado em um ambiente tridimensional e usa a biblioteca OSG (OSG, 2010), responsável pela saída gráfica, a biblioteca *Demeter* (Fowler, 2010), responsável pelo terreno irregular e a biblioteca ODE (Smith, 2006), responsável pelo realismo físico, tanto da morfologia robótica como da colisão entre os objetos presentes no ambiente (*e.g.* robôs, árvores,

inclinação de terreno). O uso da biblioteca ODE permite que os robôs, fisicamente simulados, respeitem questões como gravidade, inércia e atrito. Por exemplo, mantendo uma força f constante nos motores lineares (torque) um veículo terá velocidade v em regiões planas, em regiões de declive terá v maior e em aclives terá v menor. Uma imagem deste protótipo pode ser vista nas Figuras 6.5(e) e 6.5(f).

Ambos os protótipos possuem o mesmo modelo de simulação de propagação do fogo, simplificado em relação ao modelo proposto por Pessin *et al.* (2007a). A integração entre os protótipos se dá por meio de um arquivo texto; após realizar a evolução, o protótipo responsável pela estratégia cria um arquivo com as posições de atuação que é lido na inicialização do protótipo de ação no combate. Este arquivo texto simula a comunicação entre o agente monitor e os agentes de combate.

Para o desenvolvimento do RoBombeiros (Pessin, 2008; Pessin *et al.*, 2010a, 2007a) foi realizado um estudo extensivo sobre incêndios em ambientes naturais. Foi visto que o vento e o tipo de vegetação são os parâmetros de grande influência na direção e intensidade do fogo (Batista, 2004; CE, 2006; Koproski, 2005; LIF/UFPR, 2013; MA, 2006). Dados sobre técnicas reais de operação, combate, equipamentos e monitoração de incêndios florestais retirados de (Antunes, 2000; Batista, 2004; CE, 2006; CPTEC/INPE, 2007; Koproski, 2005; LIF/UFPR, 2013; Perera e Remmel, 2001) mostraram que equipes de combate usualmente são formadas por 2, 4 ou 8 pessoas e que, para incêndios de grande porte, um dos métodos de combate indicado é a criação de aceiros mecânicos. O uso de aceiros químicos é, em se considerando incêndios de grande porte, ainda um tópico de pesquisa na engenharia florestal. Como dito anteriormente, para incêndios em áreas com predomínio de vegetação herbácea, a criação de aceiros, em geral, faz uso de motoniveladoras (como a mostrada na Figura 6.3), onde optamos por este tipo de método de combate no nosso modelo.

6.2 Evolução de Estratégias

Como descrito no Capítulo 2, diversos métodos já foram apresentados na literatura com o objetivo de resolver problemas com o trabalho de grupos de robôs, usando as mais diversas diferentes abordagens e propondo soluções com graus diversos de eficiência e autonomia. Nossa pesquisa tem sido realizada em uma linha que tange a robótica evolucionária, mas onde a evolução ocorre em todo o grupo e não em cada um dos robôs de forma individual. Como detalhado na Seção 6.1, a aplicação do grupo de robôs é realizada no controle de incêndios florestais, por meio da criação de aceiros.



Figura 6.3: Motoniveladora (Caterpillar, 2013).

O mecanismo de planejamento usa uma técnica estocástica de otimização para definir as posições iniciais e finais de atuação de cada robô. Os algoritmos desenvolvidos buscam otimizar especificamente: (i) Posição inicial no combate, para cada robô do grupo (ponto de início da criação de aceiro) e (ii) Posição final no combate de cada robô do grupo (ponto final da criação de aceiro). Estas coordenadas são enviadas (mensagens de comando) para ativar as unidades de combate.

Para realizar a simulação, são necessários: (i) Conhecer a quantidade de robôs disponíveis; (ii) Conhecer a velocidade de operação dos robôs; (iii) Saber a posição inicial dos robôs; (iv) Possuir capacidade de simular a propagação de incêndio. Para simular a propagação de incêndio são necessários: (i) Obter a coordenada inicial do incêndio; (ii) Obter o sentido do vento; (iii) Obter a direção do vento; (iv) Obter um mapa de terreno e vegetação. Este conjunto de informações proposto pode ser totalmente obtido a partir de sensores.

Tabela 6.1: Estrutura proposta para solução (grupo com quatro robôs – ângulos e raios relativos ao ponto inicial do incêndio).

| Gene | Função | Valor Mínimo | Valor Máximo |
|------|--|--------------|--------------|
| 0 | Ângulo inicial do robô 0 | 0,0° | 360,0° |
| 1 | Ângulo final do robô 0 e inicial do robô 1 | 0,0° | 360,0° |
| 2 | Ângulo final do robô 1 e inicial do robô 2 | 0,0° | 360,0° |
| 3 | Ângulo final do robô 2 e inicial do robô 3 | 0,0° | 360,0° |
| 4 | Ângulo final do robô 3 | 0,0° | 360,0° |
| 5 | Raio inicial do robô 0 | 10,0 m | 100,0 m |
| 6 | Raio final do robô 0 e inicial do robô 1 | 10,0 m | 100,0 m |
| 7 | Raio final do robô 1 e inicial do robô 2 | 10,0 m | 100,0 m |
| 8 | Raio final do robô 2 e inicial do robô 3 | 10,0 m | 100,0 m |
| 9 | Raio final do robô 3 | 10,0 m | 100,0 m |

Propusemos inicialmente uma estrutura (cromossomo em se tratando de Algoritmos Genéticos) onde as posições de cada agente eram independentes. A execução de algumas simulações não apresentou resultados satisfatórios, assim, propomos uma nova estrutura onde a posição final de um agente é a posição inicial do próximo, como apresenta a Tabela 6.1. O tipo de atuação proposto (criação de aceiro) necessita que cada agente inicie sua operação exatamente na posição onde o agente anterior terminou a sua atuação. Assim, o uso de uma estrutura que considera a posição final de um agente com o a inicial do próximo é, de certa forma, uma simplificação para a simulação com o AG. Na estrutura apresenta na Tabela 6.1 estão presentes informações de todo o grupo dos agentes envolvidos, assim, o tamanho da estrutura é dependente da quantidade de robôs no sistema. Realizamos simulações considerando a existência de 4 agentes de combate.

As posições de atuação são calculadas aplicando no melhor indivíduo resultante da evolução as Equações 6.1 e 6.2.

$$x_f = x_a + r_i \times \cos(a_i) \quad (6.1)$$

$$y_f = y_a + r_i \times \sin(a_i) \quad (6.2)$$

Onde (x_f, y_f) é a coordenada da posição final dos robôs, (x_a, y_a) é a coordenada da posição inicial do incêndio, r_i é o raio (genes 5 a 9 na Tabela 6.1) e a_i é o ângulo (genes 0 a 4 na Tabela 6.1). O raio, bem como o ângulo, são específicos para cada operação de cada robô (coordenada inicial e final de criação de aceiro).

Uma das vantagens de se utilizar um método de otimização para o ajuste das posições é que eles não necessitam de informações locais para a correção dos erros, ou seja, eles não necessitam de uma base com os dados de treinamento. No caso do combate de incêndios, estas informações seriam muito difíceis de serem obtidas. O aprendizado ocorre por meio da interação das possíveis soluções com o ambiente e o desenrolar da simulação.

6.2.1 Escolha do *fitness*

Os métodos de otimização avaliados neste trabalho são guiados por uma função de aptidão (*fitness*). O uso de diferentes métodos de otimização (*e.g.* AG, PSO, HC, SA) em geral, podem fazer uso da mesma função de aptidão, ou seja, não existe necessidade de alteração do *fitness* para os diferentes métodos de otimização. O que muda entre os métodos é como o processo de exploração do espaço de busca é realizado, mas não a função de aptidão em si.

No caso da avaliação realizada neste trabalho, aplicada em combate de incêndio florestal, a função de aptidão tem relação com a área de vegetação salva do fogo e com a taxa de utilização

das unidades de combate (robôs bombeiros). A função de aptidão desenvolvida pode ser vista na Eq. 6.3.

$$f = \sum aq + 20 \times \sum aa + 10 \times \sum dm^2 \quad (6.3)$$

Onde aq é o total de área queimada (busca minimizar a área queimada); aa é o total de área com aceiro (busca minimizar a área de trabalho dos robôs, evitando que sejam criados aceiros em áreas onde o fogo não atingiria); e dm é a diferença média entre os aceiros úteis de cada robô (busca minimizar a diferença entre a média geral de aceiros úteis em relação ao aceiro útil de cada indivíduo, assim, o tamanho das áreas de trabalho tende a se equalizar entre os diferentes robôs). Os valores 20 e 10 que multiplicam os fatores aa e dm servem para aumentar a importância dos mesmos na função (busca-se equalizar as importâncias). Estes valores foram obtidos de forma empírica por meio da execução de diversos experimentos com diferentes valores. O fator dm é obtido pelo somatório dos quadrados das diferenças; elevar ao quadrado evita problemas com obtenção de resultados negativos devido a subtração.

No caso específico deste trabalho, otimizar o *fitness* é minimizar o valor da função, ou seja, quanto menor o valor final, menos vegetação foi queimada, menos aceiro foi criado, e menor é a diferença entre o tamanho dos aceiros. Buscamos, nas simulações, minimizar o valor do *fitness*. Deste modo, realizamos a otimização das posições dos indivíduos da população que representam os valores a serem adotados como sendo a estratégia de atuação no combate.

6.2.2 Simulador de incêndio

O simulador de incêndio desenvolvido⁵ tem um princípio probabilístico⁶ de funcionamento (Lopes *et al.*, 2002; Zinck e Grimm, 2009). Os tipos de vegetação, bem como o controle da propagação do fogo se dão sobre uma matriz (oculta sob o terreno). Esta matriz possui, para cada área do terreno, o tipo de vegetação presente, assim, considerando orientação do vento, intensidade do vento e tipo de vegetação de uma área podemos construir a simulação de propagação do fogo. A velocidade de propagação é relativa e tem como inspiração parâmetros obtidos do trabalho de (Koproski, 2005). Quanto ao vento, tanto a sua intensidade como a sua orientação podem ser geradas aleatoriamente ou configuradas a partir de dados parametrizados pelo usuário. O tempo de permanência do fogo em uma área é relacionado diretamente ao tipo da vegetação presente. Ambos os protótipos possuem um mapa que simula a integração das informações de vegetação, topografia e comportamento de fogo detalhados no Anexo B. A

⁵Código-fonte disponível em <http://goo.gl/Ns513>. Vídeo disponível em <http://goo.gl/4IakJ>.

⁶Embora tendo um princípio probabilístico, é bom lembrar que o comportamento do incêndio pode ser idêntico em diferentes rodadas quando mantendo uma semente aleatória constante.

criação dos mapas teve como base cartas topográficas e o mapa de modelos de combustíveis florestais, que podem ser vistos na Figura 6.4.

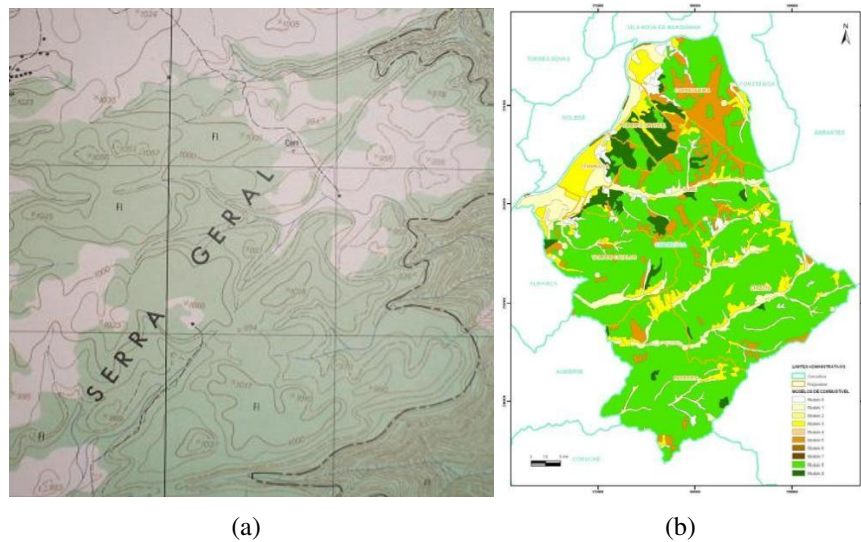


Figura 6.4: (a) Fragmento de carta topográfica de UTM (0578000, 6764000) até (0584000, 6770000) representando um quadrante de 6km x 6km. (b) Mapa de modelos de combustíveis (MA, 2006).

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

- Quantidade de combustível: de -1 a 13 , onde os valores de 1 até 13 representam tipos de combustível e suas respectivas propriedades como velocidade de propagação e tempo de permanência. O valor 0 significa que a área já queimou e o valor -1 significa área onde foi criado o aceiro; o protótipo usado na tese está implementado apenas com o Modelo 1 de combustível florestal, simplificado em relação ao descrito em Pessin *et al.* (2007c).
- Variáveis lógicas de controle de visualização do fogo.

Esta matriz não possui nenhuma informação relacionada ao posicionamento dos robôs. O terreno (mapa do protótipo 3D) possui coordenadas com dimensões de 15.000 por 15.000 e a matriz possui coordenadas com dimensões 600 por 600 , assim, o tamanho de cada quadra é de 25 por 25 unidades. No simulador 2D temos um ambiente com 280 por 280 coordenadas, onde cada coordenada representa um pixel e sendo possível pensar de forma relativa como cada pixel representando 1 metro quadrado de área de floresta.

O modelo de propagação de incêndio pode ser facilmente adaptado ou substituído por outra rotina de simulação de incêndio mais realista (considerando a umidade do ar e do solo, aclives e

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

6.3 Investigações em Formação de Grupos

Avaliações iniciais apresentadas em (Pessin, 2008) foram realizadas com o uso de Algoritmos Genéticos. Foi utilizado o modelo com *populações sobrepostas* (Jong, 2002). Para reduzir o espaço de busca foram utilizados *alelos* que limitam o conjunto de valores gerados para cada atributo, assim, o raio de atuação é de no mínimo 10,0 unidades de medida e de no máximo 100,0 unidades de medida e o raio fica entre 0,0 e 360,0. Foram adotados *genomas* do tipo real, otimizados para operações com números de ponto flutuante. Análises comparativas envolvendo representação binária e representação com ponto flutuante apresentaram que a representação em ponto flutuante apresenta vantagens significativas principalmente com relação a precisão e velocidade de convergência (Goldberg, 1991; Michalewicz, 1996). O esquema de seleção adotado foi o *Stochastic Remainder Sampling Selector* (SRSS), que segundo (Goldberg, 1989) possui um desempenho superior ao esquema da roleta (*Roulette Wheel Selector*).

Dois tipos de mutação foram utilizados em diferentes simulações com o Algoritmo Genético desenvolvido; inicialmente foi utilizada mutação do tipo Gaussiana e em seguida foi utilizada mutação do tipo Uniforme. Na mutação do tipo Gaussiana, a partir da identificação de um genoma razoável, são realizados preferencialmente pequenos ajustes até o final das gerações (genes são substituídos por números aleatórios de uma distribuição gaussiana). A mutação Uniforme substitui um gene por um número aleatório qualquer, levando a mudanças mais bruscas nos valores a serem otimizados, entretanto, ambas apresentaram capacidade de resolver o problema (Pessin, 2008).

Nesta seção, temos como objetivo apresentar os resultados de avaliações mais aprofundadas em relação a parâmetros do algoritmo genético desenvolvido. A lista completa da variação de parâmetros é apresentada na Tabela 6.2. Para verificar qual o melhor conjunto de parâmetros, fixamos as características ambientais da simulação do incêndio, bem como as posições iniciais dos robôs. Foram executadas 10 simulações com cada conjunto de parâmetros, totalizando 160

simulações e foram realizadas observações visuais em 20% dos resultados a fim de verificar quais os *fitness* ideais para o combate ao incêndio. Nestas simulações, o grupo de robôs é composto por quatro agentes que simulam a operação de atuação criando um aceiro mecânico.

Das formas de seleção avaliadas, a seleção por amostragem estocástica (*Stochastic Remainder Sampling Selector*) é feita em duas etapas, inicialmente uma nova população é preenchida com base no valor resultante da divisão do *fitness* de cada indivíduo pelo *fitness* médio da população; valores maiores que 1,0 são inseridos e as posições restantes são ocupadas com base no valor após a casa decimal, após, os indivíduos são selecionados para cruzamento por meio de seleção randômica (Michalewicz, 1996). Na seleção por torneio (*Tournament Selector*) é usado o método da roleta para obter n indivíduos, selecionando o que possuir o melhor *fitness*. Na *GAlib* (Wall, 1996) – biblioteca utilizada neste trabalho – o método de torneio implementado é de torneio de dois.

Tabela 6.2: Lista das variações de parâmetros avaliados no Algoritmo Genético para definição das posições de atuação.

| Conjunto | Seleção | Cruzamento | Tipo de mutação | Taxa de mutação |
|----------|---------|-------------|-----------------|-----------------|
| A | Torneio | Dois pontos | Uniforme | 10,00% |
| B | Torneio | Dois pontos | Uniforme | 50,00% |
| C | Torneio | Dois pontos | Gaussiana | 10,00% |
| D | Torneio | Dois pontos | Gaussiana | 50,00% |
| E | Torneio | Um ponto | Uniforme | 10,00% |
| F | Torneio | Um ponto | Uniforme | 50,00% |
| G | Torneio | Um ponto | Gaussiana | 10,00% |
| H | Torneio | Um ponto | Gaussiana | 50,00% |
| I | SRSS | Dois pontos | Uniforme | 10,00% |
| J | SRSS | Dois pontos | Uniforme | 50,00% |
| K | SRSS | Dois pontos | Gaussiana | 10,00% |
| L | SRSS | Dois pontos | Gaussiana | 50,00% |
| M | SRSS | Um ponto | Uniforme | 10,00% |
| N | SRSS | Um ponto | Uniforme | 50,00% |
| O | SRSS | Um ponto | Gaussiana | 10,00% |
| P | SRSS | Um ponto | Gaussiana | 50,00% |

Alguns dos resultados das simulações são mostrados na Figura 6.5. Considerando o ambiente proposto, depois de obtidos resultados de diversas simulações, foi possível observar que os piores valores (quando os robôs não eram capazes de extinguir o incêndio) ficavam perto de 23.000 a 25.000 unidades (não mais que 25.000). Quando os robôs apresentavam uma boa solução, a mesma ficava inferior a 3.500 unidades. Essas unidades são o resultado do *fitness* e podem ser interpretadas como a soma da área queimada de vegetação e da utilização das unidades de combate de acordo com o apresentado na Seção 6.2.1. Assim, para a apresentação dos resultados neste capítulo, os mesmos foram normalizados a fim de apresentar o erro máximo como 1,0 ao contrário de 25.000, sendo o valor limite para uma solução ser considerada de boa qualidade definido como 0,14 ou inferior. Estes valores são estritamente relacionados às ca-

racterísticas do ambiente proposto e precisariam ser reconfigurados no caso de serem adotados diferentes tamanhos de terrenos e quantidades de robôs.

A Figura 6.5 apresenta alguns resultados satisfatórios e não satisfatórios das simulações. As Figuras 6.5(a) e 6.5(b) apresentam resultados insatisfatórios, respectivamente com *fitness* de $\approx 0,24$ e $\approx 0,32$. As Figuras 6.5(c) e 6.5(d) apresentam resultados satisfatórios, com *fitness* inferior a 0,14.

As simulações realizadas, apresentadas na Tabela 6.2 não permitiram obter 100% dos resultados inferior a 0,14 para nenhum conjunto de parâmetros. Assim, uma nova bateria de testes foi realizada, utilizando os cinco melhores conjuntos, com aumento no número de indivíduos (de 100 para 150) e no número de gerações (de 500 para 700). A Tabela 6.3 apresenta o resultado deste segundo conjunto de parâmetros avaliados. Podemos ver que a experiência *I'* foi a única que permitiu obter 100% dos resultados finais com *fitness* abaixo de 0,14.

Tabela 6.3: Segunda lista das variações de parâmetros avaliados no Algoritmo Genético para definição das posições de atuação. Utilizamos como base os cinco melhores conjuntos da Tabela 6.2, com aumento no número de indivíduos (de 100 para 150) e no número de gerações (de 500 para 700). Experiência *I'* permitiu obter 100% dos resultados finais com *fitness* abaixo de 0,14.

| Conjunto | <i>Fitness</i> final < 0,14 | Média | Desvio |
|----------|-----------------------------|-------|--------|
| D' | 80% | 0,124 | 0,021 |
| E' | 80% | 0,126 | 0,028 |
| I' | 100% | 0,121 | 0,010 |
| K' | 40% | 0,151 | 0,042 |
| M' | 90% | 0,118 | 0,017 |

As experiências desta seção mostraram que o conjunto de parâmetros usando cruzamento de dois pontos, mutação uniforme, taxa de mutação de 10% e método de seleção por amostragem estocástica foi o que permitiu obter os melhores resultados (todos inferiores a 0,14).

6.3.1 Tolerância a falhas

Uma das capacidades esperadas do sistema é a possibilidade de atuação satisfatória no combate ao incêndio mesmo quando existam falhas nas unidades de combate. Desta forma, propomos um modelo de comunicação associado a uma heurística a fim de tornar o sistema capaz de combater o incêndio mesmo perdendo alguma unidade robótica. Simulamos operações de falhas em unidades robóticas a fim de realizar reorganização das estratégias de atuação. Estes resultados estão também apresentados em (Pessin e Osório, 2009b, 2010).

Com relação a reorganização da solução (capacidade de tolerância a falhas), as unidades robóticas simuladas são equipadas com capacidade de envio de mensagens. Na simulação da



Figura 6.5: (a) e (b) Resultados insatisfatórios, os aceiros são excessivamente grandes e mal distribuídos entre os robôs. (c) e (d) Resultados satisfatórios. (e) Visão detalhada da navegação dos robôs móveis. (f) Resultado satisfatório aplicado no protótipo 3D.

operação tática, configuramos um tempo de troca de mensagens entre os robôs e o agente coordenador (responsável pela evolução com o AG). Os robôs devem enviar mensagens com um certo *delay* para que o coordenador saiba de possíveis atrasos ou de falha completa de operação (similar ao conceito de *watchdog timer*). Assim, ao identificar um problema, o agente coordenador pode agir de duas formas: (i) reorganizar completamente o grupo desconsiderando um dos integrantes ou (ii) enviar novas unidades robóticas para atuar no fragmento da falha. Realizamos simulações com as duas formas.

A reorganização pode ser feita por meio do AG exatamente da mesma forma como foi feita a primeira organização, porém considerando possíveis modificações no ambiente causadas pelos robôs. Devido ao tempo para a evolução ser consideravelmente longo, propomos uma heurística a fim de agilizar a reorganização. Nesta heurística, propomos um conjunto de indivíduos novos com base no melhor indivíduo obtido pela primeira evolução.

A heurística proposta tem como princípio a criação uma população inicial com base no melhor indivíduo obtido pela primeira evolução. Na reorganização de 4 para 3 robôs (Figuras 6.6(a) e 6.6(b)), são obtidos os pontos iniciais e finais do aceiro e são calculadas as novas posições considerando a divisão da tarefa entre os três robôs, além disso, são inseridos outros indivíduos considerando acréscimo no ângulo e no raio da operação, este aumento é feito em 2%, 4%, 8%, 12%, 16%, 24%, 32%, 40%, totalizando 81 indivíduos que são usados como população inicial.

Para dois novos agentes (enviados para atuar no fragmento da falha – Figuras 6.6(c) e 6.6(d)) é usado o mesmo método com pequenas variações: considerando que existe falha em apenas um fragmento do aceiro, os novos indivíduos são gerados com base nos pontos iniciais e finais previstos da operação do robô que apresentou falha (e não no aceiro completo). Além disso, para novos dois robôs, o ponto de início de incêndio passa a ser considerado o ponto médio de onde ocorreu a falha (e não o ponto original do início de incêndio). A Figura 6.6 apresenta cenas com reorganização, podemos ver que na primeira geração, a heurística proposta já é capaz de combater o incêndio. A evolução, mesmo usando a heurística, é importante pois permite que o aceiro seja finamente ajustado às variações de vegetação.

O gráfico de evolução do *fitness* da reorganização para 3 robôs de combate pode ser visto na Figura 6.7, apresentando a média de 10 simulações (com os parâmetros da experiência *I'*). Ao contrário dos demais gráficos apresentados neste capítulo, este apresenta a normalização entre 0 e 1 com base nos valores máximos e mínimos obtidos especificamente por esta simulação, não sendo 1,0 o valor 25.000. A simulação da propagação do fogo considerou vento no sentido Leste-Oeste (270°) com velocidade relativa de 7 km/h; e velocidade de navegação relativa dos robôs de 35 km/h, sendo os robôs posicionados em uma base com distância de ≈ 2 km do ponto

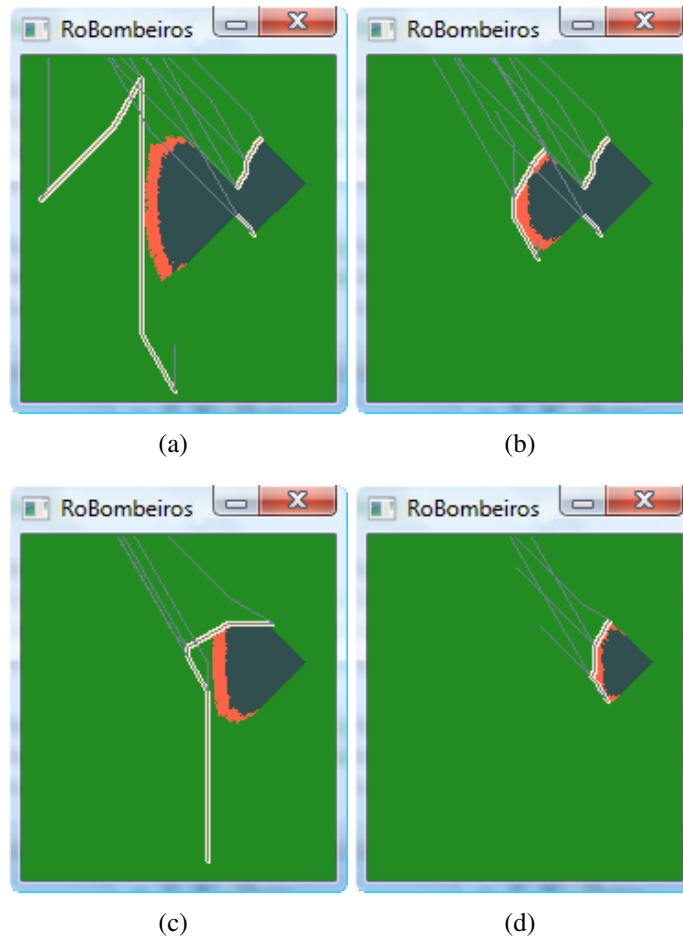


Figura 6.6: (a) e (b) Reorganização com envio de mais dois robôs: (a) Melhor indivíduo após primeira geração sem heurística e (b) Melhor indivíduo após primeira geração com heurística. (c) e (d) Reorganização para 3 robôs: (c) Melhor indivíduo após primeira geração sem heurística e (d) Melhor indivíduo após primeira geração com heurística.

inicial do incêndio. O gráfico mostra que o uso da heurística permite obter, já na população inicial, um indivíduo bastante apto a atuar no combate ao incêndio.

6.3.2 Buscando melhorias na eficiência do AG

Após as avaliações descritas na Seção 6.3, onde foram avaliados tipos de seleção, formas de cruzamento, tipos de mutação e valores de mutação, buscou-se realizar uma nova avaliação considerando outras duas questões relacionadas com diversidade de população e pressão seletiva. A diversidade da população e a pressão seletiva são fortemente relacionadas e têm impacto direto na eficiência da busca (Michalewicz, 1996). O aumento da pressão seletiva diminui a

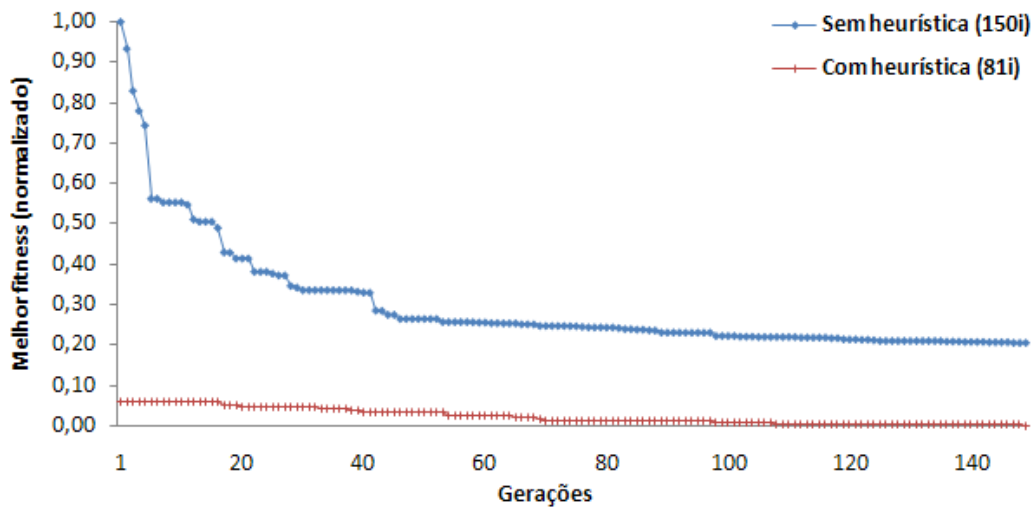


Figura 6.7: Evolução do *fitness* de acordo com o número de gerações (reorganização para 3 robôs de combate, com e sem heurística). O eixo x está limitado em 150 gerações.

diversidade da população, e vice-versa. A diversidade da população e a pressão seletiva tem relação com convergência prematura e com busca ineficiente. Assim, é importante existir um balanço entre estes parâmetros (Whitley, 1989). Dois fatores que tem forte influência na pressão seletiva e na diversidade da população são os conceitos de população sobreposta (*overlapping*) e de escalas de *fitness*.

Assim, nesta seção buscamos avaliar o comportamento do AG na busca de posições para atuação do grupo de robôs considerando diferentes formas de populações sobrepostas e de escalas de *fitness*, como apresenta a Tabela 6.4. Estes resultados foram apresentados em (Pessin *et al.*, 2010d).

Tabela 6.4: Lista de variações de parâmetros no AG considerando uso de populações sobrepostas e de escalas de *fitness*

| Conjunto | Método de escala | Sobreposição |
|-------------|------------------|----------------------|
| $A_1 - A_4$ | Sem escala | {Não; 25%; 50%; 75%} |
| $B_1 - B_4$ | <i>Linear</i> | {Não; 25%; 50%; 75%} |
| $C_1 - C_4$ | <i>Sigma</i> | {Não; 25%; 50%; 75%} |
| $D_1 - D_4$ | <i>Power</i> | {Não; 25%; 50%; 75%} |
| $E_1 - E_4$ | <i>Sharing</i> | {Não; 25%; 50%; 75%} |

A técnica de população sobreposta consiste em fazer um clone da população corrente. As operações genéticas de seleção, cruzamento e mutação são realizadas neste clone. Após a avaliação destes indivíduos – da população clonada – os indivíduos da população original são substituídos com certo percentual de sobreposição. Os piores indivíduos da população original são substituídos pelos melhores da população clonada, mantendo na população original uma certa quantidade de indivíduos originais, o que dá maior chance a um *fitness* bom de deixar

mais descendentes. A substituição de apenas uma parte da população diminui a diversidade da mesma, assim, o uso de populações sobrepostas deve ter convergência mais rápida. Entretanto, a exploração do espaço de busca é teoricamente mais restrita (Goldberg, 1989; Jong, 2002).

Escalas de *fitness* são realizadas por meio da aplicação de uma função matemática sobre o *fitness* original, de tal forma que o método de seleção não considere o valor de *fitness* original, mas sim o valor transformado (Whitley, 1989). Escalas de *fitness* tem aplicações principalmente em casos onde o valor obtido pela função pode ser pouco significativo para a busca, nestes casos, métodos de escala são sugeridos para aumentar ou diminuir a pressão seletiva sobre os indivíduos (Goldberg, 1989; Michalewicz, 1996; Whitley, 1989). No trabalho desenvolvido, além da avaliação do valor original do *fitness*, quatro tipos de escalas foram avaliados:

- Escala Linear (*Linear Scaling*): Normaliza o *fitness* com base no menor e no maior *fitness* da população. Pode ter um fator de escala. Usamos, de acordo com Goldberg (1989) o valor de 1,2.
- Escala Exponencial (*Power Law Scaling*): Mapeia o novo *fitness* com a aplicação de um expoente, de acordo com a equação $f'_i = f_i^k$, onde f_i é o *fitness* original e f'_i é o *fitness* escalado. Usamos $k = 1,005$ de acordo com o especificado por Wall (1996).
- Escala Sigma (*Sigma Truncation Scaling*): Normaliza usando a média e o desvio padrão da população, de acordo com a equação: $f'_i = f_i + (\bar{f} - c \cdot \sigma)$. Usamos $c = 2,0$ de acordo com Wall (1996).
- Escala por colaboração (*Sharing Scaling*): Reduz o *fitness* dos indivíduos que são semelhantes na população. Este método é usado para realizar especiação⁷.

Os resultados das simulações (média de 10 rodadas) pode ser visto na Figura 6.8. Podemos ver que apenas quatro conjuntos foram capazes de apresentar média e desvio padrão, para todas as simulações, com valores inferiores a 0,14. Estes conjuntos foram: {escala linear com 50% de sobreposição; escala *sharing* com 50% de sobreposição; sem escala com 75% de sobreposição e escala *sharing* com 75% de sobreposição}. Além disso, podemos ver que o resultado das simulações, para os conjuntos que não utilizaram populações sobrepostas, apresentaram resultados consideravelmente piores.

Uma vez obtidos quatro conjuntos com resultados satisfatórios (*fitness* inferior a 0,14), buscamos avaliar melhorias diminuindo a quantidade de indivíduos. Assim, realizamos uma nova bateria de simulações considerando os quatro conjuntos satisfatórios, porém, com redução do

⁷Processo evolucionário pelo qual novas espécies surgem.

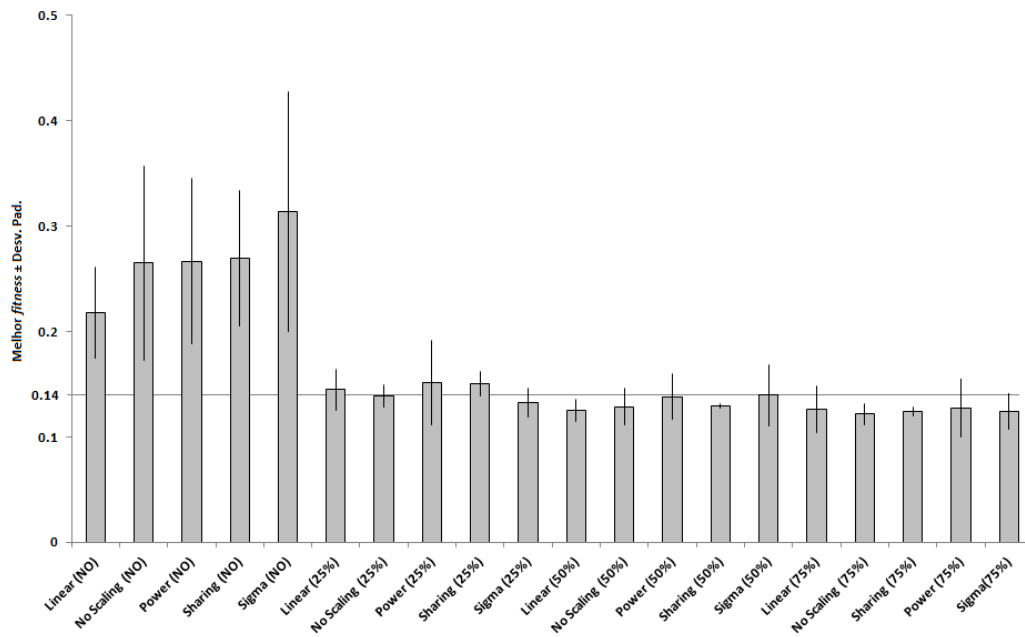


Figura 6.8: Resultados das avaliações apresentadas na Tabela 6.4. O eixo x apresenta a descrição do conjunto de parâmetros como (método de escala; percentual de sobreposição).

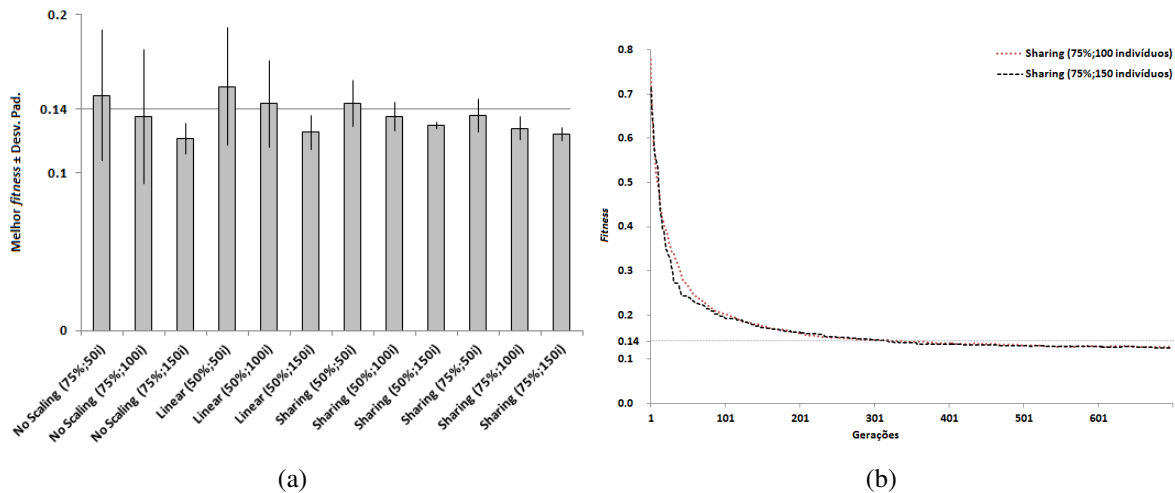


Figura 6.9: (a) Resultados das avaliações considerando os quatro melhores conjuntos de parâmetros da Figura 6.8 usando diferentes tamanhos de população. O eixo x apresenta a descrição do conjunto de parâmetros como (método de escala; percentual de sobreposição; número de indivíduos). (b) Gráfico de evolução do *fitness* de acordo com o número de gerações.

número de indivíduos de 150 para 100 e 50. O tamanho da população é um dos parâmetros mais importantes no processo de busca do AG (Michalewicz, 1996). Se a população for muito pequena, o AG pode convergir muito rápido (com resultado insatisfatório). Se a população for muito grande, pode existir desperdício de recursos computacionais.

Podemos ver na Figura 6.9(a) que, com a diminuição do número de indivíduos, os resultados das simulações pioram. Entretanto, mesmo usando 100 indivíduos no lugar de 150, um conjunto de parâmetro obteve 100% dos resultados com *fitness* inferior a 0,14. O conjunto de parâmetros que permitiu obter estes resultados é o que usou 75% de sobreposição e escala de *fitness* do tipo *sharing*. Nenhum outro conjunto permitiu obter todos os resultados com valores inferiores a 0,14 usando 100 indivíduos. A Figura 6.9(b) apresenta o gráfico de queda do *fitness* do melhor conjunto descrito no parágrafo anterior, podemos ver que usando 100 ou 150 indivíduos, as curvas são bastante similares.

Podemos perceber, a partir dos resultados das simulações, que os melhores resultados foram obtidos com sobreposição de 75% da população, ou seja, mantendo 25% dos indivíduos da população original a cada geração. Desta forma, a pressão seletiva⁸ não é muito alta. Aliada a esta característica, o uso de escala de *fitness* do tipo *sharing* tende a geração de população com grande diversidade. É importante ressaltar que esta característica é considerada boa neste trabalho, visto que o foco é a obtenção de um indivíduo final muito bom, independente das considerações sobre a média da população.

6.3.3 Otimização por enxame de partículas

Após as avaliações com AGs, buscamos avaliar como Algoritmos de Otimização por Enxame de Partículas se comportam quando aplicados à formação e atuação de grupos robóticos. A mesma configuração de ambiente foi utilizada nos trabalhos com Algoritmos Genéticos. Foram avaliados diversos parâmetros que influenciam o comportamento da otimização, como inércia, confiança, tipos de modelos sociais e tamanho de enxame. O conjunto de parâmetros avaliado pode ser visto na Tabela 6.5. Conjuntos {A..F} possuem apenas confiança no grupo (modelo social, $c_1 = 0,0$ e $c_2 = 2,0$). Conjuntos {G..L} possuem apenas confiança na própria partícula (modelo cognitivo, $c_1 = 2,0$ e $c_2 = 0,0$). Conjuntos {M..R} possuem confiança tanto no grupo como na partícula (modelo completo, $c_1 = 2,0$ e $c_2 = 2,0$). Resultados apresentados em (Pessin *et al.*, 2010c).

Dos conjuntos de parâmetros inicialmente avaliados (Tabela 6.5), somente os conjuntos {B,E} (modelo social com $w = 0,8$) e {M,P} (modelo completo com $w = 0,4$) obtiveram pelo

⁸Em se tratando de AGs, o conceito de pressão seletiva tem relação com a chance do indivíduo ser selecionado para gerar descendentes para a próxima geração.

Tabela 6.5: Conjunto de avaliações iniciais utilizando PSO.

| Conjunto | ConfP (c_1) | ConfG (c_2) | Inércia (w) | Partículas |
|----------|-----------------|-----------------|-----------------|------------|
| A | 0,0 | 2,0 | 0,4 | 80 |
| B | 0,0 | 2,0 | 0,8 | 80 |
| C | 0,0 | 2,0 | 1,2 | 80 |
| D | 0,0 | 2,0 | 0,4 | 160 |
| E | 0,0 | 2,0 | 0,8 | 160 |
| F | 0,0 | 2,0 | 1,2 | 160 |
| G | 2,0 | 0,0 | 0,4 | 80 |
| H | 2,0 | 0,0 | 0,8 | 80 |
| I | 2,0 | 0,0 | 1,2 | 80 |
| J | 2,0 | 0,0 | 0,4 | 160 |
| K | 2,0 | 0,0 | 0,8 | 160 |
| L | 2,0 | 0,0 | 1,2 | 160 |
| M | 2,0 | 2,0 | 0,4 | 80 |
| N | 2,0 | 2,0 | 0,8 | 80 |
| O | 2,0 | 2,0 | 1,2 | 80 |
| P | 2,0 | 2,0 | 0,4 | 160 |
| Q | 2,0 | 2,0 | 0,8 | 160 |
| R | 2,0 | 2,0 | 1,2 | 160 |

menos 50% dos resultados com *fitness* inferior a 0,14. Nenhum conjunto de parâmetros permitiu obter mais de 70% de resultados abaixo de 0,14. Assim, novos conjuntos de parâmetros foram propostos (Tabelas 6.6 e 6.7), sendo variações dos melhores quatro conjuntos obtidos na avaliação anterior. Além disso, aumentamos o número de gerações de 500 para 800, a quantidade de partículas (20, 50, 100 e 200) e a inércia ($\pm 0,1$). As simulações que usaram modelo cognitivo apresentaram resultados significativamente piores, assim, nenhuma avaliação com modelo cognitivo foi feita na segunda bateria de avaliações.

As Tabelas 6.6 e 6.7 apresentam a média de 10 rodadas para cada conjunto. Podemos ver que o único conjunto de parâmetros que permitiu obter 100% das simulações abaixo de 0,14 foi o conjunto F_{11} . O conjunto de parâmetros utilizado em F_{11} é $c_1 = 2,0$, $c_2 = 2,0$ (modelo completo), $w = 0,5$, 800 gerações e 200 partículas. É possível observar, nas Tabelas 6.6 e 6.7, que as experiências com 20 e 50 partículas apresentaram resultados inferiores, em comparação às experiências com 100 e 200 partículas, tanto no modelo completo (Tabela 6.7) como no modelo social (Tabela 6.6). Além disso, no modelo social (Tabela 6.6), pode-se observar que as experiências que usaram valor de inércia igual a 0,9 apresentaram resultados completamente insatisfatórios, independentemente da quantidade de partículas presentes no sistema. Outra informação que se pode observar é que as experiências com o modelo completo (Tabela 6.7) apresentaram resultados melhores em comparação com as experiências com o modelo social (Tabela 6.6).

A Figura 6.10(a) apresenta o gráfico das médias dos menores *fitness*, obtidos com base na bateria de avaliações descritas na Tabela 6.7. É apresentada a média de 10 simulações \pm desvio

Tabela 6.6: Novo conjunto de avaliações considerando PSO com modelo social.

| Conjunto | Partículas | Inércia | Resultados com $fitness < 0,14$ |
|----------|------------|---------|---------------------------------|
| S_0 | 20 | 0,7 | 0% |
| S_1 | 20 | 0,8 | 40% |
| S_2 | 20 | 0,9 | 0% |
| S_3 | 50 | 0,7 | 40% |
| S_4 | 50 | 0,8 | 70% |
| S_5 | 50 | 0,9 | 0% |
| S_6 | 100 | 0,7 | 60% |
| S_7 | 100 | 0,8 | 70% |
| S_8 | 100 | 0,9 | 0% |
| S_9 | 200 | 0,7 | 80% |
| S_{10} | 200 | 0,8 | 80% |
| S_{11} | 200 | 0,9 | 0% |

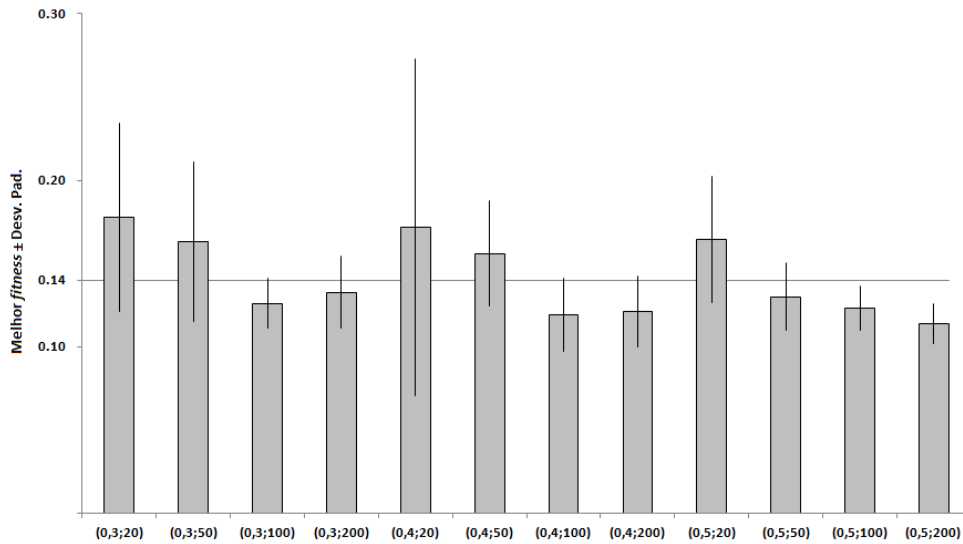
Tabela 6.7: Novo conjunto de avaliações considerando PSO com modelo completo.

| Conjunto | Partículas | Inércia | Resultados com $fitness < 0,14$ |
|----------|------------|---------|---------------------------------|
| F_0 | 20 | 0,3 | 40% |
| F_1 | 20 | 0,4 | 40% |
| F_2 | 20 | 0,5 | 40% |
| F_3 | 50 | 0,3 | 40% |
| F_4 | 50 | 0,4 | 30% |
| F_5 | 50 | 0,5 | 50% |
| F_6 | 100 | 0,3 | 80% |
| F_7 | 100 | 0,4 | 80% |
| F_8 | 100 | 0,5 | 80% |
| F_9 | 200 | 0,3 | 70% |
| F_{10} | 200 | 0,4 | 80% |
| F_{11} | 200 | 0,5 | 100% |

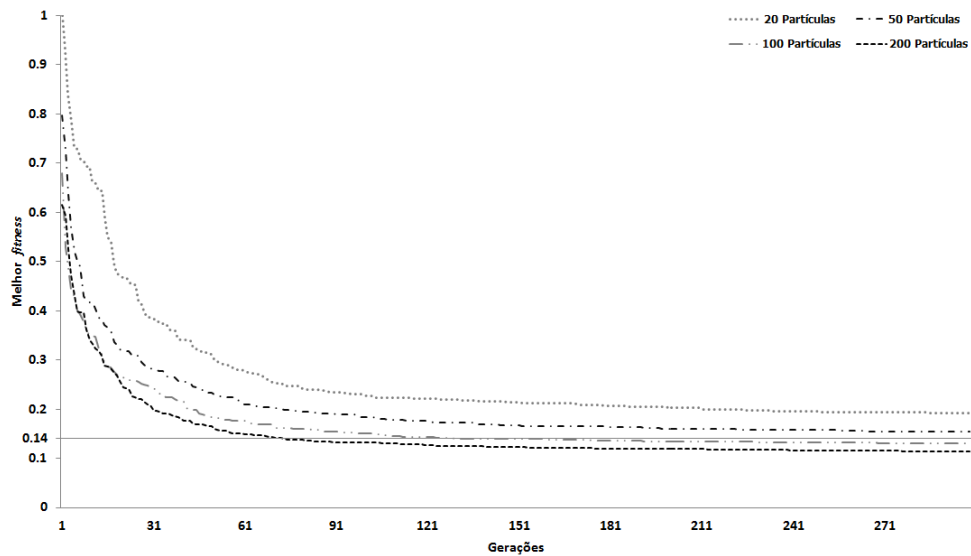
padrão. É possível observar que o conjunto de parâmetros que apresentou a melhor média com menor desvio padrão é a experiência F_{11} , de acordo com o demonstrado na Tabela 6.7.

O gráfico de evolução do $fitness$, para 4 robôs de combate, pode ser visto na Figura 6.10(b), que mostra a média de 10 simulações (modelo completo, 800 gerações, 200 partículas e inércia de 0,5). O gráfico apresentado na Figura 6.10(a) mostra que o melhor $fitness$ é obtido com uma quantidade de partículas igual a 200. A partir de 150 gerações, a otimização do $fitness$ praticamente estabiliza.

As avaliações com PSO não utilizaram métricas possíveis de serem comparadas diretamente com os Algoritmos Genéticos. Ambos, AG e PSO resolveram o problema, porém, avaliações estatísticas formais devem ser realizadas buscando avaliar qual dos métodos é o melhor para resolver o problema proposto. Considerando o uso da mesma função de aptidão em ambos os métodos, bem como dos mesmos parâmetros no simulador (velocidade e orientação do vento, velocidade dos robôs), podemos dizer que ambos foram bastante similares.



(a)



(b)

Figura 6.10: (a) Resultados das avaliações apresentadas na Tabela 6.7. O eixo x apresenta o conjunto de parâmetros do experimento como (inércia; tamanho do enxame). (b) Curvas de queda de $fitness$ de acordo com o número de gerações, para diferentes tamanhos de enxame. Inércia de 0,5.

6.3.4 Outras técnicas de otimização

Duas técnicas de otimização bastante difundidas em áreas de engenharia, e que apresentam resultados satisfatórios em diversos problemas são o *Simulated Annealing* (SA) e o *Hill Climbing* (HC). Ambos exigem menos recursos computacionais quando comparados às técnicas de AG ou PSO. Propomos então uma avaliação das técnicas de HC e SA buscando comparar seus resultados com os resultados obtidos com os trabalhos que exploraram o uso de AG e PSO. A comparação entre os resultados tem como objetivo analisar se é possível obter resultados satisfatórios de formação de grupos com técnicas de otimização computacionalmente menos demandantes. Esta comparação tem como base o resultados das simulações.

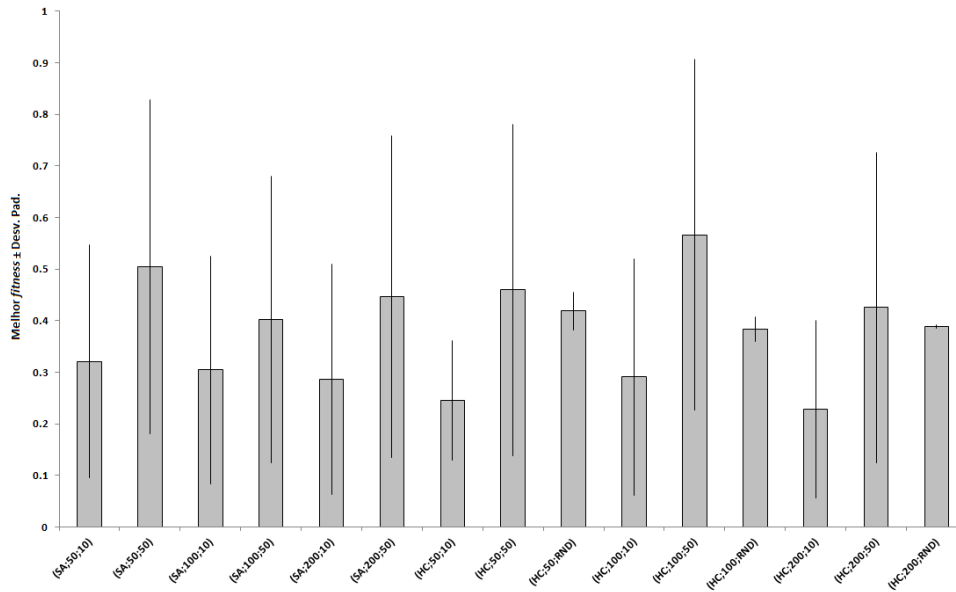
Parâmetros de grande impacto, tanto no SA como no HC, são a *quantidade de vizinhos* e a *forma de obtenção de vizinhos*. A quantidade de vizinhos é o número de avaliações realizadas em cada passo da otimização (também chamado de número de soluções candidatas – usualmente representado pela letra k). É, de certa forma, similar ao conceito de população em um AG. A forma de obtenção de vizinhos é análoga ao conceito de mutação; permite definir o quão local ou global será a busca. Em geral, a forma de obtenção dos vizinhos pode ter como base um valor percentual, usado para mais e para menos sobre o valor original, ou então, pode ter como base a geração de novos números aleatórios.

As simulações foram realizadas considerando diferentes quantidades de vizinhos e diferentes formas de obtenção de vizinhos, de acordo com o apresentado na Tabela 6.8. O número de gerações utilizado é de 800 – mesmo valor utilizado no trabalho (Pessin *et al.*, 2010c). Para verificar qual o melhor conjunto de parâmetros, tanto as mesmas características climáticas como as posições iniciais dos robôs são as mesmas dos outros trabalhos. Foram executadas vinte simulações com cada conjunto de parâmetros, totalizando 900 simulações.

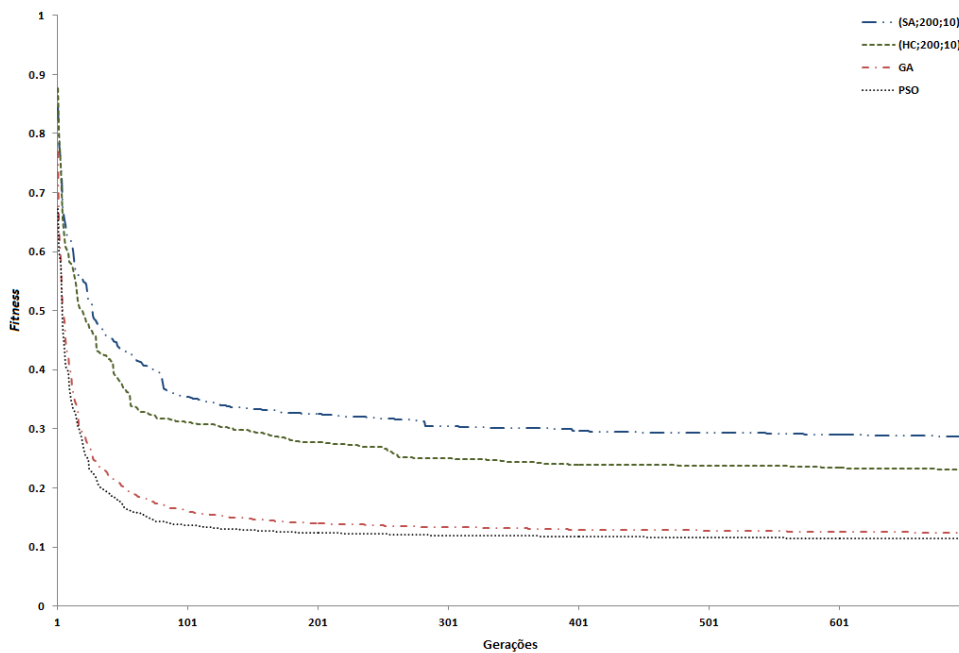
Tabela 6.8: Conjunto de parâmetros avaliados no HC e SA.

| Algoritmo | Quantidade de vizinhos avaliados em cada geração | Mutação (Forma de obtenção de novos vizinhos) |
|-----------|--|---|
| HC | {50; 100; 200} | {10%; 50%; RND} |
| HC | {50; 100; 200} | {10%; 50%; RND} |
| HC | {50; 100; 200} | {10%; 50%; RND} |
| SA | {50; 100; 200} | {10%; 50%} |
| SA | {50; 100; 200} | {10%; 50%} |
| SA | {50; 100; 200} | {10%; 50%} |

O resultado do processo de busca (otimização) pode ser visto na Figura 6.11(a). Resultados abaixo de 0,14 ocorreram apenas em 15% dos casos nos seguintes conjuntos de parâmetros: (HC, 50 vizinhos, 10% de mutação); (HC, 100 vizinhos, 10% de mutação); (HC, 200 vizinhos,



(a)



(b)

Figura 6.11: (a) Resultados das avaliações apresentadas na Tabela 6.8. O eixo x apresenta a descrição do conjunto de parâmetros como (método; número de vizinhos; mutação). (b) Curvas de queda de *fitness* dos melhores HC e SA obtidos nesta avaliação – HC e SA com 200 vizinhos e 10% de mutação. Curva de queda de *fitness* do AG (Pessin e Osório, 2009b) – AG com 150 indivíduos, 10% de mutação, cruzamento de dois pontos. Curva de queda de *fitness* do PSO (Pessin *et al.*, 2010c) – PSO com 200 partículas, inércia de 0,5 e modelo completo.

10% de mutação). E em 5% dos casos nos seguintes conjuntos de parâmetros: (SA, 100 vizinhos, 10% de mutação); (SA, 200 vizinhos, 10% de mutação). Nenhum outro conjunto de parâmetros apresentou alguma solução com *fitness* inferior a 0,14. Os conjuntos citados como capazes de encontrar algumas soluções satisfatórias, apenas as encontraram em um número muito pequeno de vezes. Sendo que o desvio padrão das simulações é altíssimo. Tal desvio padrão nos faz acreditar que tanto o HC como o SA se mostram muito suscetíveis a semente aleatória utilizada.

Podemos ver na Figura 6.11(a) que as simulações que utilizaram 10% de mutação apresentaram as menores médias finais e que as simulações que utilizaram vizinhos aleatórios apresentaram os menores desvios padrão. Este fato poderia ser explorado em futuras análises, por exemplo, mantendo n gerações randômicas seguidas de m gerações com mutação de 10%. Ainda, na Figura 6.11(a), podemos ver que as piores médias foram obtidas com mutação de 50%, independente da quantidade de vizinhos. Também podemos ver que usando 10% de mutação existe uma pequena vantagem para o HC frente ao SA, para todas as quantidades de vizinhos. Novas avaliações poderiam ser feitas a fim de encontrar melhores conjuntos de parâmetros no HC e no SA. Porém, como o número de gerações avaliado neste trabalho está similar ao utilizado no AG e no PSO, isso nos faz acreditar que o AG e o PSO oferecem vantagens. Algumas simulações utilizando 1000 gerações foram realizadas com o HC e o SA, porém, não mostraram resultados significativamente melhores.

No SA, foram utilizados parâmetros de temperatura e resfriamento de 20,0 e 0,98, respectivamente. O critério de parada foi o número de gerações (igual a 800). Considerando que o SA pode trocar soluções atuais por soluções piores, observamos que isso não ocorreu em todas as simulações; quando ocorreu ela foi sempre no máximo até a geração 35 (temperatura superior a 10,0), concentrando-se nas primeiras 10 gerações (temperatura superior a 17,0).

A Figura 6.11(b) apresenta as curvas de queda de fitness dos dois melhores conjuntos obtidos no trabalho, bem como as curvas obtidas nos trabalhos (Pessin e Osório, 2010) e (Pessin *et al.*, 2010c). Em (Pessin e Osório, 2010) foi possível observar que o AG permitiu obter 100% dos casos com valores inferiores a 0,14. Em (Pessin *et al.*, 2010c) foi possível observar que o PSO permitiu obter 100% dos casos com valores também inferiores a 0,14. Os parâmetros avaliados não permitem uma validação formal, porém, tanto o AG como o PSO apresentaram resultados muito superiores aos obtidos utilizando HC e SA.

6.4 Abordagens Baseadas em Etologia e Engenharia

Descrevemos nesta seção outras duas abordagens para trabalhos com grupos de robôs – baseadas em etologia e em engenharia – fazendo uso da nomenclatura definida por Nolfi e Floreano (2000). Essas abordagens foram descritas na Seção 2.1.1. Em resumo, a abordagem em etologia tem como princípio fundamental a tentativa de modelar sistemas robóticos com base em comportamento social de animais simples; a perspectiva na engenharia tem como princípio fundamental a divisão do sistema em partes tratáveis com conjuntos de regras.

Dois trabalhos com grupos de robôs, considerando as abordagens etológica e de engenharia, foram realizados em conjunto com as investigações da tese. Estes trabalhos não terão seu desenvolvimento detalhados aqui, porém serão objetos de discussão em comparação com a abordagem bio-inspirada utilizada nas seções anteriores. Em resumo, a abordagem bio-inspirada é relacionada com a teoria da evolução natural, e tem como princípio fundamental a tentativa de modelar sistemas que se aperfeiçoem com base na evolução.

O trabalho (Vargas *et al.*, 2012) apresenta um grupo de robôs que utiliza a abordagem etológica com objetivo de estudar movimentação de grupos. Durante o desenvolvimento, diversas abstrações tiveram de ser implementadas a fim de tornar o sistema capaz de resolver a tarefa de movimentação e troca de contexto (recolher lixo orgânico ou reciclável, movimentar para a base de coleta correta, realizar recarga de bateria). Por exemplo, os robôs tem sua posição conhecida e compartilham essa informação por meio de comunicação, o que não é comum em sistemas biológicos. Embora algumas implementações utilizem comunicação visual para isso, o reconhecimento das imagens é feito por processos não etológicos. Na sua essência, os sistemas etológicos apresentam uma troca de contexto, em geral, baseada em autômatos. A tarefa macro até pode ser aprendida por meio de modelos de comportamento de animais simples, entretanto, algumas tarefas continuam sob responsabilidade do desenvolvedor. Pudemos ver que os sistemas etológicos até servem como inspiração para certos casos, entretanto, os modelos híbridos são os que apresentam melhores resultados, quando o objetivo é resolver uma tarefa em específico. Para finalizar, era esperado um alto grau de flexibilidade com o desenvolvimento dos sistemas com base em etologia, entretanto, pudemos ver que o mesmo não se mostrou tão verdadeiro quanto o esperado. Os trabalhos (Dickson, 2012; Matear, 2012; Smith, 2012) desenvolvidos no *Robotics Lab* se mostraram capazes de realizar tarefas, entretanto, não apresentaram a capacidade de flexibilidade esperada.

Um trabalho com abordagem baseada em engenharia (da forma canônica) foi desenvolvido para (Pessin *et al.*, 2007a), sendo o primeiro trabalho na linha de grupos de robôs do autor deste documento. O trabalho se mostrou capaz de combater incêndio, entretanto, era bastante sensível a mudanças de ambiente – era necessário um conjunto específico de regras para cada mudança

ambiental, além disso, o ajuste fino nos comportamentos era difícil de ser desenvolvido pelo programador.

O desenvolvimento da abordagem bio-inspirada mostrou capacidade de lidar bem com o ajuste fino na operação específica de combate ao incêndio. A característica de escalabilidade não foi investigada, embora, com base em literatura possamos dizer que o modelo proposto é capaz de resolver o problema para grupos relativamente pequenos de robôs. É possível que, para grupos grandes, como a estrutura usada para evolução ficaria muito grande, o método de evolução pode ter dificuldade em achar soluções, na forma atualmente proposta.

A abordagem apresentada também tem certa necessidade de troca de contexto. É importante perceber que, inspirada em biologia (evolução), é realizada a escolha das posições de atuação. Entretanto, a navegação dos robôs foi realizada por meio de conjuntos de regras (ou por redes neurais treinadas, como apresentado no Capítulo 5). Ainda, a fase de identificação do incêndio e comunicação depende de características desenvolvidas por um especialista (programador). Ou seja, o que parece ser mais interessante como abordagem é a divisão das tarefas em partes tratáveis, entretanto, o desenvolvimento de cada parte pode sim fazer uso de métodos biológicos ou etológicos, dependendo das atividades e dos tipos de robôs disponíveis.

6.5 Considerações Finais

Este capítulo teve como objetivo apresentar investigações com uso de técnicas de otimização para organização e coordenação de grupos de robôs para atuação em atividades críticas. Foi descrita a proposta, o desenvolvimento, a utilização e da avaliação de modelos que permitem graus de autonomia e eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis. Apresentamos trabalhos de formação de grupo que estendem e aperfeiçoam avaliações iniciais com grupos de robôs apresentadas detalhadamente na dissertação de mestrado de Pessin (2008).

Foi apresentado um grupo de robôs que tem sua atuação em uma atividade crítica, simulada como um incêndio florestal. Uma equipe de robôs autônomos deve trabalhar cooperativamente a fim de realizar com sucesso a identificação e o combate ao incêndio. O *planejamento da formação* da estrutura do grupo de robôs se dá por meio de técnicas de otimização em uma unidade monitora. A *divisão de tarefas* se dá de forma implícita, pela função de aptidão desenvolvida. A *coordenação* se dá devido a comunicação entre as unidades de combate e a unidade monitora e o possível replanejamento das operações.

É importante perceber que, no modelo proposto, as contribuições obtidas não podem ser generalizadas para qualquer tamanho de grupo de robôs ou de modelos de coordenação. Na

proposta descrita, foi demonstrada a possibilidade de uso de grupos pequenos de robôs (foram utilizadas equipes com 2, 3 e 4 robôs). É possível que números um pouco maiores possam ser usados, entretanto, para números maiores (por exemplo, maiores que 20 ou 30 robôs), melhorias significativas devem ser feitas a fim de tornar o sistema útil para a criticidade da operação devido principalmente a restrições de tempo.

Ainda, o modelo de divisão de tarefas está implementado de forma implícita na função de aptidão desenvolvida e tem como princípio de funcionamento a subdivisão das tarefas entre os robôs. É possível que para grupos heterogêneos de robôs o desenvolvimento de um *fitness* que leve em conta a divisão de tarefas se torne bastante difícil de ser implementado, sendo necessário lançar mão de outras técnicas, possivelmente com o desenvolvimento de hierarquias entre subgrupos de robôs. No modelo atual, por usar uma estrutura onde todos os robôs estão definidos, o sistema apresenta uma inerente capacidade de divisão de tarefas, quando considerando grupos homogêneos.

Foi possível perceber, no desenvolvimento das investigações, que as quatro técnicas investigadas podem ser utilizados para ajuste das posições de atuação dos robôs no combate ao incêndio. Mostramos como a capacidade de comunicação pode ser usada a fim de identificar falhas e então replanejar a atuação dos robôs, propondo uma heurística que diminuiu muito o tempo na obtenção das novas posições, tanto no caso de reorganização de quatro para três robôs como no caso de envio de uma nova equipe de outros dois robôs.

Pudemos ver que o conjunto de parâmetros usados no AG influencia bastante a busca pelas soluções. Ainda, pudemos ver que, nesta aplicação, o uso de populações sobrepostas apresentou resultados bastante superiores aos demais. O uso de populações sobrepostas com percentual de 25%, 50% e 75% foi parecido entre si, entretanto, muito superior ao AG sem o uso de população sobreposta. O uso de escala de *fitness* apresentou resultados bastante similares quando usando 150 indivíduos, entretanto, com a diminuição no número de indivíduos (de 150 para 100) pudemos ver que com o uso do *fitness* do tipo *sharing* obtivemos resultados superiores aos demais. Embora tenhamos apresentado uma vasta investigação por parâmetros mais eficientes para este AG, é importante salientar que dadas as características de cada problema, tais parâmetros não podem ser diretamente generalizados como melhores para outras aplicações. Assim, também se vislumbra como trabalho futuro o desenvolvimento de um algoritmo evolutivo que procure pelos melhores parâmetros nos métodos de otimização.

Com o uso de Otimização por Enxame de Partículas, dado um conjunto de configurações específico, obtivemos resultados aparentemente similares aos usando Algoritmos Genéticos. Tais resultados se mostraram consideravelmente melhores quando comparados os uso dos métodos de *Hill Climbing* e *Simulated Annealing*. A descrição e a proposta de um simulador probabilístico de incêndio florestal foi realizada. Este simulador apresentou características básicas

acreditadas como suficientes para validar a operação. Entretanto, é válida a investigação com a troca do modelo por outros modelos de incêndio.

Uma breve discussão entre abordagens bio-inspiradas, etológicas e de engenharia foi realizada. Após o desenvolvimento deste trabalho, fica bastante claro que o uso de diferentes abordagens tem, em geral, diferentes expectativas (principalmente em termos de eficiência, flexibilidade, tolerância a falha e escalabilidade). Quando se tem em mente a abordagem etológica, o objetivo ainda é o estudo de uma forma mais geral do comportamento do sistema; parece que, para a maioria dos autores, não é esperada alta eficiência do mesmo. Espera-se sim alta escalabilidade, que é inerente ao conceito, e alta flexibilidade, ainda não tão demonstrada como esperado. Quando se tem em mente a abordagem bio-inspirada, o objetivo é usar métodos evolutivos (conceito aqui generalizado para métodos de otimização) que em geral demonstram sucesso na evolução de partes (módulos) do sistema; como demonstrado nesta tese, evoluímos a estrutura responsável pelo posicionamento de ação dos robôs. A abordagem em engenharia se mostrou importante pelo fato de apresentar o conceito de desmembrar um sistema em partes tratáveis. Para finalizar, dados os diversos trabalhos desenvolvidos neste capítulo bem como nos capítulos anteriores, pode-se concluir que, para o desenvolvimento de grandes sistemas, as três abordagens podem conviver em harmonia. Dada a divisão do sistema em partes tratáveis, cada parte (módulo) pode ser desenvolvida e avaliada usando uma ou outra abordagem de acordo com as características de eficiência, flexibilidade, tolerância a falha e escalabilidade desejadas.

6.6 Trabalhos Futuros

O principal trabalho futuro vislumbrado para este capítulo é a investigação do mesmo usando robôs reais. O estudo do *reality gap* pode fortalecer os conceitos inerentes as abordagens utilizadas. O desenvolvimento do sistema, fora da simulação, ainda depende do desenvolvimento de outras partes como (i) a identificação e monitoração do incêndio, (ii) a implementação das capacidades de comunicação entre todos os agentes, (iii) a implementação da capacidade de atuação no combate ao incêndio. Por outro lado, a realização de experimentos em escala reduzida (com grupos de robôs reais de pequeno porte) em ambientes fechados (possivelmente de Realidade Aumentada, no que se refere a simulação do incêndio), torna-se uma possibilidade a partir da proposta do sistema de localização *indoor* apresentado no Capítulo 4.

Outro trabalho futuro é a investigação sobre a abordagem etológica em substituição ao modelo biológico utilizado. É importante realizar uma comparação entre as capacidades de escalabilidade, flexibilidade tolerância a falhas nos modelos.

Fora do eixo de grupos de robôs, uma avaliação mais detalhada sobre o simulador de incêndio poderia mostrar que o mesmo é útil para outras investigações. O simulador de incêndio poderia assim ser utilizado em pesquisas sobre monitoração e previsão de comportamentos de incêndios.

Também é cogitada a possibilidade de aplicar a abordagem proposta nesta tese em outras aplicações, por exemplo, na contenção de vazamentos de óleo no mar, ou mesmo em atividades militares visando cercar uma tropa inimiga em deslocamento. Tarefas como estas possuem propriedades semelhantes ao combate à incêndios, pois é necessário simular o deslocamento/espalhamento de um elemento no ambiente ao mesmo tempo em que se otimiza a estratégia para se aproximar e cercar este elemento. Adaptações na função de aptidão, no comportamento dos robôs e no simulador podem ser necessários, todavia, espera-se que seja possível um aproveitamento de conceitos em relação ao trabalho realizado com combate a incêndios.

Em termos da abordagem inspirada na biologia utilizada, seria interessante um estudo detalhado a fim de tornar o critério de parada mais inteligente. O mesmo foi utilizado como número de gerações, o que não garante a obtenção de resultados positivos em todos os casos. Um critério de parada mais inteligente poderia ser avaliado a fim de tornar o método de evolução mais flexível às mudanças do ambiente.

Considerações Finais

Este capítulo apresenta as considerações finais da tese. Este é um capítulo com um forte viés de revisão, já que a maioria das considerações finais, bem como os trabalhos futuros, já foram apresentados nos capítulos de investigações (Capítulos 4, 5, e 6). Entretanto, apresentamos neste capítulo de uma forma mais condensada os objetivos, as motivações e as contribuições resultantes desta tese. Serão descritos também considerações e trabalhos futuros de uma forma mais ampla, integrando os conceitos apresentado em cada um dos capítulos de investigação.

7.1 Revisão do Tema

O contínuo aumento da complexidade no controle de sistemas robóticos, bem como na aplicação de sistemas de grupos de robôs substituindo pessoas em atividades práticas (como as atividades extenuantes, que coloquem a vida em risco, ou que sejam, de certa forma, ineficientes) tem gerado demanda por soluções mais robustas, flexíveis, e eficientes. Propomos nesta tese uma série de trabalhos que tangem a busca por sistemas de robótica móvel autônomos e inteligentes, para aplicação em ambientes dinâmicos. Como ponto inicial, a investigação sobre a atuação com grupos de robôs em atividades críticas foi o principal elemento motivador para o desenvolvimento de técnicas bio-inspiradas em simulação. Buscando o desenvolvimento de um sistema mais completo e que pudesse ter aplicação real, tornaram-se necessários o desenvolvi-

mento e a investigação em duas áreas base da robótica, necessárias para que o grupo de robôs tivesse atuação eficiente: localização e navegação. Sendo assim, o trabalho se desenvolve em três grandes áreas: localização, navegação e coordenação de grupos.

No que tange a pesquisa sobre coordenação de grupos de robôs, fizemos a escolha de trabalhar sobre uma aplicação modelada como o problema de combate a incêndios florestais. Incêndios são problemas graves e recorrentes em nossa sociedade, com soluções ainda bastante ineficientes até mesmo do ponto de vista da resolução por humanos. Os incêndios florestais causam diversos tipos de danos humanos, materiais e ambientais. O desenvolvimento de um sistema de grupo de robôs para atuação em monitoração e combate a incêndios em áreas florestais é um problema multidisciplinar. Assim, o sucesso desta tarefa envolve o uso de uma variedade de tecnologias de diferentes campos, como robótica móvel (*e.g.* controle de movimentação, fusão de sensores, modelagem robótica). Envolve a coleta de informações de especialistas sobre incêndios (*e.g.* comportamento do fogo em ambientes naturais, técnicas e métodos de monitoração, estratégias reais de combate).

Na investigação sobre formação de grupos (Capítulo 6), o objetivo geral é a busca de soluções de alto grau de autonomia para atuação de grupos de robôs em missões críticas, considerando ambientes dinâmicos. Focando em aspectos de autonomia e eficiência na resolução das tarefas e buscando avaliar e propor aspectos que suportem altos graus de escalabilidade, flexibilidade e tolerância a falhas. Sendo objetivos específicos desta parte da tese: (i) desenvolvimento e avaliação de uma arquitetura multirrobótica para combate de incêndios florestais por meio de métodos de otimização, considerando autonomia e eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis, (ii) proposta de um simulador de incêndio florestal, (iii) desenvolvimento e avaliação de um método de tolerância a falhas na operação multirrobótica, (iv) avaliações buscando otimizar o método de otimização por meio de avaliação de diversas características do algoritmo proposto, (v) avaliações com outros métodos de otimização, como Otimização por Enxame de Partículas, *Simulated Annealing* e *Hill Climbing*, e (vi) a proposta e discussão de possíveis alternativas às abordagens utilizadas.

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

Na investigação sobre navegação (Capítulo 5), o objetivo geral é ser um elo entre os conceitos de localização e coordenação de grupos, sendo o foco o desenvolvimento de um veículo autônomo com maior implicação em avanços técnicos.. Os objetivos específicos desta parte da Tese: (i) aperfeiçoamento de uma arquitetura para navegação autônoma, baseada em visão monocular e com capacidade de seguir pontos esparsos de GPS, (ii) apresentação do estudo do *gap* entre simulação e realidade entre o modelo proposto por Pessin *et al.* (2010a) e os trabalhos com robôs reais do LRM/USP (2013), (iii) avaliação de diferentes topologias de redes neurais buscando verificar as topologias que permitam o aprendizado com maior acuracidade, (iv) discussão sobre as capacidades de aprendizado, e (v) apresentação de algumas contribuições técnicas no desenvolvimento de controladores (*drivers*) para os sensores *GPS Garmin 18x 5Hz* e *True North Revolution GS*, utilizados no robô CaRINA I.

Localização é uma informação fundamental para a realização de diversas tarefas nas áreas de robótica móvel. O uso de GPS apresenta certo sucesso em ambientes externos. Versões de GPS com erros de poucos centímetros são extremamente caras e só apresentam funcionalidade em ambiente externo. O uso de GPS em ambientes internos ainda não é uma realidade. Diversas formas de localização em ambientes internos existem, como por meio de câmeras ou *lasers*, entretanto, cada uma apresenta pontos fracos específicos. O uso da potência da rede sem fio pode prover uma metáfora do modelo de GPS para ambientes internos onde podemos obter localização com um custo computacional mais baixo que usando laser, mais acurado do que usando odometria e mais prático do que usando câmeras.

Na investigação sobre localização (Capítulo 4), o objetivo geral é a busca por um método de localização alternativo ao GPS (passível de ser usado em ambientes internos) que seja fácil, prático e barato de implementar. Os objetivos específicos desta parte da tese são: (i) desenvolvimento e avaliação de uma arquitetura para localização de robôs móveis usando sinal de potência de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina, (ii) avaliação de diferentes topologias de redes neurais buscando verificar as topologias que permitam o aprendizado com maior acuracidade, (iii) avaliação de métodos de evolução de redes neurais, considerando evolução de pesos e de topologias, (iv) proposta de diferentes tipos de filtros, a serem aplicados sobre o sinal lido dos pontos de acesso, a fim de avaliar o comportamento geral do sistema bem como do aprendizado da rede neural, (v) estudo e avaliação da rede neural considerando possíveis diferentes quantidades de entradas de sinal da rede sem fio, a fim de avaliar como o aprendizado da rede se comporta com mais ou menos pontos de acesso disponíveis, (vi) estudo e proposta de métodos para evolução de pesos e topologias de redes neurais, seguido da comparação e discussão dos resultados entre os métodos de evolução com métodos clássicos, e (vii) proposta de uma possível alternativa com o objetivo de tornar o sistema proposto mais autônomo e tolerante a falhas.

7.2 Resultados e Contribuições

7.2.1 Localização

A tarefa de localização se inseriu no contexto da investigação do uso de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e aprendizado de máquina para auxiliar na tarefa de localização de robôs móveis em ambientes internos. Descrevemos propostas e avaliações de métodos de localização de agentes móveis usando RSSI obtido de APs. Embora extensamente avaliado, o trabalho sobre localização usando redes sem fio ainda apresenta temas abertos de investigação (Seção 4.5). Entretanto, os atuais resultados utilizando o RSSI se apresentaram encorajadores e motivam a continuidade da pesquisa.

Na sequência, será apresentada uma discussão sobre os resultados mais relevantes obtidos, bem como as contribuições sobre a pesquisa com localização de robôs em ambientes internos. A investigação realizada mostrou uma metodologia para localização de robôs móveis usando o valor da potência de redes sem fio (*Wireless LAN IEEE 802.11 b/g*) e Aprendizado de Máquina (por meio de redes neurais). Tal metodologia tem como objetivo prover informação de localização de forma prática, barata e acurada tanto para robôs móveis como para outros tipos de dispositivos (por exemplo, uma rede de sensores sem fio). A metodologia inicial, embora funcional, mostrou algumas limitações na questão de completa autonomia e correção contínua de erro. Entretanto, tais limitações podem vir a ser corrigidas com as propostas descritas nas Seções 4.3.6 e 4.3.7.

Pelo fato da potência do sinal de rede ter grandes oscilações, foi proposto e avaliado o uso de filtro de mediana móvel. Pudemos ver que não existe diferença estatística entre o uso do valor bruto (*raw*) ou da mediana (média) de duas leituras. Entretanto, para as demais janelas de tempo, o aumento na acuracidade do aprendizado da rede neural foi bastante promissor. Mostramos inclusive que existe correlação forte entre o tamanho da janela e a acuracidade no aprendizado. Pudemos ver que, a partir de uma janela de oito leituras, o erro mínimo, aparentemente, não diminui muito, entretanto as medidas dos erros continuam caindo. Usamos o filtro de mediana ao contrário do filtro de médias pela característica das medianas em remover inerentemente os *outliers*. É importante perceber que o tamanho da janela significa, sob um ponto de vista mais crítico, um atraso na obtenção da posição. Assim, um usuário deste tipo de sistema deveria enfrentar o dilema de escolher entre acuracidade e velocidade na obtenção da informação.

Alguns pontos de acesso apresentam valores com mais oscilações que outros. Foi um dos objetivos avaliar o aprendizado relacionando a quantidade de redes *wireless* disponíveis (quantidade de entradas no sistema) e a acuracidade no aprendizado da posição pela rede neural.

Mostramos que, mesmo com pontos de acesso com amplitude mais alta, a remoção destes não auxiliou no aprendizado. Ou seja, mesmo mantendo os pontos com nível mais alto de oscilação, para o conjunto avaliado, não obtivemos melhorias. Isso possivelmente se deve ao fato de que a amplitude nos pontos de acesso utilizados não é muito elevada e também pela capacidade inerente das redes neurais em suavizar as entradas.

Avaliações de topologias de redes neurais por meio de processo manual se mostraram eficientes na obtenção de boas redes, entretanto, se apresentaram como um trabalho bastante árduo e extenuante. Uma falta de maiores garantias na acuracidade do aprendizado também decorre deste processo manual, caso seja considerado um conjunto muito pequeno de avaliações. Tal problema motivou o desenvolvimento de um método de evolução de topologias a fim de tornar o processo de descoberta de boas redes um processo automatizado. Mostramos como um método de otimização (algoritmo genético) pôde ser utilizado para buscar (evoluir) topologias de redes neurais com resultados semelhantes com as redes obtidas por processo manual. Pudemos ver que o uso de apenas uma execução de cada rede apresentou resultados sensivelmente piores que utilizando média de 5 ou 10 execuções para obtenção do *fitness*. Execuções considerando médias de 5 ou 10 execuções apresentaram resultados mais homogêneos e com maior acuracidade. Ainda, pudemos ver que não apresenta diferença estatística para execução do treino da rede por 20.000 ciclos ou 100.000 ciclos. Esta descoberta é bastante importante visto que o treino de uma rede neural por um número menor de ciclos permite uso mais rápido do sistema e, por ser uma tarefa que usa bastante poder computacional, recursos como bateria, memória e CPU são poupados. É importante perceber a importância do AG na busca por topologias (adotando uma abordagem híbrida com o SNNS para treinar os pesos das conexões com o algoritmo *rprop*). Pudemos encontrar topologias similares com as obtidas por processo manual sem o gasto de tempo do pesquisador. O processo evolucionário é demorado (dependendo da base de dados e de outros parâmetros, pode consumir um tempo elevado de processamento), entretanto, o pesquisador não fica preso ao processo. Otimizações no algoritmo ou execução em paralelo poderiam ser realizadas a fim de tornar o processo mais rápido.

Também avaliamos um algoritmo evolucionário no que se refere à evolução de pesos de conexões. A evolução de pesos é importante e necessária em sistemas onde as bases de dados não podem ser pré-obtidas (que não é exatamente nosso caso). Entretanto, a dúvida entre evolução e treino existia. Pudemos ver que o método de otimização, eficiente para a busca de topologias, não se mostrou interessante para evolução dos pesos neste trabalho. Os resultados usando *rprop* no treino da rede foram melhores que usando o PSO tanto em termos de acuracidade como em tempo de execução. Era esperada, segundo alguns autores, que a evolução dos pesos usando PSO (ou outro método qualquer de otimização/evolução) proporcionasse resultados finais com

uma menor dispersão, dada a capacidade de fazer os pesos das conexões escaparem de mínimos locais. Entretanto, tal característica não foi observada para o tipo de problema tratado.

Para finalizar, é bom lembrar que a abordagem desenvolvida para localização em ambientes internos teve como principal motivador a possibilidade de uso de robôs de pequeno porte em ambientes indoor. O uso de robôs, em atividades práticas úteis, usualmente depende de capacidades de localização e navegação. Sendo que a navegação (a ser investigada no Capítulo 5), em geral, também depende da capacidade de localização. Ainda, o uso de robôs de pequeno porte em ambiente indoor é um dos requisitos do desenvolvimento real, em pequena escala, da investigação descrita no Capítulo 6. Além disso, os próximos pesquisadores que realizarem investigações com os robôs SRV-1Q ou Pioneer AT, disponíveis no LRM, ou com o robô NAO, disponível no Robotics Lab, podem se beneficiar desta metodologia de localização desenvolvida.

7.2.2 Navegação Autônoma

A investigação sobre navegação apresentou o desenvolvimento de um veículo autônomo de grande porte, funcional para ambientes externos, abordando sucintamente aspectos de simulação que foram base para o desenvolvimento de sistemas de navegação em robôs reais de pequeno porte seguido do desenvolvimento e investigação com um robô de grande porte.

Na sequência, será apresentada uma discussão sobre os resultados mais relevantes obtidos, bem como as contribuições sobre a pesquisa com navegação autônoma.

A investigação realizada apresentou uma arquitetura para navegação com capacidade de localização (pose atual e destino) e percepção local (área navegável e não navegável), baseado na integração de uma câmera monocular e GPS. Tal arquitetura possui capacidade reativa de desvio de obstáculos (áreas não-navegáveis) e tem certo grau de deliberação, visto que é “atraída” por pontos de GPS. Como vimos anteriormente, é possível que nossa arquitetura seja estendida para que seja utilizada em conjunto com outros métodos que utilizem mapas mais complexos, desde que estes forneçam uma lista de coordenadas a serem seguidas. Navegação autônoma é um dos fundamentos necessários para o desenvolvimento das investigações apresentadas no capítulo sobre atividades com grupos de robôs (Capítulo 6).

A arquitetura se mostrou capaz de percorrer os trajetos propostos, evitando obstáculos (áreas não navegáveis). Entretanto, algumas restrições foram encontradas no modelo. O modo como foram usadas as geometrias e a identificação de pista apresentou falta de capacidade de fazer os veículos se manter no centro da pista. Em certas situações, como em uma pista dupla ou muito larga, o veículo usa “toda” a área navegável livremente, inclusive mudando de posição e de

faixa, sem que isto seja realmente desejado. Melhorias neste aspecto estão sendo consideradas como trabalhos futuros.

O desenvolvimento dos trabalhos (Pessin, 2008; Pessin *et al.*, 2010a) mostrou uma rede neural com capacidade para realizar a navegação com desvio de obstáculos com um robô simulado. Este capítulo estendeu a pesquisa em navegação e mostrou as adaptações necessárias com relação ao uso de um robô real e para o uso de uma câmera monocular. Em comparação ao modelo simulado, que utilizava 5 sensores *laser* simulados, o novo modelo com o robô de grande porte utilizou 5 entradas de distância na rede neural (5 geometrias), enquanto em (Pessin, 2008; Pessin *et al.*, 2010a) foi utilizada também uma rede com 5 neurônios para as entradas, relacionadas não com geometrias livres ou ocupadas, mas com a distância percebida pelos sensores, ou seja, mantivemos o mesmo número de entradas relacionadas com a percepção de distância/obstáculos. Com relação à escolha do movimento (navegação ou desvio de obstáculos) o mesmo modo se manteve – o robô é “atraído” por um ponto de GPS e tem capacidade reativa para desviar de obstáculos no trajeto. O recebimento da coordenadas é similar também, dado por um arquivo texto contendo coordenadas de latitude e longitude. Uma diferença entre os modelos é a entrada, na rede neural, relacionada com o destino (em termos de orientação). A mudança na estrutura da camada de entrada da rede neural, em relação a rede utilizada em (Pessin *et al.*, 2010a) possibilitou o aprendizado satisfatório com um número menor de épocas.

Diversas topologias foram avaliadas e se mostraram estatisticamente similares. Embora não tenha sido possível até o momento fazer com que o robô aprenda com um motorista humano – o que é uma investigação corrente no LRM – a rede neural aprende com os dados provenientes de um sistema baseado em regras. É importante o desenvolvimento de métodos ou filtros que permitam o aprendizado da rede neural com dados provenientes da condução do veículo por pessoas; o atual modelo, com base em regras, apresenta a dificuldade de que, para cada diferente conjunto de sensores, conjuntos específicos de regras deveriam ser construídos. Esta é uma tarefa muito árdua e sujeita a erros. O uso de um método de aprendizado de máquina que aprenda com motorista humano deve diminuir esta dificuldade e tornar o sistema livre de possíveis erros em regras.

Outras dificuldades encontradas foram o erro alto do GPS utilizado e diversos problemas com mudanças na posição da câmera e mudanças na iluminação. Tais problemas estão sendo investigados no LRM/USP (2013) e devem ser resolvidos com a integração de mais sensores ao sistema. Apesar disso, a investigação descrita neste capítulo apresenta uma importante etapa do desenvolvimento de um veículo autônomo de grande porte.

7.2.3 Grupos de Robôs

A investigação sobre formação de grupos apresentou o desenvolvimento de métodos de otimização para organização e coordenação de grupos de robôs para atuação em atividades críticas. Foi descrita a proposta, o desenvolvimento, a utilização e da avaliação de modelos que permitem graus de autonomia e eficiência na formação, coordenação e divisão de tarefas entre os membros de um grupo de robôs móveis. Apresentamos trabalhos de formação de grupo que estendem e aperfeiçoam avaliações iniciais com grupos de robôs apresentadas detalhadamente na dissertação de mestrado de Pessin (2008).

Na sequência, será apresentada uma discussão sobre os resultados mais relevantes obtidos, bem como as contribuições sobre a pesquisa com grupos de robôs.

Foi apresentado um grupo de robôs que tem sua atuação em uma atividade crítica, simulada como um incêndio florestal. Uma equipe de robôs autônomos deve trabalhar cooperativamente a fim de realizar com sucesso a identificação e o combate ao incêndio. O *planejamento da formação* da estrutura do grupo de robôs se dá por meio de técnicas de otimização em uma unidade monitora. A *divisão de tarefas* se dá de forma implícita, pela função de aptidão desenvolvida. A *coordenação* se dá devido a comunicação entre as unidades de combate e a unidade monitora e o possível replanejamento das operações.

É importante perceber que, no modelo proposto, as contribuições obtidas não podem ser generalizadas para qualquer tamanho de grupo de robôs ou de modelos de coordenação. Na proposta descrita, foi demonstrada a possibilidade de uso de grupos pequenos de robôs (foram utilizadas equipes com 2, 3 e 4 robôs). É possível que números um pouco maiores possam ser usados, entretanto, para números maiores (por exemplo, maiores que 20 ou 30 robôs), importantes melhorias devem ser feitas a fim de tornar o sistema útil para a criticidade da operação devido principalmente a restrições de tempo.

Ainda, o modelo de divisão de tarefas está implementado de forma implícita na função de aptidão desenvolvida e tem como princípio de funcionamento a subdivisão das tarefas entre os robôs. É possível que para grupos heterogêneos de robôs o desenvolvimento de um *fitness* que leve em conta a divisão de tarefas se torne bastante difícil de ser implementado, sendo necessário lançar mão de outras técnicas, possivelmente com o desenvolvimento de hierarquias entre subgrupos de robôs. No modelo atual, por usar uma estrutura onde todos os robôs estão definidos, o sistema apresenta uma inerente capacidade de divisão de tarefas, quando considerando grupos homogêneos.

Foi possível perceber, no desenvolvimento das investigações, que as quatro técnicas investigadas podem ser utilizados para ajuste das posições de atuação dos robôs no combate ao incêndio. Mostramos como a capacidade de comunicação pode ser usada a fim de identificar

falhas e então replanejar a atuação dos robôs, propondo uma heurística que diminuiu muito o tempo na obtenção das novas posições, tanto no caso de reorganização de quatro para três robôs como no caso de envio de uma nova equipe de outros dois robôs.

Pudemos ver que o conjunto de parâmetros usados no AG influencia bastante a busca pelas soluções. Como, por exemplo, com o uso de populações sobrepostas e com o uso de escala de *fitness*. Ambos permitiram obter resultados superiores para a classe de problema tratado. Embora tenhamos apresentado uma vasta investigação por parâmetros mais eficientes para este AG, é importante salientar que dadas as características de cada problema, tais parâmetros não podem ser diretamente generalizados como melhores para outras aplicações. Assim, também se vislumbra como trabalho futuro o desenvolvimento de um algoritmo evolutivo que procure pelos melhores parâmetros nos métodos de otimização.

Com o uso de Otimização por Enxame de Partículas, dado um conjunto de configurações específico, obtivemos resultados aparentemente similares aos usando Algoritmos Genéticos. Tais resultados se mostraram consideravelmente melhores quando comparados os uso dos métodos de *Hill Climbing* e *Simulated Annealing*. A descrição e a proposta de um simulador probabilístico de incêndio florestal foi realizada. Este simulador apresentou características básicas acreditadas como suficientes para validar a operação. Entretanto, é válida a investigação com a troca do modelo por outros modelos de incêndio.

Uma breve discussão entre abordagens bio-inspiradas, etológicas e de engenharia foi realizada. Após o desenvolvimento deste trabalho, fica bastante claro que o uso de diferentes abordagens tem, em geral, diferentes expectativas esperadas (principalmente em termos de eficiência, flexibilidade, tolerância a falha e escalabilidade). Quando se tem em mente a abordagem etológica, o objetivo ainda é o estudo de uma forma mais geral do comportamento do sistema; parece que, para a maioria dos autores, não é esperada alta eficiência do mesmo. Espera-se sim alta escalabilidade, que é inerente ao conceito, e alta flexibilidade, ainda não tão demonstrada como esperado. Quando se tem em mente a abordagem bio-inspirada, o objetivo é usar métodos evolutivos (conceito aqui generalizado para métodos de otimização) que em geral demonstram sucesso na evolução de partes (módulos) do sistema; como demonstrado nesta tese, evoluímos a estrutura responsável pelo posicionamento de ação dos robôs. A abordagem em engenharia se mostrou importante pelo fato de apresentar o conceito de desmembrar um sistema em partes tratáveis. Para finalizar, dados os diversos trabalhos desenvolvidos no Capítulo 6 bem como nos Capítulos 4 e 5, pode-se concluir que, para o desenvolvimento de grandes sistemas, as três abordagens podem viver em harmonia. Dada a divisão do sistema em partes tratáveis, cada parte (módulo) pode ser desenvolvida e avaliada usando uma ou outra abordagem de acordo com as características de eficiência, flexibilidade, tolerância a falha e escalabilidade desejadas.

No desenvolvimento das investigações realizadas nesta tese, diversas publicações foram obtidas. A lista completa pode ser vista no Anexo A.

7.3 Trabalhos Futuros

Os trabalhos futuros vislumbrados em cada capítulo de investigação estão descritos no fim dos mesmos, Seções 4.5, 5.5 e 6.6. Entretanto, de uma forma mais geral, alguns outros trabalhos futuros são almejados:

- De acordo com a evolução das pesquisas dentro do LRM, as formas de avaliação dos resultados dos experimentos foram sendo melhoradas. Assim, pôde-se ver na tese o aperfeiçoamento do uso de métodos estatísticos. Esses métodos ainda não são consenso geral, nem são padrão para todas as pesquisas. Assim, sugere-se como trabalho futuro um estudo mais aprofundado e criterioso para a utilização de métodos estatísticos para a análise das investigações.
- Ainda, a rastreabilidade dos dados foi uma necessidade cada vez mais frequente por motivos como: reutilização dos mesmos para verificações e também para execução de novas análises ou novas plotagens. Assim, um trabalho futuro é o desenvolvimento de padrões para auxiliar na rastreabilidade dos resultados dos dados dos experimentos (não apenas dos códigos-fonte, o que é algo bastante comum).
- Outro trabalho futuro é o relacionamento com grupos de engenharia de software a fim de criar modelos padrão para teste de software, para padronização do código e para as diversas outras etapas onde a engenharia de software pode auxiliar na construção de software robótico.
- Em uma das investigações do Capítulo 4 foi descrito um algoritmo genético para evoluir topologias de redes neurais, avaliado na topologia da rede neural de localização. É esperado como trabalho futuro o desenvolvimento deste AG como um software de fácil utilização para evoluir outras redes neurais de outros trabalhos, como os trabalhos de navegação apresentados no Capítulo 5 bem como em demais trabalhos que usam redes neurais, como (Sales *et al.*, 2010), (Shinzato e Wolf, 2011) e (Hata *et al.*, 2009a).
- O desenvolvimento de um AG de fácil utilização para otimização de parâmetros dos métodos de otimização (como o próprio AG, PSO, HC e SA) também é vislumbrado como trabalho futuro, sendo uma contribuição técnica importante para o LRM.

- Para finalizar, como não existe um consenso geral sobre qual a melhor técnica de aprendizado de máquina a ser utilizada, outro trabalho futuro é a investigação e comparação com outras técnicas de aprendizado de máquina, como máquinas de suporte vetorial e processos gaussianos.

Esta tese visou a apresentação de um conjunto significativo de trabalhos referentes a estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs, onde foram descritas contribuições e resultados práticos voltados principalmente para a localização, navegação e coordenação de grupos robóticos.

Referências Bibliográficas

- AILINCA, M.; PARANTHAMAN, P. K.; PESSIN, G.; OSÓRIO, F. S.; SOUZA, J. R.; WOLF, D. F.; UYAMA, J.; VARGAS, P. A. Uncovering new neural network topologies in real world robot applications. In: *12th Annual Workshop on Computational Intelligence (UKCI) – Studies in Mathematical and Computer Science*, Edinburgh, UK: Heriot Watt University, 2012, p. 53–58.
- ALDEBARAN Aldebaran robotics. Acesso em 18 de janeiro, 2013.
Disponível em <http://www.aldebaran-robotics.com/en>
- ANTUNES, M. A. H. Uso de satélites para detecção de queimadas e para avaliação do risco de fogo. *Ação Ambiental*, v. 12, p. 24–27, 2000.
- ARAI, T.; PAGELLO, E.; PARKER, L. E. Guest editorial: advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, v. 18, n. 5, p. 655–661, 2002.
- ARDUINO Arduino. Acesso em 15 de janeiro, 2013.
Disponível em <http://www.arduino.cc>
- BALCH, T.; ARKIN, R. C. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, v. 14, n. 6, p. 926–939, 1998.
- BALUJA, S. Evolution of an artificial neural network based autonomous land vehicle controller. *IEEE Transactions on Systems, Man and Cybernetics*, v. 26, n. 3, p. 450–463, 1996.
- BARES, J.; WHITTAKER, W. L. Dante ii, robotics institute, carnegie mellon university. Acesso em 14 de janeiro, 2013.
Disponível em http://www.ri.cmu.edu/research_project_detail.html?project_id=163&menu_id=261

- BATISTA, A. C. Detecção de incêndios florestais por satélite. *Floresta*, v. 34, p. 237–241, 2004.
- BAYINDIR, L.; SAHIN, E. A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering & Computer Sciences*, v. 15, n. 2, p. 115–147, 2007.
- BECCENERI, J. C. Meta-heurística e otimização combinatória: aplicações em problemas ambientais. In: *Computação e Matemática Aplicadas as Ciências e Tecnologias Espaciais*, INPE, p. 65–81, 2009.
- BEKEY, G. A. *Autonomous robots: From biological inspiration to implementation and control*. MIT Press, 593 p., 2005.
- BOEING Boeing. Acesso em 14 de janeiro, 2013.
Disponível em http://www.boeing.com/history/boeing/x45_jucas.html
- BOMBARDIER Bombardier inc. website. Acesso em maio de, 2007.
Disponível em <http://www.canadair.com>
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. *Swarm intelligence: From natural to artificial systems*. Oxford University Press, 320 p., 1999.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. Inspiration for optimization from social insect behaviour. *Nature*, v. 406, n. 6791, p. 39–42, 2000.
- BONABEAU, E.; MEYER, C. Swarm intelligence: A whole new way to think about business. *Harvard Business Review*, v. 79, n. 5, p. 106–114, 2001.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory, COLT '92*, New York, NY, USA: ACM, 1992, p. 144–152 (COLT '92,).
- BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDERMIR, T. B. *Redes neurais artificiais: Teoria e aplicações*. 2 ed. Rio de Janeiro, RJ, Brasil : LTC, 283 p., 2007.
- BROOKS, R. A. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, v. 2, p. 14–23, 1986.
- BROOKS, R. A. Artificial life and real robots. In: *Proceedings of the First European Conference on Artificial Life*, 1992, p. 3–10.
- BUEHLER, M.; IAGNEMMA, K.; SINGH, S. *The 2005 darpa grand challenge: The great robot race*. Springer, 520 p., 2007.
- BUEHLER, M.; IAGNEMMA, K.; SINGH, S. *The darpa urban challenge: Autonomous vehicles in city traffic*. Springer, 626 p., 2009.
- BURGESS, N.; DONNETT, J. G.; JEFFERY, K. J.; O'KEEFE, J. Robotic and neuronal simulation of hippocampal navigation. *Philosophical Transactions of the Royal Society*, v. 352, p. 1535–1543, 1997.

- CAO, Y.; FUKUNAGA, A.; KAHNG, A. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, v. 4, n. 1, p. 7–27, 1997.
- CARVALHO, A.; BRAGA, A. P.; LUDERMIR, T. B. Computação evolutiva. In: *Sistemas Inteligentes - Fundamentos e Aplicações*, cap. 9, Solange O. Rezende (Org.) Manole, p. 225–248, 2003.
- CASTRO, F. X.; TUDELA, A.; SEBASTIA, M. T. Modeling moisture content in shrubs to predict fire risk in catalonia. *Agricultural and Forest Meteorology*, v. 116, p. 49–59, 2003.
- CATERPILLAR Caterpillar. Acesso em 14 de janeiro, 2013.
Disponível em <http://brasil.cat.com>
- CE Comissão européia: O que faz a europa? incêndios florestais. 2006.
Disponível em <http://ec.europa.eu/research/leaflets/disasters/pt/forest.html>
- CHONG, K. S.; KLEEMAN, L. Accurate odometry and error modelling for a mobile robot. In: *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, 1997, p. 2783 –2788 vol.4.
- CHUVIECO, E.; CONGALTON, R. G. Application of remote sensing and geographic information systems to forest fire hazard mapping. *Remote Sensing of the Environment*, v. 29, p. 147–159, 1989.
- CLIFF, D.; MILLER, G. F. Co-evolution of pursuit and evasion ii: Simulation methods and results. In: *From animals to animats IV: Proceedings of the fourth international conference on simulation of adaptive behavior*, 1996.
- COSTA, F. G.; BRAUN, T.; UHEYAMA, J.; PESSIN, G.; OSÓRIO, F. S. Arquitetura baseada em veículos aéreos não tripulados e redes de sensores sem fio para aplicações agrícolas. In: *VIII Congresso Brasileiro de Agroinformática (SBIAgro2011)*, 2011, p. 1–5.
- COSTA, F. G.; UHEYAMA, J.; BRAUN, T.; PESSIN, G.; OSÓRIO, F. S.; VARGAS, P. A. The use of unmanned aerial vehicles and wireless sensor network in agricultural applications. In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2012, p. 1–4.
- CPTEC/INPE Centro de previsão do tempo e estudos climáticos/instituto nacional de pesquisas espaciais. 2007.
Disponível em <http://www.cptec.inpe.br/queimadas>
- CRANDALL, J. W.; NIELSEN, C. W.; GOODRICH, M. A. Towards predicting robot team performance. In: *IEEE International Conference on Systems, Man and Cybernetics*, 2003, p. 906–911.
- DARGIE, W.; POELLABAUER, C. *Fundamentals of wireless sensor networks: Theory and practice*. Wiley, 330 p., 2010.

- DASSAULT Dassault aviation. Acesso em 14 de janeiro, 2013.
Disponível em <http://www.dassault-aviation.com/en/defense/neuron/introduction.html?L=1>
- DEERE Deere & company. Acesso em 14 de janeiro, 2013.
Disponível em http://www.deere.com/wps/dcom/en_US/products/equipment/gator_utility_vehicles/military_utility_vehicles/r_gator/r_gator_series.page, lastaccessjajajaj
- DEZFOULIAN, S. H.; WU, D.; AHMAD, I. S. A generalized neural network approach to mobile robot navigation and obstacle avoidance. In: LEE, S.; CHO, H.; YOON, K.-J.; LEE, J., eds. *Intelligent Autonomous Systems 12*, v. 193 de *Advances in Intelligent Systems and Computing*, Springer Berlin Heidelberg, p. 25–42, 2013.
- DIAS, M. B.; ZLOT, R.; KALRA, N.; STENTZ, A. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, v. 94, n. 7, p. 1257–1270, 2006.
- DICKSON, B. *Robotics in a search and rescue task*. Final year dissertation, Heriot-Watt University, 2012.
- DORIGO, M.; SAHIN, E. Guest editorial. *Autonomous Robots*, v. 17, p. 111–113, 2004.
- DORIGO, M.; TRIANNI, V.; SAHIN, E.; GROSS, R.; LABELLA, T. H.; BALDASSARRE, G.; NOLFI, S.; DENEUBOURG, J.; MONDADA, F.; FLOREANO, D.; GAMBARDELLA, L. M. Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, v. 17, p. 223–245, 2004.
- DPI/INPE Divisão de processamento de imagens/instituto nacional de pesquisas espaciais. 2007.
Disponível em <http://www.dpi.inpe.br/proarco/bdqueimadas>
- DUDEK, G.; JENKIN, M. *Computational principles of mobile robotics*. Cambridge University Press, 280 p., 2000.
- DUDEK, G.; JENKIN, M. R. M.; MILIOS, E.; WILKES, D. A taxonomy for multi-agent robotics. *Autonomous Robots*, v. 3, n. 4, p. 375–397, 1996.
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, Nagoya, Japan: IEEE Press, 1995, p. 39–43.
- EBERHART, R. C.; SHI, Y. Comparison between genetic algorithms and particle swarm optimization. *Evolutionary Programming VII (Lecture Notes in Computer Science)*, v. 1447, p. 611–616, 1998.
- EBERHART, R. C.; SHI, Y.; KENNEDY, J. *Swarm intelligence*. Morgan Kaufmann, 510 p., 2001.

- ELNAHRAWY, E.; LI, X.; MARTIN, R. The limits of localization using signal strength: a comparative study. In: *IEEE SECON*, 2004, p. 406–414.
- ENGEDY, I.; HORVATH, G. Artificial neural network based mobile robot navigation. In: *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, 2009, p. 241–246.
- ENGELBRECHT, A. P. *Fundamentals of computational swarm intelligence*. John Wiley & Sons, 672 p., 2006.
- ESPINACE, P.; SOTO, A.; TORRES-TORRITI, M. Real-time robot localization in indoor environments using structural information. *IEEE LARS*, 2008.
- FAHLMAN, S. E. *An empirical study of learning speed in back-propagation networks – computer science technical report*. Relatório Técnico, Carnegie Mellon University, 1988.
- FARINELLI, A.; IOCCHI, L.; NARDI, D. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 34, n. 5, p. 2015–2028, 2004.
- FARINELLI, A.; IOCCHI, L.; NARDI, D.; ZIPARO, V. A. Assignment of dynamically perceived tasks by token passing in multirobot systems. *Proceedings of the IEEE*, v. 94, n. 7, p. 1271–1288, 2006.
- FERNANDES, L. C.; SOUZA, J. R.; SHINZATO, P. Y.; PESSIN, G.; MENDES, C. C. T.; OSÓRIO, F. S.; WOLF, D. F. Intelligent robotic car for autonomous navigation: Platform and system architecture. In: *II Brazilian Conference on Critical Embedded Systems (CBSEC)*, Los Alamitos, CA, USA: IEEE Computer Society, 2012, p. 12–17.
- FLOREANO, D.; KELLER, L. Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS Biology*, v. 8, n. 1, p. 1–8, 2010.
- FLOREANO, D.; MONDADA, F. Automatic creation of an autonomous agent: genetic evolution of a neural network driven robot. In: *Proceedings of third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, 1994.
- FLOREANO, D.; MONDADA, F. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, v. 26, p. 396–407, 1996.
- FLOREANO, D.; NOLFI, S. God save the red queen! competition in co-evolutionary robotics. In: *Proceedings of the Second Annual Conference on Genetic Programming*, 1997.
- FLOREANO, D.; NOLFI, S.; MONDADA, F. Competitive co-evolutionary robotics: From theory to practice. In: *From Animals to Animats 5*, 1998.
- FOWLER, C. Demeter terrain engine home page. Acesso em 9 de julho, 2010.
Disponível em <http://demeter.sourceforge.net>

- FOX, B. D.; KO, J.; KONOLIGE, K.; LIMKETKAI, B.; SCHULZ, D.; STEWART, B. Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, v. 94, p. 1325–1339, 2006.
- FRANCESCHINI, F.; GALETTO, M.; MAISANO, D.; MASTROGIACOMO, L. Mobile spatial coordinate measuring system (mscms) – introduction to the system. *The International Journal of Advanced Manufacturing Technology*, v. 47, p. 3867–3889, 2009.
- FRITSCH, D.; WEGENER, K.; SCHRAFT, R. D. Control of a robotic swarm for the elimination of marine oil pollutions. In: *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, 2007, p. 29–36.
- GARCIA, A. C. B.; SICHMAN, J. S. Agentes e sistemas multiagentes. In: *Sistemas Inteligentes - Fundamentos e Aplicações*, cap. 11, Solange O. Rezende (Org.) Manole, p. 269–306, 2003.
- GAST, M. S. *802.11 wireless networks: The definitive guide*. O'Reilly, 464 p., 2002.
- GO, J.; BROWNING, B.; VELOSO, M. Accurate and flexible simulation for dynamic, vision-centric robots. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, IEEE Computer Society, 2004, p. 1388–1389.
- GOLDBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 432 p., 1989.
- GOLDBERG, D. E. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, v. 5, p. 139–167, 1991.
- GOUMA, V.; CHRONOPOULOU, A. Wildland fire danger zoning: A methodology. *International Journal of Wildland Fire*, v. 8, n. 1, p. 37–43, 1998.
- GUARDA Comissão organizadora do concurso robô bombeiro, escola superior de tecnologia e gestão da guarda. Acesso em 25 de julho, 2010.
Disponível em <http://robobombeiro.ipg.pt>
- GUSTAFSSON, F.; GUNNARSSON, F. Positioning using time-difference of arrival measurements. In: *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, 2003.
- HALL, D. National geographic. Acesso em 9 de julho, 2010.
Disponível em <http://ngm.nationalgeographic.com>
- HARA, S.; ANZAI, D. Experimental performance comparison of rssi- and tdoa-based location estimation methods. In: *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, 2008, p. 2651–2655.
- HARTER, D. Evolving neurodynamic controllers for autonomous robots. In: *Proceedings of International Joint Conference on Neural Networks*, 2005.

- HARVEY, I.; PAOLO, E. D.; WOOD, R.; QUINN, M.; TUCI, E. Evolutionary robotics: a new scientific tool for studying cognition. *Artificial Life*, v. 11, n. 1-2, p. 79–98, 2005.
- HATA, A. Y.; WOLF, D. F.; PESSIN, G.; OSÓRIO, F. S. Terrain mapping and classification in outdoor environments using neural networks. *International Journal of u- and e- Service, Science and Technology*, v. 2, p. 51–61, 2009a.
- HATA, A. Y.; WOLF, D. F.; PESSIN, G.; OSÓRIO, F. S. Terrain mapping and classification using neural networks. In: *International Conference on Hybrid Information Technology (ICHIT)*, New York: ACM, 2009b, p. 438–442.
- HATA, A. Y.; WOLF, D. F.; SHINZATO, P. Y. Mapeamento e classificação de terrenos utilizando aprendizado supervisionado. In: *Congresso Brasileiro de Automática*, 2010, p. 122–129.
- HEINEN, M. R.; OSORIO, F. S.; HEINEN, F. J.; KELBER, C. Seva3d: Using artificial neural networks to autonomous vehicle parking control. In: *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2006.
- HEINEN, M. R.; OSÓRIO, F. S. Evolving morphologies and gaits of physically realistic simulated robots. In: *Proceedings of the 2009 ACM symposium on Applied Computing table of contents*, 2009, p. 1161–1165.
- HOFFMANN, L. T.; CASTRO, A. P. A.; SILVA, J. D. S. Controle inteligente de um robô móvel utilizando imagens. In: *Jornada de Robótica Inteligente (JRI), Anais do XXIV Congresso da Sociedade Brasileira de Computação (CSBC'04)*, 2004.
- HOLLAND, J. H. *Adaptation in natural and artificial systems*. The University of Michigan Press, 1975.
- HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, p. 359–366, 1989.
- HUANG, G.-B. Learning capability and storage capacity of two-hidden-layer feedforward networks. *Neural Networks, IEEE Transactions on*, v. 14, n. 2, p. 274 – 281, 2003.
- HUHNS, M. N.; STEPHENS, L. M. Multiagent systems and societies of agents. In: WEISS, G., ed. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, cap. 2, MIT Press, p. 79–120, 2000.
- HUSBANDS, P.; HARVEY, I. Evolution versus design: Controlling autonomous robots. In: *Proceedings of the Third Annual Conference of AI, Simulation and Planning in High Autonomy Systems – Integrating Perception, Planning and Action*, 1992, p. 139–146.
- IDMIND Projecto raposa, robot semi-autónomo para operações de salvamento, idmind - engenharia de sistemas. Acesso em 27 de julho, 2010.
Disponível em <http://raposa.idmind.pt>

- IOCCHI, L.; NARDI, D.; PIAGGIO, M.; SGORBISSA, A. Distributed coordination in heterogeneous multi-robot systems. *Autonomous Robots*, v. 15, n. 2, p. 155–168, 2003.
- IROBOT irobot corporation. Acesso em 15 de janeiro, 2013.
Disponível em <http://store.irobot.com/product/index.jsp?productId=11305110&cp=2501652>
- JACOBSSON, M.; NIEMEGERES, I.; DE GROOT, S. H. *Personal networks: Wireless networking for personal devices*. Wiley, 228 p., 2010.
- JAKOBI, N. Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In: *Proceedings of Fourth European Conference on Artificial Life*, 1993, p. 348–357.
- JAKOBI, N.; HUSB, P.; HARVEY, I. Noise and the reality gap: The use of simulation in evolutionary robotics. In: *Advances in Artificial Life: Proceedings of 3rd European Conference on Artificial Life*, 1995, p. 704–720.
- JIANGUO, W.; YILONG, Z.; LINLIN, X. Adaptive genetic algorithm enhancements for path planning of mobile robots. In: *2010 International Conference on Measuring Technology and Mechatronics Automation*, 2010.
- JOCHEM, T.; POMERLEAU, D.; THORPE, C. Next generation neurally based autonomous road follower. In: *Proceedings of the International Conference on Intelligent Autonomous Systems*, IOS Press, 1993.
- JOCHEM, T.; POMERLEAU, D.; THORPE, C. Vision guided lane-transition. In: *Proceedings of the Intelligent Vehicles Symposium*, 1995, p. 30–35.
- JONG, K. A. D. *Evolutionary computation*. Bradford, 272 p., 2002.
- JPL/NASA Jet propulsion laboratory (jpl), nasa. Acesso em 15 de janeiro, 2013.
Disponível em <http://marsrovers.jpl.nasa.gov/home/index.html>
- JUN, H.; QINGBAO, Z. Multi-objective mobile robot path planning based on improved genetic algorithm. In: *2010 International Conference on Intelligent Computation Technology and Automation*, 2010.
- KARTALOPOULOS, S. V. *Understanding neural networks and fuzzy logic: Basic concepts and applications*. Wiley, 1995.
- KENNEDY, J. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, IEEE Computer Society, 1999, p. 1931–1938.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, 1995, p. 1942–1948.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.

- KOPROSKI, L. P. *O fogo e seus efeitos sobre a herpeto e a mastofauna terrestre no parque nacional de ilha grande*. Dissertação de Mestrado, Universidade Federal do Paraná (UFPR), 2005.
- KOULAMAS, C.; ANTONYA, S. R.; JAENA, R. A survey of simulated annealing applications to operations research problems. *Omega*, v. 22, p. 41–56, 1994.
- KRAMER, J.; SCHEUTZ, M. Development environments for autonomous mobile robots: A survey. *Autonomous Robots*, v. 22, n. 2, p. 101–132, 2007.
- LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. Introdução aos algoritmos genéticos. In: *XVIII Jornada de Atualização em Informática (JAI), XIX Congresso Nacional da Sociedade Brasileira de Computação*, 1999, p. 51–126.
- LADD, A.; BEKRIS, K.; RUDYS, A.; WALLACH, D.; KAVRAKI, L. On the feasibility of using wireless ethernet for indoor localization. *Robotics and Automation, IEEE Trans. on*, v. 20, n. 3, p. 555 – 559, 2004.
- LANG, H.; WANG, Y.; DE SILVA, C. Mobile robot localization and object pose estimation using optical encoder, vision and laser sensors. In: *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, 2008, p. 617 –622.
- LEONARD, J. J.; DURRANT-WHYTE, H. F. Simultaneous map building and localization for an autonomous mobile robot. In: *Proceedings of IEEE International Workshop on Intelligent Robots and Systems (IROS)*, 1991, p. 1442–1447.
- LIF/UFPR Laboratório de incêndios florestais: Pesquisas e projetos em prevenção e combate de incêndios florestais, universidade federal do paraná (ufpr). Acesso em 14 de janeiro, 2013.
Disponível em <http://floresta.ufpr.br/firelab>
- LIU, H.; DARABI, H.; BANERJEE, P.; LIU, J. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 37, n. 6, p. 1067 –1080, 2007.
- LOPES, A.; CRUZ, M.; VIEGAS, D. Firestation – an integrated software system for the numerical simulation of firespread on complex topography. *Environmental Modelling & Software*, v. 17, p. 269–285, 2002.
- LRM/USP Laboratório de robótica móvel (usp). Acesso em 15 de janeiro, 2013.
Disponível em <http://www.lrm.icmc.usp.br>
- MA *Guia metodológico para elaboração de plano municipal/intermunicipal de defesa da floresta contra incêndios, apêndice 1: Metodologia para a elaboração da carta dos combustíveis florestais*. Ministério da Agricultura, 2006.
- MACEDO, A. R. M.; MACEDO, A. R. L.; DUARTE, J. B. F. Robótica aplicada ao combate a incidentes. *Revista TN Petróleo*, v. 53, p. 108–113, 2007.

- MAISANO, D. A.; JAMSHIDI, J.; FRANCESCHINI, F.; MAROPOULOS, P.; MASTROGIACOMO, L.; MILEHAM, A.; OWEN, G. Indoor gps: System functionality and initial performance evaluation. *International Journal of Manufacturing Research*, v. 3, n. 3, p. 335–349, 2008.
- MARINELLO, S. H. M.; UHEYAMA, J.; PESSIN, G.; OSÓRIO, F. S.; VARGAS, P. A. Determinando posições relativas de nós móveis por meio de bluetooth e aprendizado de máquina. *Revista de Exatas e Tecnológicas (RETEC) ISSN 2236-739X*, v. 3, p. 1–10, 2012.
- MARTINELLI, A. The odometry error of a mobile robot with a synchronous drive system. *Robotics and Automation, IEEE Trans. on*, v. 18, n. 3, p. 399–405, 2002.
- MATEAR, C. *Applying swarm intelligence and evolutionary robotics to robot soccer*. Final year dissertation, Heriot-Watt University, 2012.
- MATTHIES, L. Jet propulsion laboratory. Acesso em 14 de janeiro, 2013.
Disponível em <http://www-robotics.jpl.nasa.gov/tasks/showTask.cfm?TaskID=107&tdaID=999992>
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943.
- MENDES, C. C. T.; SHINZATO, P. Y.; WOLF, D. F.; OSORIO, F. S. Anais do xix congresso brasileiro de automática. In: *Deteção de Obstáculos em Tempo Real para Sensores de Profundidade Utilizando uma GPU*, 2012, p. 1000–1007.
- MENDES, C. C. T.; WOLF, D. F. Desvio de obstáculos utilizando um método estéreo semi-global. In: *VIII Encontro Nacional de Inteligência Artificial*, 2011, p. 788–799.
- METCALFE, K. Galeria de fotos. Acesso em 8 de julho, 2010.
Disponível em <http://www.flickr.com/photos/kasimetcalfe>
- MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. Springer, 387 p., 1996.
- MICHALEWICZ, Z.; FOGEL, D. B. *How to solve it: Modern heuristics*. Springer, 467 p., 2004.
- MIGLINO, O.; LUND, H. H.; NOLFI, S. Evolving mobile robots in simulated and real environments. *Artificial Life*, v. 2, p. 417–434, 1996.
- MILLER, L.; RODRIGUEZ, D.; ALLEN, K.; MAKEEV, M.; PLEW, J.; MOORE, M.; SCHWARTZ, E. M. Firebot: Design of an autonomous fire fighting robot. In: *Proceedings of Florida Conference on Recent Advances in Robotics (FCRAR'03)*, 2003.
- MITCHELL, M. *An introduction to genetic algorithms*. The MIT Press, 221 p., 1998.
- MITCHELL, T. M. *Machine learning*. McGraw-Hill, 432 p., 1997.

- MITRI, S.; FLOREANO, D.; KELLER, L. The evolution of information suppression in communicating robots with conflicting interests. *PNAS*, v. 106, n. 37, p. 15786–15790, 2009.
- MOBILEROBOTS Adept mobilerobots llc. Acesso em 15 de janeiro, 2013.
Disponível em <http://www.mobilerobots.com>
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes - Fundamentos e Aplicações*, cap. 4, Solange O. Rezende (Org.) Manole, p. 89–114, 2003.
- MOSHAIQV, A.; ASHRAM, A. Multi-objective evolution of robot neuro-controllers. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC'09)*, 2009.
- NAPIER, A.; SIBLEY, G.; NEWMAN, P. Real-time bounded-error pose estimation for road vehicles using vision. In: *IEEE Conf. on Intelligent Transp. Systems*, 2010.
- NIEHAUS, C.; RÖFER, T.; LAUE, T. Gait optimization on a humanoid robot using particle swarm optimization. In: *Proceedings of the Second Workshop on Humanoid Soccer Robots in conjunction with the 2007 IEEE-RAS International Conference on Humanoid Robots*, 2007.
- NIKOLOPOULOS, C. *Expert systems: Introduction to first and second generation and hybrid knowledge based systems*. CRC Press, 336 p., 1997.
- NILSSON, N. J. A mobile automaton: An application of artificial intelligence techniques. *Autonomous Mobile Robots: Control, Planning and Architecture*, v. 2, p. 233–239, 1969.
- NL/CMU Navigation laboratory, carnegie mellon university. Acesso em 14 de janeiro, 2013.
Disponível em <http://www.cs.cmu.edu/afs/cs/project/alv/www/index.html>
- NOC National oceanography centre. Acesso em 14 de janeiro, 2013.
Disponível em http://www.noc.soton.ac.uk/nmf/sea_sys_index.php?page=isis
- NOLFI, S.; FLOREANO, D. Coevolving predator and prey robots: Do "arms races" arise in artificial evolution? *Artificial Life*, v. 4, p. 311–335, 1998.
- NOLFI, S.; FLOREANO, D. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. The MIT Press, 384 p., 2000.
- NREC/CMU National robotics engineering center (nrec), carnegie mellon. Acesso em 15 de janeiro, 2013.
Disponível em http://www.rec.ri.cmu.edu/projects/auto_harvesting
- ORLANDO, M. F.; AKOLKAR, H.; DUTTA, A.; SAXENA, A.; BEHERA, L. Optimal design and control of a hand exoskeleton. In: *2010 IEEE Conference on Robotics Automation and Mechatronics (RAM)*, 2010.

- OSAGIE, P. *Distributed control for networked autonomous vehicles*. Dissertação de Mestrado, Royal Institute of Technology (KTH), Stockholm, Sweden, 2006.
- OSG Open scene graph. Acesso em 30 de julho, 2010.
Disponível em <http://www.openscenegraph.org/projects/osg>
- OSÓRIO, F. S.; WOLF, D. F.; BRANCO, K. C.; PESSIN, G. Mobile robots design and implementation: From virtual simulation to real robots. In: *Proceedings of IDMMÉ – Virtual Concept 2010*, 2010, p. 1–6.
- PACK, D. J.; AVANZATO, R.; AHLGREN, D. J.; VERNER, I. M. Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives. *IEEE Transactions on Education*, v. 4, p. 369–376, 2004.
- PARKER, L. E. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, v. 2, n. 1, p. 5–14, 2008.
- PENNSSTATE Penn state abington, mini grand challenge 2010. Acesso em 26 de julho, 2010.
Disponível em <http://www.cede.psu.edu/users/avanzato/robots/contests/outdoor/index.htm>
- PERERA, A. H.; REMMEL, T. K. Fire mapping in a northern boreal forest: assessing avhrr/ndvi methods of change detection. *Forest Ecology and Management*, v. 152, p. 119–129, 2001.
- PERKINS, C.; LEI, L.; KUHLMAN, M.; LEE, T.; GATEAU, G.; BERGBREITER, S.; ABSHIRE, P. Distance sensing for mini-robots: Rssi vs. tdoa. In: *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, 2011, p. 1984–1987.
- PESSIN, G. *Evolução de estratégias e controle inteligente em sistemas multi-robóticos robustos*. Dissertação de Mestrado, Universidade do Vale do Rio dos Sinos (UNISINOS), 2008.
Disponível em <http://goo.gl/udDsr>
- PESSIN, G.; OSÓRIO, F.; HATA, A.; WOLF, D. Intelligent control and evolutionary strategies applied to multirobotic systems. In: *Industrial Technology (ICIT), 2010 IEEE International Conference on*, 2010a, p. 1427–1432.
- PESSIN, G.; OSÓRIO, F. S. Algoritmos genéticos aplicados à formação e atuação de grupos robóticos. In: *Encontro Nacional de Inteligência Artificial (VII ENIA), Anais do XXIX Congresso da Sociedade Brasileira de Computação*, Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2009a, p. 1019–1028.
- PESSIN, G.; OSÓRIO, F. S. Avaliação de um modelo tolerante a falhas para atuação tática de grupos robóticos utilizando algoritmos genéticos. In: *XXXV Conferencia Latinoamericana de Informática (XXXV CLEI)*, 2009b, p. 1–10.
- PESSIN, G.; OSÓRIO, F. S. Otimização por enxame de partículas aplicado à formação e atuação de grupos robóticos. *Scientia (Unisinos)*, v. 20, p. 94–106, 2009c.

- PESSIN, G.; OSÓRIO, F. S. Evaluation of a fault-tolerant model for tactic operations of mobile robotic groups using genetic algorithms. *CLEI Electronic Journal*, v. 13, p. 1–8, 2010.
- PESSIN, G.; OSÓRIO, F. S.; DIAS, M. A.; SOUZA, J. R.; NETO, D. F. Avaliação de técnicas de otimização aplicadas à formação e atuação de grupos robóticos. In: *III Workshop on Computational Intelligence (WCI 2010), Joint Conference 2010 Workshop Proceedings (SBIA/SBRN/JRI)*, Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2010b, p. 494–499.
- PESSIN, G.; OSÓRIO, F. S.; MUSSE, S.; NONNENMACHER, V.; FERREIRA, S. S. Simulação virtual de agentes autônomos para a identificação e controle de incêndios em reservas naturais. In: *IX Symposium on Virtual and Augmented Reality (SVR)*, 2007a, p. 236–245.
- PESSIN, G.; OSÓRIO, F. S.; SOUZA, J. R.; COSTA, F. G.; UHEYAMA, J.; WOLF, D. F.; BRAUN, T.; VARGAS, P. A. Evolving an indoor robotic localization system based on wireless networks. In: *13th International Conference on Engineering Applications of Neural Networks (EANN)*, 2012, p. 1–10.
- PESSIN, G.; OSÓRIO, F. S.; UHEYAMA, J.; SOUZA, J. R.; WOLF, D. F.; BRAUN, T.; VARGAS, P. A. Evaluating the impact of the number of access points in mobile robots localization using artificial neural networks. In: *The Fifth International Conference on COMMunication System softWare and MiddlewaRE (COMSWARE)*, New York: ACM, 2011a, p. 1–9.
- PESSIN, G.; OSÓRIO, F. S.; UHEYAMA, J.; WOLF, D. F.; BRAUN, T. Mobile robot indoor localization using artificial neural networks and wireless networks. In: *First Brazilian Conference on Critical Embedded Systems (CBSEC)*, São Paulo: EPUSP, 2011b, p. 89–94.
- PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Particle swarm optimization applied to intelligent vehicles squad coordination. In: *7th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2010c, p. 1–6.
- PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F.; BRASIL, C. R. S. Improving efficiency of a genetic algorithm applied to multi-robot tactic operation. In: *12th Ibero-American Conference on Artificial Intelligence (IBERAMIA), Advances in Artificial Intelligence, Lecture Notes in Computer Science (LNCS)*, 2010d, p. 50–59.
- PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F.; DIAS, M. A. Genetic algorithm applied to robotic squad coordination. In: *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, Los Alamitos, California: IEEE Computer Society, 2009, p. 169–174.
- PESSIN, G.; OSÓRIO, F. S.; MUSSE, S. R.; NONNENMACHER, V.; FERREIRA, S. S. Desenvolvimento de um ambiente para simulação de robótica móvel em Áreas florestais. In: *Anais do VI Simpósio de Informática da Região Centro do RS (SIRC-RS'07)*, Santa Maria, RS, Brasil, 2007b.
- PESSIN, G.; OSÓRIO, F. S.; MUSSE, S. R.; NONNENMACHER, V.; FERREIRA, S. S. Simulação virtual de agentes autônomos para a identificação e controle de incêndios em reservas naturais. In: *Anais do IX Symposium on Virtual and Augmented Reality (SVR'07)*, 2007c, p. 236–245.

- PESSIN, G.; OSÓRIO, F. S.; MUSSE, S. R.; NONNENMACHER, V.; FERREIRA, S. S. Utilizando agentes autônomos com aprendizado para a identificação e combate de incêndios em Áreas florestais. In: *Anais do VII Simpósio de Informática do Planalto Médio (SIPM'07)*, Passo Fundo, RS, Brasil, 2007d.
- PIERI, E. R. *Curso de robótica móvel*. Universidade Federal de Santa Catarina (UFSC), 2002.
- POMERLEAU, D. Defense and civilian applications of the alvinn robot driving system. In: *Proceedings of Government Microcircuit Applications Conference*, 1994, p. 358–362.
- PRADO, M. G. *Planejamento de trajetória para estacionamento de veículos autônomos*. Dissertação de Mestrado, Universidade de São Paulo (USP), 2013.
- PRESTRI, M. National geographic. Acesso em 9 de julho, 2010.
Disponível em <http://ngm.nationalgeographic.com>
- PUGH, J.; MARTINOLI, A. Multi-robot learning with particle swarm optimization. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS'06)*, Hakodate, Japan: ACM, 2006, p. 441–448.
- PYNADATH, D. V.; TAMBE, M. Multiagent teamwork: analyzing the optimality and complexity of key theories and models. In: *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2 (AAMAS '02)*, ACM, 2002, p. 873–880.
- QUINLAN, J. Induction of decision trees. *Machine Learning*, v. 1, p. 81–106, 1986.
- RAMOS, P. C. M. Sistema nacional de prevenção e combate a incêndios florestais. In: *Anais do I Forum Nacional sobre Incêndios Florestais (IPEF'95)*, Instituto de Pesquisas e Estudos Florestais, 1995, p. 29–38.
- RASMUSSEN, C. E. Gaussian processes in machine learning. In: *Advanced Lectures on Machine Learning*, v. 3176 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 63–71, 2004.
- REN, T. R.; KWOK, N. M.; LIU, D. K.; HUANG, S. D. Path planning for a robotic arm sand-blasting system. In: *Proceedings of the 2008 IEEE International Conference on Information and Automation*, 2008.
- RIBEIRO, G. A.; LIMA, G. S.; DE OLIVEIRA, A. L. S.; DE CAMARGOS, V. L.; MAGALHÃES, M. U. Eficiência de um retardante de longa duração na redução da propagação do fogo. *Revista Árvore*, v. 30, n. 6, p. 1025–1031, 2006.
- RIEDMILLER, M.; BRAUN, H. *Rprop – description and implementation details (technical report)*. Relatório Técnico, Universitat Karlsruhe, Deutschland, 1994.
- RIGOLOT, E. Fuel-break assessment with an expert appraisal approach. In: *Proceedings of IV International Conference on Forest Fire Research*, Coimbra, Portugal: Millpress Science Publishers, 2002.

- ROBLES, J.; DEICKE, M.; LEHNERT, R. 3d fingerprint-based localization for wireless sensor networks. In: *Positioning Navigation and Communication (WPNC)*, 2010.
- RONG, C.; WANG, Q.; HUANG, Y.; XIE, G.; WANG, L. Autonomous evolution of high-speed quadruped gaits using particle swarm optimization. *RoboCup 2008: Robot Soccer World Cup XII (Lecture Notes in Computer Science)*, v. 5399, p. 259–270, 2009.
- ROSEMBLATT, F. *Principles of neurodynamics*. Spartan, 1962.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol*, v. 65, n. 6, p. 386–408, 1958.
- ROSS, J. *The book of wireless: A painless guide to wi-fi and broadband wireless*. No Starch Press, 352 p., 2008.
- ROTHERMEL, R. C. *General technical report int-143: How to predict the spread and intensity of forest and range fires*. United States Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, 161p, 1983.
- RUMELHART, D. E.; MCCLELLAND, J. L. *Parallel distributed processing*. MIT Press, 1987.
- RUSSEL, S.; NORVIG, P. *Artificial intelligence: A modern approach*. Prentice Hall, 1132 p., 2002.
- SAHIN, E.; WINFIELD, A. Special issue on swarm robotics. *Swarm Intelligence*, v. 2, p. 69–72, 2008.
- SALES, D. O.; SHINZATO, P. Y.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Vision-based autonomous navigation system using ann and fsm control. In: *Robotics Symposium and Intelligent Robotic Meeting (LARS), 2010 Latin American*, 2010, p. 85–90.
- SAXENA, A. K.; VORA, M. Novel approach for the use of small world theory in particle swarm optimization. In: *Proceedings of 16th International Conference on Advanced Computing and Communications (ADCOM)*, 2008, p. 363–366.
- SCHMAJUK, N. A.; BLAIR, H. T. Place learning and the dynamics of spatial navigation: A neural network approach. *Adaptive Behavior*, v. 1, p. 353–385, 1993.
- SDL Simple directmedia layer. Acesso em 15 de janeiro, 2013.
Disponível em <http://www.libsdl.org>
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, 1998, p. 69–73.
- SHINZATO, P. Y.; WOLF, D. F. A road following approach using artificial neural networks combinations. *Journal of Intelligent & Robotic Systems*, v. 62, p. 527–546, 2011.
- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. The MIT Press, 472 p., 2011.

- SILVA, I. N. D.; SPATTI, D. H.; FLAUZINO, R. A. *Redes neurais artificiais para engenharia e ciências aplicadas – curso prático*. Artliber, 399 p., 2010.
- SILVEIRA, T.; DE OLIVEIRA, H. C. B.; DA SILVA, L. E. Controle de inércia para fuga de mínimos locais de funções não-lineares na otimização por enxame de partículas. In: *Anais do XXIX Congresso da Sociedade Brasileira de Computação - Encontro Nacional de Inteligência Artificial (VII ENIA)*, Bento Gonçalves, RS: Porto Alegre, RS: SBC, 2009, p. 739–748.
- SMITH, D. *Swarm applications: Robotic soccer*. Final year dissertation, Heriot-Watt University, 2012.
- SMITH, R. *Open dynamics engine v0.5 user guide*. 2006.
- SMITH, R. G.; DAVIS, R. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, v. 11, n. 1, p. 61–70, 1981.
- SOUZA, J. R.; PESSIN, G.; EBOLI, G. B.; MENDES, C. C. T.; OSÓRIO, F. S.; WOLF, D. F. Vision and gps-based autonomous vehicle navigation using templates and artificial neural networks. In: *27th Annual ACM Symposium on Applied Computing (SAC)*, New York: ACM, 2012a, p. 280–285.
- SOUZA, J. R.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Avaliação de árvores de decisão no controle de navegação robótica. In: *III Workshop on Computational Intelligence (WCI 2010), Joint Conference 2010 Workshop Proceedings (SBIA/SBRN/JRI)*, Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2010, p. 488–493.
- SOUZA, J. R.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Vision-based autonomous navigation using supervised learning techniques. In: *12th Engineering Applications of Neural Networks (EANN), IFIP Advances in Information and Communication Technology*, Boston: Springer, 2011a, p. 11–20.
- SOUZA, J. R.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F.; VARGAS, P. A. Combining evolution and training in a robotic controller for autonomous vehicle navigation. In: *Towards Autonomous Robotic Systems (TAROS), Advances in Autonomous Robotics, Lecture Notes in Computer Science (LNCS)*, 2012b, p. 426–427.
- SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; OSÓRIO, F. S.; WOLF, D. F. Vision-based autonomous navigation using neural networks and templates in urban environments. In: *First Brazilian Conference on Critical Embedded Systems (CBSEC)*, São Paulo: EPUSP, 2011b, p. 55–60.
- SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; OSÓRIO, F. S.; WOLF, D. F. Vision-based waypoint following using templates and artificial neural networks. *Neurocomputing*, v. 107, p. 77 – 86, 2013.
- SUMAN, B.; KUMAR, P. A survey of simulated annealing as a tool for single and multi-objective optimization. *Journal of the Operational Research Society*, v. 57, p. 1143–1160, 2006.

- SURVEYOR Surveyor srv-1 blackfin robot. Acesso em 14 de julho, 2010.
Disponível em <http://www.surveyor.com>
- TEOH, E.; TAN, K.; XIANG, C. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *Neural Networks, IEEE Transactions on*, v. 17, n. 6, p. 1623–1629, 2006.
- THRUN, S.; BURGARD, W.; FOX, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In: *Proc. IEEE International Conference on Robotics and Automation ICRA '00*, 2000, p. 321–328.
- TRINITY Trinity college fire-fighting home robot contest and the robowaiter contest. Acesso em 22 de julho, 2010.
Disponível em <http://www.trincoll.edu/events/robot>
- VARGAS, P. A.; BENHALEN, A. M.; PESSIN, G.; OSÓRIO, F. S. Applying particle swarm optimization to a garbage and recycling collection problem. In: *12th Annual Workshop on Computational Intelligence (UKCI)*, 2012, p. 1–8.
- WAIBEL, M.; KELLER, L.; FLOREANO, D. Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Computation*, v. 13, p. 648–660, 2009.
- WALL, M. *Galib: A c++ library of genetic algorithm components, version 2.4, user manual*. 1996.
- WALTER, W. An imitation of life. *Scientific American*, v. 182, p. 42–45, 1950.
- WELCH, G.; BISHOP, G.; VICCI, L.; BRUMBACK, S.; KELLER, K.; COLUCCI, D. High-performance wide-area optical tracking: The hiball tracking system. *Presence: Teleoper. Virtual Environ.*, v. 10, n. 1, p. 1–21, 2001.
- WERFEL, J.; INGBER, D.; NAGPAL, R. Collective construction of environmentally-adaptive structures. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, p. 2345–2352.
- WETTERGREEN, D. Depthx: Deep phreatic thermal explorer. Acesso em 14 de janeiro, 2013.
Disponível em http://www.ri.cmu.edu/research_project_detail.html?type=publication&project_id=544&menu_id=261
- WHITLEY, D. The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In: *Proceedings of the third international conference on Genetic algorithms*, 1989, p. 116–121.
- WIDROW, B.; HOFF, M. E. Adaptive switching circuits. In: *Institute of Radio Engineers (IRE) Western Electronic Show and Convention Record (WESCON)*, 1960, p. 96–104.

- WIKIPÉDIA:VALOR-P Wikipédia: a enciclopédia livre: Valor-p. Acesso em 11 de fevereiro, 2013.
Disponível em <http://pt.wikipedia.org/wiki/Valor-p>
- WILD, A. Alex wild photography. Acesso em 8 de julho, 2010.
Disponível em <http://www.alexanderwild.com>
- WOLF, D.; HOWARD, A.; SUKHATME, G. S. Towards geometric 3d mapping of outdoor environments using mobile robots. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, p. 1507–1512.
- WOOLDRIDGE, M. *An introduction to multiagent systems*. John Wiley & Sons, 348 p., 2002.
- ZELL, A.; MAMIER, G.; MACHE, M.; HÜBNER, R.; HERRMANN, S.; SOYEZ, T.; SCHMALZL, M.; SOMMER, T.; HATZIGEORGIOU, A.; POSSELT, D.; SCHREINER, T.; KETT, B.; CLEMENTE, G.; WIELAND, J. *Stuttgart neural network simulator - user manual, version 4.1 (report no. 6/95)*. Relatório Técnico, University of Stuttgart, 1995.
- ZINCK, R. D.; GRIMM, V. Unifying wildfire models from ecology and statistical physics. *The American Naturalist*, v. 174, p. E170–E185, 2009.
- ZOU, A. M.; HOU, Z. G.; FU, S. Y.; TAN, M. Neural networks for mobile robot navigation: A survey. In: WANG, J.; YI, Z.; ZURADA, J.; LU, B.-L.; YIN, H., eds. *Advances in Neural Networks - ISNN 2006*, v. 3972 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 1218–1226, 2006.

Artigos Publicados

A.1 Publicados em periódicos

1. SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; OSÓRIO, F. S.; WOLF, D. F. Vision-based waypoint following using templates and artificial neural networks. *Neurocomputing*, vol. 107, p. 77–86, 2013. <http://dx.doi.org/10.1016/j.neucom.2012.07.040>
2. MARINELLO, S. H. M.; UHEYAMA, J.; PESSIN, G.; OSÓRIO, F. S.; VARGAS, P. A. Determinando posições relativas de nós móveis por meio de bluetooth e aprendizado de máquina. *Revista de Exatas e Tecnológicas (RETEC) ISSN 2236-739X*, vol. 3, p. 1–10, 2012.
3. PESSIN, G.; OSÓRIO, F. S. Evaluation of a fault-tolerant model for tactic operations of mobile robotic groups using genetic algorithms. *CLEI Electronic Journal*, vol. 13, p. 1–8, 2010.
4. HATA, A. Y.; WOLF, D. F.; PESSIN, G.; OSÓRIO, F. S. Terrain mapping and classification in outdoor environments using neural networks. *International Journal of u- and e-Service, Science and Technology*, vol. 2, p. 51–61, 2009.
5. PESSIN, G.; OSÓRIO, F. S. Otimização por enxame de partículas aplicado à formação e atuação de grupos robóticos. *Scientia (Unisinos)*, vol. 20, p. 94–106, 2009. <http://dx.doi.org/10.4013/sct.2009.20.2.03>

A.2 Publicados em conferências

1. PESSIN, G.; OSÓRIO, F. S.; SOUZA, J. R.; COSTA, F. G.; UEYAMA, J.; WOLF, D. F.; BRAUN, T.; VARGAS, P. A. Evolving an indoor robotic localization system based on wireless networks. *13th International Conference on Engineering Applications of Neural Networks (EANN)*, 2012, p. 1–10.
2. SOUZA, J. R.; PESSIN, G.; EBOLI, G. B.; MENDES, C. C. T.; OSÓRIO, F. S.; WOLF, D. F. Vision and gps-based autonomous vehicle navigation using templates and artificial neural networks. *27th Annual ACM Symposium on Applied Computing (SAC)*, New York: ACM, 2012, p. 280–285. <http://dx.doi.org/10.1145/2245276.2245332>
3. FERNANDES, L. C.; SOUZA, J. R.; SHINZATO, P. Y.; PESSIN, G.; MENDES, C. C. T.; OSÓRIO, F. S.; WOLF, D. F. Intelligent robotic car for autonomous navigation: Platform and system architecture. *II Brazilian Conference on Critical Embedded Systems (CBSEC)*, Los Alamitos, CA, USA: IEEE Computer Society, 2012, p. 12–17. <http://dx.doi.org/10.1109/CBSEC.2012.26>
4. COSTA, F. G.; UEYAMA, J.; BRAUN, T.; PESSIN, G.; OSÓRIO, F. S.; VARGAS, P. A. The use of unmanned aerial vehicles and wireless sensor network in agricultural applications. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2012, p. 5045-5048. <http://dx.doi.org/10.1109/IGARSS.2012.6352477>
5. AILINCA, M.; PARANTHAMAN, P. K.; PESSIN, G.; OSÓRIO, F. S.; SOUZA, J. R.; WOLF, D. F.; UEYAMA, J.; VARGAS, P. A. Uncovering new neural network topologies in real world robot applications. *12th Annual Workshop on Computational Intelligence (UKCI) – Studies in Mathematical and Computer Science*, Edinburgh, UK: Heriot-Watt University, 2012, p. 53–58.
6. VARGAS, P. A.; BENHALEN, A. M.; PESSIN, G.; OSÓRIO, F. S. Applying particle swarm optimization to a garbage and recycling collection problem. *12th Annual Workshop on Computational Intelligence (UKCI)*, 2012, p. 1–8. <http://dx.doi.org/10.1109/UKCI.2012.6335780>
7. SOUZA, J. R.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F.; VARGAS, P. A. Combining evolution and training in a robotic controller for autonomous vehicle navigation. *Towards Autonomous Robotic Systems (TAROS), Advances in Autonomous Robotics, Lecture Notes in Computer Science (LNCS)*, 2012, p. 426–427.
8. PESSIN, G.; OSÓRIO, F. S.; UEYAMA, J.; WOLF, D. F.; BRAUN, T. Mobile robot indoor localization using artificial neural networks and wireless networks. *First Brazilian Conference on Critical Embedded Systems (CBSEC)*, São Paulo: EPUSP, 2011, p. 89–94.
9. SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; OSÓRIO, F. S.; WOLF, D. F. Vision-based autonomous navigation using neural networks and templates in urban environments. *First Brazilian Conference on Critical Embedded Systems (CBSEC)*, São Paulo: EPUSP, 2011, p. 55–60.

10. PESSIN, G.; OSÓRIO, F. S.; UEYAMA, J.; SOUZA, J. R.; WOLF, D. F.; BRAUN, T.; VARGAS, P. A. Evaluating the impact of the number of access points in mobile robots localization using artificial neural networks. *The Fifth International Conference on Communication System softWare and MiddlewaRE (COMSWARE)*, New York: ACM, 2011, p. 1–9. <http://dx.doi.org/10.1145/2016551.2016561>
11. SOUZA, J. R.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Vision-based autonomous navigation using supervised learning techniques. *12th Engineering Applications of Neural Networks (EANN), IFIP Advances in Information and Communication Technology*, Boston: Springer, 2011, p. 11–20.
http://dx.doi.org/10.1007/978-3-642-23957-1_2
12. COSTA, F. G.; BRAUN, T.; UEYAMA, J.; PESSIN, G.; OSÓRIO, F. S. Arquitetura baseada em veículos aéreos não tripulados e redes de sensores sem fio para aplicações agrícolas. *VIII Congresso Brasileiro de Agroinformatica (SBIAGro2011)*, 2011, p. 1–5.
13. PESSIN, G.; OSÓRIO, F.; HATA, A.; WOLF, D. Intelligent control and evolutionary strategies applied to multirobotic systems. *Industrial Technology (ICIT), 2010 IEEE International Conference on*, 2010, p. 1427–1432.
<http://dx.doi.org/10.1109/ICIT.2010.5472498>
14. PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Particle swarm optimization applied to intelligent vehicles squad coordination. *7th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2010, p. 1–6.
<http://dx.doi.org/10.3182/20100906-3-IT-2019.00070>
15. OSÓRIO, F. S.; WOLF, D. F.; BRANCO, K. C.; PESSIN, G. Mobile robots design and implementation: From virtual simulation to real robots. *Proceedings of IDMMME – Virtual Concept 2010*, 2010, p. 1–6.
16. PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F.; BRASIL, C. R. S. Improving efficiency of a genetic algorithm applied to multi-robot tactic operation. *12th Ibero-American Conference on Artificial Intelligence (IBERAMIA), Advances in Artificial Intelligence, Lecture Notes in Computer Science (LNCS)*, 2010, p. 50–59.
http://dx.doi.org/10.1007/978-3-642-16952-6_6
17. SALES, D. O.; SHINZATO, P. Y.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Vision-based autonomous navigation system using ann and fsm control. *Robotics Symposium and Intelligent Robotic Meeting (LARS), 2010 Latin American*, 2010, p. 85–90.
<http://dx.doi.org/10.1109/LARS.2010.26>
18. SOUZA, J. R.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Avaliação de árvores de decisão no controle de navegação robótica. *III Workshop on Computational Intelligence (WCI 2010), Joint Conference 2010 Workshop Proceedings (SBIA/SBRN/JRI)*, Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2010, p. 488–493.
19. PESSIN, G.; OSÓRIO, F. S.; DIAS, M. A.; SOUZA, J. R.; NETO, D. F. Avaliação de técnicas de otimização aplicadas à formação e atuação de grupos robóticos. *III Workshop*

- on Computational Intelligence (WCI 2010), Joint Conference 2010 Workshop Proceedings (SBIA/SBRN/JRI)*, Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2010, p. 494–499.
20. PESSIN, G.; OSÓRIO, F. S. Algoritmos genéticos aplicados à formação e atuação de grupos robóticos. *Encontro Nacional de Inteligência Artificial (VII ENIA), Anais do XXIX Congresso da Sociedade Brasileira de Computação*, Porto Alegre, RS: Sociedade Brasileira de Computação (SBC), 2009, p. 1019–1028.
 21. PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F.; DIAS, M. A. Genetic algorithm applied to robotic squad coordination. *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, Los Alamitos, California: IEEE Computer Society, 2009, p. 169–174.
<http://dx.doi.org/10.1109/CERMA.2009.17>
 22. HATA, A. Y.; WOLF, D. F.; PESSIN, G.; OSÓRIO, F. S. Terrain mapping and classification using neural networks. *International Conference on Hybrid Information Technology (ICHIT)*, New York: ACM, 2009, p. 438–442.
<http://dx.doi.org/10.1145/1644993.1645074>
 23. OSÓRIO, F. S.; WOLF, D. F.; PESSIN, G.; SIMOES, E.; BRANCO, K. C. Simulação Virtual de Carros em Jogos e Aplicações de I.A. *Tutorial no VIII Brazilian Symposium on Games and Digital Entertainment (SBGames)*, 2009.
 24. PESSIN, G.; OSÓRIO, F. S. Avaliação de um modelo tolerante a falhas para atuação tática de grupos robóticos utilizando algoritmos genéticos. *XXXV Conferencia Latinoamericana de Informática (XXXV CLEI)*, 2009, p. 1–10.

Incêndios em Ambientes Naturais

Este anexo apresenta uma compilação de trabalhos sobre incêndios, combustíveis florestais, técnicas e equipamentos reais de operação em combate a incêndios, bem como apresenta algumas considerações sobre incêndios em ambientes naturais. As informações apresentadas neste anexo foram utilizadas como base do desenvolvimento do simulador de incêndio e da proposta de combate de incêndio apresentada no Capítulo 6. As informações contidas neste anexo estendem as informações previamente descritas em (Pessin *et al.*, 2007a) e (Pessin, 2008).

Fogo, de um modo geral, é o termo aplicado ao fenômeno físico resultante da rápida combinação entre o oxigênio e uma substância combustível com produção de calor, luz e chamas. A reação da combustão completa da madeira, que poderia ser generalizada para todo o material combustível da floresta, envolve três elementos básicos: combustível, oxigênio e calor. Em qualquer incêndio florestal é necessário haver combustível para queimar, oxigênio para manter as chamas e calor para iniciar e continuar o processo de queima. Essa interrelação entre os três elementos básicos da combustão é conhecida como triângulo do fogo. A ausência, ou redução abaixo de certos níveis, de qualquer um dos componentes do triângulo do fogo inviabiliza o processo da combustão (LIF/UFPR, 2013).

B.1 Combustíveis Florestais

Os combustíveis florestais, produto da cobertura vegetal e sua dinâmica, são importantes parâmetros nos processos de ignição e propagação dos incêndios. O conhecimento das características básicas dos combustíveis, tais como: tipo, quantidade, continuidade e arranjo são muito úteis, tanto na estimativa do risco como na previsão do comportamento dos incêndios (Castro *et al.*, 2003; Chuvieco e Congalton, 1989; Gouma e Chronopoulou, 1998). Estudos dos modelos de florestas e resíduos florestais são de grande importância para o aprimoramento dos modelos de simulação a serem implementados em ambientes virtuais. Além destes modelos, também é importante que seja feito um mapeamento do terreno e do tipo de vegetação da cobertura deste terreno, caso se deseje realizar uma reprodução mais fiel de um determinado ambiente.

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

- **Modelo 1, herbáceo:** Pasto fino, seco e baixo, com altura abaixo do joelho, que cobre completamente o solo. O mato ou as árvores cobrem menos de 1/3 da superfície. Os incêndios propagam-se com grande velocidade pelo pasto fino.
- **Modelo 2, herbáceo:** Pasto contínuo, fino, seco e baixo, com presença de mato ou árvores que cobrem entre 1/3 e 2/3 da superfície. Os combustíveis são formados pelo pasto seco, folhada e ramos caídos da vegetação lenhosa. Os incêndios propagam-se rapidamente pelo pasto fino.
- **Modelo 3, herbáceo:** Pasto contínuo, espesso, com mais de 1 m sendo que 1/3 ou mais do pasto deverá estar seco. Os incêndios são mais rápidos e de maior intensidade.
- **Modelo 4, arbustivo:** Matos ou árvores jovens muito densos, com cerca de 2 m de altura. Continuidade horizontal e vertical do combustível. Abundância de combustível lenhoso morto (ramos) sobre as plantas vivas. O fogo propaga-se rapidamente sobre as copas dos matos com grande intensidade e com chamas grandes.
- **Modelo 5, arbustivo:** Mato denso baixo, com uma altura inferior a 0,6 m. Apresenta cargas ligeiras de folhada do mesmo mato, que contribui para a propagação do fogo em situação de ventos fracos. Fogos de intensidade moderada.
- **Modelo 6, arbustivo:** Mato mais velho do que no modelo 5, com alturas compreendidas entre os 0,6 m e 2 m de altura. Os combustíveis vivos são mais escassos e dispersos. No conjunto é mais inflamável do que o modelo 5. O fogo propaga-se através do mato com ventos moderados a fortes.

- **Modelo 7, arbustivo:** Mato de espécies muito inflamáveis, de 0,6 m a 2 m de altura, que propaga o fogo debaixo das árvores. O incêndio desenvolve-se com teores mais altos de umidade do combustível morto do que nos outros modelos, devido à natureza mais inflamável dos outros combustíveis vivos.
- **Modelo 8, manta morta:** Folhada em bosque denso. A folhada forma uma capa compacta ao estar formada de agulhas pequenas (5 cm ou menos) ou por folhas planas não muito grandes. Os fogos são de fraca intensidade, com chamas curtas e que avançam lentamente.
- **Modelo 9, manta morta:** Folhada em bosque denso que se diferencia do modelo 8 por formar uma camada pouco compacta e arejada. Fogo mais rápido e com chamas mais compridas do que as do modelo 8.
- **Modelo 10, manta morta:** Restos lenhosos originados naturalmente, incluindo lenha grossa caída como consequência de vendavais, pragas intensas ou excessiva maturação da massa, com presença de vegetação herbácea que cresce entre os restos lenhosos.
- **Modelo 11, resíduos lenhosos:** Resíduos ligeiros com diâmetro menor que 7,5 cm, recentes, de tratamentos silvícolas ou de aproveitamentos, formando uma capa pouco compacta de altura por volta de 30 cm. Incêndios com intensidades elevadas.
- **Modelo 12, resíduos lenhosos:** Resíduos de exploração mais pesados do que no modelo 11, formando uma capa contínua de altura até 60 cm. Incêndios com intensidades elevadas.
- **Modelo 13, resíduos lenhosos:** Grandes acúmulos de resíduos de exploração grossos com diâmetro maior que 7,5 cm e pesados, cobrindo todo o solo.

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

B.2 Técnicas Reais de Operação

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

B.2.1 Operações de combate

A operação de combate ou supressão de um incêndio envolve seis etapas distintas. Essas etapas são as seguintes (LIF/UFPR, 2013):

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

6. **Combate:** Tempo consumido na operação de combate ou eliminação do incêndio. Devem ser consideradas para o tempo de combate as diferentes técnicas de combate a incêndios.

B.2.2 Equipamentos de combate

O tipo e a quantidade de equipamento para combate a incêndios florestais em uma instituição florestal dependem de fatores como tipo de vegetação, topografia, tamanho da área e pessoal disponível. De um modo geral, as ferramentas e equipamentos usados no combate aos incêndios florestais são os seguintes (LIF/UFPR, 2013):

- **Ferramentas manuais:** enxadas, machados, foices, pás, abafadores, extintores e lança-chamas (usado na criação de contra-fogos).
- **Equipamentos de apoio:** lanternas (durante a noite), capacetes, cantis e equipamentos de primeiros socorros (com medicamentos para queimaduras, cortes e soro antiofídico).
- **Equipamentos motorizados:** moto-serra (usada para derrubar rapidamente árvores que estejam queimando ou para abrir pequenos aceiros) e atomizador costal (usado para lançar água ou retardante químico em incêndios superficiais).
- **Equipamentos pesados:** trator com lâmina (indispensável no combate a grandes incêndios, principalmente quando se precisa abrir aceiros¹ em vegetação pesada) e motoniveladora (usada para abrir ou ampliar aceiros).
- **Equipamentos de bombeamento de água:** moto-bomba portátil (útil quando existe uma boa rede de pontos de captação de água), carro-tanque (somente opera com eficiência em boas estradas), helicóptero e avião-tanque (Figura B.1) (Bombardier, 2007) (despejam água retardando a propagação do fogo até que as equipes de terra realizem a extinção total).

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

¹ Aceiro é uma área livre de vegetação, onde o fogo, por não encontrar material combustível, se extingue.



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

B.2.3 Métodos e estratégias de combate

Existem pelo menos quatro métodos de combate ao fogo nos incêndios florestais (LIF/UFPR, 2013):

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

- **Método aéreo:** Usado nos incêndios de copa, de grande intensidade e área e em locais de difícil acesso às brigadas de incêndio. São usados aviões e helicópteros especialmente construídos ou adaptados para o combate ao incêndio.

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

B.2.4 Sistemas de detecção

Quanto mais rápida a detecção de um foco de incêndio, maior é a chance de êxito no combate e na minimização dos danos. Os métodos de detecção podem ser, de acordo com Ramos (1995):

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

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

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

(CNMC) em Brasília, onde são gerados mapas nos quais são identificadas a localização e a intensidade do foco.

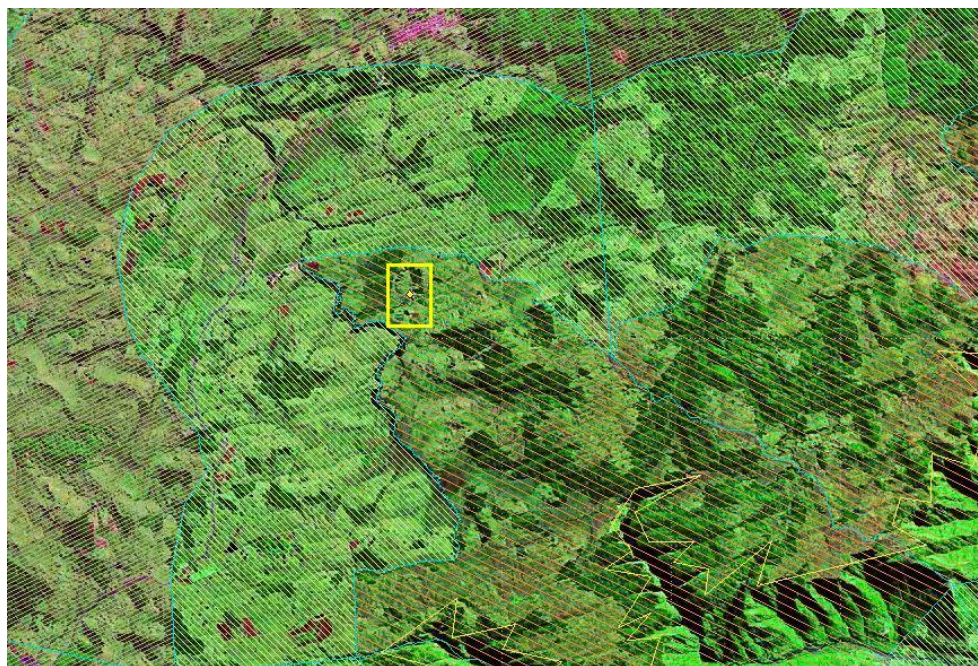
O Instituto Nacional de Pesquisas Espaciais (INPE), através do Sistema de Monitoramento, Prevenção e Controle de Incêndios Florestais na Amazônia (PROARCO), monitora as ocorrências de incêndios para todos os estados brasileiros através de sensores espaciais (Batista, 2004). Sensores AVHRR captam e registram qualquer temperatura acima de 47°C, interpretadas como sendo um foco de calor. Apesar da resolução espacial ser baixa (1,1 km), queimadas com áreas de no mínimo 900 m² podem ser detectadas (Antunes, 2000). Dados obtidos através das imagens termais de satélites meteorológicos são integrados a dois sistemas de informação geográfica que podem ser acessados pela internet, como mostram as Figuras B.2(a) e B.2(b) obtidas no endereço <http://www.dpi.inpe.br/proarco/bdqueimadas>. Todo o Brasil e grande parte da América do Sul são cobertos pelas imagens (CPTEC/INPE, 2007).

B.3 Considerações sobre Incêndios em Ambientes Naturais

Os aceiros têm sido uma alternativa viável e de amplo uso no meio florestal, nas unidades de conservação e nas margens de rodovias para a redução na propagação do fogo (Ribeiro *et al.*, 2006). O principal objetivo de um aceiro é o de quebrar a continuidade do material combustível, dividindo uma área em compartimentos. O surgimento de novos produtos e de novos equipamentos de aplicação tem sugerido que aceiros químicos podem ser muito eficientes em combate a incêndios florestais (Ribeiro *et al.*, 2006). A eficiência de um aceiro depende da composição da vegetação e da estrutura do material combustível existente em suas proximidades, principalmente pelas características das espécies dominantes, altura e proximidade da vegetação, assim, vários estratos da vegetação devem ser considerados (Rigolot, 2002).

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

Apresentamos neste anexo uma compilação de trabalhos sobre incêndios, combustíveis florestais, técnicas e equipamentos reais de operação em combate a incêndios, bem como apre-



(a)

| CAMPO | VALOR |
|--|------------------------------|
| ID | S290654050084820060926171400 |
| Lat | -29.115000 |
| Long | -50.146667 |
| LatGMS | S 29 6 54.00 |
| LongGMS | O 50 8 48.00 |
| Data-Hora/Date-Time/Fecha-Hora | 2006-09-26 17:14:00 |
| Satélite/Satellite/Satélite | AQUA |
| Município/City/Localidad | Cambará do Sul |
| Estado/State/Provincia | RS |
| Unidade Conservacao/Conservation Unit/Area de Conservación | P.N. de Aparados da Serra |
| Terra Indigena/Reserved for Indian/Área Indígena | Amarelo |
| Floresta Nacional/National Forest/Floresta Nacional | 0.00 |

(b)

Figura B.2: (a) Foco de incêndio próximo ao Parque Nacional dos Aparados da Serra, imagem da Divisão de Processamento de Imagens do INPE, satélite *LandSat* 2000 (DPI/INPE, 2007). (b) Tabela de informações disponibilizada junto a imagem do foco de incêndio (DPI/INPE, 2007).

senta algumas considerações sobre incêndios em ambientes naturais. As informações apresentadas neste anexo foram utilizadas como base do desenvolvimento do simulador de incêndio e da proposta de combate de incêndio apresentada no Capítulo 6. As informações contidas neste anexo estendem as informações previamente descritas em (Pessin *et al.*, 2007a) e (Pessin, 2008). Propomos um agente monitor responsável pelas etapas de detecção, planejamento e comunicação. Sendo a forma de combate a incêndio implementada utilizando equipamento pesado, assim os veículos de combate a incêndio são motoniveladoras; e o método de combate é o indireto; utilizando as motoniveladoras para criação de um aceiro. O terreno escolhido é semelhante aos parques nacionais da região nordeste do Rio Grande do Sul, que apresentam predomínio de combustível florestal do Modelo 1.