
Rede neural hierárquica para aprendizado de
exames de robôs em tempo real

Murillo Rehder Batista

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Rede neural hierárquica para aprendizado de enxames de robôs em tempo real

Murillo Rehder Batista

Orientadora: Profa. Dra. Roseli Aparecida Francelin Romero

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

USP – São Carlos
Junho de 2014

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

B333r Batista, Murillo Rehder
Rede neural hierárquica para aprendizado de
exames de robôs em tempo real / Murillo Rehder
Batista; orientadora Roseli Aparecida Francelin
Romero. -- São Carlos, 2014.
98 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2014.

1. Robótica. 2. Exames de Robôs. 3. Aprendizado
de Robôs. 4. Diagramas de Voronoi. I. Romero, Roseli
Aparecida Francelin, orient. II. Título.

Agradecimentos

Primeiramente, agradeço a Deus pela força espiritual nos momentos difíceis, que foram muitos.

Agradeço meus pais, Marcos e Vera, pelas muitas oportunidades que tive na vida desde pequeno. Agradeço, também, a meus irmãos Milena e Marcos, que foram uma inspiração para entrar em uma boa universidade.

Agradeço à Professora Roseli por ter me orientado desde a Iniciação Científica, que foi lá pro final do quarto ano de Graduação, mesmo não tendo um histórico “ideal” de graduação.

Agradeço ao Rodrigo Calvo, cujo trabalho foi o ponto de partida do meu Mestrado, e que me ajudou a sanar diversas dúvidas de seu trabalho, e ao Prof. Maurício Figueiredo, com o qual tive uma conversa que ajudou a fazer alguns aspectos do projeto fluírem.

Agradeço à equipe Warthog Robotics, e por consequência ao USPDroids, porque sem a criação deste grupo eu dificilmente teria entrado na Robótica, e provavelmente não teria tido a chance de realizar uma pós-graduação.

Agradeço aos professores que ministraram as disciplinas que cursei durante meu Mestrado. Foram muito trabalhosas, mas certamente valeram a pena.

Agradeço as pessoas que tiveram que ouvir eu falando deste projeto repetidas e repetidas vezes ou que me ajudaram em algum momento. Em particular Raphael, Eduardo, Marcelo, Adam, Luiz, Valéria, Bilzã, Patrick, Jean, Murillo Carneiro, Filipe Neto, Fabiano, Alberione e muitas outras pessoas que não me lembro enquanto escrevo isto mas existem.

Agradeço ao Rafael, Juninho, Liuri e Yago por alguns bons momentos jogando ou falando de coisas não relacionadas ao projeto, além de vários dos nomes citados acima pelo mesmo motivo.

Agradeço à Janaína, que está conseguindo me tolerar já faz bastante tempo, e que é um enorme exemplo de dedicação e de disciplina para mim.

Agradeço à CAPES e à FAPESP, pelo financiamento do projeto através de bolsas.

Resumo

Uma tendência crescente entre os pesquisadores da Robótica Móvel é a elaboração de sistemas robóticos descentralizados denominados enxames de robôs, nos quais a ação conjunta de cada agente leva à execução de tarefas de maneira mais robusta que quando realizada por um único robô. Um acréscimo adicional à robustez é conveniente em tais sistemas para que eles sejam de maior confiabilidade no mundo real. Neste trabalho, uma rede neural hierárquica desenvolvida para o aprendizado em tempo real inicialmente elaborada para o aprendizado de navegação de um único robô será estendida para controlar um enxame de robôs. O sistema realiza um balanceamento da influência de comportamentos implementados previamente em um robô de acordo com conhecimentos obtidos através da interação do mesmo com o ambiente. Cada robô possui sua própria rede neural, adquirindo seu conhecimento tanto independentemente quanto com o compartilhamento de informações com outros robôs. Espera-se que o uso de tal arquitetura permita uma adaptação mais rápida dos robôs ao ambiente, permitindo uma mudança em tempo real de seus parâmetros de acordo com as peculiaridades do ambiente no qual os robôs estão inseridos. A tarefa de escolta de um robô pelos demais é adotada para a avaliação de desempenho do modelo de rede neural proposto. Dois comportamentos são ponderados pela rede neural hierárquica: o de manutenção de uma distância preestabelecida a um agente e um outro de cobertura de área baseado em Diagramas Centroidais de Voronoi. Os testes foram feitos nos ambientes Player/Stage e indicam que a rede neural hierárquica torna os robôs capazes não apenas de aprender à medida que interagem com ambiente como de utilizar este conhecimento em tempo real para realizar a escolta de forma bem sucedida.

Abstract

A growing trend among Mobile Robotics researchers is developing robot swarms, in which a decentralized robot team solves tasks by combining simple behaviors. It is convenient to have mechanisms to increase a robot system's robustness. In this work, a neural network inspired in behavioral analysis is used to make robots from a swarm to learn how to act properly. This network combines two innate behaviors and, according to its experience, learns with the robots' mistakes how to make this combination. Each robot has access to its own independent neural network, and can share its knowledge with neighboring robots. It is expected that such architecture learns by itself when to stimulate or suppress each behavior's influence as it interacts with the environment. The task chosen to evaluate the proposed system is the escorting of a mobile agent. Two behaviors are balanced to achieve an escorting behavior: maintenance of a minimum distance between a robot and the escort target and an area coverage method based on Centroidal Voronoi Tessellations. Tests were made using the Player/Stage simulator, and they show that the robots not only are capable of adapting themselves but also are able to use the stored knowledge to improve their effectiveness in doing the desired task.

Sumário

Resumo	iii
Abstract	v
Sumário	vii
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivos	5
1.3 Organização do Texto	7
2 Revisão Bibliográfica	9
2.1 Sistemas Multirroboóticos	9
2.1.1 Enxames de Robôs	12
2.1.2 Gerenciamento de <i>software</i> dos robôs	15
2.2 Aprendizado em Robôs Móveis em tempo real	16
2.2.1 Aprendizado em Sistemas Multirroboóticos	19
3 Fundamentação Teórica	21
3.1 A rede neural hierárquica ANS	21
3.2 Diagramas Centroidais de Voronoi	26
3.2.1 Diagramas Centroidais de Voronoi em Sistemas Multirroboóticos	28
3.3 Considerações Finais	29
4 Descrição da Pesquisa	31
4.1 Adaptações e implementação da rede AIS	31
4.2 Implementação do neurônio e da Rede do Módulo Coordenador	32
4.3 Implementação do módulo de saída e da AIS	34
4.4 Cobertura de área: SLACS	36
4.5 Aplicação do SLACS para a Escolta	39
4.6 Adaptações da Rede Neural Hierárquica para Enxames de Robôs	41
4.6.1 Compartilhamento de dados	43
5 Experimentos realizados	45
5.1 Avaliação da cobertura de área	45
5.2 Avaliação da escolta	51

5.2.1	Modelo de Robô Adotado	54
5.2.2	Avaliação da escolta	54
5.2.3	Testes com influências constantes	58
5.2.4	Primeiro conjunto de testes de escolta utilizando a rede AIS	64
5.2.5	Segundo conjunto de testes de escolta utilizando a rede AIS	75
5.3	Discussão	84
6	Conclusão	89
6.1	Sugestões de trabalhos futuros	90
	Referências Bibliográficas	98

Introdução

1.1 Contextualização

Um robô móvel autônomo pode ser definido como uma máquina que navega e atua em um ambiente de modo independente de um operador humano, de acordo com funções previamente definidas (Arkin, 1998). Objetiva-se nas pesquisas relacionadas aos robôs móveis projetar tais máquinas de modo que elas sejam capazes de realizar atividades diversas. Tais robôs são dotados de sensores capazes de obter dados do ambiente no qual o robô atua, como a distância em relação a obstáculos, iluminação e temperatura do ambiente.

Além de sensores, robôs móveis autônomos têm dispositivos capazes de alterar o ambiente denominados atuadores, que podem ser tanto seu mecanismo de locomoção (rodas, esteiras, patas) quanto garras mecânicas para capturar objetos. É importante salientar que o próprio robô é considerado parte integrante do ambiente no qual está situado, razão pela qual instrumentos de locomoção são classificados como atuadores: eles alteram a pose de um agente físico do ambiente, que é o robô.

Dentre as tarefas que podem ser executadas por robôs móveis, estão atividades monótonas ou de grande risco para pessoas. Muitas destas tarefas, em especial as de alto risco, podem vir a causar danos ao robô suficientes para impossibilitá-lo de prosseguir com sua atividade, como danos em suas rodas ou em sua bateria. Portanto, é importante que uma

solução para um problema a partir da utilização de robôs seja robusta. Mesmo com o avanço intenso e contínuo dos *hardwares*, qualquer aparelho eletrônico está suscetível a falhas internas. Este fato não desencoraja, entretanto, o esforço empenhado na área por pesquisadores das mais diversas áreas como Ciência de Computação, Engenharias Mecânica, Eletrônica, Mecatrônica, Mecânica, etc.

Uma solução a nível físico para o problema de eventuais falhas de um robô é, basicamente, utilizar mais de um robô para executar a mesma tarefa, premissa básica dos sistemas multirrobóticos. Existindo um maior número de robôs, caso um seja incapaz de prosseguir, os demais robôs poderão ser capazes de realizar a tarefa. Além disso, em um grande número de tarefas, como a exploração de ambientes desconhecidos e a vigilância de ambientes, a existência de vários robôs implica em maior eficiência porque há um maior número de entidades realizando a mesma atividade.

Outra vantagem importante está no fato de que robôs podem interagir com os demais robôs através de comunicação. Isto permite que, por exemplo, um robô pertencente ao grupo que está procurando por uma vítima em um desmoronamento comunique aos demais a localização da mesma, para que os demais robôs ajudem a resgatá-la. Ou ainda que um grupo que está fazendo o mapeamento de um ambiente desconhecido compartilhe informações para que não haja necessidade de outros robôs mapearem novamente.

A interação entre unidades que realizam atividades simples que resultam em efeitos mais complexos quando considera-se o comportamento do todo é um assunto de grande interesse da comunidade científica, particularmente dos pesquisadores de Ciências de Computação, não se limitando à Robótica, sendo denominado estudo de sistemas multiagentes.

Um trabalho clássico sobre este tema é o Game of Life (Conway, 1970), pioneiro no tópico de vida artificial, no qual um ambiente é representado por uma matriz cujas células, ou posições, são classificadas como “vivas” ou “mortas”. Quando uma posição “viva” possui um número de células adjacentes “vivas” igual a três, esta “morre”, enquanto uma posição “morta” que possua três vizinhos “vivos” torna-se “viva”. Apesar de simples, este conjunto de regras permite a geração de uma gama de sistemas complexos.

Esta possibilidade de realização de ações de maior complexidade através de ações simples é, também, válida na Robótica. Um exemplo claro está no Futebol de Robôs, em que um grupo de robôs navega em um campo com o

objetivo de jogar futebol. Enquanto um robô é designado para ser um goleiro, que evita que a bola entre no gol, outros visam avançar a bola até o campo adversário, e alguns buscam chutar a gol. O investimento neste problema é suficiente para a existência de grandes competições de Robótica, como a Robocup, competição mundial de robótica, e a LARC, a competição Latino-americana.

Um trabalho pioneiro neste tópico foi o de Balch et al. (1995), em que uma equipe de robôs foi utilizada para coletar lixo. Grupos de robôs que vigiam um ambiente são propostos no trabalho de Lee et al. (2010). O uso de múltiplos robôs para mapeamento e exploração foi apresentado por Simmons et al. (2000).

Um subconjunto da área de sistemas multirrobóticos é o da Robótica de Enxame, que pode ser definido como o estudo do desenvolvimento de um grupo de robôs descentralizado, onde cada robô tem capacidade de desempenhar tarefas individuais que, em conjunto, realiza tarefas de maior complexidade ou a tarefa individual em questão com maior eficácia.

Enxames de Robôs são inspirados em animais sociais, tais como formigas e abelhas que possuem como características marcantes uma robustez maior que a dos demais sistemas multirrobóticos, flexibilidade e escalabilidade (Şahin, 2005). As diferenças fundamentais entre um enxame de robôs e os demais sistemas multirrobóticos são que um enxame de robôs atua possuindo apenas um conhecimento local e de maneira totalmente independente de um sistema central que lhe dê instruções, além de atuar com um número consideravelmente grande de robôs, geralmente em torno de dezenas. Um enxame de robôs é capaz, por exemplo, de se dividir em grupos menores arbitrariamente para realizar uma tarefa e se unirem novamente em um instante futuro, e tais ações ocorrem de maneira natural.

Apesar da grande robustez existente em um sistema composto por vários robôs, não há garantias de que os comandos programados nos robôs estão aptos para guiá-los em qualquer situação, o que é um outro grande problema da Robótica. Por mais elaboradas que sejam as formulações matemáticas de controle para um robô, elas podem não lidar com situações inesperadas. Um enxame de robôs pode até realizar uma tarefa quando seus parâmetros e comandos são inadequados para realizar a tarefa, mas é possível que poucos robôs irão de fato completá-la. Para isso, um mecanismo de aprendizado para que um robô ou um grupo de robôs tenha a autonomia necessária para realizar as tarefas designadas pode ser utilizado. Entende-se por aprendizado

um método pelo qual o robô adquire conhecimento de como melhorar seu desempenho.

Segundo Pfeifer e Scheier (1999), assume-se que o aprendizado de um agente biológico emerge de interações com o ambiente. Ou seja, à medida em que um agente interage com o ambiente, ele adquire informações sobre seu desempenho. Quando, por exemplo, uma criança coloca a mão em um objeto muito quente, ela recebe um *feedback* de seu sistema nervoso, que atua como um sensor, indicando que esta ação é ruim. A partir daquele momento, a criança evitará colocar suas mãos em um objeto muito quente. Um método de aprendizado para robôs que visa o aprendizado desta maneira é apresentado por Sharkey (1998), por meio do qual robôs possuem uma rede neural que indica as velocidades linear e angular de um robô de acordo com as informações sensoriais disponíveis. Os pesos das redes neurais são treinados a partir de um *feedback* de desempenho computado pelo próprio robô.

Aplicações nas quais técnicas de aprendizado de máquina envolvem sistemas multirrobóticos visam obter uma robustez e flexibilidade ainda maiores do enxame e podem tirar proveito das peculiaridades deste conceito. Pugh e Martinoli (2006) apresentaram uma representação do conhecimento adquirido por um robô na forma de pesos de uma rede neural cujo desempenho é avaliado por uma função de aptidão. Os pesos dos neurônios de um robô são atualizados de maneira descentralizada, através de uma otimização por enxame de partículas que considera os pesos dos neurônios associados aos robôs mais próximos e seus respectivos valores de desempenho.

Pugh e Martinoli (2006) e Sharkey (1998) utilizaram uma rede neural cuja estrutura de aprendizado visa uma convergência de parâmetros para um “ponto ótimo”, o que pode resultar em demora para adaptação a cenários novos, sendo um problema relacionado ao dilema estabilidade-plasticidade (Abraham e Robins, 2005). De acordo com este dilema, uma rede neural projetada para ser muito estável tem dificuldade em aprender padrões novos, enquanto o oposto, ou seja, uma arquitetura de aprendizado muito plástica, apresenta dificuldade em manter o que foi aprendido.

Calvo et al. (2010) desenvolveram uma rede neural para robôs baseada na teoria do condicionamento animal. Este sistema controla dois comportamentos conflitantes que o robô conhece inicialmente (desvio de obstáculos e busca a uma meta), ditos inatos, através de uma rede neural

hierárquica. O sistema foi testado tanto em simulação quanto em um robô real realizando a tarefa de exploração de ambientes. Tal sistema inteligente é dotado inicialmente de um conjunto de comportamentos inatos, e consegue aprender através de *feedback* de seu desempenho em relação à sua interação com o ambiente e, assim, gerenciar de maneira mais eficaz a influência destes comportamentos.

Apesar de desenvolvida com a finalidade de ser aplicável para solucionar diversos problemas distintos de aprendizado e de possuir potencial para adaptação a ambientes novos em tempo real, o sistema inteligente desenvolvido por Calvo et al. (2010) não foi experimentado em cenários diferentes do proposto no trabalho original, tendo uma formulação que não aborda cenários de uso distintos.

1.2 Objetivos

O objetivo deste trabalho é controlar um enxame de robôs, de modo descentralizado, adaptando e aperfeiçoando a arquitetura de rede neural hierárquica proposta por Calvo et al. (2010). Os robôs deverão aprender, de maneira autônoma, o quanto cada comportamento deve influenciar em determinadas situações de modo que realizem uma tarefa específica em conjunto. O conhecimento será, inclusive, compartilhado entre robôs vizinhos de modo a acelerar o aprendizado.

A tarefa a ser utilizada para testar tal arquitetura de aprendizado é a de escolta ou aprisionamento de entidades (Antonelli et al., 2008), onde um grupo de robôs deve se posicionar a uma distância preestabelecida de um agente móvel que é considerado a meta de escolta, visando acompanhar este agente para protegê-lo de eventuais ameaças externas. O aprendizado, neste caso, trabalhará com o balanceamento da influência de dois comportamentos: manutenção de distâncias mínimas e cobertura de área. Juntos, estes comportamentos são capazes de realizar a escolta.

Um objetivo secundário deste trabalho é desenvolver um algoritmo de cobertura de área para enxames de robôs simples e capaz de lidar com quaisquer tipo de obstáculo. Para isto, é utilizado o método probabilístico de Lloyd para a obtenção de centros de massa de um Diagrama Centroidal de Voronoi (Ju et al., 2002). Este método é utilizado neste trabalho juntamente com um comportamento de manutenção de distâncias mínimas para a realização da escolta.

A importância deste trabalho, em âmbito geral, dá-se pela tendência de desenvolvimento de enxames de robôs para a realização de tarefas distintas, e pelas vantagens de adaptação às condições do ambiente que um método de aprendizado por reforço em tempo real oferece. Em uma passagem estreita, por exemplo, os robôs teriam a tendência de colidir com maior frequência com obstáculos. Com um método baseado em aprendizado por reforço adequado, as colisões seriam indicativas de uma intensificação do comportamento de desvio de obstáculos. Com isto, é esperado um número menor de colisões e um distanciamento melhor entre os robôs e a meta de escolta.

O uso de um sistema de aprendizado para robôs de um enxame é visado como estratégia para aumentar a eficácia de operação do sistema e para permitir que o mesmo seja capaz de ajustar os pesos que indicam a influência de cada comportamento nas ações do robô em tempo real. O uso da arquitetura de aprendizado adotada justifica-se pelo grande número de iterações que algoritmos de aprendizado por reforço tradicionais exigem, pela simplicidade de obtenção de conhecimento, e para a experimentação de tal estrutura em um sistema multirrobótico, com a possibilidade de compartilhamento de informação. Arquiteturas de aprendizado online na forma de redes neurais crescentes foram, também, alternativas para a solução de aprendizado online em outros trabalhos recentes (Heinen e Engel, 2010) (Silva et al., 2012).

O problema da escolta foi escolhido porque sua eficácia é sensivelmente dependente de um número de robôs razoável, visto que o agente precisa ser coberto pelos robôs, e porque se trata de um problema de navegação de dificuldade considerável porque os robôs não só devem perseguir um agente móvel como evitar colisões entre os robôs e eventuais obstáculos do ambiente.

Justifica-se a elaboração do comportamento de cobertura de área através de Diagramas Centroidais de Voronoi por, principalmente, dois motivos. O primeiro deles é relacionado à necessidade da implementação de tal comportamento para a realização da escolta. O segundo motivo para a criação de um novo método de cobertura de área baseado neste método é a falta de abordagens que realizem esta tarefa de maneira que lidem bem com obstáculos detectados em tempo real, o que é esperado durante a escolta.

No Capítulo 3 será apresentado em detalhes o método de cobertura de área proposto no presente trabalho. Espera-se que este método seja capaz de realizar a cobertura de ambientes desconhecidos de maneira eficaz e que

mantenha uma coesão de posicionamento entre os robôs ao mesmo tempo que evite colisões quando combinada à manutenção de distâncias mínimas para a realização da escolta.

1.3 Organização do Texto

Este texto está organizado do seguinte modo. No Capítulo 2, um levantamento bibliográfico sobre pesquisas relevantes a este projeto é apresentado. No Capítulo 3, os métodos escolhidos para a elaboração deste projeto são explicados em maiores detalhes. No Capítulo 4, o trabalho realizado é descrito, em especial as modificações dos métodos vistos no Capítulo 3. No Capítulo 5, os testes realizados para avaliar este trabalho são explicados e seus resultados discutidos. As conclusões sobre o trabalho realizado são apresentadas no Capítulo 6. Por fim, as referências utilizadas para a elaboração do texto são listadas.

Revisão Bibliográfica

Neste Capítulo, é feito um levantamento bibliográfico sobre os temas abordados neste trabalho, tais como sistemas multirrobóticos, enxames de robôs e técnicas de aprendizado para robôs. Há, também, observações sobre características das pesquisas apontadas neste Capítulo que são interessantes para o desenvolvimento deste trabalho, tais como descentralização e aprendizado.

2.1 Sistemas Multirrobóticos

Em muitas tarefas realizadas por robôs móveis autônomos, a utilização de um único agente robótico pode ser inviável ou até mesmo impossível, emergindo o interesse pelo uso de múltiplos robôs.

Equipes de robôs já foram propostas para realizar diversas tarefas. Jennings et al. (1997) utilizaram uma equipe de robôs para resgates em desastres. Os robôs buscaram por vítimas de maneira paralela, e, ao localizar uma delas, os robôs procuraram “empurrá-la” para um local seguro. A busca pela vítima foi feita através de navegação aleatória no ambiente. Este trabalho especificou que a grande dificuldade enfrentada foi garantir que os robôs sejam capazes de colaborar através de comunicação para realizar a tarefa, além da garantia de que esta comunicação seja utilizada de maneira que o desempenho do enxame melhore.

Uma tarefa clássica de sistemas multirrobóticos é a de exploração de

ambientes desconhecidos com a finalidade de gerar um mapa. Utilizando uma equipe de robôs, o problema se trata de definir direções para cada robô de modo que a cobertura de área seja feita rapidamente. O problema foi tratado em Burgard et al. (2000) a partir de funções de utilidade para cada ponto do próprio mapa métrico que está sendo criado em tempo real. Quando um robô se direciona para um ponto em particular para explorar sua respectiva região, estima-se o quanto cada ponto da região será mapeado quando o robô alcançar o local, e uma função de utilidade é aplicada. Quanto maior for o número de robôs que estime mapear um local em particular, menor será o valor da função de utilidade.

Uma solução diferente foi apresentada por Wurm et al. (2008) dividindo o ambiente em segmentos que seriam correspondentes a salas ao invés de pontos de um mapa. O uso de um particionamento automático do ambiente permitiu um planejamento mais rápido para os robôs.

Batista et al. (2012) abordaram este problema a partir de Campos Potenciais Harmônicos, que utilizam a estrutura do mapa para computar potenciais de navegação. Locais não explorados exerciam forças atrativas, enquanto obstáculos aplicavam forças repulsivas. O método utiliza estes potenciais não somente para guiar os robôs como para evitar colisões entre os mesmos. Mesmo sem um método que force um robô a ir para um ponto específico do mapa, eles fluíram naturalmente para porções não exploradas distintas do ambiente.

Outra aplicação de sistemas multirrobóticos consiste em tarefas de escolta ou aprisionamento de entidades, em que um grupo de robôs deve cercar uma entidade para protegê-la de ameaças externas. O uso de escolta foi indicado como um mecanismo explícito de uma equipe de robôs de resgate em Hsu e Liu (2005), o que reforçou o potencial de aplicações de um método de escolta e serviu como estímulo para a escolha da escolta para este trabalho. Kamano et al. (2000) utilizaram mecanismos de escolta para cercar um agente e capturá-lo. Um conjunto de regras nebulosas foi aplicado a partir de informações de posicionamento dos robôs e da meta para controlar as velocidades lineares e angulares do robô. O desempenho de captura é avaliado e os parâmetros das regras nebulosas ajustados por um algoritmo genético.

Antonelli et al. (2008) abordaram este problema através do acionamento de quatro comportamentos básicos distintos: movimentação do robô em direção à meta, movimentar o robô de forma a circular a meta, distribuir o

posicionamento entre o robô e seus dois vizinhos mais próximos e desviar de obstáculos, que são os próprios robôs e a meta de escolta. Tais comportamentos são gerenciados por um método cinemático de controle chamado de Comportamental Nulo baseado em Espaço (*null space-based behavioral*, NSB) que balanceia a influência de comportamentos de modo a respeitar as restrições de controle do robô. A abordagem tem como prioridade forçar um posicionamento circular bastante rígido e igualmente espaçado.

Mas et al. (2009) resolveram o problema da escolta por um mecanismo centralizado de espaço de *cluster*, em que o grupo de robôs é considerado uma única entidade e a movimentação dos robôs é regida através de uma função de atributos do *cluster*, tais como a posição e a velocidade dos robôs. Sugeriu-se no trabalho que um controle de múltiplos robôs através deste mecanismo facilita a operação da equipe de robôs por um operador humano, visto que o cluster é visto como uma única entidade.

Os mecanismos de escolta descritos acima, entretanto, não consideram obstáculos além dos próprios robôs. Nos testes realizados nos mesmos, foi utilizado um ambiente vazio. Um dos poucos trabalhos de sistemas multirrobóticos que considerou adaptar a escolta de um agente em ambientes de maior complexidade é o de Penders et al. (2011), que está descrito na seção 2.1.1, mas sem mostrar resultados relacionados a tal cenário.

Uma aplicação de múltiplos robôs interessante é a vigilância de ambientes, problema onde um grupo de robôs deve circular em um local de modo a verificar a entrada de intrusos. Lee et al. (2010) gerenciaram um time de robôs para realizar a vigilância através de regras nebulosas. Eventos relevantes como a detecção de um intruso, o desaparecimento de um objeto ou a perda de comunicação com algum robô que estava em sua proximidade foram combinados com informações internas do robô, como a quantidade de bateria restante. Uma tabela de regras linguísticas foi acionada para indicar a força de cada comportamento e isto decidiu a ação a ser realizada pelo robô. As ações do robô variavam de perseguições a intrusos até um estado de vigilância no qual o robô não se movimentava, utilizando a energia restante para sensoriamento.

(Chao et al., 2006) utilizaram uma equipe de robôs equipados com dispersores para realizar um controle de difusão de poluição. Para realizar uma boa distribuição dos robôs, foi utilizado o conceito de Diagramas Centoridais de Voronoi (Ju et al., 2002): diagramas de Voronoi cujo ponto gerador é o centro de massa do polígono no qual está inserido. Um

posicionamento através desta técnica permitiu uma dispersão dos robôs que respeita a concentração de substâncias poluidoras no ambiente porque o método considerou a densidade de poluição para o cálculo do centro de massa do polígono de Voronoi. Para alcançar tal posicionamento, foi utilizado o algoritmo de Lloyd (1982).

A tarefa de dispersão foi de certa forma adotada no trabalho de Pessin et al. (2010), que utilizaram otimização por enxames de partículas para treinar o posicionamento de uma equipe de veículos autônomos de modo a realizar uma dispersão de focos de incêndio mais eficaz. As partículas a serem treinadas foram as poses (posicionamento linear e angular) finais dos robôs, e foi simulado o desempenho do controle de incêndio para a obtenção do desempenho dos robôs.

2.1.1 Enxames de Robôs

Dentro da definição geral de sistemas multirrobóticos, existe o conceito de enxames de robôs. Segundo Şahin (2005), tais sistemas são constituídos de um grande número de robôs que consegue realizar tarefas de maior complexidade através da interação entre os agentes e dos agentes com o ambiente. São também dotados de três propriedades: robustez, flexibilidade e escalabilidade.

A robustez de um sistema é dada pela tolerância tanto a falhas quanto a modificações imprevistas do ambiente. Enxames de robôs possuem esta propriedade por que a perda de alguns indivíduos não é uma falha crítica do sistema, já que os demais podem cobrir esta falta, mesmo que com uma queda de desempenho. Além disso, o uso de um sistema que consista de um único robô demanda comportamentos de complexidade muito maior para realizar a mesma tarefa que um enxame, cujos comportamentos simples são menos propensos a falhas.

Flexibilidade é uma característica obtida por enxames devido à capacidade de responder adequadamente a cenários diferentes. No caso de robôs pertencentes a um enxame, isto equivale a ser capaz de realizar tarefas diferentes dependendo do que seus sensores observam ou ao menos ser capaz de reagir a situações distintas de acordo com a necessidade.

A escalabilidade advém da não existência ou obrigatoriedade de um controlador central. Enxames de robôs são compostos de agentes robóticos que, de modo geral, somente precisam da comunicação com robôs dentro de sua área de comunicação. Esta comunicação pode ocorrer de maneira direta

(rádio) ou de maneira indireta (luz, feromônio).

Um trabalho clássico no que se refere a controle descentralizado de vários agentes, apesar de não ter surgido especificamente para a Robótica de Enxame, é o de Reynolds (1987). Este trabalho surgiu visando simular de maneira mais natural a movimentação de um bando de aves. Cada ave é controlada por um mecanismo que combina três comportamentos distintos: desvio de obstáculos, equalização da velocidade linear à das aves vizinhas e tentativa de proximidade às demais aves. O movimento obtido mostrou-se bastante fluido e natural, mostrando movimentação, desvio de obstáculos e coesão do bando.

Chaimowicz et al. (2005) trabalharam com a formação de padrões de um enxame de robôs a partir de uma abordagem focada na escalabilidade. Uma função implícita foi especificada e seguida pelos robôs através de uma técnica de gradiente descendente. As funções foram geradas através de interpolação de um conjunto de pontos que restringem a função ao formato desejado no formato de funções de base radial. O método foi capaz de gerar formações complexas, como letras do alfabeto.

O enxame de robôs do projeto GUARDIANS (Penders et al., 2011) trabalhou especificamente com a tarefa de resgate, com foco na assistência a bombeiros em operações que ocorrem em grandes galpões. Os robôs serviam como guias, escoltando o bombeiro e realizando localização e mapeamento simultâneos. A localização unida ao mapeamento ofereceu ao sistema a capacidade de informar a posição do bombeiro, visto que os robôs estavam sempre o acompanhando. Os robôs também podiam adotar um comportamento de cobertura a partir da posição do humano que acompanham, criando uma rede de comunicação que foi capaz de alcançar regiões mais distantes do ambiente.

Krishnanand et al. (2006) utilizaram uma técnica de otimização baseada em vagalumes para direcionar um enxame de robôs com o objetivo de controlar fontes emissoras de poluição. O método foi escolhido por possuir um bom desempenho na busca de múltiplos ótimos em funções multimodais, o que facilitou a divisão dos robôs em subgrupos. Os robôs se comunicaram de maneira indireta através de emissão de luz: quanto maior a quantidade de radiação encontrada, mais intensa foi a luz emitida pelos robôs. Cada robô escolheu probabilisticamente um dos demais robôs do enxame que estivesse dentro de sua área de comunicação e seguia em sua direção. Por não ser escolhido necessariamente o robô com maior emissão de luz, os robôs tendiam

a se dividir em grupos, permitindo a busca por diferentes fontes de radiação. Neste trabalho, é interessante que haja um mecanismo que force uma coesão do enxame, ou seja, que evite que o enxame se divida em vários subgrupos desnecessariamente, mas é importante que esta divisão aconteça em algumas situações de modo a evitar colisões desnecessárias e para realizar um melhor acompanhamento durante a escolta.

Calvo et al. (2011) aplicaram o conceito de otimização por colônias de formigas para realizar uma tarefa de vigilância de ambientes. À medida em que os robôs navegaram, eles liberaram um “feromônio” virtual no ambiente cuja intensidade diminui à medida que o tempo passa. Este feromônio serviu como um indicador do quão recente foi a última visita de um robô em um lugar em particular. Cada robô é capaz de detectar seu próprio rastro de feromônio e o dos demais robôs. A navegação foi gerenciada pelo gradiente do feromônio, de modo que os robôs tendiam a se direcionar a locais onde a concentração de feromônio era baixa.

Um outro trabalho que lidou com vigilância, mas com o uso de veículos aéreos não tripulados (VANTs) para monitoramento ambiental, foi apresentado por Varela et al. (2011). A ação dos VANTs foi dividida em etapas: primeiro, eles se espalharam no ambiente; a seguir, eles buscaram manter o espalhamento ao mesmo tempo em que utilizaram seus sensores para identificar sinais de poluição; e, enfim, quando identificava um sinal, ele comunicava a descoberta aos robôs dentro de seu alcance, que o auxiliaram a detectar o foco de poluição. Não foi especificado, entretanto, o método de espalhamento dos VANTs. O trabalho descrito a seguir é uma possível solução.

Em Tan et al. (2004), foi utilizado o mecanismo do algoritmo de Lloyd para realizar a tarefa da cobertura de área através de Diagramas Centroidais de Voronoi. O robô se comunicava com todos os seus vizinhos, e com esta informação construiu um diagrama de Voronoi local, que foi utilizado para a aplicação do algoritmo de Lloyd. Este trabalho, apesar de não referenciar explicitamente mecanismos de enxames de robôs, mostrou muitas características desejáveis em um enxame. A mais importante foi, talvez, a divisão e reunião dos robôs do enxame à medida em que o ambiente oferece dificuldades. Isto ocorreu porque cada robô computa em tempo real seu polígono de Voronoi, considerando somente os dados de seus vizinhos mais próximos. Ou seja, o processamento de navegação foi local. Esta capacidade foi crucial para o uso de Diagramas Centroidais de Voronoi para o trabalho.

Outra característica interessante é que esta estrutura possui potencial para gerenciar não somente o desvio de obstáculos mas também a coesão do enxame.

Rounds e Chen (2009) utilizaram o método probabilístico de Lloyd para o controle de enxames de robôs de comportamento fototrópico, ou seja, que buscam uma fonte luminosa. Para evitar conflitos provenientes de outras fontes de luz, foi utilizada uma distribuição Gaussiana a partir das fontes de luz para a geração de amostras aleatórias. Este método probabilístico descarta a necessidade de construção implícita de diagramas de Voronoi para a localização dos centroides. O método probabilístico de Lloyd, pela sua importância neste trabalho, é explicado em detalhes no Capítulo 3.

2.1.2 Gerenciamento de *software* dos robôs

Desenvolver um sistema multirrobótico para desempenhar uma tarefa envolve enfrentar um grande conjunto de obstáculos. Uma das dificuldades encontradas para o desenvolvimento de um sistema multirrobótico está em gerenciar todos os componentes de *software* aos sensores e atuadores do robô. Por isso, é interessante verificar um sistema adequado para realizar esta tarefa, facilitando o desenvolvimento do trabalho. Um dos trabalhos pioneiros neste assunto foi o sistema ACTRESS (Asama et al., 1989), desenvolvido para gerenciar o reconhecimento do ambiente, as ações do robô, a identificação de tarefas e a comunicação entre os agentes. A maneira na qual este sistema foi desenvolvido visa a síntese de vários elementos, incluindo não só robôs como qualquer entidade capaz de tomar decisões e de estabelecer comunicação, como sensores e simuladores, para tornar tal sistema o mais generalizado possível.

Bruyninckx (2001) desenvolveu o sistema OROCOS com a mesma finalidade: nele, objetiva-se gerenciar um sistema robótico através de um *software* de código aberto que utiliza princípios de Engenharia de Software em sua arquitetura para melhor manutenção e atualização.

Uma solução bastante popular para o problema de comunicação com os sensores e atuadores do robô é a interface Player (Gerkey et al., 2003), que consiste em um conjunto de *drivers* capazes de se comunicar com várias arquiteturas robóticas diferentes. Outro motivo pelo qual ele é bastante utilizado é a sua fácil integração com o simulador de robôs Stage. Esta foi a estrutura adotada para este trabalho devido à facilidade de uso, fácil integração com um simulador, oferecimento de todos os recursos necessários

para o desenvolvimento da pesquisa e pela experiência do aluno com estas ferramentas.

2.2 Aprendizado em Robôs Móveis em tempo real

O ajuste adequado dos parâmetros que definem propriedades de tomada de decisão e de navegação do robô é crucial para a obtenção de um bom desempenho. Este desempenho, entretanto, depende do ambiente apresentado ao robô: um ajuste realizado em um ambiente controlado pode não ser eficaz quando o robô é posto para realizar a tarefa desejada em um ambiente real. Uma alternativa para resolver este problema é através de técnicas de aprendizado por reforço, pelas quais robôs podem aprender através da interação com o ambiente, ou seja, a partir de diversos acertos e erros que o robô detecta. Em todas estas técnicas, o robô possui um meio de avaliar seu desempenho.

Um dos primeiros trabalhos que utilizaram este conceito foi o de Mataric (1997). Um método de aprendizado por reforço baseado em diferença temporal foi aplicado para treinar o momento no qual cada comportamento foi realizado. Para a realização deste aprendizado, entretanto, foi necessário estabelecer associações de estados do ambiente e de ações, necessários para técnicas de aprendizado por reforço. Considerando que robôs não são capazes de identificar todas as informações do local onde está e que a grande maioria dos dados captados foram valores contínuos, o que impossibilitaria o uso de dados brutos para um aprendizado por reforço, foram utilizadas discretizações binárias das informações obtidas pelos sensores disponíveis, descritas na forma de perguntas como "o robô está na sala?" ou "o robô está próximo de colidir?". Para representar as ações, ao invés do uso da força exercida pelos mecanismos de navegação do robô (rodas, esteiras, pernas), foram utilizados comportamentos modulares como "avançar", "virar à direita" ou "parar".

Este formato de aprendizado, entretanto, possui uma limitação no seu potencial de aprendizado: não seria possível treinar o quanto um robô teria que virar à esquerda, por exemplo. Uma alternativa para a solução deste problema é apresentada por Sharkey (1998), em que o conhecimento do robô é representado por uma rede neural. Isto permite um aprendizado a partir das informações brutas dos sensores disponíveis, ou seja, sem a necessidade de discretização. O objetivo da rede neural no sistema é indicar o quanto

cada um dos comportamentos implementados previamente (conhecidos como inatos) deverá influenciar no comportamento do robô. As saídas do sistema são as velocidades linear e angular das rodas do robô. A Figura 2.1 ilustra esta estrutura neural.

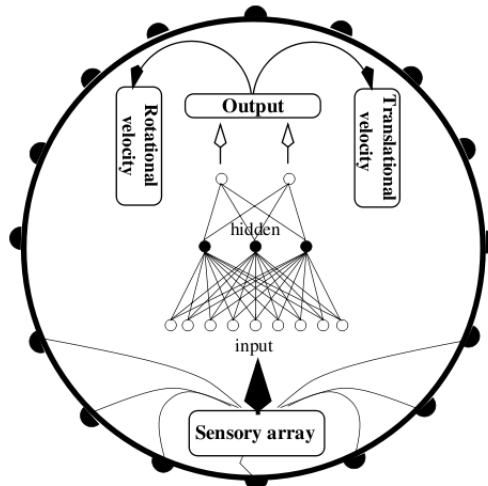


Figura 2.1: Representação das entradas (leituras de sensores) e saídas (velocidades) da rede neural. Retirada de Sharkey (1998).

Uma dificuldade de realizar o treinamento em tempo real de uma rede neural é o desconhecimento da estrutura ideal da rede. Uma estrutura inadequada pode ser incapaz de aprender padrões ou ter dificuldade em generalizar a informação oferecida (*overfitting*). Uma solução é utilizar uma rede neural onde não só os pesos são treinados como também a própria arquitetura. Stanley e Miikkulainen (2002) apresentaram uma estrutura com tal capacidade, chamada de Neuro Evolução de Topologias Incrementais (NEAT, de *NeuroEvolution of Augmenting Topologies*). Trata-se de um processo evolutivo no qual o aprendizado começa de um conjunto de estruturas neurais simples, sem camadas intermediárias. À medida em que o agente interagia com o ambiente, operadores genéticos foram aplicados para alterar a estrutura dos neurônios, adicionando neurônios ou conexões.

A grande vantagem prática de redes que podem reconstruir suas arquiteturas, como a NEAT, em relação a mecanismos neurais para aprendizado em tempo real como o de Sharkey (1998) é que torna-se desnecessário estimar através de experimentos o número de neurônios ideal, o que é um acréscimo à independência de conhecimento inicial. Torna-se necessário, porém, definir parâmetros adequados para gerenciar os acréscimos e remoções de neurônios.

No trabalho de Silva et al. (2012), foi apresentada uma modificação da

rede NEAT que trabalha com o conceito de neuromodulação, que oferece maior controle à plasticidade de uma rede neural. A plasticidade é uma característica que determina a capacidade de absorção de conhecimento de uma rede neural. A grande vantagem deste método está no maior controle sobre o momento no qual uma rede neural decide aprender com uma informação recebida.

Uma alternativa que combina conceitos de aprendizado por reforço em arquiteturas neurais com o objetivo de gerenciar a navegação de um robô foi apresentada por Figueiredo (1997) através de uma rede neural nebulosa hierárquica. A arquitetura foi dividida em um módulo de comportamentos inatos e um módulo de sistemas nebulosos. O módulo de comportamentos inatos possuía dois métodos que realizam comportamentos simples: desvio de obstáculos e avanço até uma meta. Ambos os comportamentos retornavam à rede um direcionamento angular para o robô. Os sistemas nebulosos indicaram o quanto cada comportamento influi no resultado final da rede. A ocorrência de uma colisão ou de alcance de uma meta ativava um mecanismo de aprendizagem no qual as saídas das regras nebulosas eram ajustadas. Esta rede, entretanto, apresentou problemas para lidar com uma situação: a captura de metas muito próximas de um obstáculo.

Para resolver este tipo de problema, um aprimoramento desta rede nomeado ANS (*Autonomous Navigation System*) foi apresentado em Calvo e Figueiredo (2003) e testado em robôs reais em Calvo et al. (2010). Nele, duas modificações principais foram realizadas: além de modificar o método de aprendizado por reforço para uma melhor adequação do balanceamento em cenários difíceis, os módulos de sistemas nebulosos foram substituídos por um conjunto de redes neurais, em que cada rede representa um sensor do robô. Quando um evento relevante ocorre, não somente o aprendizado por reforço é ativado como um neurônio é inserido na camada intermediária da rede neural associada ao evento, trazendo à rede um treinamento da própria arquitetura neural, assim como ocorre em trabalhos como o de Stanley e Miikkulainen (2002) e Sharkey (1998), e com a interessante característica de registro de um neurônio que armazena a informação do sensor. Desta maneira, quando uma situação como a que trazia dificuldades no trabalho de Figueiredo (1997) ocorre, um registro da informação sensorial de distância de obstáculos é memorizado. Tal estruturação da informação bruta e a combinação com o aprendizado com a mesma, além da falta de experimentação de tal método em outros cenários, foi determinante para a

adoção deste modelo neste Mestrado. Esta rede é explicada em detalhes no Capítulo 3.

Zuo et al. (2010) utilizaram o algoritmo Q-learning (Watkins, 1989) para que um grupo de robôs aprendesse autonomamente como manter uma formação, ou seja, um posicionamento estratégico entre os robôs para realizar uma determinada tarefa. Os estímulos de aprendizado dos robôs foram relacionados ao objetivo individual de evitar colisões e ao objetivo coletivo de alcançar uma posição dada como meta, reforçando ou punindo as ações dos robôs a cada iteração através do algoritmo Q-learning. Apesar de projetado para uma aplicação envolvendo vários robôs, o aprendizado foi feito de maneira individual, sem a intervenção do aprendizado dos outros robôs. Na próxima seção, alguns trabalhos que aproveitam esta vantagem são explicados.

2.2.1 Aprendizado em Sistemas Multirrobóticos

Além das vantagens de robustez típicas de um sistema composto por vários robôs móveis, existe a possibilidade de compartilhamento do conhecimento obtido durante a navegação. Isto torna a adaptação dos robôs significativamente mais rápida porque cada robô estará exposto a uma situação distinta. Um robô que ajusta seus parâmetros devido a, por exemplo, uma colisão, é capaz de transferir este conhecimento aos demais robôs, evitando que cada robô tenha que passar por experiências similares.

Uma maneira de realizar este aprendizado é a partir de métodos que, de alguma maneira, representam várias possibilidades de parâmetro de modo distribuído. Hagrais et al. (2002) utilizaram algoritmos genéticos para este fim: cada robô possui um conjunto de parâmetros pessoal (ou genótipo). Após um período de navegação dos robôs no ambiente, uma avaliação baseada no desempenho do robô na tarefa requisitada (ir até uma meta sem colidir com obstáculos) é realizada por cada robô. A seguir, cada agente do grupo compartilha seu genótipo e avaliação de desempenho com outros robôs dentro da área de comunicação. Com os valores obtidos, o robô executa uma iteração de um algoritmo genético para otimizar seus parâmetros. Um mecanismo como este é uma maneira simples de compartilhar a informação obtida, o que pode acelerar bastante o aprendizado dos robôs.

Pugh e Martinoli (2006) utilizaram-se dos mesmos mecanismos de comunicação visando a aplicação em um enxame de robôs, mas aplicaram a técnica de otimização por enxames de partículas para realizar a otimização,

obtendo resultados superiores aos do algoritmo genético e argumentando que este resultado deveu-se à maior eficácia de enxames de partículas para um número de amostras de parâmetros menor em contraste ao algoritmo genético.

Uma comparação entre a otimização de enxame de partículas e variações do algoritmo Q-learning para o aprendizado de múltiplos robôs através da aplicação da tarefa de desvio de obstáculos foi feita por Di Mario et al. (2013). Os testes demonstraram que o uso de uma rede neural para contornar a discretização das ações do Q-learning apresentou significativa melhora. A comparação direta com a otimização de enxames de partículas indica que, para o aprendizado utilizando múltiplos robôs, o desempenho de tal técnica contra o Q-learning produziu resultados melhores. Ainda, segundo os autores, a convergência de aprendizado foi alcançada mais rapidamente pela otimização por enxames de partículas devido à natureza distribuída do método, o que encoraja o uso de compartilhamento de informações. Outro resultado relevante apontado foi que a discretização dos dados sensoriais de distância a obstáculos impactou sensivelmente o desempenho dos robôs em comparação ao uso de dados contínuos.

Diversas variações de parâmetros da otimização baseada em partículas foram testadas para o mesmo problema por Di Mario e Martinoli (2013). Dentre os experimentos realizados, um treinamento inicial em ambiente de simulação foi aplicado em robôs reais, o que acelerou a convergência. Um trabalho futuro apontado foi a descentralização do método, visto que considerou-se uma otimização global dos robôs para o estudo.

Fundamentação Teórica

Neste Capítulo, são descritos os métodos adotados para a elaboração do trabalho. É feita uma descrição da rede neural hierárquica ANS e, em seguida, de Diagramas Centroidais de Voronoi e sua aplicação em enxames de robôs.

3.1 A rede neural hierárquica ANS

A maneira pela qual animais respondem a estímulos externos de modo que lhes permita sobreviver é um tópico de grande interesse não só a etólogos como para pesquisadores de Robótica, pelas suas aplicações. Esse interesse serve de inspiração para o paradigma reativo de robôs inteligentes, que trata de reações dependentes apenas do estímulo imediato do robô, sem que o mesmo possuísse um mapa ou qualquer outra informação armazenada sobre o ambiente (Murphy, 2000).

Um trabalho pioneiro sobre a resposta dos animais a estímulos foi o de Pavlov (1927), que aborda o comportamento de um indivíduo animal quando submetido a estímulos não-condicionados. Seu trabalho concluiu que estímulos não-condicionados, ou seja, que o animal possui de forma inata, são capazes de gerar respostas não-condicionadas. Com o devido treinamento, a resposta não-condicionada de um estímulo pode ser ativada por um segundo estímulo apresentado em conjunto com um primeiro estímulo não-condicionado. Este segundo estímulo é dito condicionado, ou

seja, foi treinado pela interação do animal com o ambiente.

Segundo a Teoria da Seleção de Grupos Neurais (Edelman, 1987), um indivíduo não possui conhecimento de como realizar tarefas complexas previamente. Tal conhecimento seria obtido através da interação com o ambiente, com a exceção de alguns comportamentos ditos inatos desenvolvidos a partir da evolução natural da espécie do indivíduo. Estes comportamentos inatos ditam as regras iniciais pelas quais ele interage com o ambiente, permitindo que o aprendizado de tarefas específicas ocorra de maneira que o indivíduo seja capaz de sobreviver às adversidades do ambiente. À medida em que o indivíduo interage com o ambiente, seus conhecimentos adquiridos unem-se aos conhecimentos inatos para gerar comportamentos complexos que permitem a execução de tarefas específicas.

A partir desta teoria, foi desenvolvido um sistema de aprendizado por reforço para robôs móveis na forma de uma rede neural hierárquica capaz de gerenciar o conhecimento inato e o conhecimento adquirido com o objetivo de controlar um robô até um conjunto de metas, denominado Sistema de Navegação Autônoma (ANS) (Calvo et al., 2010). Esta rede possui módulos de representação de conhecimento inato e uma estrutura responsável pelo conhecimento adquirido por meio da interação com o ambiente. A Figura 3.1 mostra a arquitetura geral do ANS. Os módulos OA (*obstacle avoidance*, ou desvio de obstáculo) e TS (*target seeking*, ou busca ao alvo) são responsáveis por gerar os comportamentos inatos do robô, enquanto que os módulos OD (*obstacle distance*, ou distância do obstáculo), TA (*target distance*, ou distância do alvo) e TD (*target direction*, ou direção do alvo) são redes neurais que fazem parte do Módulo Coordenador (ou *Coordinator Module*, CM), responsável por gerenciar a influência dos comportamentos inatos na ação efetuada pelos robôs.

As entradas destes sistemas são, respectivamente, as informações provenientes dos sensores de distância a obstáculos e de direção ao alvo, ambos acoplados ao robô. A informação fornecida pelo último equivale ao ângulo relativo ao robô que esteja mais alinhado à posição da meta (ou alvo).

Os módulos inatos são responsáveis pelo envio da informação de como o robô deve se comportar para realizar uma ação inata, seja ela desviar de obstáculos (OA) ou buscar um alvo (TS). No trabalho original, elas são representadas por sistemas nebulosos do tipo Mamdani (Mamdani e Assilian, 1975) cujas regras indicam o deslocamento angular que o robô deve exercer para realizar o comportamento.

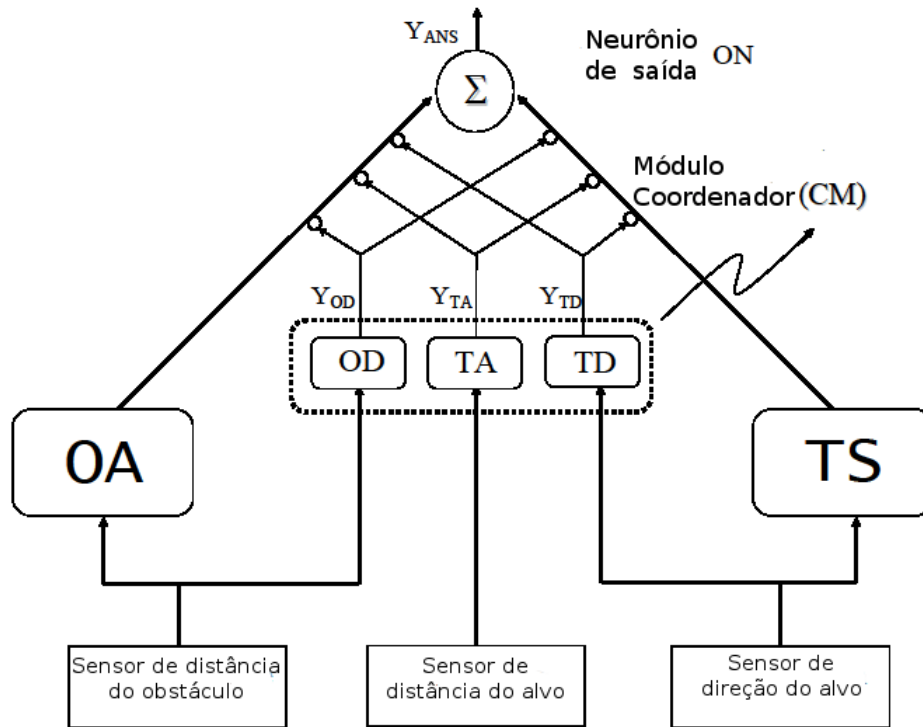


Figura 3.1: Visão geral da rede neural hierárquica ANS. Retirada de Calvo et al. (2010).

O módulo CM é composto por redes neurais. Estas redes foram modeladas utilizando conceitos de sistemas nebulosos para a elaboração de funções de similaridade e para o cômputo da saída da rede. Inicialmente, não existem neurônios na camada intermediária. Para cada evento específico ocorrente, um neurônio é acrescentado nesta camada. Para o comportamento de desvio de obstáculos, eventos de colisões ocasionam o acréscimo de neurônios na rede correspondente ao módulo OD, enquanto a captura de metas resulta no acréscimo de neurônios nas redes dos módulos TA e TD quando o comportamento de busca ao alvo é considerado. Considerando N_n o número total de neurônios em uma rede neural do CM, cada entrada de cada neurônio $i = 1..N_n$ da camada intermediária acrescentado possui peso sináptico w_i correspondente às informações obtidas pelos sensores correspondentes no momento em que o evento (colisão ou captura) ocorre e um peso de similaridade s_i que indica a relevância de w_i para a saída de uma sinapse. Os conjuntos de peso de w_i e s_i são referidos como W e S . Pode-se considerar, visto o contexto de criação do ANS como um algoritmo de aprendizado por reforço, que os neurônios são os representantes dos estados do robô.

O número de entradas de cada neurônio da primeira camada é igual ao

número de sensores relacionados à rede. Para a rede do módulo OD é utilizada a informação de distância a obstáculos, enquanto a rede TA recebe a distância à meta e a rede TD recebe a diferença de direcionamento angular. Na saída desta camada, os pesos w_i são utilizados para a construção das funções $\Phi_{w,i}$, enquanto as entradas são captadas para a geração de $\Omega_{x,i}$, ambas de formato gaussiano e dispersão constante definida pelo parâmetro σ . A saída de cada neurônio da primeira camada é proporcional à similaridade entre a informação armazenada nos pesos sinápticos e as informações provenientes dos sensores. A medida de similaridade, denotada por $\Delta(\cdot)$, corresponde ao maior ponto de intersecção entre $\Omega_{x,i}$, função gaussiana representante de um valor de entrada do neurônio, e de $\Phi_{w,i}$, representante do peso presente no neurônio. Quanto maior for o valor de σ , maior será a dispersão, e portanto menor será a variação do valor de similaridade em relação à diferença de valores. A Figura 3.2 ilustra a obtenção do valor de inferência $\Delta(\Phi_{w,i}, \Omega_{x,i})$. A Equação 3.1 resume a operação de comparação entre uma entrada e o peso correspondente, onde σ é um parâmetro de dispersão da função gaussiana. A similaridade pode ser obtida pela Equação 3.2.

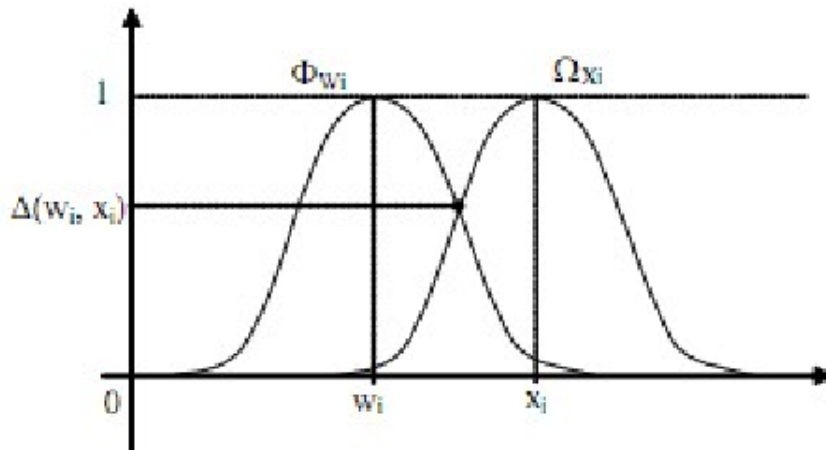


Figura 3.2: Visualização dos conjuntos Ω_{x_i} e Φ_{w_i} e da extração do valor $\Delta(\Phi_{w_i}, \Omega_{x_i})$. Retirada de (Calvo et al., 2010).

$$\Delta(\Phi_{w,i}, \Omega_{x,i}) = e^{-\frac{(x_i - w_i)^2}{2\sigma^2}} \quad (3.1)$$

$$\Delta(\Phi_w, \Omega_x) = \max_{i=1}^{N_n} \Delta(\Phi_{w_i}, \Omega_{x_i}) \quad (3.2)$$

A partir do segundo neurônio inserido na primeira camada, os pesos

sinápticos dos neurônios já armazenados podem ser atualizados. Quando um evento que não seja o primeiro relacionado à rede em questão ocorre e um limiar de similaridade Lim_{SIMIL} é alcançado entre o neurônio mais similar e o candidato à inserção, a informação de cada sensor correspondente é comparada com todos os pesos w_i de todos os neurônios da primeira camada. O neurônio que possuir maior similaridade tem seus pesos ajustados de maneira que se aproxime do valor enviado pelo sensor de acordo com a Equação 3.3. O peso s do neurônio i é atualizado de acordo com a Equação 3.4, onde α é uma taxa de controle de ajuste de neurônio: quanto maior for α , menor será o ajuste. N_{ev} é o número de eventos associados a esta rede que ocorreram.

$$w_{i,j}^{t+1} = w_{i,j}^t + (x_i - w_{i,j}^t)/(1 + s_{i,j}^t) \quad (3.3)$$

$$s_{i,j}^{t+1} = s_{i,j}^t + \frac{(\Delta\Phi_{w_i}, \Omega_{x_i} - s_{i,j}^t)}{(1 + \alpha)}, \alpha = N_{ev} - i + 1 \quad (3.4)$$

A taxa α está diretamente relacionada com o número de eventos e o número de neurônios existentes. Os pesos S são, portanto, ajustados gradualmente para que se aproximem do valor de similaridade. Assim, à medida em que eventos nos quais neurônios são ajustados ocorrem, as modificações em S , em particular em neurônios mais velhos, serão menores. Este ajuste reflete no ajuste de W : quanto maior o peso de similaridade, menor será a aproximação deste peso.

A segunda camada das redes do CM possui apenas um neurônio, que computa todas as entradas obtidas através do valor máximo obtido por todos os resultados de inferência obtidos pelos neurônios representativos dos pesos. (Equação 3.5). O valor y representa a saída de uma rede do CM.

$$y = \max_{i=1}^{N_n} (\Delta(\Phi_{w_i}, \Omega_{x_i}) * s_i) \quad (3.5)$$

A saída do ANS é gerada por um único neurônio. Nele, os resultados obtidos pelos módulos inatos e pelo CM são agregados de modo a tomar a decisão final. Este neurônio é, portanto, responsável por gerar a ação a partir da relevância dos estados identificados no CM. No caso do Sistema de Navegação Autônomo, essa saída, Y , é um ajuste angular do robô dado pela Equação 3.6. U_{OA} e U_{TS} representam os ajustes angulares enviados pelos comportamentos inatos, ϵ , $\epsilon \ll 1$ é uma constante positiva, w são pesos estimuladores de uma saída de uma rede neural do CM (OD , TD e TA), \bar{w} são

pesos inibitórios, e y são as saídas das redes neurais do CM. Os pesos \bar{w} realizam o serviço de diminuição de estímulo dos comportamentos, acelerando mudanças na influência dos comportamentos quando necessário.

$$Y = \frac{w_{OA}U_{OA} + w_{TS}U_{TS}}{w_{OA} + w_{TS}}$$

$$w_{OA} = \epsilon + w_{od}y_{od} - (\bar{w}_{ta}y_{ta})(\bar{w}_{td}y_{td}) \quad (3.6)$$

$$w_{TS} = \epsilon - \bar{w}_{od}y_{od} + (w_{ta}y_{ta})(w_{td}y_{td})$$

A cada execução do sistema, os pesos de influência de cada rede neural do CM são atualizados conforme a Equação 3.7, onde ζ e η são taxas de aprendizado situadas no intervalo $[0..1]$, e $f(\cdot)$ e $g(\cdot)$ são funções de ativação sigmóides. As constantes ζ e η regulam a intensidade de ajuste dos pesos w e \bar{w} : quanto maiores forem estes valores, menor será a influência. O ajuste também depende, segundo a fórmula, das saídas do CM e da intensidade dos comportamentos inatos.

$$\begin{aligned} w_{od}^{t+1} &= w_{od}^t + \zeta(1 - w_{od}^t)f(y_{od})g(U_{OA}) \\ \bar{w}_{od}^{t+1} &= \bar{w}_{od}^t + \zeta(1 - \bar{w}_{od}^t)f(y_{od})g(U_{TS}) \\ w_{ta}^{t+1} &= w_{ta}^t + \eta(1 - w_{ta}^t)f(y_{ta})g(U_{TS}) \\ \bar{w}_{ta}^{t+1} &= \bar{w}_{ta}^t + \eta(1 - \bar{w}_{ta}^t)f(y_{ta})g(U_{OA}) \\ w_{td}^{t+1} &= w_{td}^t + \eta(1 - w_{td}^t)f(y_{td})g(U_{TS}) \\ \bar{w}_{td}^{t+1} &= \bar{w}_{td}^t + \eta(1 - \bar{w}_{td}^t)f(y_{td})g(U_{OA}) \end{aligned} \quad (3.7)$$

A função de w e de \bar{w} é suavizar a influência das saídas do CM. Esta suavização mostra-se interessante para a aplicação em problemas de controle de robôs, evitando problemas com ruído.

3.2 Diagramas Centroidais de Voronoi

Uma estrutura matemática bastante conhecida é o diagrama de Voronoi, constituído de polígonos e construído a partir de amostras ditas geradoras em um plano n -dimensional. Estes polígonos são construídos de modo que qualquer ponto no interior dos mesmos respeite a seguinte condição: dentro de uma região V_i de ponto gerador x_i , correspondente a um polígono $i = 1..n$ e pertencente a um domínio Ω dado que $\cup V_i = \Omega$, todo ponto x incluso em V respeita a condição imposta pela Equação 3.8.

$$V_i = \{x \in \Omega \mid \|x - x_i\| < \|x - x_j\| \text{ para } j = 1, \dots, n \text{ e } j \neq i\} \quad (3.8)$$

Existe um caso especial de diagramas de Voronoi nos quais as amostras geradoras são, também, os centros de massa (ou centroides) dos polígonos. Neste caso, temos um Diagrama Centroidal de Voronoi (DCV). Tais estruturas possuem uma propriedade especial de otimização da Equação 3.9, onde \tilde{x} é tanto o ponto gerador quanto o centro de massa. Tal equação pode ser utilizada para a formulação de problemas de otimização envolvendo conceitos como custo, variância e erro de quantização (Du et al., 2010). A equação aborda, inclusive, o uso de diferentes densidades (representadas como $\rho(x)$). Tais densidades podem ser levadas em consideração na otimização via DCVs através do uso no cálculo do centro de massa de cada polígono. A Figura 3.3 mostra a diferença entre um diagrama de Voronoi típico e um DCV.

$$K(\tilde{X}, \tilde{V}) = \sum_{i=1}^n \int_{\tilde{V}_i} \rho(x) \|x - \tilde{x}_i\|^2 dx \quad (3.9)$$

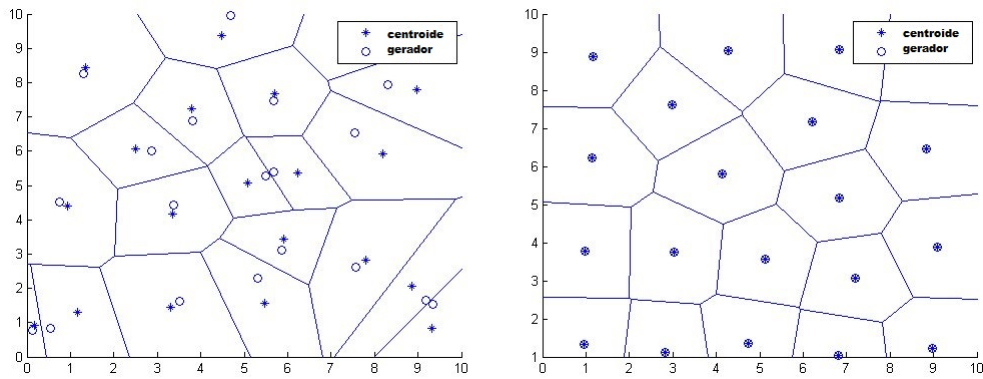


Figura 3.3: Esquerda: diagrama de Voronoi. Direita: DCV. Para o cômputo do DCV, foi utilizado o método de Lloyd limitando Ω para o fechamento de polígonos infinitos.

Um algoritmo para a obtenção de DCVs é o método de Lloyd (1982). Ele consiste em gerações sucessivas de diagramas de Voronoi. Inicialmente, os polígonos são obtidos a partir de amostras geradoras aleatórias. Após a primeira iteração, os polígonos seguintes são gerados a partir dos centros de massa dos obtidos anteriormente. Este processo é capaz de convergir o diagrama para um DCV, e pode ser interrompido quando um critério arbitrário de convergência for alcançado.

O custo para a construção do diagrama de Voronoi, entretanto, pode vir a ser caro, ou o fechamento do polígono pode-se mostrar um problema significativo. Certas aplicações sequer buscam o diagrama em si, mas os valores \tilde{x}_i associados. Ju et al. (2002) propuseram um método híbrido, usando o algoritmo de Lloyd e um algoritmo baseado em várias amostragens aleatórias conhecido como método de MacQueen (MacQueen, 1967), cujo funcionamento ocorre da seguinte maneira. É produzido um conjunto de amostras aleatórias em Ω a partir de uma $\rho(x)$ específica. Em seguida, verifica-se qual é a amostra geradora mais próxima de cada uma destas amostras. O centroide dos pontos relacionados a um gerador representa o local para onde o gerador deverá se direcionar, e é uma estimativa do centro de massa de um diagrama de Voronoi construído a partir das amostras geradoras. Este método pode, também, ser utilizado de maneira paralela, com cada amostra geradora produzindo amostras aleatórias e obtendo quais amostras estão mais próximas de si que das demais amostras geradoras. Tal método é conhecido também como método probabilístico de Lloyd.

3.2.1 Diagramas Centroidais de Voronoi em Sistemas Multirrobo

A propriedade de otimização da Equação 3.9 através de DCVs já foi explorada em problemas de robótica móvel, normalmente utilizando as posições dos robôs como amostras geradoras. Uma aplicação frequente de DCVs encontrada na literatura é na solução do problema da cobertura de área. Nele, um grupo de robôs dotados de sensores deve se posicionar em um ambiente de modo que seus sensores sejam capazes de cobrir o máximo possível de um ambiente visando manter redes de comunicação, vigilância de ambientes, exploração de regiões desconhecidas, entre outras aplicações.

Tan et al. (2004) utilizaram o algoritmo de Lloyd para coordenar o espalhamento de robôs em um ambiente sem obstáculos. A posição de cada robô é considerada uma amostra geradora, e o centro de massa é calculado delimitando os polígonos a partir do limite de sensoriamento dos robôs. A Figura 3.4 mostra a configuração inicial dos robôs e suas posições quando o espalhamento é estabilizado.

Breitenmoser et al. (2010) trataram a cobertura de área em um ambiente conhecido mas não-convexo, em que a simples aplicação do método de Lloyd apresenta dificuldades de otimização. Este problema foi sanado neste trabalho combinando o método de Lloyd com planejadores de caminhos.

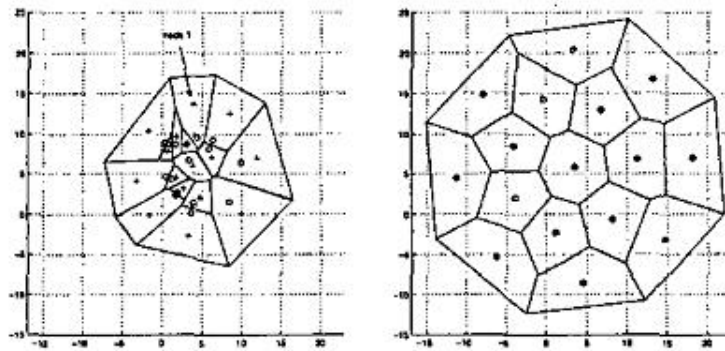


Figura 3.4: Esquerda: configuração inicial dos robôs. Direita: Configuração estável dos robôs. Retirada de (Tan et al., 2004).

Alguns trabalhos aproveitam o conceito de densidade para uma melhor solução do problema. Mishra et al. (2012) propuseram o uso de funções de densidade específicas para um menor gasto de energia de comunicação entre robôs durante a cobertura de área, além de tratar obstáculos previamente conhecidos através da soma de forças repulsivas.

3.3 Considerações Finais

Neste Capítulo, foram apresentados os conceitos gerais utilizados para a elaboração do trabalho. No Capítulo seguinte, será mostrado como e quais dos trabalhos citados foram efetivamente assimilados ao trabalho, e como serão unidos os tópicos sobre aprendizado em robôs, DCVs e robótica de enxame.

Descrição da Pesquisa

Neste capítulo, serão apresentadas as adaptações realizadas nos métodos vistos previamente para o desenvolvimento do sistema constituído de múltiplos robôs para a tarefa de escolta. Além disso, será descrito como a rede neural hierárquica foi integrada com o método de cobertura de área.

4.1 Adaptações e implementação da rede AIS

Durante o desenvolvimento deste trabalho, foram identificadas algumas deficiências do ANS em seu uso em problemas de âmbito geral. Durante a adaptação da mesma para uso na tarefa de escolta, alguns aspectos desta rede foram modificados. Tais modificações são explicadas ao longo desta Seção.

A rede ANS, apesar de projetada com fins de uso em vários problemas de aprendizado em tempo real, não possui fórmulas adaptadas para um número variável de redes neurais no CM e de comportamentos distintos, tendo as mesmas sido elaboradas especificamente para o uso para navegação de um único robô. Para possibilitar um uso mais amplo desta arquitetura, foram feitas algumas generalizações nas fórmulas de obtenção da saída Y e na atualização dos pesos das redes neurais do CM, além de acrescentada uma política de eliminação de neurônios. Visto que o ANS estará modificada de forma significativa, a rede hierárquica decorrente dessas modificações será chamada de *Autonomous Intelligent System (AIS)* em contraste à denominação específica ao problema de navegação da rede original. A arquitetura da rede

AIS é idêntica à da rede ANS, composta de módulos inatos, de um Módulo Coordenador e de um Módulo de Saída.

A implementação da rede neural AIS foi feita a partir da elaboração de três classes na linguagem de programação C++:

- Neurônio: operações de criação, ajuste de valores e comparação de neurônios pertencentes a uma rede do Módulo Coordenador;
- Rede do Módulo Coordenador: associada a um comportamento e a um conjunto de dados sensoriais, gerencia o ajuste de seus neurônios e computa sua saída $y_{i,j}$.
- AIS: a rede como um todo. Possui um módulo coordenador e combina os resultados deste módulo com os comportamentos inatos.

A implementação da rede AIS foi realizada de modo que seu código seja facilmente utilizado para outros problemas fora do proposto neste trabalho, visando seu uso em outros trabalhos que demandem aprendizado em tempo real de um agente inteligente.

4.2 Implementação do neurônio e da Rede do Módulo Coordenador

Para codificar a rede AIS, é necessário construir duas estruturas essenciais: o neurônio e a rede do Módulo Coordenador. A formulação original da AIS para estas duas estruturas foi feita de modo que a sua aplicação para problemas de natureza diversa fosse possível. Dessa maneira, as equações referentes a estas estruturas não foram alteradas.

A implementação da classe do Neurônio foi feita de maneira simples, armazenando dois vetores que representam, respectivamente, os conjuntos de pesos W e S . Cada neurônio é inicializado com seus pesos S com um valor inicial fixo (0.5, como sugerido no trabalho original), e com o número de pesos que o neurônio irá armazenar. As operações desta classe são de atualização e obtenção de pesos.

Cada rede do Módulo Coordenador é dotada de um *vector*, pertencente à biblioteca padrão da linguagem C++ *Standard Template Library* (STL). Tal estrutura de dados atua como um vetor de alocação dinâmica. Este vetor é responsável pelo armazenamento de neurônios inseridos previamente. Esta

estrutura foi escolhida porque o tratamento de inserções e remoções é facilitado. A rede também armazena o número de pesos W e S de seus neurônios e possui métodos para medir a similaridade entre neurônios (Equação 3.1) e para atualizar adequadamente neurônios quando necessário, além de inseri-los.

Um problema do ANS é a falta de um mecanismo de controle de inserção de neurônios nas redes neurais do Módulo Coordenador. Se a cada evento ocorrente um neurônio for inserido, a rede pode vir a acrescentar uma quantidade exacerbada de neurônios. A existência de um grande número de neurônios pode levar a dois problemas que comprometem o desempenho da ANS: o vício nos padrões inseridos (*overfitting*) e uma demora no cômputo das saídas do CM.

Para evitar este problema neste trabalho, antes da atualização de um neurônio ser realizada, é verificado se existe um neurônio cuja similaridade com o neurônio candidato é superior a Lim_{SIMIL} . Se existir, tal ajuste é feito no neurônio com maior similaridade e a inserção é cancelada. Caso contrário, o neurônio é inserido. Desta forma, nas redes do Módulo Coordenador do AIS, dois parâmetros tornam-se responsáveis pelo controle do crescimento de uma rede do CM: Lim_{SIMIL} e o parâmetro de suavização das funções Gaussianas σ .

Como um acréscimo ao controle de crescimento, definiu-se uma constante Lim_{DEL} . Caso a média dos pesos S de um neurônio seja inferior a Lim_{DEL} após uma atualização, o neurônio é removido. Este critério de eliminação de neurônios considera que um neurônio cujos pesos S possuam valor muito baixo dificilmente enviará um sinal suficientemente forte para influenciar na saída y da rede. Os valores do vetor S somente estariam suficientemente baixos caso um neurônio seja atualizado diversas vezes a partir de outro neurônio que tenha baixa similaridade (Equação 3.4).

Durante testes preliminares do AIS, foi verificado que a equação para a obtenção da saída das redes do Módulo Coordenador enfrenta uma dificuldade no cálculo de similaridade quando se trata da rede responsável pelas medições de distância a obstáculos. Em várias situações, o robô lê a distância máxima captada pelo sensor laser. Se em cada captura houver medições máximas, muitos neurônios terão por consequência a similaridade idêntica, comprometendo o desempenho da AIS. Por isso, será capturada a média das similaridades entre cada medida sensorial e cada peso w_i ao invés do máximo, como mostra a Equação 4.1.

$$y = \frac{\sum_{i=1}^{N_n} \Delta(\Phi_{w,i}, \Omega_{x,i})}{N_n} \quad (4.1)$$

Um último ajuste na estrutura das redes do Módulo Coordenador foi no valor de α . O cálculo original deste valor considera o número total de eventos ocorrentes. Uma tradução direta da fórmula seria trocar o número de eventos pelo número de neurônios, dados que eram equivalentes no ANS, entretanto tal valoração de α insere um viés no ajuste dos neurônios de acordo com o instante de inserção. A ocorrência de uma colisão em um evento tardio, por exemplo, seria pouco ajustada com o tempo, enquanto uma colisão similar a algo que aconteça no início da atuação da ANS sofreria mudanças drásticas. Para evitar que seja dada prioridade a um tipo específico de evento em detrimento de outro, o AIS utiliza um valor constante maior do que 1 para α .

4.3 Implementação do módulo de saída e da AIS

O AIS contém várias redes do Módulo Coordenador, que estão armazenadas em um vetor. Além destas redes, armazena os pesos w e \bar{w} . A classe do AIS possui métodos para captar dados dos sensores e comportamentos inatos, para ajustar os seus pesos e para computar sua própria saída. Os comportamentos inatos, tal como na rede ANS, são ações simples que são combinadas para realizar comportamentos complexos. Para a rede AIS, ao contrário da formulação vista para a rede ANS, considera-se que a saída destes módulos pode ser de um número qualquer de dimensões. Para a escolha realizada neste trabalho, esta saída é um vetor em \mathbb{R}^2 .

Uma generalização da equação para a obtenção dos comportamentos combinados Y (Equação 3.6) pode ser obtida através da Equação 4.2, onde m_i é a influência do comportamento inato i , x_i é o valor enviado pelo módulo inato associado, e N é o número de comportamentos. A Equação 4.3 mostra como m_i é obtido: ϵ , $\epsilon \ll 1$ é uma constante positiva, $y_{i,j}$ é a saída da j -ésima rede neural do CM associada ao comportamento i , $w_{i,j}$ é o peso excitatório, ou seja, que estimula o comportamento i de saída de cada rede neural do CM, $\bar{w}_{k,j}$ é o peso inibitório produzido por uma rede neural j associada a outro comportamento k , e C e D são, respectivamente, o número de redes neurais associadas e de não associadas ao comportamento i . A escolha da operação de máximo ao invés da operação de produto foi feita observando que uma saída nula de um módulo coordenador irá anular toda a resposta associada

àquele comportamento, mesmo que este seja o mais adequado. Este problema ocorreu nos testes preliminares da rede para a escolha quando o robô estava em um status de colisão mas não havia inserido nenhum neurônio associado à falta de comunicação com um vizinho. Para evitar resultados negativos, o valor de m_i tem um limite inferior de ϵ para garantir valores não-negativos.

$$Y = \sum_{i=1}^N \dot{m}_i x_i, \quad \dot{m}_i = \frac{m_i}{\sum_{j=1}^N m_j} \quad (4.2)$$

$$m_i = \max(\epsilon, \epsilon + \max_{j=1}^C w_{i,j} y_{i,j} - \max_{j=1}^D \overline{w}_{i,j} y_{i,j}) \quad (4.3)$$

Foram notadas, entretanto, algumas deficiências na Equação 4.3. A equação resultante mostrou-se problemática para o problema abordado devido à subtração, resultando em muitas situações em que um dos valores de m_i torna-se ϵ devido à subtração (o valor tornou-se negativo). Para evitar este problema, foi adotado neste trabalho um valor para ϵ maior que o sugerido no trabalho original ($\epsilon \ll 1, \epsilon > 0$). Na avaliação do método (Capítulo 5), ϵ recebe o valor de 0.75.

A generalização da fórmula de obtenção dos pesos de influência das redes do CM (Equação 3.7) foi realizada neste trabalho e mostrada na Equação 4.4. U_b representa a resposta dada pelo módulo inato do comportamento b ao qual a rede neural cuja saída Y_b está associada, U_i é a resposta dos demais módulos inatos, e ζ é uma taxa de aprendizado. A operação $\max(\cdot)$ foi escolhida porque eram considerados apenas dois comportamentos inatos, e uma generalização adequada exige a consideração de aplicações onde três ou mais comportamentos devem ser balanceados. As funções $g(\cdot)$ e $h(\cdot)$ são sigmoidais, com valores entre 0 e 1. A função sigmoide é apresentada na Equação 4.5. Neste trabalho, ambas as funções serão idênticas, tendo como parâmetros $\phi = 0.5$ e $\lambda = 0.5$.

$$\begin{aligned} w_b^{t+1} &= w_b^t + \zeta(1 - w_b^t)h(Y_b)g(U_b) \\ \overline{w}^{t+1} &= \overline{w}^t + \zeta(1 - \overline{w}^t)g(Y_b)\max(h(U_i)) \end{aligned} \quad (4.4)$$

$$f(i), g(i) = \frac{1}{1 + e^{-\phi i - \lambda}} \quad (4.5)$$

Uma análise tanto da atualização de w no AIS quanto no ANS indica um segundo problema. Esta atualização faz com que o peso necessariamente cresça: o valor de w situa-se no intervalo de 0 a 1, e por consequência $(1 - w^t)$

nunca será negativo. Além disso, as funções sigmoidais também não podem apresentar valores nulos. Desta forma, em algum instante da navegação, todos os pesos w e \bar{w} estarão com valores saturados, próximos de 1, e portanto incapazes de impor alguma diferença no ajuste de neurônios, tornando as saídas do CM o único critério que realmente influencia no comportamento. Mesmo em intervalos de simulação menores e com uma taxa ζ baixa, o problema da baixa representatividade de w e de \bar{w} mantém-se, pois os pesos acabam se tornando excessivamente dependentes de eventos anteriores, dificultando a adaptação a situações novas. Para a solução deste problema, os pesos w são modularizados de modo que sua soma dê 1, como mostrado na Equação 4.6.

$$w_b^{t+1} = \frac{w_b^{t+1}}{\sum_{i=1}^N w_i^{t+1}} \quad (4.6)$$

Para os pesos barrados, outra abordagem foi adotada: considerou-se que que $\bar{w} = 1$. Isto foi feito para que, em combinação com o valor ϵ , fossem encorajadas influências de comportamento que tendessem fortemente para um comportamento específico quando uma situação na qual, por exemplo, o robô esteja colidindo por muito tempo com um obstáculo. Mesmo que as respostas do CM y indiquem uma influência maior de um comportamento, é possível que a discrepância entre os valores y não seja suficientemente alta para que o robô saia de um estado de colisão, o que pode vir a acontecer pela independência existente entre cada rede do CM. Assim, a constante ϵ ganha uma função de regular a amplitude de variação de parâmetros. Quanto maior for ϵ , menor será esta discrepância.

4.4 Cobertura de área: SLACS

Como visto na Seção 3.2, um mecanismo que permite que um sistema multirrobótico realize uma cobertura de área de um ambiente desconhecido pode ser elaborado através da obtenção de centroides de um diagrama de Voronoi, direcionando os robôs para tais centroides, a fim de se obter um posicionamento aproximado de um Diagrama Centroidal de Voronoi. Estes métodos exigem, entretanto, a construção de diagramas de Voronoi por cada robô, o que exige dados de vários robôs pertencentes a sua vizinhança. Além disso, esta construção exige dados de todos os robôs dentro do alcance de comunicação de um robô, e precisa de um método para o fechamento de um polígono de Voronoi. Este fechamento deve considerar tanto eventuais

obstáculos enfrentados pelo robô quanto evitar polígonos infinitos.

No trabalho de Tan et al. (2004), existe um fechamento de polígono planejado para o caso de polígonos infinitos, mas não existe tratamento para obstáculos na construção destes polígonos. O trabalho de Calvo (2012) realiza o tratamento truncando os polígonos de acordo com as medidas dos sensores de distância a obstáculos do robô. O método, entretanto, exige a construção explícita dos polígonos para o posterior cálculo dos centroides.

O trabalho de Rounds e Chen (2009) resolve o problema do cálculo de centroides sem a montagem explícita do diagrama de Voronoi, mas não trabalha com medidas de distância de obstáculos.

O método de cobertura de área escolhido para ser um dos comportamentos foi desenvolvido neste trabalho: o SLACS (*Sample Lloyd based Area Coverage System*) (Batista et al., 2013). Este sistema possui o tratamento de obstáculos e polígonos infinitos do mesmo modo que o de Calvo (2012), mas não exige o cômputo dos polígonos de Voronoi, como no trabalho de Rounds e Chen (2009), sendo também descentralizado e sem restrições de dependência de vizinhos.

A abordagem do SLACS resume-se em cinco etapas:

- Obtenção das amostras;
- Detecção de vizinho mais próximo;
- Captura de amostras do vizinho mais próximo;
- Cálculo do próprio centroide;
- Navegação em direção ao centroide.

A obtenção de amostras utiliza diretamente os dados sensoriais do robô. O trabalho original do SLACS, assim como é adotado neste trabalho, utiliza vinte feixes de um sensor laser ao redor do robô, que cobrem 360 graus. As amostras, neste caso, são geradas de modo determinístico a partir do posicionamento relativo do robô, na direção dos feixes *laser*. As amostras de um determinado feixe são geradas de modo que estejam equidistantes entre si. O Algoritmo 1 descreve o processo de geração das amostras, denotadas por A_{i*j} .

O SLACS trata o fechamento de polígonos porque as amostras utilizadas para a estimativa do centroide são limitadas em relação à posição do robô pelo

Algoritmo 1: Geração de amostras do algoritmo SLACS.

Entrada: Número de feixes de um sensor m , alcance máximo dos feixes R_D , número máximo de amostras em uma direção A^{MAX} , medidas de distância dos feixes sr_i $i=1..m$, ângulo relativo de dispersão do feixe em relação à frente do robô θ_i $i=1..m$.

```
1: para  $i=1..m$  faça
2:    $j = 1$ ;
3:   Intervalo :=  $R_D/A^{MAX}$ ;
4:   enquanto Intervalo  $\leq sr_i$  faça
5:      $A_{i*j} = [\cos(\theta_i)*Intervalo, \sin(\theta_i)*Intervalo]$ ;
6:     Intervalo := Intervalo +  $R_D/A^{MAX}$ ;
7:      $j = j + 1$ ;
8:   fim enquanto
9: fim para
```

alcance dos sensores e, também, por ter esta geração limitada pela medida do sensor, evitando que surjam amostras em regiões com obstáculos.

Na etapa seguinte, o robô identifica seu vizinho mais próximo, procedimento que é também necessário para o compartilhamento de conhecimento da AIS. Considerando que os robôs possuem dispositivos de comunicação sem fio, esta identificação pode ser feita através do cálculo da intensidade de sinal de mensagens dos demais robôs. Considera-se que há um mecanismo que permita o cálculo da pose relativa de um robô e seu vizinho. Com a pose relativa calculada, basta que o robô vizinho envie o número de amostras criadas por feixe para que a posição relativa destas amostras seja calculada. É vantajoso o envio do número de amostras por feixe ao invés da medida de distância pela diferença que é o envio de um número inteiro (um número discreto de amostras) e um número de ponto flutuante (a medida de distância do feixe). Se desejado pelo usuário, o método não limita o número de vizinhos adotados, sendo possível a adoção de quantos vizinhos for desejado. O uso de um único vizinho foi escolhido por motivos de simplicidade e para estar de acordo com a política de comunicação utilizada pelo algoritmo de aprendizado adotado.

Considerando as amostras geradas pelo robô e as amostras de seu vizinho mais próximo, o centroide pode ser calculado. O procedimento de cálculo é idêntico ao utilizado no algoritmo de Lloyd probabilístico: calcula-se quais amostras estão mais próximas do robô do que de seu vizinho, e é feita uma média destas amostras. A média é a estimativa do centro de massa do polígono. O robô, então, navega em direção a este centroide. A Figura 4.1

ilustra robôs, suas amostras e o diagrama de Voronoi que está implícito neste procedimento. Se não houver nenhum robô dentro do alcance de comunicação, o robô calcula seu centroide a partir de suas próprias amostras.

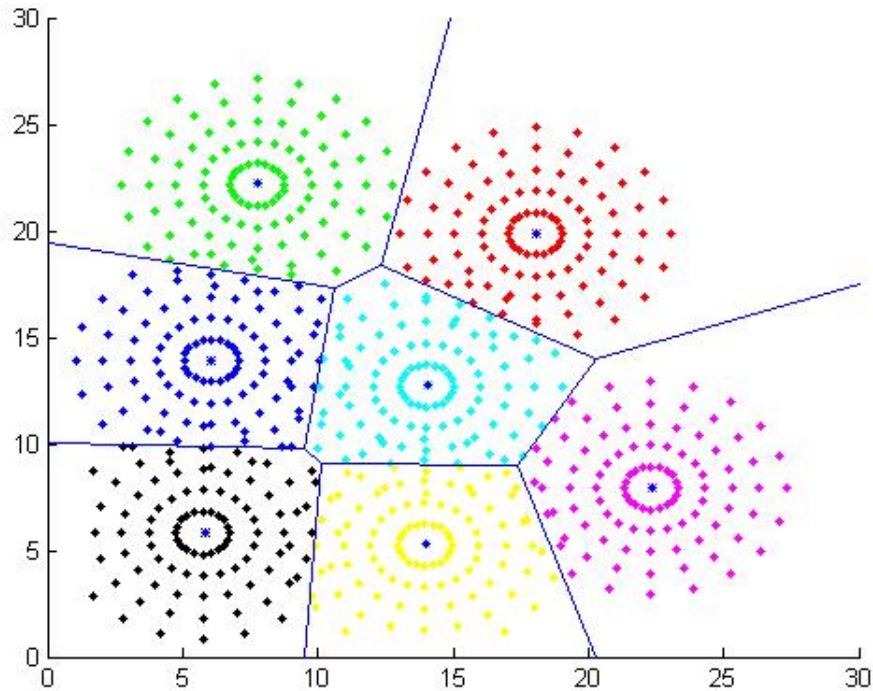


Figura 4.1: Posicionamento de robôs adotando o algoritmo SLACS. Para esta imagem, consideram-se todos os robôs para o cômputo individual de cada centroide.

4.5 Aplicação do SLACS para a Escolta

O comportamento de escolta pretendido neste trabalho será obtido a partir da combinação dos resultados de dois comportamentos distintos. Ambos os comportamentos são descritos na forma de um vetor bidimensional, que direciona o robô até um determinado local.

O mais simples destes comportamentos é o de manutenção de distâncias mínimas: ele simplesmente retorna um vetor que direciona um robô até uma meta enquanto o mesmo estiver a uma distância maior do que a desejada. Caso contrário, o vetor é o mesmo, mas com sentido contrário, de modo a distanciar o robô da meta de sua escolta. O objetivo deste comportamento é, portanto, perseguir a meta evitando colisões com a mesma.

O outro comportamento é o algoritmo SLACS descrito anteriormente (Seção 4.4). Seu objetivo é manter uma coesão entre os robôs ao mesmo tempo que evita-se a colisão com obstáculos e com os demais robôs do enxame.

As Figuras 4.2, 4.3 e 4.4 mostram a diferença de posicionamento de 20 partículas simuladas realizando a tarefa de escolta a partir do uso de um valor fixo em um ambiente vazio: uma influência muito alta do comportamento de cobertura de área na navegação do robô resulta em uma menor proximidade média do agente a ser escoltado, enquanto uma influência elevada do comportamento de manutenção de distância mínima resulta em um grande número de robôs buscando o posicionamento adequado em relação à meta em um espaço insuficiente para o posicionamento de todos, resultando em colisões. Um balanceamento mais adequado da influência de comportamentos pode concentrar um número razoável de robôs ao redor do agente a ser escoltado sem que um alto número de colisões entre os robôs ocorra. É importante salientar que tais pesos fixos foram capazes de obter estes resultados em um cenário onde o ambiente é livre de obstáculos e que a meta de escolta não está em movimento.

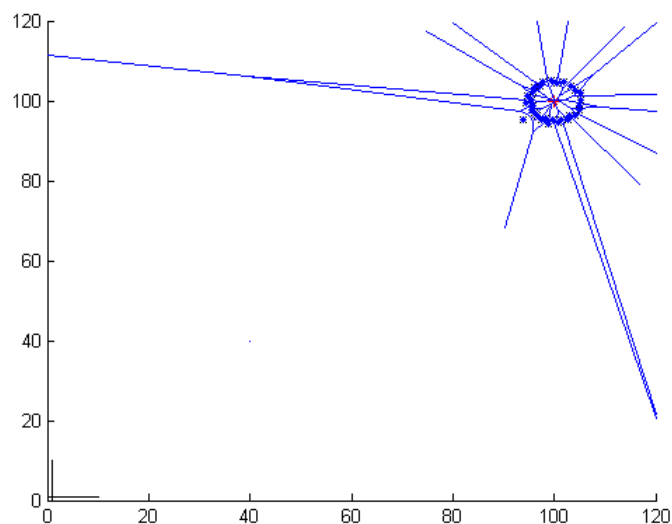


Figura 4.2: Escolta realizada com peso 0.8 no comportamento de avanço até a meta e 0.2 no comportamento de cobertura de área. Um grande número de partículas atingiu o posicionamento ideal, entretanto ficaram muito próximas entre si. Caso robôs se posicionassem desta maneira, várias colisões ocorreriam.

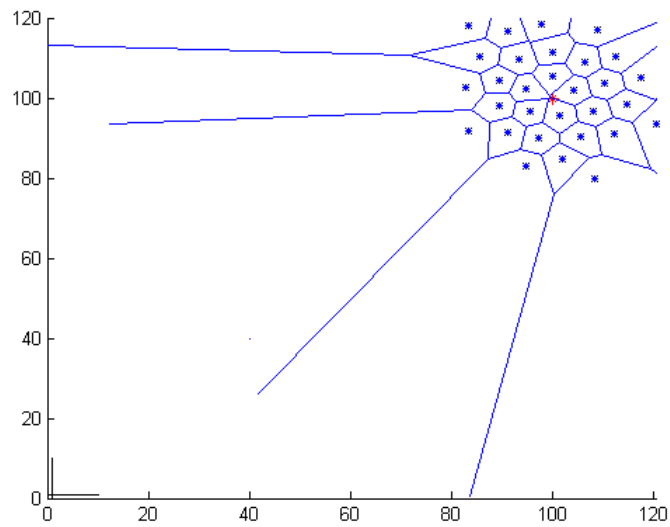


Figura 4.3: Escolta realizada com peso 0.2 no comportamento de avanço até a meta e 0.8 no comportamento de cobertura de área. Apesar do espaçamento entre os robôs evitar riscos de colisão, poucos robôs mantêm uma distância suficientemente próxima.

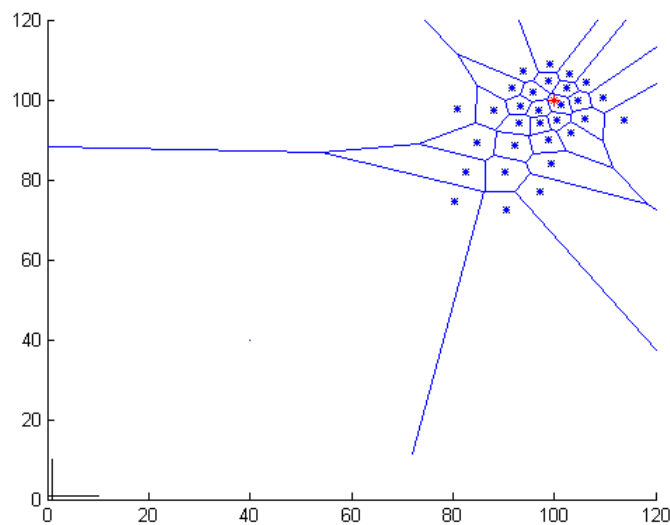


Figura 4.4: Escolta realizada com pesos iguais. As partículas se posicionaram de maneira que uma “cobertura de área localizada” foi realizada.

4.6 Adaptações da Rede Neural Hierárquica para Enxames de Robôs

Para o gerenciamento do aprendizado neste trabalho, cada robô do enxame possui uma arquitetura similar à de Calvo et al. (2010). Os dois

comportamentos inatos serão a manutenção de uma distância mínima até o agente a ser escoltado e o método de cobertura de área baseado em Diagramas Centroidais de Voronoi, tal como descrito na Seção 4.5. O objetivo do método de aprendizado é, portanto, balancear a influência de cada comportamento na realização da tarefa de acordo com as informações obtidas pelos sensores dos robôs visando uma maior eficácia na realização da tarefa de escolta. Neste trabalho, pretende-se realizar uma escolta minimizando a diferença entre a distância do robô até um agente a ser escoltado e uma distância desejada fixa, considerando um ambiente desconhecido, onde obstáculos estão presentes e evitando colisões. Uma influência muito alta do comportamento de avanço até a meta pode provocar colisões, enquanto que o contrário dificulta ou até mesmo impossibilita a navegação do enxame até o local desejado.

As redes neurais pertencentes ao CM neste trabalho serão duas, que representam os dados sensoriais descritos a seguir. Os dois primeiros dados serão entradas da rede do Módulo Coordenador responsável pela manutenção de distâncias mínimas, enquanto os demais dados estão associados à rede que estimula o comportamento de cobertura de área.

- Diferença da distância entre o robô e a entidade a ser escoltada e a distância mínima desejada. Um limite superior equivalente ao dobro da distância mínima será adotado pois esta informação só deverá ser relevante a níveis de aprendizado e influência no comportamento do robô caso ele esteja próximo de quem escoltará. Neurônios inseridos na rede neural associada estimularão a manutenção de distâncias;
- Distância entre um robô e seu obstáculo mais próximo que não seja a meta de escolta. Esta informação sensorial é capaz de indicar o risco de colisão entre dois robôs do enxame ou o risco de colisão a um obstáculo fixo. O limite superior de distância terá o mesmo valor que o alcance de comunicação do robô, e será utilizado quando não houver vizinhos comunicáveis e não houver obstáculos detectáveis pelo sensor de feixes *laser*. A rede neural associada é a mesma da informação descrita anteriormente.
- Informações de um sensor de distâncias através de feixes *laser*. Determina a distância entre um robô e um obstáculo qualquer, seja ele uma parede, outro robô, ou a meta cuja escolta é objetivada. Tal sensor utiliza diversos feixes, que medem distâncias em várias direções. O valor

é limitado pelo alcance dos *lasers*, denotado por R_D . A rede associada estimula a cobertura de área.

A segunda informação sensorial, referente à distância entre um robô e seu vizinho mais próximo, era anteriormente associada a uma rede neural própria, que estimulava a cobertura de área, visando coesão entre os robôs do enxame. Entretanto, verificou-se que as regras consideradas para a inserção de neurônios das demais redes já cobriam todos os casos relevantes para a tarefa de escolta. Além disso, foi verificada uma dificuldade dos robôs para a manutenção de distâncias seguras quando um robô está próximo tanto da meta de escolta quanto de outro robô. Para evitar que tal conflito faça com que o robô colida na meta de escolta, este dado foi associado à rede de manutenção de distâncias mínimas.

Visto que todos os robôs irão possuir uma rede hierárquica tal como a descrita e por consequência as mesmas redes neurais no CM, e também que os neurônios recém-inseridos são essencialmente dados brutos de um sensor, é possível desenvolver um mecanismo simples de compartilhamento de informação: para cada evento detectado pelo robô, ou seja, em qualquer momento em que um neurônio é inserido em sua arquitetura, este mesmo neurônio é transmitido para os demais robôs dentro de sua área de comunicação. Robôs que receberem este neurônio podem, em outra iteração, transferir o mesmo neurônio caso não detectem um evento pelo qual passam.

4.6.1 Compartilhamento de dados

O enxame de robôs apresentado neste trabalho considera que há uma comunicação ativa entre robôs que estejam dentro de um alcance de comunicação R_C . Considera-se que os robôs podem transmitir todos os seus dados sensoriais (distâncias a obstáculos, ao robô vizinho mais próximo, à meta de escolta). Para a aplicação do algoritmo SLACS, assume-se que exista um mecanismo que estime a posição relativa de um robô vizinho, o que permite a estimação das amostras a partir do dado bruto de distância obtido pelos feixes *laser*. Esta estimativa, apesar de exigir que cada robô calcule tanto suas amostras quanto as de seu vizinho, diminui consideravelmente o número de dados transmitidos em relação ao envio e posterior conversão de dados de cada robô.

Outra alternativa seria o envio do número de amostras geradas, que iria requerir apenas o envio de números inteiros representando o número de

amostras na direção de cada feixe *laser*. Para a aplicação da escolta utilizando influências constantes ou do SLACS isoladamente, é uma solução vantajosa. Para o uso do AIS, entretanto, enfrenta-se o problema de lidar com a perda de dados através da discretização caso apenas o número de amostras seja enviado, o que poderia prejudicar o desempenho do aprendizado (Di Mario e Martinoli, 2013). Para este projeto, portanto, utiliza-se o dado bruto dos sensores de distância a obstáculos.

Mostra-se conveniente aproveitar a existência de vários robôs para a adoção de um modo de compartilhar conhecimento entre os robôs. Trabalhos como o de Pugh e Martinoli (2006) e de Zuo et al. (2010) permitem que dados sejam compartilhados entre os robôs para a aceleração da convergência.

No caso da AIS, estruturada para o aprendizado por reforço de um único robô, o compartilhamento de conhecimento é um acréscimo, não o mecanismo fundamental de aprendizado. Visto que cada neurônio representa inicialmente os dados puros de um conjunto de dados sensoriais, é possível transmitir estes dados para um robô vizinho no instante em que o evento ocorre. Deste modo, o robô vizinho aprende sobre a condição no qual o evento ocorre, seja este evento uma colisão, um afastamento do robô em relação à meta, ou outro problema. Um robô é capaz de transmitir estes neurônios para todos os robôs dentro de seu alcance de comunicação R_C . O mesmo mecanismo que se comunica com os demais robôs do enxame para compartilhar as amostras do SLACS pode ser utilizado para este fim.

No próximo Capítulo, são descritas as avaliações feitas para validar a eficiência da proposta.

Experimentos realizados

Neste capítulo, o mecanismo de escolta proposto será avaliado por meio de dois conjuntos de experimentos. Para o primeiro conjunto, o mecanismo de escolta será avaliado utilizando influências constantes e no segundo experimento, utilizando-se o método de aprendizado AIS. Depois, uma análise comparativa entre eles será apresentada. Antes, porém, da avaliação do sistema de escolta, será apresentada uma avaliação do mecanismo de cobertura também proposto neste trabalho. Todos os testes foram realizados utilizando-se o simulador Player/Stage.

5.1 Avaliação da cobertura de área

Um produto deste trabalho foi a elaboração de um mecanismo de cobertura de área baseado no método de Lloyd para cálculo de centroides de um Diagrama Centroidal de Voronoi. As métricas de avaliação da cobertura de área são:

- A porcentagem de área coberta pelos sensores do enxame;
- Média das distâncias entre um robô e seu vizinho mais próximo, para verificar se o espalhamento dos robôs ocorre de maneira eficaz;
- Número de robôs sem comunicação com algum outro membro do enxame.

Os testes apresentados são os vistos no artigo de Batista et al. (2013), decorrente do desenvolvido neste trabalho.

O método de cobertura de área proposto para ser um dos comportamentos do enxame de robôs foi avaliado separadamente para conferir se o desempenho é similar ao da cobertura de área através da montagem e cálculo de centros de massa do método de Lloyd tradicional, que utiliza como referência todos os vizinhos dentro de sua área de comunicação, enquanto o método probabilístico de Lloyd proposto utiliza somente o vizinho mais próximo dentro da capacidade de comunicação dos robôs.

A comparação foi realizada com um método que constrói explicitamente diagramas de Voronoi implementado por Calvo (2012) que é chamado neste trabalho de *Truncated Closure Lloyd-based Area Coverage System (TCLACS)*. O TCLACS utiliza dados sensoriais para realizar o fechamento dos polígonos de Voronoi de modo a comportar informações de obstáculos. Três cenários de cobertura de área foram considerados: um local vazio, sem a presença de obstáculos, um ambiente dividido em salas por paredes, e uma redução do cenário de salas. No primeiro cenário, os robôs estão no centro de um ambiente de grande escala, como visto na Figura 5.1. No segundo cenário, todos os robôs estão concentrados na sala do canto superior, à esquerda, como mostrado na Figura 5.2. Uma versão reduzida do mapa dividido em salas é utilizada para o terceiro cenário de cobertura de área (Figura 5.3). Para estes testes, considerou-se que o alcance de comunicação dos robôs equivale à metade do alcance sensorial, e que, se houvesse uma parede entre dois robôs, a comunicação entre os mesmos seria desconsiderada, pois impossibilitaria a estimativa de distância entre os robôs a partir da intensidade dos sinais.

O código-fonte utilizado no trabalho de Calvo (2012) para a implementação do TCLACS foi escolhido como base e comparação para a avaliação de cobertura de área. Em tal estrutura, cada robô possui controle diferencial, podendo aplicar uma velocidade linear ou angular a cada ciclo, e nunca ambas ao mesmo tempo. A migração para a estrutura de outro trabalho foi feita para que a comparação entre os dois métodos de cobertura de área seja de maior validade, possuindo as mesmas mecânicas de navegação.

Optou-se por migrar o SLACS para a estrutura na qual o TCLACS foi implementado porque o SLACS oferece maior simplicidade e portabilidade. Estas duas características do SLACS em relação ao TCLACS são dadas, principalmente, pela dispensa de construção de polígonos e de métodos de fechamento, que estão implícitos nas operações de criação e de geração de

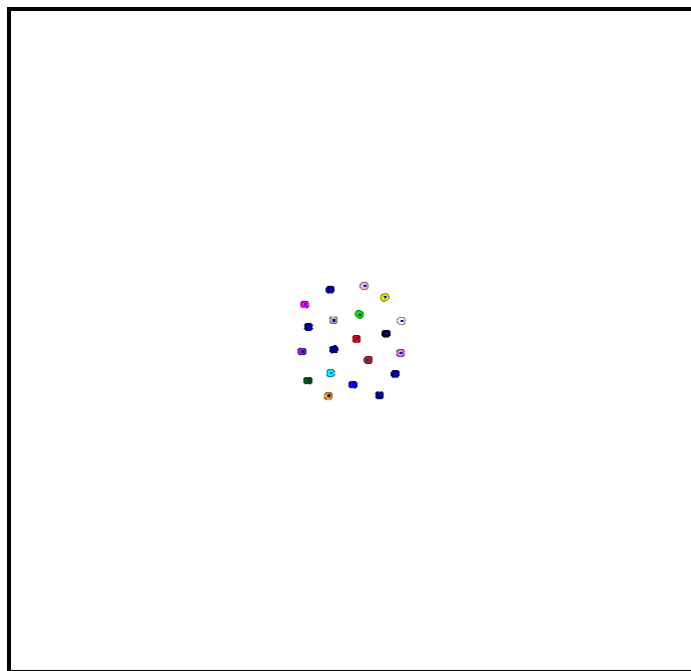


Figura 5.1: Primeiro ambiente de testes da cobertura de área e configuração inicial dos robôs.

amostras.

Um comportamento adicional é dado ao SLACS para evitar problemas de dispersão: quando um robô encontra-se a uma distância de seu vizinho mais próximo que ultrapasse um limiar, o robô para a fim de evitar a desconexão. Para os testes de cobertura apresentados a seguir, este limiar equivale a 80% do alcance de comunicação do robô. O alcance sensorial do robô nestes testes é de 6 metros, e o alcance de comunicação chega a 8 metros, portanto o limiar de pausa do robô é de 6.4 metros. Para o cálculo da área coberta, é considerada uma diminuição gradual da confiança dos sensores de distância de obstáculos: quanto mais distante do robô estiver a área coberta, menor será a confiança da métrica e portanto menor será a contabilização de tal região como área coberta.

A Figura 5.4 mostra as configurações finais dos robôs realizando a cobertura de área a partir dos métodos TCLACS e SLACS. Os resultados foram bastante similares: O TCLACS cobriu 46.05% do ambiente sem obstáculos, enquanto o SLACS cobriu 41.45%. Nota-se que a dispersão dos robôs no SLACS foi maior, deixando regiões não cobertas maiores. A provável causa deste problema é a comunicação com um único vizinho: enquanto o TCLACS reúne dados de um maior número de vizinhos, ele tende a não deixar áreas não cobertas internas vazias.

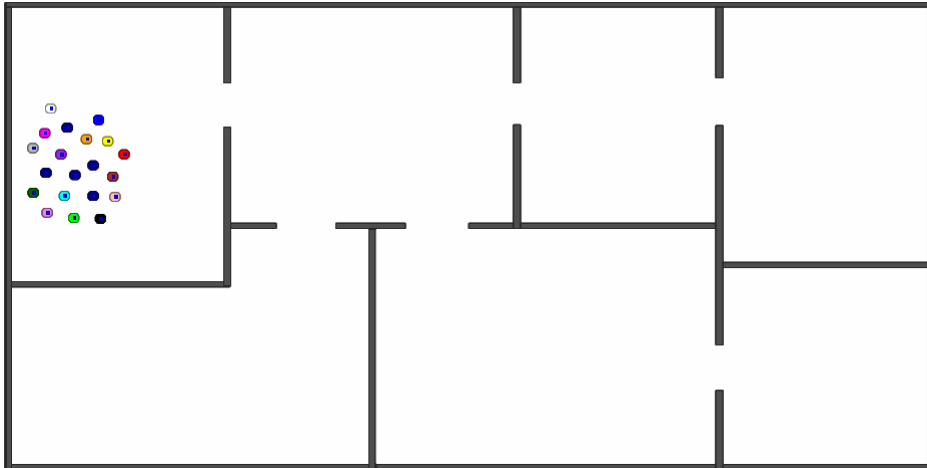


Figura 5.2: Segundo ambiente de testes da cobertura de área e configuração inicial dos robôs.

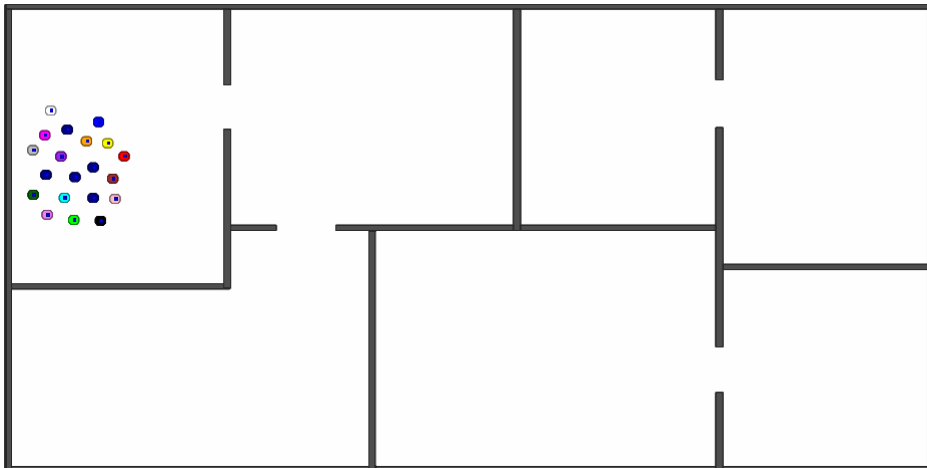


Figura 5.3: Terceiro ambiente de testes da cobertura de área e configuração inicial dos robôs.

Os gráficos de distância entre robôs vizinhos e o desvio padrão destas distâncias para ambos os testes podem ser vistos na Figura 5.5. As distâncias foram bastante parecidas, mas no caso do SLACS o desvio padrão foi menor. Este é um resultado esperado, pois este método realiza a estimativa do centroide considerando apenas o vizinho mais próximo. Observa-se que a distância média entre os robôs executando o SLACS é muito próxima do limiar de distância de comunicação de 6.4 metros.

O segundo cenário, composto por sete salas, apresenta o desafio da cobertura de área em um ambiente de maior complexidade. Para tornar o cenário mais realista, considera-se que dois robôs não podem se comunicar caso haja uma parede entre eles. Neste cenário, a cobertura de área através

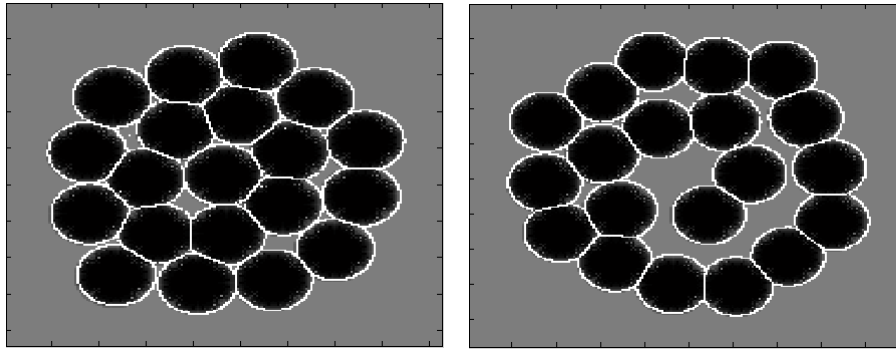


Figura 5.4: Configuração final da cobertura de área no ambiente vazio: esquerda: TCLACS; direita: SLACS. Retirado de Batista et al. (2013).

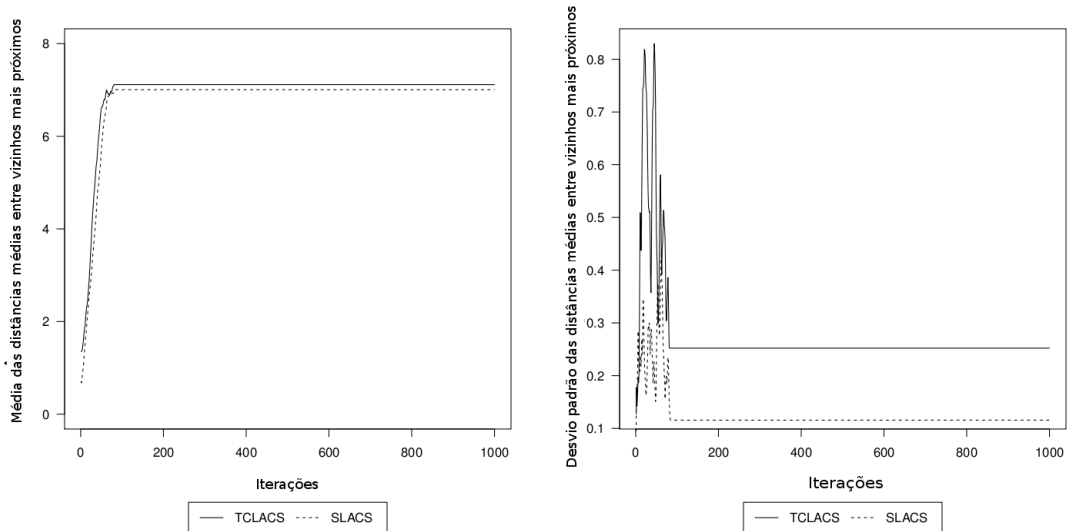


Figura 5.5: Gráficos da distância média (esquerda) e do desvio padrão de distância (direita) entre os robôs no primeiro cenário de cobertura de área. Adaptado de Batista et al. (2013).

do algoritmo desenvolvido por Calvo et al. (2010) apresenta 72.46% de cobertura de área, enquanto o SLACS apresenta 73.07% de cobertura. O posicionamento final dos robôs pode ser visto na Figura 5.6.

Observa-se que foi formado um grupo de robôs desconexo no experimento envolvendo o SLACS, o que resulta tanto do espalhamento maior observado no experimento no ambiente vazio como também da desconexão causada pelas paredes. É possível observar que o número de robôs em cada sala é mais

balanceado na execução com o SLACS devido ao espalhamento mais rápido.

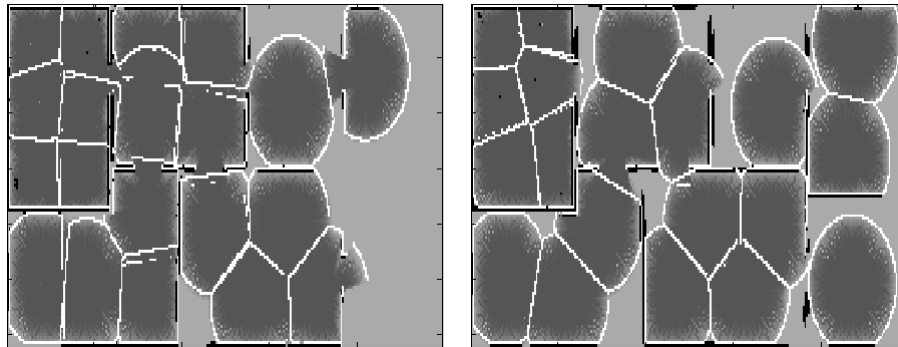


Figura 5.6: Configuração final da cobertura de área no ambiente de 7 salas: esquerda: TCLACS; direita: SLACS. Retirado de Batista et al. (2013).

Os gráficos de distância entre os robôs e desvio padrão (Figura 5.7) indicam, novamente, que as distâncias entre os robôs vizinhos são similares para ambos. O desvio padrão destas distâncias evidencia, novamente, que o desvio padrão é menor no teste do SLACS.

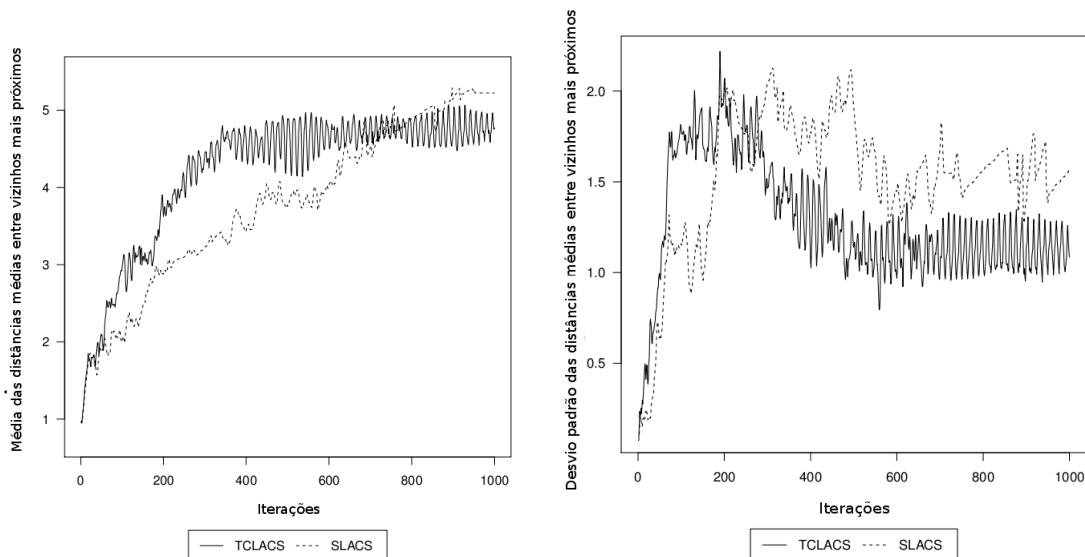


Figura 5.7: Gráficos da distância média (esquerda) e do desvio padrão de distância (direita) entre os robôs no segundo cenário de cobertura de área. Adaptado de Batista et al. (2013).

O experimento do segundo cenário analisa como ocorre o espalhamento em um ambiente que não pode ser amplamente coberto pelos robôs. No terceiro cenário, algumas salas do lado esquerdo do mapa do segundo cenário para observar o desempenho da cobertura de área em um ambiente de menores dimensões. A Figura 5.8 mostra a disposição final dos robôs. Vê-se, novamente, que a dispersão do SLACS mostrou-se maior. Ao contrário do cenário de cobertura de área anterior, não houve desconexões dos robôs que executavam o SLACS. O método TCLACS cobriu 76.43%, enquanto o SLACS cobriu 74.72%.

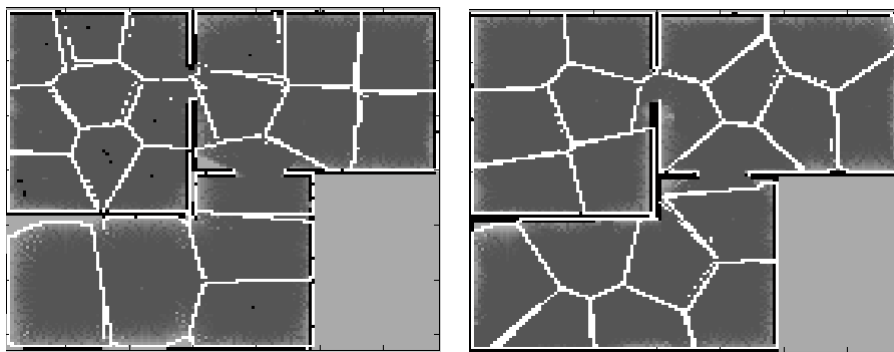


Figura 5.8: Configuração final da cobertura de área no ambiente de 3 salas: esquerda: TCLACS; direita: SLACS. Retirado de Batista et al. (2013).

Como nos cenários anteriores, a distância entre os robôs e o desvio padrão destas distâncias (Figura 5.9) indicam resultados de distância similares e um desvio padrão menor do SLACS.

Pode ser observado, entretanto, que o método probabilístico simplificado proposto teve menor desempenho em relação à conectividade no segundo cenário, visto que um pequeno grupo de três robôs se separou dos demais robôs. Isto pode ser devido à adoção de um vizinho único no método proposto em contraste ao método de Lloyd típico, que utilizou todos os vizinhos comunicáveis, e ao modo no qual os obstáculos estão posicionados.

5.2 Avaliação da escolta

O sistema de escolta a partir dos comportamentos de manutenção de distâncias e de cobertura de área via DCVs proposta neste trabalho será avaliada. Espera-se que as ações escolhidas pelos robôs e as experiências

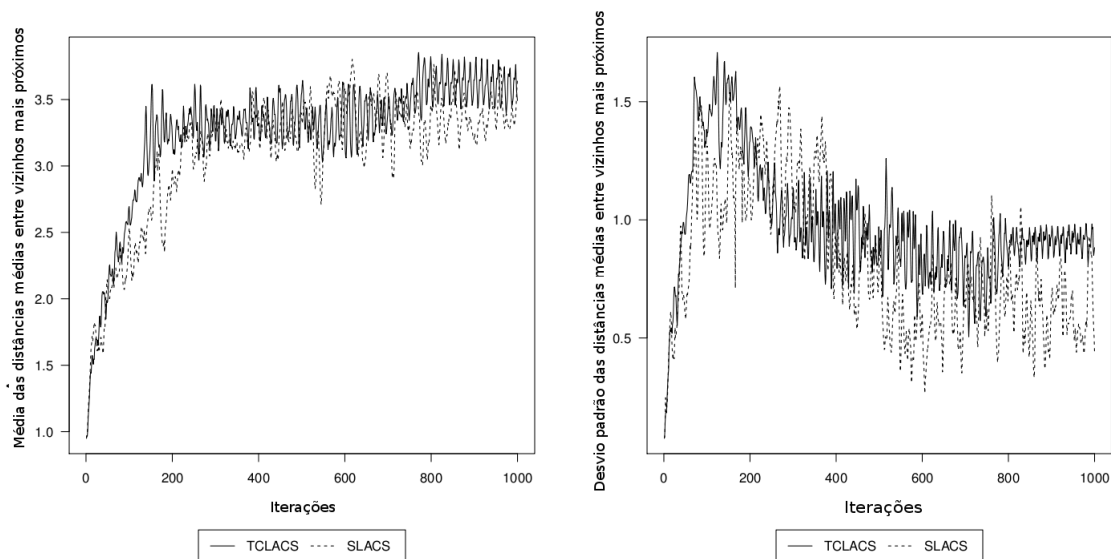


Figura 5.9: Gráficos da distância média (esquerda) e do desvio padrão de distância (direita) entre os robôs no terceiro cenário de cobertura de área. Adaptado de Batista et al. (2013).

obtidas através de aprendizado por reforço gerem comportamentos coletivos que possuam maior aptidão para realizar a tarefa desejada. Para realizar a escolta, dois comportamentos inatos serão adotados, um avanço até uma meta desejada e um comportamento de cobertura de área baseado em Diagramas Centroidais de Voronoi.

Os testes levarão em conta as seguintes métricas de avaliação, que são coletadas a cada segundo de simulação:

- Média da distância atual entre cada robô e a meta de escolta, o que representa a eficácia da cobertura ao redor do agente alvo;
- Média das distâncias entre um robô e seu vizinho mais próximo, indicando se há um espaçamento adequado entre os robôs. Se os robôs estiverem próximos demais entre si, a navegabilidade torna-se difícil por conta do risco de colisões;
- Número de colisões, capaz de indicar situações nas quais o comportamento de cobertura de área deveria ter uma maior influência;
- Número de robôs sem comunicação com algum outro membro do enxame, para avaliar a coesão entre os robôs para a realização da tarefa.

Robôs sem comunicação perdem grande parte de seu potencial de execução de comportamentos complexos devido à falta de interação entre os robôs.

Apesar de, aparentemente, uma média de distanciamento entre os robôs mais próxima de 1 ser melhor, é importante considerar que isso significa, dependendo do número de robôs realizando a escolta, um risco elevado de colisões entre os próprios robôs. A Figura 5.10 demonstra uma configuração espacial ideal dos robôs. Apenas seis robôs situam-se na região de distância mínima, enquanto os outros quatro robôs estão um pouco mais distantes. Este posicionamento coletivo seguro pode ser considerado um comportamento complexo gerado pela interação entre os robôs e pelo aprendizado. Pode-se dizer, portanto, que um bom desempenho depende de todos os critérios de avaliação acima descritos: um excelente desempenho em um quesito não compensa um desempenho muito fraco em outro. É necessário que os robôs estejam comunicáveis entre si, que se mantenham a um distanciamento razoável entre si mesmos e à meta, e que não ocorram colisões com muita frequência.

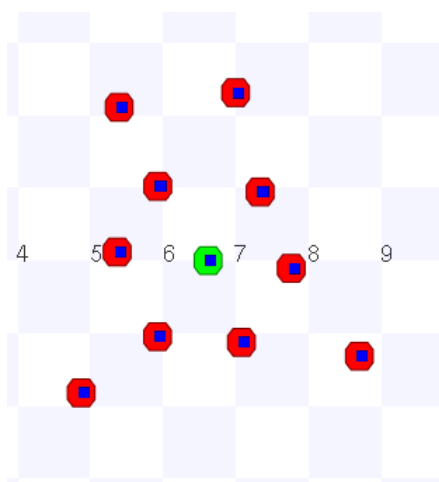


Figura 5.10: Exemplo de posicionamento adequado para a escolta.

Foi aplicado um leve ruído branco Gaussiano aditivo de igual intensidade em cada dado sensorial obtido, de modo a verificar a tolerância a ruídos da escolta e do AIS. Esta escolha foi feita levando em consideração que é um ruído amplamente utilizado em trabalhos de Robótica (Di Mario e Martinoli, 2013)(Balaguer e Carpin, 2008)(Genchev et al., 2008). O ruído R_{bg} é gerado com uso de intensidade linear (Int) igual a 0.0001, e é gerado a partir da transformação de Box-Muller (Box e Muller, 1958). A Equação 5.1 mostra

esta transformação, onde $Rand_1$ e $Rand_2$ são valores aleatórios de distribuição uniforme no intervalo [0..1].

$$R_{bg} = \sqrt{-2 * \ln(Rand_1)} (\cos(2\phi * Rand_2)) \quad (5.1)$$

Para a realização dos testes de escolta, um robô simulado foi adotado segundo as especificações descritas a seguir.

5.2.1 Modelo de Robô Adotado

Para a realização dos experimentos, um modelo de robô foi definido.

Assume-se que os robôs realizam comunicação através de rádio. O alcance do rádio é de 3 metros, portanto $R_C = 3$. O robô possui um sensor de feixes *laser* de proximidade dotado de 20 sensores, que distribuem-se em torno dos 360 graus do robô e alcançam até 2 metros ($R_D = 2$), um sensor que indica a posição relativa do agente que precisa ser escoltado em relação ao robô, e um sensor que indica a a posição relativa do robô a seus vizinhos dentro de sua área de comunicação.

Estes dois últimos sensores são, na verdade, informações que podem ser extraídas através de dados como a intensidade do sinal, e a implementação dos mesmos não foi feita neste trabalho por não ser o foco do mesmo.

O robô se movimenta de maneira onidirecional, a uma velocidade de 20 cm/s. As dimensões do robô, de forma aproximadamente retangular, são de 44x38x22cm. A navegação onidirecional foi adotada porque tem maior facilidade para seguir as trajetórias até os centroides gerados pelo SLACS, que podem surgir em locais de difícil acesso imediato para um robô de controle diferencial.

5.2.2 Avaliação da escolta

Nesta etapa, são apresentados os testes realizados para avaliar a eficácia do AIS para o balanceamento de comportamentos. Para isto, três cenários de escolta foram elaborados. Todos os testes ocorrerão por um período de 5 minutos. Cada teste foi repetido 20 vezes, e a média destas execuções foi considerada.

Para a realização destes testes, uma pequena modificação no algoritmo SLACS foi feita para uma melhor adequação à tarefa de escolta. Quando um robô está colidindo, ele irá desconsiderar as amostras de seu vizinho mais próximo. Isto foi feito porque, em alguns cenários onde dois robôs estão

próximos entre si e colidindo, eles apresentaram dificuldades para se recuperarem mesmo quando o comportamento de cobertura de área tem uma influência significativamente maior do que a manutenção de distâncias mínimas.

Os cenários foram elaborados de modo a explorar diferentes situações para os robôs, avaliando forças e fraquezas das abordagens utilizadas no projeto.

No primeiro cenário, mais simples, o alvo a ser escoltado navegará fazendo curvas em um ambiente vazio. Este cenário é ilustrado na Figura 5.11. Os rastros em verde indicam o formato das curvas presentes na trajetória realizada pelo agente meta. A velocidade linear mantém-se constante durante toda a trajetória, sem períodos de pausa. Apesar de apresentar um cenário simples, o padrão de movimentação da meta de escolta é o mais complicado entre os três cenários. Avalia-se aqui a capacidade do método de escolta proposto para aplicação do AIS de escoltar sem a presença de obstáculos, dando maior ênfase aos obstáculos dinâmicos que são os próprios robôs.

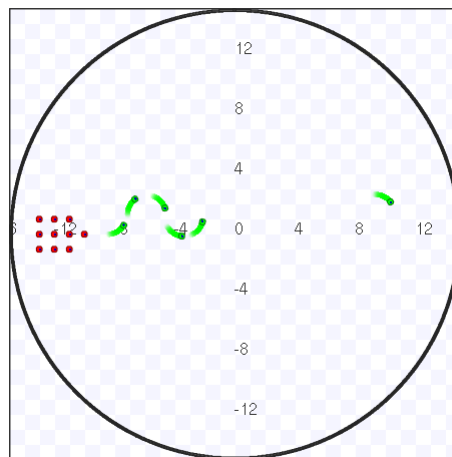


Figura 5.11: Primeiro cenário de teste de escolta.

Para o segundo cenário, o alvo irá percorrer um corredor, como mostrado na Figura 5.12. Inicialmente, o alvo fica estacionado por 50 segundos, e, quando atinge a posição mostrada à direita da imagem, que é alcançada por volta de 225 segundos após o início da simulação, para. Este cenário foi elaborado para verificar como os robôs irão reagir quando há a necessidade de alguns robôs ficarem distantes da meta de escolta, visto que a manutenção dos robôs a distâncias desejáveis da meta de escolta resulta em um alto número de colisões.

O segundo cenário apresenta um momento crítico no momento em que a meta de escolta alcança o corredor, que ocorre por volta de 100 segundos

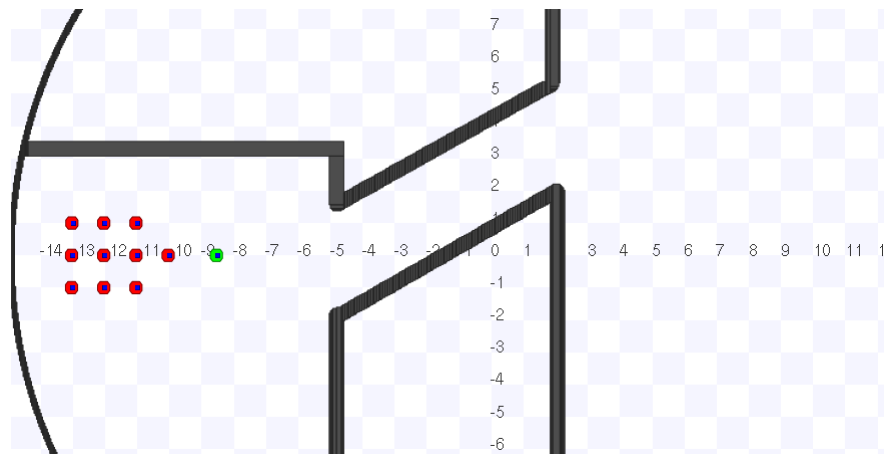


Figura 5.12: Segundo cenário de teste de escolta.

após o início do experimento. Neste momento, várias colisões devem acontecer. Visto que a meta de escolta não segue precisamente no ponto médio do corredor, durante a travessia pela mesma, robôs que estejam acompanhando mais próximos à parede mais próxima do alvo têm maiores chances de colisão.

O terceiro cenário, visto na Figura 5.13, apresenta uma maior variação de situações que os demais cenários, apesar de tais situações serem de solução mais fácil que a travessia do corredor presente no segundo cenário. Tal como no segundo cenário, a meta de escolta começa a navegar a partir de 50 segundos. Em seguida, ele andarà em linha reta até que se completem 200 segundos. O primeiro obstáculo a ser enfrentado apresenta-se na parte superior do mapa, e avalia o quão bem os robôs conseguem se deslocar para evitar colisões com tal obstáculo. O obstáculo seguinte é similar ao primeiro, mas encontra-se na parte inferior do mapa, levando a um deslocamento maior. O terceiro e último obstáculo consiste de duas barras verticais. Este obstáculo verifica se os robôs conseguem se dividir entre as três regiões de navegação geradas pelas barras (acima, abaixo, e entre as barras) para então alcançarem a meta de escolta e o restante do enxame.

Para os testes envolvendo o uso da AIS, a tabela de prioridades (Tabela 5.2.2) apresentada a seguir mostra as condições para um neurônio ser inserido. Caso um evento de maior prioridade ocorra, apenas o neurônio associado ao mesmo é verificado para inserção ou atualização de neurônios. Esta tabela foi construída levando em consideração o impacto de cada erro no desempenho da escolta.

A prioridade de cada inserção pode ser justificada através de uma análise

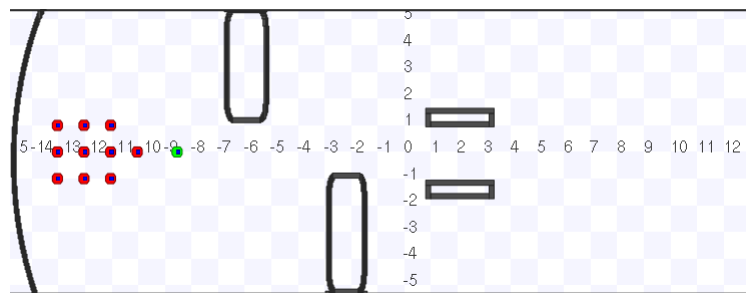


Figura 5.13: Terceiro cenário de teste de escolta.

Prioridade	Regra	Módulo Coordenador
1	Robô está a menos de 1 metro da meta de escolta	Manutenção de Distância Mínima
2	Robô está colidindo ou está a menos de 1 metro de um robô vizinho	Desvio de Obstáculos
3	Distância entre o robô e o alvo maior do que R_C e menor distância entre o robô e um vizinho ou um obstáculo é menor do que 1	Manutenção de Distância Mínima

da tarefa de escolta. O erro mais grave que pode ser cometido durante a escolta é a colisão entre um robô e a meta de escolta, pois isto prejudica a trajetória da mesma, afetando criticamente o desempenho da escolta independentemente do desempenho dos demais robôs. É importante salientar que o comportamento associado à primeira regra é o de manutenção de distâncias mínimas, pois este comportamento em si distancia um robô mais próximo da meta de escolta independentemente do alcance do sensor de feixes *laser*. O segundo erro mais grave é a colisão nos demais casos, porque ela prejudica o desempenho do próprio robô, apesar de não afetar criticamente a escolta como um todo por si. A gravidade da colisão com um robô vizinho é maior do que uma colisão com um obstáculo do ambiente, pois prejudica o desempenho de dois robôs, razão pela qual a mesma regra de distância adotada para evitar a colisão entre o robô e a meta de escolta é adotada. Por fim, o erro de menor gravidade é o distanciamento excessivo da meta de escolta quando há bastante espaço para navegação, pois apesar de prejudicar o desempenho da escolta em geral, não resulta em danos ao robô.

5.2.3 Testes com influências constantes

Para fins de controle de resultados do desempenho da escolta utilizando o conhecimento do AIS, alguns testes foram realizados com influências de comportamentos inatos constantes. Estas constantes foram escolhidas a partir de análise empírica, de modo a permitir uma identificação de desempenhos distintos em cada cenário. O valor constante é um indicativo da porcentagem de influência do comportamento da manutenção de distâncias mínimas. Uma constante de influência de comportamento igual a 0.8 significaria que o vetor de comportamento de manutenção de distâncias mínimas seria multiplicado por 0.8, enquanto o vetor do comportamento de cobertura de área seria multiplicado por 0.2 ($1 - 0.8$). Pode-se dizer que os testes apresentados nesta etapa do trabalho refletem a eficácia do método de escolta escolhido para este projeto.

Testes com constante = 0.51

Para o primeiro teste, uma constante que tende de forma levemente favorável para a manutenção de distâncias mínimas é utilizada, com o valor de 0.51.

O resultado das execuções do enxame na escolta no primeiro cenário após 20 simulações pode ser visto no gráfico da Figura 5.14. Observa-se que as colisões só iniciaram a partir do momento em que os robôs já circundavam a meta de escolta. A influência maior da manutenção de distâncias mínimas, mesmo que baixa, foi suficiente para resultar em colisões constantes entre os robôs. Identifica-se nesta bateria de testes a consequência observada anteriormente de uma influência maior deste comportamento. A distância média entre os robôs e a meta de escolta pode ser considerada sob uma análise superficial boa, mas não compensa o alto número de colisões em um cenário sem obstáculos fixos. Além disso, nota-se que há leves oscilações das médias de distância tanto dos robôs à meta de escolta quanto dos robôs e seus vizinhos, o que atribui-se à trajetória do agente que está sendo escoltado.

No segundo cenário de escolta (Figura 5.15), onde o alvo da escolta atravessa um corredor, nota-se que o número de colisões aumentou significativamente a partir do início da movimentação da meta de escolta. O número de colisões no início da travessia do corredor foi bastante elevado, chegando a picos de duas colisões em média. Ocorre uma diminuição

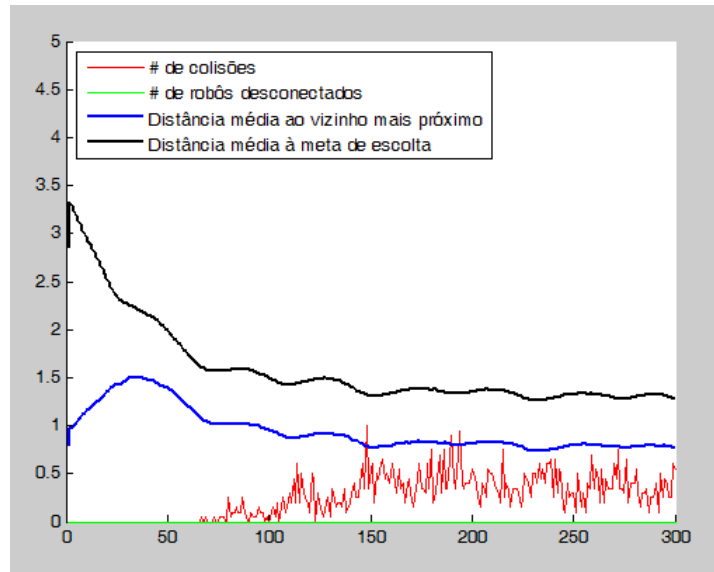


Figura 5.14: Primeiro cenário de teste de escolta com constante igual a 0.51.

considerável da média de colisões após a travessia do corredor. Apesar do elevado número médio de colisões, poucos robôs foram incapazes de completar a trajetória.

Observou-se, novamente, uma média baixa de distanciamento à meta de escolta. A média de desconexões mostra que, na maior parte dos cenários onde um robô foi incapaz de percorrer a trajetória, apenas um robô foi incapaz de completar o percurso, mas em alguns casos houve trava de mais de um robô. A média de distanciamento entre os próprios robôs, novamente baixa, indica que os mesmos ficaram excessivamente próximos entre si, o que indica que grande parte das colisões ocorreram entre os próprios robôs.

O resultado da escolta no terceiro cenário (Figura 5.16) mostra que, novamente, a chegada aos obstáculos resultou em um elevado número de colisões. Desta vez, a média de robôs incapazes de terminar o percurso foi elevada, chegando à média de um robô falhando em realizar a escolta por colisão a cada dois experimentos. As sucessivas colisões no final da trajetória continuaram ocorrendo. Os robôs quase sempre visaram atravessar o espaço entre as duas barras, sem a redistribuição dos robôs nos três espaços livres.

Testes com constante = 0.5

Considerando a alta taxa de colisão presente nos testes que utilizam a constante de influência igual a 0.51, os mesmos experimentos foram realizados com uma constante igual a 0.5 para observar o impacto de uma

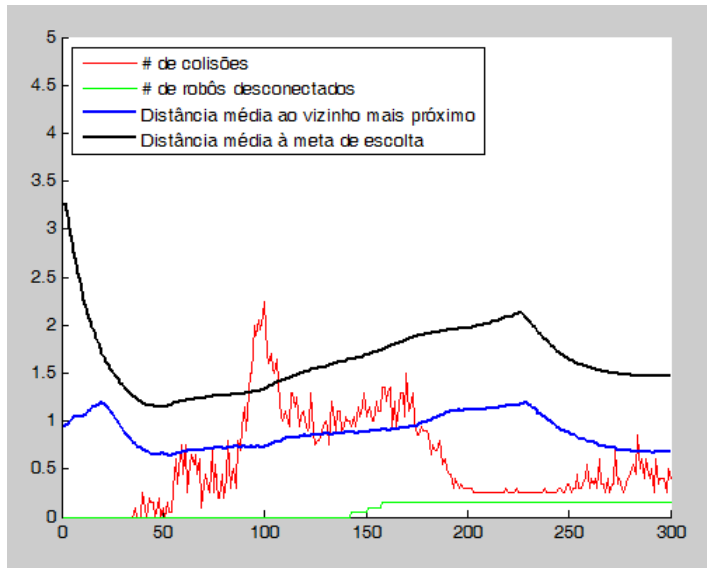


Figura 5.15: Segundo cenário de teste de escolta com constante igual a 0.51.

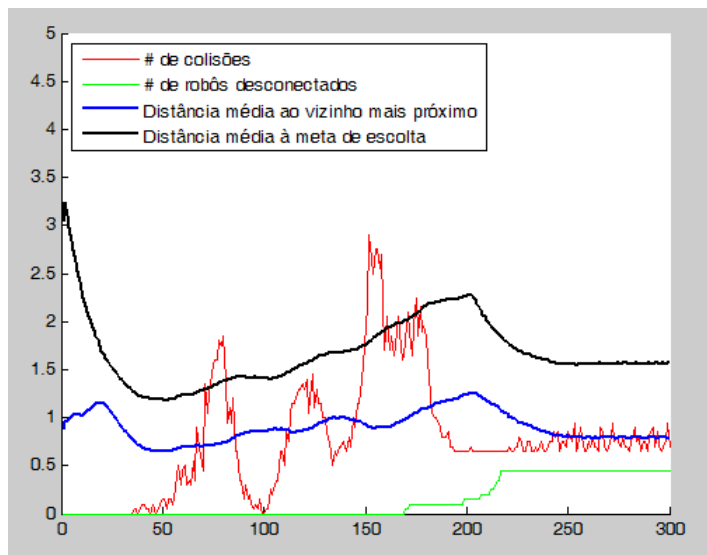


Figura 5.16: Terceiro cenário de teste de escolta com constante igual a 0.51.

mudança pequena na influência dos comportamentos no desempenho da escolta.

Nota-se nos resultados do primeiro cenário mostrados na Figura 5.17 que a média de colisões entre os robôs diminuiu substancialmente com a mudança de parâmetro, enquanto a média de distâncias cresceu. É possível verificar já no primeiro cenário de testes que a mudança no desempenho foi significativa. As oscilações das médias de distâncias, em comparação às vistas com os testes utilizando a constante igual a (Figura 5.14), tornaram-se maiores. É importante observar, também, que as colisões apenas iniciaram

quando as médias de distância diminuíram.

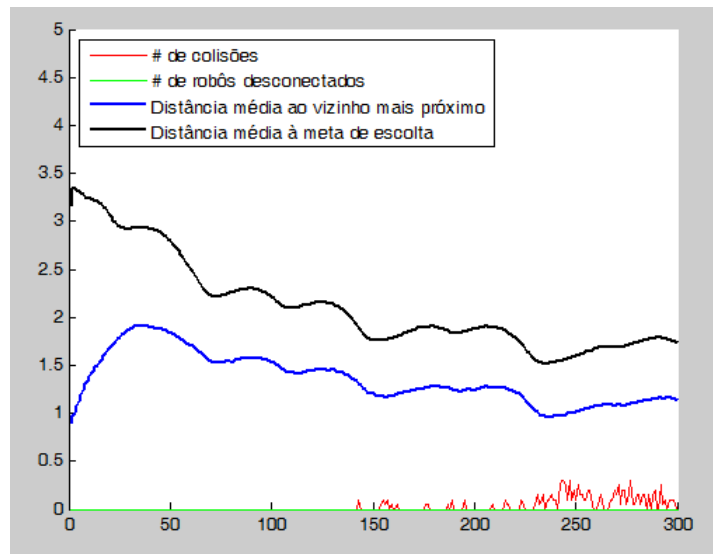


Figura 5.17: Primeiro cenário de teste de escolta com constante igual a 0.5.

Os resultados do segundo cenário de escolta (Figura 5.18) também indicam menores picos de média de colisão. Entretanto, o distanciamento médio entre os robôs e a meta de escolta foi bastante elevado, e um número maior de robôs foi incapaz de terminar o percurso.

Um crescimento da média de desconexão pode ser observado a partir de 200 segundos de simulação, e pode ser atribuído a robôs que se recuperaram de uma colisão na entrada do corredor e se uniram ao restante do enxame tardiamente, explicando a diminuição desta média à medida em que o tempo decorreu.

O resultado da escolta no terceiro cenário (Figura 5.19) indica que o ajuste de parâmetro resultou em uma melhora significativa do desempenho do enxame em um cenário que, apesar de possuir obstáculos, oferece espaço suficiente para o espalhamento dos robôs. Tanto o número de robôs capazes de percorrer o cenário quanto a média de colisões foram melhores. Os robôs visaram se distribuir nas três regiões delimitadas pelas duas barras paralelas para acompanhar a meta de escolta, razão pela qual a média de distanciamento entre os robôs e a meta de escolta apresentou um pico.

Testes com constante = 0.49

Concluindo os testes de uso de influências constantes para a ponderação dos comportamentos de cobertura de área e de manutenção de distância mínima, foram realizados testes em que a constante de influência é igual a

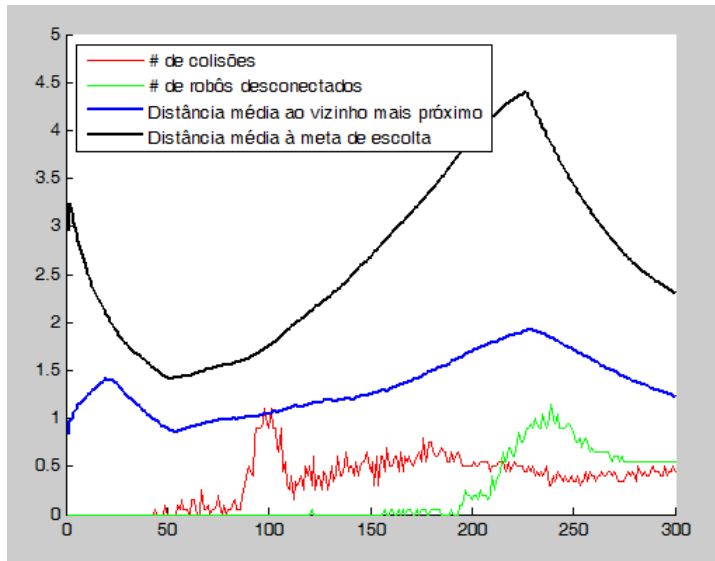


Figura 5.18: Segundo cenário de teste de escolta com constante igual a 0.5.

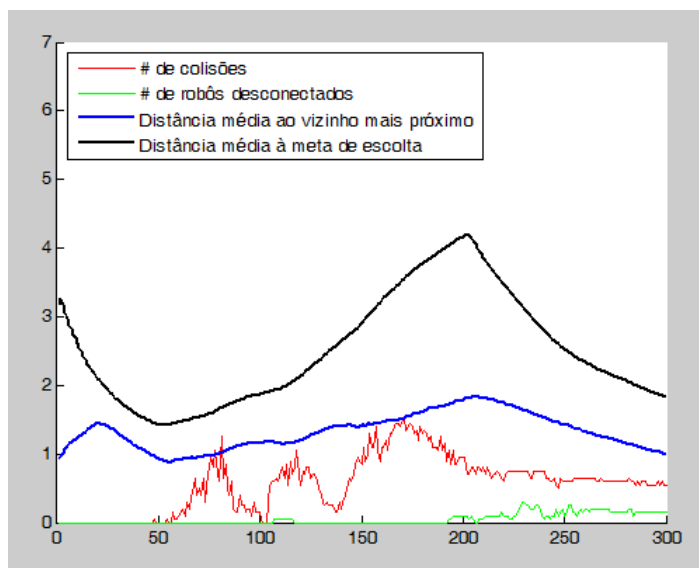


Figura 5.19: Terceiro cenário de teste de escolta com constante igual a 0.5.

0.49, ou seja, exercendo uma influência levemente maior do SLACS em relação à manutenção de distância mínima.

Durante o teste no primeiro cenário, livre de obstáculos (Figura 5.20), a escolta foi realizada sem a ocorrência de colisões. O distanciamento dos robôs em relação à meta de escolta, entretanto, aumentou substancialmente, assim como a média de distanciamento entre os robôs e seus vizinhos mais próximos. A oscilação da média de distância entre os robôs do enxame e a meta de escolta foi maior que a encontrada entre os próprios robôs, o que era esperado visto que a cobertura de área exercia maior influência

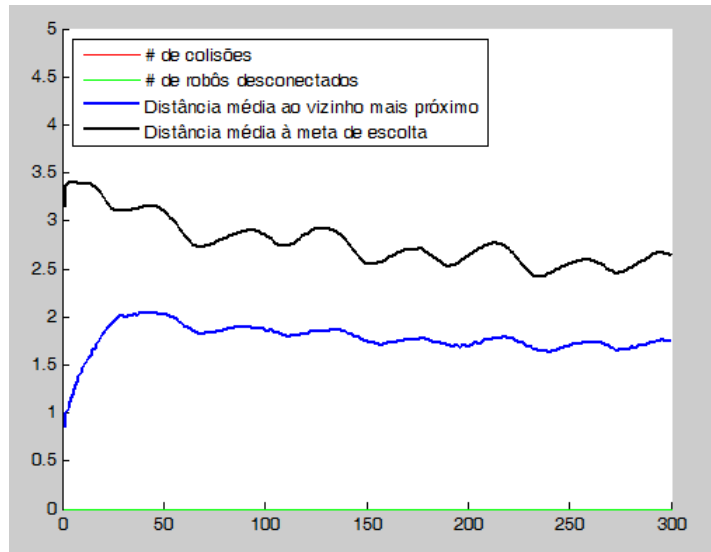


Figura 5.20: Primeiro cenário de teste de escolta com constante igual a 0.49.

Os resultados do segundo cenário de escolta (Figura 5.21) apresentaram médias de colisão significativamente menores. Entretanto, o distanciamento médio entre os robôs foi bastante elevado, e um número maior de robôs foi incapaz de terminar o percurso porque enfrentaram dificuldade em atravessar o corredor devido à maior influência do SLACS. O fato de a média de desconexões ter sido maior que a de colisões indica que a maior parte dos robôs incapazes de terminar o trajeto não estavam colidindo.

O aumento da média de desconexões no final da simulação também é notável, e ocorreu a partir do momento em que a maior parte dos robôs foi capaz de atravessar o corredor, deixando um ou mais robôs sem vizinhos em sua área de comunicação.

Os experimentos realizados no terceiro cenário (Figura 5.22) mostram mais uma vez uma média de colisões drasticamente reduzida, ao custo de uma média de distanciamento entre os robôs muito elevada. Muitos robôs foram incapazes de atravessar os obstáculos porque a prioridade do espalhamento maior foi suficiente para desencorajar os mesmos de avançar rapidamente.

Nota-se pelos resultados dos testes realizados que o desempenho do enxame na realização da escolta é altamente dependente do ambiente que será enfrentado e da influência dos comportamentos. Os resultados da escolta sem o uso de um mecanismo de aprendizado sugerem, portanto, que o desempenho da tarefa pode ser significativamente aprimorados adotando um método que varie a influência dos comportamentos, como o AIS. Os testes através do aprendizado são mostrados a seguir.

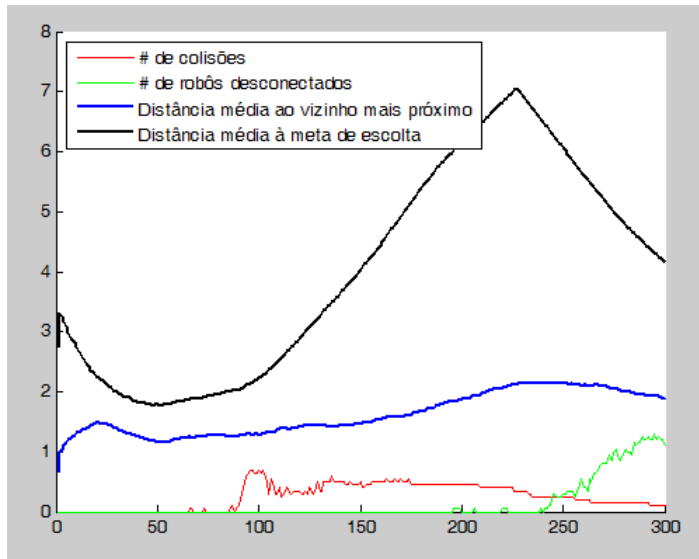


Figura 5.21: Segundo cenário de teste de escolta com constante igual a 0.49.

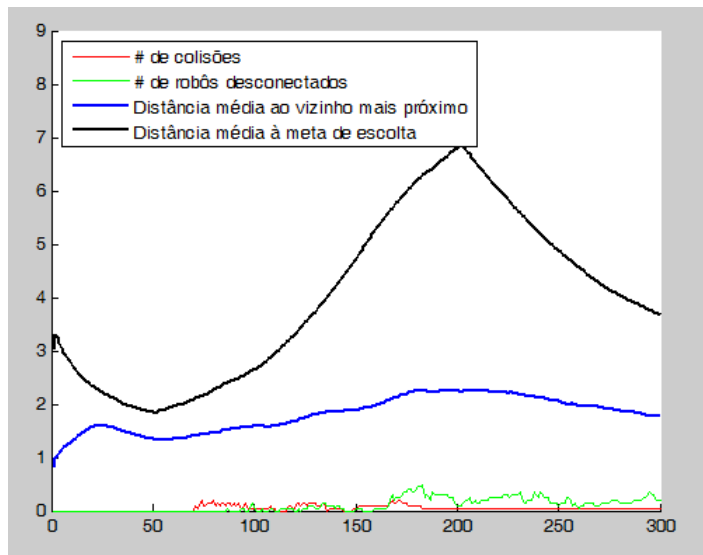


Figura 5.22: Terceiro cenário de teste de escolta com constante igual a 0.49.

5.2.4 Primeiro conjunto de testes de escolta utilizando a rede AIS

Após a mostra dos testes de uso de influências constantes, os testes nos quais o AIS é adotado são apresentados. Estes testes apresentam, portanto, os principais resultados desta Dissertação. Cada cenário de teste foi testado em duas etapas distintas, cada uma repetida vinte vezes, tal como nos testes adotando influências constantes. Primeiramente, avaliou-se o desempenho da escolta considerando um enxame cujos robôs não tenham conhecimento prévio armazenado, ou seja, sem neurônios no Módulo Coordenador. Em seguida, os robôs são testados reutilizando o conhecimento obtido

anteriormente, para verificar se de fato houve aprendizado. É importante salientar que, apesar de os cenários serem os mesmos na reutilização, as situações enfrentadas pelos robôs são na prática distintas em cada teste, sobretudo pelo efeito do ruído aplicado. Portanto, a reutilização do aprendizado não será aplicada para valores exatamente iguais. Cada robô individualmente passa, inclusive, por dificuldades distintas em muitos casos.

Os parâmetros que serão utilizados durante os testes são os apresentados na Tabela 5.2.4. Tais valores foram definidos de modo a gerar uma quantidade limitada de neurônios, com pouca variação nos valores de S para evitar que o AIS torne-se um ajustador pontual de parâmetros, e em que não haja uma demora significativa para a mudança de influência de comportamentos quando necessário.

Parâmetro	Descrição	Valor
σ	Suaviza as funções Gaussianas do cálculo de $\Delta(\Phi_{w,i}, \Omega_{x,i})$	0.8
Lim_{SIMIL}	Determina se um neurônio será inserido ou utilizado para atualização de W e S de outro neurônio	0.8
Lim_{DEL}	Determina se um neurônio será eliminado após sua atualização	0.3
α	Controla a variação de S	20
ζ	Controlam a variação dos pesos de influência de comportamento W	0.5
ϵ	Controla a intensidade de estímulo comportamental	0.75

Além dos gráficos de avaliação da escolta, serão apresentados gráficos que indicam o crescimento médio das redes neurais presentes no Módulo Coordenador. Esta informação é importante para acompanhar o processo de crescimento das redes do CM, indicando os momentos nos quais o aprendizado recorreu à inserção de neurônios. Uma análise deste dado em contraste a dados de número médio de colisões e distâncias entre os robôs e a meta de escolta ajudam a diferenciar entre eventos que resultaram no crescimento das redes neurais do CM e eventos que culminaram em ajustes em neurônios.

O resultado no primeiro cenário de escolta, vazio, é apresentado no gráfico da Figura 5.23. Mesmo sem conhecimento prévio, devido à escolha de regras que priorizam aspectos de segurança da escolta, os robôs realizam a escolta mantendo uma média de distância um pouco maior à vista com o uso de uma constante de influência fixa igual a 0.5 (Figura 5.17), e sem a ocorrência de

colisões. Foi observado que os robôs que se mantiveram na distância de escolta desejada de 1 metro mantiveram uma influência superior da manutenção de distâncias mínimas. Além disso, é possível observar que as médias de distâncias mantiveram-se estáveis a partir de 70 segundos de simulação, ou seja, o posicionamento resultante dos robôs não foi significativamente alterado pelas curvas realizadas pela meta de escolta.

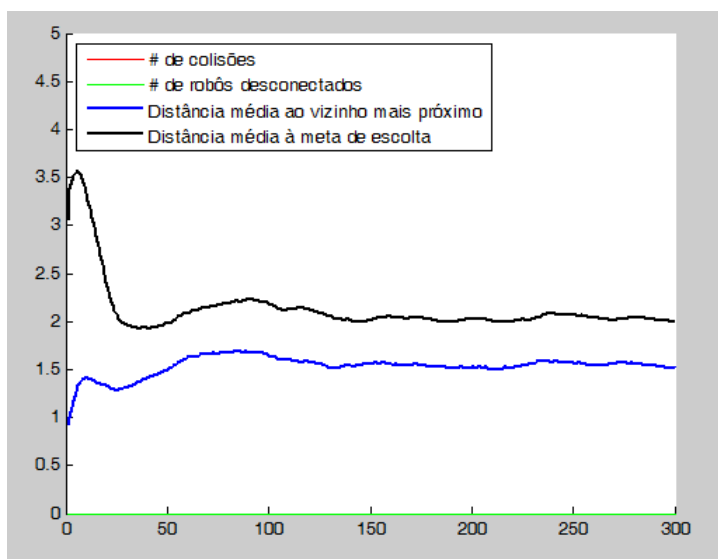


Figura 5.23: Primeiro cenário de teste de escolta com o uso da rede AIS.

O gráfico da Figura 5.24 mostra a média de crescimento de neurônios nos Módulos Coordenadores dos robôs. O crescimento da camada intermediária foi interrompido a partir de 70 segundos de simulação, ou seja, também estabilizou-se ao mesmo tempo em que o distanciamento relativo entre os robôs e sua meta, e também seus vizinhos mais próximos, se estabilizou.

São apresentados, agora, os resultados da reutilização de comportamentos para a escolta no cenário vazio a partir da Figura 5.25. É interessante observar que o desempenho médio foi, na prática, o mesmo, com exceção do tempo para estabilização da distância média à meta de escolta, que ocorreu antes dos 50 segundos de simulação. Isto deve-se ao aprendizado de avanço quando o robô está distante da meta de escolta ter sido estimulado nas simulações anteriores, levando os robôs a avançarem diretamente para a meta de escolta. Apesar do avanço mais rápido e de um período onde o distanciamento médio entre os robôs ter sido baixo durante o período, o estímulo para a manutenção de distâncias mínimas não provocou colisões.

A média de neurônios mostrada na Figura 5.26 indica que poucos neurônios associados à manutenção de distâncias mínimas foram inseridos,

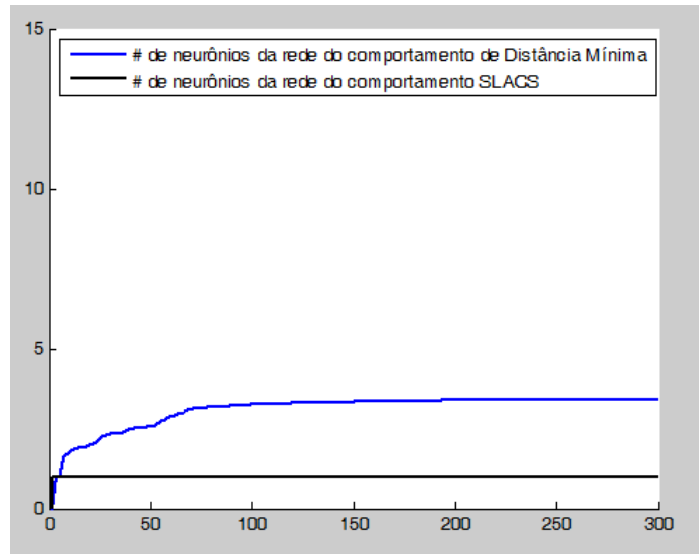


Figura 5.24: Crescimento das redes neurais do Módulo Coordenador - primeiro conjunto de parâmetros, cenário vazio.

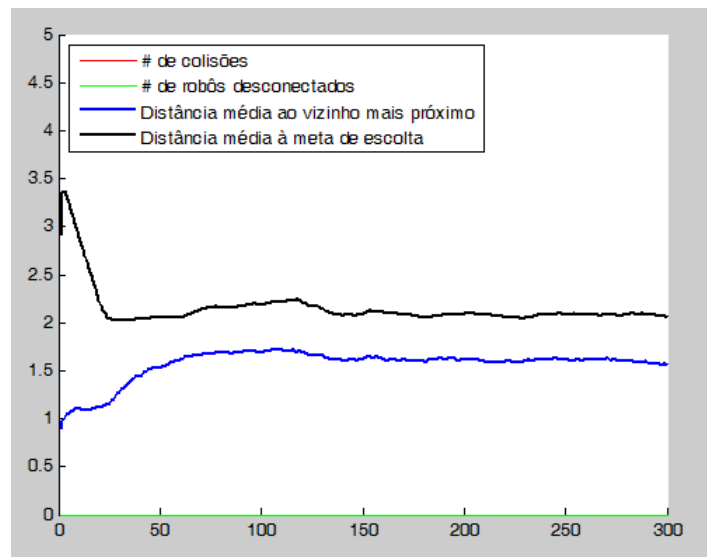


Figura 5.25: Primeiro cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido.

e apenas em algumas simulações. Este é um sinal de que não foi necessário produzir mais neurônios para que as regras definidoras de eventos deixassem de ser ativadas com frequência. Pode-se afirmar pelo desempenho da escolta e pelo crescimento do número de neurônios que o AIS foi capaz de aprender como escoltar neste cenário, mesmo sendo relativamente simples. A eficácia do AIS poderá ser melhor definida nos próximos cenários de escolta.

No segundo cenário de escolta, onde a meta de escolta atravessa um corredor, o desempenho da escolta pode ser analisado na Figura 5.27. Uma

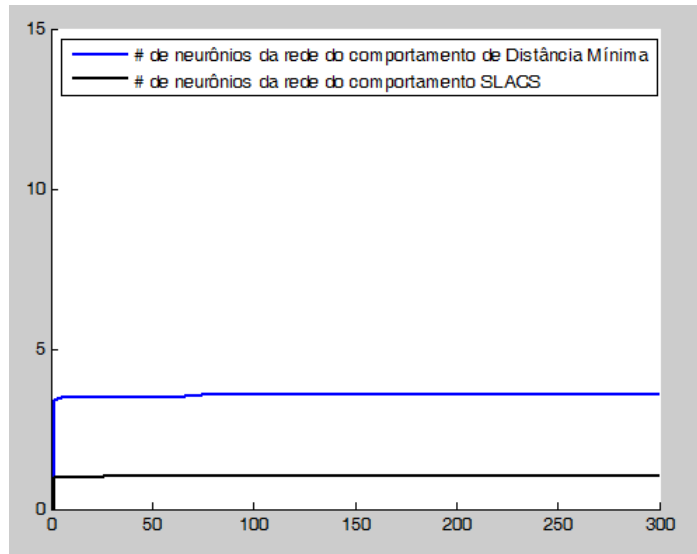


Figura 5.26: Crescimento das redes neurais do Módulo Coordenador - primeiro conjunto de parâmetros, cenário vazio, reutilização de aprendizado.

observação inicial interessante é que, em alguns testes, um robô chega a colidir na meta de escolta enquanto a mesma está parada, recuperando-se antes que os 50 segundos iniciais se completem e a navegação da meta de escolta inicie. A provável causa da colisão é o grande estímulo inicial à execução do SLACS que ocorre inicialmente devido à proximidade entre os robôs. A distribuição esparsa dos sensores laser mostrou dificuldade para identificar a meta de escolta, o que não permitiu o desvio realizado pelo SLACS. O robô se afasta em seguida da meta de escolta porque o comportamento de manutenção de distâncias mínimas começa a se fortalecer. No momento no qual a meta de escolta inicia a travessia do corredor, há um pico da média de colisões, o que era esperado considerando que os robôs ainda não haviam obtido experiência de colisão com obstáculos. A recuperação dos robôs da situação de colisão foi, em geral, rápida. Em todas as simulações, apenas dois robôs não foram capazes de terminar o percurso: um robô que colidiu na parte superior da entrada do corredor e seu vizinho. O posicionamento do robô que estava colidindo entre o corredor e seu vizinho impossibilitou que este robô terminasse o trajeto. Uma observação interessante é que a influência de comportamentos do robô que estava colidindo foi equivalente ao uso de uma influência constante de 0.2, que priorizaria muito o desvio de obstáculos, ou seja, não foi um problema causado por incapacidade da AIS. De modo geral, o desempenho da escolta foi satisfatório, especialmente em comparação às influências constantes, que

apresentaram resultados com um número de colisões significativamente maior, com exceção do uso da constante de influência igual a 0.49, onde a alta influência do SLACS evitou que muitos robôs fossem sequer capazes de iniciar a travessia do corredor.

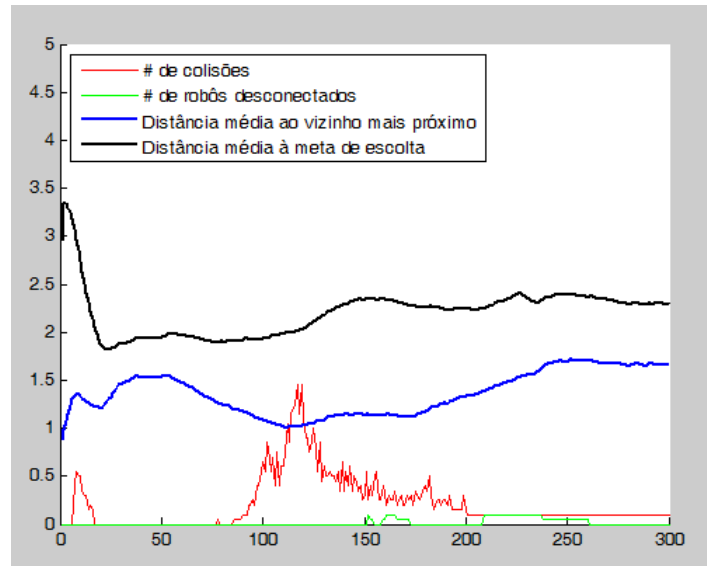


Figura 5.27: Segundo cenário de teste de escolta com o uso da rede AIS.

A Figura 5.28, correspondente ao crescimento médio do CM, indica que o crescimento dos neurônios antes do início da travessia do corredor foi bastante similar ao apresentado no primeiro cenário de escolta (Figura 5.24). A partir do momento em que a travessia do corredor se iniciou, como era esperado, foram inseridos vários neurônios relacionados ao evento de colisão. Um afastamento dos robôs após a travessia do corredor em relação à meta de escolta em um momento em que os mesmos encontram-se fora do mesmo causou um leve aumento no número de neurônios associados à manutenção de distâncias mínimas.

A reutilização do aprendizado no cenário do corredor (Figura 5.29) mostrou um desempenho significativamente maior. Apenas um robô em todas as simuações colidiu brevemente com a meta de escolta, o pico da média de colisões no início da travessia do corredor foi substancialmente menor, e o número de colisões durante a travessia do corredor foi muito baixo. É interessante observar que a melhora do desempenho ocorre ao mesmo tempo em que o distanciamento médio entre os robôs e a meta de escolta sobe durante a passagem pelo corredor, mas sem um aumento significativo da média de distâncias entre os próprios robôs e seus vizinhos mais próximos. Isto indica que os robôs entraram no corredor com maior

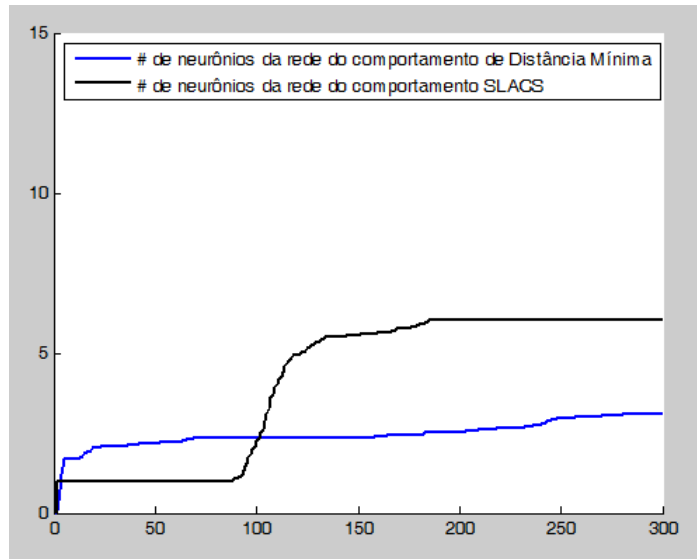


Figura 5.28: Crescimento das redes neurais do Módulo Coordenador - segundo cenário.

cautela, mas sem prejudicar a coesão entre os próprios robôs.

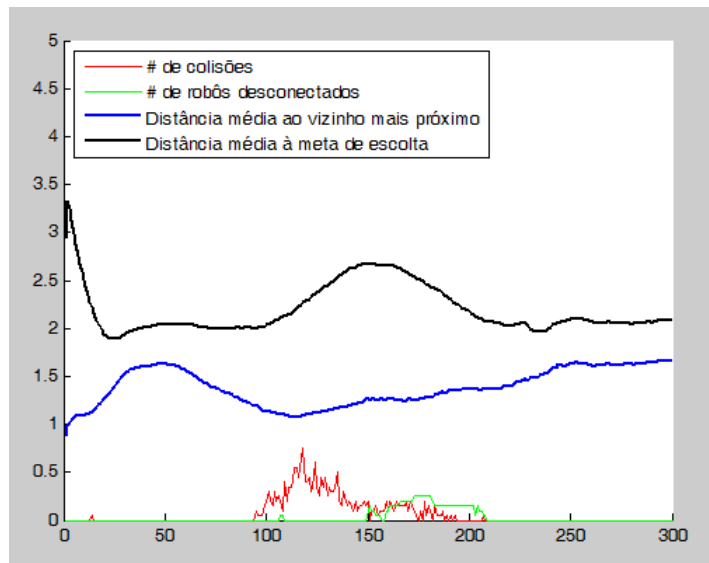


Figura 5.29: Segundo cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido.

A média de neurônios mostrada na Figura 5.30 mostra, curiosamente, um aumento significativo no número de neurônios associados ao desvio de obstáculos na entrada do corredor, mesmo que menor comparado ao visto antes da reutilização do aprendizado. É possível que os robôs tenham colidido em situações diferentes, como colisões na parte de baixo do corredor, cujo distanciamento é identificado por um conjunto de sensores de feixes

laser diferente das colisões com a parte de cima do corredor, onde a grande maioria das colisões ocorreu nos testes sem aprendizado prévio. O fenômeno do acréscimo de neurônios associados à manutenção de distâncias mínimas é razoável, visto que muitas vezes nas quais isto acontece um robô insere sem contato com muitos robôs vizinhos por estarem distantes de seu alcance de comunicação. Então, quando um robô diferente encontra-se nesse cenário, é possível que ele não tenha recebido o neurônio correspondente. De modo geral, entretanto, o crescimento das camadas intermediárias do CM foi substancialmente menor, e o desempenho foi melhor, o que sugere com maior força que o aprendizado através do AIS ocorreu.

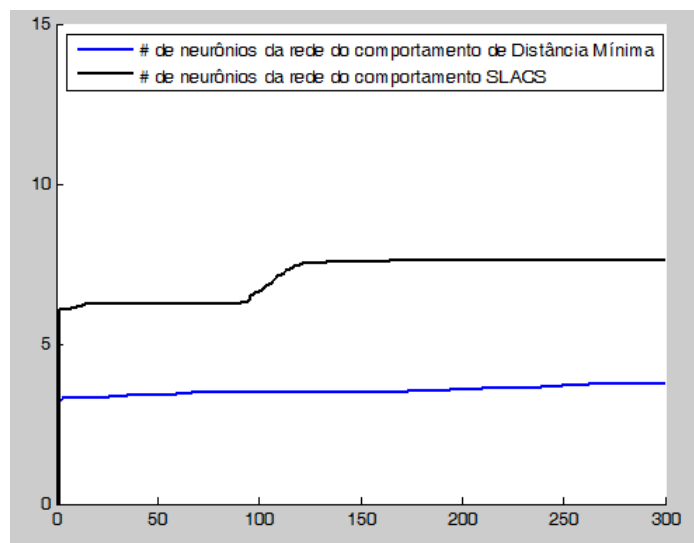


Figura 5.30: Crescimento das redes neurais do Módulo Coordenador - segundo cenário, reutilização de aprendizado.

Por fim, foram realizados os testes da escolta utilizando o AIS no terceiro cenário de testes, onde um número mais distinto de obstáculos é apresentado. O desempenho dos robôs pode ser analisado através da Figura 5.31. O fenômeno das colisões com a meta de escolta observado no segundo cenário repetiu-se em alguns poucos experimentos. A média de colisões, especialmente durante a travessia do trecho com duas barras paralelas, foi elevada em comparação ao obtido com o uso de influência de comportamentos fixa de 0.5 (Figura 5.19), mas todos os robôs foram capazes de se recuperar das mesmas e percorrer todo o trajeto. Além disso, vê-se que os robôs não se distanciaram muito. Apesar do número de colisões e do distanciamento, em alguns casos, os robôs procuraram navegar nas três regiões divididas pelas barras paralelas.

A Figura 5.32 indica o aumento de neurônios no CM nestes experimentos.

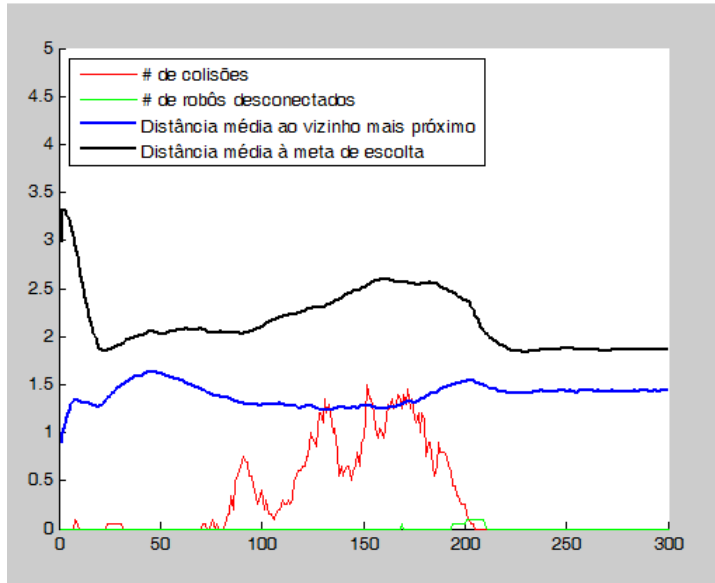


Figura 5.31: Terceiro cenário de teste de escolta com o uso da rede AIS.

Como nos demais cenários, apenas um neurônio associado ao SLACS foi inserido durante os 50 segundos iniciais de simulação, enquanto o crescimento foi mais acentuado na rede associada à manutenção de distância mínima à meta de escolta. Houve um crescimento constante de neurônios entre 100 e 200 segundos de simulação, o que é justificável não somente pelo número de colisões como pela diferença de situações de colisão: as condições nas quais cada colisão aconteceu foi diferente.

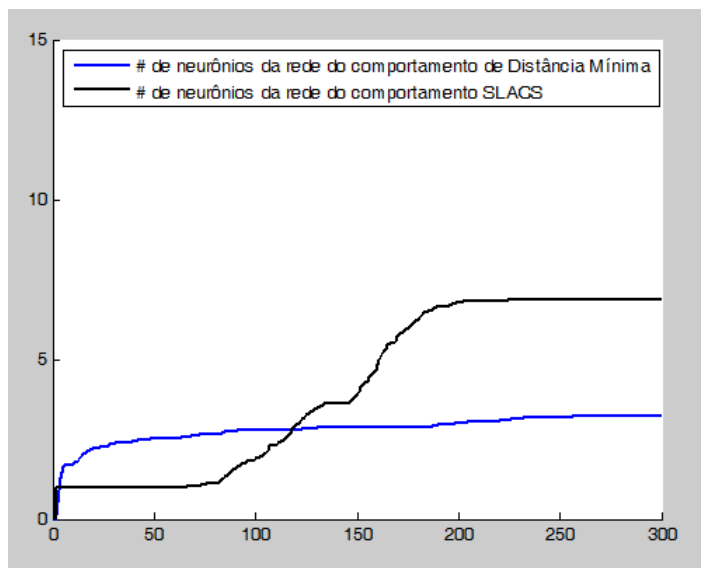


Figura 5.32: Crescimento das redes neurais do Módulo Coordenador - terceiro cenário.

A reutilização do aprendizado neste cenário (Figura 5.34) indica um desempenho melhor em relação ao número de colisões, em especial nos dois primeiros obstáculos, o que indica que o deslocamento dos robôs para lidar com a ocorrência seguida de obstáculos nas porções superior e inferior foi aprendido. A média de colisões durante a travessia dos obstáculos em forma de duas barras paralelas também foi melhor, mantendo uma média de distanciamento entre os robôs e a meta de escolta ligeiramente maior. Os robôs apresentaram maior facilidade de percorrer as três regiões navegáveis entre as barras. Em um dos cenários, dois robôs vieram a colidir nas pontas de uma das barras e foram incapazes de se recuperar, apesar da rede AIS ter indicado a ambos um estímulo elevado ao SLACS, o que é um indicativo de que a trava dos robôs não foi responsabilidade de deficiências de aprendizado, mas de outras particularidades, como o SLACS ou a dificuldade de detecção de obstáculos (Figura ??). Apesar do desempenho da reutilização não ter sido tão significativo em comparação ao observado no segundo cenário (comparar Figuras 5.27 e 5.34), deve-se levar em consideração a maior complexidade do ambiente enfrentado no terceiro cenário, em especial as barras paralelas, que tem reconhecimento pelos feixes *laser* dificultado. Na posição na qual os robôs travaram no teste onde isto ocorreu, ambos não enxergaram obstáculos logo à sua frente, o que dificultou que o SLACS recuperasse-os desta colisão.

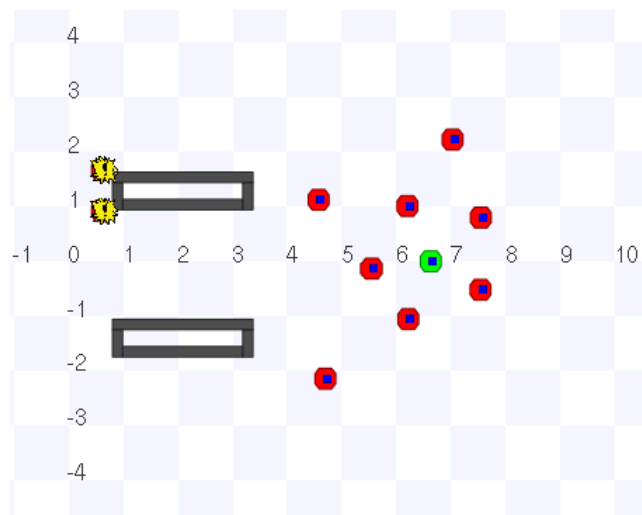


Figura 5.33: Terceiro cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido.

A média de neurônios mostrada na Figura 5.35 indica um crescimento mínimo da rede do CM associada ao SLACS, mas manteve-se parecida,

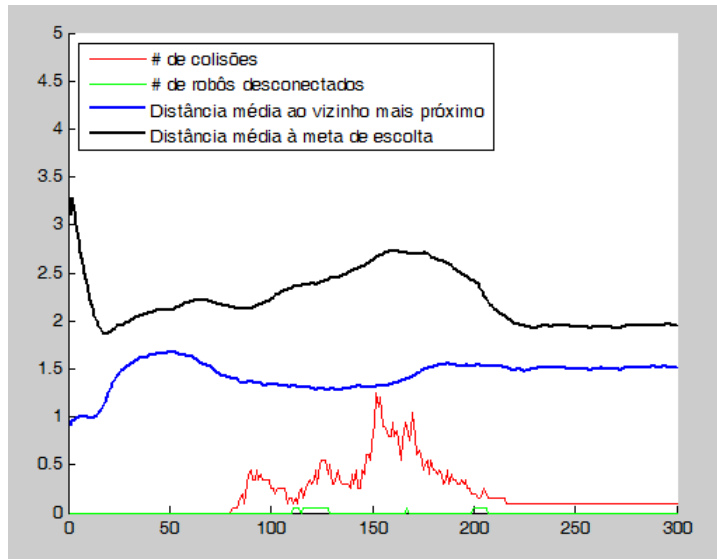


Figura 5.34: Terceiro cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido.

indicando estabilidade do aprendizado.

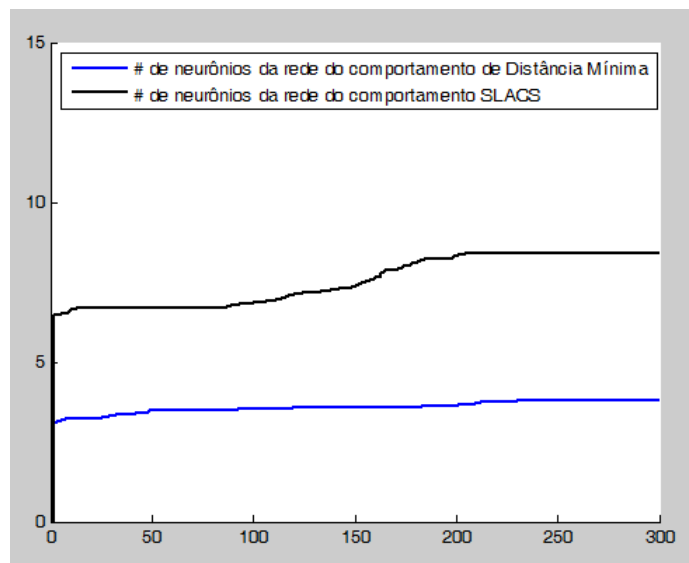


Figura 5.35: Crescimento das redes neurais do Módulo Coordenador - terceiro cenário, reutilização de aprendizado.

Considerando os resultados apresentados, pode-se dizer que há a ocorrência de aprendizado. Os testes a seguir investigam o desempenho da rede AIS permitindo a inserção de um maior número de neurônios.

5.2.5 Segundo conjunto de testes de escolta utilizando a rede AIS

No segundo conjunto de testes utilizando a rede AIS, é verificada a mudança no desempenho a partir da mudança do aumento do limiar de similaridade Lim_{SIMIL} em 0.1, ou seja, considerando $Lim_{SIMIL} = 0.9$. O objetivo é verificar se é vantajoso permitir um crescimento maior das redes do Módulo Coordenador a despeito do maior armazenamento e exigência de maior processamento que é resultante de tal mudança. A atualização dos pesos dos neurônios (W , S) é, por consequência, mais difícil de ocorrer. Os mesmos testes serão realizados.

O desempenho médio dos testes realizados no ambiente vazio sem reutilização de aprendizado são vistos na Figura 5.36. O desempenho mostrou-se muito parecido com o visto nos testes onde $Lim_{SIMIL} = 0.8$ (Figura 5.17).

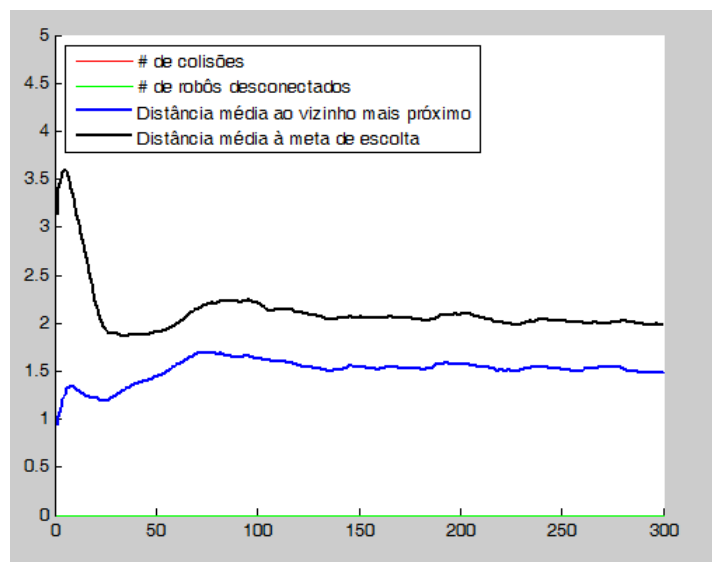


Figura 5.36: Primeiro cenário de teste de escolta com o uso da rede AIS, com $Lim_{SIMIL} = 0.9$.

O gráfico da Figura 5.37 mostra a média de crescimento de neurônios nos Módulos Coordenadores dos robôs. Nota-se aqui que o crescimento foi bastante distinto do apresentado nos testes anteriores, com um aumento no número de neurônios associados ao SLACS. É provável que neurônios extras tenham sido gerados a partir da detecção de algum robô vizinho pelos feixes *laser*, ou ainda pelo efeito do ruído. Os principais fenômenos de inserção de neurônios foram, novamente, o distanciamento excessivo da meta de escolta e a proximidade com robôs vizinhos.

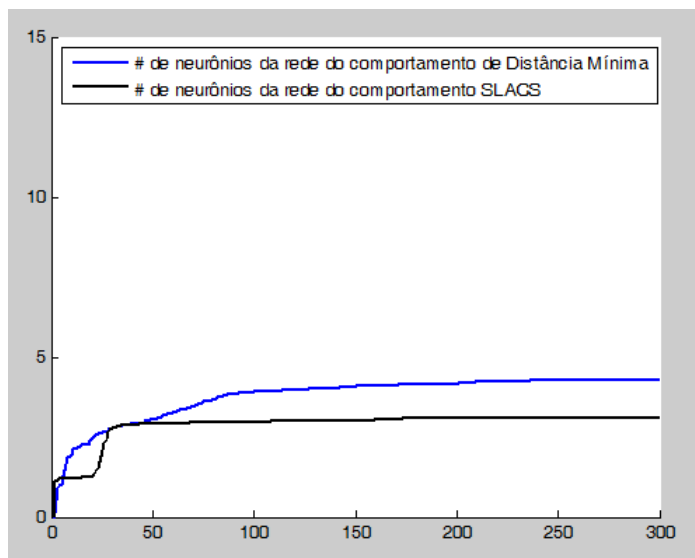


Figura 5.37: Crescimento das redes neurais do Módulo Coordenador - primeiro conjunto de parâmetros, cenário vazio, com $Lim_{SIMIL} = 0.9$.

A reutilização do aprendizado adquirido no cenário vazio foi avaliada a seguir, a partir da Figura 5.25. Os resultados foram bastante parecidos, mais uma vez, com os obtidos com $Lim_{SIMIL} = 0.8$. Por se tratar de um cenário que apresenta poucas dificuldades para o posicionamento criado naturalmente entre os robôs do enxame, não foi possível verificar diferenças significativas de desempenho.

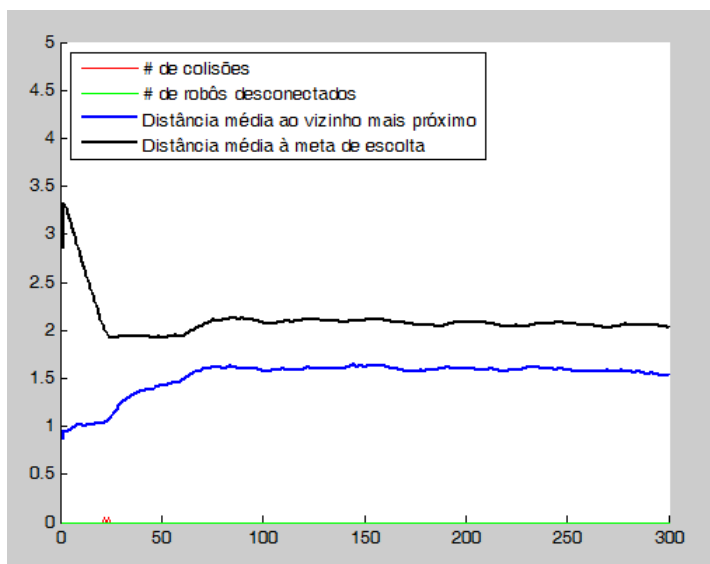


Figura 5.38: Primeiro cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido, com $Lim_{SIMIL} = 0.9$.

A média de neurônios mostrada na Figura 5.39 indica que, mesmo com

um desempenho similar, o crescimento neuronal foi bastante diferente. A média de neurônios associados ao desvio de obstáculos chega a dobrar, enquanto a média de neurônios na rede do CM associada à manutenção de distâncias mínimas mantém-se parecida. Os resultados sugerem, portanto, que o aumento no número de neurônios na rede associada ao SLACS será ainda maior nos próximos testes, onde obstáculos fixos, além dos demais robôs e da meta de escolta, estão presentes.

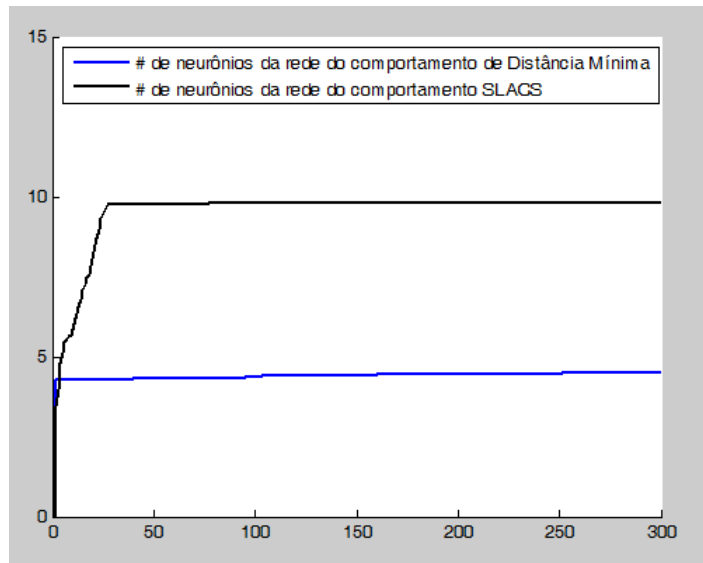


Figura 5.39: Crescimento das redes neurais do Módulo Coordenador - primeiro conjunto de parâmetros, cenário vazio, reutilização de aprendizado, com $Lim_{SIMIL} = 0.9$.

São apresentados, a seguir, os testes no segundo cenário de escolta, onde um corredor deve ser atravessado pelos robôs em busca da meta de escolta. O desempenho pode ser visto na Figura 5.40. O resultado, de modo geral, foi bastante parecido com o anterior. As diferenças estão no aumento do número de colisões durante a saída do corredor, possivelmente causada pela pouca generalização da representação de medidas de distanciamento em casos de colisão feita pelos neurônios do CM correspondente. O distanciamento entre os robôs também foi maior, devido ao maior tempo necessário para a recuperação de um robô que esteja preso na entrada do corredor.

Na Figura 5.41, que corresponde ao número médio de neurônios nas redes do CM, pode-se notar uma maior dificuldade para representar situações de risco adequadamente. Não somente o número de neurônios na rede neural associada ao desvio de obstáculos foi muito maior como cresceu por praticamente todo o tempo de travessia do corredor.

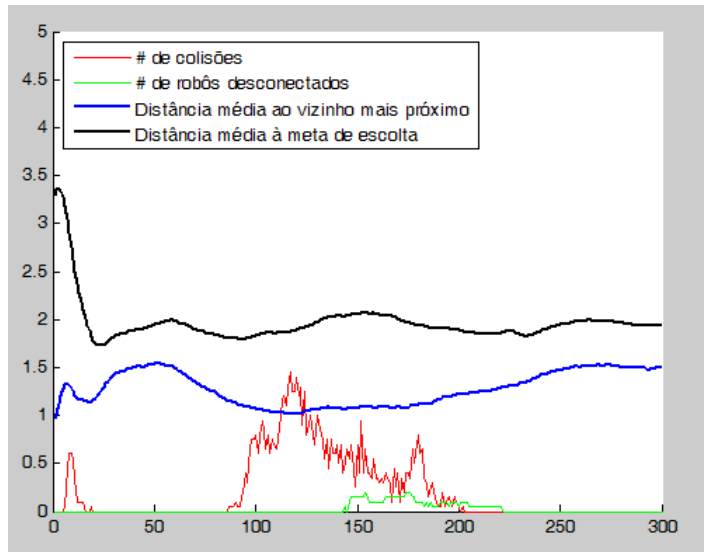


Figura 5.40: Segundo cenário de teste de escolta com o uso da rede AIS, com $Lim_{SIMIL} = 0.9$.

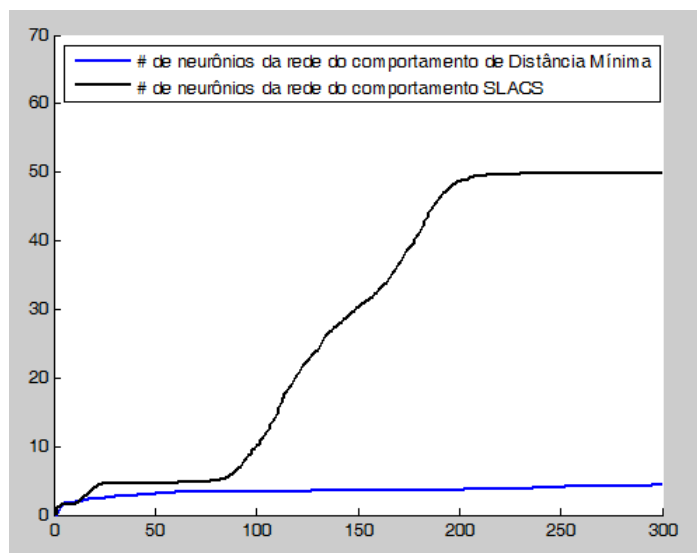


Figura 5.41: Crescimento das redes neurais do Módulo Coordenador - segundo cenário, com $Lim_{SIMIL} = 0.9$.

A reutilização do aprendizado no cenário do corredor (Figura 5.42) apresentou uma melhora no desempenho, em especial na etapa de saída do corredor, que ocorre entre 150 e 200 segundos de simulação, com $Lim_{SIMIL} = 0.8$ (Figura 5.29).

A média de neurônios mostrada na Figura 5.43 mostra que o crescimento da rede continuou durante a entrada no corredor, mas estabilizou-se a seguir, o que era esperado vista a redução de colisões durante a saída do corredor. O número de neurônios associados à manutenção de distâncias manteve-se

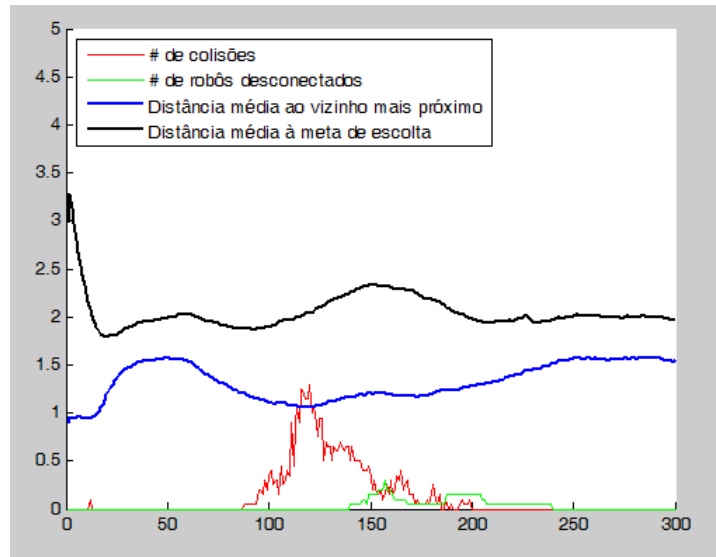


Figura 5.42: Segundo cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido, com $Lim_{SIMIL} = 0.9$.

novamente estável.

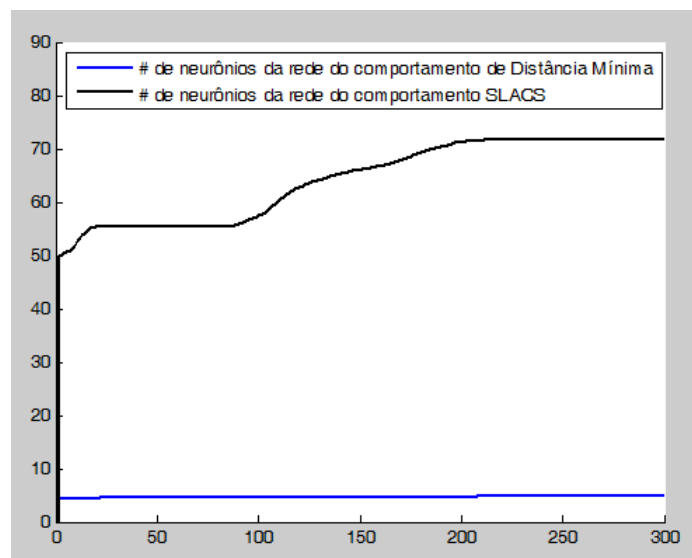


Figura 5.43: Crescimento das redes neurais do Módulo Coordenador - segundo cenário, reutilização de aprendizado, com $Lim_{SIMIL} = 0.9$.

Os testes com $Lim_{SIMIL} = 0.9$ no terceiro cenário de escolta são apresentados a seguir, a partir da Figura 5.44, e indicam, novamente, uma dificuldade inicial em generalizar cenários de colisão. A grande diferença de desempenho em relação ao mesmo experimento, apresentado na Figura 5.31, foi na situação de obstáculos mais tardia, onde neste caso estão as barras paralelas. É importante salientar que a diferença entre o distanciamento

médio entre os robôs e a meta de escolta foi parecida, mas o crescimento da distância foi mais abrupto. Este distanciamento deve-se, portanto, a robôs que apresentaram maior dificuldade para atravessar as barras paralelas, hipótese reforçada pela diferença significativa da média de colisões após 200 segundos de simulação, quando a meta de escolta está parada. Alguns robôs não foram capazes de terminar o percurso devido a colisões, em uma condição igual à vista em outros testes: a rede AIS sinaliza uma influência drasticamente maior do SLACS, mas a diferença ainda não se mostrou suficiente.

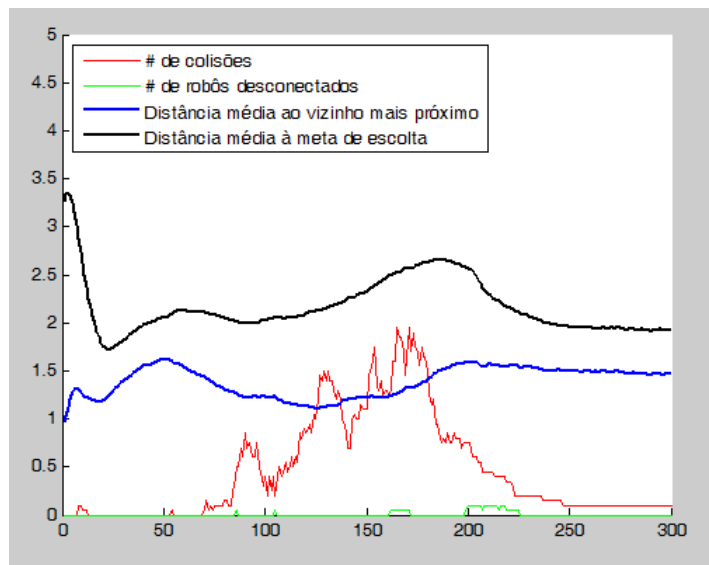


Figura 5.44: Terceiro cenário de teste de escolta com o uso da rede AIS, com $Lim_{SIMIL} = 0.9$.

A Figura 5.45 indica o aumento de neurônios nas redes do CM nestes experimentos. O crescimento foi, como esperado, maior durante a navegação da meta de escolta, o número de neurônios da rede do CM associada ao SLACS foi maior, e o crescimento da rede associada à manutenção de distâncias mínimas manteve-se o mesmo.

A reutilização do aprendizado neste cenário (Figura 5.46), adotando $Lim_{SIMIL} = 0.9$, indicou desta vez um desempenho significativamente superior, obtendo uma média de colisões consideravelmente menor, com um desempenho comparável ao obtido com $Lim_{SIMIL} = 0.8$. O contraste do desempenho em relação ao apresentado na reutilização do aprendizado adquirido no segundo cenário de escolta, com menor complexidade de obstáculos, sugere que é vantajoso que as redes neurais do CM possuam um número maior de neurônios.

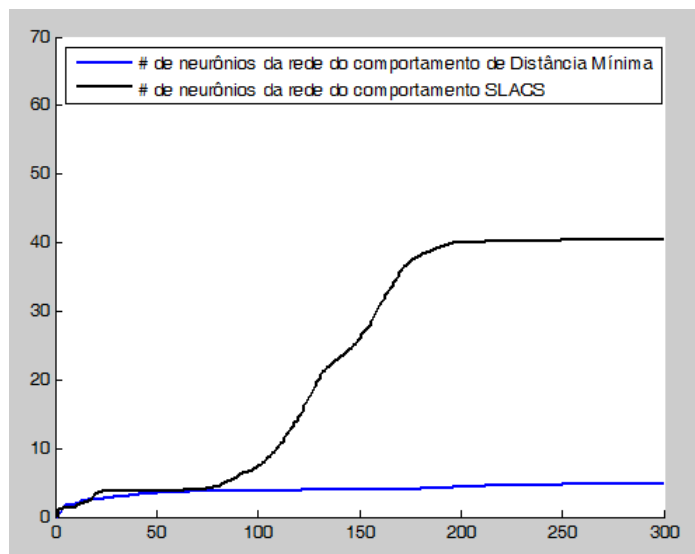


Figura 5.45: Crescimento das redes neurais do Módulo Coordenador - terceiro cenário, com $Lim_{SIMIL} = 0.9$.

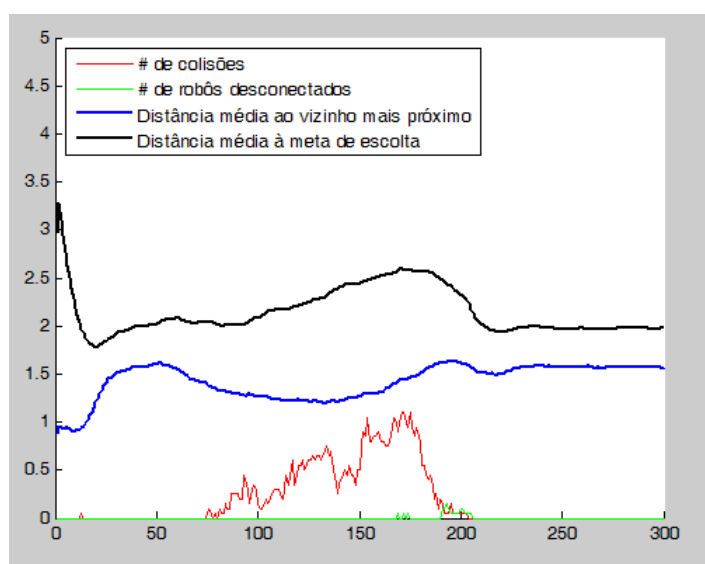


Figura 5.46: Terceiro cenário de teste de escolta com o uso da rede AIS, reutilizando conhecimento previamente adquirido, com $Lim_{SIMIL} = 0.9$.

O mesmo padrão de crescimento do número de neurônios visto em testes anteriores é apresentado neste experimento (Figura 5.47), com o número de neurônios associados ao comportamento de manutenção de distâncias mínimas estável e um crescimento do número de neurônios associados ao SLACS.

Pode-se ver que, de modo geral, o desempenho da aplicação da rede AIS para moderar os comportamentos que resultam na realização da escolta, que é um comportamento de maior complexidade, foi superior ao obtido com a

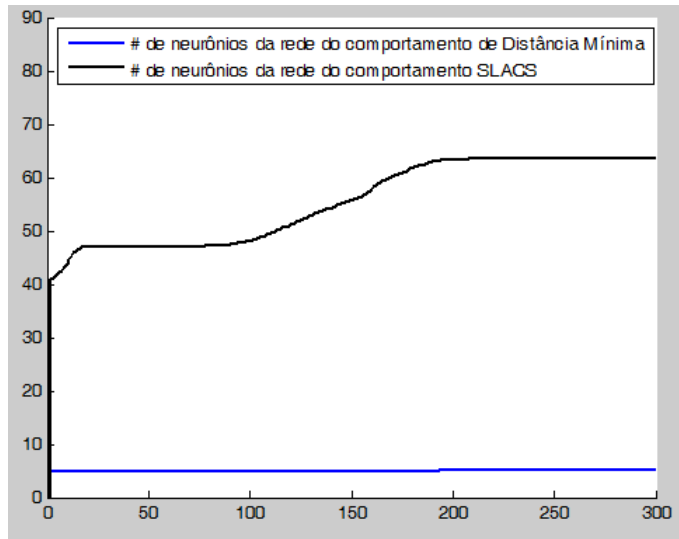


Figura 5.47: Crescimento das redes neurais do Módulo Coordenador - terceiro cenário, reutilização de aprendizado, com $Lim_{SIMIL} = 0.9$.

adoção de constantes de influência de comportamento fixas. Ainda, o desempenho reutilizando o conhecimento adquirido foi, em todos os experimentos, muito parecido ou superior.

Para fins de comparação numérica, são apresentadas as Tabelas 5.1, 5.2 e 5.3, que apresentam as médias totais dos parâmetros para cada conjunto de experimentos. M_m corresponde à média de distâncias até a meta, M_v corresponde à média de distâncias ao vizinho mais próximo, Col corresponde à média de colisões e $Desc$ à média de desconexões.

Tabela 5.1: Resultados da escolha no Cenário 1

Cenário 1	Cte = 0.51	Cte = 0.5	Cte = 0.49	Lim _{STMIL} = 0.8	Lim _{STMIL} = 0.8, <i>reuso</i>	Lim _{STMIL} = 0.9	Lim _{STMIL} = 0.9, <i>reuso</i>
M_m	1.5922	2.1190	2.7767	2.1284	2.1496	2.1156	2.0990
M_v	0.9477	1.3547	1.7753	1.5323	1.5657	1.5109	1.5161
Col	0.2522	0.0315	0	0	0	0	0.0003
$Desc$	0	0	0	0	0	0	0

Tabela 5.2: Resultados da escolha no Cenário 2

Cenário 2	Cte = 0.51	Cte = 0.5	Cte = 0.49	Lim _{STMIL} = 0.8	Lim _{STMIL} = 0.8, <i>reuso</i>	Lim _{STMIL} = 0.9	Lim _{STMIL} = 0.9, <i>reuso</i>
M_m	1.5258	2.6511	4.0110	2.2013	2.1973	1.9636	2.0486
M_v	0.8721	1.3379	1.6185	1.3579	1.3801	1.2734	1.3230
Col	0.3450	0.3612	0.2550	0.2162	0.0675	0.2172	0.1352
$Desc$	0.0275	0.2230	0.1485	0.0173	0.0300	0.0260	0.0300

Tabela 5.3: Resultados da escolha no Cenário 3

Cenário 3	Cte = 0.51	Cte = 0.5	Cte = 0.49	Lim _{STMIL} = 0.8	Lim _{STMIL} = 0.8, <i>reuso</i>	Lim _{STMIL} = 0.9	Lim _{STMIL} = 0.9, <i>reuso</i>
M_m	1.6804	2.5121	4.0133	2.1578	2.2312	2.1708	2.1491
M_v	0.9038	1.3245	1.8039	1.3915	1.4368	1.3851	1.4222
Col	0.7622	0.5660	0.0545	0.3008	0.2107	0.4658	0.1942
$Desc$	0.1477	0.0525	0.1118	0.0045	0.0042	0.0093	0.0042

Na próxima Seção, uma discussão sobre os resultados obtidos baseados nos experimentos realizados será apresentada.

5.3 Discussão

Em face dos resultados obtidos, pode-se dizer que o método de escolta proposto neste trabalho é adequado para a realização desta tarefa. Apesar de ser um problema de navegação de múltiplos robôs pouco explorado, a escolta pode ser aplicada para captura de objetos, vigilância, monitoramento e rastreamento de alvos. Na abordagem proposta, supõe-se que os robôs tenham a capacidade de se comunicar entre si. Entretanto, é comum que enxames de robôs não possuam um método de comunicação ativa. Este trabalho considerou, de fato, que os robôs do enxame possuem uma capacidade de processamento razoável, com maior foco em manter um pouco das características gerais apontadas como essenciais para um enxame de robôs: robustez, flexibilidade e escalabilidade (Şahin, 2005).

O método de cobertura de área desenvolvido neste trabalho, o SLACS, respeita estas três propriedades. A atuação para cobertura de área tem boa tolerância a falhas, consegue agir em cenários diversos com bom desempenho e pode ser executada com um número indeterminado de robôs. Apesar do teste ter considerado um sensor que não condiz com a filosofia de baixo custo da Robótica de Enxames (que é o sensor de feixes *laser*), o método pode ser ajustado para qualquer instrumento ou método medidor de distância. É possível, inclusive, que uma adaptação de sensores como sonares, em que a verificação possui um alcance em forma de cone ao invés do formato linear dos feixes *laser*, não enfrentasse os problemas de recuperação com colisões em cantos de obstáculos que foram identificadas. O teste é bastante rígido com relação a limites de comunicação se considerados robôs reais, o que é um fator que deve ser levado em conta para uma análise adequada do SLACS.

É evidente a melhora da robustez no enxame quando o AIS é aplicado, em primeira instância, porque os robôs tornam-se capazes de se recuperar de situações nas quais o uso de influências constantes de comportamento não podem. A AIS, quando corretamente aplicada, não interfere na flexibilidade, e tampouco na escalabilidade, visto que a transmissão de conhecimento não é absolutamente necessária para o desempenho do enxame e nem depende de um sistema global. O aprendizado não chega a depender de comunicação

para ocorrer, mas a tem como um complemento. Este sistema de aprendizado teria de ser adaptado, entretanto, no caso de um sistema de robôs heterogêneos, onde cada robô pode ter forma, sensores e atuadores distintos. Para o funcionamento do compartilhamento de aprendizado, é necessário que os robôs sejam idênticos, ou ao menos tratem de comportamentos similares.

Observou-se nos testes que praticamente todos os neurônios associados à manutenção de distâncias mínimas foram inseridos nos segundos iniciais dos experimentos. Isto pode ser explicado pelo posicionamento inicial dos robôs, que começam em uma posição distante da meta de escolta, e pelas poucas entradas x_i associadas (menor distância entre o robô e algum obstáculo, seja ele fixo ou um vizinho, e a distância à meta de escolta) em contraste à 20 leituras realizadas pelos feixes *laser*. Mesmo quando Lim_{SIMIL} assume o valor 0.9, a rede associada à manutenção de distâncias mínimas se mantém com poucos neurônios. Esta rede do CM é, entretanto, importante para a manutenção da formação de escolta entre os robôs, e é necessária quando um robô está muito distante da meta de escolta e precisa recuperar seu posicionamento.

Também notou-se que o desempenho médio do exame reutilizando o conhecimento previamente adquirido mostrou-se superior, indicativo de que ocorreu de fato aprendizado durante as simulações. Este aprendizado ocorreu em menor escala utilizando $Lim_{SIMIL} = 0.9$ no segundo cenário de escolta devido ao aumento significativo do número de neurônios e conseqüente falta de generalização da representação dos dados sensoriais. Isto ocorreu porque trata-se de um cenário onde existem poucas variações possíveis de amostras sensoriais de distanciamento à meta de escolta, e portanto é de maior conveniência um número reduzido de neurônios.

Nos dois cenários onde foram apresentados obstáculos fixos, notou-se que um maior número de neurônios levou a uma dificuldade no reconhecimento de condições de maior risco de colisão. Durante o aprendizado sem reutilização de conhecimento, a generalização obtida com $Lim_{SIMIL} = 0.8$ alcançou uma média de colisões consideravelmente menor em contraste à vista utilizando $Lim_{SIMIL} = 0.9$.

O desempenho no terceiro cenário de escolta, entretanto, chega a mostrar que a reutilização do aprendizado obteve um desempenho similar para ambos os valores de Lim_{SIMIL} . O maior gasto de memória e de processamento, entretanto, não compensa o uso de um limiar elevado, especialmente levando-se em consideração que o AIS está sendo aplicado em

robôs que devem ter baixo custo, como tipicamente ocorre na Robótica de Enxame. Estes resultados são um indicativo de que a rede é capaz de aprender e de inserir neurônios quando necessário, mas esta capacidade é dependente de σ e Lim_{SIMIL} . Apesar de não terem sido experimentadas mudanças em σ , é provável que tais mudanças resultassem na variação do número de neurônios.

Os valores adotados para σ e Lim_{SIMIL} foram suficientemente elevados, o que levou os valores de S a manterem-se, também, elevados, pois não somente a similaridade para a atualização de pesos teria de ser elevada para a atualização como o valor de σ encoraja valores próximos de 1. Isto fez com que não ocorressem eliminações de neurônios durante os testes, pois seria muito difícil o alcance de uma média inferior a Lim_{DEL} para o vetor de pesos S .

O valor escolhido para α resultou em variações menores para os pesos S , o que pode ser o motivo pelo qual em muitos casos o robô, mesmo colidindo ou relativamente distante da meta, não se recuperou rapidamente: foi necessário um maior número de eventos para que este encorajamento a escapar do obstáculo ou para avançar acontecesse. Outro motivo que encorajou a adoção de um valor elevado para α foi o compartilhamento de neurônios entre os robôs: uma convergência acelerada da influência de um neurônio transmitida aos demais poderia aumentar rapidamente a influência do mesmo no resultado, podendo potencialmente viciar a rede. Um caso no qual isto poderia ocorrer é no início da simulação, onde quase todos os neurônios associados à manutenção de distâncias mínimas foram inseridos devido ao posicionamento inicial dos robôs, distantes da meta de escolta.

Algo que pode ter prejudicado os resultados da escolta é o mecanismo de controle simplificado adotado, que sempre aplica velocidades constantes aos robôs do enxame. Esta característica acabou apresentando um desafio adicional para o SLACS, a escolta e principalmente para o AIS. Muitas das colisões ocorrentes poderiam não acontecer caso este método fosse mais elaborado. Visto que o objetivo principal do trabalho era verificar se o AIS era capaz de aprender a lidar com as dificuldades e mudanças do ambiente, e que os próprios robôs são parte deste ambiente, este problema não interfere na análise central do trabalho. Em muitos dos testes onde a escolta utilizou o AIS, observou-se que os robôs conseguiam ficar bastante próximos entre si sem muitas colisões, mesmo sem haver um controle de velocidade.

A propriedade de espaçamento dos robôs do SLACS, que acabou

mantendo alguns robôs distantes do alcance ideal de 1 metro estipulado neste trabalho, mostrou ter benefícios para o desempenho, em especial considerando-se o número de colisões. Quando todos os robôs tentam entrar neste alcance, colisões ocorrem. O balanceamento de comportamentos criado pelo SLACS conseguiu encontrar um bom equilíbrio destas distâncias, mantendo distâncias seguras. Isto é, também, mérito das regras adotadas para a inserção de neurônios. A escolha das regras, apesar de ter grande impacto no desempenho, não foi explorada a fundo porque não é uma indicadora de aprendizado, mas sim um uso de conhecimento especialista, mesmo que rudimentar.

As modificações no módulo de saída para estimular variações na influência dos comportamentos além das apontadas pelas saídas do Módulo Coordenador y mostraram-se importantes, em especial para evitar que robôs não conseguissem escapar de obstáculos. Variações elevadas dos valores podem parecer desnecessárias visto a diferença significativa no desempenho dos robôs a partir de um leve ajuste da constante de influência de comportamentos, mas em algumas situações específicas, em particular quando o robô colide com o canto de um obstáculo, o centro de massa obtido pelo SLACS pode não ser suficientemente distante do desejado, exigindo que a influência do mesmo seja muito alta para que o robô escape.

O processo elaborado para a obtenção de m_i , apesar de ter sido suficiente para a obtenção de bons resultados, depende de muitos fatores. Um destes fatores, o módulo de b_i , não chegou a ser explorado devido à modularização dos valores. É provável que em um cenário de uso do AIS, por exemplo, no controle de velocidade de rodas de um robô, tratando a movimentação do robô em menor abstração do que no que foi apresentado neste trabalho, beneficiasse melhor do ajuste dos parâmetros da função sigmoial $f()$.

No próximo Capítulo, serão apresentadas conclusões sobre o sistema proposto para a tarefa de escolta, utilizando-se vários robôs, e os resultados obtidos, bem como os vários itens de investigação para trabalhos futuros.

Conclusão

Neste trabalho de Mestrado, foi proposto o uso de um mecanismo de aprendizado para robôs inspirado em fenômenos comportamentais chamado ANS (*Autonomous Navigation System*) em um enxame de robôs. A formulação desta rede foi generalizada para a aplicação em diversos cenários, e com isto o nome da mesma tornou-se AIS (*Autonomous Intelligent System*). O objetivo era testar esta rede em um problema diferente, abordando a tarefa de escolta por um enxame de robôs, um sistema multirrobótico independente de um operador central que é capaz de auto-cooperação e é altamente escalável. Este sistema realiza o aprendizado a partir de reforços recebidos quando um erro é cometido pelos robôs. O mecanismo de aprendizado da AIS, que insere como neurônio amostras diretas do ambiente, foi aproveitado para a realização de transmissão de conhecimento.

Para o desenvolvimento do algoritmo de escolta, um comportamento de manutenção de distâncias mínimas e um comportamento de cobertura de área foram combinados através de soma de vetores. O método permitiu que os robôs acompanhassem a meta de escolta, e, com um ajuste adequado da influência de seus comportamentos, é capaz de manter uma parcela dos robôs a uma distância maior dos demais de maneira natural para evitar colisões.

Um objetivo secundário do trabalho foi o desenvolvimento de um mecanismo de cobertura de área apto para um enxame de robôs, o SLACS *Sample Lloyd-based Area Coverage System*. Este método, inspirado no

algoritmo de Lloyd probabilístico para cálculo de centroides de polígonos de Voronoi, permitiu uma cobertura de área que lida naturalmente com a presença de obstáculos, utilizando os sensores de proximidade disponíveis para uma geração determinística das amostras utilizadas pelo algoritmo de Lloyd probabilístico. O método mostrou-se capaz de resultados comparáveis a uma cobertura de área que constrói explicitamente polígonos de Voronoi, e, ao contrário deste procedimento, utiliza somente um vizinho que esteja dentro de sua área de comunicação.

A aplicação do AIS conferiu ao enxame, que já foi capaz de um desempenho aceitável através do uso de influências constantes, uma robustez adicional, permitindo a identificação de situações de risco através de seus neurônios. O controle da plasticidade do AIS mostrou-se fundamental: os resultados nos quais um número mais balanceado de neurônios foram superiores aos apresentados pelo AIS quando um grande número de neurônios é inserido. O potencial de generalização a partir de poucas instâncias reforça aspectos interessantes do AIS para o aprendizado de um enxame de robôs: o armazenamento de amostras de conhecimento na forma de neurônios permite que o conhecimento recém-adquirido seja facilmente compartilhado com um vizinho.

Apesar do uso de um número relativamente pequeno de robôs considerando-se que se trata de

6.1 Sugestões de trabalhos futuros

Apesar dos bons resultados obtidos neste trabalho, alguns tópicos e análises estão em aberto. Para um sistema multirroboótico que não seja um enxame de robôs, em que os robôs possuam um maior potencial de processamento, seria possível explorar a estruturação do conhecimento para um compartilhamento mais profundo do que a transmissão de algo obtido em um momento específico. Métodos que consigam combinar conjuntos de neurônios de uma rede específica do Módulo Coordenador já existentes no AIS podem resultar em um mecanismo de compartilhamento de conhecimento poderoso. Ainda, o uso da atribuição de *fitness* combinado ao AIS poderia permitir o ajuste dos pesos dos neurônios de forma similar ao apresentado no trabalho de Pugh e Martinoli (2006), utilizando um algoritmo como a otimização de enxame de partículas.

Trabalhos futuros envolvendo um estudo de maior profundidade de

aspectos não investigados a fundo neste trabalho também são pretendidos, em especial investigando situações nas quais remoções de neurônios podem ocorrer. Outra possibilidade é a da aplicação do conceito de idade nos neurônios do CM, o que poderia vir a permitir uma maior amplitude de valores sem a necessidade dos pesos \bar{w} .

Experimentos nos quais a escolha seja realizada da maneira sugerida neste trabalho, mas com um mecanismo de controle bem elaborado, poderiam trazer excelentes resultados. O problema da escolha tem potenciais usos e é, ainda, pouco explorado. Uma sugestão seria uma adaptação desta escolha para robôs com movimentação não-holonômica, que apresenta dificuldades para acompanhar os centroides gerados pelo SLACS, motivo pelo qual uma movimentação onidirecional foi escolhida. A análise de outros mecanismos de transmissão de conhecimento também poderia trazer resultados interessantes.

Um trabalho futuro que será realizado é o de uma análise mais profunda do SLACS, medindo seu desempenho com a adoção de um número maior de vizinhos e experimentando a adaptação do código original, que considera feixes *laser*, com outros métodos que obtenham distâncias.

Este trabalho foi apoiado com auxílio financeiro da CAPES e da FAPESP. Processo nº 2012/14820-7, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências Bibliográficas

- Abraham, W. C. e Robins, A. (2005). Memory retention—the synaptic stability versus plasticity dilemma. *Trends in neurosciences*, 28(2):73–78. Citado na página 4.
- Antonelli, G., Arrichiello, F., e Chiaverini, S. (2008). The entrapment/escorting mission. *IEEE Robotics and Automation Magazine*, 15(1):22–29. Citado nas páginas 5 e 10.
- Arkin, R. C. (1998). *Behavior-based robotics*. MIT press. Citado na página 1.
- Asama, H., Matsumoto, A., e Ishida, Y. (1989). Design of an autonomous and distributed robot systems: Actress. In *IEEE/RJS International Conference on Intelligent Robots and Systems*, páginas 283–290. Citado na página 15.
- Balaguer, B. e Carpin, S. (2008). Where am I? A simulated GPS sensor for outdoor robotic applications. In *Simulation, Modeling, and Programming for Autonomous Robots*, páginas 222–233. Citado na página 53.
- Balch, T., Boone, G., Collins, T., Forbes, H., MacKenzie, D., e Santamar, J. (1995). Io, ganymede, and callisto: A multiagent robot trash-collecting team. *AI Magazine*, páginas 39–51. Citado na página 3.
- Batista, M., Calvo, R., e Romero, R. (2013). A robot on-line area coverage approach based on the probabilistic lloyd method. In *Proceedings of the International Joint Conference on Neural Networks*, páginas 2145–2152. Citado nas páginas 37, 46, 49, 50, 51, e 52.
- Batista, M., Silva, M., e Romero, R. (2012). Harmonic potential fields applied to frontier-based exploration problem using multiple robots. In *5th Workshop in Applied Robotics and Automation*, páginas 1–6. Citado na página 10.

- Box, G. E. P. e Muller, M. E. (1958). A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611. Citado na página 53.
- Breitenmoser, A., Schwager, M., Metzger, J., Siegwart, R., e Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *IEEE International Conference on Robotics and Automation*, páginas 4982–4989. Citado na página 28.
- Bruyninckx, H. (2001). Open robot control software: the orocos project. In *IEEE International Conference on Robotics and Automation*, volume 3, páginas 2523–2528. Citado na página 15.
- Burgard, W., Moors, M., Fox, D., Simmons, R., e Thrun, S. (2000). Collaborative multi-robot exploration. In *IEEE International Conference on Robotics and Automation*. Citado na página 10.
- Calvo, R. (2012). *Sistemas bio-inspirados para coordenação de múltiplos robôs móveis*. Tese de Doutorado, Universidade de São Paulo. Citado nas páginas 37 e 46.
- Calvo, R. e Figueiredo, M. (2003). Reinforcement learning for hierarchical and modular neural network in autonomous robot navigation. In *International Joint Conference on Neural Networks*, volume 2, páginas 1340 – 1345. Citado na página 18.
- Calvo, R., Oliveira, J., Figueiredo, M., e Romero, R. (2010). Autonomous cognition and reinforcement learning for mobile robots. In *International Joint Conference on Neural Networks*, páginas 1–8. Citado nas páginas 4, 5, 18, 22, 23, 24, 41, e 49.
- Calvo, R., Oliveira, J., Figueiredo, M., e Romero, R. (2011). Inverse aco applied for exploration and surveillance in unknown environments. In *The Third International Conference on Advanced Cognitive Technologies and Applications*, páginas 142–147. Citado na página 14.
- Chaimowicz, L., Michael, N., e Kumar, V. (2005). Controlling swarms of robots using interpolated implicit functions. In *IEEE International Conference on Robotics and Automation*, páginas 2487–2492. Citado na página 13.
- Chao, H., Chen, Y., e Ren, W. (2006). A study of grouping effect on mobile actuator sensor networks for distributed feedback control of diffusion

- process using central voronoi tessellations. In *IEEE International Conference on Mechatronics and Automation*, páginas 769–774. Citado na página 11.
- Conway, J. (1970). The game of life. *Scientific American*. Citado na página 2.
- Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. *Lecture Notes in Computer Science*, 3342:10–20. Citado nas páginas 3, 12, e 84.
- Di Mario, E. e Martinoli, A. (2013). Distributed particle swarm optimization for limited-time adaptation with real robots. *Robotica*, páginas 1–16. Citado nas páginas 20, 44, e 53.
- Di Mario, E., Talebpour, Z., e Martinoli, A. (2013). A comparison of pso and reinforcement learning for multi-robot obstacle avoidance. In *IEEE International Conference on Evolutionary Computation*, volume 1, páginas 149–156. Citado na página 20.
- Du, Q., Gunzburger, M., e Ju, L. (2010). Advances in studies and applications of centroidal voronoi tessellations. *Numerical Mathematics: Theory, Methodologies and Applications*, 28:1477–1500. Citado na página 27.
- Edelman, G. (1987). *Neural Darwinism: The theory of neuronal group selection*. Basic Books. Citado na página 22.
- Figueiredo, M. F. (1997). *Redes neurais nebulosas aplicadas em problemas de modelagem e controle autônomo*. Tese de Doutorado, Tese de doutorado, Universidade Estadual de Campinas. Citado na página 18.
- Genchev, S., Venkov, P., e Vidolov, B. (2008). Trilateration analysis for movement planning in a group of mobile robots. In *Artificial Intelligence: Methodology, Systems, and Applications*, páginas 353–364. Citado na página 53.
- Gerkey, B. P., Vaughan, R. T., e Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *International Conference on Advanced Robotics*, páginas 317–323. Citado na página 15.
- Hagras, H., Colley, M., Callaghan, V., e Carr-West, M. (2002). Online learning and adaptation of autonomous mobile robots for sustainable agriculture. *Autonomous Robots*, 13:37–52. Citado na página 19.

- Heinen, M. R. e Engel, P. M. (2010). Aprendizado e controle de robôs móveis autônomos utilizando atenção visual. *Revista de Informática Teórica e Aplicada*, 17(3):349–354. Citado na página 6.
- Hsu, H. C. e Liu, A. (2005). Multiagent-based multi-team formation control for mobile robots. *Journal of Intelligent and Robotic Systems*, 42(4):337–360. Citado na página 10.
- Jennings, J., Whelan, G., e Evans, W. (1997). Cooperative search and rescue with a team of mobile robots. In *8th International Conference on Advanced Robotics*, páginas 193–200. Citado na página 9.
- Ju, L., Du, Q., e Gunzburger, M. (2002). Probabilistic methods for centroidal voronoi tessellations and their parallel implementations. *Parallel Computing*, 28:1477–1500. Citado nas páginas 5, 11, e 28.
- Kamano, T., Yasuno, T., Suzuki, T., Hasegawa, Y., Harada, H., e Kataoka, Y. (2000). Design and implementation of fuzzy cooperative catching controller for multiple mobile robots. In *26th Annual Conference of the IEEE Industrial Electronics Society*, volume 3, páginas 1749–1754. Citado na página 10.
- Krishnanand, K., Amruth, P., Guruprasad, M., Bidargaddi, S., e Ghose, D. (2006). Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources. In *IEEE International Conference on Robotics and Automation*, páginas 958–963. Citado na página 13.
- Lee, M., Tarokh, M., e Cross, M. (2010). Fuzzy logic decision making for multi-robot security systems. *Artificial Intelligence*, páginas 177–194. Citado nas páginas 3 e 11.
- Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137. Citado nas páginas 12 e 27.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, páginas 281–297. Citado na página 28.
- Mamdani, E. H. e Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of man-machine studies*, 7(1):1–13. Citado na página 22.

- Mas, I., Li, S., Acain, J., e Kitts, C. (2009). Entrapment/escorting and patrolling missions in multi-robot cluster space control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 5855–5861. Citado na página 11.
- Matarić, M. J. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4:73–83. Citado na página 16.
- Mishra, A., Singh, S., Bhattacharya, S., e Pattnaik, P. (2012). A novel area coverage management scheme for sensor network with mobile sensor nodes. *Indian Journal of Science and Technology*, 5(8):3122–3127. Citado na página 29.
- Murphy, R. (2000). *Introduction to AI Robotics*. The MIT Press. Citado na página 21.
- Pavlov, I. P. (1927). *Conditioned Reflexes*. Oxford University Press. Citado na página 21.
- Penders, J., Alboul, L., Witkowski, U., Naghsh, A., Saez-Pons, J., Herbrechtsmeier, S., e El-Habbal, M. (2011). A robot swarm assisting a human fire-fighter. *Advanced Robotics*, 25(1-2):93–117. Citado nas páginas 11 e 13.
- Pessin, G., Osório, F., e Wolf, D. (2010). Particle swarm optimization applied to intelligent vehicles squad coordination. In *IFAC Symposium on Intelligent Autonomous Vehicles*, páginas 1–6. Citado na página 12.
- Pfeifer, R. e Scheier, C. (1999). *Understanding intelligence*. The MIT Press. Citado na página 4.
- Pugh, J. e Martinoli, A. (2006). Multi-robot learning with particle swarm optimization. In *Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, páginas 441–448. Citado nas páginas 4, 19, 44, e 90.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *14th annual conference on Computer graphics and interactive techniques*, páginas 25–34. Citado na página 13.
- Rounds, S. e Chen, Y. (2009). Cooperative phototaxis using networked mobile sensors and centroidal voronoi tessellations. In *Proceedings of the 2009*

- American Control Conference*, páginas 3274–3279. Citado nas páginas 15 e 37.
- Sharkey, N. (1998). Learning from innate behaviors: A quantitative evaluation of neural network controllers. *Machine Learning*, 31(1):115–139. Citado nas páginas 4, 16, 17, e 18.
- Silva, F., Urbano, P., e Christensen, A. L. (2012). Adaptation of robot behaviour through online evolution and neuromodulated learning. In *Advances in Artificial Intelligence à IBERAMIA 2012*, volume 7637 of *Lecture Notes in Computer Science*, páginas 300–309. Springer. Citado nas páginas 6 e 17.
- Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., e Younes, H. (2000). Coordination for multi-robot exploration and mapping. In *AAAI Proceedings on Robotics and Automation*. Citado na página 3.
- Stanley, K. O. e Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127. Citado nas páginas 17 e 18.
- Tan, J., Lozano, O., Xi, N., e Sheng, W. (2004). Multiple vehicle systems for sensor network area coverage. In *Fifth World Congress on Intelligent Control and Automation*, volume 5, páginas 4666–4670. Citado nas páginas 14, 28, 29, e 37.
- Varela, G., Caamamo, P., Orjales, F., Deibe, A., Lopez-Pena, F., e Duro, R. (2011). Swarm intelligence based approach for real time uav team coordination in search operations. In *Third World Congress on Nature and Biologically Inspired Computing*, páginas 365–370. Citado na página 14.
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Tese de Doutorado, Universidade de Cambridge, Inglaterra. Citado na página 19.
- Wurm, K., Stachniss, C., e Burgard, W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 1160–1165. Citado na página 10.
- Zuo, G., Han, J., e Han, G. (2010). Multi-robot formation control using reinforcement learning method. In *Advances in Swarm Intelligence*, páginas 667–674. Citado nas páginas 19 e 44.