

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

HENRIQUE SEITI KONDO

**MAPAS COGNITIVOS FUZZY DINÂMICOS APLICADOS NA EXPLORAÇÃO DE
AMBIENTES DESCONHECIDOS INSPIRADOS EM ROBÓTICA DE ENXAME**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2019

HENRIQUE SEITI KONDO

**MAPAS COGNITIVOS FUZZY DINÂMICOS APLICADOS NA EXPLORAÇÃO DE
AMBIENTES DESCONHECIDOS INSPIRADOS EM ROBÓTICA DE ENXAME**

Trabalho de Conclusão de Curso de graduação do curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Márcio Mendonça

CORNÉLIO PROCÓPIO
2019



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento Acadêmico de Elétrica
Curso de Engenharia de Controle e Automação



FOLHA DE APROVAÇÃO

Henrique Seiti Kondo

Mapas Cognitivos Fuzzy Dinâmicos aplicados na exploração de ambientes desconhecidos inspirados em robótica de enxame

Trabalho de conclusão de curso apresentado às 15:00hs do dia 22/04/2019 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). Márcio Mendonça - Presidente (Orientador)

Prof(a). Dr(a). Wagner Fontes Godoy - (Membro)

Prof(a). Dr(a). Rodrigo Rodrigues Sumar - (Membro)

AGRADECIMENTOS

Agradeço à minha família por permitirem estar aqui e serem a base da minha vida, sempre apoiando minhas decisões e estando ao meu lado nos momentos mais difíceis. Meus pais, Edson e Sandra, que fizeram de tudo para que eu e meu irmão, Fernando, conseguíssemos estudar; meus avôs que me mostraram que com dedicação no trabalho podemos conquistar nossos objetivos; meus tios, Fernando e Lisa, que são um dos meus modelos de vida.

Ao professor Márcio Mendonça, por acreditar em meu potencial e aceitar ser meu orientador. Pelos ensinamentos desde as aulas de Robótica e da monitoria dessa mesma disciplina até os dias de hoje.

Ao pessoal da AARLab, em especial, Lucas, pelas horas de convivência e oferecer um ambiente agradável ao trabalho, pelos artigos publicados e pelos ensinamentos e sugestões para o trabalho.

Aos meus amigos da minha cidade, Ibiporã, principalmente Nanami, Bruno, Miguel e Shina que moraram comigo durante boa parte dessa caminhada.

Ao grupo de robótica Overload, que foi um dos principais responsáveis pelo meu contato com a Robótica e proporcionou grande experiência nessa área. Agradeço aos meus colegas de turma, em especial, Gabriel e Filipe que tornaram grandes amigos durante essa caminhada.

Agradeço ao pessoal da HS Technology, local do meu estágio e de muitos ensinamentos, tanto profissional, quanto pessoal.

Aos professores Sumar, Godoy e Endo, por aceitarem ser parte da banca da minha apresentação de Trabalho de Conclusão de Curso.

Por fim, à UTFPR e seus funcionários por serem os responsáveis por tornar o dia a dia dessa vida de estudante universitário possível.

RESUMO

KONDO, Henrique S. **Mapas cognitivos fuzzy dinâmicos aplicados na exploração de ambientes desconhecidos inspirados em robótica de enxame.** 2019. 65 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Este trabalho propõe o estudo da utilização dos Mapas Cognitivos *Fuzzy* Dinâmicos (DFCM) para a coordenação de robôs em um sistema multi-agentes (SMA), inspirado na robótica de enxame e na arquitetura dos agentes, em especial, a arquitetura de subsunção de Brooks. A modelagem é feita por meio dos DFCM e os agentes devem explorar ambientes semi ou totalmente desconhecidos. Os testes para este estudo foram simulados, entretanto são realizados em diferentes ambientes através do *software* MatLab® para verificar a autonomia e validar a arquitetura proposta. O trabalho compara os resultados obtidos com a simulações feitas com *Fuzzy Logic Controller* (FLC) utilizado por SOARES *et al.* (2017), afim de verificar seu baixo custo computacional. Esse trabalho contribui como base para o desenvolvimento de outros trabalhos que possam utilizar o DFCM.

Palavras-chave: Arquitetura de navegação robótica. Exploração de ambiente desconhecido. Mapas Cognitivos *Fuzzy* Dinâmicos. Robô móvel autônomo. Robótica de enxame.

ABSTRACT

KONDO, Henrique S. **Dynamic fuzzy cognitive maps applied in the exploration of unknown environments inspired in swarm robotics**. 2019. 65 f. Monograph – Control and Automation Engineering. Federal University of Technology - Paraná. Cornélio Procópio, 2019.

This work proposes the study of the use of Dynamic Fuzzy Cognitive Maps (DFCM) for the coordination of robots in a multi-agent system (SMA), inspired by swarm robotics and agent architecture, especially the subsumption architecture of Brooks. The modeling is done through the DFCM and the agents must explore semi or totally unknown environments. The tests for this study were simulated, but are performed in different environments through MatLab® software to verify the autonomy and validate the proposed architecture. The work compares the results obtained with simulations made with Fuzzy Logic Controller (FLC) used by SOARES et al. (2017), in order to verify its low computational cost. This work contributes to the development of other works that can use the DFCM.

Keywords: Robotic navigation architecture. Exploration of unknown environment. Dynamic Fuzzy Cognitive Maps. Autonomous mobile robot. Swarm robotics.

LISTA DE FIGURAS

Figura 1 - Pilares para a implementação da Indústria 4.0	8
Figura 2 - Exemplo do modelo de um veículo de Braintenberg	13
Figura 3 - Robôs (Kilobots) baseados na Robótica de Enxame	16
Figura 4 - Exemplo de um FCM clássico	19
Figura 5 - Modelo Geométrico para um Robô Móvel	24
Figura 6 - Movimento Infinitesimal	25
Figura 7 - Sensor Ultrassônico HC-SR04	27
Figura 8 - Funcionamento de um Sensor Ultrassônico	27
Figura 9 - Alcance dos Sensores	28
Figura 10 - Planos auxiliares com robô em diferentes poses.	29
Figura 11 - Distância entre pontos	30
Figura 12 - Quadrante em função de senos e cossenos.	31
Figura 13 - Síntese da alternância entre os modos de operação	35
Figura 14 - Máquina de Estados	36
Figura 15 - FCM dos Modos de Operação 1 e 2	37
Figura 16 - FCM dos Modos de Operação 3	39
Figura 17 - Ambiente de simulação I	41
Figura 18 - Ambiente de simulação II	42
Figura 19 - Ambiente de simulação III	42
Figura 20 - Exemplo de área explorada e não explorada	44
Figura 21 - Simulação do ambiente I com 1 robô.	45
Figura 22 - Pulsos do robô na simulação do ambiente I	46
Figura 23 - Área explorada pelo robô no ambiente I.	46
Figura 24 - Simulação do ambiente I com 4 robôs.	47
Figura 25 - Pulsos dos robôs na simulação do ambiente I	48
Figura 26 - Área explorada pelos robôs no ambiente I.	48
Figura 27 - Área total explorada pelos robôs no ambiente I	49
Figura 28 - Simulação do ambiente II com 1 robô.	51
Figura 29 - Pulsos do robô na simulação do ambiente II	51
Figura 30 - Área explorada pelo robô no ambiente II.	52
Figura 31 - Simulação do ambiente II com 4 robôs.	53
Figura 32 - Pulsos dos robôs na simulação do ambiente II	53
Figura 33 - Área explorada pelos robôs no ambiente II.	54
Figura 34 - Área total explorada pelos robôs no ambiente II.	54
Figura 35 - Simulação do ambiente III com 1 robô.	56
Figura 36 - Pulsos do robô na simulação do ambiente III	57
Figura 37 - Área explorada pelo robô no ambiente III	57
Figura 38 - Simulação do ambiente III com 4 robôs.	58
Figura 39 - Pulsos dos robôs na simulação do ambiente III	59
Figura 40 - Área explorada pelos robôs no ambiente III.	59
Figura 41 - Área total explorada pelos robôs no ambiente III.	60

Sumário

1. INTRODUÇÃO	7
1.1. Problema a ser pesquisado	8
1.2. Motivação.....	9
1.3. Objetivo geral	11
1.4. Objetivos específicos	11
1.5. Estrutura do trabalho.....	12
2. FUNDAMENTAÇÃO TEÓRICA.....	13
2.1. Robótica Móvel Autônoma	13
2.2. Robótica de Exame.....	14
2.3. Arquitetura de subsunção	16
2.4. Mapas Cognitivos <i>Fuzzy</i> Dinâmicos (DFCM)	18
3. METODOLOGIA	23
3.1. Estrutura do robô.....	23
3.1.1. Modelo Cinemático.....	23
3.1.2. Simulação dos Sensores.....	26
3.1.3. Modelo Robótico Simulado	31
3.2. Sistema de Controle do Robô	32
3.2.1. Arquitetura de Subsunção.....	32
3.2.2. Modos de Operação.....	32
3.2.2.1. Modo de operação zero: movimento livre.....	33
3.2.2.2. Modo de operação um: desvio de obstáculos	33
3.2.2.3. Modo de operação dois: iminência de colisão	34
3.2.2.4. Modo de operação três: captura de alvos	34
3.2.2.5. Síntese dos modos de operação	34
3.2.3. DFCM	36
3.3. Ambiente de teste.....	40
3.4. Simulação	43
3.4.1. Parâmetros de teste	43
4. RESULTADOS E DISCUSSÕES.....	45
4.1. Resultados do ambiente de teste I	45
4.2. Resultados do ambiente de teste II	50
4.3. Resultados do ambiente de teste III	56

5. CONCLUSÃO	62
5.1. Trabalhos Futuros	62
6. REFERÊNCIAS	63

1. INTRODUÇÃO

A evolução da ciência e tecnologia trouxe a sofisticação da noção de robôs. Podemos definir robôs como agentes que podem sentir e agir no ambiente para completar seus objetivos (MATARIC, 2008) e, para isso, realizam manipulações no mundo físico, equipados de efetores como pernas, rodas, articulações e garras. Além disso, conta com um conjunto cada vez mais diversificado de sensores para a percepção do ambiente, como câmeras e lasers, e para medir o movimento do robô, como giroscópios e acelerômetros (RUSSELL e NORVIG, 2013).

Essa prática da robótica e visão de máquina envolve a aplicação de algoritmos computacionais e dados coletados a partir desses sensores que medem, por exemplo, a intensidade dos milhões de pixels que compõem uma imagem do mundo. Para muitas aplicações robóticas a quantidade de dados que precisa ser processada em tempo real é enorme. Para visão do robô sobre o mundo, pode ser da ordem de dezenas a centenas de megabytes por segundo (CORKE, 2017).

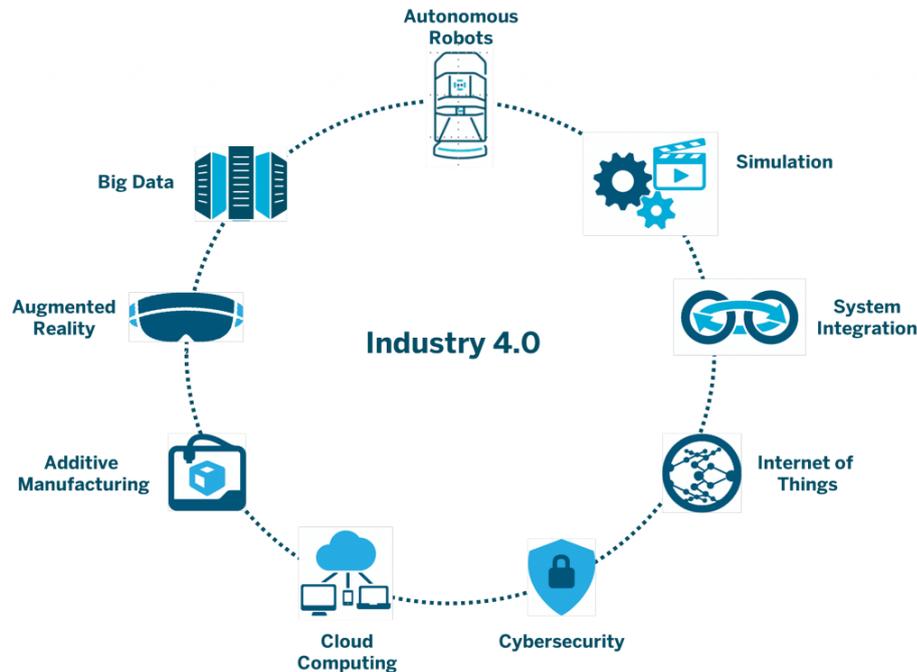
Uma das categorias dentro da robótica que vem se desenvolvendo na comunidade científica são os robôs autônomos, no qual podem executar diversas tarefas e substituir o ser humano, como entregar comida em hospitais, mover containers em docas, dentre outros trabalhos similares (RUSSELL e NORVIG, 2013).

A utilização de robôs autônomos, como podemos ver na Figura 1, é um dos pilares para a implementação de um dos conceitos recentemente abordados, a Indústria 4.0. Nesse conceito, tem se a integração de todo o sistema de produção, maior flexibilidade e eficiência, utilizando sistemas inteligentes baseados em rede (GOMEZ *et. al.*, 2017).

Os robôs que atuam nessa área compartilham de uma grande quantidade de dados e informações obtidas, o chamado *Big Data* – outro conceito presente na Indústria 4.0, permitindo que seu sistema tome decisões com maior precisão de forma colaborativa. Seguindo essa linha de pensamento, a utilização de um sistema inteligente de baixo custo computacional que consiga processar tais dados é de suma importância. No trabalho realizado por PEREIRA (2017), robôs são feitos com base em celulares e utilizam da “Nuvem” (*Cloud Computing*) para compartilhar as informações mais rapidamente. Esse trabalho parte de outros que utilizam do chamado Robótica em Rede (*Networked Robotics*) que é grupo de sistemas

robóticos conectados através de uma rede de comunicação de dados, sendo esta cabeada ou sem fio (PEREIRA, 2017). Na figura a seguir podemos observar todos os pilares para a implementação da Indústria 4.0.

Figura 1 - Pilares para a implementação da Indústria 4.0



Fonte: <<https://www.embalagemmarca.com.br/2017/05/conheca-os-nove-pilares-para-implantacao-da-industria-4-0/>> Acessado em: 28/03/2019.

1.1. Problema a ser pesquisado

Inicialmente, a pesquisa no desenvolvimento de sistemas autônomos inteligentes teve foco em ambientes com obstáculos estáticos e com um único robô (CALVO, 2007). Porém há algumas tarefas como, busca e resgate de vítimas, mapeamento do ambiente, plantio e pulverização de uma plantação e vigilância onde a utilização de mais de um robô contribui para o sucesso e a maior eficiência para o cumprimento de seus objetivos.

Devido a isso, pesquisas utilizando vários robôs, os chamados Sistemas Multi-Agentes (SMA), foram iniciados. A partir dessas pesquisas foram listadas algumas vantagens de um SMA em relação ao sistema de um único agente, como: distribuição do espaço com conseqüente redução do tempo de execução da tarefa, decréscimo da complexidade individual, possibilidade de permitir o compartilhamento remoto de informações, tolerância a falhas através de agentes

redundantes e custo (ARAI et al., 2002; ARKIN, 1999). Por outro lado, com o uso de sistemas com múltiplos robôs surgiram situações que não existiam, como: o problema das colisões entre os agentes, a coordenação dos mesmos e a confiabilidade das informações transmitidas entre esses robôs (PUIG; GARCIA; WU, 2011).

Uma possível divisão do conceito de SMA é a Robótica de Enxame, do inglês “*Swarm Robotics*”, no qual vários robôs considerados simples simulam comportamentos coletivos observados na natureza em várias espécies vivas. As interações locais entre os indivíduos e o ambiente conduz o grupo de agentes autônomos a solucionar tarefas complexas em maneiras distribuídas, sem uma unidade de controle central (BAYINDIR, 2016).

A possibilidade de alcançar objetivos globais no nível de enxame por meio de algoritmos distribuídos que atuam no nível individual tem seu preço: é muito difícil modelar o comportamento de robôs individualmente que maximize o desempenho global (BAYINDIR, 2016). Esse problema é amplamente estudado por pesquisadores, e tem sido abordado principalmente em simulações, no qual um cenário do SMA é replicado e, as capacidades do robô (sensores e atuadores) e suas interações são simuladas por um computador. Isso permite avaliar o desempenho do programa através de repetidas execuções, eliminando ou mitigando os experimentos com robôs reais que necessitam de tempo e facilitando a otimização do algoritmo.

Assim para este trabalho, pretende-se utilizar conceitos da Robótica de Enxame para auxiliar robôs em missão de resgate, como em casos do acidente nuclear de Fukushima, desabamento do prédio Wilton Paes de Almeida, rompimento da represa de Brumadinho, incêndio na cidade de São Paulo, pessoas perdidas em florestas ou vítimas de terremotos, onde operariam em locais com obstáculos estáticos e dinâmicos desconhecidos e quantidade incerta de alvos (vítimas).

1.2. Motivação

Mesmo com o avanço de dispositivos e funcionalidades dos robôs e com a utilização de conceitos como o da Robótica de Enxame, existem problemas na criação de um robô autônomo para a exploração de ambientes parcialmente e, especialmente, totalmente desconhecidos. Dentre esses, os principais são: a

modelagem do ambiente, percepção, localização e o controle das tomadas de decisões (COSTA e GOUVEA, 2010). Os sistemas de navegação autônomos devem ser capazes de tomar decisões sequenciais que levem ao agente se locomover em ambientes estáticos e dinâmicos, completando seu objetivo principal, sempre garantindo sua integridade física.

Geralmente as tarefas que utilizam SMA possuem requisitos que envolvem custos para se cumprir o objetivo. Por exemplo, em uma tarefa de busca e resgate de sobreviventes em desastres, o requisito crucial é o tempo, ou seja, quanto menor o tempo para se encontrar a vítima, melhor. Ou ainda a otimização das trajetórias para se economizar recursos, como o combustível consumido (DE ALMEIDA et al., 2019).

Na área de robôs autônomos móveis, há alguns estudos recentes de arquiteturas, baseadas em técnicas de inteligência computacional, para o controle das decisões na Robótica de Enxame. Tais como Controle Lógico *Fuzzy* (SOARES et al., 2017), Mobile Ad Hoc Network (MANET) (DUCATELLE et al., 2014), Mapas Cognitivos *Fuzzy* Dinâmico (CHRUN, 2016).

Os Mapas Cognitivos *Fuzzy*, do inglês *Fuzzy Cognitive Maps* (FCM), é uma técnica recente usada para o desenvolvimento de sistemas autônomos. Utilizando do conhecimento heurístico, ou seja, expressando o conhecimento de especialistas por meio de palavras ou expressões de linguagem - do mesmo modo que um Controlador *Fuzzy* - interligados por uma simples relação de causa e efeito para modelar as tomadas de decisões em navegação autônoma (TOLMAN, 1997).

Essa ferramenta surgiu da combinação dos aspectos de Redes Neurais Artificiais e da Lógica *Fuzzy*, sendo a versão clássica proposta por KOSKO (1986) e formalizado por YUAN e ZHI-QIANG (2000). Como uma técnica computacional, os FCMs são capazes de modelar sistemas complexos e podem ser aplicados em uma ampla gama de problemas de engenharia e áreas correlatas (PAPAGEORGEIOU, 2012). Porém MIAO et al. (1999) aponta em seu trabalho que o FCM possui dificuldade em modelar relações de causa e efeito em sistemas dinâmicos ou temporais e, portanto, para o problema de um sistema real proposto, o FCM pode haver inferências contraditórias.

Assim, com um estudo mais avançado sobre os FCMs, surgiram os Mapas Cognitivos *Fuzzy* Dinâmicos, do inglês *Dynamics Fuzzy Cognitive Maps* (DFCM), que são modelos cognitivos que utilizam o mesmo conhecimento heurístico para

construção dos modelos, juntamente com dados on-line para sua sintonia e auto adaptação dinâmica.

A ontologia apresentada por MENDONÇA e ARRUDA (2015) no desenvolvimento de um sistema de navegação autônomo serviu de ponto de partida para o modelo desenvolvido neste trabalho. Ainda para este trabalho foi utilizado o modelo cinemático apresentado por SOARES *et al.* (2017), sendo este trabalho utilizado para fazer a comparação entre o modelo *Fuzzy* e o DFMC, avaliando as vantagens e desvantagens do modelo proposto.

Assim a motivação ficará por conta da criação de um sistema de navegação autônoma, baseada num modelo DFMC, com a capacidade de aprendizado, sintonia, adaptação, gerenciamento de ações e evolução, utilizando conceitos da Robótica de Enxame.

1.3. Objetivo geral

Este trabalho tem como objetivo principal o desenvolvimento de uma arquitetura, baseada em um DFMC proposto por MENDONÇA *et al.* (2017) para o controle de robôs móveis autônomos, aplicando o conceito da Robótica de Enxame. O DFMC desenvolvido deve ser capaz de controlar e gerenciar as tomadas de decisão dos robôs em um ambiente desconhecido, atingindo os objetivos propostos de forma a garantir sua integridade.

Para isso, o explorador (robô móvel) deve ser simulado através de um *software* computacional diante de pelo menos três ambientes diferentes, para este ser considerado autônomo (MAES, 1995; MATARIC, 2007; RUSSELL e NORVIG, 2013).

1.4. Objetivos específicos

- Projetar uma arquitetura DFMC para o controle da tomada de decisão dos robôs que serão empregados nos enxames para a procura de vítimas;
- Implementar em tempo de testes simulados as funções e modos de operação necessários para interação com o ambiente;
- Implementar agentes que caracterizam Robótica de Enxame (Desenvolver os cenários de teste);

- Comparar e analisar possíveis melhorias comparadas ao trabalho precursor SOARES, em especial a complexidade computacional que tem sido observado em trabalhos publicados na literatura;
- Testar e validar o modelo desenvolvido por meio de experimentos simulados;

1.5. Estrutura do trabalho

Os próximos capítulos deste trabalho são organizados da seguinte maneira:

O Capítulo 2 apresenta o embasamento teórico do problema proposto, uma fundamentação teórica sobre Robôs Móveis Autônomos, Robótica de Enxame, Arquitetura de Subsunção e Mapas Cognitivos *Fuzzy* Dinâmicos.

O Capítulo 3 aborda a metodologia utilizada para a resolução do problema apresentado e as etapas elaboradas deste trabalho.

O Capítulo 4 mostra os resultados e os comentários sobre estes.

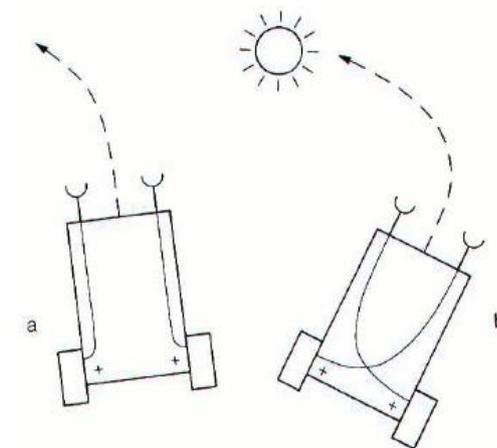
No Capítulo 5 são feitos os comentários finais acerca deste trabalho e os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Robótica Móvel Autônoma

Por volta de 1940, um neurofisiologista chamado William Gray Walter, interessado na maneira de como o cérebro funciona, construiu uma variedade de máquinas que ele chamou de “*turtles*” ou “*tortoises*” (tartarugas – devido a sua forma física). Essas máquinas eram robôs simples construídos com três rodas, como um triciclo, e sensores sensíveis à luz (MATARIC, 2007). Inspirado por esse trabalho, Valentino Braitenberg escreveu seu livro *Vehicles* em 1984, uma das principais fontes para modelos de robôs/veículos móveis, como pode ser visto um exemplo na Figura 2.

Figura 2 - Exemplo do modelo de um veículo de Braitenberg.



Fonte: *Vehicles: Experiments in Synthesic Psychology*, publicado pelo The MIT Press

Esses dois trabalhos (Walter e Braitenberg) apresentam modelos de agentes somente reativos (ações de baixo nível), ou seja, que reagem somente a estímulos, tais como decisões para virar à esquerda ou à direita quando os sensores acusam algum obstáculo.

Todo robô capaz de se movimentar e interagir com o ambiente está inserido na robótica móvel (RUSSELL e NORVIG, 2013). Os robôs móveis são geralmente aqueles que podem se mover de um lugar para outro. De acordo com o início do parágrafo, pode-se dizer que a capacidade de adaptação dá ao robô maior

flexibilidade para executar tarefas novas e complexas, na qual o ambiente não precisa ser modificado (GORIS, 2005).

Para o robô ser considerado autônomo deve ser capaz de reagir a obstáculos e responder a eventos não modelados ou previstos (SMITHERS, 1997; MATARIC, 2007), autonomamente, ou seja, sem interferências externas (RUSSELL e NORVIG, 2013), visando à execução de tarefas para as quais ele foi projetado, em um ambiente complexo e dinâmico (MAES, 1995).

Assim, o conhecimento chamado de deliberativo (ações de alto nível), como por exemplo, o gerenciamento ou planejamento da trajetória do agente, estão relacionados com a tomada de decisão e por isso são de extrema importância para a modelagem de um robô móvel autônomo (CHRUN, 2016).

2.2. Robótica de Enxame

A Robótica de Enxame, ou *Swarm Robotics*, é um campo de pesquisa de Sistemas Multi-Agentes, para a coordenação de um largo grupo de robôs relativamente simples que interagem e cooperam uns com os outros para resolver em conjunto, tarefas que estão fora dos recursos de cada robô (DORIGO e SAHIN, 2004; DUDEK et al., 1996). Esse conceito foi inspirado na observação de insetos sociais - formigas, cupins, vespas e abelhas - que são conhecidos por coordenar suas ações na realização de tarefas que estão além da capacidade de um único indivíduo, como na construção de grandes e complexos montes pelos cupins, forrageamento (busca por alimento) e transporte coletivamente de grandes presas pelas formigas (SAHIN, 2005).

Essa abordagem foi proposta nos anos 80, como aplicação de *Swarm Intelligence* (Inteligência de Enxames) para sistemas multi-robôs. O termo *Swarm Intelligence* (BONABEAUE et al., 1999) foi concebido por BENI (1989) para denotar uma classe de sistemas robóticos celulares.

Estudos de DENEUBOURG (2001) revelaram que não existem mecanismos centralizados de coordenação por trás da operação sincronizada dos insetos sociais, essa coordenação ocorre a partir de interações locais entre os robôs individuais ou entre os robôs e o ambiente (DE WOLF e HOLVOET, 2005).

De acordo com SAHIN (2005) o seu funcionamento é em nível de sistema robusto, flexível e escalável, sendo essas propriedades motivações para a abordagem da Robótica de Enxame:

- **Robustez:** o sistema Robótico de Enxame deve ser capaz de operar apesar dos distúrbios do ambiente ou do mau funcionamento de seus indivíduos. Alguns fatores da robustez podem ser observados em insetos sociais:
 1. Os indivíduos são redundantes, ou seja, a perda de um agente pode ser compensada imediatamente por outro.
 2. A coordenação é descentralizada e, portanto, a perda de um indivíduo não interrompe a operação do sistema.
 3. Os indivíduos que compõem o enxame são relativamente simples, e em comparação a um único sistema complexo, este é menos propensos à falha.
 4. A detecção (sensoriamento) é distribuída, portanto, o sistema é robusto contra as perturbações locais do ambiente.
- **Flexibilidade:** o sistema tem a capacidade de gerar soluções para tarefas diferentes, utilizando diferentes estratégias de coordenação em resposta às mudanças no ambiente.
- **Escalabilidade:** o sistema deve ser capaz de operar sob uma ampla faixa de número de agentes, ou seja, os mecanismos de coordenação que asseguram o funcionamento do enxame e não devem ser perturbados por mudanças no tamanho do grupo.

DUDEK *et al.* (1996) identificaram as seguintes categorias de tarefas de interesse da robótica: tarefas que são inerentes único agente, tarefas que podem se beneficiar com o uso de múltiplos robôs, tarefas que são tradicionalmente multi-agentes, tarefas que exigem o uso de um SMA. A Robótica de Enxame tem o foco nas últimas três categorias e, para este trabalho temos o caso de uma tarefa que é tradicionalmente multi-agentes, pois uma busca por uma vítima é geralmente feita por mais de um indivíduo, para aumentar a eficiência e diminuir o tempo.

Nesse trabalho utilizou-se das propriedades da Robótica de Enxame para o desenvolvimento do sistema, principalmente por ter um controle descentralizado. Devido a isso, denominamos o sistema como inspirado em Robótica de Enxame.

Como trabalho futuro pode-se implementar os feromônios utilizados pelas formigas ou outras estratégias de forrageamento por parte desses insetos.

Existem atualmente vários trabalhos para o desenvolvimento físico desse conceito, dentre eles temos o *Swarm-bot*, um enxame de robôs (s-bots) relativamente simples que são capazes de se conectarem; o *Swarmoid* que é um enxame composto por três tipos de robôs, *eye-bots* (voadores), *hand-bots* (escaladores) e os *foot-bots* (transportadores); os *TERMES* que são robôs inspirados em cupins para a construção de estruturas; o *Thousand-robot Swarm* que é um enxame composto por 1024 *Kilobots* (Figura 3), robôs pequenos e de baixo custo equipados apenas de motores de vibração para o movimento e um sensor infravermelho para a comunicação e detecção da distância. Todos esses sistemas desenvolvidos foram criados inspirados nos comportamentos observados nos insetos sociais e são capazes de trabalharem cooperativamente na realização de trabalhos complexos.

Figura 3 - Robôs (Kilobots) baseados na Robótica de Enxame.



Fonte:

<https://static.projects.iq.harvard.edu/files/styles/os_files_xlarge/public/ssr/files/img_8886.jpg?m=1494006979&itok=_hVStNSS> Acessado em 09/05/2018.

2.3. Arquitetura de subsunção

A arquitetura é um conjunto de especificações e técnicas utilizadas para a definição funcional de um robô. Um sistema de controle desenvolvido para um agente autônomo deve realizar o processamento de várias informações e sua arquitetura deve suportar tal processamento. Para o trabalho proposto, o robô deve ter um sistema de navegação autônomo, que necessita tomar decisões sequenciais

em ambientes desconhecidos. Assim, suas condições limites podem mudar rapidamente, inferindo decisões contraditórias, como quando um alvo aparece na mesma direção que o agente detecta um obstáculo. O nível de complexidade desse problema exige uma arquitetura com capacidade de decomposição dos comportamentos em uma série de sub-comportamentos (BROOKS, 1986). A arquitetura que propõe esse tipo de resolução é denominada de subsunção e foi proposta por Rodney Brooks em 1986 (BROOKS, 1986). Essa arquitetura é uma das mais clássicas e inspirou outras arquiteturas mais sofisticadas, com a orientada a eventos, visto no trabalho de MENDONÇA (2012).

Essa ferramenta trouxe uma nova perspectiva na modelagem de robôs, influenciando diversas pesquisas na área da robótica autônoma e inteligência artificial (IA) em tempo real. Diferente ao conceito tradicional de IA, que organiza o comportamento através de representações simbólicas do ambiente, esta permite simular um comportamento complexo a partir da sua decomposição em sub-comportamentos, ordenando em camadas (hierárquicas ou não) e resolvendo cada um (BROOKS, 1986).

A hierarquização dos comportamentos ou como chamados nesse trabalho, modos de operação do agente, é seguida pelo princípio da incompatibilidade onde quanto mais complexo for um sistema, mais difíceis e imprecisas será prever como ele vai reagir ao ambiente (BRAITENBERG, 1986). Assim, são desenvolvidas as camadas de controle, correspondendo a cada nível de prioridade, e adicionando uma nova camada sobre a anterior, seguindo de comportamentos mais simples para os comportamentos mais sofisticados. De forma geral, é implementado primeiramente os comportamentos reativos (camadas de baixo nível), em direção aos deliberativos (camadas de alto nível) (CHRUN, 2016).

Os modos de operação alternam entre si a partir das informações obtidas pelos sensores, sendo inibidos ou ativados. Todos os modos funcionam paralelamente e, muitos comportamentos podem ser simultaneamente ativados. As camadas de comportamento mais elevadas podem observar o estado das camadas mais baixas, podendo inibir suas saídas e substituir suas entradas, ajustando seu comportamento (BROOKS e STEIN, 1994).

2.4. Mapas Cognitivos *Fuzzy* Dinâmicos (DFCM)

Os Mapas Cognitivos foram propostos por AXELROD (1976) para representar palavras, ideias, tarefas ou outros itens ligados a um conceito central. Baseado na teoria dos grafos e operações com matrizes, os Mapas Cognitivos são representados por um conjunto de “nós” ligados por “arcos” direcionados e ponderados com valores numéricos (pesos). Esses nós representam os conceitos e os arcos são as relações de causa-efeito que eles possuem entre si, sendo a direção para determinar a orientação da influência e o peso se este possui uma relação causal negativa (quando um conceito aumenta, o outro diminui) ou positiva (quando um aumenta, o outro também aumenta) (KOK, 2009; TOLMAN, 1997).

Com a inclusão da incerteza, através da *lógica Fuzzy*, KOSKO (1986) apresentou os Mapas Cognitivos *Fuzzy* (*Fuzzy Cognitive Maps - FCM*). O FCM herda os conceitos dos Mapas Cognitivos, adicionando os valores dos pesos representados por números *Fuzzy*, ou seja, uma quantidade cujo valor é incerto, ao invés de exato, sendo assim uma combinação da *lógica Fuzzy* com o mapeamento cognitivo, para representar o conhecimento de sistemas complexos que são caracterizados pela incerteza. Tornando um método para modelar sistemas complexos utilizando do conhecimento e da experiência humana.

O FCM pode ser considerado um tipo de rede cognitiva neuro-difusa, que é inicialmente desenvolvido por aquisição de conhecimento especializado, mas podem ser treinados por vários supervisores e/ou técnicas não supervisionadas (PAPAGEORGIU, 2014).

Numa perspectiva da inteligência artificial, os FCM são redes dinâmicas com capacidades de aprendizado, quanto maior a quantidade de dados disponíveis para modelar o sistema, este se torna melhor em se adaptar e alcança uma solução (PAPAGEORGIU, 2014). Essas funcionalidades os tornam essenciais e melhora o desempenho nas tarefas de modelagem, análise e tomada de decisões.

Para o processamento do raciocínio do FCM, é gerada uma formulação matemática simples. Os valores do conceito C_i no tempo t são representados pelo vetor de estado $A_i(k)$, sendo $A(0)$ o vetor de entrada. O valor A_i de cada conceito C_i em um momento $k + 1$ é calculado pela soma do valor anterior de A_i em um momento precedente t com o produto do valor A_j do nó C_j no momento k e o valor do arco de causa-efeito w_{ij} . Assim temos a representação matemática mais

utilizada:

$$A_i(k+1) = f\left(A_i(k) + \sum_{j=1}^N A_j(k) * w_{ij}\right) \quad (1)$$

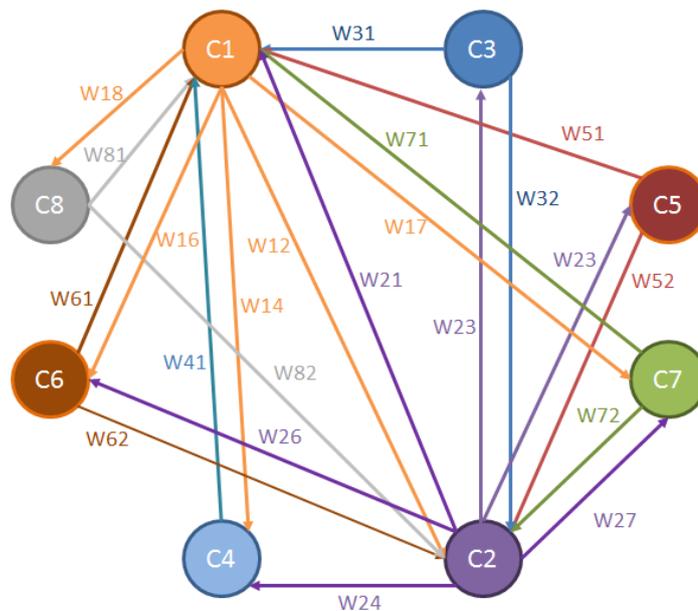
Onde f é uma função de limite (ativação). Uma função de limite do tipo sigmoide tem se os valores de conceito no intervalo $[0, 1]$, sendo seu modelo matemático:

$$f(x) = \frac{1}{1 + e^{-m.x}} \quad (2)$$

Onde m é um número real positivo (fator de esquecimento) e x é o valor de $A_i(k)$ no ponto de equilíbrio.

Um exemplo de um FCM é mostrado na Figura 4, o grafo identifica as respectivas relações causais e seus conceitos. Sua respectiva matriz de pesos é apresentada na matriz 3.

Figura 4 - Exemplo de um FCM clássico.



Fonte: (MENDONÇA et al., 2015d)

$$W = \begin{bmatrix} 0 & w_{12} & 0 & w_{14} & 0 & w_{16} & w_{17} & w_{18} \\ w_{21} & 0 & w_{23} & w_{24} & 0 & w_{26} & w_{27} & 0 \\ w_{31} & w_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{41} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{51} & w_{52} & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{61} & w_{62} & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{71} & w_{72} & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{81} & w_{82} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

Porém, MIAO *et al.* (1999) apontam que o FCM tem dificuldade em modelar relações de causa e efeito em sistemas dinâmicos ou temporais, e, portanto, pode haver inferências contraditórias no FCM quando há uma modelagem de um sistema real complexo.

Como uma evolução do FCM, o DFCM funciona de acordo com a complexidade do problema modelado. Para decisões em que ambiguidades estão envolvidas, mas sem uma precedência temporal entre eles, essa ferramenta se comporta como um FCM clássico em que as relações causais são conceitos *Fuzzy*, mas ainda considerando que todas as causalidades ocorrem simultaneamente. Um modelo DFCM pode dinamicamente mudar os valores das relações causais entre seus conceitos e/ou alterar a estrutura do grafo representando o sistema. O DFCM pode mudar através a partir dos estados do sistema, tonando-se dinamicamente através de eventos.

Além disso, também é possível adicionar ou excluir relações causais entre conceitos. Essas características torna o DFCM uma ferramenta sofisticada para a modelagem de sistemas dinâmicos não-lineares (MENDONÇA *et al.*, 2013a).

Cada peso da matriz pode também ser modelado como uma tupla:

$$(N, Ci, Co, r, U, Br) \quad (4)$$

No qual:

- N identifica a camada ou nível em que as relações pertencem, isto é, uma relação causal pura tem $N = 0$, uma vez que esta pertence ao menor nível da camada;
- Ci são os conceitos de entrada;
- Co são os conceitos de saída;

- r é o tipo de relação, sendo uma relação causal, relação causal variante no tempo, relação *Fuzzy* ou uma relação de seleção (resultante da execução de uma base de regras);
- U descreve o universo discurso da relação, podendo representar um valor numérico, um intervalo ou uma variável linguística;
- Br é o índice que representa a base de regras relevantes para a relação, assim uma relação puramente causal ou variante no tempo tem $Br = 0$.

O design de modelos DFCM, mesmo para sistemas pequenos, requer uma grande quantidade de conceitos e relacionamentos que precisam ser estabelecidos, o que dificulta o desenvolvimento manual do sistema. Para facilitar este processo e permitir uma análise consistente dos mapas, o modelo DFCM deve ser desenvolvido de forma incremental, de acordo com o tipo de conexões modeladas de causa-efeito e organizadas em uma ontologia, como a proposta por MENDONÇA e ARRUDA (2015) (PAPAGEORGIOU, 2014).

Tabela 1 – Etapas de desenvolvimento da DFCM (ontologia).

Descrição	
Etapa 1:	Identificação de conceitos elementares, suas funções (entrada, saída, decisão e nível) e suas correlações, determinando sua natureza causal (positiva, negativa, neutra).
Etapa 2:	Configuração inicial dos conceitos e relações causais. Os valores dos estados iniciais do mapa (nós/arcos) podem ser adquiridos através do conhecimento de especialistas, análise de dados históricos e / ou simulação do sistema.
Etapa 3:	Determinação da influência ontológica entre os conceitos. Concepção dos diferentes pontos de vista do sistema.
Etapa 4:	Para cada ponto de vista do sistema, projetar uma base de regra <i>Fuzzy</i> e funções variantes no tempo para calcular dos valores dos pesos das relações causais do DFCM.
Etapa 5:	Processamento da informação, adaptação e otimização do modelo DFCM, ajustando suas respostas aos valores desejados de saída. Se necessário, um método de treinamento de máquina (Aprendizagem por Reforço, Aprendizado de Hebb, Algoritmos Evolutivos, ou outras técnicas) pode ser utilizado para sintonia dinâmica do modelo.
Etapa 6:	A modelagem do nível de gestão correspondendo ao desenvolvimento da base de regras associadas aos conceitos de fatores e relações de seleção; a implementação do algoritmo de aprendizado online.
Etapa 7:	Desenvolver métodos ou algoritmos para compartilhar informações (esta etapa é opcional).
Etapa 8:	Teste final e validação do modelo desenvolvido.

Fonte: MENDONÇA e ARRUDA (2015)

Um guia para projetar de forma incremental um modelo DFCM baseado nessa ontologia é dado na Tabela 1. Os passos 1 e 2 correspondem ao desenvolvimento clássico do FCM (KOSKO, 1986). Os passos 3 e 4 estão relacionados à inclusão das relações *Fuzzy* nos grafos e das variáveis no tempo para esse modelo de relações de causa e efeito. O passo 5 corresponde a um passo do modelo de ajuste/adaptação, muito semelhante ao aprendizado em redes neurais. No passo 6, estão incluídos as bases de regras associadas ao nível de decisão estratégica. Na etapa 7, se exigido pelo problema, os métodos ou algoritmos para compartilhar informações adquiridas linha são considerados. Finalmente, o passo 8 corresponde à etapa de validação do modelo que é usual na modelagem acionada por dados.

3. METODOLOGIA

A proposta desse trabalho é o desenvolvimento de uma arquitetura, baseada em Mapas Cognitivos *Fuzzy* Dinâmicos (DFCM), para coordenar robôs autônomos na navegação de ambientes semidesconhecidos e utilizar os resultados obtidos para comparar com o trabalho de SOARES *et al.* (2017), que utiliza o Controlador Lógico *Fuzzy* (FLC) para tal coordenação.

Assim, as configurações para simulação como, a estrutura dos robôs (modelo cinemático, sensores e modelo robótico) e os ambientes de teste (a posição dos robôs e dos alvos) para ambos os sistemas, DFCM e FLC, devem ser iguais. Essas configurações foram baseadas no trabalho anterior (SOARES *et al.*, 2017) com algumas alterações para o aperfeiçoamento das simulações. Todas essas simulações foram realizadas no *software Matlab®*.

Nesta seção é apresentada a metodologia empregada no desenvolvimento deste trabalho. Cada subseção é dividida para a explicação mais detalhada dos materiais utilizados.

3.1. Estrutura do robô

O trabalho foi iniciado com o estudo da estrutura dos robôs utilizado por SOARES *et al.* (2017).

3.1.1. Modelo Cinemático

Nesta subseção é explicado como foi desenvolvido o modelo cinemático dos exploradores, ou seja, como é feita a simulação de seus movimentos, a fim de deixar o mais próximo da realidade.

Os robôs utilizados possuem duas rodas, ou seja, possuem dois motores para seu movimento. Esses motores se movimentam a partir de pulsos entre $[-1$ e $1]$, sendo os valores positivos responsáveis por fazerem o motor se movimentar em um sentido, empregado neste trabalho como “para frente”, e os valores negativos responsáveis pelo sentido inverso, definido como “para trás”. Quanto maior o valor do pulso mandado para o servo motor, maior o movimento do robô, por exemplo, o pulso com valor 1 fará com que o motor dê um giro completo e o pulso com valor 0,

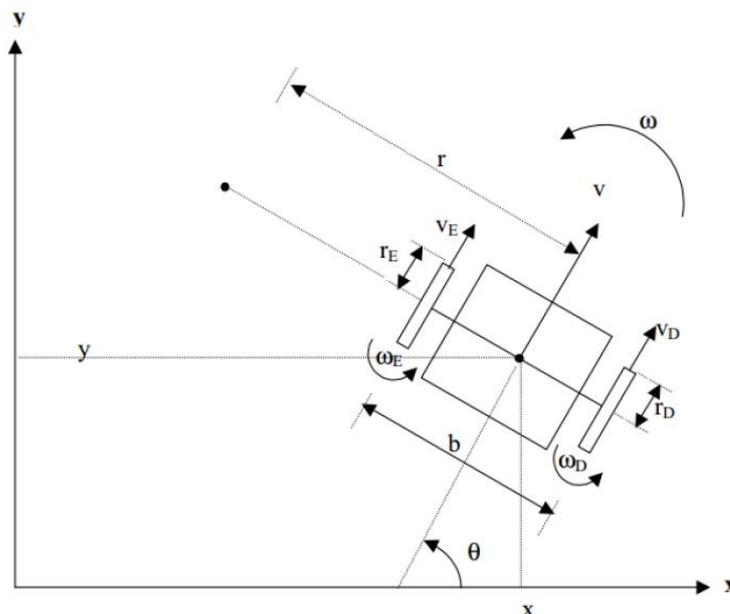
metade de um giro completo.

Assim, para fazer com que o robô se desloque para frente ou para trás, os dois motores devem receber dois sinais de módulos iguais, mudando somente o sinal. Caso o robô queira virar para a direita, porém continuando sua trajetória, o motor da direita deve receber um pulso com valor maior que o da esquerda. O mesmo serve para caso queira virar para a esquerda, no qual o valor de pulso recebido pelo motor esquerdo deve ser maior. O módulo do pulso interfere na quantidade deslocada e no ângulo de curva feito pelo robô.

Para essa simulação, não foi implementado a ré para o robô, porém este recebe pulsos negativos caso tenha que desviar de uma iminente colisão quando, por exemplo, encontra com um obstáculo dinâmico (outros robôs) ou um obstáculo estático em sua frente. Assim, um dos motores receberá um valor positivo e outro motor, um valor negativo, fazendo com que o robô gire sobre seu próprio eixo.

A seguir são demonstrados as equações desenvolvidas até chegar na matriz final (Equação 6) utilizada na simulação e empregada no trabalho anterior (SOARES *et al.*, 2017).

Figura 5 – Modelo Geométrico para um Robô Móvel

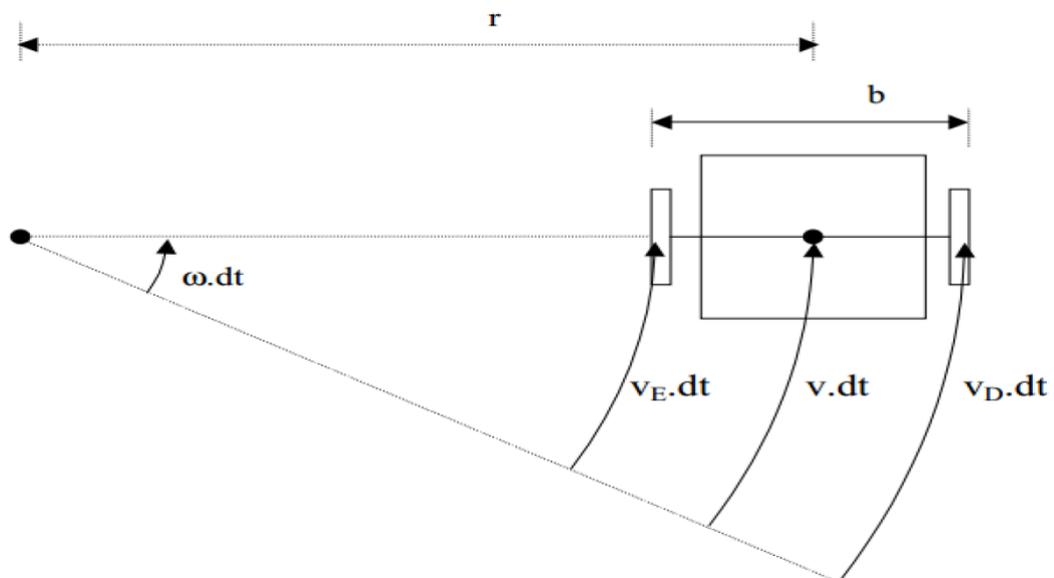


Fonte: Adaptado de Alsina (2016).

A partir do diagrama de um robô móvel demonstrado na Figura 5, podemos observar as variáveis:

- $[x, y]$: Posição do robô.
- θ : Ângulo de direção.
- b : Comprimento do eixo.
- r : Raio de giro.
- r_d e r_e : Raio das rodas direita e esquerda, respectivamente.
- ω : Velocidade angular do robô.
- ω_d e ω_e : Velocidade angular nas bordas das rodas direita e esquerda, respectivamente.
- v : Velocidade linear do robô.
- v_d e v_e : Velocidade linear nas bordas das rodas direita e esquerda, respectivamente.

Figura 6 - Movimento Infinitesimal



Fonte: Adaptado de Alsina (2016).

A partir da análise do movimento infinitesimal do robô (Figura 6) podemos demonstrar a velocidade linear v e a angular ω em função dos pulsos ω_d e ω_e . Inicialmente temos que:

$$v = \omega \cdot r \quad (5)$$

De acordo com o movimento infinitesimal, podemos afirmar que:

$$v_d \cdot dt = \omega_d \cdot \left(r \pm \frac{b}{2}\right) dt \quad (6)$$

$$v_e \cdot dt = \omega_e \cdot \left(r \pm \frac{b}{2}\right) dt \quad (7)$$

Utilizando as equações (6) e (7) temos que:

$$v_d + v_e = \omega_d \cdot r_d + \omega_e \cdot r_e = 2 \cdot \omega \cdot r = 2 \cdot v \quad (8)$$

$$v_d - v_e = \omega_d \cdot r_d - \omega_e \cdot r_e = \omega \cdot b \quad (9)$$

Isolando os termos v e ω das equações (8) e (9), obtemos a matriz:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r_d}{2} & \frac{r_e}{2} \\ \frac{r_d}{b} & -\left(\frac{r_e}{b}\right) \end{bmatrix} \cdot \begin{bmatrix} w_d \\ w_e \end{bmatrix} \quad (10)$$

A matriz obtida e demonstrada pela equação (Equação 10) é a utilizada para simular o movimento dos robôs. A partir dos pulsos ω_d e ω_e que são fornecidos pelo DFCM e informações dos exploradores, como o raio de cada roda e o comprimento do seu eixo, podemos simular o movimento do robô e saber a posição deste a cada iteração.

3.1.2. Simulação dos Sensores

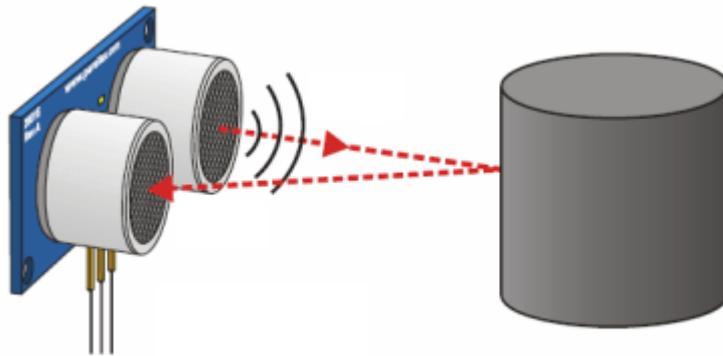
A simulação dos sensores é feita através de um algoritmo, que foi desenvolvido utilizando um modelo real de sensor, o sensor ultrassônico HC-SR04 (Figura 7). Esse sensor funciona através da emissão de um pulso sonoro de alta frequência que, ao atingir um objeto, recebe de volta um sinal de eco. A partir desse sinal refletido, tem-se a distância entre o sensor e o objeto. A imagem de seu funcionamento pode ser visto na Figura 8.

Figura 7 - Sensor Ultrassônico HC-SR04



Fonte: <https://www.robocore.net/loja/sensores/sensor-de-distancia-ultrassonico-hc-sr04>

Figura 8 - Funcionamento de um Sensor Ultrassônico.

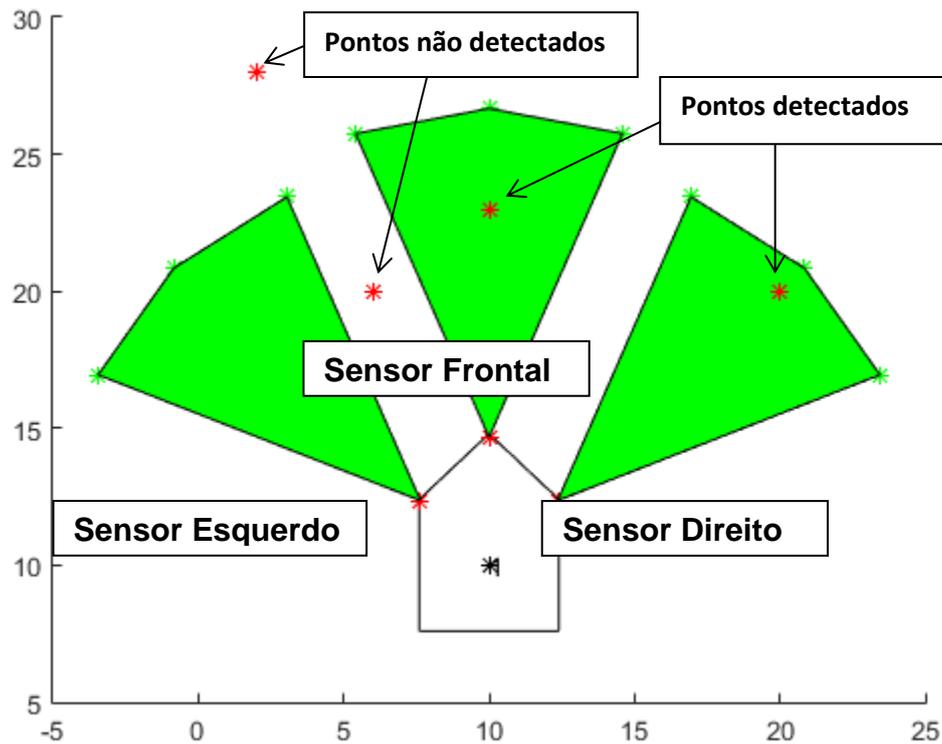


Fonte: <https://portal.vidadesilicio.com.br/hc-sr04-sensor-ultrassonico/>

A partir de testes com o sensor real, foram obtidos o alcance e o ângulo de abertura dos sensores. Assim, o robô foi desenvolvido para possuir três sensores, responsáveis pela interação com o ambiente, detectando obstáculos e alvos, e mapeando o ambiente. Os sensores foram posicionados na área frontal e distribuídos a fim de permitir uma melhor interação com o ambiente. A região verde visto na Figura 9 é a zona de leitura dos sensores, ou seja, os alvos e os obstáculos são detectados apenas quando estão dentro dessa região.

Mesmo posicionando os sensores desta forma, é possível ver a existência de pontos “cegos” e, por isso, durante a simulação alguns obstáculos podem não ser mapeados ou alvos não serem encontrados em um primeiro instante. Esse problema é comum e é descrito em trabalhos como de BROOKS (1986).

Figura 9 - Alcance dos Sensores



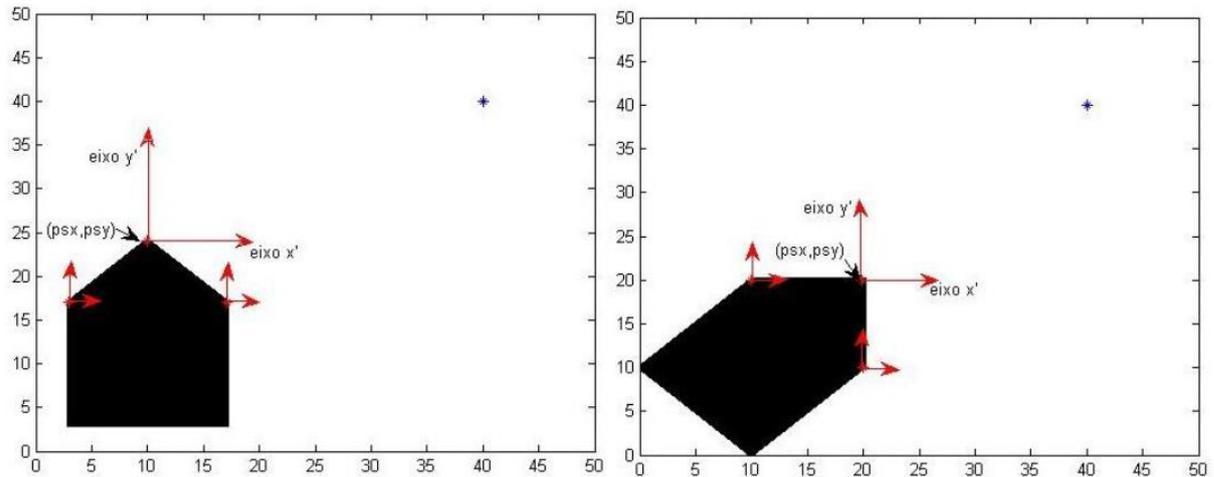
Fonte: Autoria Própria

Para obter as informações dos pontos de detecção por parte de cada sensor, é feita a cada iteração, a verificação da distância e angulação entre cada um dos pontos em relação a cada um dos sensores do robô. Isso é feito através do algoritmo de simulação dos sensores.

Adotando-se um plano cartesiano $[x,y]$ e sabendo a posição de cada sensor, é possível aplicar o teorema de Pitágoras para o cálculo da distância e do ângulo, do triângulo formado pelos eixos x , y e a distância do ponto ao sensor, como é visto na Figura 10.

Para esse trabalho, a origem do plano cartesiano $(0,0)$ está fixa e cada sensor possui seu próprio plano cartesiano auxiliar $[x',y']$ e a posição do sensor $[psx,psy]$ para cálculo de distância e ângulo. Os eixos dos planos auxiliares sempre estão paralelos aos eixos do plano cartesiano para facilitar os cálculos e diminuir a complexidade da simulação.

Figura 10 - Planos auxiliares com robô em diferentes poses.



Fonte: Retirado de SOARES *et al.* (2017).

Na Figura 10 e nas equações a seguir, o sensor frontal é utilizado como exemplo, assim para cada iteração é necessário realizar os cálculos para cada um dos sensores (Sensor Direito, Sensor Frontal e Sensor Esquerdo).

Primeiro, são calculadas as distâncias Δx e Δy , que são as distâncias relativas entre o ponto de referência do sensor (psx , psy) e o obstáculo (xf , yf), através das equações (11) e (12), respectivamente:

$$\Delta x = xf - psx \quad (11)$$

$$\Delta y = yf - psy \quad (12)$$

O próximo passo é calcular a distância entre a posição do sensor e o alvo com a equação (13).

$$dist = \sqrt{\Delta x^2 + \Delta y^2} \quad (13)$$

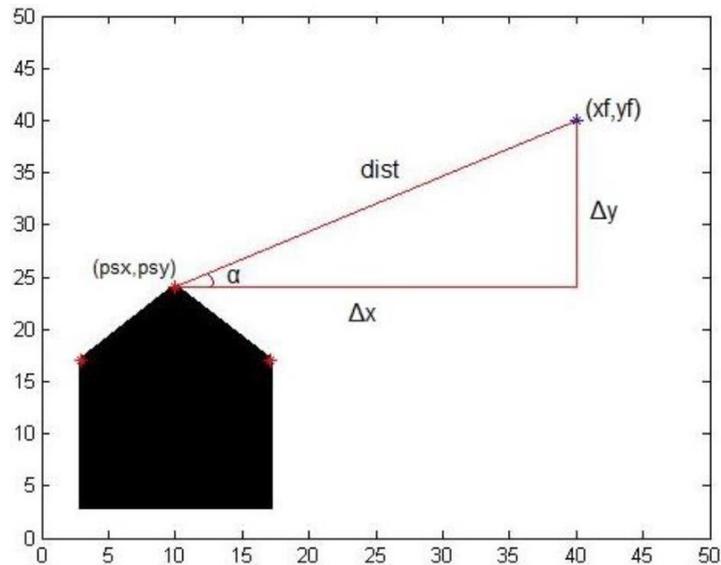
A partir dos resultados das equações (11), (12) e (13), é possível obter mais informações do ângulo α (ângulo formado entre o alvo e o eixo auxiliar x') com valores de $[0^\circ, 360^\circ]$, como visto na Figura 11.

$$sen\alpha = \frac{\Delta y}{dist} \quad (14)$$

$$cos\alpha = \frac{\Delta x}{dist} \quad (15)$$

$$\alpha = \operatorname{tg}^{-1}\left(\frac{\Delta y}{\Delta x}\right) \quad (16)$$

Figura 11 - Distância entre pontos.



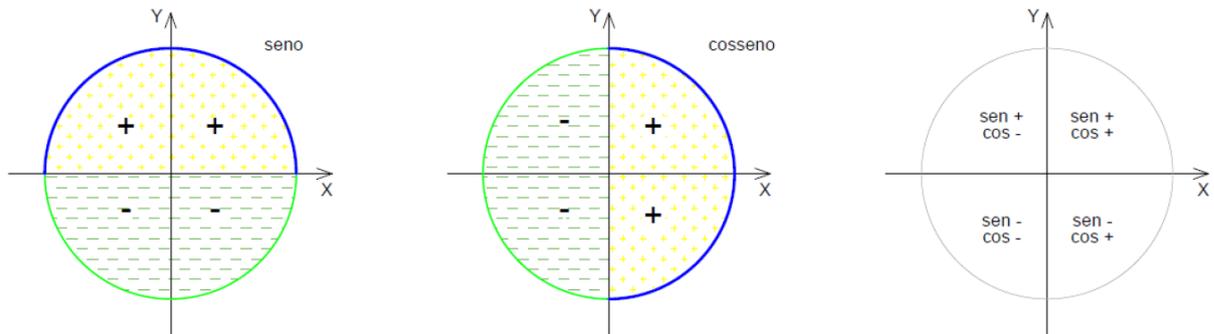
Fonte: Retirado de SOARES *et al.* (2017).

Porém, apenas com o resultado da equação 16 não é possível obter o real valor do ângulo α entre o alvo e o eixo x' , pois ângulos em diferentes quadrantes podem ter o mesmo valor de tangente. Para determinar em qual quadrante o ponto está em relação ao robô, utiliza as informações obtidas pelas equações 14 e 15. A partir das seguintes regras, podemos determinar o quadrante do ângulo (SOARES, 2017):

- Seno de α positivo e cosseno de α positivo: o alvo está no primeiro quadrante e, portanto não precisa de alterações, o valor do ângulo α está correto;
- Seno de α positivo e o cosseno de α negativo: o alvo está no segundo quadrante e, portanto deve-se adicionar 180° ao valor do ângulo α ;
- Seno de α negativo e cosseno de α positivo: o alvo está no quarto quadrante e, portanto deve-se adicionar 360° ao valor do ângulo α .
- Seno de α negativo e cosseno de α negativo: o alvo está no terceiro quadrante e, portanto deve-se adicionar 180° ao valor do ângulo α ;

Os fundamentos dessa base de regras podem ser vistos nos círculos trigonométricos da Figura 12.

Figura 12 - Quadrante em função de senos e cossenos.



Fonte: Retirado de SILVA (2015).

Assim, como visto na Figura 9, se o ponto estiver dentro da região do sensor, ou seja, possuir distância calculada menor que o alcance do sensor e o ângulo α estar dentro do ângulo de captação do sensor, este ponto será mapeado ou será marcado para captura (no caso de alvos). Já para o caso do ponto estar a uma distância calculada maior que o alcance e/ou o ângulo α estiver fora do ângulo de captação do sensor, nada acontecerá e o ponto continuará oculto.

Quando o robô detecta um obstáculo ou alvo, ele manda informações da distância entre o ponto e o sensor que o detectou. Caso não tenha nenhum obstáculo em sua região de captação, essa informação volta como o alcance máximo do sensor.

As variáveis utilizadas para cada sensor foi: SD (sensor direito), SF (sensor frontal) e SE (sensor esquerdo).

3.1.3. Modelo Robótico Simulado

Para realizar a simulação dos robôs, foi necessário estabelecer os valores das variáveis, algumas já apresentadas, como o alcance e o ângulo dos sensores e, outros no quesito da estrutura física do robô, como as dimensões dos robôs. Os valores utilizados na simulação para o modelo robótico é apresentado a seguir:

- Distância de segurança do robô: para garantir sua integridade frente a um obstáculo, foi definido como 4 unidades de distância da simulação. Esta distância é importante para que o robô consiga desviar a tempo do obstáculo, visto que não consegue ir para trás;
- Dimensões do robô: raio da roda e comprimento da base, ambos de extrema

importância para o modelo cinemático, foram definidos como 0,8333 e 4,8 respectivamente;

- Alcance dos sensores e ângulo de captação: necessário para saber o quão distante um alvo será detectado e se ele está dentro da área de alcance dos sensores, foram definidos como 12 unidades de distância e 45°;
- Direção de cada sensor em relação à direção inicial do robô: o sensor frontal (SF) será sempre igual à direção atual do robô, enquanto o sensor esquerdo (SE) e o direito (SD) estarão defasados 45° positivamente e negativamente em relação ao frontal;
- Erro desejado de captura: o centro de massa do robô deve estar a 3 unidades de distância do alvo para ele ser considerado capturado.

Os robôs possuem um compartilhamento de informações onde eles sabem a quantidade de alvos e quando um é capturado, essa informação é passada para os demais.

3.2. Sistema de Controle do Robô

Após definir a estrutura do robô para a simulação, começou a ser desenvolvido sua arquitetura e seu sistema de controle.

3.2.1. Arquitetura de Subsunção

Seguindo a metodologia do trabalho, o sistema de controle dos robôs foi iniciado empregando a arquitetura de subsunção de Brooks, no qual foi desenvolvido a partir de um modelo simples até um modelo mais complexo, através da implementação de funções conforme a complexidade do problema (BRAINTENBERG, 1986). Essa arquitetura permite simular um comportamento complexo a partir da decomposição em sub-comportamento, que neste trabalho é chamado de modos de operação. Assim foram desenvolvidos tais sub-comportamentos, realizando testes a cada implementação na arquitetura.

3.2.2. Modos de Operação

Os Modos de Operação foram formulados através da necessidade do robô

durante sua simulação e a partir do pensamento de especialistas. Foram então, desenvolvidos quatro modos de operação sendo:

- Modo 0 – Movimento Livre;
- Modo 1 – Desvio de Obstáculos;
- Modo 2 – Iminência de Colisão;
- Modo 3 – Captura de Alvos.

Os Modos de Operação 1, 2 e 3 utilizam do DFCM para mandarem pulsos para os motores. Cada um desses Modos de Operação possui seu DFCM e seus parâmetros são apresentados na Subseção 3.2.3.

3.2.2.1. Modo de operação zero: movimento livre

Na simulação, todos os robôs começam com o Modo de Operação 0 e a cada iteração, seu modo é atualizado para este, por isso foi definido como modo “zero”. Neste modo, o pulso que os motores recebem não é ajustado pelo DFCM, pois nesse modo o robô segue em linha reta. Quando entra nesse modo, os motores do robô recebem metade do valor máximo (0,5), caso continue nesse modo, irá aumentar (0,1) a cada iteração até chegar a seu valor máximo (1). Assim caso não encontre nenhum obstáculo ou alvo nas cinco próximas iterações, o robô estará em sua velocidade máxima.

Porém, para que o robô não fique “preso” dentro de um ambiente, se este ativar o modo de operação 2 por mais de 70 iterações, a próxima vez que este estiver no modo de operação 0, este se movimentará para um dos lados e não em linha reta.

3.2.2.2. Modo de operação um: desvio de obstáculos

Quando o robô detecta algum obstáculo e este não está em uma distância menor que sua distância segura (definida como 4 unidades), este Modo de Operação é ativado. Ele é responsável pelas curvas mais suaves do robô.

Os valores de saída dos pulsos para cada motor é determinado pelo DFCM correspondente a esse modo e varia de acordo com a leitura dos sensores quanto à

distância entre o sensor e o obstáculo.

3.2.2.3. Modo de operação dois: iminência de colisão

Este modo é ativado quando um dos três sensores ultrassônicos simulados detecta a presença de um obstáculo com uma distância menor do que a de segurança do explorador (definida como 4 unidades). Quando isso ocorre, o DFCM responsável por esse modo gera os pulsos para os motores do robô que o gira para evitar a colisão.

Este modo também é responsável por impedir as colisões entre os robôs. Ao utilizar mais de um robô em cada ambiente, eles são considerados como obstáculos dinâmicos aos demais indivíduos e ao avistar outro robô e ativar este modo de operação, cada robô desvia para um lado. Caso um robô encontre um robô em sua frente, este diminui a velocidade para desviar deste, podendo até a quase parar para tal.

3.2.2.4. Modo de operação três: captura de alvos

Este modo é ativado após a detecção de algum alvo dentro da área de alcance dos sensores. Os pulsos são gerados pelo DFCM desse modo de forma a alinhar o explorador ao alvo e seguir em linha reta em direção ao mesmo, em uma velocidade variável de acordo com a distância entre os dois.

Como o robô deve sempre garantir sua integridade, mesmo quando um alvo é encontrado e este está indo capturá-lo, o robô deve priorizar o desvio do obstáculo e, por isso, o Modo de Operação 2 pode ser ativado mesmo quando o robô encontra-se no Modo de Operação 3. Assim, o robô deixa de capturar seu alvo, que volta a ficar desconhecido, e prioriza o desvio.

Cada alvo só é capturado pelo explorador que o encontrou, assim, os demais indivíduos ficam responsáveis por desviar do explorador que estiver nesse modo.

3.2.2.5. Síntese dos modos de operação

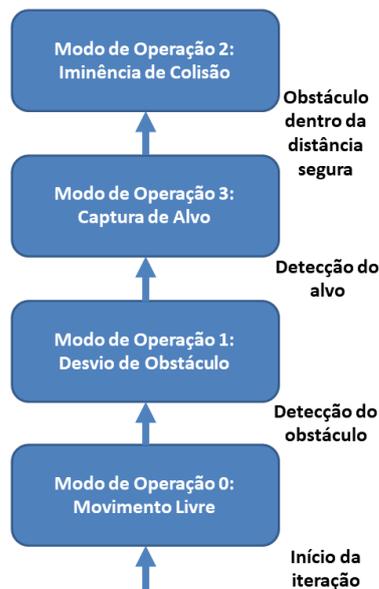
Durante a execução da sua tarefa, o robô deve buscar cumprir seus objetivos, garantindo sempre sua integridade. Devido a isso, os Modos de Operação foram

organizados de forma hierárquica como é visto na Figura 13. A partir disso foi possível implementar e conferir seu funcionamento, de acordo com a máquina de estados que é apresentado na Figura 14 e com seus eventos descritos na Tabela 2.

O Modo Iminência de Colisão é o de maior prioridade seguido do Modo Captura de Alvos, depois o de Desvio de Obstáculos e por fim o de Movimento Livre. Essa sequência define quais são as prioridades do explorador ao longo da simulação:

1. Garantir integridade do robô;
2. Buscar e capturar alvos;
3. Explorar e mapear o ambiente.

Figura 13 - Síntese da alternância entre os modos de operação

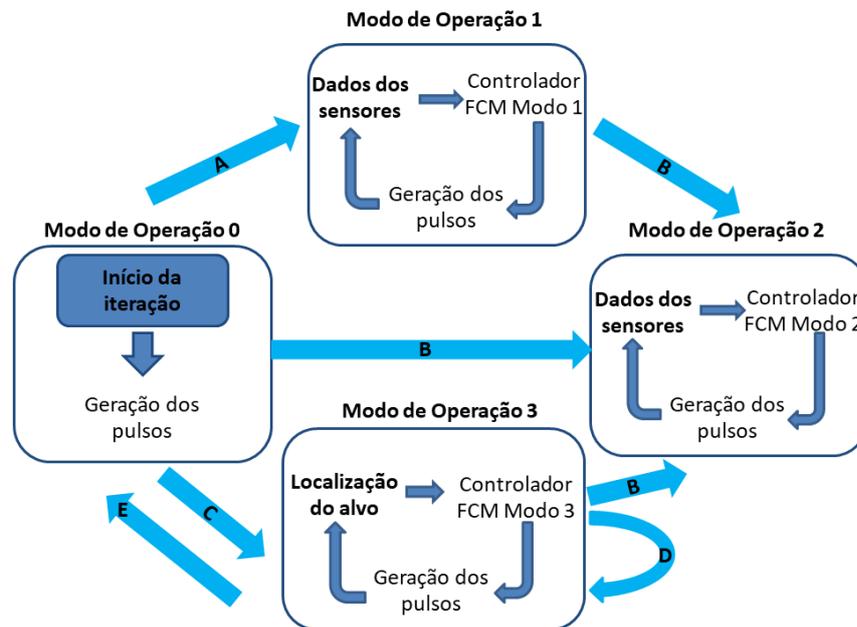


Fonte: Retirado de SOARES *et al.* (2017).

Tabela 2 - Eventos da Máquina de Estado

Evento	Descrição
A	Obstáculo detectado pelo sensor a uma distância maior que 4m
B	Obstáculo detectado pelo sensor a uma distância menor que 4m
C	Alvo encontrado
D	Alvo sendo resgatado
E	Algo resgatado

Figura 14 - Máquina de Estados



Fonte: Autoria Própria

3.2.3. DFCM

Como explicado, os conceitos utilizados no FCM são criados através do estudo do sistema por especialistas, adicionando e ajustando-os caso seja preciso. Assim o FCM foi desenvolvido utilizando as entradas de cada sensor (Sensor Direito, Sensor Frontal e Sensor Esquerdo) e a saída de cada motor (Motor Direito e Motor Esquerdo), podendo ser visto na Figura 15.

Os conceitos de entrada Sensor Direito (SD) é o conceito 1 (C1); o Sensor Frontal (SF) é o 2 (C2) e o Sensor Esquerdo (SE) é o 3 (C3). Já os conceitos de saída Motor Direito (MD) é o conceito 4 (C4) e o Motor Esquerdo (ME) foi definido como o 5 (C5). Cada um desses conceitos é interligado pelos arcos que são as relações de causa-efeito que possuem entre si. Essa relação é representada pelas flechas em vermelho na Figura 15. A direção da flecha determina a orientação da influência e o sinal que encontra dentro da flecha é o tipo de relação, se este possui uma relação causal negativa (quando um conceito aumenta, o outro diminui) ou positiva (quando um aumenta, o outro também aumenta) (KOK, 2009).

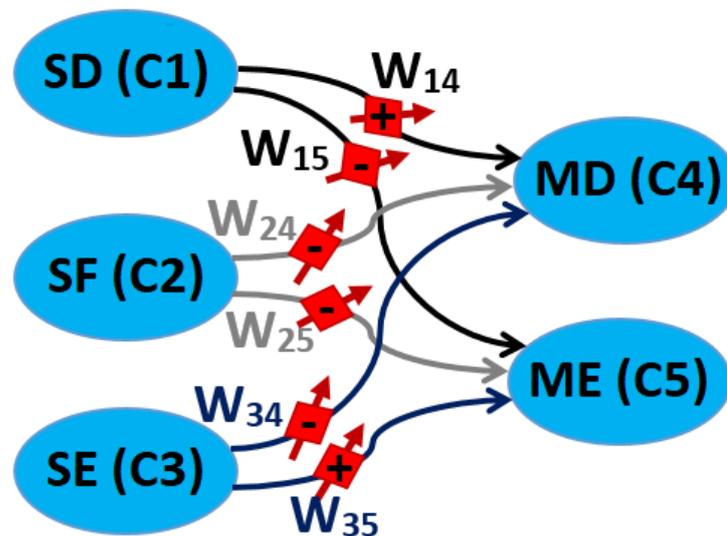
Como os sensores são os conceitos de entrada, eles influenciam o valor dos motores, que são os conceitos de saída. Por isso o sentido das flechas vai dos sensores para os motores.

Como descrito na fundamentação teórica, o FCM pode ser dinâmico caso ele seja alterado através de eventos. Assim, para esse trabalho, o DFCM utilizado é um sistema dinâmico que se altera diante dos estados do robô.

Para os Modos de Operação 1 e 2, ao detectar um objeto pelo Sensor Direito, o objetivo do DFCM é fazer com que o robô mova para direita, para isso, como apresentado no modelo cinemático, o Motor Direito deve receber um pulso maior que o Motor Esquerdo. Devido a isso a relação causal do SD com o MD é positivo e representado pelo arco W_{14} , enquanto a relação causal do SD com o ME é negativo, visto no arco W_{15} . O mesmo acontece com a relação causal de SE com ME (W_{35}), que é positivo e SE com MD (W_{34}), que é negativo.

A relação causal do Sensor Frontal é diferente, isto por que ao encontrar com um obstáculo o robô deve diminuir sua velocidade. Devido a isso, sua relação causal tanto com o MD (W_{24}), quanto com o ME (W_{25}) é negativo.

Figura 15 - FCM dos Modos de Operação 1 e 2



Fonte: Autoria Própria

Como o FCM é baseado na teoria dos grafos e operações com matrizes, podemos representar o grafo apresentado na Figura 15 através de uma matriz, assim temos a matriz (Equação 17):

$$W = \begin{bmatrix} - & \mathbf{SD} & \mathbf{SF} & \mathbf{SE} & \mathbf{MD} & \mathbf{ME} \\ \mathbf{SD} & w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ \mathbf{SF} & w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ \mathbf{SE} & w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \\ \mathbf{MD} & w_{41} & w_{42} & w_{43} & w_{44} & w_{45} \\ \mathbf{ME} & w_{51} & w_{52} & w_{53} & w_{54} & w_{55} \end{bmatrix} \quad (17)$$

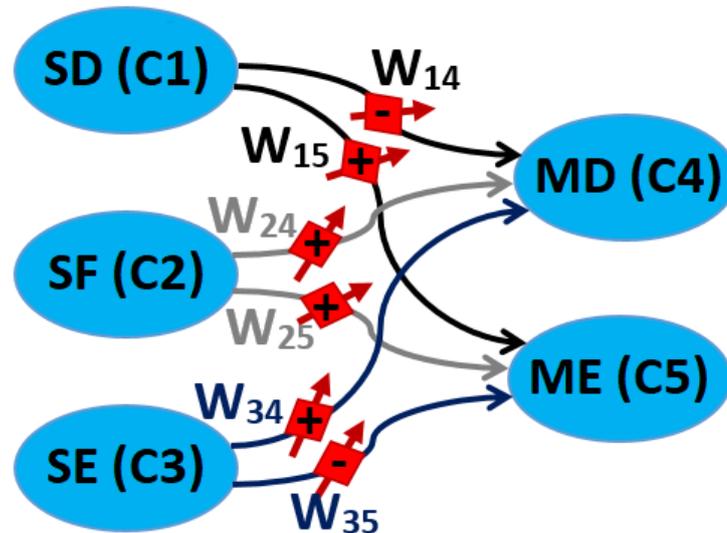
Como nem todos os conceitos possuem relação entre si, podemos simplificar a matriz de acordo com o grafo apresentado na Figura 15. Foi colocado a relação causal positiva ou negativa, sendo assim:

$$W = \begin{bmatrix} - & \mathbf{SD} & \mathbf{SF} & \mathbf{SE} & \mathbf{MD} & \mathbf{ME} \\ \mathbf{SD} & 0 & 0 & 0 & +w_{14} & -w_{15} \\ \mathbf{SF} & 0 & 0 & 0 & -w_{24} & -w_{25} \\ \mathbf{SE} & 0 & 0 & 0 & -w_{34} & +w_{35} \\ \mathbf{MD} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{ME} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

Já para o Modo de Operação 3 - Captura de Alvos, diferentemente dos outros dois modos, quando o explorador localiza um alvo, seu objetivo é capturá-lo, ou seja, ele deve ir ao encontro do alvo e não desviar, como acontece com os obstáculos. Devido a isso a relação causal dos sensores é inversa. Podemos ver seu grafo na Figura 16 e sua matriz na Equação 19.

- SD com MD: relação causal negativo;
- SD com ME: relação causal positivo;
- SF com MD: relação causal positivo;
- SF com ME: relação causal positivo;
- SE com MD: relação causal positivo;
- SE com ME: relação causal negativo;

Figura 16 - FCM dos Modos de Operação 3



Fonte: Autoria Própria

$$W = \begin{bmatrix} - & SD & SF & SE & MD & ME \\ SD & 0 & 0 & 0 & -w_{14} & +w_{15} \\ SF & 0 & 0 & 0 & +w_{24} & +w_{25} \\ SE & 0 & 0 & 0 & +w_{34} & -w_{35} \\ MD & 0 & 0 & 0 & 0 & 0 \\ ME & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (19)$$

Nesse trabalho, O DFCM utilizado muda de acordo com o Modo de Operação do agente. A matriz de peso para cada conceito foi desenvolvido a partir do conhecimento dos especialistas e com refinamento empírico de acordo com as simulações feitas.

Assim, o DFCM para cada um dos três Modos de Operação que utilizam do DFCM para mandar os pulsos aos motores do robô são:

- Modo 1 – Desvio de Obstáculos;

$$W = \begin{bmatrix} - & SD & SF & SE & MD & ME \\ SD & 0 & 0 & 0 & +0,6 & -0,2 \\ SF & 0 & 0 & 0 & -0,3 & -0,3 \\ SE & 0 & 0 & 0 & -0,2 & +0,6 \\ MD & 0 & 0 & 0 & 0 & 0 \\ ME & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (20)$$

- Modo 2 – Iminência de Colisão;

$$W = \begin{bmatrix} - & \mathbf{SD} & \mathbf{SF} & \mathbf{SE} & \mathbf{MD} & \mathbf{ME} \\ \mathbf{SD} & 0 & 0 & 0 & +0,8 & -0,6 \\ \mathbf{SF} & 0 & 0 & 0 & -0,8 & -0,8 \\ \mathbf{SE} & 0 & 0 & 0 & -0,6 & +0,8 \\ \mathbf{MD} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{ME} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (21)$$

- Modo 3 – Captura de Alvos

$$W = \begin{bmatrix} - & \mathbf{SD} & \mathbf{SF} & \mathbf{SE} & \mathbf{MD} & \mathbf{ME} \\ \mathbf{SD} & 0 & 0 & 0 & -0,4 & +0,5 \\ \mathbf{SF} & 0 & 0 & 0 & +0,3 & +0,3 \\ \mathbf{SE} & 0 & 0 & 0 & +0,5 & -0,4 \\ \mathbf{MD} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{ME} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (22)$$

3.3. Ambiente de teste

Neste trabalho, como mencionado, para que o robô seja considerado autônomo, este deve ser simulado diante de pelo menos três ambientes diferentes (MAES, 1995; MATARIC, 2007; RUSSELL e NORVIG, 2013), completando seu objetivo de forma eficiente em todos os ambientes propostos.

Os três ambientes de teste utilizados foram os mesmo utilizados nos testes de SOARES (2017). Esses ambientes são delimitados por uma área de dimensão 100 x 100 unidades (considerado 1 unidade = 1 metro), considerado a área de busca dos exploradores. Dentro de cada ambiente são apresentados três elementos: a arena, os obstáculos fixos e os alvos. Os obstáculos dinâmicos (outros robôs) são encontrados somente nas simulações com mais de um robô.

A arena é a área de busca dos robôs, por isso já é inicialmente conhecida e é representado por pontos pretos.

Os obstáculos fixos que, são os únicos a mudarem de posição em cada ambiente, são inicialmente desconhecidos pelos robôs e denotados por pontos azuis. Porém, a partir do momento que os exploradores mapeiam os obstáculos fixos, estes se tornam parte do mapa e por isso, se tornam pontos pretos. O mapeamento completo do mapa muitas vezes não é totalmente completado devido

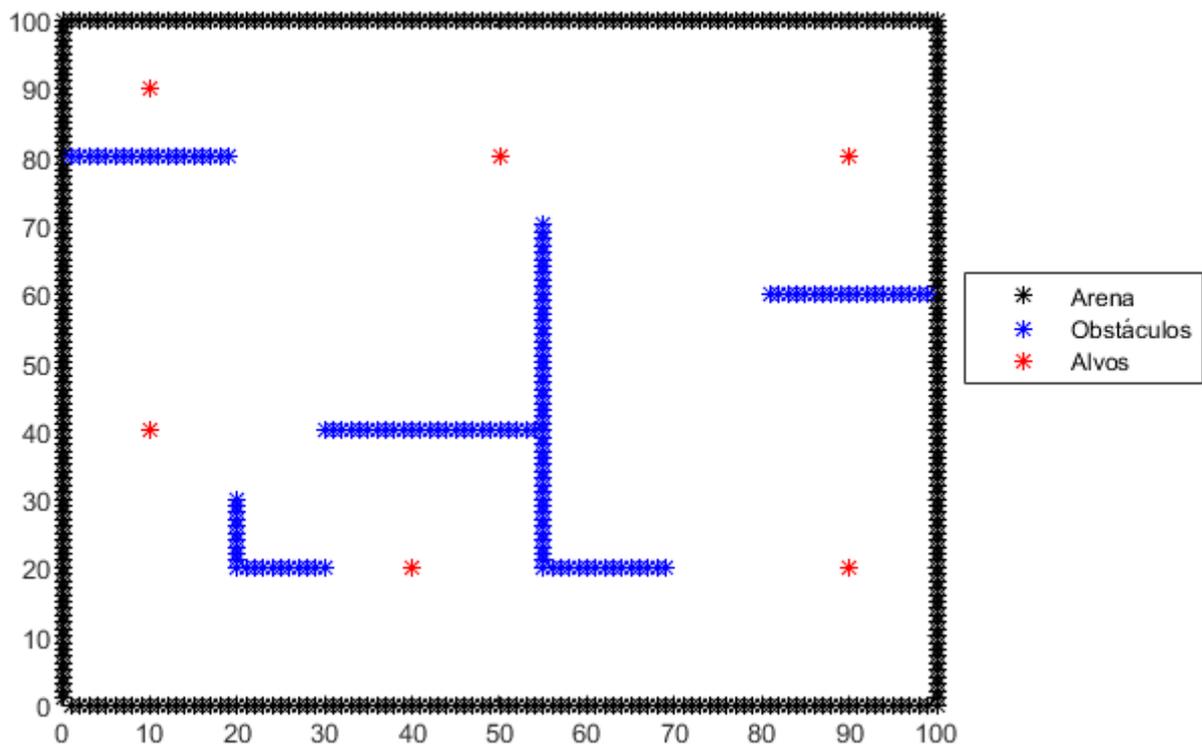
ao objetivo principal ser encontrar todos os alvos.

Os alvos que são representados pelos seis pontos vermelhos, são dispostos nas mesmas posições nos três ambientes e quando encontrados por algum robô, tornam-se verdes. As posições dos alvos são (10 40), (10 90), (40 20), (50 80), (90 20) e (90 80).

Já os obstáculos dinâmicos são detectados através dos sensores juntamente com as posições fornecidas entre os indivíduos. Assim como a sua posição atual, os robôs fornecem entre si a posição dos obstáculos estáticos e a quantidade de alvos capturados.

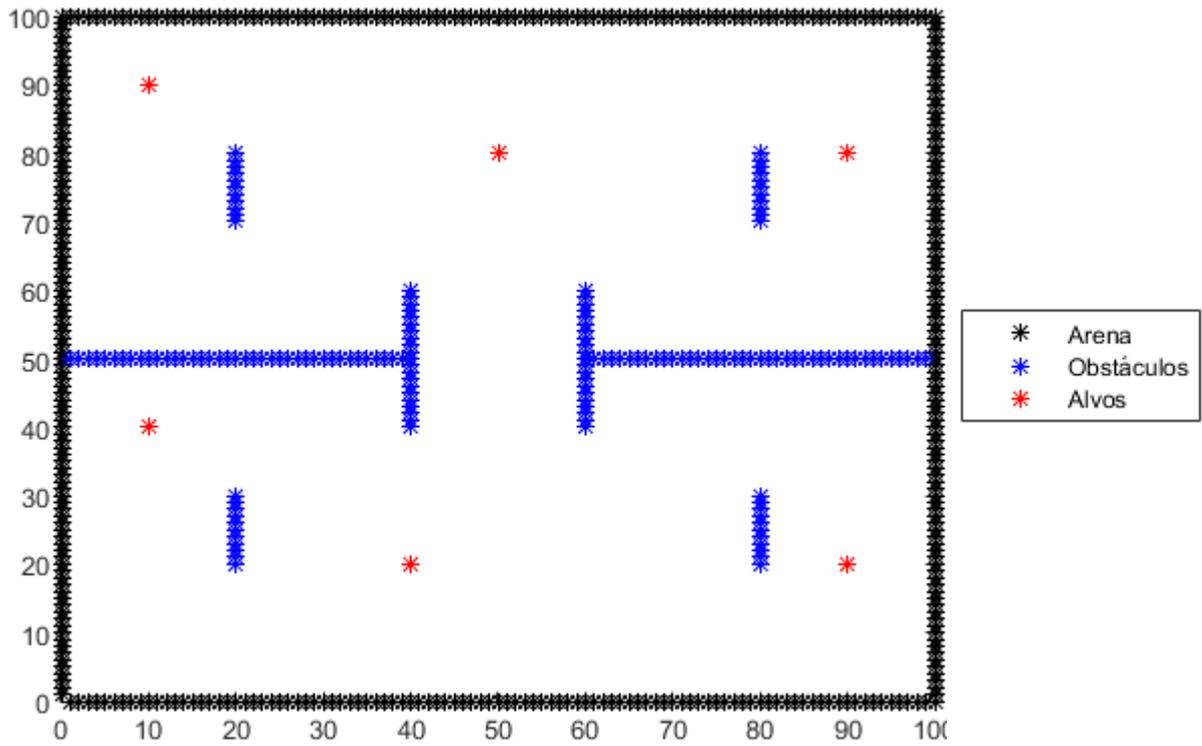
Os ambientes de teste são apresentados nas Figuras 17, 18 e 19:

Figura 17 - Ambiente de simulação I



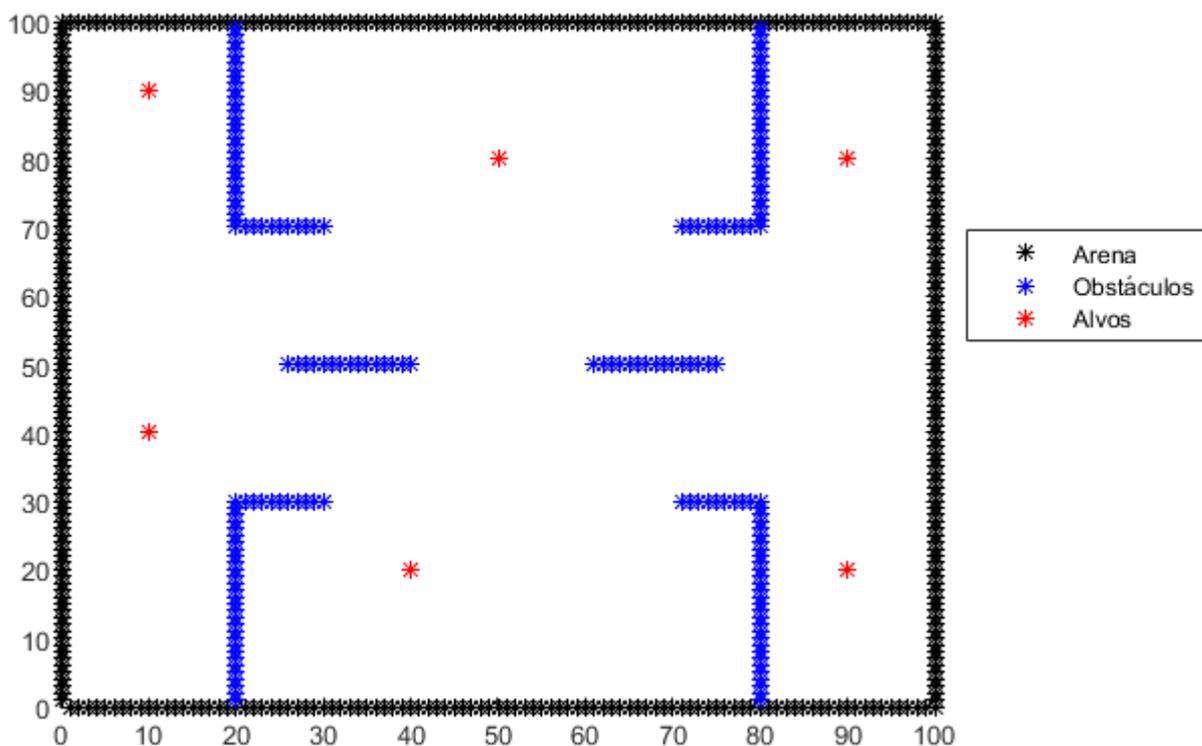
Fonte: Autoria Própria

Figura 18 - Ambiente de simulação II



Fonte: Autoria Própria

Figura 19 - Ambiente de simulação III



Fonte: Autoria Própria

A posição de cada robô, sua disposição e sua cor em relação ao ambiente foi definida como padrão para todos os ambientes. Para os testes com um único explorador este se encontra na posição cartesiana [40 10], direção 0° e cor preta. Já para a simulação dos testes com quatro robôs:

- Robô 1: posição cartesiana [40 10], direção 0° e cor preta;
- Robô 2: posição cartesiana [60 10], direção 180° e cor azul escuro;
- Robô 3: posição cartesiana [90 40], direção 90° e cor azul claro;
- Robô 4: posição cartesiana [90 70], direção 270° e cor magenta;

3.4. Simulação

O *software* utilizado para a realização das simulações foi o *Matlab*® e para efeito de comparação, todos os testes foram realizados no computador SAMSUNG® com processador Intel® Core™ i5-5200 CPU@2.20GHz e 8,00GB de memória RAM.

3.4.1. Parâmetros de teste

Para cada ambiente foi realizado inicialmente um teste utilizando apenas um robô e, posteriormente com quatro robôs. Esses testes foram realizados tanto para o DFCM, como para o FLC devido a algumas alterações na estrutura da simulação feita no trabalho anterior (SOARES, 2017). A estrutura *Fuzzy* desenvolvida por SOARES foi mantida.

Foram aferidos o tempo de processamento da simulação com o auxílio dos comandos *tic* e *toc* e o tempo de vídeo, simulando um parâmetro de tempo real utilizando as funções *getframe* e *VideoWrite* com trinta quadros por segundo. Foi gerado um gráfico com a relação dos pulsos gerados ao longo das simulações.

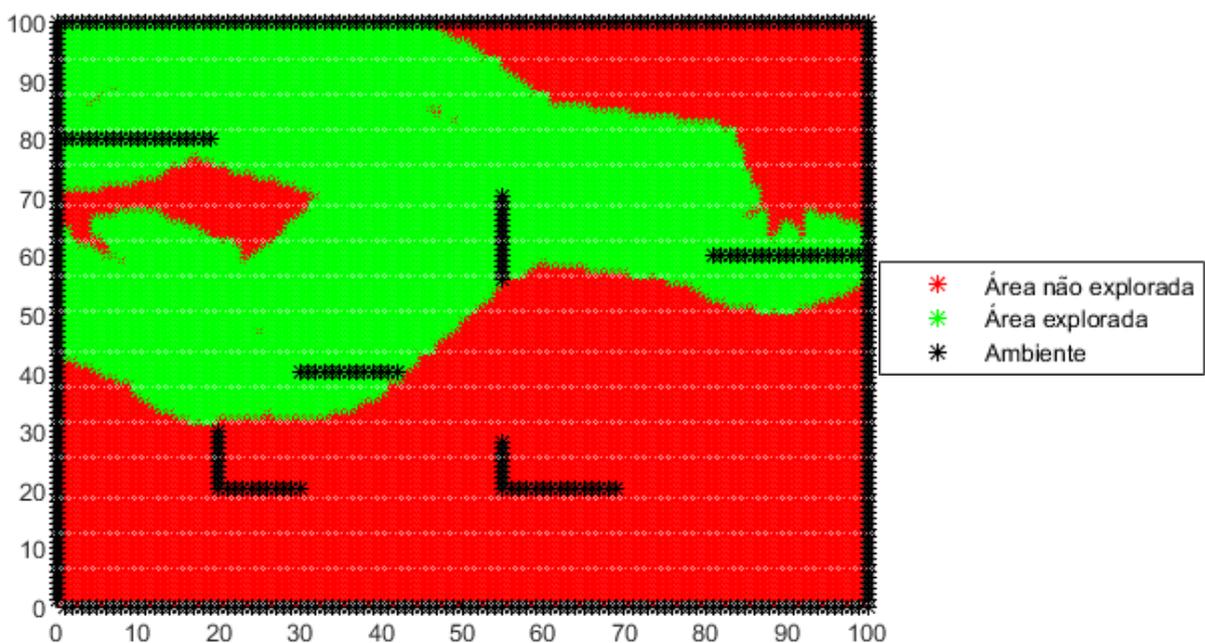
Além do tempo de processamento, principal comparação, outros dados foram utilizados para comparar os resultados:

- Iterações: Foi utilizado o número de iterações necessário para completar cada cenário (Eficiência do tempo para executar a tarefa).
- Distância Percorrida: Através do modelo cinemático do robô, foi possível criar uma variável responsável por contar a distância percorrida por cada robô e

assim comparar entre eles no mesmo ambiente e com controladores diferentes. Foi criada outra variável responsável por juntar todas as distâncias percorridas (Eficiência no gasto de recurso).

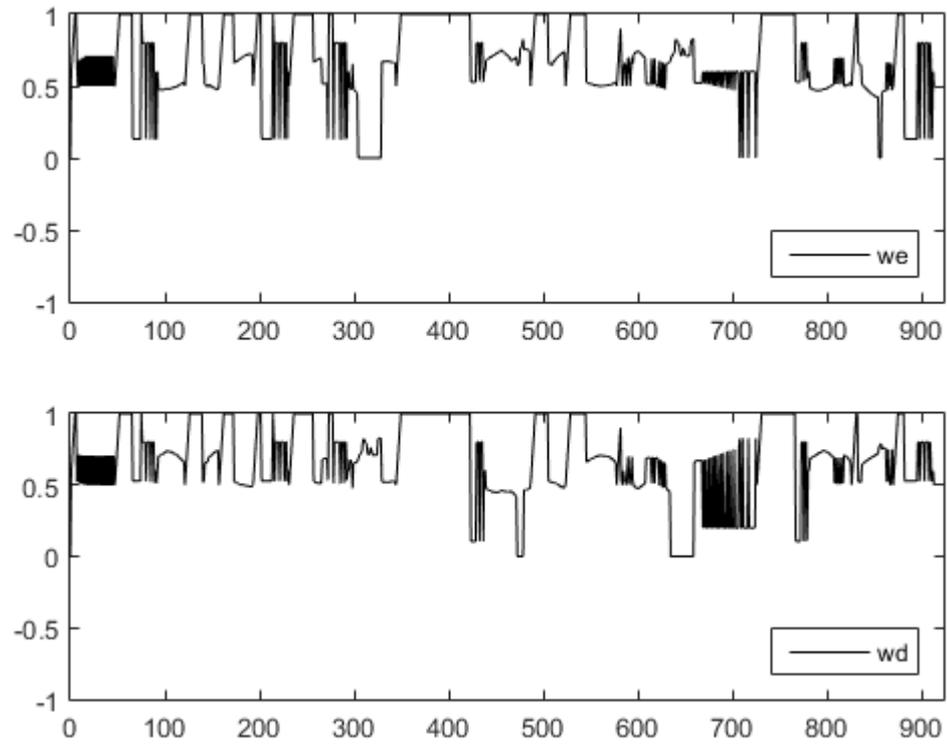
- **Área Explorada:** A partir de mapeamento foi desenvolvido outra função, a da área explorada pelo robô. Como o mapa é restringido em uma área de 100 x 100 m (10.000 m²), foi criada uma variável responsável por guardar parte do ambiente não explorado, ou seja, todos os 10.000 pontos cartesianos. A cada iteração do robô, é verificado se a região dos sensores foi mapeado e em caso negativo, ele guarda essa posição em outra variável, a da área já explorada. Isso é feito para cada robô e também para todos os robôs juntos para obter sua exploração total. Não é possível somente juntar a área explorada individual, pois os robôs podem explorar o mesmo ambiente. Ao final, temos os valores em metros quadrados da área explorada por cada robô e o conjunto. Na figura 20, podemos ver um exemplo da área que não foi explorada, representada pelos pontos em vermelho, a área já explorada, pontos em verde e os obstáculos mapeados, mostrados em pontos pretos. Lembrando que os robôs compartilham a posição dos obstáculos estáticos.

Figura 20 - Exemplo de área explorada e não explorada.



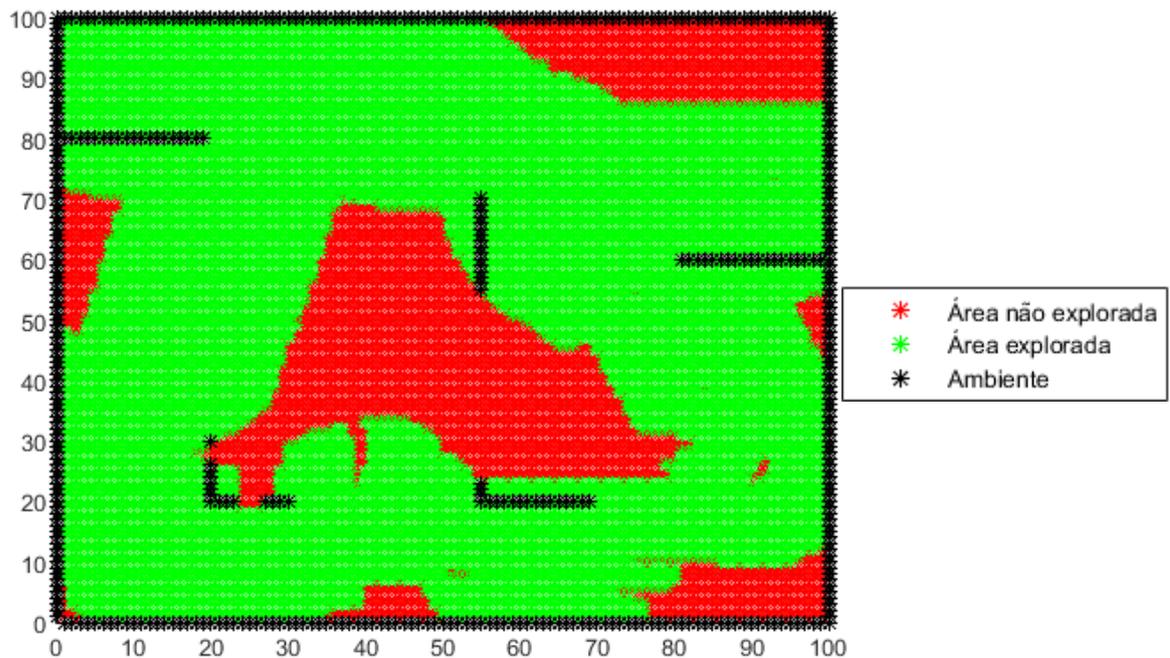
Fonte: Autoria Própria

Figura 22 - Pulsos do robô na simulação do ambiente I.



Fonte: Autoria Própria

Figura 23 - Área explorada pelo robô no ambiente I.



Fonte: Autoria Própria

Já para o ambiente I utilizando quatro robôs, o resgate de todas as vítimas

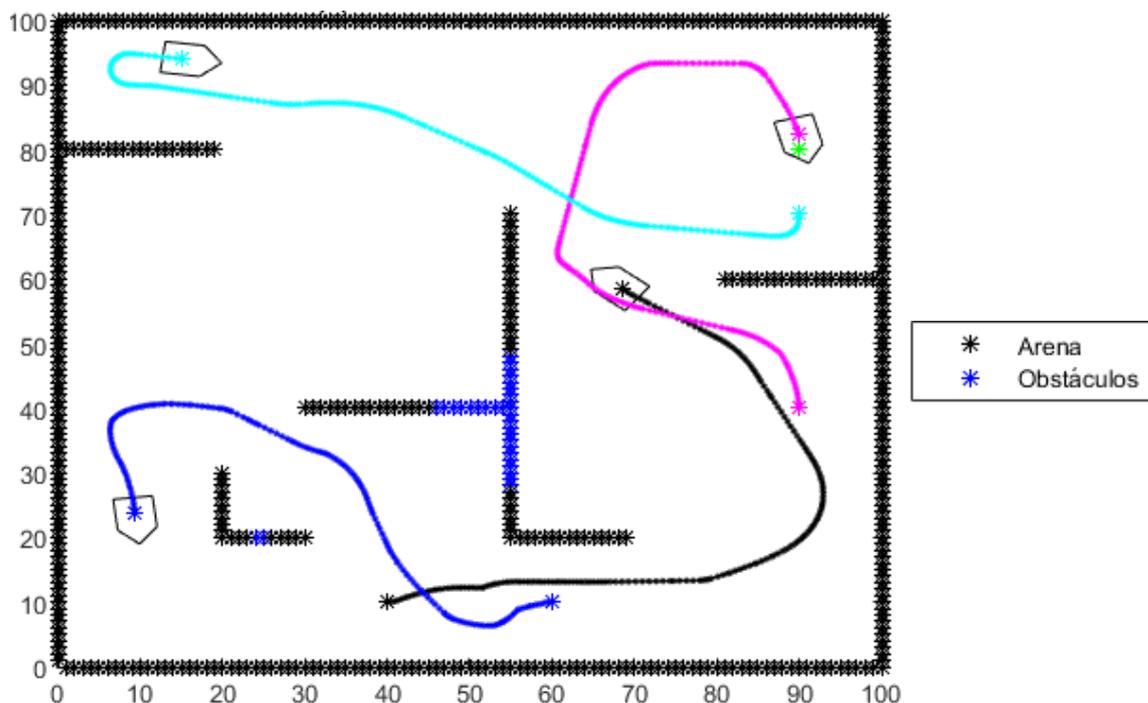
durou 185 iterações, levando um tempo de processamento de 7 minutos e 22 segundos e gerando um vídeo de 6 segundos. O vídeo gerado pode ser acessado pelo link: <<https://youtu.be/LDdZCCDeCYo>>.

O comportamento de cada robô no ambiente pode ser visto na Figura 24 e os pulsos gerados pelo DFCEM para cada robô na Figura 25. Quando um robô encontra outro (obstáculo móvel) ele recebe pulsos negativos para virar e evitar a colisão. Podemos observar esse comportamento nos robôs 1, 2 e 3, sendo as regiões marcadas em vermelho na Figura 25.

O explorador I percorreu uma distância de 101,2969 m e explorou uma área de 2486 m². O explorador II percorreu uma distância de 86,6182 m e explorou uma área de 2210 m². O explorador III percorreu uma distância de 97,7908 m e explorou uma área de 2437 m². O explorador IV percorreu uma distância de 102,9609 m e explorou uma área de 2493 m².

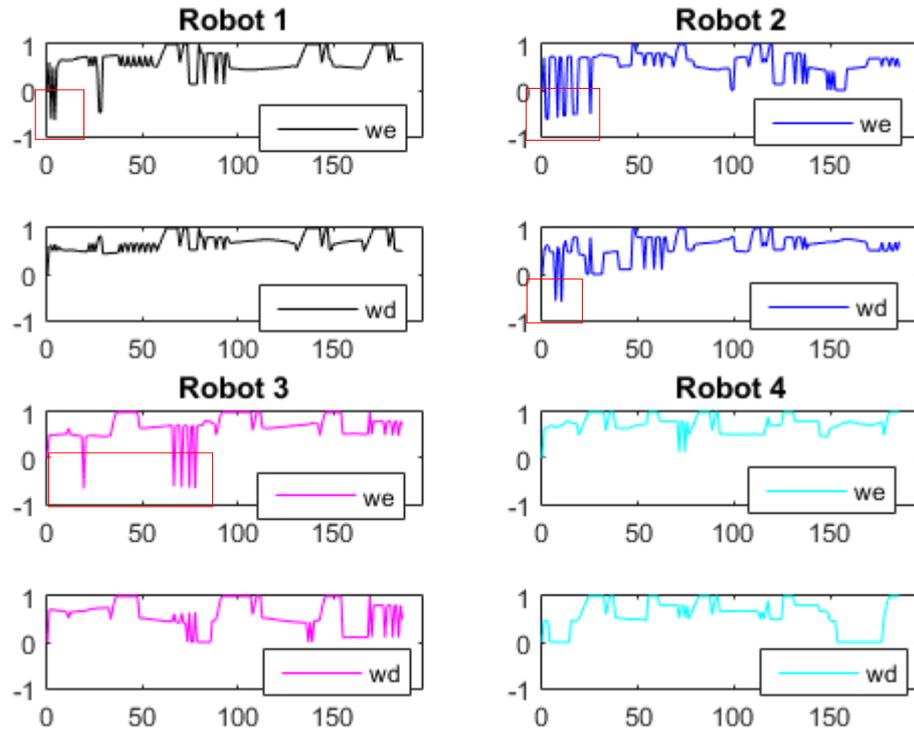
Portanto os robôs percorreram uma distância total de 388,668 m e sua área explorada coletivamente foi de 7498 m², que representa 74,95% da área de busca. A área explorada por cada robô está representada na Figura 26 e a área total explorada pela Figura 27.

Figura 24 - Simulação do ambiente I com 4 robôs.



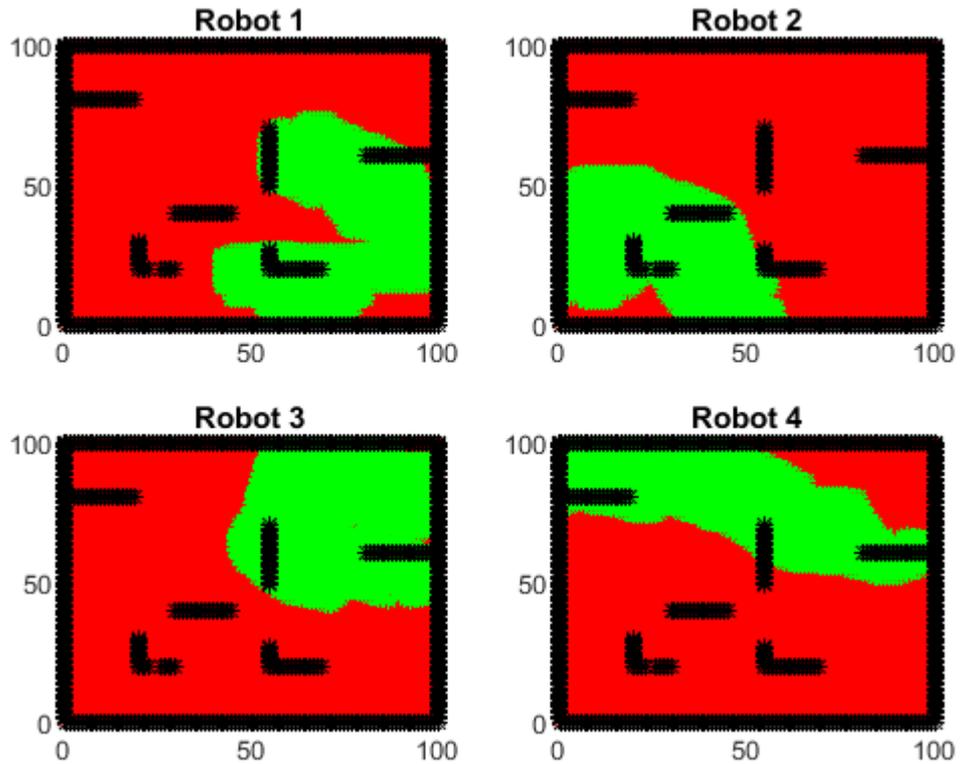
Fonte: Autoria Própria

Figura 25 - Pulsos dos robôs na simulação do ambiente I.



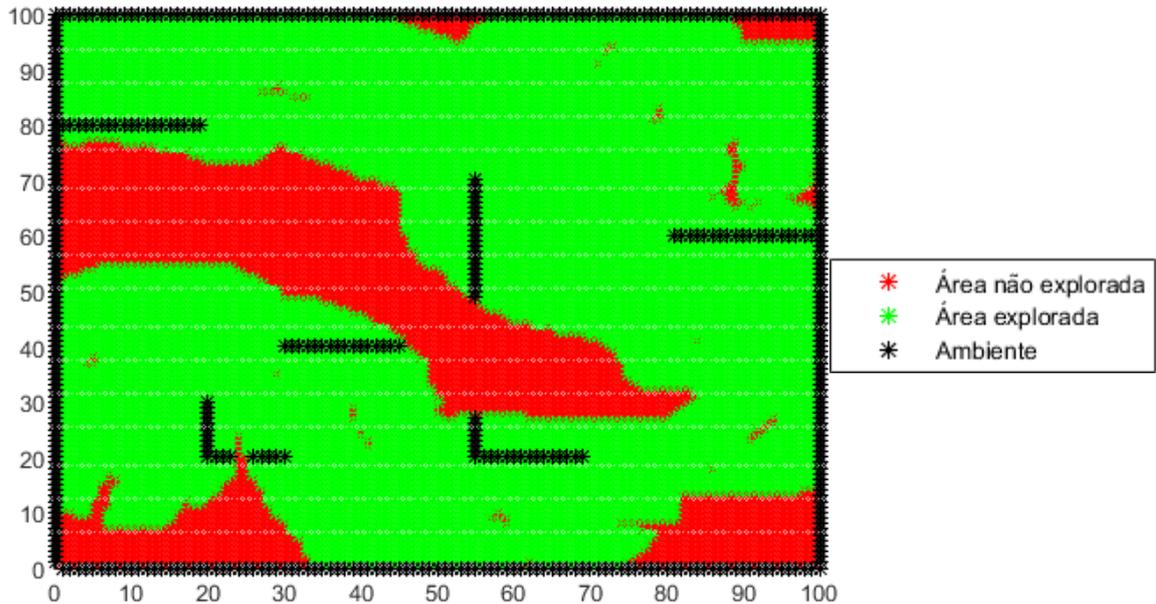
Fonte: Autoria Própria

Figura 26 - Área explorada pelos robôs no ambiente I.



Fonte: Autoria Própria

Figura 27 - Área total explorada pelos robôs no ambiente I.



Fonte: Autoria Própria

Para o primeiro ambiente podemos perceber que para a simulação com um robô o DFCM teve um tempo de processamento menor que o do FLC, porém devido seu objetivo principal ser o resgate dos alvos, a área total explorada foi menor. A distância percorrida pelo robô é maior e com isso podemos perceber que o movimento do robô com DFCM é mais polido, visto que, por exemplo, para que o robô busque o alvo após o encontrar, o *Fuzzy* movimenta o explorador em seu eixo até que este fique de frente com o alvo e siga reto em direção a este. Já para o DFCM, devido ao polimento do movimento que este faz o agente continua seu movimento fazendo os devidos ajustes de curva.

Para o ambiente com quatro exploradores, o DFCM apresentou uma grande vantagem, visto que cumpriu seu objetivo com quase metade das iterações necessária ao FLC. Devido a isso, a distância total percorrida pelos robôs e sua área de exploração total foi menor em relação à simulação utilizando *Fuzzy*.

Comparando o tempo do processamento utilizando um e quatro robôs, o com maior número de robôs apresentou maior tempo principalmente em consequência da simulação da área explorada por cada robô, visto que é necessário criar uma variável para cada agente. Esses resultados podem ser visualizados nas tabelas a seguir (Tabelas 3, 4 e 5).

Tabela 3 - Comparação do tempo DFCM e FLC Ambiente I

Robôs	Controlador	Iterações	Tempo
1	DFCM	912	7'15"
1	FLC	976	7'36"
4	DFCM	185	7'22"
4	FLC	346	11'20"

Tabela 4 - Comparação Distância Percorrida Ambiente I

Distância Percorrida						
Robôs	Controlador	Robô 1	Robô 2	Robô 3	Robô 4	Total
1	DFCM	508,8717	-	-	-	508,8717
1	FLC	508,7000	-	-	-	508,7000
4	DFCM	101,2969	86,6182	97,7908	102,9609	388,6668
4	FLC	233,4808	178,9572	181,6575	203,2476	797,3431

Tabela 5 - Comparação Área Explorada Ambiente I

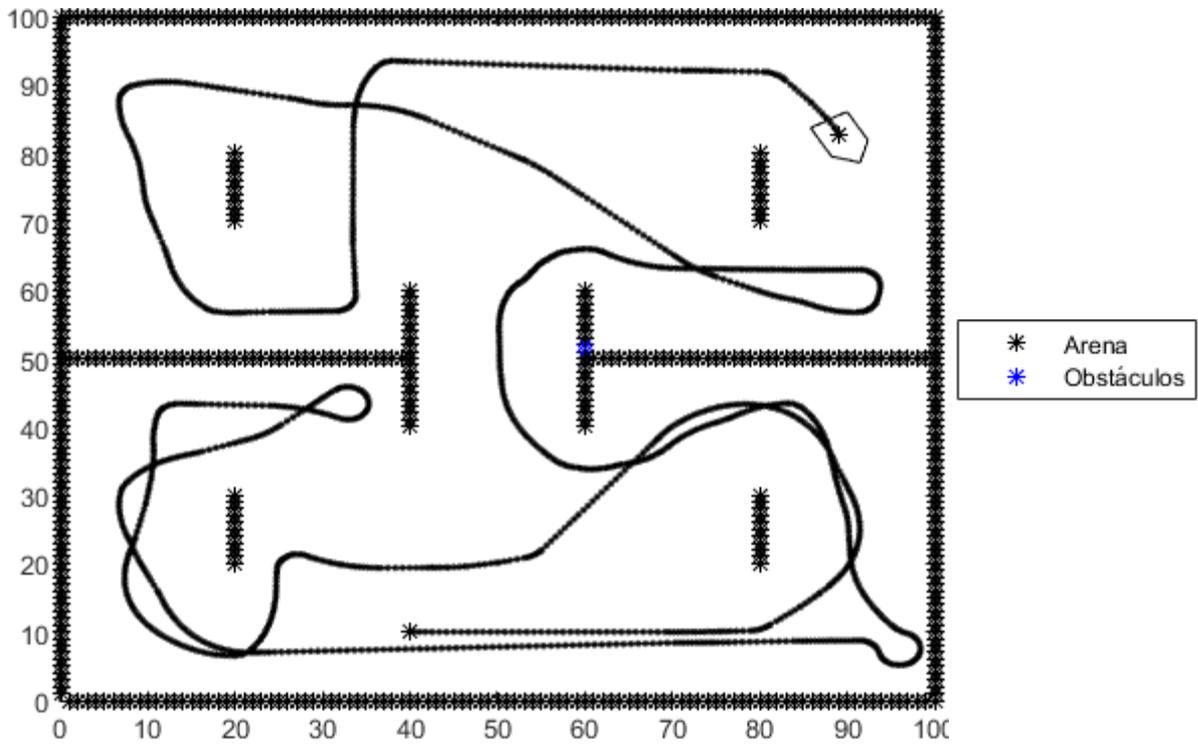
Área Explorada						
Robôs	Controlador	Robô 1	Robô 2	Robô 3	Robô 4	Total
1	DFCM	8073	-	-	-	8073
1	FLC	8374	-	-	-	8374
4	DFCM	2486	2210	2553	2493	7498
4	FLC	4465	4160	4352	4110	9026

4.2. Resultados do ambiente de teste II

Para o ambiente II utilizando somente um robô, o resgate de todas as vítimas durou 1466 iterações, tendo um tempo de processamento de 9 minutos e 50 segundos e gerando um vídeo de 48 segundos. Sua simulação no ambiente pode ser visto na Figura 28 e o vídeo gerado pode ser acessado pelo link: <https://youtu.be/OUuiDz_8Fbw>.

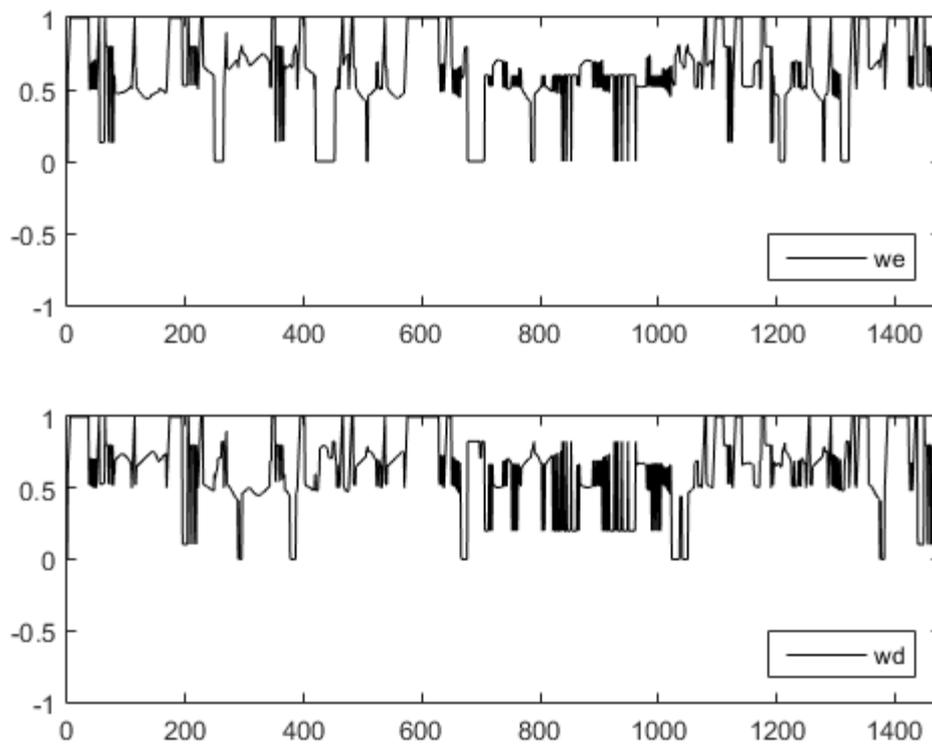
O explorador percorreu uma distância de 763,9753 m, explorando uma área de 9421 m², que corresponde a 94,21% da área total. Essa área pode ser vista na Figura 30.

Figura 28 - Simulação do ambiente II com 1 robô.



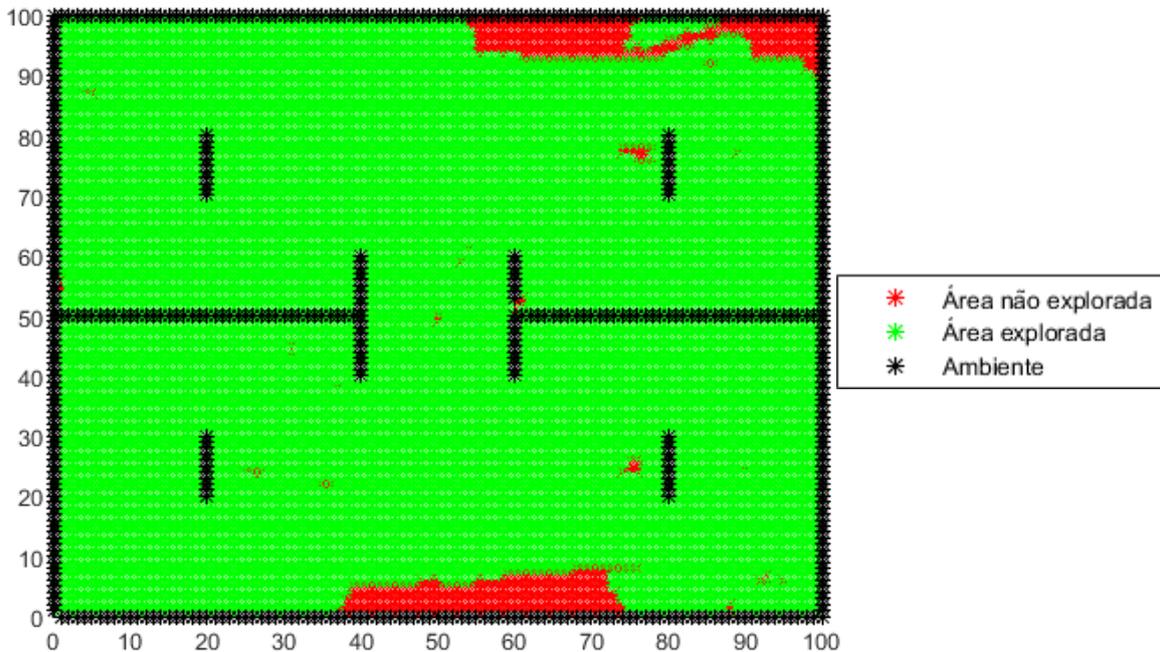
Fonte: Autoria Própria

Figura 29 - Pulsos do robô na simulação do ambiente II.



Fonte: Autoria Própria

Figura 30 - Área explorada pelo robô no ambiente II.



Fonte: Autoria Própria

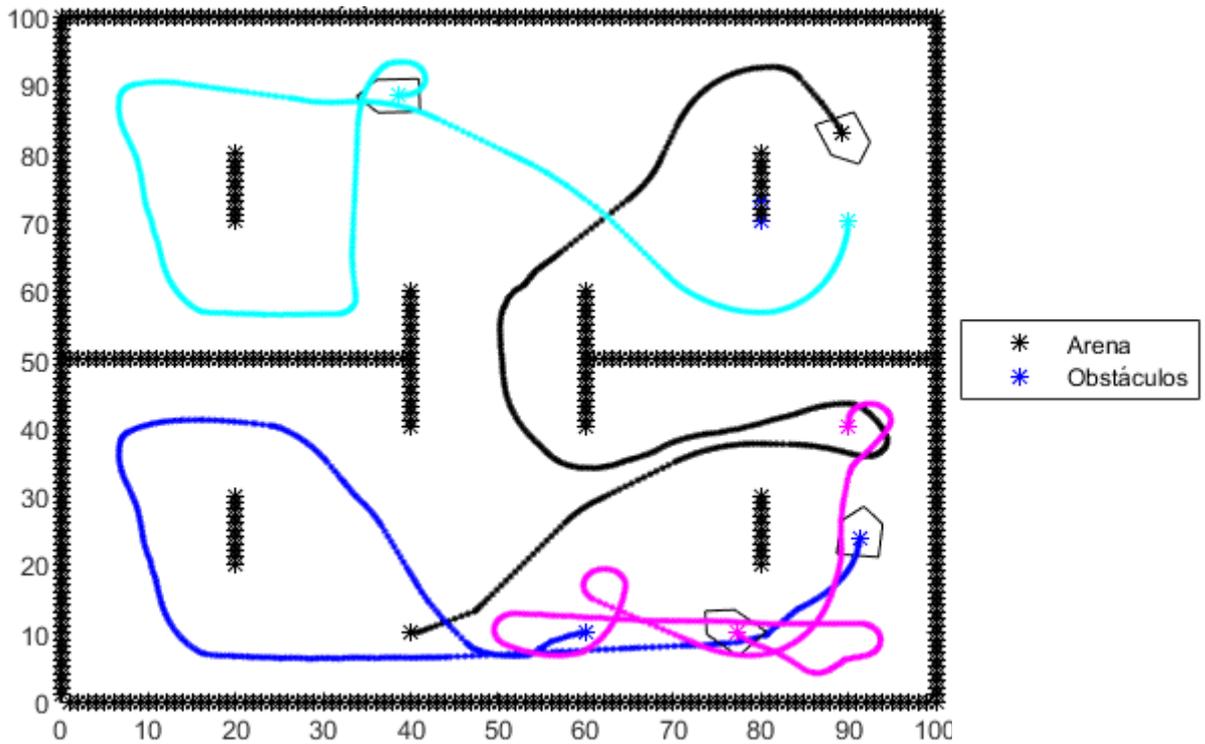
Já para o ambiente II utilizando quatro robôs, o resgate de todas as vítimas durou 385 iterações, tendo um tempo de processamento de 12 minutos e 40 segundos e gerando um vídeo de 12 segundos. O vídeo gerado pode ser acessado pelo link: <<https://youtu.be/TuLz0Y-Pbic>>.

Os pulsos gerados pelo DFCM para cada robô na Figura 32, sendo as regiões marcadas em vermelho as iterações que encontram com outros robôs.

O explorador I percorreu uma distância de 194,3038 m e explorou uma área de 4048 m². O explorador II percorreu uma distância de 188,5215 m e explorou uma área de 3593 m². O explorador III percorreu uma distância de 169,2457 m e explorou uma área de 2437 m². O explorador IV percorreu uma distância de 201,4758 m e explorou uma área de 4141 m².

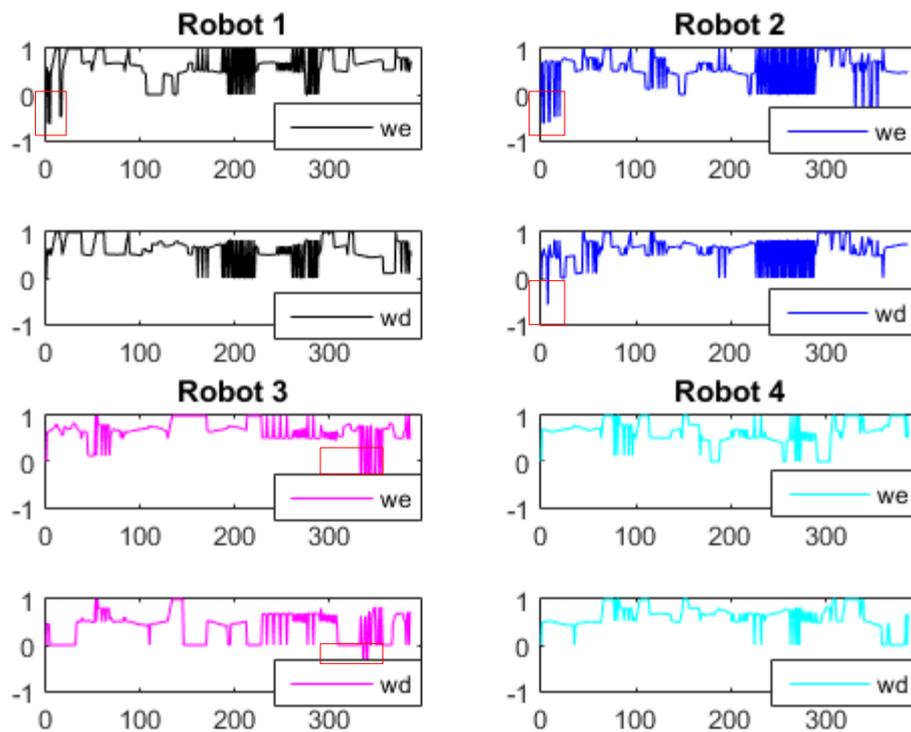
Portanto os robôs percorreram uma distância total de 753,5468 m e sua área explorada coletivamente foi de 9743 m², que representa 97,43% da área de busca. A área explorada por cada robô está representada na Figura 33 e a área total explorada pela Figura 34.

Figura 31 - Simulação do ambiente II com 4 robôs.



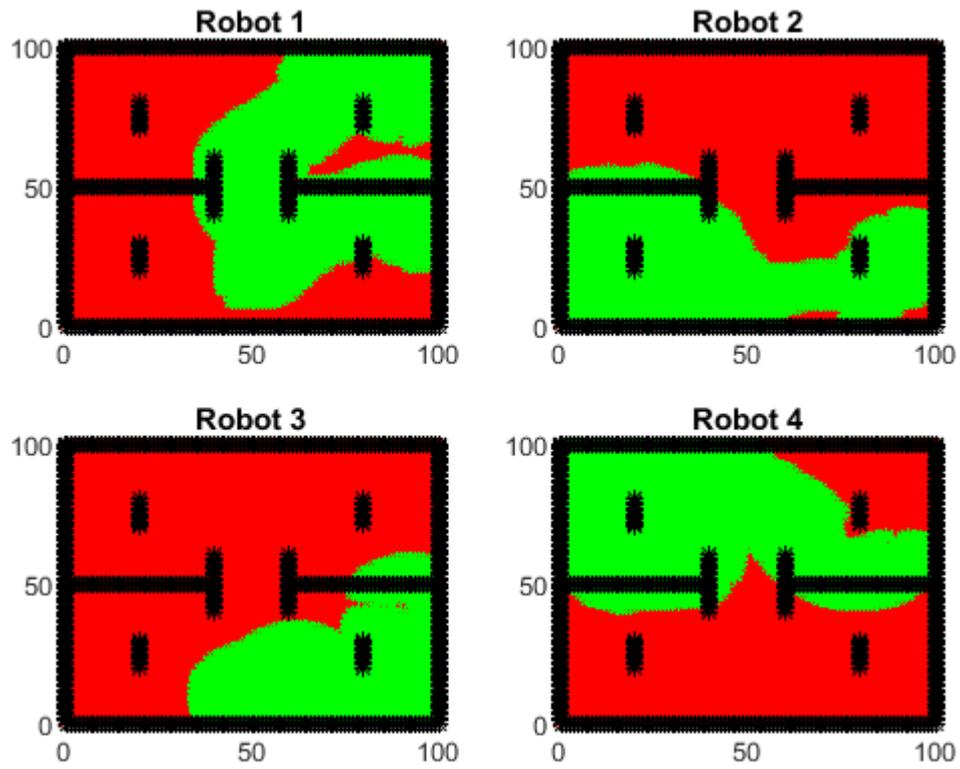
Fonte: Autoria Própria

Figura 32 - Pulsos dos robôs na simulação do ambiente II.



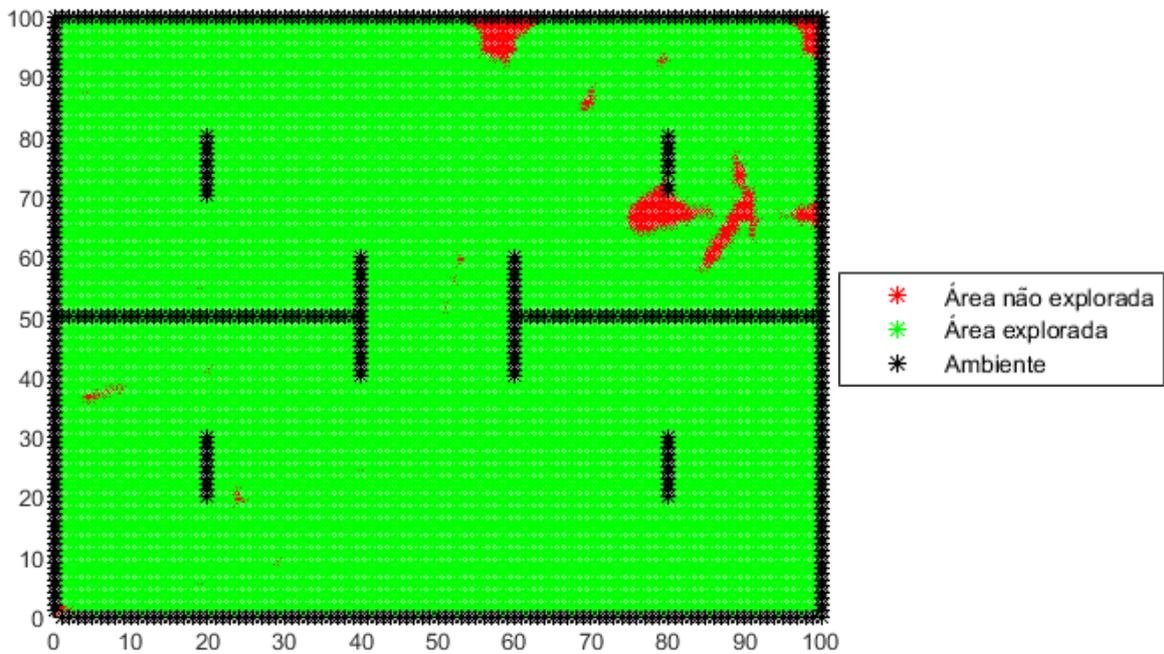
Fonte: Autoria Própria

Figura 33 - Área explorada pelos robôs no ambiente II.



Fonte: Autoria Própria

Figura 34 - Área total explorada pelos robôs no ambiente II.



Fonte: Autoria Própria

O ambiente II com um robô possui um corredor ao centro que aumenta a

dificuldade na simulação. Devido a isso, o robô utilizou-se da função implementada, na qual o agente muda de direção caso esteja no Modo de Operação 0 – Movimento Livre. Para o ambiente com um robô, a simulação com DFCM terminou com um número de iterações e tempo de processamento menor que o FLC. Já no ambiente II com quatro robôs, o número de iterações necessário para resgatar todas as vítimas foi maior e, mesmo assim, possui menor tempo de processamento, comprovando o menor custo de processamento necessário para o DFCM.

Os agentes percorreram uma menor distância em ambos os casos para o DFCM e, além disso, sua área total explorada foi maior, pois estes exploraram novas áreas. No FLC, os robôs ficaram “presos” e acabaram explorando os mesmos ambientes. Os resultados para melhor comparação pode ser vistos nas tabelas a seguir (Tabelas 6, 7 e 8).

Tabela 6 - Comparação do tempo DFCM e FLC Ambiente II

Robôs	Controlador	Iterações	Tempo
1	DFCM	1466	9'50"
1	FLC	3708	18'31"
4	DFCM	385	12'40"
4	FLC	368	12'59"

Tabela 7 - Comparação Distância Percorrida Ambiente II

Distância Percorrida						
Robôs	Controlador	Robô 1	Robô 2	Robô 3	Robô 4	Total
1	DFCM	662,3315	-	-	-	662,3315
1	FLC	2063,4000	-	-	-	2063,4000
4	DFCM	194,3038	188,5215	169,2457	201,4758	753,5468
4	FLC	208,1097	213,9753	197,8093	231,1755	851,0698

Tabela 8 - Comparação Área Explorada Ambiente II

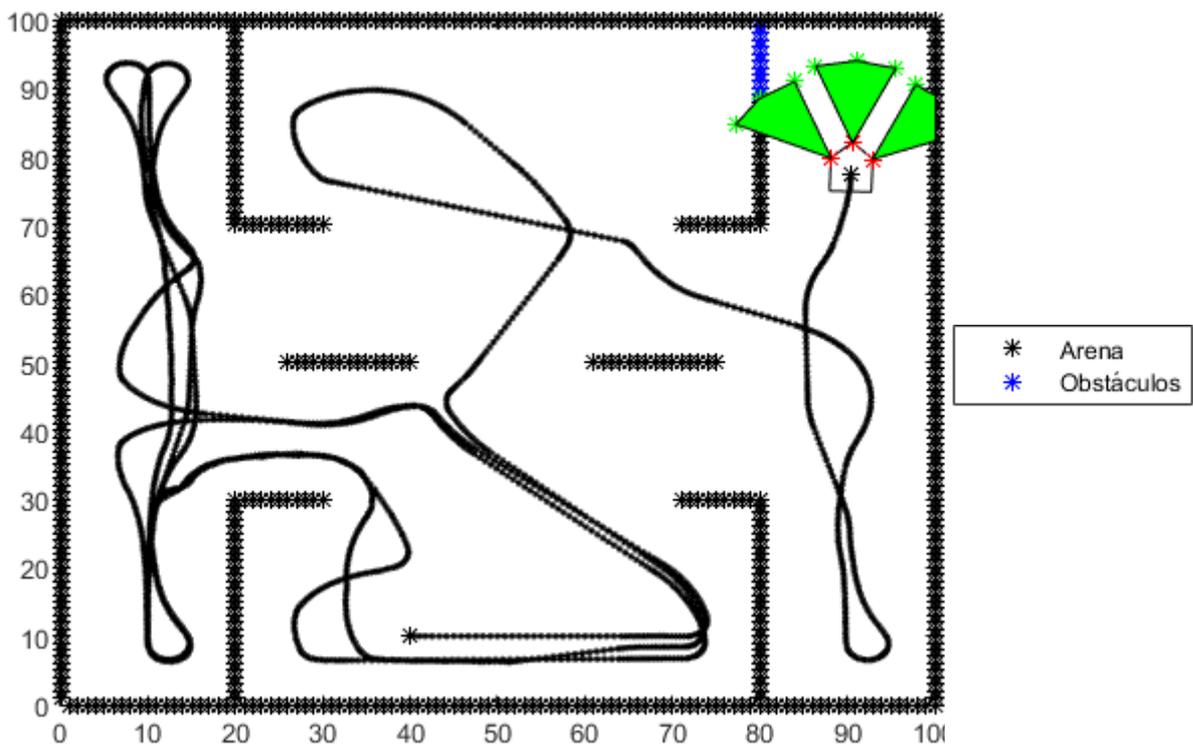
Área Explorada						
Robôs	Controlador	Robô 1	Robô 2	Robô 3	Robô 4	Total
1	DFCM	9258	-	-	-	9258
1	FLC	8880	-	-	-	8880
4	DFCM	4048	3593	2437	4141	9743
4	FLC	3911	3655	4273	4073	9394

4.3. Resultados do ambiente de teste III

Para o ambiente III utilizando somente um robô, o resgate de todas as vítimas durou 2215 iterações, tendo um tempo de processamento de 15 minutos e 10 segundos e gerando um vídeo de 73 segundos. O comportamento do agente e seu pulso podem ser vistos nas Figuras 35 e 36, respectivamente. O vídeo gerado pode ser acessado pelo link: <https://youtu.be/CsfKNDdn_Oc>.

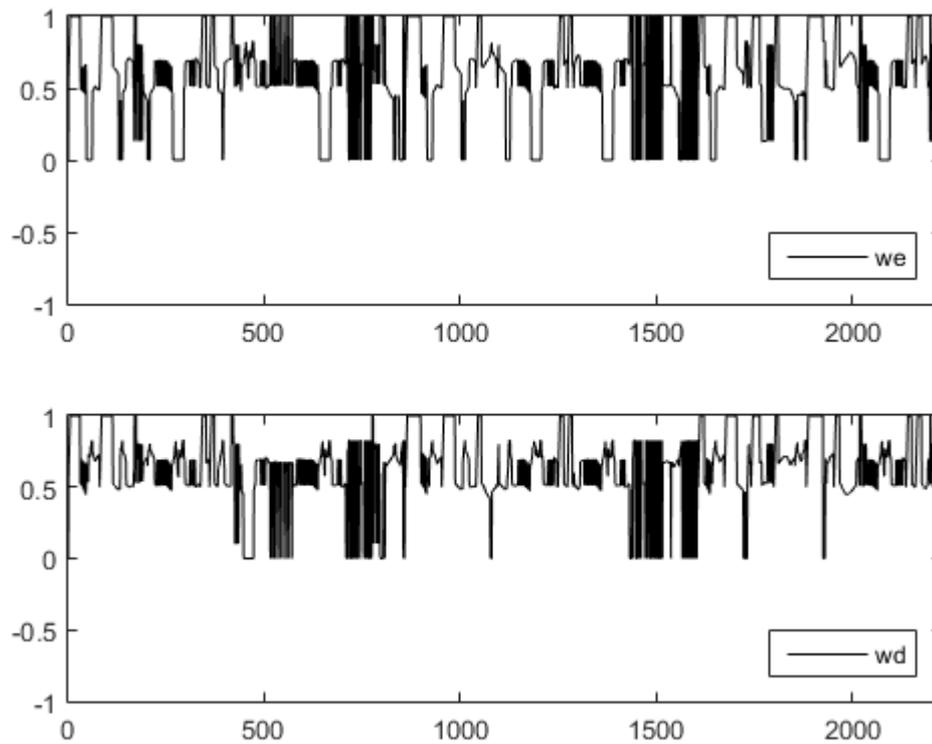
O explorador percorreu uma distância de 1134,48 m, explorando uma área de 9196 m², que corresponde a 91,96% da área total. Essa área pode ser vista na Figura 37.

Figura 35 - Simulação do ambiente III com 1 robô.



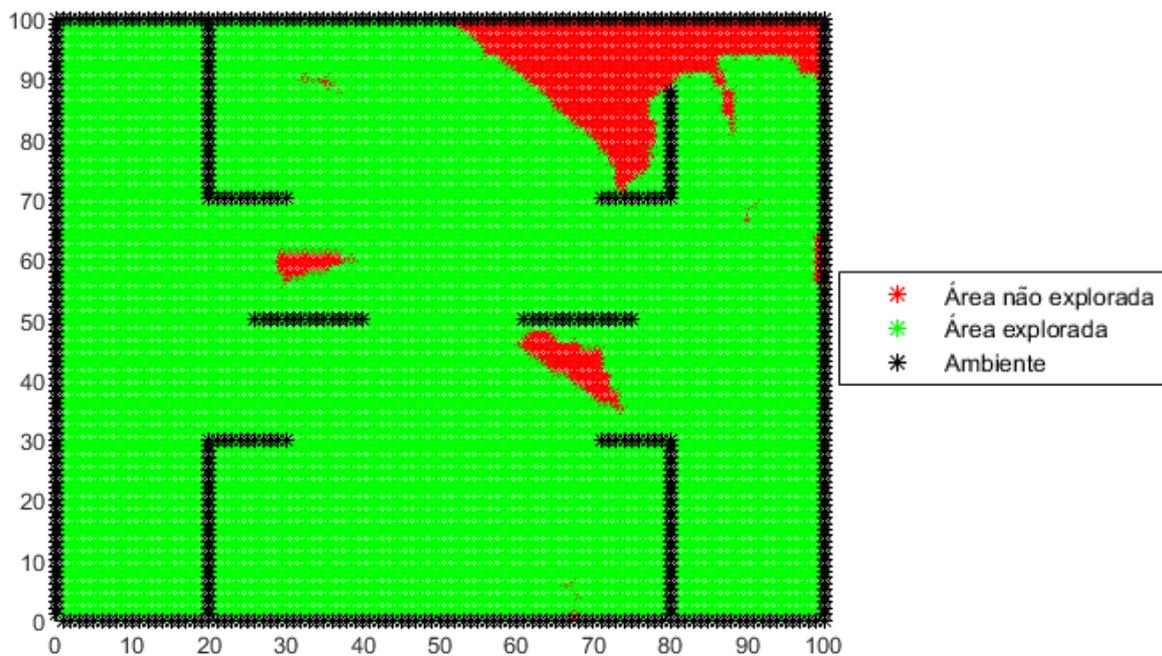
Fonte: Autoria Própria

Figura 36 - Pulsos do robô na simulação do ambiente III.



Fonte: Autoria Própria

Figura 37 - Área explorada pelo robô no ambiente III.



Fonte: Autoria Própria

Já para a simulação utilizando quatro robôs, o resgate durou 421 iterações, tendo um tempo de processamento de 12 minutos e 53 segundos e gerando um

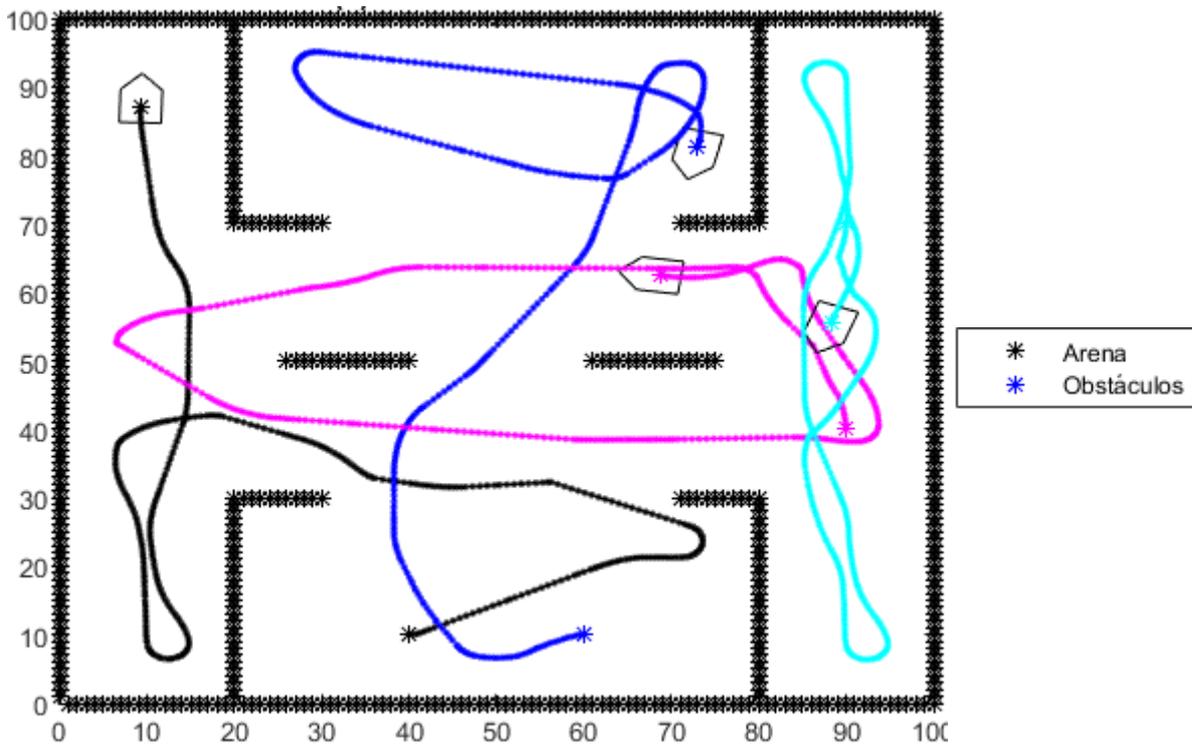
vídeo de 14 segundos. O vídeo gerado pode ser acessado pelo link: <<https://youtu.be/qrBEDn3dwQ0>>.

O comportamento de cada agente é demonstrado na Figura 38 e seu pulso pode ser vistos na Figura 39, respectivamente.

O explorador I percorreu uma distância de 227,5513 m e explorou uma área de 4256 m². O explorador II percorreu uma distância de 231,1775 m e explorou uma área de 4088 m². O explorador III percorreu uma distância de 234,8727 m e explorou uma área de 4328 m². O explorador IV percorreu uma distância de 202,1128 m e explorou uma área de 2753 m².

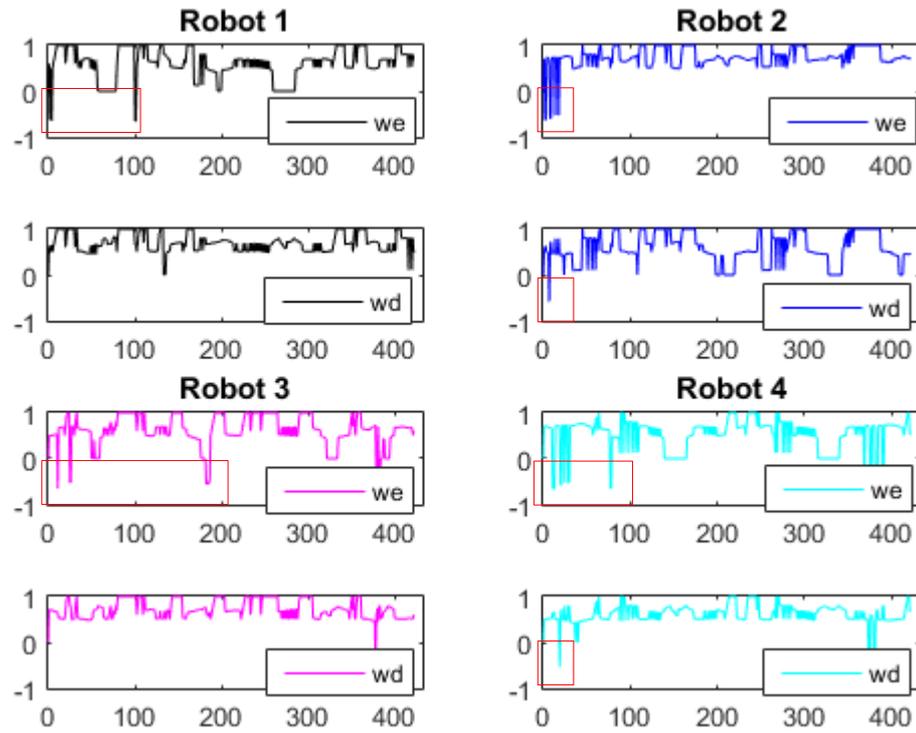
Portanto os robôs percorreram uma distância total de 895,7143 m e sua área explorada coletivamente foi de 9577 m², que representa 95,77% da área de busca. A área explorada por cada robô está representada na Figura 40 e a área total explorada pela Figura 41.

Figura 38 - Simulação do ambiente III com 4 robôs.



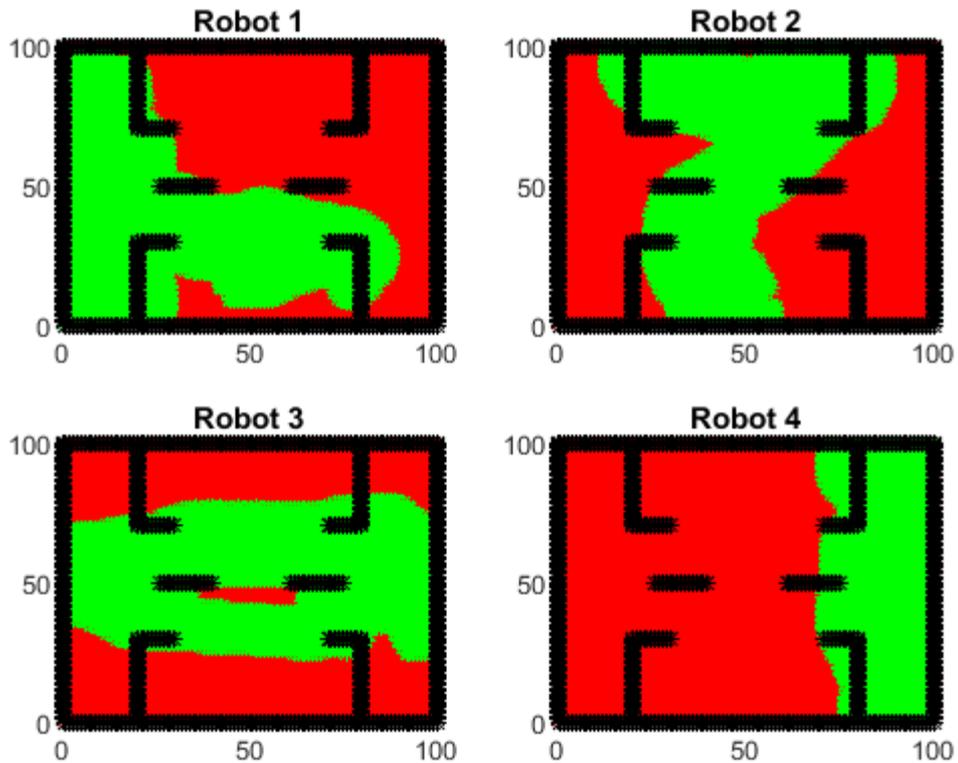
Fonte: Autoria Própria

Figura 39 - Pulsos dos robôs na simulação do ambiente III.



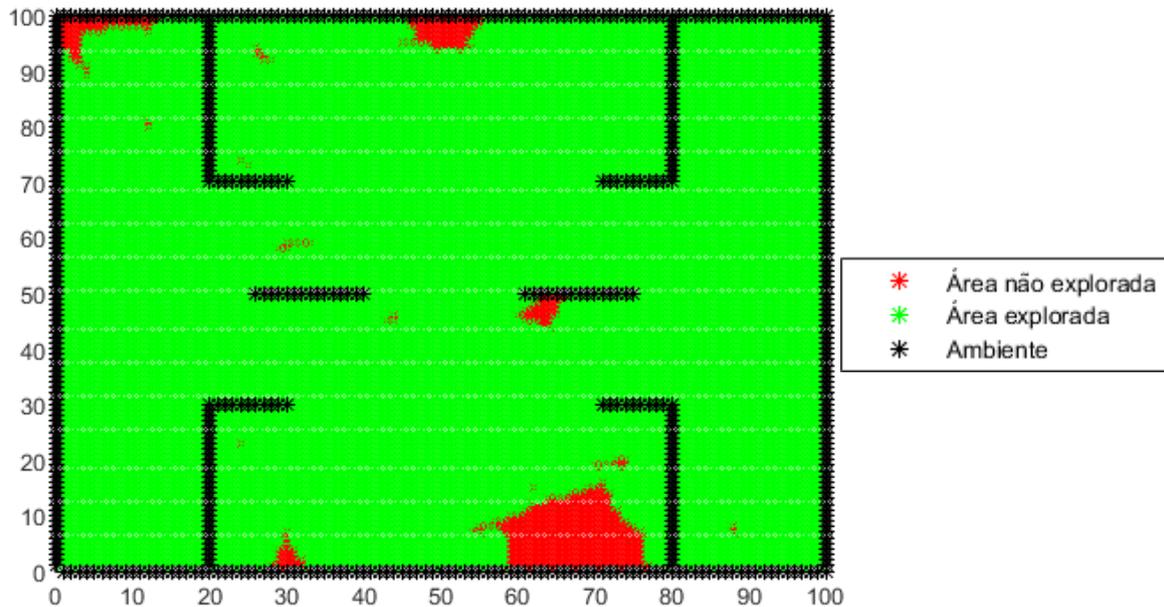
Fonte: Autoria Própria

Figura 40 - Área explorada pelos robôs no ambiente III.



Fonte: Autoria Própria

Figura 41 - Área total explorada pelos robôs no ambiente III.



Fonte: Autoria Própria

No último ambiente de teste, podemos visualizar também na simulação com um robô, a função implementada para que não fique preso numa mesma rota.

Para o ambiente com um e quatro robôs, a simulação com DFCM terminou com um número de iterações e, conseqüentemente, um tempo de processamento menor que o FLC. Os agentes percorreram uma menor distância, mas mesmo assim, a área total explorada foi maior, demonstrando eficiência na busca e exploração. Os resultados para comparação pode ser vistos nas tabelas a seguir (Tabelas 9, 10 e 11).

Tabela 9 - Comparação do tempo DFCM e FLC Ambiente III

Robôs	Controlador	Iterações	Tempo
1	DFCM	2215	15'10"
1	FLC	5536	17'43"
4	DFCM	421	12'53"
4	FLC	465	14'21"

Tabela 10 - Comparação Distância Percorrida Ambiente III

Distância Percorrida						
Robôs	Controlador	Robô 1	Robô 2	Robô 3	Robô 4	Total
1	DFCM	1134,4800	-	-	-	1134,4800
1	FLC	3524,7900	-	-	-	3524,7900
4	DFCM	227,5513	231,1775	234,8727	202,1128	895,7143
4	FLC	301,9569	233,6686	274,4438	278,7807	1088,8500

Tabela 11 - Comparação Área Explorada Ambiente III

Área Explorada						
Robôs	Controlador	Robô 1	Robô 2	Robô 3	Robô 4	Total
1	DFCM	9196	-	-	-	9196
1	FLC	8362	-	-	-	8362
4	DFCM	4256	4088	4328	2753	9577
4	FLC	2779	4724	4089	5004	9236

5. CONCLUSÃO

Em resumo, os resultados mostraram a aplicação de um controlador DFCM em um grupo de robôs e sua comparação com o trabalho usando um FLC (SOARES, 2017). Esses aspectos sugerem que o controle DFCM pode ser utilizado com sucesso para robôs autônomos, uma vez que nosso controlador apresentou resultados semelhantes aos da FLC anterior.

Foi aprimorado o desempenho computacional (número de iterações e tempo de processamento) com a abordagem DFCM, o que leva à implementação de um grupo maior de robôs desde que a DFCM concluiu as metas. Possivelmente, em cenários maiores, com maior número de robôs, a presença de *big data* pode ser introduzida devido ao volume de informações lidas dos sensores.

Este trabalho foi aceito no congresso On Fuzzy Systems da IEEE e serve como base para o desenvolvimento de outros sistemas utilizando o DFCM. Além disso, sugere o desenvolvimento de outros projetos FCM para o controle de agentes autônomos, como o apresentado por LI *et al.* (2018), no qual eles desenvolvem um sistema para o controle de carros autônomos em um jogo, verificando e validando esse sistema.

5.1. Trabalhos Futuros

Como continuidade desse trabalho, espera-se testar a técnica proposta em ambientes estressados, ou seja, com muitos robôs e poucas vítimas ou poucos robôs para muitas vítimas, buscando detectar comportamentos emergentes. Experimentos para a robustez do grupo de robôs devem ser investigados, por exemplo, o comportamento diante de uma possível falha de um ou mais robôs, a introdução da comunicação por feromônios, ou ainda, a implementação da quantidade de combustível disponível para os agentes. Pretende-se desenvolver também um algoritmo de busca e mapeamento como o SLAM (*Simultaneous Localization and Mapping*), utilizado no trabalho de PANCHPOR (2018).

Além disso, está previsto a implementação de um sensor de termovisão para detectar se as vítimas ainda estão com vidas e assim, priorizar o resgate destas. Por fim, a construção de protótipos para os testes em ambientes reais.

6. REFERÊNCIAS

- ARKIN, R. C. **Behavior-beased robotics**. MIT press, 1999.
- ARAI, T.; PAGELLO, E.; PARKER, L. E. **Guest editorial advances in multirobot systems**. IEEE Transaction on robotics and automation, v. 18, n. 5, p. 655-661, 2002.
- AUGUSTO, B. et al. **Heuristic Search Applied to Fuzzy Cognitive Maps Learning**. Search Algorithms for Engineering Optimization, 2013.
- AXELROD, R. **Structure of Decision: The Cog-nitive Maps of Political, Elites**. Princeton, NJ: Princeton Univ. Press, 1976.
- BAYINDIR, L. **A review of swarm robotics tasks**. Neurocomputing, v. 172, p. 292– 321, 2016.
- BENI, G. **From swarm intelligence to swarm robotics**. Ant Colony Optimization and Swarm Intelligence Proceedings, v. 3342, n. July 2004, p. 1–9, 2005.
- BRAITENBERG, V. **Vehicles: Experiments in Synthetic Psychology**. 1. ed. Cambridge, USA: MIT Press, 1986.
- BROOKS, R. **A robust layered control system for a mobile robot**. IEEE Journal on Robotics and Automation, v. 2, n. 1, p. 14–23, 1986.
- BROOKS, R. A.; STEIN, L. A. **Building Brain for Bodies**. Autonomous Robots, v. 1, 1994.
- CARVALHO J. P.; TOMÉ J. A. B. **Rule based fuzzy cognitive maps-expressing time in qualitative system dynamics**. 10th IEEE International Conference on Fuzzy Systems, 2001, pp. 280–283.
- CALVO, R. **Arquitetura híbrida inteligente para navegação autônoma de robôs**. v. Master, 2007.
- CALVO, R. **Sistemas bio-inspirados para coordenação de múltiplos robôs móveis**. 2012.
- CHAUVIN, L. et al. **Ontological Cognitive Map**. 2008 20th IEEE International Conference on Tools with Artificial Intelligence, 2008, 3-5 Nov. 2008. p.225-232.
- CHRUN, I. R. **Mapas cognitivos fuzzy dinâmicos aplicados em vida artificial e robótica de enxame**. 2016.
- CORKE P. **Robotics, Vision and Control**. Fundamental Algorithms in MATLAB, 1st ed., vol. 73, no. 6. Heidelberg: Springer-Verlag Berlin Heidelberg, 2011.
- COSTA, E. D. S.; GOUVEA, M. M. **Autonomous Navigation in Dynamic Environments with Reinforcement Learning and Heuristic**. Machine Learning and Applications (ICMLA), Ninth International Conference, 2010.
- DE ALMEIDA, J. P. L. S. et al. **Bio-inspired on-line path planner for cooperative exploration of unknown environment by a Multi-Robot System** *Robotics and Autonomous Systems*, 2019.
- DE RANGO F.; Palmieri N.; Yang X. S.; Marano S. **Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks**. Soft Comput., vol. 22, no. 13, pp. 4251–4266, 2018.
- DUCATELLE, F. et al. **Cooperative navigation in robotic swarms**. Swarm Intelligence, v. 8, n. 1, p. 1–33, 2014.
- DUDEK, G. et al. **A taxonomy for multi-agent robotics**. Autonomous Robots, v. 3, n. 4, p. 4–29, 1996.
- GORIS, K. **Autonomous mobile robot mechanical design**. Verije University, Brussels, 2005.

- KOK, K. **The potential of Fuzzy Cognitive Maps for semi-quantitative scenario development, with an example from Brazil.** *Global Environmental Change* 19 122– 133, 2009.
- KOSKO, B. **Fuzzy cognitive maps.** *Int. J. Man-Machine Studies*, v. 24, n. 1, p. 65-75, 1986.
- LI, N. et al. **Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems.** *IEEE Transactions on Control Systems Technology*, v. 26, n. 5, p. 1782–1797, 2018.
- MAES, P. **Artificial life meets entertainment: lifelike autonomous agents.** *Communications of the ACM*, v. 38, n. 11, p. 108–114, 1995.
- MATARIC M. J. **The robotics primer.** v. 45 2007
- MENDONÇA, M. et al. **Fuzzy Cognitive Maps Applied to Student Satisfaction Level in an University.** *IEEE Latin America Transactions*, v. 13, n. 12, p. 3922-3927, 2015d. 27
- MENDONÇA, M.; ANGÉLICO, B.; ARRUDA, L.V.R.; NEVES, F. (2013). **A dynamic fuzzy cognitive map applied to chemical process supervision.** *Engineering Applications of Artificial Intelligence*, v. 26, p. 1199-1210.
- MENDONÇA, M.; DE ARRUDA, L. V. R.; NEVES, F. **Autonomous navigation system using Event Driven-Fuzzy Cognitive Maps.** *Applied Intelligence*, v. 37, n. 2, p. 175–188, 2012.
- MENDONÇA, M. et al. **A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps.** *Engineering Applications of Artificial Intelligence*, v. 59, p. 122–132, 1 mar. 2017.
- MIAO, Y. **Dynamical cognitive network-an extension of fuzzy cognitive map.** *Tools with Artificial Intelligence*, 1999. *Proceedings. 11th IEEE International Conference on*, 1999, 1999. p.43-46.
- NÁPOLES, G. et al. **On the convergence of sigmoid Fuzzy Cognitive Maps.** *Information Sciences*, v. 349–350, p. 154–171, 1 jul. 2016.
- PALMIERI N.; YANG X. S.; DE RANGO F.; SANTAMARIA A. F. **Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team.** *Neurocomputing*, vol. 306, pp. 17–36, 2018.
- PAPAGEORGIOU, EI, J. S. **A review of fuzzy cognitive map research at the last decade.** *leexplore.ieee.Org*, n. c, 2013.
- PAPAGEORGIOU, E. I. **Fuzzy Cognitive Maps for Applied Sciences and Engineering.** Vol. 54, Springer, Berlin, 2014.
- PAPAGEORGIOU, E. I. **Learning algorithms for fuzzy cognitive maps.** *IEEE Trans. Syst. Cybern. Part C Appl. Rev.* 42, 150–163, 2012.
- PANCHPOR, A. A.; SHUE, S.; CONRAD, J. M. **A survey of methods for mobile robot localization and mapping in dynamic indoor environments.** *2018 Conference on Signal Processing And Communication Engineering Systems, SPACES 2018*, v. 2018–Janua, p. 138–144, 2018.
- PEREIRA D. S; SANTANA, A. S.; MAIA R. S.; SOUZA A. **A Cloud Robotics Architecture Clone Based for a Cellbots Team.** *IEEE Latin America Transactions*, v. 15, n. 9, p. 1587–1594, 2017.
- PUIG, D.; GARCIA, M. A.; WU, L. **A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation.** *Robotics and Autonomous Systems*, v. 59, n. 9, p. 635–653, 2011.
- ROBLA-GOMEZ S.; BECERRA V. M.; LLATA J. R.; GONZALEZ-SARABIA E.; TORRE-FERRERO C.; PEREZ-ORIA J. **Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments.** *IEEE*, 2017.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence A Modern Approach**. 2013.

SOARES, P. P. **Group of Robots Inspired by Swarm Robotics Exploring Unknown Environments**. 2017.

SAHIN E.; GIRGIN S.; BAYINDIR L.; TURGUT A. E. **Swarm Robotics**. Chapter, 2008.

SICILIANO B.; KHATIB O. **Springer Handbook of Robotics**, 2nd ed. Heidelberg: Springer-Verlag Berlin Heidelberg, 2016.

SMITHERS, T. **Autonomy in Robots and Other Agents**. *Brain and Cognition*, v. 34, n. 1, p. 88-106, 1997.

TOLMAN, E.C.; **Cognitive maps in rats and men**. *Psychol. Rev.* 55(4), 189–208 (1997)

YUAN,M.; ZHI-QIANG, L. **On causal inference in fuzzy cognitive maps**. *IEEE Transactions on Fuzzy Systems*, v.8, n.1, p.107-119, 2000.