

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CAMPUS CORNÉLIO PROCÓPIO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

LUCAS BOTONI DE SOUZA

**SISTEMA MULTIRROBÔ PARA EXPLORAÇÃO DE AMBIENTES  
DESCONHECIDOS UTILIZANDO CONTROLADORES FUZZY**

DISSERTAÇÃO

CORNÉLIO PROCÓPIO  
2020

LUCAS BOTONI DE SOUZA

**SISTEMA MULTIRROBÔ PARA EXPLORAÇÃO DE AMBIENTES  
DESCONHECIDOS UTILIZANDO CONTROLADORES FUZZY**

Dissertação apresentada como requisito para obtenção do título de Mestre em Engenharia Mecânica do Programa de Pós-Graduação em Engenharia Mecânica da Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio.

Orientador: Prof. Dr. Márcio Mendonça  
Coorientador: Prof. Dr. Rodrigo Henrique Cunha Palácios

CORNÉLIO PROCÓPIO  
2020

---

### Dados Internacionais de Catalogação na Publicação

---

S729 Souza, Lucas Botoni de

Sistema multirrobo para exploração de ambientes desconhecidos utilizando controladores fuzzy / Lucas Botoni de Souza. – 2020.  
121 p. : il. color. ; 31 cm.

Orientador: Márcio Mendonça.

Coorientador: Rodrigo Henrique Cunha Palácios.

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Mecânica. Cornélio Procópio, 2020.

Bibliografia: p. 105-113.

1. Robótica. 2. Robôs móveis. 3. Mapas cognitivos (Psicologia). 4. Engenharia Mecânica – Dissertações. I. Molina, Fabian Andres Lara, orient. II. Palácios, Rodrigo Henrique Cunha, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

CDD (22. ed.) 620.1

---

### Biblioteca da UTFPR - Câmpus Cornélio Procópio

Bibliotecário/Documentalista responsável:  
Romeu Righetti de Araujo – CRB-9/1676



**Título da Dissertação Nº 044:**

# **“Sistema Multirrobo Para Exploração De Ambientes Desconhecidos Utilizando Controladores Fuzzy”.**

Por

## **Lucas Botoni de Souza**

Orientador: **Prof. Dr. Márcio Mendonça**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de **MESTRE EM ENGENHARIA MECÂNICA** – Área de Concentração: **Ciências Mecânicas**, linha de pesquisa: **Sistemas Dinâmicos**, pelo Programa de Pós-Graduação em Engenharia Mecânica – PPGEM – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 13h50min do dia 11 de MARÇO de 2020. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

---

Prof.Dr. Márcio Mendonça  
(Orientador – UTFPR-CP)

---

Prof. Dr. Adailton Silva Borges  
(UTFPR-CP)

---

Prof. Dr. Ricardo Breganon  
(IFPR-Jacarezinho)

Dedico este trabalho à minha família e a todos que trazem o melhor em mim.

## **AGRADECIMENTOS**

Agradeço minha família, Sidnei, Angélica e Tiago, pelo incessante incentivo a realizar um curso de pós-graduação e pelo apoio depositado em todas as etapas desse processo.

Agradeço também a família da minha namorada Larissa, em especial Vilma e Gilberto, por me ajudarem sempre que possível e por atuarem como minha segunda família.

Aos meus amigos de verdade, dos quais sentirei muita falta, agradeço pelos momentos juntos e por sempre poder contar com eles.

Também merece agradecimento todo o corpo docente e demais profissionais do PPGEM-CP, em especial o coordenador Prof. Dr. Vagner Rigo e Wandressa Bonfim por todo o auxílio prestado a mim.

Gostaria de agradecer ao meu orientador Prof. Dr. Márcio Mendonça pelo convite a participar do PPGEM e também pela orientação durante esse período, assim como meu Coorientador Prof. Dr. Rodrigo Henrique Cunha Palácios, cujas sugestões e ajuda foram fundamentais para o andamento do trabalho.

Por fim, meus sinceros agradecimentos à CAPES/Fundação Araucária, a Superintendência de Ciência, Tecnologia e Ensino Superior do Paraná (SETI), ao governo do Estado do Paraná e a Universidade Tecnológica Federal do Paraná pelo auxílio financeiro durante o período, seja na forma de bolsa ou apoio para participação em eventos.

“The true measure of a man is not his intelligence or how high he rises in this freak establishment. No, the true measure of a man is this: how quickly can he respond to the needs of others and how much of himself he can give.”

– Philip K. Dick

## RESUMO

DE SOUZA, L. B. **Sistema Multirrobo para Exploração de Ambientes Desconhecidos Utilizando Controladores Fuzzy**. 121 p. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia Mecânica. Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2020.

Uma das utilizações dos Sistemas Multirrobo (SMR) está em operações de salvamento de vítimas. O principal desafio encontrado por equipes de salvamento em desastres é o tempo de resposta. As chances de encontrar sobreviventes diminuem significativamente com o tempo e reduzidas drasticamente após 48 horas. Nesse sentido, a motivação deste trabalho é apresentar um SMR inspirado nos conceitos de robótica de enxame para resgate de vítimas em ambientes desconhecidos. Neste caso, os robôs não possuem conhecimento dos limites da área de busca e obstáculos, conhecendo apenas o número de vítimas a serem resgatadas como critério de parada para os experimentos simulados no Matlab<sup>®</sup>. Para isso, três abordagens que herdam os principais aspectos da lógica *Fuzzy* são utilizadas com base em trabalhos anteriores: um controlador lógico *Fuzzy* (FLC), um controlador baseado em um Mapa Cognitivo *Fuzzy* Dinâmico (DFCM) e um DFCM inspirado na metaheurística de Otimização por Colônia de Formigas (DFCM-ACO). A tarefa proposta simula operações de resgate de desastres na vida real, ou mesmo humanos perdidos em ambientes desconhecidos, como florestas. As simulações foram realizadas em três ambientes, a fim de testar a robustez global contra situações imprevisíveis, autonomia, área explorada e tempo de processamento computacional para as três abordagens, usando uma arquitetura reativa baseada em subsunção. Em geral, os resultados sugerem que as abordagens SMR baseadas em DFCM são capazes de concluir as tarefas propostas consumindo menos tempo de processamento, com os robôs percorrendo menores distâncias para explorar uma quantidade semelhante a área da abordagem FLC, com o DFCM-ACO apresentando resultados equilibrados entre as técnicas: a princípio é mais eficiente, porém é mais complexo computacionalmente que o DFCM. Finalmente, um protótipo inicial para utilização em experimentos reais é apresentado, assim como trabalhos futuros são endereçados.

**Palavras-chave:** Robótica de enxame. Sistema Multirrobo. Ambientes Desconhecidos. Robôs Móveis Autônomos. Mapas Cognitivos *Fuzzy* Dinâmicos.



## ABSTRACT

DE SOUZA, L. B. **Multi-robot System for Unknown Environments Exploration Using Fuzzy Controllers.** 121 p. Master thesis (Graduation) – Mechanical Engineering Graduate Program. Federal University of Technology – Paraná, Cornélio Procópio, 2020.

One of the application fields of Multi-Robot Systems (MRS) is within victim rescue operations. The main challenge faced by disaster rescue teams is response time. The chances of finding survivors decrease significantly over time and dramatically decrease after 48 hours. In this context, the motivation of this work is to present an MRS inspired by the concepts of swarm robotics to rescue victims in unknown environments. In this case, the robots are unaware of the search area boundaries and obstacles, knowing only the number of victims to be rescued as a stopping criterion for the simulation made in Matlab<sup>®</sup>. Therefore, three approaches inheriting the main aspects of fuzzy logic are used based on previous works: a fuzzy logic controller (FLC), a dynamic fuzzy cognitive map (DFCM) controller, and a DFCM inspired by the ant colony optimization metaheuristic (DFCM-ACO). The proposed task simulates real life disaster rescue operations, or even humans lost in unfamiliar environments such as forests. The simulations were performed in three environments in order to test the overall robustness against unpredictable situations, autonomy, explored area and computer processing time for the three approaches using a subsumption-based reactive architecture. In general, the results suggest that the DFCM-based MRS approaches are able to complete the tasks consuming less processing time, with robots travelling shorter distances to explore a similar environment to the FLC approach and with the DFCM-ACO presenting balanced results between the other techniques: at first it is more efficient, but it is more computationally complex than the DFCM. Finally, future works are outlined.

**Keywords:** Swarm Robotics. Multi-Robot System. Unknown Environments. Autonomous Mobile Robots. Dynamic Fuzzy Cognitive Maps.

## LISTA DE FIGURAS

Figura 2.1 – Robô móvel utilizado em armazéns .....	23
Figura 2.2 – Tesla Model S em modo de condução autônoma .....	24
Figura 2.3 – A integração dos conceitos da Indústria 4.0.....	27
Figura 2.4 – Exemplo de sistema multirrobo .....	28
Figura 2.5 – Distribuição de comportamentos .....	30
Figura 2.6 – Representação gráfica de um neurônio artificial .....	32
Figura 2.7 – Comparação entre lógica <i>crisp</i> (a) e <i>Fuzzy</i> (b) .....	35
Figura 2.8 – Diagrama de blocos de um Controlador Lógico <i>Fuzzy</i> .....	36
Figura 2.9 – Exemplo de superfície <i>Fuzzy</i> .....	37
Figura 2.10 – Exemplo de um Mapa Cognitivo <i>Fuzzy</i> .....	40
Figura 2.11 – Ilustração de um experimento com formigas.....	45
Figura 3.1 – Modelo cinemático para o robô móvel.....	47
Figura 3.2 – Máquina de estados finitos para o comportamento dos robôs .....	52
Figura 3.3 – Visão geral da arquitetura dos robôs.....	53
Figura 3.4 – Ambientes simulados .....	54
Figura 3.5 – Representação das características dos robôs simulados .....	57
Figura 3.6 – Sensor Ultrassônico HC-SR04.....	57
Figura 3.7 – Princípio de funcionamento HC-SR04.....	58
Figura 3.8 – Módulo GPS GY-NEO6MV2 .....	58
Figura 3.9 – Representação gráfica do sistema de coordenadas locais .....	59
Figura 3.10 – Estrutura do FLC utilizado.....	61
Figura 3.11 – Funções de pertinência utilizadas .....	62
Figura 3.12 – Superfícies <i>Fuzzy</i> : SD e SF com (a) $w_E$ e (b) $w_D$ .....	63
Figura 3.13 – Superfícies <i>Fuzzy</i> : SE e SD com (a) $w_E$ e (b) $w_D$ .....	63
Figura 3.14 – Superfície <i>Fuzzy</i> : SE e SF com (a) $w_E$ e (b) $w_D$ .....	64
Figura 3.15 – Mapa Cognitivo <i>Fuzzy</i> Dinâmico proposto .....	64
Figura 3.16 – DFCM-ACO proposto .....	66
Figura 4.1 – Trajeto de um robô com FLC: ambientes (a) I, (b) II e (c) III .....	68
Figura 4.2 – Trajeto de um robô com DFCM: ambientes (a) I, (b) II e (c) III.....	68
Figura 4.3 – Trajeto de um robô com DFCM-ACO: ambientes (a) I, (b) II e (c) III.....	68
Figura 4.4 – Pulsos de um robô com FLC: ambiente I .....	69
Figura 4.5 – Pulsos de um robô com FLC: ambiente II .....	69
Figura 4.6 – Pulsos de um robô com FLC: ambiente III .....	69
Figura 4.7 – Pulsos de um robô no ambiente I: (a) DFCM, (b) DFCM-ACO .....	70
Figura 4.8 – Pulsos de um robô no ambiente II: (a) DFCM, (b) DFCM-ACO .....	70
Figura 4.9 – Pulsos de um robô no ambiente III: (a) DFCM, (b) DFCM-ACO .....	70
Figura 4.10 – Área explorada por um robô com FLC: ambientes (a) I, (b) II e (c) III .....	71
Figura 4.11 – Área explorada por um robô com DFCM: ambientes (a) I, (b) II e (c) III .....	71
Figura 4.12 – Área explorada por um robô com DFCM-ACO: ambientes (a) I, (b) II e (c) III .....	71

Figura 4.13 – Trajeto de quatro robôs do ambiente I: (a) FLC, (b) DFCM e (c) DFCM-ACO .....	73
Figura 4.14 – Pulsos de quatro robôs com FLC: ambiente I .....	73
Figura 4.15 – Pulsos de quatro robôs com DFCM: ambiente I.....	74
Figura 4.16 – Pulsos de quatro robôs com DFCM-ACO: ambiente I.....	74
Figura 4.17 – Área explorada pelo SMR com FLC: ambiente I .....	75
Figura 4.18 – Área explorada pelo SMR com DFCM: ambiente I.....	75
Figura 4.19 – Área explorada pelo SMR com DFCM-ACO: ambiente I.....	76
Figura 4.20 – Trajeto de quatro robôs do ambiente II: (a) FLC, (b) DFCM e (c) DFCM-ACO .....	76
Figura 4.21 – Pulsos de quatro robôs com FLC: ambiente II .....	77
Figura 4.22 – Pulsos de quatro robôs com DFCM: ambiente II.....	77
Figura 4.23 – Pulsos de quatro robôs com DFCM-ACO: ambiente II.....	78
Figura 4.24 – Área explorada pelo SMR com FLC: ambiente II .....	78
Figura 4.25 – Área explorada pelo SMR com DFCM: ambiente II.....	78
Figura 4.26 – Área explorada pelo SMR com DFCM-ACO: ambiente II.....	79
Figura 4.27 – Trajeto de quatro robôs do ambiente III: (a) FLC, (b) DFCM e (c) DFCM-ACO .....	79
Figura 4.28 – Pulsos de quatro robôs com FLC: ambiente III .....	80
Figura 4.29 – Pulsos de quatro robôs com DFCM: ambiente III.....	80
Figura 4.30 – Pulsos de quatro robôs com DFCM-ACO: ambiente III.....	81
Figura 4.31 – Área explorada pelo SMR com FLC: ambiente III, 4 robôs .....	81
Figura 4.32 – Área explorada pelo SMR com DFCM: ambiente III, 4 robôs.....	81
Figura 4.33 – Área explorada pelo SMR com DFCM-ACO: ambiente III, 4 robôs.....	82
Figura 4.34 – Trajeto de oito robôs do ambiente I: (a) FLC, (b) DFCM e (c) DFCM-ACO .....	84
Figura 4.35 – Pulsos de oito robôs com FLC: ambiente I, robôs 1 a 4.....	84
Figura 4.36 – Pulsos de oito robôs com FLC: ambiente I, robôs 5 a 8.....	85
Figura 4.37 – Pulsos de oito robôs com DFCM: ambiente I, robôs 1 a 4 .....	86
Figura 4.38 – Pulsos de oito robôs com DFCM: ambiente I, robôs 5 a 8 .....	86
Figura 4.39 – Pulsos de oito robôs com DFCM-ACO: ambiente I, robôs 1 a 4 .....	87
Figura 4.40 – Pulsos de oito robôs com DFCM-ACO: ambiente I, robôs 5 a 8 .....	87
Figura 4.41 – Áreas individuais e total exploradas pelo SMR com FLC: ambiente I, 8 robôs .....	88
Figura 4.42 – Áreas individuais e total exploradas pelo SMR com DFCM: ambiente I, 8 robôs .....	88
Figura 4.43 – Áreas individuais e total exploradas pelo SMR (DFCM-ACO): ambiente I, 8 robôs .....	88
Figura 4.44 – Trajeto de oito robôs do ambiente II: (a) FLC, (b) DFCM e (c) DFCM-ACO .....	89
Figura 4.45 – Pulsos de oito robôs com FLC: ambiente II, robôs 1 a 4.....	89
Figura 4.46 – Pulsos de oito robôs com FLC: ambiente II, robôs 5 a 8.....	90
Figura 4.47 – Pulsos de oito robôs com DFCM: ambiente II, robôs 1 a 4 .....	90
Figura 4.48 – Pulsos de oito robôs com DFCM: ambiente II, robôs 5 a 8 .....	91

Figura 4.49 – Pulsos de oito robôs com DFCM-ACO: ambiente II, robôs 1 a 4 .....	91
Figura 4.50 – Pulsos de oito robôs com DFCM-ACO: ambiente II, robôs 5 a 8 .....	92
Figura 4.51 – Áreas individuais e total exploradas pelo SMR com FLC: ambiente II, 8 robôs .....	92
Figura 4.52 – Áreas individuais e total exploradas pelo SMR com DFCM: ambiente II, 8 robôs .....	93
Figura 4.53 – Áreas individuais e total exploradas pelo SMR com DFCM-ACO: ambiente II, 8 robôs.....	93
Figura 4.54 – Trajeto de oito robôs do ambiente III: (a) FLC, (b) DFCM e (c) DFCM-ACO .....	94
Figura 4.55 – Pulsos de oito robôs com FLC: ambiente III, robôs 1 a 4 .....	94
Figura 4.56 – Pulsos de oito robôs com FLC: ambiente III, robôs 5 a 8 .....	95
Figura 4.57 – Pulsos de oito robôs com DFCM: ambiente III, robôs 1 a 4 .....	95
Figura 4.58 – Pulsos de oito robôs com DFCM: ambiente III, robôs 5 a 8 .....	96
Figura 4.59 – Pulsos de oito robôs com DFCM-ACO: ambiente III, robôs 1 a 4 .....	96
Figura 4.60 – Pulsos de oito robôs com DFCM-ACO: ambiente III, robôs 5 a 8 .....	97
Figura 4.61 – Áreas individuais e total exploradas pelo SMR com FLC: ambiente III, 8 robôs .....	98
Figura 4.62 – Áreas individuais e total exploradas pelo SMR com DFCM: ambiente III, 8 robôs .....	98
Figura 4.63 – Áreas individuais e total exploradas pelo SMR com DFCM-ACO: ambiente III, 8 robôs.....	98
Figura 4.64 – Comparativo gráfico entre os tempos de processamento .....	101
Figura B.1 – Vistas do robô protótipo .....	119
Figura B.2 – Sequência de eventos do teste inicial.....	120
Figura B.3 – Sequência de eventos do teste final .....	121
Figura B.4 – Comparação entre coordenadas obtidas entre Google Maps e monitor serial.....	121

## LISTA DE TABELAS

Tabela 3.1 – Descrição dos subdomínios do problema abordado.....	50
Tabela 3.2 – Descrição dos eventos do SMR .....	52
Tabela 3.3 – Poses para os cenários com 4 robôs .....	56
Tabela 3.4 – Poses para os cenários com 8 robôs .....	56
Tabela 3.5 – Valores dos pesos para o DFCM.....	65
Tabela 3.6 – Valores dos pesos para o DFCM-ACO.....	66
Tabela 4.1 – Resultados das simulações para um robô.....	72
Tabela 4.2 – Número de iterações e tempo de processamento para 4 robôs.....	82
Tabela 4.3 – Área explorada para 4 robôs .....	83
Tabela 4.4 – Distância percorrida para 4 robôs.....	83
Tabela 4.5 – Número de iterações e tempo de processamento para 8 robôs.....	99
Tabela 4.6 – Área explorada para 8 robôs .....	99
Tabela 4.7 – Distância percorrida para 8 robôs.....	100

## LISTA DE SIGLAS E ACRÔNIMOS

FCM	Mapas Cognitivos <i>Fuzzy</i> , do inglês, <i>Fuzzy Cognitive Maps</i>
DFCM	Mapas Cognitivos <i>Fuzzy</i> Dinâmicos, do inglês, <i>Dynamic Fuzzy Cognitive Maps</i>
RNA	Rede Neural Artificial
FLC	Controlador Lógico <i>Fuzzy</i>
SMR	Sistema(s) Multirrobo(s)
B&R	Busca e Resgate
RBFCM	Mapas Cognitivos <i>Fuzzy</i> Baseados em Regras, do inglês, <i>Rule-Based Fuzzy Cognitive Maps</i>
ACO	Otimização por Colônia de Formigas, do inglês, <i>Ant Colony Optimization</i>
HD-FCM	Mapas Cognitivos <i>Fuzzy</i> Dinâmicos Híbridos, do inglês, <i>Hybrid-Dynamic Fuzzy Cognitive Maps</i>
TAFCM	Mapas Cognitivos <i>Fuzzy</i> Baseados em Autômatos Temporizados, do inglês, <i>Timed Automata-based Fuzzy Cognitive Maps</i>
FCN	Rede Cognitiva <i>Fuzzy</i> , do inglês, <i>Fuzzy Cognitive Network</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
WCCI	Congresso Mundial em Inteligência Computacional, do inglês, <i>World Congress on Computational Intelligence</i>
ED-FCM	Mapas Cognitivos <i>Fuzzy</i> Acionados por Eventos, do inglês, <i>Event-Driven Fuzzy Cognitive Maps</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>14</b>
1.1 MOTIVAÇÕES.....	18
1.2 OBJETIVOS .....	19
1.2.1 Objetivo geral .....	20
1.2.2 Objetivos específicos.....	20
1.3 ESTRUTURA DO TRABALHO.....	20
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>22</b>
2.1 ROBÓTICA MÓVEL AUTÔNOMA.....	22
2.2 SISTEMAS MULTIROBÔS E ROBÓTICA DE ENXAME .....	27
2.3 PRINCÍPIOS DA ARQUITETURA DE SUBSUNÇÃO DOS ROBÔS MÓVEIS ....	30
2.4 REDES NEURAIS ARTIFICIAIS.....	31
2.5 SISTEMAS <i>FUZZY</i> .....	33
2.5.1 Mapas Cognitivos Fuzzy.....	37
2.5.2 Revisão bibliográfica .....	41
2.6 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS .....	44
<b>3 DESENVOLVIMENTO.....</b>	<b>46</b>
3.1 MODELAGEM CINEMÁTICA DOS ROBÔS MÓVEIS.....	46
3.2 ARQUITETURA DE SUBSUNÇÃO .....	50
3.3 CONSIDERAÇÕES DOS EXPERIMENTOS SIMULADOS .....	53
3.4 SISTEMA DE CONTROLE <i>FUZZY</i> .....	61
3.5 SISTEMAS DE CONTROLE POR MAPAS COGNITIVOS <i>FUZZY</i> DINÂMICOS	64
<b>4 RESULTADOS E DISCUSSÃO .....</b>	<b>67</b>
4.1 RESULTADOS PARA UM ROBÔ .....	67
4.2 RESULTADOS PARA QUATRO ROBÔS .....	72
4.3 RESULTADOS PARA OITO ROBÔS .....	83
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>102</b>
5.1 TRABALHOS GERADOS .....	103
5.2 TRABALHOS FUTUROS.....	103
<b>REFERÊNCIAS.....</b>	<b>105</b>
<b>APÊNDICE A – REGRAS DO SISTEMA <i>FUZZY</i>.....</b>	<b>114</b>
<b>APÊNDICE B – RESULTADOS INICIAIS DO PROTÓTIPO .....</b>	<b>119</b>

## 1 INTRODUÇÃO

A utilização de robôs está aumentando nos últimos anos, principalmente impulsionada pela implementação dos conceitos da Indústria 4.0. Atualmente, eles são utilizados nos campos educacional, médico, nas indústrias, na agricultura, no combate a incêndios e no resgate de pessoas perdidas, na vigilância, nas operações militares (terrestres e subaquáticas) (SICILIANO; KHATIB, 2016), na exploração espacial e em áreas inóspitas aos seres humanos, como inspeção de vulcões e minas (BEN-ARI; MONDADA, 2018). Eles podem executar melhor a maioria dessas tarefas quando um grupo de robôs móveis autônomos é utilizado. Nessa configuração, os robôs podem alcançar um objetivo global complexo por meio de interações locais simples e cooperativas (MENDONÇA *et al.*, 2019).

Em particular, tarefas de exploração e busca que envolvem Sistemas Multirrobôs (SMR) exigem que um conjunto de habilidades seja executado individualmente por cada robô móvel. Dentre elas estão a percepção, auto localização, locomoção, comunicação e tomada de decisões, a fim de alcançar um comportamento descentralizado e cooperativo (DE RANGO *et al.*, 2018; PALMIERI *et al.*, 2018). Esses princípios levam esses robôs a serem capazes de reagir a obstáculos estáticos ou dinâmicos, agentes externos e eventos ou comportamentos emergentes (RUSSELL; NORVIG, 2009).

Desse modo, o campo de pesquisa que engloba os SMR com inteligência coletiva é denominado de robótica de enxame, caracterizado pelo grande número de robôs homogêneos ou heterogêneos de relativa simplicidade estrutural e de componentes (BARCA; SEKERCIOGLU, 2013). A robótica de enxame difere dos sistemas robóticos tradicionais devido à sua forma de controle não centralizada e não hierárquica, ou seja, os robôs não possuem uma coordenação central (controlador único). Os sistemas de robótica de enxame são totalmente distribuídos e auto organizados, e geralmente são inspirados pelo comportamento de animais coletivos, como pássaros, formigas e abelhas (BONABEAU; DORIGO; THERAULAZ, 1999).

A robótica de enxame está emergindo como uma abordagem alternativa para a coordenação de um SMR. No entanto, a demonstração física de enxames de robôs é um desafio tanto de hardware quanto de software (SICILIANO; KHATIB, 2016). Para o processo de tomada de decisões cooperativas, cada robô deve ter uma



visão do ambiente ao qual está inserido e, desse modo, pode ter que executar um processamento na ordem de dezenas ou centenas de megabytes por segundo (CORKE, 2011). Além disso, a comunicação local entre todos os robôs, robustez, flexibilidade, comportamento baseado em estigmergia e controle descentralizado são desafios recorrentes para a robótica de enxame (SICILIANO; KHATIB, 2016).

Visando esses recursos, o uso de técnicas de computação suave aumentou em SMRs, resultando em arquiteturas de robôs mais flexíveis. Eles podem lidar com a infinidade de estados que os robôs podem alcançar em ambientes desconhecidos. Por exemplo, vários robôs devem explorar cooperativamente um ambiente desconhecido para encontrar e desativar minas terrestres, como em De Rango *et al.* (2018) e Palmieri *et al.* (2018). Um cenário recorrente visto em SMRs, assim como para robôs em geral, é atribuir à equipe de robôs tarefas que expõem a vida humana ao perigo. Nessas situações, os robôs podem concluir as tarefas de maneira mais eficiente/barata, adaptável e com mais robustez do que os humanos (PALMIERI *et al.*, 2018).

Esse trabalho é parte de uma pesquisa iniciada em 2016 no campo da robótica móvel autônoma. A primeira publicação, de Mendonça *et al.* (2015) apresentou uma aplicação de FCMs em robôs móveis e implementou dois sistemas de navegação usando Mapas Cognitivos *Fuzzy* Dinâmicos Híbridos (HD-FCM) e controladores lógicos *Fuzzy* hierárquicos (FLC) em um sistema composto por um robô. O próximo passo foi a construção de um protótipo e a validação do modelo cinemático do robô através da sobreposição das trajetórias simulada e real, usando um HD-FCM embarcado (MENDONÇA *et al.*, 2016).

No terceiro trabalho, de Mendonça *et al.* (2017), a arquitetura foi escalada para seu uso em sistemas baseados em robótica de enxame, aplicando também conceitos de Otimização por Colônia de Formigas (ACO) para a evolução da trajetória dos robôs. A abordagem utiliza apenas um robô liberado de cada vez no ambiente, deixando feromônios durante o percurso. O próximo robô é liberado quando o anterior termina a rota ou colide com os obstáculos. Por fim, as duas últimas contribuições dessa pesquisa utilizam tanto FLCs quanto Mapas Cognitivos *Fuzzy* (FCMs) para um grupo de robôs homogêneos trabalhando simultaneamente. Os resultados de ambas as abordagens foram comparados em dois cenários experimentais simulados, com 1 e 4 robôs operando (SOARES *et al.*, 2018; MENDONÇA *et al.*, 2019).

Assim, o principal problema do trabalho proposto consiste na busca estocástica de um número conhecido de vítimas em áreas desconhecidas. Cada robô contém um algoritmo de memória que armazena a área explorada e a distância percorrida para mapear os ambientes. Outras abordagens de mapeamento podem ser vistas em Colares e Chaimowicz (2015, 2016). Ademais, a presente contribuição aprimora os últimos resultados, de Mendonça *et al.* (2019), adicionando um terceiro ambiente nas simulações, assim como experimentos para oito robôs.

De acordo com Russell e Norvig (2009), se os robôs concluírem suas tarefas com êxito em pelo menos três ambientes, os robôs poderão ser denominados autônomos. Conseqüentemente, em cenários reais, as condições não são ideais, como nos simulados. Por esse motivo, o autor adicionou condições não ideais para todos os três ambientes: os robôs começam próximos um do outro (condição de desvio de obstáculos dinâmicos). Por fim, os experimentos com oito robôs lidam com uma maior proporção de robôs por vítima, a fim de verificar a detecção e capacidades de resgate do SMR.

A partir dos conceitos acima, pode-se elencar outras contribuições similares que utilizam os conceitos dos SMRs. Uma tarefa de exploração cooperativa é realizada pelo SMR em Almeida *et al.* (2019), na qual um mecanismo de tomada de decisão e um sistema de controle *Fuzzy* são apresentados para fornecer o comportamento dos robôs, visando um planejamento de trajetória projetado para evitar colisões. O comportamento de capturar alvos é chamado de tarefa coletiva de forrageamento, inspirado pelas ações de formigas e outros animais sociais na busca de alimentos e recursos para seus ninhos (BAYINDIR, 2016).

Outro exemplo de comportamento de forrageamento pode ser visto em Schmickl e Crailsheim (2006), inspirado pela estratégia de agregação de mofo. Nesse trabalho, a principal tarefa cooperativa consiste na localização e disposição de lixo nos locais desejados, e as subtarefas incluem exploração de cenários, localização de caminhos e uma análise da propagação de informações entre os robôs. Em resumo, para os trabalhos citados, ao alterar vítimas deste trabalho para pontos de troca de informações, tem-se que as aplicações são semelhantes. Em outras palavras, a estratégia proposta pode ser adaptada para interagir com alvos e ambientes dinâmicos, por exemplo usando um SMR para aplicações militares.

De maneira simplista, os FCMs são uma metodologia de computação suave que surgiu como uma combinação de lógica *Fuzzy* e Redes Neurais Artificiais

(RNAs). Os FCMs constituem um método computacional capaz de examinar situações durante as quais o processo de pensamento humano envolve descrições nebulosas ou incertas. A representação gráfica dos FCMs descreve as relações de causa e efeito entre os nós (conceitos), dando-nos a oportunidade de descrever o comportamento de um sistema de maneira simples e simbólica (PAPAGEORGIU, 2014).

Para garantir a operação do sistema, os FCMs incorporam o conhecimento e a experiência acumulados de especialistas que conhecem como o sistema se comporta em diferentes circunstâncias. Os FCMs têm algumas características vantajosas específicas sobre os métodos tradicionais de mapeamento: eles capturam mais informações nos relacionamentos entre conceitos, são dinâmicos, combináveis, ajustáveis e podem expressar relacionamentos ocultos.

Dessa forma, o modelo resultante pode ser usado para analisar, simular e testar a influência dos parâmetros e prever o comportamento do sistema. Os FCMs ganharam considerável interesse de pesquisa na última década e foram usados na modelagem de uma grande variedade de sistemas, como visto nos trabalhos (BEHROOZ *et al.*, 2018; MPELOGIANNI; GROUMPOS, 2018a, 2018b; ZHANG; JETTER, 2018).

Com o presente trabalho, espera-se contribuir para a área de sistemas inteligentes por meio da aplicação de uma derivação dos FCMs, um Mapa Cognitivo *Fuzzy* Dinâmico (DFCM), a um sistema de navegação para SMRs para exploração de ambientes desconhecidos, resgate de vítimas e verificação de robustez através de testes em cenários de possível estresse (8 robôs). A principal vantagem dessa abordagem é a simplificação dos processos de aquisição e representação de conhecimento usando DFCMs.

Além disso, ao comparar o tempo de processamento e a complexidade computacional do DFCM com um FLC, espera-se que a abordagem proposta seja mais simples de ser implementada, executada e incorporada a microcontroladores, resultando no uso de hardware de robôs mais compactos e de menor custo. Com isso, são mostrados os recursos de um protótipo inicial de hardware usando o Arduino® Mega 2560 – um microcontrolador conhecido em instituições de ensino por sua facilidade de uso.

## 1.1 MOTIVAÇÕES

A utilização de SMRs e seus subdomínios, como a robótica de enxame, vem ganhando destaque no panorama da robótica móvel. Nesse tipo de sistema, há o emprego de múltiplos robôs para realizar tarefas em comum em ambientes compartilhados (BAYINDIR, 2016). No entanto, embora essas equipes de robôs possam ser mais eficazes, elas também apresentam novos desafios. Os robôs podem ter cinemática não-holonômica; as informações captadas pelos sensores podem ser limitadas, e os ruídos e distúrbios externos podem dificultar o processamento das informações disponíveis. Com isso, pode-se afirmar que manter vários objetivos e realizar a fusão das informações sensoriais são desafios não triviais de projeto (HSIEH; CHIRIKJIAN, 2014).

Uma das possíveis utilizações dos SMRs está em operações de salvamento de vítimas, cenário recorrente na atualidade. O principal desafio encontrado por equipes de salvamento em desastres é o tempo de resposta. As taxas de encontrar pessoas e/ou animais sobreviventes diminuem significativamente com o tempo e, devido a fatores como desidratação e ferimentos, são reduzidas drasticamente após 48 horas depois da ocorrência (DIN *et al.*, 2018). Portanto, os socorristas precisam se mover o mais rápido possível, evitando obstáculos estáticos e dinâmicos, e cobrir a maior área para alcançar todas as vítimas em potencial para salvar suas vidas (SICILIANO; KHATIB, 2016). Uma alternativa emergente ao uso de socorristas humanos está surgindo nos robôs de resgate em desastres (RD) ou de busca e resgate (B&R).

As tarefas de B&R são caracterizadas pela exploração de cenários e exigem a determinação de todos os locais das vítimas, de forma a reduzir significativamente as chances de prejudicar a vida dos socorristas. Essas restrições tornam a B&R tarefas desafiadoras para os seres humanos, e uma possível solução para a desvantagem é o uso de sistemas de robôs, em particular os SMRs (DIN *et al.*, 2018). No entanto, devido à sua relativa novidade, os robôs B&R não foram amplamente adotados, mas esse cenário está mudando à medida que a tecnologia evolui. No período entre 2001 e 2012, robôs ou SMRs foram utilizados 37 vezes em desastres meteorológicos, geológicos, de mineração e causados por humanos, obtendo sucesso em 30 situações (81%). Desde 2012 tem-se um aumento constante

na utilização de SMRs em desastres, com o uso em aproximadamente 30 desastres em seis países, principalmente em colapsos na mineração e em edifícios (SICILIANO; KHATIB, 2016).

É possível encontrar na literatura uma variedade de trabalhos que abordam tarefas de exploração e B&R utilizando SMRs baseados na robótica de enxame, similarmente ao que ocorre neste trabalho. No primeiro (DIN *et al.*, 2018), apresenta um enxame de robôs baseado em comportamento para executar operações de B&R em ambientes desconhecidos. Nesse trabalho, o critério de parada para as simulações é um número fixo de etapas. Ademais, um líder é selecionado dinamicamente para guiar o enxame. Dessa forma o controle, realizado por um FLC, é centralizado, porém dinâmico.

O trabalho de Bakhshipour, Ghadi e Namdari (2017) possui uma abordagem semelhante, utilizando um robô líder para guiar o enxame em operações de B&R. Os autores utilizaram um algoritmo heurístico para encontrar a solução mais adequada para resgatar as vítimas em um local de desastre, ou seja, um planejamento da trajetória dos robôs. Nesse caso, pode-se empregar a utilização de uma câmera aérea apontada para o chão do local a ser explorado, fato que dificulta sua implementação em situações reais pela dificuldade de visualização das vítimas.

Outro exemplo pode ser visto no trabalho de Venayagamoorthy, Grant e Doctor (2009), que combina aspectos da inteligência de enxame com controladores lógicos *Fuzzy* para uma tarefa de pesquisa coletiva. A primeira abordagem consistiu na utilização de um algoritmo de otimização de enxame de partículas (PSO) fundido com a lógica *Fuzzy*, criando assim um enxame de robôs fuzzificado, enquanto que a segunda abordagem utilizou controladores *Fuzzy*-enxame. Por fim, Couceiro *et al.* (2014) apresentam uma pesquisa de métodos em aplicações de B&R de robótica de enxame, discutindo suas vantagens e desvantagens.

## 1.2 OBJETIVOS

Os objetivos deste trabalho são divididos em geral e específicos, explanados como segue.

### 1.2.1 Objetivo geral

O objetivo geral deste trabalho consiste no desenvolvimento de um SMR autônomo capaz de realizar tarefas de busca de alvos em ambientes simulados. O controle de movimento e orientação dos robôs foi realizado por três estratégias baseadas em lógica *Fuzzy*, a primeira baseada em um controlador *Fuzzy* do tipo Mamdani (FLC), a segunda por um Mapa Cognitivo *Fuzzy* Dinâmico (DFCM) e por fim um Mapa Cognitivo *Fuzzy* Dinâmico otimizado por um algoritmo de Otimização por Colônia de Formigas (DFCM-ACO).

### 1.2.2 Objetivos específicos

A partir do objetivo geral, foram traçados os objetivos específicos, que são constituídos por:

1. Desenvolver uma arquitetura composta por subcomportamentos para o sistema multirrobôs, com conceitos baseados na robótica de enxame;
2. Projetar os controladores baseados em lógica *Fuzzy* para o controle de movimento e orientação dos robôs;
3. Comparar os resultados das estratégias utilizadas para tempo de processamento computacional, área explorada, distância percorrida por robô e consumo de bateria simulado para três cenários simulados, utilizando três configurações de sistema: um, quatro e oito robôs.

## 1.3 ESTRUTURA DO TRABALHO

O presente trabalho encontra-se dividido em 5 capítulos. Após o capítulo introdutório, o Capítulo 2 objetiva apresentar os fundamentos teóricos utilizados para o desenvolvimento da proposta apresentada. Nele, são discutidas brevemente

fundamentos da robótica móvel autônoma, conceitos da robótica de enxame e de Redes Neurais Artificiais (RNAs), que compõe a arquitetura dos FCMs. Com isso, apresenta-se a modelagem e arquitetura utilizadas para os robôs desse trabalho. Por fim, o capítulo apresenta uma síntese dos sistemas computacionais inteligentes baseados em lógica *Fuzzy* utilizados para o controle do SMR.

O terceiro capítulo aborda a aplicação dos conceitos fundamentados no Capítulo 2 no presente trabalho. O modelo cinemático utilizado é primeiramente exibido, seguido da arquitetura proposta para a tomada de decisões dos robôs. Com isso, os sistemas de controle desenvolvidos são apresentados. Os resultados obtidos com as diferentes estratégias de controle são mostrados no Capítulo 4 juntamente com as discussões acerca dos parâmetros analisados. Por fim, no Capítulo 5 são realizadas as considerações finais do trabalho e seus possíveis desdobramentos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada uma visão global acerca da robótica móvel autônoma, de enxame e sistemas multirrobo. Na sequência, são descritos o processo de modelagem cinemática para robôs móveis e a arquitetura de subsunção utilizada para a modelagem dos processos de tomada de decisão do SMR. Assim, são explanados os conceitos dos sistemas inteligentes aplicados no trabalho: RNAs, sistemas *Fuzzy*, FCMs e sua extensão DFCM e a Otimização por Colônia de Formigas (ACO). Por fim, são expostos trabalhos relacionados.

### 2.1 ROBÓTICA MÓVEL AUTÔNOMA

Entre as aplicações de robôs controlados remotamente pode-se citar a execução de tarefas como inspeção, fotografia aérea, identificação de minas terrestres, entre outros (SUTTER *et al.*, 2018). Entretanto, esses tipos de robôs não são autônomos, uma vez que utilizam seu sensoriamento para assessorar seus operadores quando há acesso remoto a locais perigosos ou inacessíveis para os seres humanos. Quando são controlados externamente, mas também realizam subtarefas automaticamente, podem ser denominados semiautônomos. Um exemplo recorrente são os drones mais recentes, nos quais o piloto automático estabiliza o voo enquanto o operador determina a trajetória. (BEN-ARI; MONDADA, 2018).

Para ser considerado completamente autônomo, um robô pode obter informações sobre o ambiente, trabalhar por períodos prolongados sem intervenção humana, mover-se sem assistência humana, evitar situações que são prejudiciais aos seres vivos e a si mesmos (WAHDE, 2016). Um robô autônomo também pode aprender ou adquirir novos conhecimentos, como ajustar-se a novos métodos para realizar suas tarefas ou adaptar-se às mudanças do ambiente. Dessa forma, pode-se dizer que robôs móveis autônomos devem ter o mínimo de recursos e autonomia para que operem em ambientes não estruturados, dinâmicos, parcialmente observáveis ou incertezas no sensoriamento e nas ações (NIKU, 2011; XIA, 2016).



Como não dependem de um operador, os robôs móveis devem possuir um sistema de tomada de decisões para realizar tarefas sem intervenção humana, como transportar material enquanto exploram terrenos incertos (interior de casas) e em ambientes dinâmicos (operar perto de pessoas, em meio ao tráfego de veículos). Alguns exemplos são os robôs aspiradores, como o Roomba® (IROBOT, 2019), e os robôs autônomos vistos em armazéns de grandes empresas, como a Amazon®. Nela, robôs são utilizados para controle dos estoques e separação de produtos. Um exemplo dessa utilização é mostrado na Figura 2.1. Mesmo operando em fábricas muito estruturadas, esses robôs compartilham o ambiente com seres humanos e, portanto, seus sensores devem ser extremamente confiáveis para evitar acidentes (SIEGWART; NOURBAKHS, 2014).

**Figura 2.1 – Robô móvel utilizado em armazéns**



Fonte: (RBR, 2018).

Entretanto, outras aplicações parecem tomar para si o foco de pesquisas na área da robótica autônoma e semiautônoma devido à complexidade das tarefas envolvidas: veículos autônomos e exploradores (*rovers*) espaciais. O primeiro caso pode ser visto na Figura 2.2 por meio do Tesla® Model S, lançado em 2012 nos Estados Unidos da América. Nessa aplicação, tanto modelagem quanto controle são complexos devido aos ambientes serem altamente incertos e também dos padrões de segurança impostos a esse tipo de veículo.

No espaço, pode-se destacar os *rovers* semiautônomos da expedição *Curiosity* para Marte, ativos desde seu lançamento em 2012. Um operador controla da Terra os detalhes de cada missão (rotas e experimentos científicos a serem realizados), enquanto que os *rovers* tem capacidade de desvio de obstáculos. Isso ocorre pela diferença de tempo entre o comando do operador e seu recebimento pelos *rovers* (tempo de resposta), o que poderia causar danos a esses veículos. Nesse caso, a navegação autônoma requer a integração da medição de profundidade com um sistema de visão, por análise de visão estéreo ou demais métodos determinantes de alcance. Ela também requer regras de comportamento heurísticas para o dispositivo navegar nesse ambiente desconhecido (NIKU, 2011).

**Figura 2.2 – Tesla Model S em modo de condução autônoma**



Fonte: (DUNDON; COURIER, 2018).

Atualmente, parte considerável da pesquisa e desenvolvimento no campo da robótica almeja tornar os robôs mais autônomos por meio do refinamento de sensores já existentes e da criação de novos, que possibilitam estratégias de controle ainda mais inteligentes para os robôs. Essa evolução nas tecnologias sensoriais leva à utilização de técnicas mais flexíveis e adaptativas para esses sistemas, como as baseadas em lógica *Fuzzy* apresentadas neste trabalho. Esses esforços de pesquisa levaram ao desenvolvimento de novas áreas ou subdivisões no estudo de sistemas

robóticos, como a visão computacional e as interações entre humanos e máquinas (BEN-ARI; MONDADA, 2018; HEXMOOR, 2013).

Há competições que fomentam e criam desafios para a área da robótica móvel autônoma. Como exemplos, tem-se as promovidas pela *Defense Advanced Research Projects Agency* (DARPA), como a *DARPA Grand Challenge* (DGC) no campo de veículos autônomos, e a *DARPA Robotics Challenge* (DRC) para robôs humanoides. Além de organizar as competições, a DARPA também é encarregada de patrocinar as equipes participantes.

O DGC foi criado inicialmente para estimular o desenvolvimento de tecnologias necessárias para criar os primeiros veículos terrestres autônomos, capazes de concluir um percurso *off-road* dentro de um tempo limitado. A partir disso, os eventos seguintes adicionam dificuldade aos desafios, como a inserção de ambientes simulados e a utilização dos veículos em situações de manutenção de emergência. A primeira edição do evento ocorreu em 2004 e terminou sem vencedor declarado, pois nenhuma das equipes conseguiu cumprir o desafio (DARPA, 2019).

Semelhantemente, o DRC tem como objetivo desenvolver robôs terrestres semiautônomos que poderiam executar tarefas em ambientes que oferecessem perigo aos seres humanos. O evento foi realizado entre 2012 e 2015, dividido em três competições: *Virtual Robotics Challenge*, um desafio virtual de robótica (2013) e dois desafios em ambiente real: *DRC Trials* (2013) e *DRC Finals* (2015). Além de objetivar o incentivo a pesquisa em robótica, o DRC também visava o emprego de soluções de *hardware* mais acessíveis, financiando simuladores de robôs, como o Gazebo (DARPA, 2015).

Nos âmbitos do ensino fundamental e médio, o maior campeonato é a *FIRST Robotics Competition*. Nele, as equipes devem arrecadar fundos e criar uma marca para disputar uma das quatro possíveis divisões: duas para robôs programados LEGO® e outras duas de estilo livre (FIRST, 2019). Outra competição relevante é a Robocup, criada em 1997 e focada no futebol de robôs. Seu objetivo é desenvolver, até o final do século 21, uma equipe de robôs humanoides autônomos capazes de derrotar a seleção humana campeã mundial (ROBOCUP, 2019).

No Brasil também podem ser destacadas algumas competições de robótica, como a *Latin American Robotics Competition/Brazilian Robotics Competition* (LARC/BRC), promovendo competições de futebol, robôs domésticos, para resgate e educacionais (CBR, 2019).

Recentemente, pode-se destacar a *ROSI Challenge*, competição promovida no Simpósio Brasileiro de Automação Inteligente (SBAI) 2019. Ela consiste na simulação de um robô móvel para inspeção de transportadores de correia utilizados na mineração, trabalho atualmente realizado de forma presencial por humanos. Entretanto, a grande extensão desses transportadores e os riscos envolvidos – como incêndios e risco de choques mecânicos – sugere o emprego de soluções mais seguras, como um robô autônomo. Com sua utilização, os operadores podem gerenciar os aspectos operacionais dos transportadores remotamente e, além disso, podem aumentar a eficiência de processos envolvidos por meio da padronização da coleta de dados (SBA, 2019).

Como dito anteriormente, a robótica autônoma vem emergindo como tema recorrente devido à sua participação nos conceitos da Indústria 4.0. Esse termo foi criado em 2011 e, em resumo, refere-se à quarta revolução industrial (RAJ *et al.*, 2019). Nela, as máquinas industriais ou sistemas ativos são caracterizados pelo uso de sistemas inteligentes em rede, a fim de proporcionar maior flexibilidade, eficiência e integração com toda a cadeia produtiva. Desse modo, os robôs podem promover o compartilhamento de dados e informações, além de realizar atividades específicas (ROBLA-GOMEZ *et al.*, 2017).

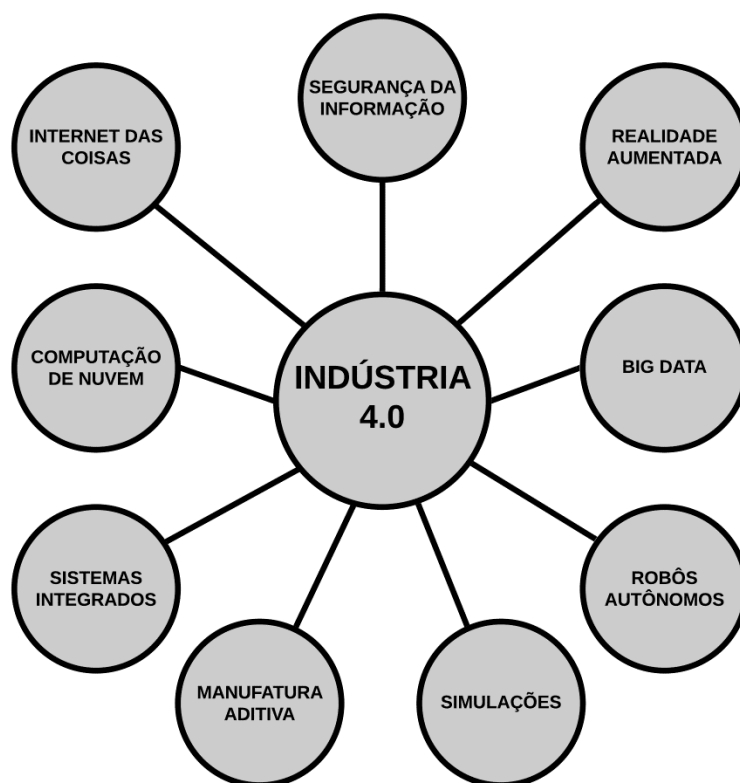
Portanto, essas entidades podem estar alinhadas a funções mecânicas na colaboração logística, em questões variáveis de monitoramento e até na identificação e busca de vítimas em acidentes de larga escala. A partir do amplo conhecimento do ambiente, com grande quantidade de informações obtidas (*big data*), é possível, por meio de sistemas inteligentes, tomar decisões com maior precisão de forma colaborativa.

A Indústria 4.0 possui três princípios básicos: integração entre sistemas computacionais e físicos (do inglês, *Cyber-physical Systems*), Internet das Coisas (IoT) e Fábricas Inteligentes. A decomposição desses princípios resulta em nove pilares de avanço tecnológico (MARIANI; BORGHI, 2019), apresentados abaixo e na Figura 2.3:

1. Robótica autônoma;
2. Integração de sistemas verticais e horizontais;
3. Simulação;
4. IoT industrial;

5. Cibersegurança (do inglês *cybersecurity*);
6. Computação em nuvem;
7. Manufatura aditiva, ou seja, a operação das diversas ferramentas que compõem o processo conhecido como Impressão 3D;
8. Utilização de Realidade Aumentada;
9. *Big data* e analítica.

Figura 2.3 – A integração dos conceitos da Indústria 4.0



Fonte: Adaptado de (MACIEL, 2018).

## 2.2 SISTEMAS MULTIRROBÔS E ROBÓTICA DE ENXAME

A área de pesquisa dos SMRs engloba a utilização de um conjunto de robôs para o desempenho de comportamentos que convergem para a realização de um objetivo em comum, seja em terra, água ou ar (SICILIANO; KHATIB, 2016). Por sua vez, uma das definições para o termo “robótica de enxame” é dada como o conjunto de técnicas inspiradas na natureza para controlar grandes grupos de robôs de natureza estrutural relativamente simples. Exemplos de enxames na natureza podem

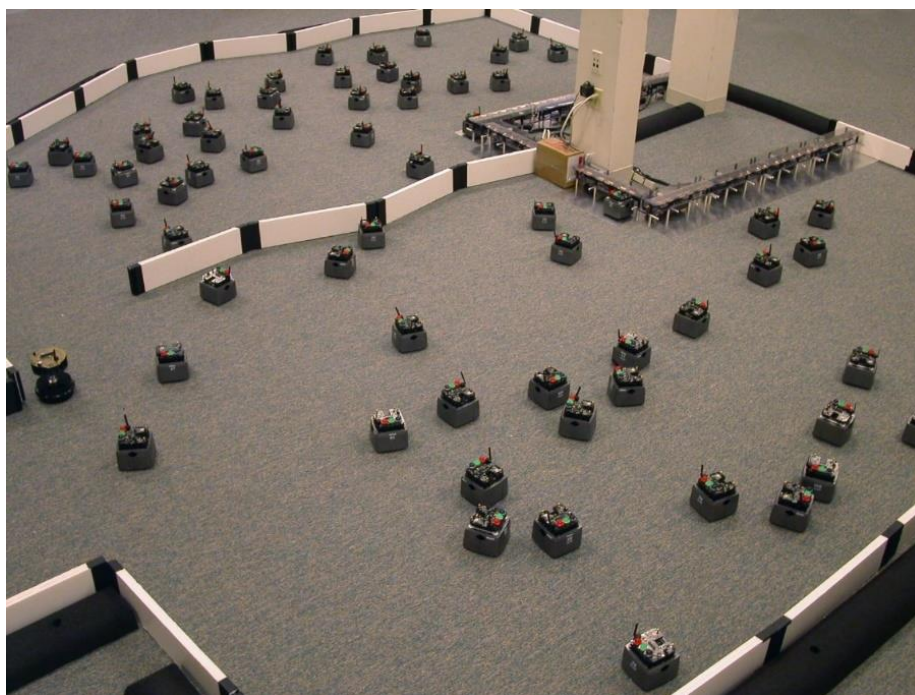


ser vistos em uma revoada de pássaros ou em um cardume (GAZI; PASSINO, 2011; RIZK; AWAD; TUNSTEL, 2018). Nesse campo de pesquisa, oriundo dos SMRs, os robôs muitas vezes desconhecem as ações de outros robôs distantes, ou seja, a comunicação é feita apenas entre robôs próximos, dependendo da estigmergia na maior parte do tempo (SICILIANO; KHATIB, 2016).

Um sistema estigmergico possui um processo que sofre alterações a cada transformação no ambiente. Em outras palavras, as características do ambiente servem como estímulos aos comportamentos do sistema. Nos enxames naturais, a estigmergia é frequentemente a força motriz por trás dos fenômenos, como trilhas de formigas. Portanto, a estigmergia é um importante campo de pesquisa em robótica de enxames, por exemplo, em trilhas de feromônios artificiais (HAMMAN, 2010).

Outra característica necessária aos SMRs de enxame diz respeito a dificuldade das tarefas propostas: apenas há necessidade da utilização de um enxame de robôs se as tarefas apenas puderem ser executadas a nível de equipe, ou também se o tempo de conclusão for um fator determinante para o cumprimento dos objetivos. Em outras palavras, o aumento no número de robôs em um grupo pode reduzir significativamente o tempo de conclusão da tarefa (KHAMIS; HUSSEIN; ELMOGY, 2014). Um exemplo de SMR é mostrado na Figura 2.4.

**Figura 2.4 – Exemplo de sistema multirrobôs**



Fonte: (MCLURKIN, 2019).

Os robôs de um SMR podem ter diferentes tipos de comportamentos, como agrupamento, encadeamento, busca, agregação e forrageamento. Esses comportamentos são classificados em coletivo (este trabalho, como discutido anteriormente), cooperativo (os robôs estão cientes dos outros), colaborativo (cada robô ajuda os outros a alcançar seus objetivos) e coordenativo (os robôs estão cientes dos outros, mas não compartilham um objetivo comum) (PARKER, 2008).

Outro aspecto inerente a robótica de enxame é a coordenação de movimento. Ela pode ser relativa apenas a outros robôs, ao ambiente, a agentes externos e combinações entre eles (SICILIANO; KHATIB, 2016; DIN *et al.*, 2018). Neste trabalho, os movimentos dos robôs são relativos às três categorias discutidas: eles mudam seus movimentos devido a outros robôs, obstáculos estáticos e dinâmicos (ambiente) e detecção de vítimas (agentes externos).

Para a construção de sistemas multiagentes (SMAs) ou SMRs, a primeira definição deve ser como os robôs serão desenvolvidos e como eles interagirão entre si e com o ambiente. Neste trabalho, o SMR utilizado tem aspectos inspirados nos conceitos da robótica de enxame: robustez, flexibilidade e escalabilidade (DORIGO; BONABEAU, 1999; BENI, 2004). Para SMRs, a robustez está ligada à conclusão dos objetivos apesar do mal funcionamento dos robôs ou de distúrbios ambientais. A flexibilidade aborda a capacidade de gerar soluções para tarefas utilizando diferentes estratégias de coordenação em resposta a ambientes dinâmicos. Ademais, um SMR é escalável para não ser perturbado devido a alterações no tamanho do grupo dos robôs.

Outros conceitos da robótica de enxame também foram utilizados para desenvolver as simulações neste trabalho. Os robôs são redundantes (ou seja, facilmente intercambiáveis) e a coordenação é descentralizada: a perda de um robô é compensada imediatamente pelos outros e/ou pela implantação de outro agente, não interrompendo a operação do sistema. Por fim, visou-se a simplicidade estrutural e de componentes para facilitar os reparos e diminuir custos de implementação, resultando em um sistema menos propenso a falhas (CALVO *et al.*, 2012).

Além disso, neste trabalho a percepção do SMR é distribuída, de forma a possibilitar robustez contra distúrbios locais do ambiente (obstáculos dinâmicos). O sistema proposto também possui um algoritmo de comunicação local, responsável pelas únicas informações que os robôs compartilham entre eles: o número de vítimas já resgatadas - para interromper as simulações quando os robôs encontram todas as

vítimas - e os obstáculos já detectados. Por fim, os algoritmos de memória armazenam a posição e a direção de cada robô para mapeamento dos ambientes.

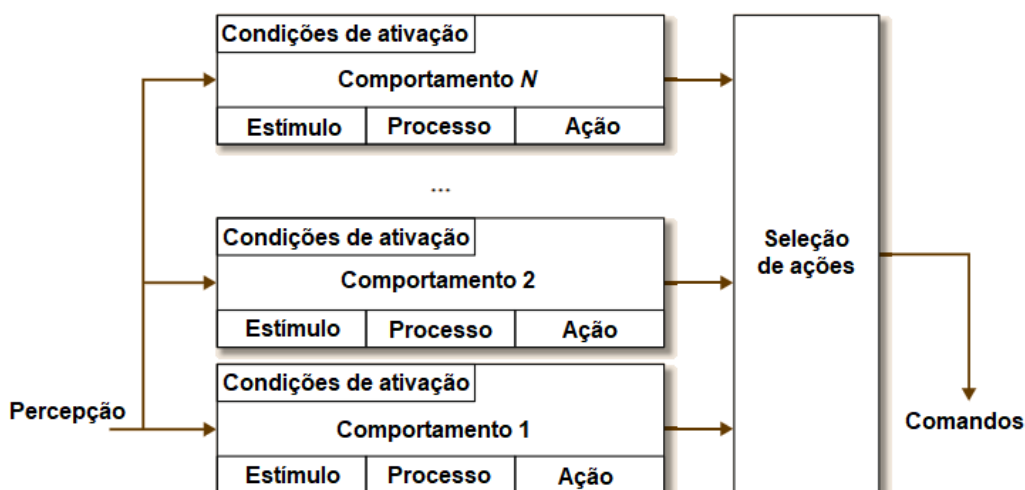
Com esses recursos, espera-se que os robôs tenham autonomia suficiente para operar em ambientes desconhecidos ou semidesconhecidos. Por exemplo, a flexibilidade de operar sob sensores de ruído ou mau funcionamento, falha de robôs, ambientes dinâmicos e cenários de estresse.

### 2.3 PRINCÍPIOS DA ARQUITETURA DE SUBSUNÇÃO DOS ROBÔS MÓVEIS

Uma arquitetura pode ser entendida como o conjunto de instruções que definem o comportamento/funcionamento de um sistema de acordo com suas características sensoriais e físicas. No contexto dos robôs móveis autônomos, Brooks (1986) desenvolveu a arquitetura de subsunção, uma das mais utilizadas por sua simplicidade de implementação.

A arquitetura de subsunção foi provavelmente uma das primeiras tentativas de compor comportamentos complexos por meio de camadas de comportamentos, hierárquicas ou não, mais simples (KUMOVA; HEYE, 2017). Essa característica vai de encontro ao conceito de organização de comportamentos através de representações simbólicas dos ambientes, como visto nos conceitos clássicos da inteligência artificial (BROOKS, 1986).

Figura 2.5 – Distribuição de comportamentos



Fonte: Adaptado de (SICILIANO; KHATIB, 2016).



Uma possível interpretação para essa arquitetura pode ser vista na Figura 2.5. Em resumo, o sistema de percepção ativa os comportamentos por meio de um conjunto de condições (estímulos externos). Dessa forma, cada comportamento (ou subcomportamento), processa as informações recebidas e direciona as ações correspondentes, que são transcritas como comandos para o sistema.

No contexto deste trabalho, fatores como o sistema de mapeamento (navegação), recorrente necessidade de tomada de decisões e necessidade de cumprimento de sub objetivos variados sugerem a utilização de uma arquitetura baseada em subsunção. Nela, a adaptação dos agentes (robôs) é observada pela adição de novos subcomportamentos ou módulos. Ademais, a arquitetura de subsunção é inicialmente classificada como reativa.

No presente trabalho, há hierarquia nos comportamentos do SMR. Cada subcomportamento do sistema respeita um nível de prioridade, partindo das camadas de baixo nível (reativas) para as de alto nível, representadas pelos controladores inteligentes e as características deliberativas dos robôs. A modelagem desses subcomportamentos, assim como a utilização da arquitetura de subsunção, foi inspirada pelo princípio da incompatibilidade de Zadeh (1976). Nele, afirma-se que, quanto maior a complexidade de um sistema, mais improvável e impreciso se torna a previsão de suas reações aos ambientes. Esse fato é visto no trabalho de Braitenberg (1986), que modela o funcionamento de SMRs baseado em emoções e comportamentos humanos.

Por fim, os subcomportamentos utilizados neste trabalho são alternados em decorrência dos dados de sensoriamento dos robôs, funcionando de forma sequencial, enquanto que o SMR funciona paralelamente. Em outras palavras, a dado instante, todos os robôs podem ter um subcomportamento diferente ativado. Porém, em cada robô, apenas um subcomportamento opera por vez.

## 2.4 REDES NEURAIIS ARTIFICIAIS

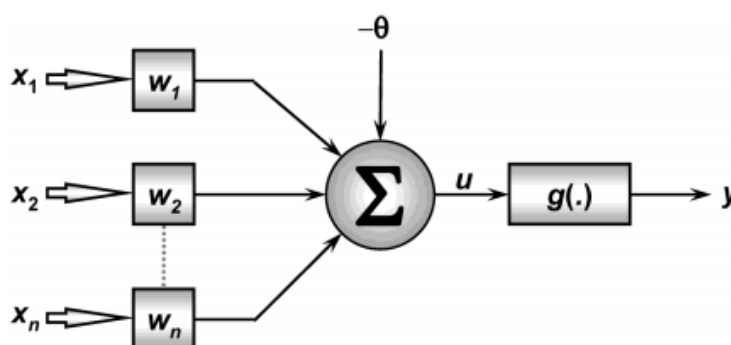
As Redes Neurais Artificiais (RNAs) possuem base na natureza, inspiradas de forma simplista na forma de um neurônio biológico. De modo geral, uma RNA é um processador distribuído paralelamente, constituído de pequenas estruturas que,

combinadas, tem a capacidade de armazenar conhecimento a partir da experiência. As RNAs possuem duas características principais (HAYKIN, 2009):

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizado;
2. As forças de conexão interneurônio, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento adquirido.

Na Figura 2.6 é mostrado uma das possíveis formas de neurônio artificial, na qual  $\{x_1, \dots, x_n\}$  é o vetor contendo as entradas da rede,  $\{w_1, \dots, w_n\}$  correspondem aos pesos sinápticos,  $\Sigma$  é o combinador,  $g(\cdot)$  a função de ativação,  $\Theta$  o limiar de ativação do neurônio e  $y$  a saída.

**Figura 2.6 – Representação gráfica de um neurônio artificial**



Fonte: Adaptado de (DA SILVA *et al.*, 2017).

A representação de uma RNA é feita na forma de grafos direcionados, de maneira que os nós representam os elementos de processamento (neurônios) e as arestas são caminhos de condução unidirecional de sinais, garantindo que os elementos recebam conexões de entradas. As RNAs possuem características que as permitem ter aplicações em diferentes áreas de pesquisa (HAYKIN, 2009; DA SILVA *et al.*, 2017):

1. Não-linearidade: podem ser lineares ou não, já que são constituídas de interconexões não-lineares de neurônios;

2. Adaptação por experiência: a partir da modificação dos parâmetros internos da rede, é possível adquirir conhecimento através da experimentação sucessiva;
3. Tolerância a falhas: mesmo em condições adversas, em que parte da estrutura é comprometida, uma RNA é capaz de operar graças a elevada quantidade possível de interconexões entre seus neurônios;
4. Capacidade de generalização: uma RNA tem a capacidade de realizar a generalização do conhecimento adquirido através de treinamento, possibilitando a obtenção de informações não apenas de um padrão selecionado, mas também conseguindo verificar a confiabilidade das informações obtidas.

Devido à sua flexibilidade, as RNAs podem ser usadas em diferentes aplicações, seja como aproximador universal de funções, controle de processos e robótica (HE *et al.*, 2017; MEYER, 1998), classificador de padrões (BAZAN *et al.*, 2019; PALÁCIOS *et al.*, 2016), previsão de séries temporais, otimização de funções e visão computacional (AHMED; HEMAYED; FARAG, 1999).

Resumidamente, o funcionamento de uma RNA é composto pela apresentação dos sinais de entrada para a rede. Cada sinal é multiplicado pelo respectivo peso sináptico associado à sua influência na saída. Consequentemente, um nível de atividade é produzido pela soma ponderada dos sinais (combinador) e, por fim, se esse nível excede um limitante uma saída é produzida (HAYKIN, 2009).

## 2.5 SISTEMAS FUZZY

A compreensão dos seres humanos em relação à maioria dos processos físicos baseia-se em aspectos imprecisos. Quando comparada com a precisão usualmente utilizada por computadores, essa imprecisão pode ser útil para modelagem de sistemas. Assim, a capacidade de incorporar essa forma de raciocínio em problemas até então intratáveis e complexos é uma das características principais da lógica *Fuzzy*. Nesse contexto, pode-se afirmar que o impacto da utilização da lógica *Fuzzy* seja até mesmo desfavorável para problemas que exigem considerável

precisão, como a usinagem de componentes com precisão na escala microscópica. No entanto, problemas que não exigem tamanha escala de precisão, como manobrar e realizar a navegação de um veículo, se beneficiam dessa lógica (ROSS, 2010).

Para problemas de engenharia, aspectos como a precisão vêm acompanhados de uma elevação significativa nos custos e prazos de desenvolvimento e produção. Ao considerar o uso da lógica *Fuzzy* para um determinado problema, deve-se considerar a necessidade de explorar a tolerância a imprecisões. Uma precisão alta não apenas determina o aumento dos custos, mas também implica no aumento da dificuldade em tratar o problema (ROSS, 2010).

A partir da constante aquisição de informações sobre um problema, sua formulação e solução tornam-se proporcionalmente menos incertas. Contudo, problemas caracterizados por pouca informação e complexos são caracterizados por um alto grau de incerteza. Em um sistema, as incertezas podem ser de diferentes naturezas, dentre elas (KLIR, 2007):

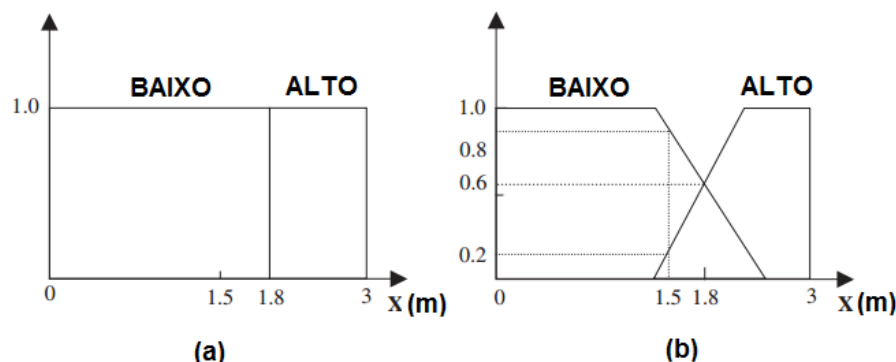
1. Imprecisa (não nítidas, incertas, imprecisas, aproximadas);
2. Vaga (não específicas, amorfas);
3. Ambígua (muitas opções, contraditórias);
4. Devidas à ignorância (dissonantes, sem conhecimento);
5. Devidas à variabilidade natural (conflituosas, aleatórias, caóticas ou imprevisíveis).

Em Klir (2007), Lotfali A. Zadeh, o criador da lógica *Fuzzy* (ZADEH, 1965), apresenta alguns exemplos simples dessas naturezas, que podem ser expressados em termos das declarações de uma pessoa sobre quando ela deve retornar a um local atual no tempo. A afirmação "retornarei em breve" é vaga, enquanto a afirmação "voltarei em alguns minutos" é imprecisa – não se sabe se o primeiro está associado a alguma unidade de tempo, enquanto que o segundo está associado a uma incerteza conhecida por estar na ordem de minutos.

Um exemplo de um comportamento *Fuzzy* é a busca pela formação de uma "classe de pessoas altas". Essa classe pode mudar de acordo com a percepção de mundo de quem a modela, ou seja, diferentes pontos de vista expressos por termos linguísticos (ZADEH, 1965). A Figura 2.7 mostra a comparação entre possíveis

sistemas de lógica *crisp* (onde há pertinência binária a um certo conjunto) e *Fuzzy*, para a análise da altura de um indivíduo.

Figura 2.7 – Comparação entre lógica *crisp* (a) e *Fuzzy* (b)



Fonte: Adaptado de (PEDRYCZ; GOMIDE, 2007).

Dessa forma, conjuntos *Fuzzy* oferecem uma forma de modelagem de informações cujos elementos podem pertencer a diferentes graus de associação (pertinência), o que auxilia no entendimento de conceitos comumente encontrados no mundo real - como baixa tolerância e pequeno erro de aproximação - que são exemplos de conceitos aos quais a quantificação sim-não não se aplica ou se torna restritiva. Em outras palavras, nesses conceitos não há como quantificar os limites booleanos, pois há elementos cuja associação aos conceitos é apenas parcial e não pode ser descrito somente por 0 (não pertencente) e 1 (pertencente) (PEDRYCZ; GOMIDE, 2007).

Um elemento é pertencente a dado conjunto *Fuzzy*  $A$  a partir de sua atribuição a diferentes níveis de pertinência. Essa quantificação é realizada pelas chamadas funções de pertinência. Assim, quanto maior o grau de pertinência  $A(x)$ , mais forte é o nível de pertencimento desse elemento a  $A$  (ZADEH, 1965).

Como pode-se observar na Figura 2.6, toda função de pertinência é associada a um termo linguístico – nesse caso “baixo” e “alto” são as funções. Com isso, é feito um processo de inferência (operações *Fuzzy*) a partir de uma base de regras estabelecidas pelo operador. Duas formas de inferência são predominantes na literatura: o método de Takagi-Sugeno (TAKAGI; SUGENO, 1985) e o de Mamdani, (1974), utilizado neste trabalho. No último, a base de regras opera da seguinte forma:

$$SE < condição > ENTÃO < ação >$$

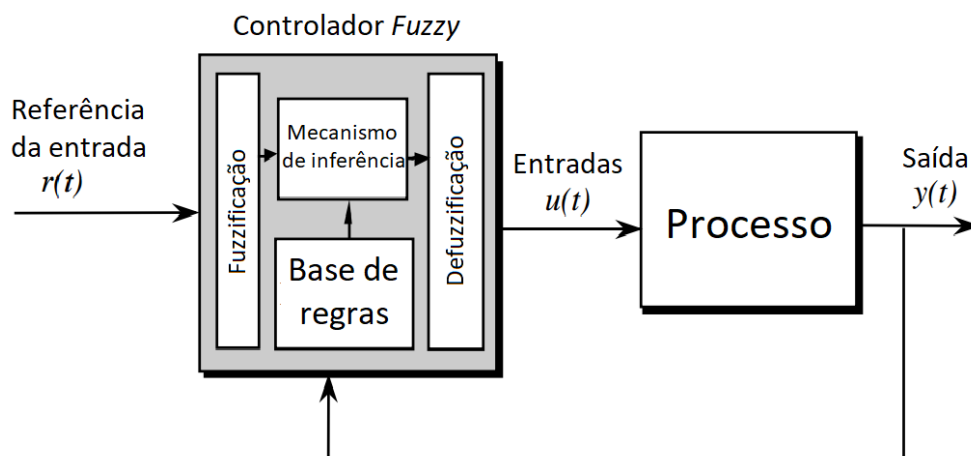
De modo que:

- <condição>: a valor da variável  $X$  é  $E$  (entrada);
- <ação>: a variável  $Y$  (saída) assume valor  $S$ .

Dessa forma, um Controlador Lógico *Fuzzy* (FLC) do tipo Mamdani pode ser descrito a partir do diagrama da Figura 2.8. Seu funcionamento pode ser descrito por quatro elementos (PASSINO; YURKOVICH, 1998):

1. Base de regras: contém a quantificação *Fuzzy* do conhecimento do operador baseado em sua descrição linguística das ações de controle necessárias;
2. Mecanismo de inferência: sistema de tomada de decisões que interpreta o conhecimento do especialista em relação ao sistema modelado;
3. Interface de fuzzificação: realiza a transformação das entradas do controlador de acordo com as funções de pertinência para que o mecanismo de inferência possa ativar e aplicar as regras correspondentes;
4. Interface de defuzzificação: transformação das saídas do mecanismo de inferência em saídas do controlador.

Figura 2.8 – Diagrama de blocos de um Controlador Lógico *Fuzzy*



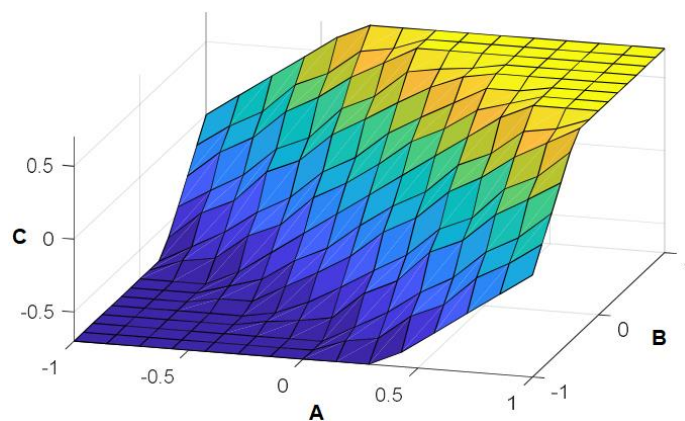
Fonte: Adaptado de (PASSINO; YURKOVICH, 1998).

A partir das combinações entre entradas e saídas de um sistema *Fuzzy*, pode-se gerar uma superfície *Fuzzy* correspondente como forma de avaliação das regras utilizadas. Para auxiliar na construção de um FLC, pode-se seguir as seguintes etapas (ROSS, 2010):

1. Determinação do universo de discurso das variáveis, ou seja, os limites máximos e mínimos que cada uma possui;
2. Definição dos termos linguísticos associados a cada função de pertinência, assim como os níveis de pertinência para cada faixa de valores;
3. Construção da base de regras para o mecanismo de inferência escolhido a partir dos parâmetros de atuação desejados.

Um exemplo de superfície *Fuzzy* é mostrado na Figura 2.9. Nela, é possível observar a relação não linear entre as duas variáveis de entrada – *A* e *B* – com uma variável de saída *C*.

**Figura 2.9 – Exemplo de superfície *Fuzzy***



Fonte: Adaptado de (KORKMAZ *et al.*, 2018).

### 2.5.1 Mapas Cognitivos Fuzzy

Os Mapas Cognitivos *Fuzzy* (FCMs) são sistemas dinâmicos para modelar o conhecimento causal. Eles foram introduzidos por Kosko (1986) como uma extensão

dos mapas cognitivos desenvolvidos por Axelrod (1976) e Tolman (1948). Esses mapas cognitivos são constituídos por um conjunto de nós vinculados por conexões direcionadas e ponderadas. Os nós representam os conceitos relevantes para uma determinada operação. As conexões (relações causais) entre esses conceitos são representadas por setas orientadas para mostrar a direção da influência, e são ponderadas para mostrar um efeito de promoção ou inibição (PAPAGEORGIOU, 2014).

Nos FCMs, os conceitos são interconectados por números *Fuzzy* que simbolizam conceitos linguísticos. Essas conexões modelam as relações entre conceitos através de um nível ou intensidade de causalidade (NÁPOLES *et al.*, 2016). Combinado com a representação gráfica de um FCM – um grafo orientado – esses recursos tornam os FCMs simples e intuitivos de entender em termos de seu modelo formal e processamento (ACAMPORA; PEDRYCZ; VITIELLO, 2015). No entanto, tem uma desvantagem de não modelar o tempo (MENDONÇA *et al.*, 2013a).

De forma geral, a modelagem de FCMs incorpora o conhecimento e experiência de especialistas para a operação do sistema abordado, possuindo características vantajosas sobre outros métodos: eles são capazes de abstrair mais informações dos relacionamentos, combináveis, ajustáveis e podem expressar relacionamentos ocultos, além de poder ser decompostos em subestruturas mais simples para facilitar a modelagem (PAPAGEORGIOU, 2014).

Seus usos estão nas áreas de modelagem de sistemas, simulação e aplicações embarcadas. Os FCMs vêm construindo uma popularidade emergente com base em sua facilidade de uso e capacidade de combinar conceitos qualitativos e quantitativos. Entre as áreas de aplicação, podem ser citadas:

- Predição de sistemas não lineares e séries temporais (ANEZAKIS *et al.*, 2016; ŠTULA; MARAS; MLADENVIĆ, 2017; POZETA; PAPAGEORGIOU; YASTREBOV, 2018);
- Medicina (AMIRKHANI *et al.*, 2017; BEVILACQUA; CIARAPICA; MAZZUTO, 2018);
- Reconhecimento de expressões faciais (ZHANG; WEI, 2017);
- Robótica e navegação autônoma (VAŠČÁK, 2010; MENDONÇA; ARRUDA; NEVES-JR, 2011; MENDONÇA *et al.*, 2013b, 2019);



- Modelagem de vida artificial (ARRUDA *et al.*, 2018);
- Economia (HATWÁGNER *et al.*, 2019);
- Engenharia e manutenção (MENDONÇA; CHUN; ROCHA, 2017);
- Controle de sistemas (MENDONÇA *et al.*, 2013a; SALMERON; PAPAGEORGIOU, 2014; AMIRKHANI *et al.*, 2016);
- Agricultura (NATARAJAN; SUBRAMANIAN; PAPAGEORGIOU, 2016; MOURHIR *et al.*, 2017);
- Entre outras (PAPAGEORGIOU; SALMERON, 2013).

Neste trabalho, será adotada a representação do FCM de tupla quádrupla  $(C, W, A, f)$  para intervalos de  $K = [-1 \ 1]$  e  $L = [0 \ 1]$ , descrita a seguir (MAGO *et al.*, 2012).

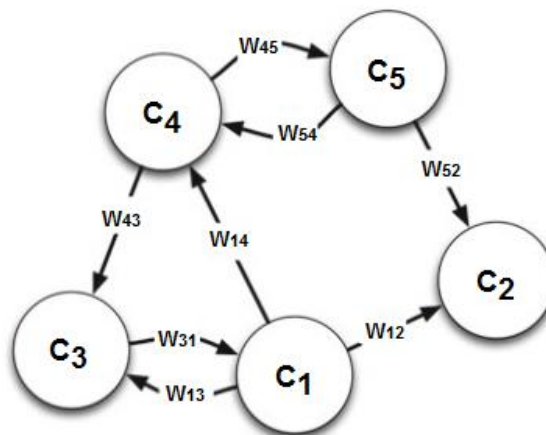
1.  $C = \{C_1, C_2, \dots, C_n\}$  é o grupo de  $n$  conceitos do FCM;
2.  $W: (C_i, C_j) \rightarrow W_{ij}$  representa uma relação causal (peso) conectando os conceitos de entrada e saída. Respectivamente,  $W_{ij} < 0$  e  $W_{ij} > 0$  indicam relações causais negativa e positiva.  $\|W_{ij}\|$  é a intensidade da relação causal. Finalmente,  $W^{(C \times C)} = W_{ij} \in K^{(n \times n)}$  é a matriz de conexão;
3.  $A: C_i \rightarrow A_i$  é o grau de ativação de um conceito, dado pela Equação (2.1).  $A(0)$  é o vetor inicial, que especifica os valores de todos os nós dos conceitos.  $A(t) \in L^n$  é o vetor de estados na iteração  $t$ ;

$$A_i(k + 1) = f \left( A_i(k) + \sum_{j \neq i}^N A_j(k) \cdot W_{ji} \right) \quad (2.1)$$

4.  $f(x): R \rightarrow L_i$  é a função de decisão, dada pela Equação (2.2). Neste trabalho foi utilizada a função sigmoide. No entanto, podem ser utilizadas outras funções conforme a natureza do problema, como apresentado em (PAPAGEORGIOU, 2014), que inclui a relação recorrente em  $t \geq 0$  entre  $A(t + 1)$  e  $A(t)$ .  $\lambda$  é um número positivo que indica a taxa de aprendizado ou a sensibilidade às mudanças de  $A$   $\lambda = 1$  neste trabalho).

$$f(x) = \frac{1}{1+e^{-\lambda x}} \quad (2.2)$$

Figura 2.10 – Exemplo de um Mapa Cognitivo *Fuzzy*



Fonte: Adaptado de (PAPAGEORGIU, 2014).

Embora não seja o escopo deste trabalho, os estudos de estabilidade e convergência dos FCMs são apresentados em (BOUTALIS; KOTTAS; CHRISTODOULOU, 2009; NÁPOLES; BELLO; VANHOOF, 2013; NÁPOLES *et al.*, 2016; ELENI; PETROS, 2017; HARMATI; KÓCZY, 2019a, 2019b). A Figura 2.10 apresenta um exemplo de FCM. Para ele, a matriz de conexão é dada pela Equação (2.3).

$$W_{ij} = \begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \\ W_{51} & W_{52} & W_{53} & W_{54} & W_{55} \end{pmatrix} \quad (2.3)$$

Uma possível metodologia para construção de um modelo FCM consiste de cinco passos (MENDONÇA, 2011):

1. Determinação de todos os conceitos. Estabelecimento das relações e sua natureza (positiva, negativa ou neutra);
2. Aquisição dos dados de entrada por conhecimento de especialistas ou por meio dos dados conhecidos do sistema analisado;
3. Realização dos cálculos das relações causais do FCM a partir dos dados de entrada;

4. Otimização do modelo FCM pelo tratamento da resposta obtida;
5. Validação do FCM nas condições de funcionamento propostas.

### 2.5.2 Revisão bibliográfica

Existem pelo menos 20 extensões de FCMs, que melhoram ou contornam suas desvantagens, como representação do conhecimento baseada em regras e modelagem de tempo (SALMERON, 2010; PAPAGEORGIOU; SALMERON, 2013). Entretanto, à medida que a utilização dos FCMs aumenta, várias limitações de sua abordagem clássica surgiram. Algumas delas são:

- a) Falta de conhecimento do sistema: a natureza dos conceitos (insumos, estados, produtos) não é levada em consideração (PAPAGEORGIOU, 2014);
- b) Dependência de especialistas: os FCMs são altamente dependentes da intervenção de especialistas para o design dos Mapas Cognitivos *Fuzzy* (PAPAGEORGIOU; FROELICH, 2012);
- c) Incapacidade de autoaprendizagem: o design das abordagens de adaptação é mais difícil devido à sua estrutura e variabilidade complexas (VAŠČÁK, 2010);
- d) Definição da causalidade: como as causalidades podem ser definidas com um alto grau de credibilidade/confiabilidade (KOULOURIOTIS; DIAKOULAKIS; EMIRIS, 2001);
- e) Equação para cálculo: como pode ser calculado um valor unificado, para descrever a alteração total causada por um dos conceitos por todos os outros (MPELOGIANNI; GROUMPOS, 2016);
- f) Ignorância do fator tempo: como pode um atraso de tempo diferente para cada relação causal ser implementada (KOSKO, 1986);
- g) Uso da função sigmoide e interpretação dos resultados: o uso da curva para ajustar os resultados ao intervalo desejado pode distorcer a interpretação final dos resultados (KARAGIANNIS; GROUMPOS, 2013).

Devido às limitações mencionadas, surge a necessidade de uma nova abordagem dos Mapas Cognitivos *Fuzzy*. Desse modo, Mapas cognitivos *Fuzzy* baseados em regras (RBF<sub>CM</sub>) (CARVALHO; TOMÉ, 2001), DF<sub>CM</sub>s (MENDONÇA *et al.*, 2013a) e Mapas Cognitivos *Fuzzy* Baseados em Autômatos Temporizados (TAF<sub>CM</sub>) (ACAMPORA; LOIA; VITIELLO, 2011) e F<sub>CM</sub>s de Estado (MPELOGIANNI; GROUMPOS, 2018b) podem ser citados como inspirações para este trabalho.

Em uma abordagem semelhante à observada nas F<sub>CM</sub>s de Estado (MPELOGIANNI; GROUMPOS, 2018b), o DF<sub>CM</sub> deste trabalho possui conceitos de entrada e saída. As entradas são os dados de três sensores ultrassônicos localizados na parte frontal dos robôs, respectivamente no lado direito, ao centro e do lado esquerdo ( $SD$ ,  $SF$  e  $SE$ ) e as saídas são os pulsos enviados aos motores CC ( $w_D$  e  $w_E$ ). No entanto, em oposição aos F<sub>CM</sub>s de Estado que contém conceitos que se alteram conforme o estado do sistema, o DF<sub>CM</sub> proposto neste trabalho possui pesos relacionados ao estado, ou seja, uma matriz de pesos diferente de acordo com cada subcomportamento do SMR.

Alguns trabalhos da literatura podem ser destacados por apresentar aplicações semelhantes a este no campo dos sistemas *Fuzzy*/F<sub>CM</sub>s aplicados na área da robótica.

O primeiro estudo desenvolveu um sistema de navegação autônomo utilizando Mapas Cognitivos *Fuzzy* Acionados por Eventos (ED-F<sub>CM</sub>). Nela, os arcos do ED-F<sub>CM</sub> são atualizados a partir da ocorrência de eventos especiais como detecção dinâmica de obstáculos. Como resultado, o modelo desenvolvido é capaz de representar o comportamento dinâmico do robô na presença de mudanças no ambiente. Essa habilidade é alcançada adaptando os relacionamentos do F<sub>CM</sub> entre os conceitos. Os resultados obtidos nas simulações indicaram que a arquitetura ED-F<sub>CM</sub> proposta constitui um sistema de navegação flexível e robusto, capaz de processar imprecisões e incertezas no ambiente. Uma das principais vantagens da abordagem proposta é que a aquisição e representação de conhecimento são simplificadas pelo uso de modelos F<sub>CM</sub> (MENDONÇA; ARRUDA; NEVES-JR, 2011).

A evolução do trabalho acima é apresentada em (MENDONÇA *et al.*, 2016), na qual dois sistemas de navegação são desenvolvidos, uma variante dos F<sub>CM</sub>s clássicos, chamada Mapas Cognitivos *Fuzzy* Híbridos Dinâmicos (HD-F<sub>CM</sub>), é utilizada para modelar tarefas de decisão e/ou fazer inferência no contexto da navegação móvel, enquanto que um controlador lógico *Fuzzy* ponderado hierárquico

(HW-FLC) compõe o segundo sistema de navegação. Nos resultados obtidos com as simulações e um experimento real utilizando a plataforma Arduino®, o agente móvel alcançou seu objetivo de atingir as metas em diferentes configurações de cenários. Em todas as simulações, o agente alcançou os alvos e evitou os obstáculos.

A utilização de FLCs na robótica também pode ser vista em (ALMEIDA *et al.*, 2019), trabalho no qual um grupo de robôs autônomos apresenta uma estratégia de navegação cooperativa e distribuída, chamada de planejador de trajetória online. Os robôs são homogêneos e navegam/exploram um ambiente desconhecido a fim de encontrar e alcançar pontos de passagem obrigatórios e, em seguida, alcançar uma posição final conhecida, interagindo entre eles e o ambiente para decidir sobre seus caminhos e tarefas. O compartilhamento de informações é realizado diretamente quando os robôs estão em uma área de comunicação e/ou indiretamente por estigmergia. Nesse caso, o feromônio artificial, como campo repulsivo, é utilizado para marcar regiões que já foram exploradas por outros membros da equipe, evitando, assim, exploração redundante e perda de tempo. O movimento dos robôs é realizado por controladores *Fuzzy*, e o desempenho planejador de trajetórias é avaliado em seis diferentes cenários simulados.

No campo de robótica aérea, mais especificamente quadrotores, pode-se citar (AMIRKHANI *et al.*, 2016). Nesse trabalho, o controle inteligente de um robô quadrotor voador é realizado por um FCM com o método de servo visual baseado em imagem (IBVS), uma vez que se supõe que as características da imagem estejam dentro do campo de visão da câmera. Os resultados da simulação obtidos mostraram que, apesar dos desafios existentes, o método proposto foi implementado com sucesso para controlar o robô quadrotor. Os ângulos de rotação e inclinação foram utilizados para a re-projeção das imagens captadas, durante a qual os pontos de imagem obtidos da câmera foram projetados em um plano de projeção sempre vertical e paralelo ao plano do objeto. Por fim, também pode-se citar os resultados da atual linha de pesquisa deste trabalho, como visto em (SOARES *et al.*, 2018) e (MENDONÇA *et al.*, 2019) já descritos anteriormente.

## 2.6 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

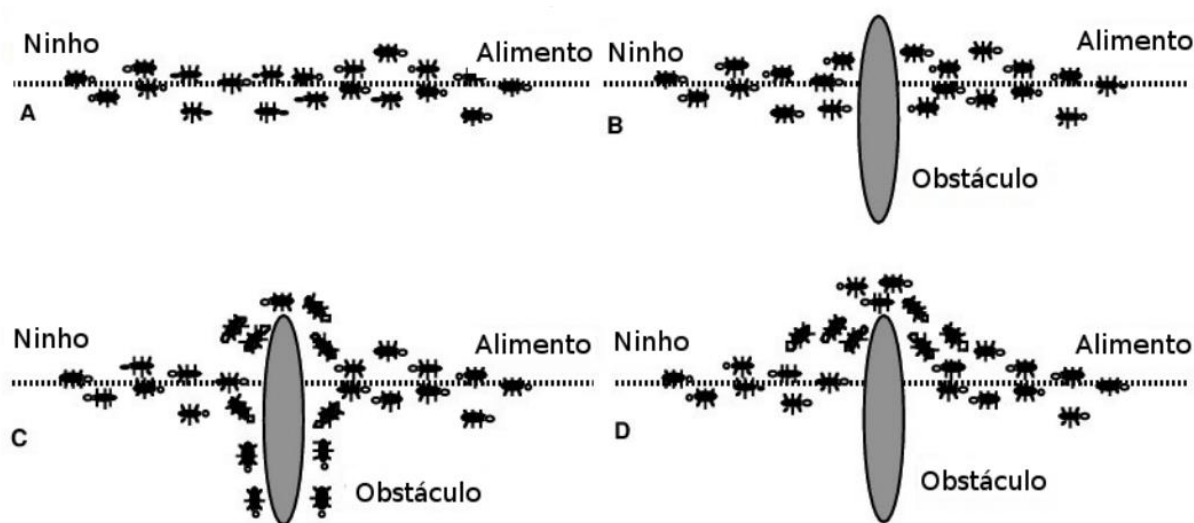
A Otimização por Colônia de Formigas (ACO) é uma metaheurística para problemas de otimização combinatória apresentada por Marco Dorigo e seus colaboradores (DORIGO; BLUM, 2005; DORIGO; MANIEZZO; COLORNI, 1996). Em resumo, uma metaheurística consiste de um conjunto de algoritmos genéricos que podem ser aplicados a diferentes classes de problemas com relativamente poucas modificações em sua estrutura (LOPES; RODRIGUES; STEINER, 2013). A ACO foi inspirada pelo comportamento de forrageamento das formigas e, em particular, como elas podem encontrar caminhos mais curtos entre as fontes de alimentos e seus ninhos. Ao procurar por comida, as formigas inicialmente exploram a área ao redor do ninho de maneira aleatória (DORIGO; BIRATTARI; STÜTZLE, 2006).

A partir de observações de seu comportamento, descobriu-se que a comunicação entre as formigas que caminhavam pelas mesmas trilhas ocorria por meio de uma substância química denominada feromônio. Enquanto se movem, elas deixam um rastro de feromônio no caminho por onde passam. Desse modo, as próximas formigas decidem qual trilha seguir pela quantidade de feromônio detectada, ou seja, escolhem o caminho mais utilizado pelo grupo (DORIGO; BLUM, 2005). Como o feromônio evapora com o tempo, quanto maior a concentração de formigas em um caminho, mais atrativo ele será para as próximas (LOPES; RODRIGUES; STEINER, 2013).

Assim que uma formiga encontra uma fonte de alimento, ela avalia a quantidade e a qualidade da comida e a leva de volta ao ninho. Durante a viagem de volta, a quantidade de feromônio que uma formiga deixa no chão pode depender da quantidade e qualidade dos alimentos. As trilhas de feromônios guiarão outras formigas para a fonte de alimento (DORIGO; BIRATTARI; STÜTZLE, 2006).

Um exemplo desse comportamento é mostrado na Figura 2.11. Como visto em (A), as formigas seguem um caminho de seu ninho até o alimento. Entretanto, com a inserção de um obstáculo (B), as formigas tendem a escolher um dos possíveis caminhos (C). Por fim, tem-se que o caminho com a maior frequência de formigas é o mais curto, como visto em (D).

Figura 2.11 – Ilustração de um experimento com formigas



Fonte: (LOPES; RODRIGUES; STEINER, 2013).

Assim, inspirado pelos conceitos de feromônio presentes na ACO, este trabalho apresenta uma otimização das trajetórias do SMR controlado por DFCM por meio de um feromônio repulsivo, chamado de DFCM-ACO. Em outras palavras, quando um robô detectar concentrações elevadas de feromônios – sejam eles deixados pelos outros robôs ou ele próprio – desvia sua rota a fim de não ficar preso em zonas de navegação relativamente mais difícil, como corredores estreitos.

Os efeitos esperados dessa estratégia são um aumento na complexidade computacional dos algoritmos, com valores de tempo de processamento entre aqueles do FLC e do DFCM. Contudo, pela otimização da trajetória dos robôs causada pela implementação do DFCM-ACO, espera-se uma maior área explorada por unidade de distância percorrida, ou seja, uma maior dispersão dos robôs dentro dos ambientes simulados propostos.

### 3 DESENVOLVIMENTO

Este capítulo é responsável por apresentar todas as etapas do desenvolvimento deste trabalho. O primeiro passo é realizar a modelagem cinemática dos robôs para sua utilização nas simulações, bem como a arquitetura de subsunção. Com isso, é necessário definir todos os parâmetros das simulações, como os ambientes, critérios de parada e as métricas de comparação entre as abordagens apresentadas. Por fim, os controladores utilizados para o SMR são descritos.

#### 3.1 MODELAGEM CINEMÁTICA DOS ROBÔS MÓVEIS

Neste trabalho, os robôs são modelados através de uma arquitetura baseada em subsunção, como visto em Mendonça *et al.* (2019), com aspectos reunidos da abordagem original proposta por Brooks (1986). Nela, um robô deve reagir apenas a estímulos externos, e não de forma abrangente. Nessa arquitetura, o comportamento global (alto nível) é decomposto em subcomportamentos de níveis de complexidade mais baixos para facilitar fatores como modelagem e escalabilidade. Essa abordagem é justificada pelo princípio de incompatibilidade de Zadeh (1976), e está presente no trabalho de veículos de Braitenberg (1986): o aumento da complexidade de um sistema é diretamente proporcional a dificuldade de prever seu comportamento.

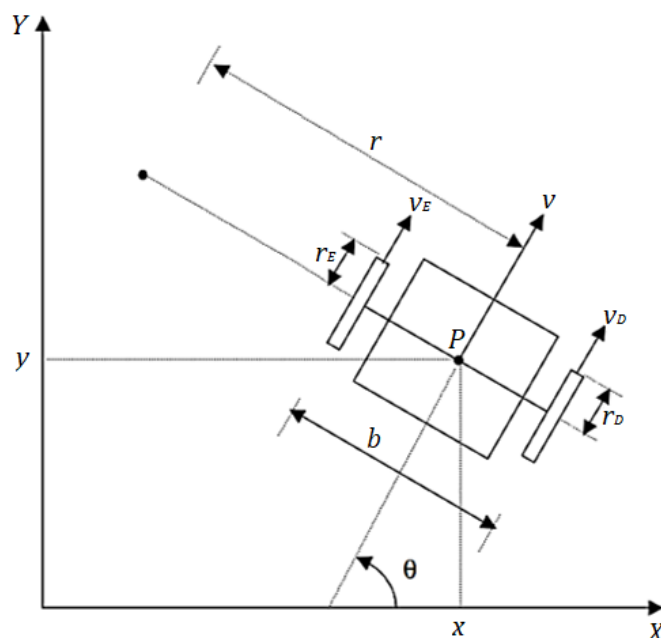
A partir da representação da Figura 3.1, a modelagem cinemática dos robôs é dada pelas equações de (3.1) a (3.10) neste trabalho. O robô simulado possui duas rodas dianteiras motorizadas e uma roda traseira de movimento livre para estabilizar as curvas. Neste trabalho, há imposição de que os robôs não se movimentem em marcha ré, tornando-os não holonômicos, ou seja, possuem ao menos uma restrição de movimento. Desse modo, para realizar a mesma trajetória da “ida” na “volta”, o robô deve girar 180°.

As variáveis  $v$  e  $w$  correspondem às velocidades linear e angular dos robôs, respectivamente.  $b$  é o comprimento do eixo (14,4 cm),  $r_D$  e  $r_E$  são o raio das rodas (2,5 cm). Os índices  $D$  e  $E$  indicam a roda direita e esquerda, respectivamente.



Este modelo cinemático também pode ser usado em robôs de quatro rodas sem rodas esterçantes. As variáveis utilizadas são mostradas como segue.

Figura 3.1 – Modelo cinemático para o robô móvel



Fonte: Adaptado de (KALIŃSKI; MAZUR, 2016).

- $P = (x, y)$ : posição dos robôs;
- $\theta$ : ângulo de orientação dos robôs;
- $b$ : comprimento do eixo rígido que liga as duas rodas;
- $r$ : raio de giro dos robôs;
- $r_D$  e  $r_E$ : raio das rodas direita e esquerda, respectivamente;
- $v$ : velocidade linear dos robôs;
- $v_D$  e  $v_E$ : velocidade linear na extremidade das rodas direita e esquerda, respectivamente;
- $w$ : velocidade angular dos robôs;
- $w_D$  e  $w_E$ : velocidade angular na extremidade das rodas direita e esquerda, respectivamente.

Para especificar a posição do robô móvel, o primeiro passo é estabelecer uma relação entre os sistemas global  $(X, Y)$  e local  $(X', Y')$  de coordenadas do plano cartesiano. Consequentemente, escolhe-se um ponto  $P$  em seu chassi – neste caso  $(x, y)$  no sistema global – no qual  $(X', Y')$  define os eixos de referência local relativos

a  $P$ . Assim, a diferença angular entre os sistemas é definida por  $\theta$ , completando a representação da pose dos robôs, definida pela Equação (3.1).

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (3.1)$$

Com isso, a próxima etapa consiste na utilização de uma matriz de rotação ortogonal  $R(\theta)$  (Equação 3.2) para realizar as transformações do sistema local (plano do robô) em relação as coordenadas do plano global.

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

A matriz na Equação (3.2) pode então ser utilizada para decompor o movimento em relação ao plano global a partir das velocidades lineares e angulares  $(\dot{x}, \dot{y}, \dot{\theta})$ , resultando na Equação (3.3) e conseqüentemente na Equação (3.4).

$$\dot{\xi}_R = R(\theta) \dot{\xi}_I \quad (3.3)$$

$$\dot{\xi}_R = R(\theta) \dot{\xi}_I = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (3.4)$$

Para a representação das velocidades lineares e angulares de cada roda, deve-se estabelecer sua relação, dada pela Equação (3.5). Com isso, utilizando deslocamentos infinitesimais, pode-se decompor a velocidade angular em suas componentes em  $x$  e  $y$ , dadas pelas Equações (3.6) e (3.7).

$$v = w \cdot r \quad (3.5)$$

$$v_D \cdot dt = w_D \cdot \left( r \pm \frac{b}{2} \right) dt \quad (3.6)$$

$$v_E \cdot dt = w_E \cdot \left( r \pm \frac{b}{2} \right) dt \quad (3.7)$$

Com isso, utiliza-se as Equações (3.6) e (3.7) para estabelecer a relação entre as componentes da velocidade linear e angular (Equações 3.8 e 3.9):

$$v_D + v_E = w_D \cdot r_D + w_E \cdot r_E = 2 \cdot w \cdot r = 2 \cdot v \quad (3.8)$$

$$v_D - v_E = w_D \cdot r_D - w_E \cdot r_E = w \cdot b \quad (3.9)$$

Por fim, isolando as velocidades linear e angular nas Equações (3.8) e (3.9), pode-se obter a matriz dada pela Equação (3.10).

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r_D}{2} & \frac{r_E}{2} \\ \frac{r_D}{b} & -\left(\frac{r_E}{b}\right) \end{bmatrix} \cdot \begin{bmatrix} w_D \\ w_E \end{bmatrix} \quad (3.10)$$

Em resumo, as saídas dos controladores geram os pulsos dos motores ( $w_D$  e  $w_E$ , 0 a 100%) das rodas direita e esquerda, de acordo com a estratégia adotada em cada subcomportamento. Se ambos os pulsos forem positivos, o robô avança. Se  $w_D > w_E$ , o robô faz uma curva para a direita, e vice-versa. Finalmente, um robô gira rapidamente se um pulso é negativo e o outro é positivo, situação que ocorre quando há iminência de colisão (MENDONÇA *et al.*, 2019).

A utilização da cinemática neste trabalho ao invés da modelagem dinâmica é justificada pela sua correspondência de resultados nessa aplicação (ROMERO *et al.*, 2014). Assim, como um dos princípios da robótica de enxame e dos SMRs é a simplicidade de implementação, optou-se pela utilização da cinemática para simplificar os experimentos simulados e aplicar e comparar as técnicas propostas.

As entradas são inspiradas nos dados reais de três sensores ultrassônicos (HC-SR04) – frontal (*SF*), esquerdo (*SE*) e direito (*SD*) – localizados na frente dos robôs. Esses sensores possuem um alcance de 12 cm e ângulos de feixe de 45°. Além disso, um ruído branco foi adicionado nas simulações para aproximar o modelo de condições reais e testar a robustez dos controladores.

### 3.2 ARQUITETURA DE SUBSUNÇÃO

Para modelar os subcomportamentos, a primeira etapa é determinar os cenários para classificá-los em subdomínios e descrever seus comportamentos esperados, conforme mostrado na Tabela 3.1.

**Tabela 3.1 – Descrição dos subdomínios do problema abordado**

Subdomínio	Descrição
Objetivos Individuais	<ul style="list-style-type: none"> <li>• Exploração do ambiente</li> <li>• Desvio de obstáculos</li> <li>• Resgate de vítimas</li> </ul>
Objetivo global	O SMR deve resgatar todas as vítimas
Subcomportamentos individuais	Ilustrados pela máquina de estados finitos da Figura 3.2 e Tabela 3.2
Comportamento global	Os robôs devem resgatar vítimas e continuar com a exploração do ambiente até que todas sejam resgatadas

**Fonte: Autoria própria.**

O trabalho proposto utiliza a mesma abordagem de trabalhos anteriores (SOARES *et al.*, 2018; MENDONÇA *et al.*, 2019) para a modelagem do comportamento do SMR. O sistema possui três tarefas, apresentadas em ordem de prioridade:

- 1) Detectar e evitar obstáculos estáticos e dinâmicos (outros robôs);
- 2) Localizar e resgatar todas as vítimas;
- 3) Mapear os ambientes.

Para concluí-las, o comportamento global foi dividido em quatro subcomportamentos, desenvolvidos de acordo com as necessidades emergentes do sistema:

- 1) Movimento livre;
- 2) Desvio de obstáculos;
- 3) Situação de iminência de colisão;
- 4) Captura das vítimas.

O subcomportamento 1 – movimento livre – é ativado quando não há detecção dos sensores dentro de sua faixa de operação. A função dos controladores é gerar um pulso inicial a 50% da capacidade dos motores, que depois é incrementado até o valor máximo de acordo com o número de iterações ou outro subcomportamento seja ativado.

De forma a evitar que os robôs fiquem presos em certas situações, principalmente corredores estreitos, esse subcomportamento é alterado caso haja a ativação do terceiro subcomportamento por 70 iterações. Caso isso ocorra, o robô virará para um dos lados.

O segundo subcomportamento – desvio de obstáculos – é ativado se o robô detectar um obstáculo dentro da faixa entre a região de segurança (4 cm) e a detecção máxima dos sensores (12 cm). Com esse modo, o controlador é responsável por atribuir curvas mais suaves aos robôs.

O terceiro subcomportamento aborda a detecção de uma iminência de colisão: é ativado quando os três sensores ultrassônicos frontais detectam um obstáculo mais próximo que a distância de segurança. A saída do controlador força um giro rápido para desviar do obstáculo, de acordo com os dados dos sensores. Para impedir que o robô fique preso no mesmo laço, esse algoritmo altera a direção da curva sempre que o robô entra no subcomportamento 1.

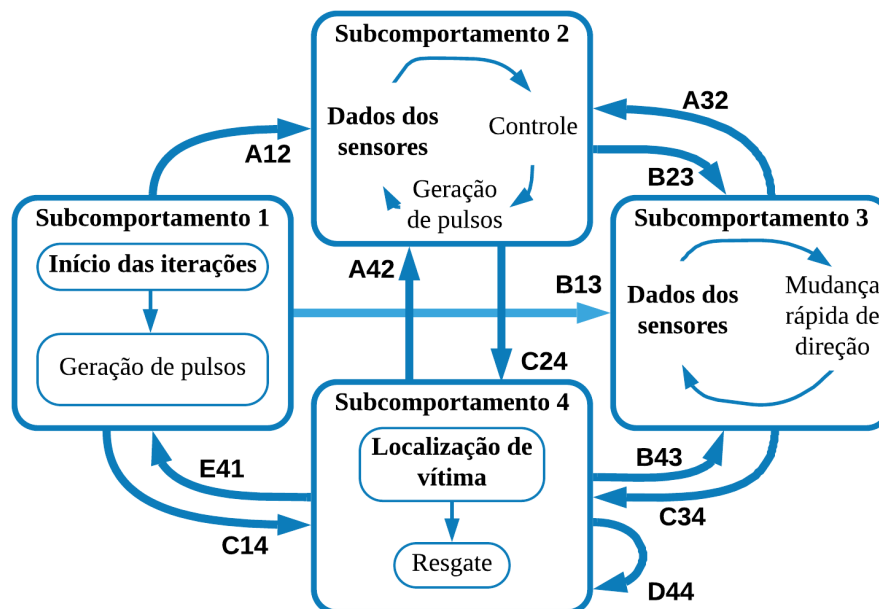
O subcomportamento 4 é responsável pelo resgate das vítimas. Quando uma delas é localizada sua cor é alterada para verde nas simulações. Com isso, o robô responsável se aproxima para realizar a operação. Nesse subcomportamento, por medida de segurança, o controlador diminui a velocidade do robô ao se aproximar da vítima ou se houver um obstáculo dentro da zona de segurança.

Dessa forma, possíveis novos recursos, como o planejamento de trajetória, podem desclassificar os robôs como puramente reativos. Com isso, a navegação dos robôs seria híbrida, ou seja, a partir de um dado planejamento de trajetória, o DFCM utilizado neste trabalho atuaria para desvio de obstáculos desconhecidos previamente pelo sistema.

Além disso, os robôs apresentam um algoritmo de memória utilizado para mapeamento do ambiente e em uma situação particular. Se um robô encontra uma vítima, mas não pode resgatá-la após um certo período de tempo, os robôs próximos recebem sua localização para resgatá-la. Isso ocorre porque, imediatamente após

localizar a vítima, seu resgate se torna o principal objetivo do robô. Assim, se um obstáculo for encontrado, o robô abandonará seu alvo (vítima), permitindo que um robô em melhores condições a resgate.

Figura 3.2 – Máquina de estados finitos para o comportamento dos robôs



Fonte: Autoria própria.

Tabela 3.2 – Descrição dos eventos do SMR

Evento	Descrição
A	Obstáculo detectado por qualquer sensor ultrassônico
B	Todos os sensores detectaram um obstáculo a uma distância inferior à de segurança
C	O robô localizou uma vítima
D	Vítima e robô estão em processo de resgate
E	Ativação do algoritmo de memória do robô

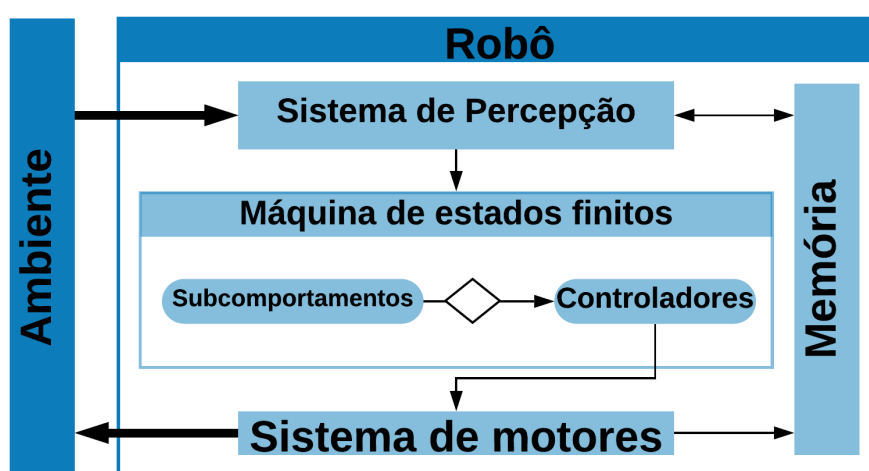
Fonte: Autoria própria.

Os subcomportamentos operam em paralelo, cada um implementado individualmente de acordo com a máquina de estados finitos mostrada na Figura 3.2, e seus eventos descritos na Tabela 3.2. Como visto na figura, os dados atuais dos sensores inibem ou ativam as rotinas dos subcomportamentos, gerando as saídas desejadas de acordo com os modelos dos controladores. As inscrições nas setas da Figura 3.2 indicam a letra do evento que desencadeou o comportamento listado no alfabeto presente na Tabela 3.2. Dessa forma, o primeiro número indica o

subcomportamento antes da ocorrência do evento, enquanto que o segundo indica o subcomportamento atual gerado pela ocorrência.

Uma visão geral da arquitetura do sistema pode ser vista na Figura 3.3 e pode ser utilizada para auxiliar na construção de modelos para sistemas de navegação autônomos. Os sinais recebidos pelos sensores entram em um processo de tomada de decisão que modifica os subcomportamentos dentro da máquina de estados finitos de acordo com a ocorrência dos eventos da Tabela 3.2.

Figura 3.3 – Visão geral da arquitetura dos robôs



Fonte: Autoria própria.

Neste trabalho, todo subcomportamento aciona uma matriz de pesos diferente no caso do DFCM, conforme mostrado na Seção 3.5. Na próxima etapa, as saídas dos controladores são enviadas para o sistema de motores. Os dados da pose e as localizações visitadas são armazenadas na memória dos robôs para o mapeamento.

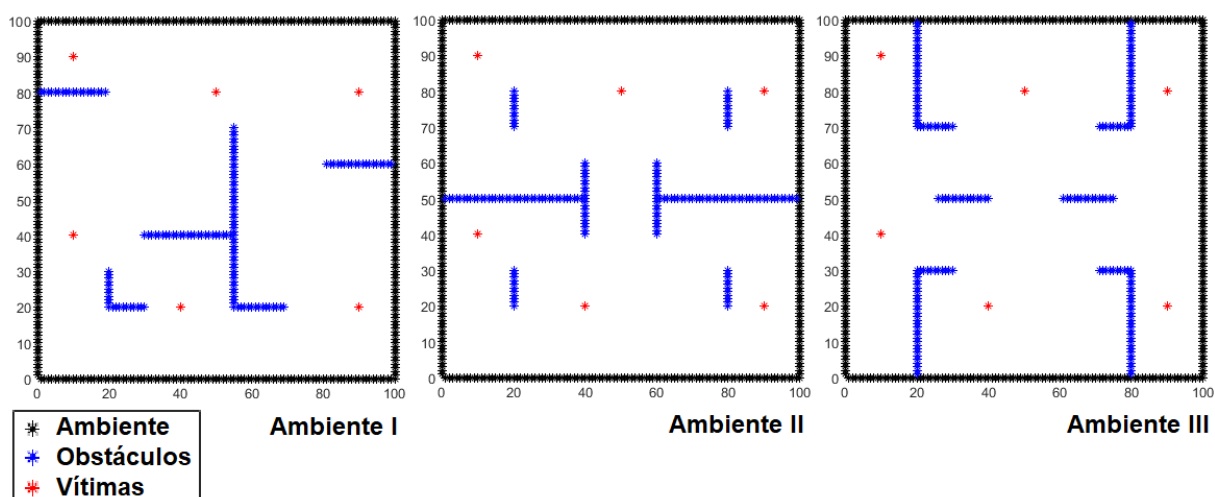
### 3.3 CONSIDERAÇÕES DOS EXPERIMENTOS SIMULADOS

Na literatura, umas das definições possíveis de autonomia é o cumprimento dos objetivos propostos de forma eficiente em pelo menos três ambientes simulados (MAES, 1995; MATARIC, 2007; RUSSELL; NORVIG, 2009). Os controladores FLC e DFCM foram comparados nas mesmas condições ambientais: as simulações

começaram com um robô, seguidas por quatro e depois expandidas para oito. As simulações pressupõem que todos os sensores ultrassônicos tenham uma distância máxima de percepção de 12 cm e uma distância de segurança de 4 cm (que dispara o subcomportamento 3).

Neste trabalho, os experimentos simulados são realizados no Matlab® nos mesmos ambientes dos experimentos simulados vistos em (MENDONÇA *et al.*, 2019), delimitados em um quadrado de 10000 cm<sup>2</sup> de área. No entanto, essas dimensões podem ser readequadas para mudanças em cenários reais de busca de vítimas. Em todos os ambientes, as vítimas estão nas coordenadas (10 40), (10 90), (40 20), (50 80), (90 20) e (90 80) (MENDONÇA *et al.*, 2019). Contudo, os robôs não possuem conhecimento da localização das vítimas. Assim, o resgate de todas as vítimas é utilizado como um dos critérios de parada das simulações.

**Figura 3.4 – Ambientes simulados**



Fonte: Autoria Própria.

Os ambientes podem ser vistos na Figura 3.4, na qual as vítimas estão marcadas como asteriscos vermelhos e os obstáculos internos em azul. As extremidades de cada ambiente são demarcadas por paredes (em preto) que impedem que o SMR ultrapasse os limites estabelecidos. Para os ambientes, objetivou-se que apresentassem diferentes níveis de dificuldade, a fim de testar os controladores propostos em diferentes condições. Assim, os experimentos simulados foram realizados em um computador com as seguintes especificações: processador de 6 núcleos e 12 *threads*, frequências entre 3.4 – 3.9GHz; memória RAM 8GB com frequência de 2666Hz em *single channel*.



O autor tem conhecimento do número de vítimas (seis) como um dos critérios de parada (apenas a nível dos experimentos simulados). No entanto, outros critérios poderiam ser adotados de forma a manter essa classificação: os robôs poderiam parar de operar quando explorassem uma determinada porcentagem da área (conhecida), alcançassem um número máximo de iterações, ou ainda esgotassem suas baterias simuladas. O outro critério de parada definido é o número máximo de 5000 iterações. Quando um dos robôs detecta um dos obstáculos internos, a sua representação passa a ser em preto no experimento simulado atual. Ademais, para as simulações com mais de um robô, considerou-se o restante da equipe como obstáculos dinâmicos.

Os parâmetros utilizados para comparar o esforço computacional de ambos os controladores são o número de iterações necessárias para resgatar todas as vítimas e o tempo de processamento do Matlab® (através das funções incorporadas *tic* e *toc*). Além disso, outros aspectos foram comparados, como a suavidade dos pulsos enviados às rodas ( $w_D$  e  $w_E$ ). A eficácia e a eficiência de ambos os controladores foram comparadas por meio da área explorada e da distância percorrida de cada robô e pelo SMR. Por fim, foram gerados vídeos de todas as simulações por meio das funções incorporadas *getframe* e *VideoWrite*. Abaixo, tem-se uma possível interpretação para os parâmetros de comparação utilizados neste trabalho:

- Número de iterações: fragmento constituinte da análise da eficiência das técnicas de controle utilizadas e dos algoritmos. Essa variável é diretamente proporcional ao tempo de processamento, e pode indicar qual dos controladores envia os maiores pulsos para os motores;
- Distância percorrida: similarmente ao parâmetro anterior, é inversamente proporcional ao consumo de bateria dos robôs (em aplicações reais). Em outras palavras, diz respeito a eficiência na utilização de um recurso esgotável;
- Área explorada: foi dividida entre individual e total. Com a primeira, pode-se tomar conclusões sobre a trajetória de cada robô – se percorreu o mesmo caminho várias vezes – e, pela segunda, tem-se uma forma de medir a dispersão do SMR, ou seja, uma métrica para avaliar qual dos controladores obteve a maior taxa de área mapeada. Nesse parâmetro, leva-se em consideração a superposição das áreas cobertas por robô.

As poses iniciais dos robôs consistem em suas posições ( $x$   $y$ ) e ângulo em relação às coordenadas globais. Esses parâmetros – juntamente com as cores dos robôs nas simulações – são mostrados nas Tabelas 3.3 e 3.4 de acordo com a configuração do SMR. Nos cenários com apenas um robô a pose é (40, 10, 0°).

**Tabela 3.3 – Poses para os cenários com 4 robôs**

Robô	X (cm)	Y (cm)	Ângulo (°)	Cor
1	15	5	90	Azul
2	30	5	90	Verde
3	45	5	90	Vermelho
4	65	5	90	Ciano

Fonte: Autorial Própria.

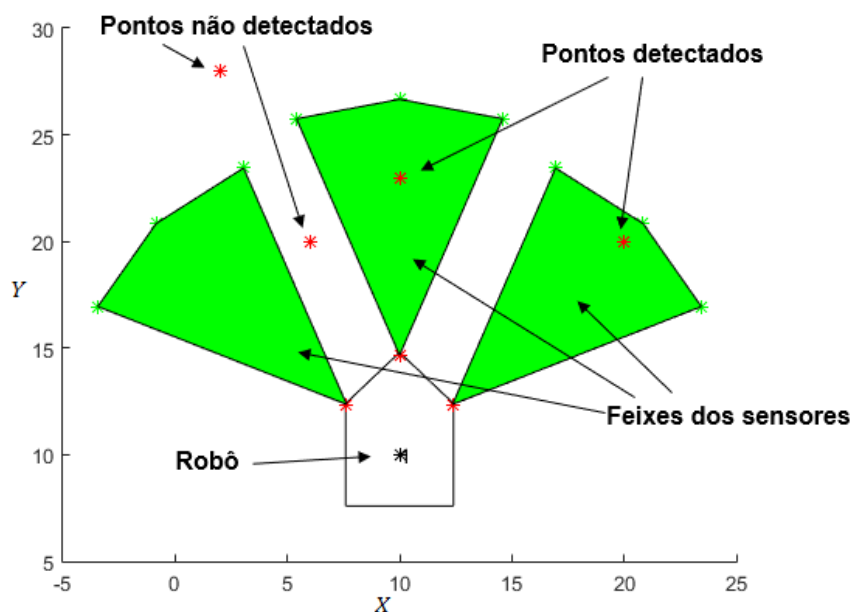
**Tabela 3.4 – Poses para os cenários com 8 robôs**

Robô	X (cm)	Y (cm)	Ângulo (°)	Cor
1	15	5	90	Azul
2	25	5	90	Verde
3	35	5	90	Vermelho
4	45	5	90	Ciano
5	55	5	90	Magenta
6	65	5	90	Amarelo
7	75	5	90	Preto
8	85	5	90	Azul

Fonte: Autorial Própria.

No Matlab<sup>®</sup>, os sensores foram simulados usando a distância cartesiana entre os robôs e obstáculos/vítimas. O perímetro de resgate foi escolhido com base na distância entre o centro de massa dos robôs e as vítimas, considerando um erro máximo de 3 cm. Como abordado anteriormente, as variáveis de entrada dos controladores são inspiradas nos dados reais de três sensores ultrassônicos HC-SR04 localizados na frente dos robôs conforme observado na Figura 3.5. Seu alcance é de 12 cm e possui ângulos de feixe de 45°.

**Figura 3.5 – Representação das características dos robôs simulados**



**Fonte: Autoria Própria.**

A fim de aproximar o modelo de possíveis condições reais, adicionou-se um ruído branco nas simulações. Além disso, é importante ressaltar que, a nível de simulação, uma vítima é automaticamente detectada quando ela entra no raio de alcance de um dos sensores ultrassônicos, o que não seria possível em uma aplicação real sem a utilização de outro sensor, como uma câmera.

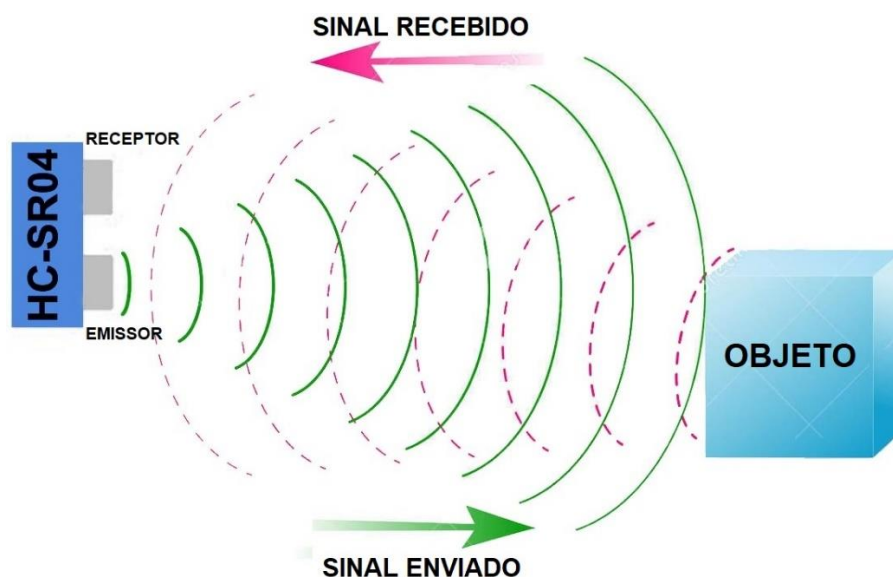
**Figura 3.6 – Sensor Ultrassônico HC-SR04**



**Fonte: (THOMSEN, 2011).**

De forma geral, os sensores ultrassônicos tem dois componentes principais: emissor e receptor. Seu princípio de funcionamento consiste na emissão de um sinal sonoro de alta frequência que, ao atingir algum objeto, retorna um sinal de eco, captado pelo receptor do HC-SR04. Um exemplo desse sensor pode ser visto na Figura 3.6, com funcionamento ilustrado na Figura 3.7.

Figura 3.7 – Princípio de funcionamento HC-SR04



Fonte: Adaptado de (THOMSEN, 2011).

Para o experimento real, conduzido com base nos ambientes simulados, foi utilizado um módulo de GPS GY-NEO6MV2, mostrado na Figura 3.8, para enviar a localização, data e hora dos alvos localizados. O sistema de posicionamento global (GPS, do inglês *global positioning system*), é um sistema de navegação por satélite que fornece as características de posição e horário a um receptor, desde que este se encontre na área de trabalho de três ou mais satélites GPS.

Figura 3.8 – Módulo GPS GY-NEO6MV2



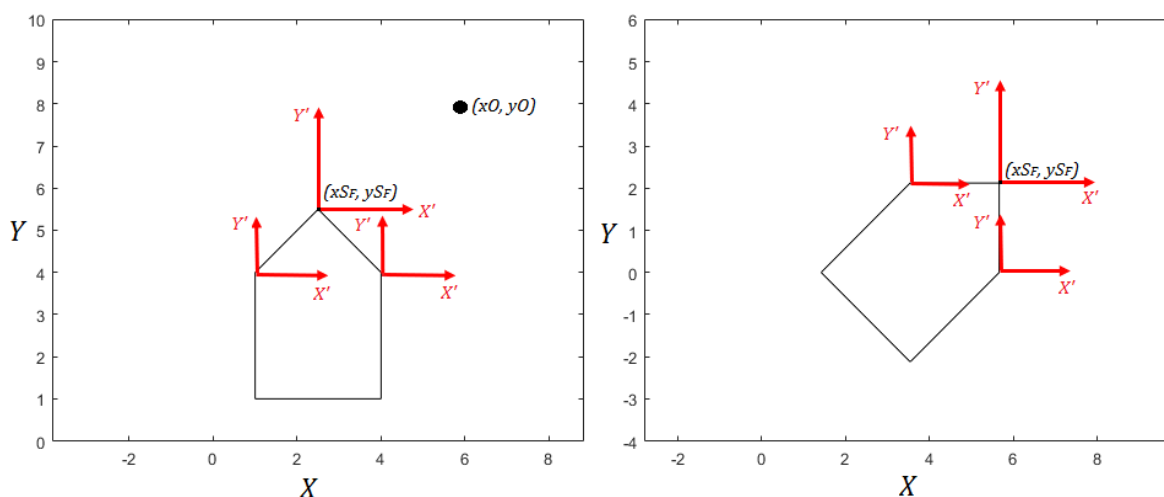
Fonte: (NEWEGG, 2020).

Especificamente, o GY-NEO6MV2 foi escolhido por ser um módulo de fácil utilização e obtenção, além de possui comunicação serial com o microcontrolador escolhido, possuindo uma precisão de aproximadamente 2,5 m. Para sua utilização, deve-se inserir três bibliotecas no Arduino® MEGA 2560: *SoftwareSerial* (para o envio

de informações ao Arduino<sup>®</sup>), *TinyGPS* (configurações do GPS) e *TimeLib* (ajuste do fuso horário).

Nas simulações, um algoritmo é responsável por realizar a operação dos sensores a cada iteração. A partir de sistema global  $(X, Y)$ , distância e angulação em relação aos obstáculos são calculadas para os três sensores de acordo com um sistema de coordenadas locais  $(X', Y')$ , para cada sensor, sempre paralelos ao plano global. Assim, a posição relativa a  $(X', Y')$  é dada pela distância euclidiana ( $dE$ ) conforme (3.11) a partir da posição global de cada sensor  $i = \{E, F, D\}$ , dada por  $(xS_i, yS_i)$ . Em (3.11) a (3.14),  $(xO, yO)$  representam a posição de cada obstáculo. A Figura 3.9 ilustra o princípio utilizado para os cálculos apresentados utilizando o sensor frontal como exemplo.

**Figura 3.9 – Representação gráfica do sistema de coordenadas locais**



Fonte: Autoria Própria

$$dE = \sqrt{(xO - xS_i)^2 + (yO - yS_i)^2} \quad (3.11)$$

A partir de (3.11), também pode-se obter a informação do ângulo formado entre objeto e o sistema local para cada sensor. O processo de obtenção do ângulo, dado por  $\alpha$ , é descrito pelas Equações (3.12), (3.13) e (3.14).

$$\text{sen}(\alpha) = \frac{(yO - yS_i)}{dE} \quad (3.12)$$

$$\text{cos}(\alpha) = \frac{(xO - xS_i)}{dE} \quad (3.13)$$

$$\alpha = \operatorname{tg}^{-1} \left[ \frac{(yO - yS_i)}{(xO - xS_i)} \right] \quad (3.14)$$

A partir da utilização de (3.12), (3.13) e (3.14), determina-se então qual o quadrante no qual o objeto está localizado em relação ao robô. Para isso, os seguintes fatores são considerados:

- $\operatorname{sen}(\alpha) > 0$  e  $\operatorname{cos}(\alpha) > 0$ : o obstáculo ou vítima encontra-se no primeiro quadrante. Nesse caso, o valor dado pela tangente inversa (3.14) não necessita de alteração;
- $\operatorname{sen}(\alpha) > 0$  e  $\operatorname{cos}(\alpha) < 0$ : o obstáculo ou vítima encontra-se no segundo quadrante. Assim, adiciona-se  $180^\circ$  ao valor obtido em (3.14);
- $\operatorname{sen}(\alpha) < 0$  e  $\operatorname{cos}(\alpha) < 0$ : o obstáculo ou vítima encontra-se no terceiro quadrante. Assim como no caso anterior, adiciona-se  $180^\circ$  ao valor de  $\alpha$ ;
- $\operatorname{sen}(\alpha) < 0$  e  $\operatorname{cos}(\alpha) > 0$ : o obstáculo ou vítima encontra-se no quarto quadrante. Nesse caso, para obter o valor correto de  $\alpha$ , adiciona-se  $360^\circ$  ao valor obtido em (3.14).

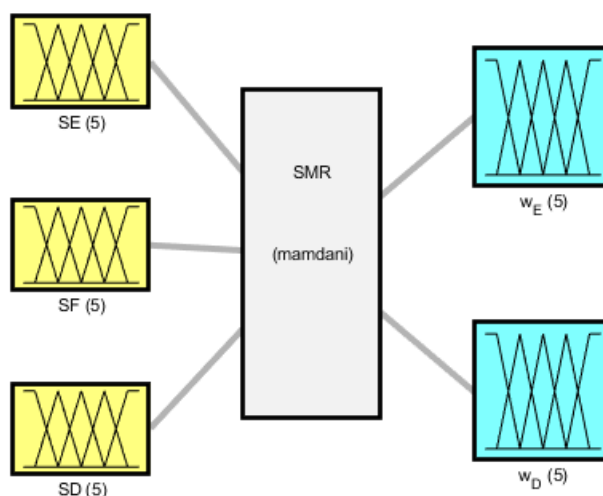
Assim, todos os obstáculos dentro da região de captura dos sensores com  $\alpha$  na faixa do ângulo de mapeamento serão mapeados e, no caso das vítimas, serão marcadas para a operação de resgate. Caso contrário, os obstáculos e vítimas serão considerados ocultos para os robôs. Para não haver detecção de vítimas além dos obstáculos (“robô enxergando um alvo através das paredes”), quando ocorre uma detecção de obstáculo, o robô verifica se a posição da vítima está além da parede fixa. Se a resposta for afirmativa, o alcance do sensor analisado passa a ser a distância entre ele e a parede detectada, e o alvo é descartado.

Por fim, no caso do algoritmo da ACO, o princípio de detecção de feromônios é o mesmo daquele encontrado nos sensores simulados. Contudo, para evitar poluição visual nos resultados e não gerar peso computacional desnecessário, tem-se que os robôs despejam uma “unidade” de feromônio a cada 10 iterações nos experimentos simulados.

### 3.4 SISTEMA DE CONTROLE FUZZY

O sistema de controle *Fuzzy* desenvolvido para este trabalho é descrito como segue. Possui três entradas (sensores ultrassônicos), duas saídas (pulsos enviados aos motores CC) e é do tipo Mamdani, como observa-se na Figura 3.10.

**Figura 3.10 – Estrutura do FLC utilizado**



Fonte: Autoria própria.

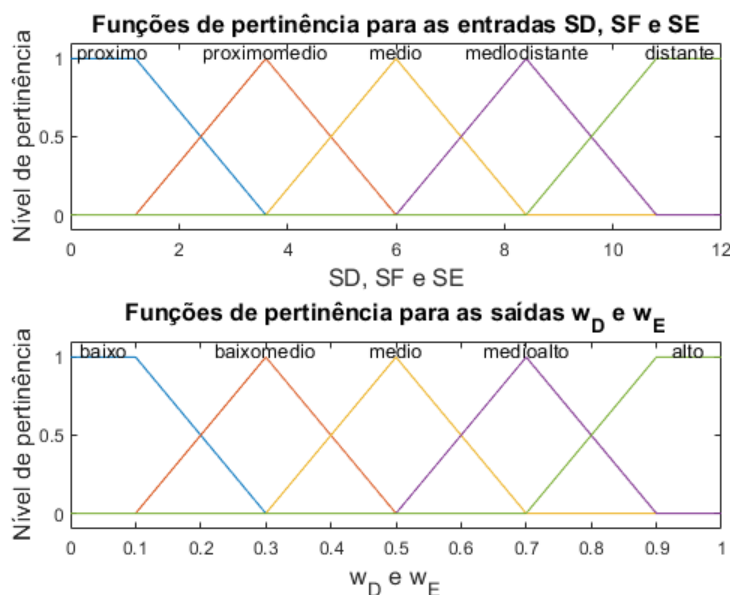
As entradas são os dados obtidos dos sensores ultrassônicos simulados: esquerdo ( $SE$ ), frontal ( $SF$ ) e direito ( $SD$ ). As saídas são representadas pela intensidade dos pulsos enviados às rodas direita e esquerda dos robôs, respectivamente  $w_D$  e  $w_E$ . Os universos de discurso desses parâmetros, mostrados abaixo, respeitam os dados dos sensores reais e a porcentagem de utilização dos motores:

- Detecção dos sensores ultrassônicos: a distância de detecção dos três sensores varia na faixa [0 12] cm, conforme datasheet do HC-SR04;
- Envio de pulsos aos motores: seu universo de discurso representa a taxa de utilização de cada motor, ou seja, de 0 a 100%. Nesse caso, a saída foi normalizada na faixa [0 1].

Os universos de discurso das variáveis do FLC são divididos em cinco funções de pertinência: duas trapezoidais nas bordas e três triangulares ao centro,

como apresentado pela Figura 3.11. As variáveis linguísticas das entradas descrevem a distância entre o sensor e o objeto detectado: “próximo”, “médio”, “distante” e suas combinações possíveis “proximomédio” e “médiodistante”.

**Figura 3.11 – Funções de pertinência utilizadas**



**Fonte: Autoria própria.**

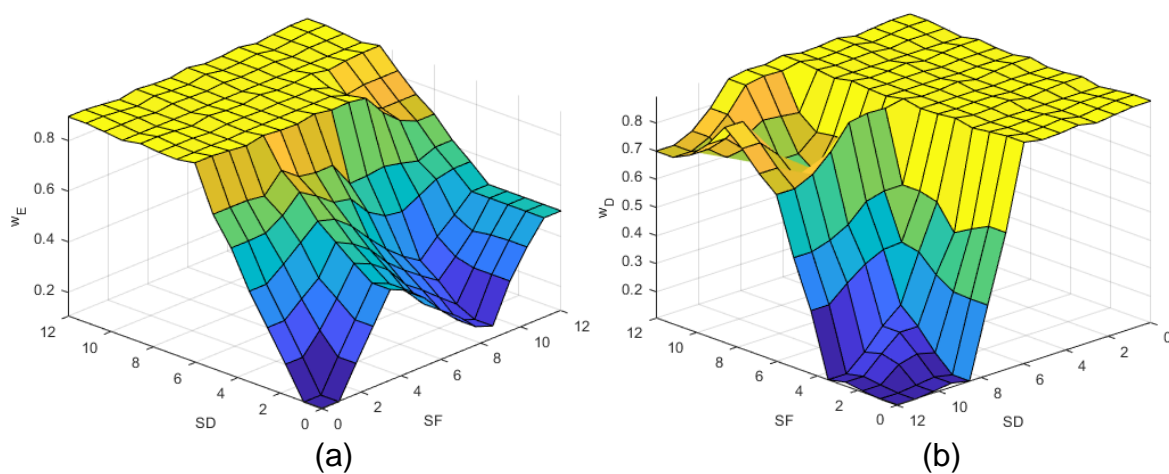
Como cada uma das entradas possui 5 funções de pertinência, tem-se que o FLC possui 125 regras a partir da combinação entre essas funções. Todas as regras possuem a mesma ponderação (1) e, como exemplo, 3 delas são mostradas a seguir. O restante das regras pode ser visto no Apêndice A.

1. Se ( $SE$  é próximo) e ( $SF$  é próximo) e ( $SD$  é próximo) então ( $w_E$  é baixo) ( $w_D$  é alto);
2. Se ( $SE$  é próximo) e ( $SF$  é próximo) e ( $SD$  é proximomédio) então ( $w_E$  é alto) ( $w_D$  é médio);
3. Se ( $SE$  é próximo) e ( $SF$  é próximo) e ( $SD$  é médio) então ( $w_E$  é alto) ( $w_D$  é baixomédio).

As relações entre as entradas e saídas podem ser observadas também pelas superfícies *Fuzzy*. Como são três entradas e duas saídas, são seis superfícies possíveis. A Figura 3.12 descreve a relação entre as entradas  $SD$  e  $SF$  com as saídas  $w_E$  (a) e  $w_D$  (b).



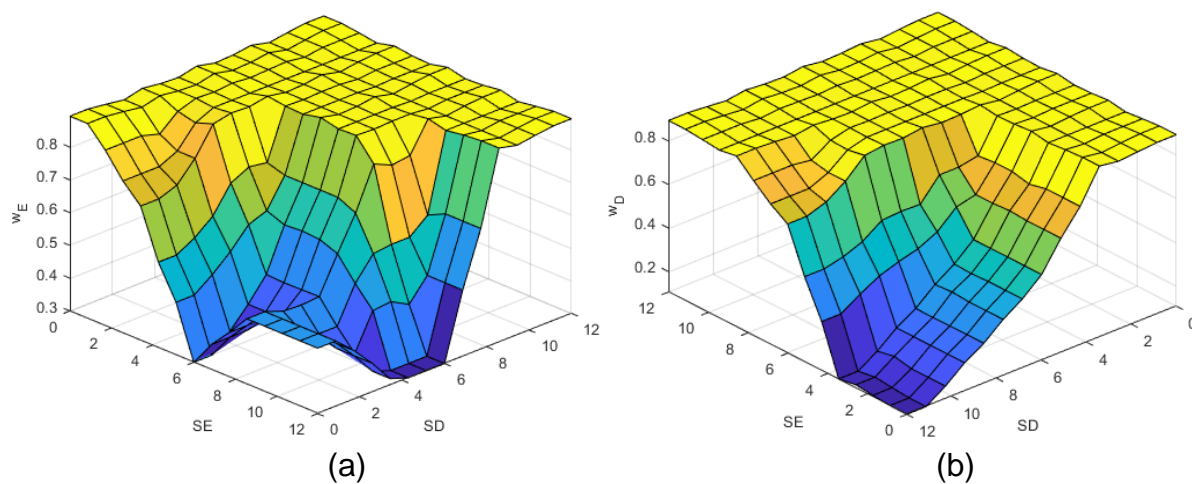
Figura 3.12 – Superfícies *Fuzzy*: SD e SF com (a)  $w_E$  e (b)  $w_D$



Fonte: Autoria própria.

Na Figura 3.13 pode-se observar a relação entre as entradas  $SD$  e  $SE$  com as saídas  $w_E$  (a) e  $w_D$  (b), rotacionada para melhor visualização.

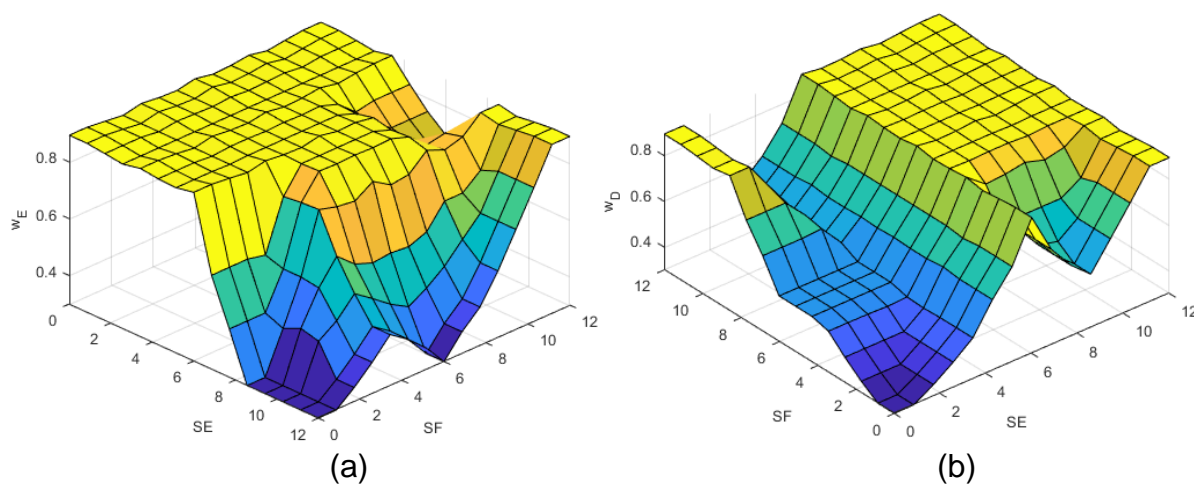
Figura 3.13 – Superfícies *Fuzzy*: SE e SD com (a)  $w_E$  e (b)  $w_D$



Fonte: Autoria própria.

Por fim, a Figura 3.14 descreve as relações entre as entradas  $SE$  e  $SF$  com a saída  $w_E$  (a) e  $SF$  e  $SE$  com  $w_D$  (b).

Figura 3.14 – Superfície *Fuzzy*: SE e SF com (a)  $w_E$  e (b)  $w_D$

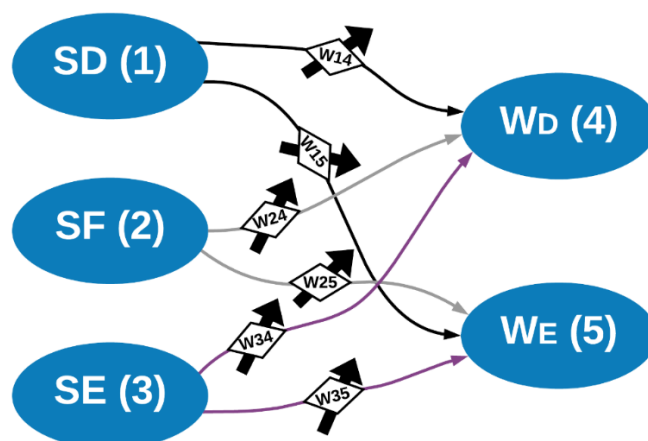


Fonte: Autoria própria.

### 3.5 SISTEMAS DE CONTROLE POR MAPAS COGNITIVOS *FUZZY* DINÂMICOS

O DFCM desenvolvido para este trabalho modifica/atualiza sua matriz de pesos de acordo com os dados dos sensores e o subcomportamento atual dos robôs. Em outras palavras, o DFCM é dirigido pelos eventos que descrevem o estado atual dos sensores, de forma semelhante a (MENDONÇA; ARRUDA; NEVES-JR, 2011).

Figura 3.15 – Mapa Cognitivo *Fuzzy* Dinâmico proposto



Fonte: Autoria própria.

A estrutura do DFCM, mostrada na Figura 3.15, foi inspirada no sistema *Fuzzy* e segue a mesma lógica de representação das variáveis: as entradas são os

dados dos sensores ultrassônicos: esquerdo (*SE*), frontal (*SF*) e direito (*SD*), e as saídas são as intensidades dos pulsos enviados às rodas esquerda ( $w_E$ ) e direita ( $w_D$ ) dos robôs. Na Figura 3.15, os losangos correspondem ao processo de tomada de decisões DFCM, que é responsável por alterar a matriz de pesos de acordo com o subcomportamento atual. Particularmente, se um obstáculo estiver dentro da zona de segurança, todos os pesos  $W_{ij}$  são modificados para permitir que os robôs desacelerem e façam curvas mais acentuadas (MENDONÇA *et al.*, 2019).

Os níveis de causalidades de cada matriz foram ajustados empiricamente de acordo com a experiência obtida pela operação do SMR e guiados pelos comportamentos desejados para o sistema. Assim, de acordo com a Figura 3.15, essas relações são mostradas em sua forma matricial na Equação (3.15), com seus valores apresentados na Tabela 3.5. Nela, os pesos não indicados são nulos.

$$W_{ij} = \begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \\ W_{51} & W_{52} & W_{53} & W_{54} & W_{55} \end{pmatrix} \quad (3.15)$$

**Tabela 3.5 – Valores dos pesos para o DFCM**

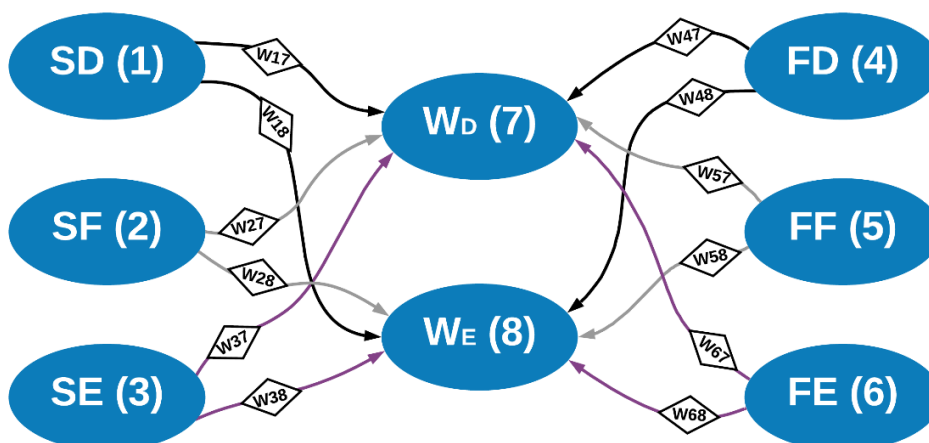
Peso	Subcomportamento			
	1	2	3	4
$W_{14}$	-0.40	0.60	0.80	-0.40
$W_{15}$	0.50	-0.20	-0.60	0.50
$W_{24}$	0.30	-0.30	-0.60	0.30
$W_{25}$	0.30	-0.30	-0.60	0.30
$W_{34}$	0.50	-0.20	0.60	0.50
$W_{35}$	-0.40	0.60	-0.70	-0.40

**Fonte: Autoria Própria.**

De forma a adaptar de forma otimizada os caminhos percorridos pelos robôs que utilizam a estratégia DFCM, optou-se por utilizar o conceito de feromônio repulsivo oriundo da ACO. Para isso, foi desenvolvido um DFCM-ACO (Figura 3.16) que leva em consideração os caminhos percorridos por todos os robôs do sistema. Assim, caso algum dos robôs detecte uma quantidade significativa de feromônio numa área específica dos ambientes simulados, há a capacidade de alterar sua rota atual

para explorar novas áreas. Esse comportamento estimula a exploração de regiões maiores e, de certa forma, otimiza a bateria dos robôs, já que os desestimula a percorrer sempre o mesmo caminho.

Figura 3.16 – DFCM-ACO proposto



Fonte: Autoria própria.

Tabela 3.6 – Valores dos pesos para o DFCM-ACO

SC	Peso											
	$W_{17}$	$W_{18}$	$W_{27}$	$W_{28}$	$W_{37}$	$W_{38}$	$W_{47}$	$W_{48}$	$W_{57}$	$W_{58}$	$W_{67}$	$W_{68}$
1	0.60	-0.40	-0.30	-0.30	-0.20	0.20	0.80	-0.20	-0.50	-0.50	-0.20	0.20
2	0.60	-0.40	-0.30	-0.30	-0.20	0.20	0.80	-0.20	-0.50	-0.50	-0.20	0.20
3	0.80	-0.60	-0.80	-0.80	0.60	-0.40	0.20	-0.20	-0.20	-0.20	-0.20	0.20
4	-0.40	0.50	0.30	0.30	0.50	-0.40	0.00	0.00	0.00	0.00	0.00	0.00

Fonte: Autoria Própria.

O funcionamento dos conceitos relativos aos feromônios detectados à direita (*FD*), ao centro (*FF*) e à esquerda (*FE*) segue o mesmo princípio dos sensores ultrassônicos: o robô, ao detectar feromônio em uma das três posições, deve desviar para o sentido com menor concentração do mesmo, seja ele deixado por outros robôs ou por ele mesmo. Assim, os pesos utilizados nessa abordagem são mostrados na Tabela 3.6, na qual a primeira coluna refere-se aos subcomportamentos (SC).

## 4 RESULTADOS E DISCUSSÃO

O objetivo deste capítulo é de apresentar os resultados provenientes dos três controladores apresentados e realizar as comparações propostas nas seções anteriores utilizando o hardware apresentado na Seção 3.3. Os resultados estão separados pelo número de robôs de cada cenário (1, 4 e 8). Por fim, os valores obtidos são comparados e analisados.

### 4.1 RESULTADOS PARA UM ROBÔ

O primeiro passo foi implementar um robô controlado por FLC e DFCM em todos os ambientes para servir de base de comparação com as simulações com 4 e 8 robôs. Nesse cenário, espera-se que os robôs não concluam a tarefa (resgate das seis vítimas) ou a concluam levando mais tempo do que o desejado, considerando que uma tarefa de resgate deve ser concluída no menor tempo possível que, a tempo dos experimentos simulados, corresponde ao número de iterações.

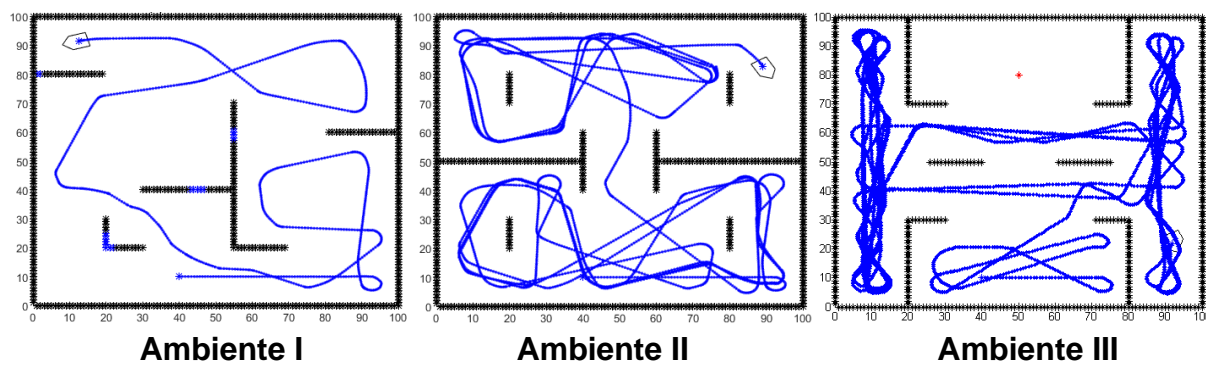
Os trajetos do robô em cada ambiente podem ser vistos nas Figuras 4.1, 4.2 e 4.3 respectivamente para FLC, DFCM e DFCM-ACO. No primeiro, o robô enfrentou algumas dificuldades no quadrante superior esquerdo, um espaço confinado. Esse fato levou o robô a não explorar a área central esquerda desse ambiente, pois o experimento foi encerrado quando a última vítima foi encontrada.

Pelo terceiro ambiente da Figura 4.1, tem-se que o robô controlado pelo FLC não foi capaz de cumprir os objetivos no terceiro ambiente. Após 5000 iterações, os movimentos ficaram concentrados nos corredores esquerdo e direito desse ambiente. Desse modo o experimento simulado foi encerrado sem o resgate de uma das vítimas. Esse fato ilustra o comportamento esperado para os cenários com apenas um robô, e sugere a implementação do SMR.

No segundo ambiente, a forma de corredor também levou o robô a percorrer a área inferior duas vezes, momento no qual o algoritmo de randomização atuou para alterar a orientação do robô, o que permitiu a exploração da seção superior.

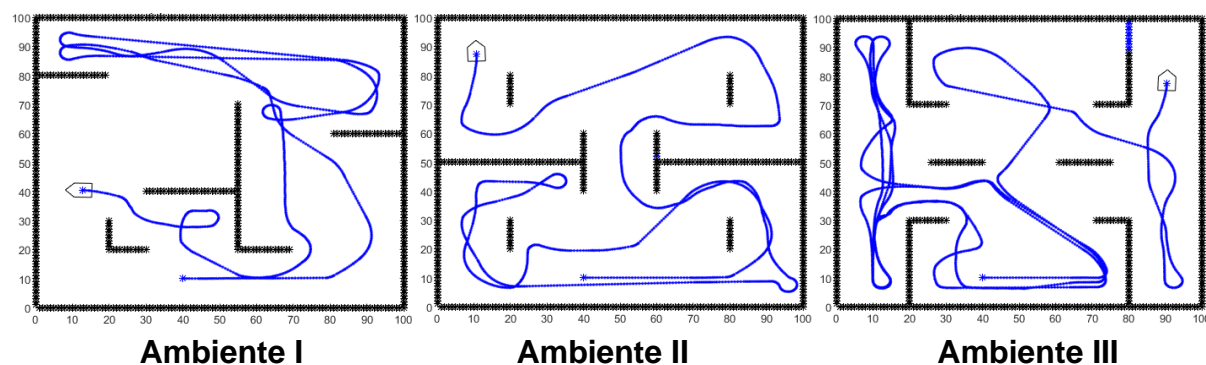
O último ambiente também apresentou uma situação semelhante, na qual o robô percorreu a seção diagonal inferior algumas vezes antes de alterar sua orientação.

**Figura 4.1 – Trajeto de um robô com FLC**



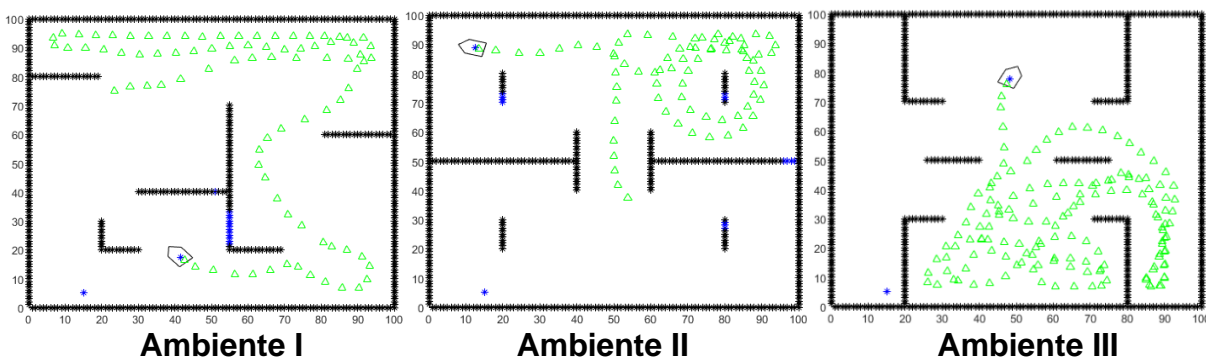
Fonte: Autoria própria.

**Figura 4.2 – Trajeto de um robô com DFCM**



Fonte: Autoria própria.

**Figura 4.3 – Trajeto de um robô com DFCM-ACO**

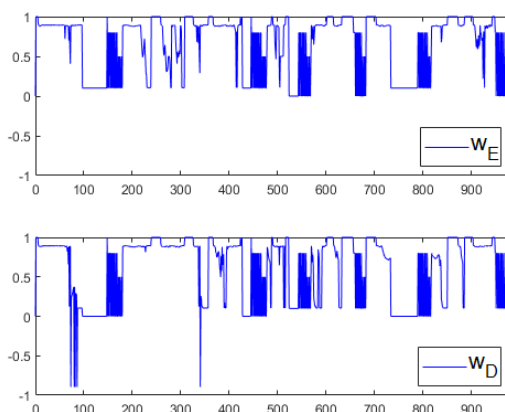


Fonte: Autoria própria.

A partir dos pulsos dos motores dados pelas Figuras 4.4, 4.5 e 4.6 para o FLC e pelas Figuras 4.7, 4.8 e 4.9 (a) para o DFCM e (b) para o DFCM-ACO, pode-se notar que os robôs não entraram em estado iminente de colisão

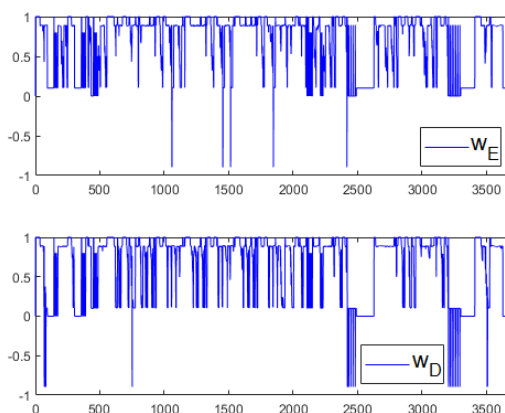
(subcomportamento 3), pois não há pulsos negativos nos motores, o que indicaria uma curva rápida. As áreas em que os pulsos estão mais concentrados são aquelas nas quais o processo de tomada de decisões ocorre, o que sugere melhorias em trabalhos futuros, a fim de reduzir o tempo de processamento para esse mecanismo.

**Figura 4.4 – Pulsos de um robô com FLC: ambiente I**



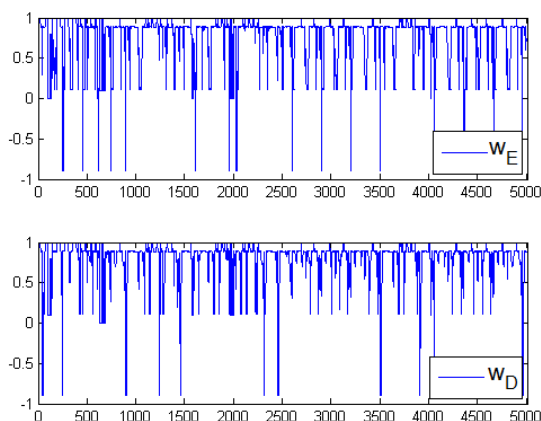
Fonte: Autoria própria.

**Figura 4.5 – Pulsos de um robô com FLC: ambiente II**



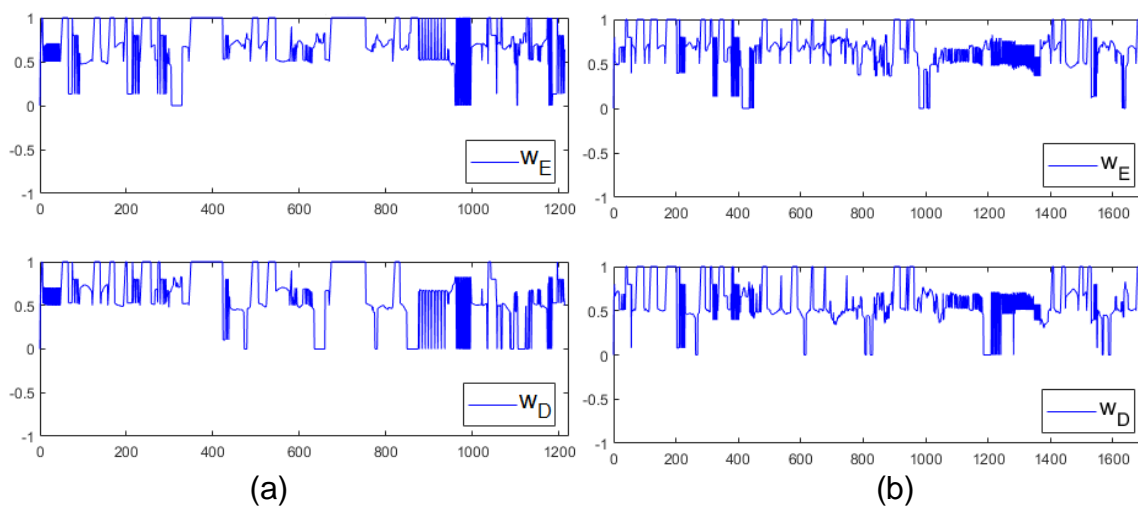
Fonte: Autoria própria.

**Figura 4.6 – Pulsos de um robô com FLC: ambiente III**



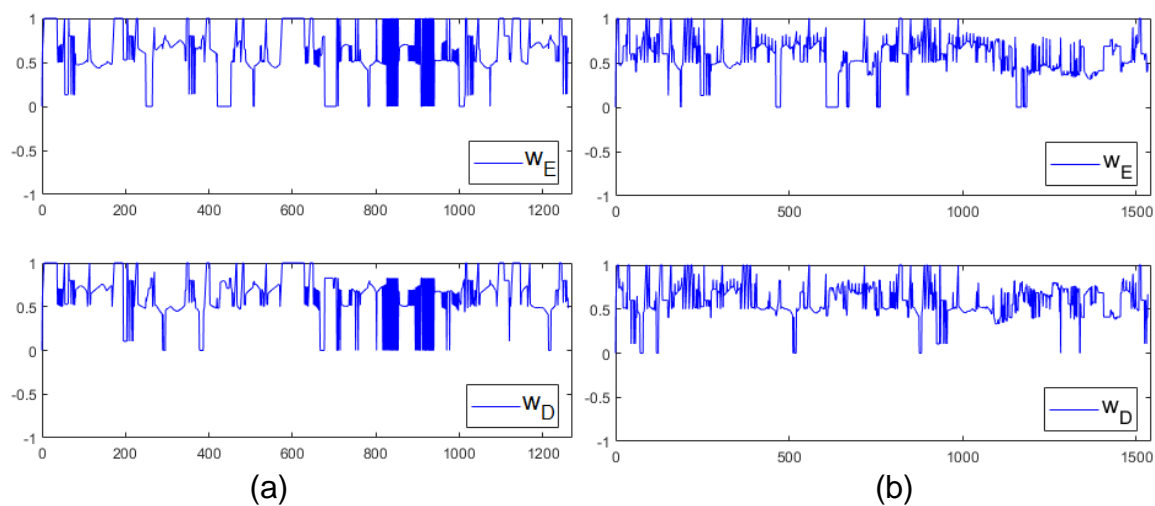
Fonte: Autoria própria.

Figura 4.7 – Pulsos de um robô no ambiente I: (a) DFCM, (b) DFCM-ACO



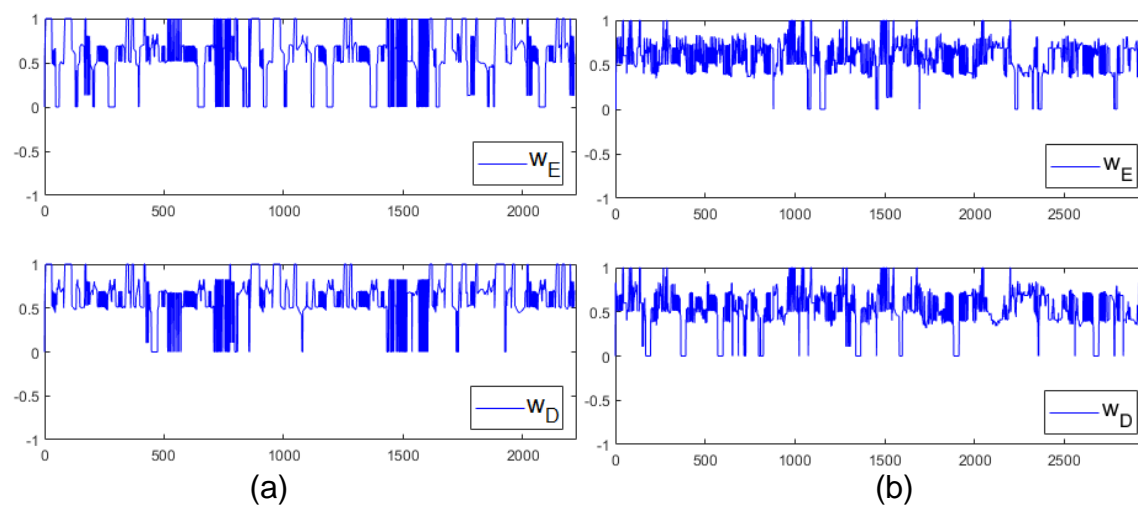
Fonte: Autoria própria.

Figura 4.8 – Pulsos de um robô no ambiente II: (a) DFCM, (b) DFCM-ACO



Fonte: Autoria própria.

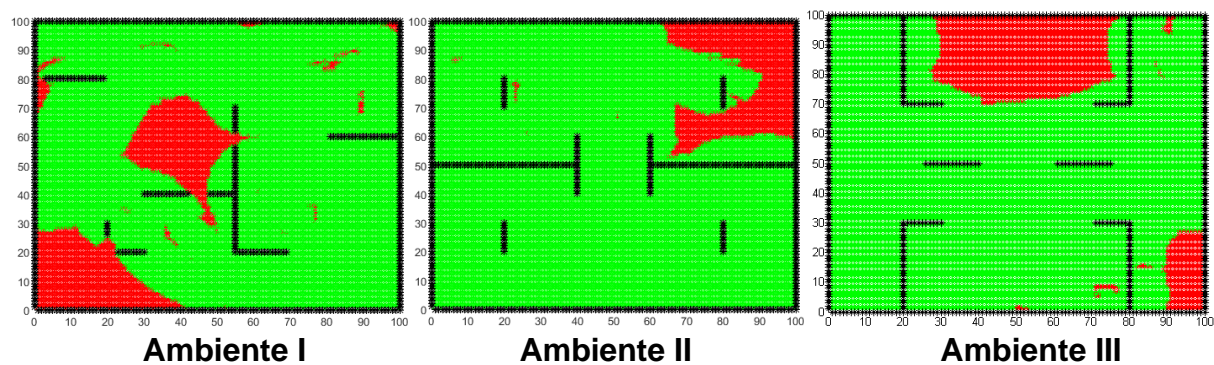
Figura 4.9 – Pulsos de um robô no ambiente III: (a) DFCM, (b) DFCM-ACO



Fonte: Autoria própria.

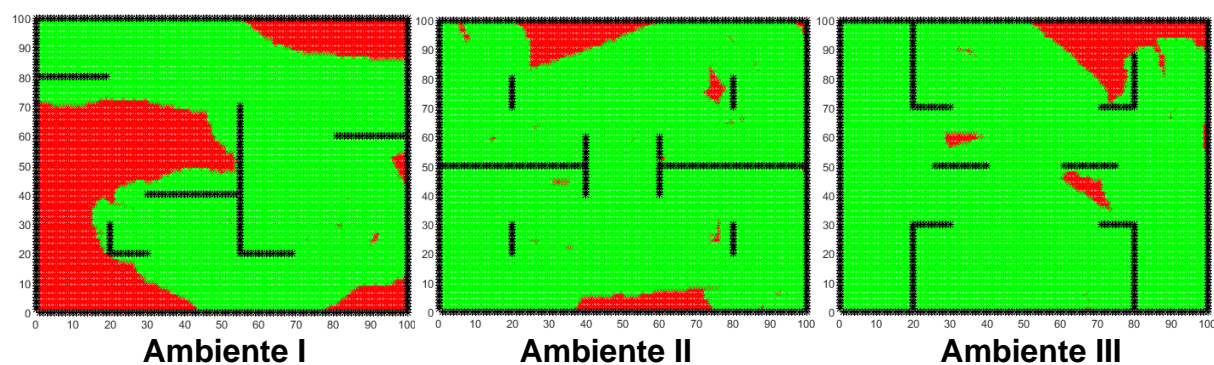


Figura 4.10 – Área explorada por um robô com FLC



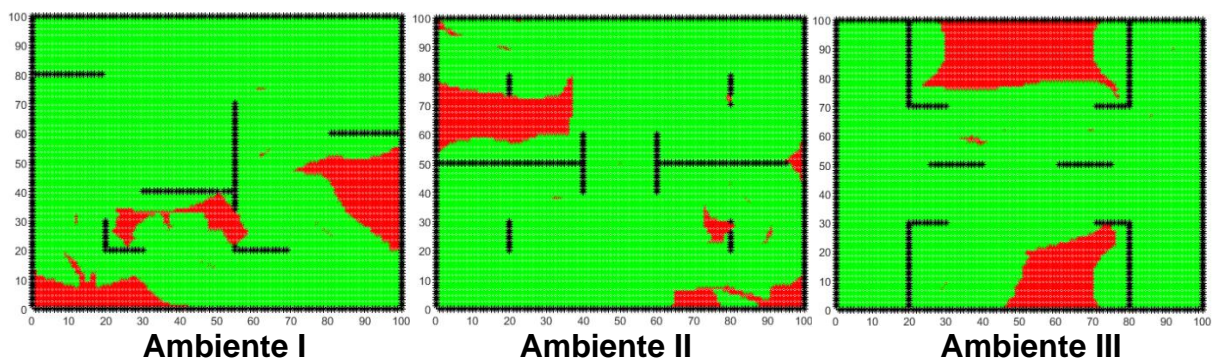
Fonte: Autoria própria.

Figura 4.11 – Área explorada por um robô com DFCM



Fonte: Autoria própria.

Figura 4.12 – Área explorada por um robô com DFCM-ACO



Fonte: Autoria própria.

Os resultados da área explorada estão representados na Figura 4.10 para o FLC, e nas Figuras 4.11 e 4.12 para o DFCM e DFCM-ACO, respectivamente. No primeiro ambiente, menos de 70% da área foi coberta pelo DFCM e 83% foram cobertos pelo FLC, além de 89% completados pelo DFCM-ACO. Considerando um padrão de 85% como o mínimo desejado neste trabalho, os dois primeiros controladores não atingiram essa proporção, enquanto que o DFCM-ACO ficou 4%

acima do mínimo desejado. No entanto, esses resultados foram melhores nos ambientes II e III. Desta vez, o DFCM cobriu uma média de 92,27% da área, o FLC 85,81% e o DFCM-ACO 87,02%. Esses parâmetros podem ser observados no ambiente I, onde uma área significativa não foi coberta pelo DFCM. Nesse caso em particular, uma vítima localizada na seção vermelha não seria detectada.

A Tabela 4.1 mostra os resultados para os controladores no cenário para um robô. Vale ressaltar que, apesar da abordagem DFCM demorar mais para concluir a tarefa no ambiente I, nos outros dois ela mostrou ser pelo menos duas vezes mais rápida, levando em consideração o número de iterações (NI), tempo de processamento (TP), a distância percorrida (DP) e a área explorada (AE).

**Tabela 4.1 – Resultados das simulações para um robô**

1 robô	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO
<b>NI</b>	976,00	1212,00	1705,00	3701,00	1260,00	1534,00	5000,00	2215,00	2976,00
<b>TP (s)</b>	339,00	600,95	262,27	707,01	192,94	227,73	1063,00	331,38	386,78
<b>DP (cm)</b>	508,70	668,75	892,94	2062,58	662,33	748,22	3174,75	1134,47	1372,30
<b>AE (cm<sup>2</sup>)</b>	8374,00	6894,00	8839,00	8800,00	9258,00	9006,00	8362,00	9196,00	8398,00

Fonte: Autoria própria.

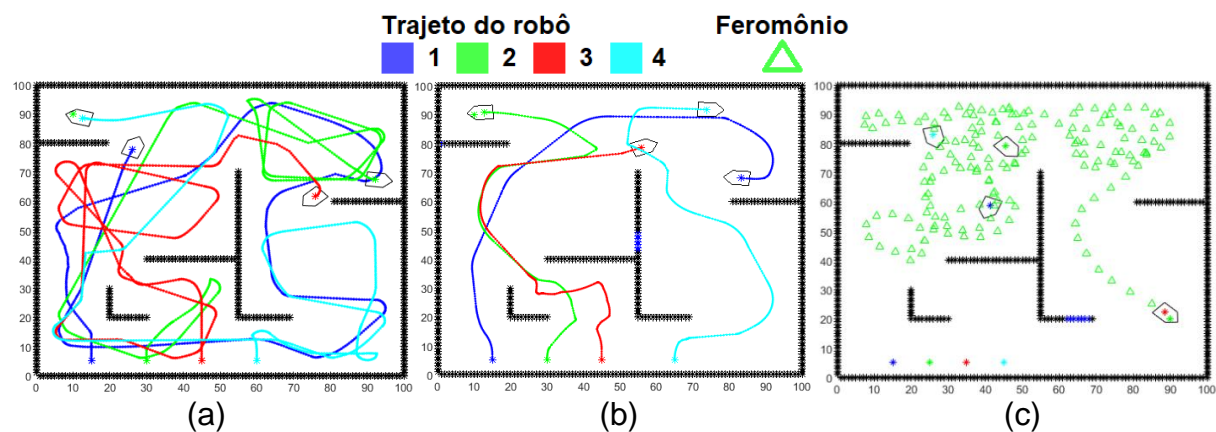
No entanto, considerando a razão entre esses dois parâmetros nos ambientes II e III, o DFCM explorou maiores áreas mesmo percorrendo distâncias menores do que a abordagem FLC. Esse recurso pode sugerir que os robôs consumirão menos bateria para pesquisar áreas maiores em cenários reais.

## 4.2 RESULTADOS PARA QUATRO ROBÔS

Após a implementação com um robô operando sozinho nos três ambientes, um SMR de quatro robôs foi implementado. As trilhas de cada robô no ambiente I são

mostradas respectivamente nas Figuras 4.13 (a) para FLC, 4.13 (b) para o DFCM e 4.13 (c) para o DFCM-ACO.

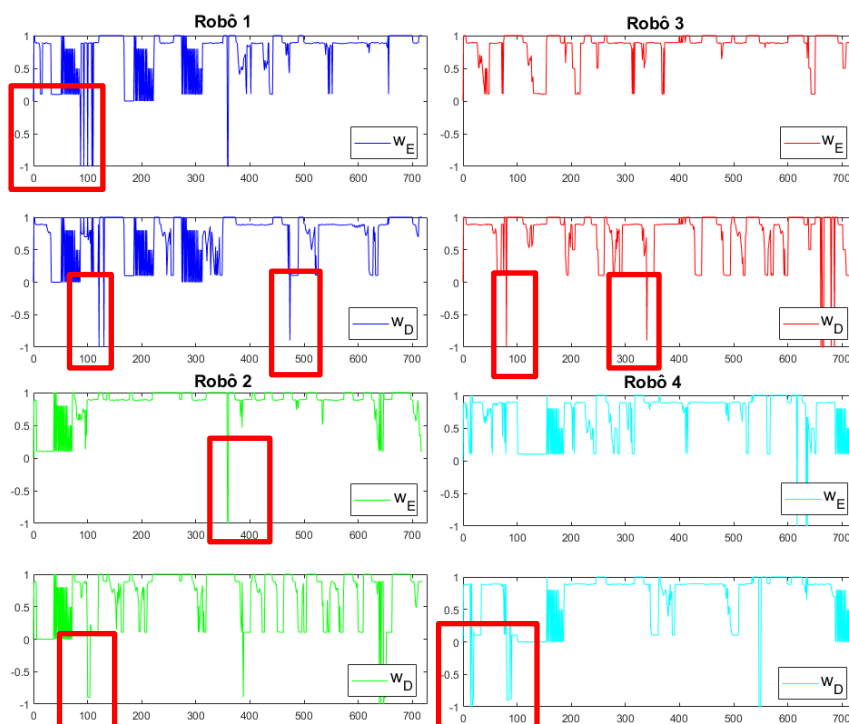
Figura 4.13 – Trajeto de quatro robôs do ambiente I: (a) FLC, (b) DFCM e (c) DFCM-ACO



Fonte: Autoria própria.

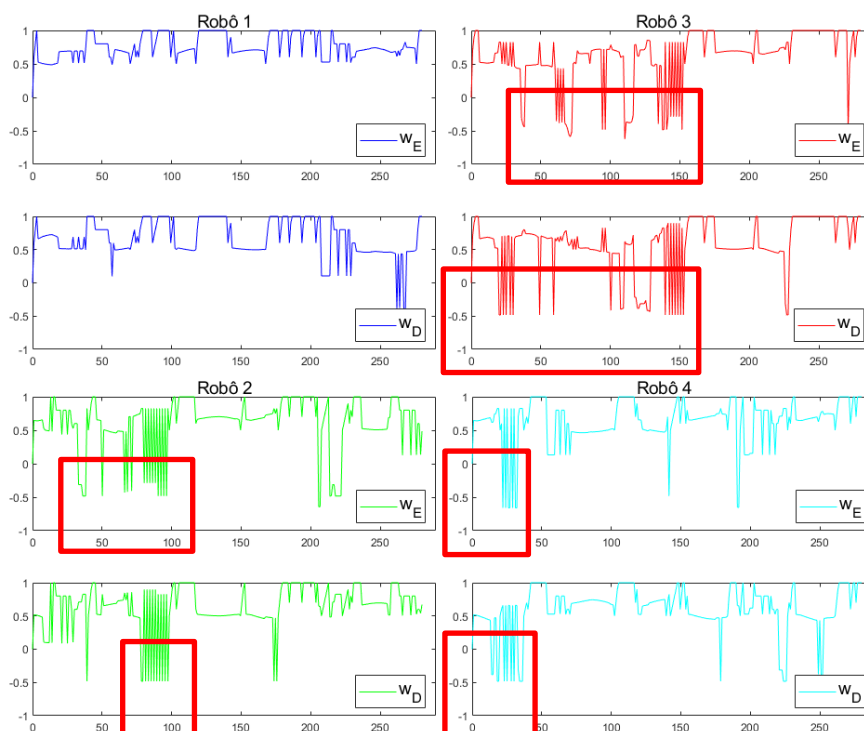
Como visto nas Figuras 4.14 (FLC), 4.15 (DFCM) e 4.16 (DFCM-ACO), as regiões com concentração de pulsos indicando desvio de obstáculos dinâmicos, ou seja, os robôs evitando colisões entre si. Para ambas as estratégias utilizadas, as curvas acentuadas que descrevem esse comportamento são representadas pelos pulsos negativos, destacados em vermelho nas figuras citadas.

Figura 4.14 – Pulsos de quatro robôs com FLC: ambiente I



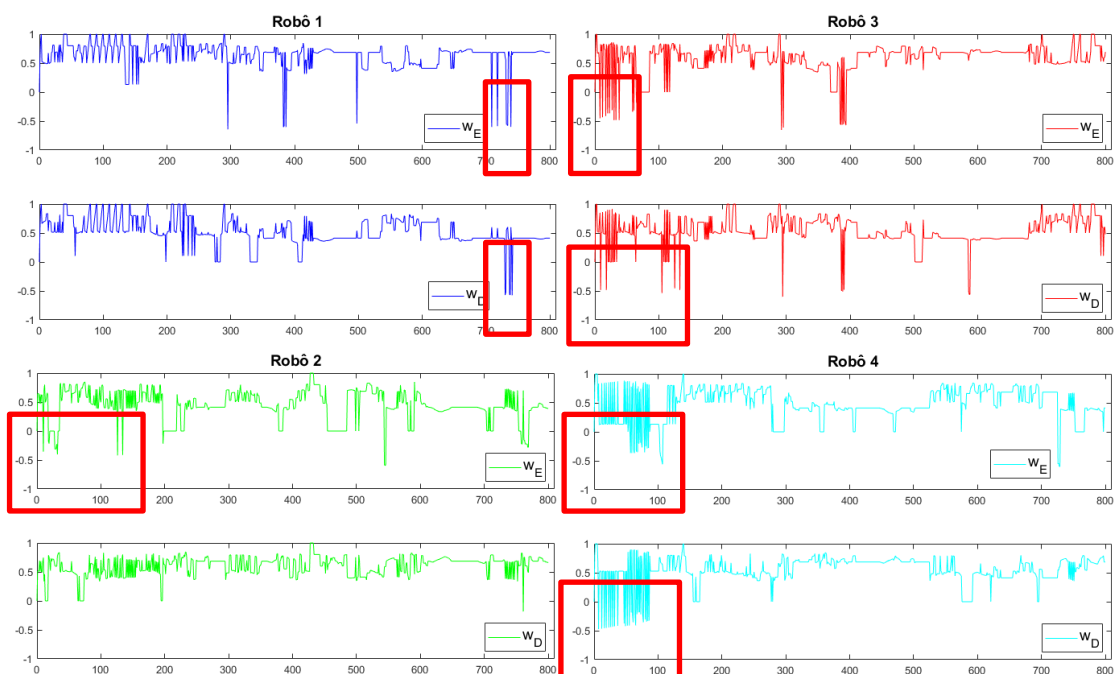
Fonte: Autoria própria.

Figura 4.15 – Pulsos de quatro robôs com DFCM: ambiente I



Fonte: Autoria própria.

Figura 4.16 – Pulsos de quatro robôs com DFCM-ACO: ambiente I



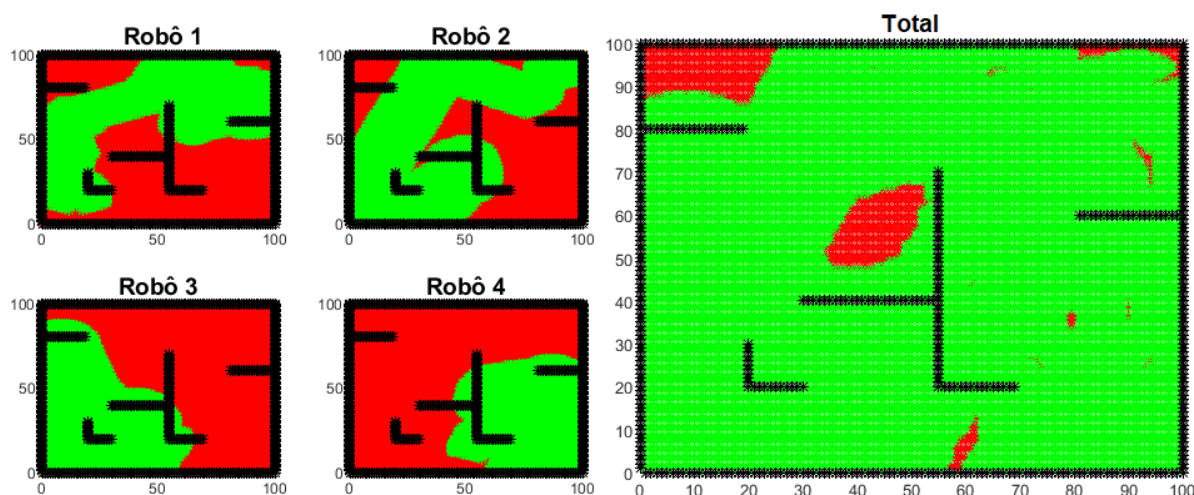
Fonte: Autoria própria.

As regiões cobertas por cada robô e pelo SMR são mostradas na Figura 4.17 (FLC), 4.18 (DFCM) e 4.19 (DFCM-ACO) para o ambiente I, representando os dados da Tabela 4.3 na Seção 4.4. Para o caso do DFCM, o robô 1 cobriu uma área



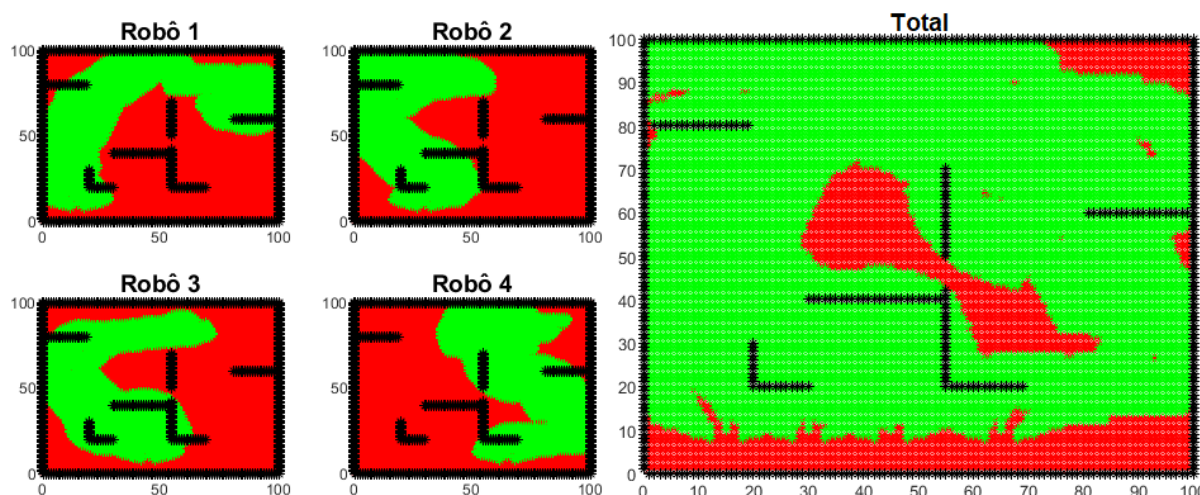
em forma de L invertido. Os robôs 2 e 3 cobriram uma região em forma de C no lado esquerdo do ambiente I. O robô 4 percorreu o lado direito nesse cenário. Em relação a área total, pode-se observar que as áreas não exploradas na seção central e inferior desse ambiente decorreram do cumprimento do objetivo. Em outras palavras, o SMR não explorou esses trechos devido ao critério de parada: encontrar todas as vítimas.

**Figura 4.17 – Área explorada pelo SMR com FLC: ambiente I**



Fonte: Autoria própria.

**Figura 4.18 – Área explorada pelo SMR com DFCM: ambiente I**

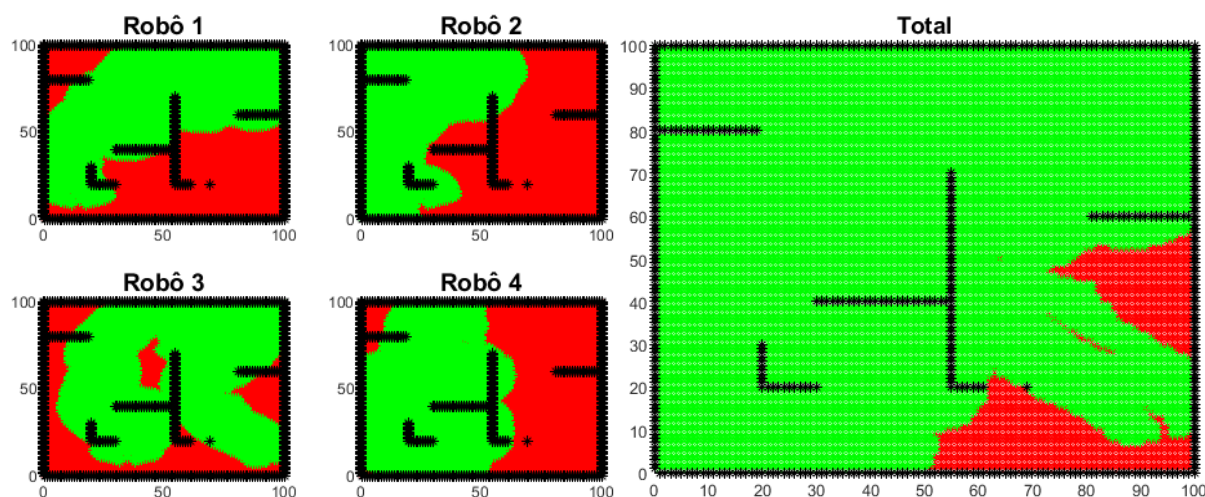


Fonte: Autoria própria.

Considerando o segundo ambiente, observa-se pela Figura 4.20 que um problema da abordagem DFCM é que somente um dos robôs alcança o quadrante superior, enquanto que utilizando FLC e DFCM-ACO dois robôs conseguem esse fato. No entanto, a área explorada não foi prejudicada, pois apresentou 87% de cobertura

(Figuras 4.24, 4.25 e 4.26) e os valores individuais desse parâmetro são próximos o suficiente do FLC (90,15%) e do DFCM-ACO (98,09%), que demorou mais tempo e consumiu mais tempo de processamento nos dois primeiros ambientes.

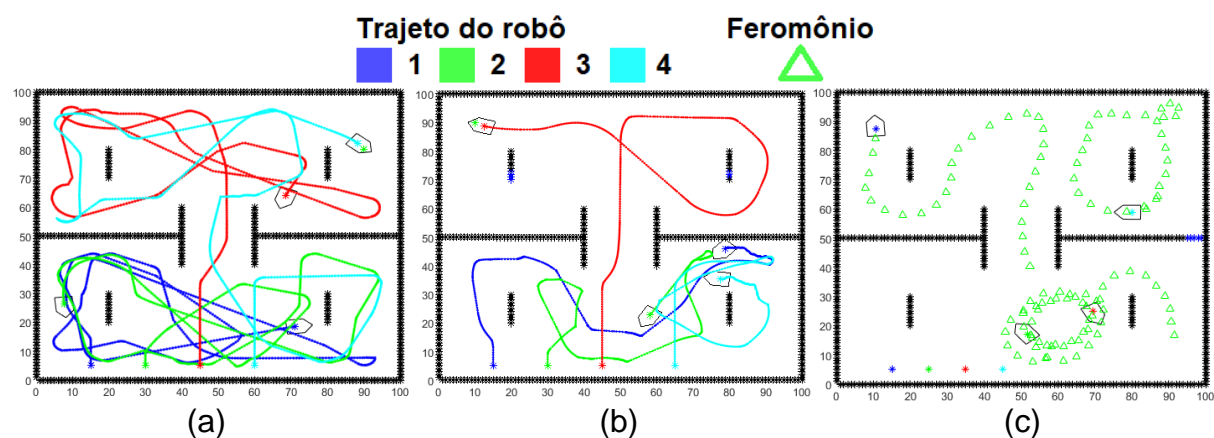
**Figura 4.19 – Área explorada pelo SMR com DFCM-ACO: ambiente I**



Fonte: Autoria própria.

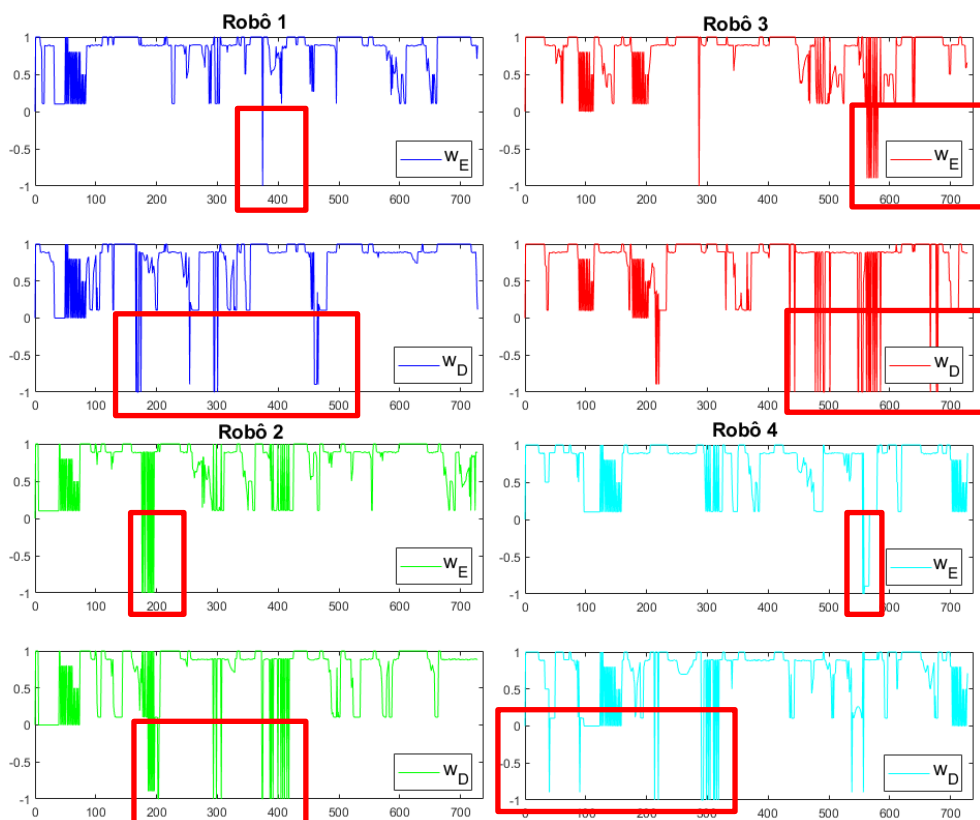
Na abordagem FLC, a iminência de colisão pode ser vista na Figura 4.21 nos robôs 2 e 3, principalmente nas regiões próximas de 200 e 600 iterações. Já para o DFCM, faz-se presente nos robôs 1 e 4 (Figura 4.22) nos quais os pulsos alternam repetidamente no intervalo entre 300 a 400 iterações. Pela Figura 4.23, pode-se constatar que a abordagem DFCM-ACO foi a que menos ativou o subcomportamento 3, algo atingido pela utilização de feromônios. Essa ativação é mostrada por meio dos retângulos vermelhos em destaque nas Figuras 4.21 a 4.23 para o ambiente II.

**Figura 4.20 – Trajeto de quatro robôs do ambiente II: (a) FLC, (b) DFCM e (c) DFCM-ACO**



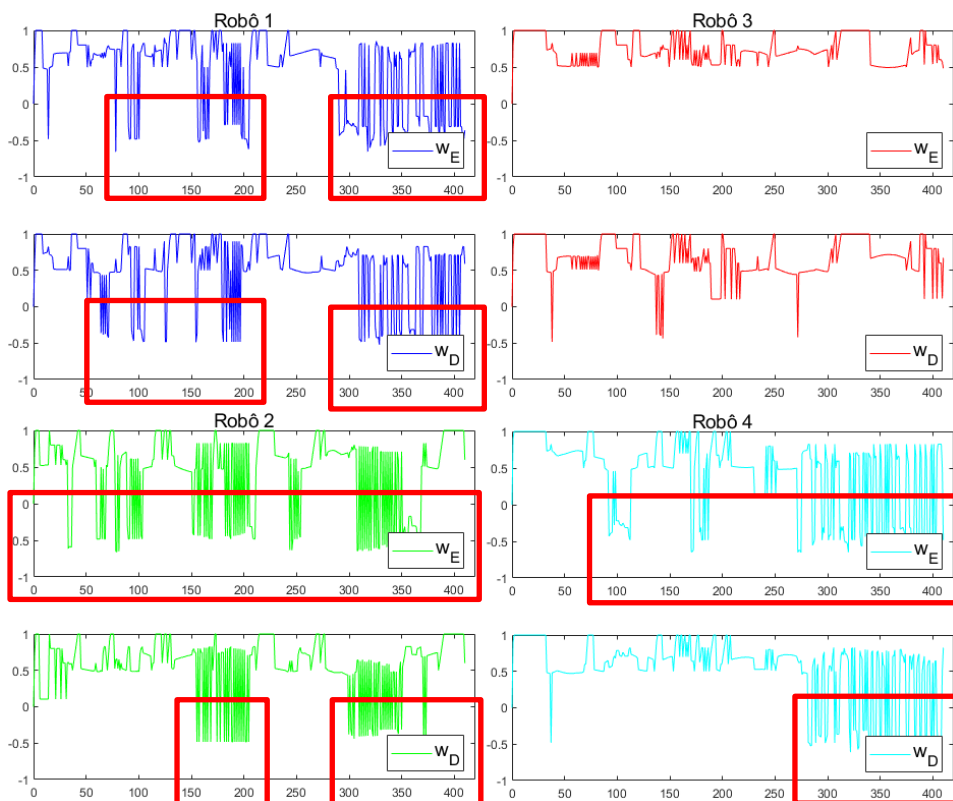
Fonte: Autoria própria.

Figura 4.21 – Pulsos de quatro robôs com FLC: ambiente II



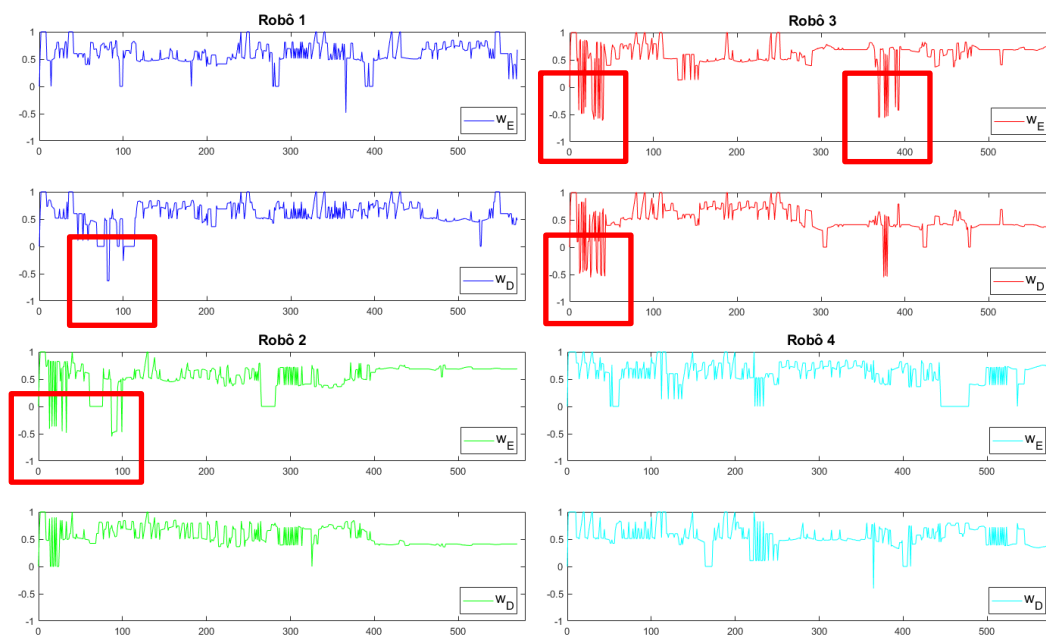
Fonte: Autoria própria.

Figura 4.22 – Pulsos de quatro robôs com DFCM: ambiente II



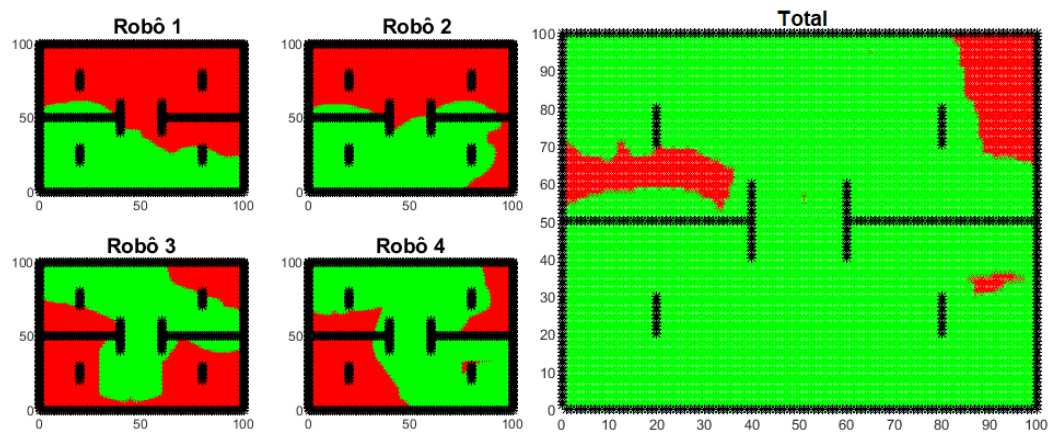
Fonte: Autoria própria.

Figura 4.23 – Pulsos de quatro robôs com DFCM-ACO: ambiente II



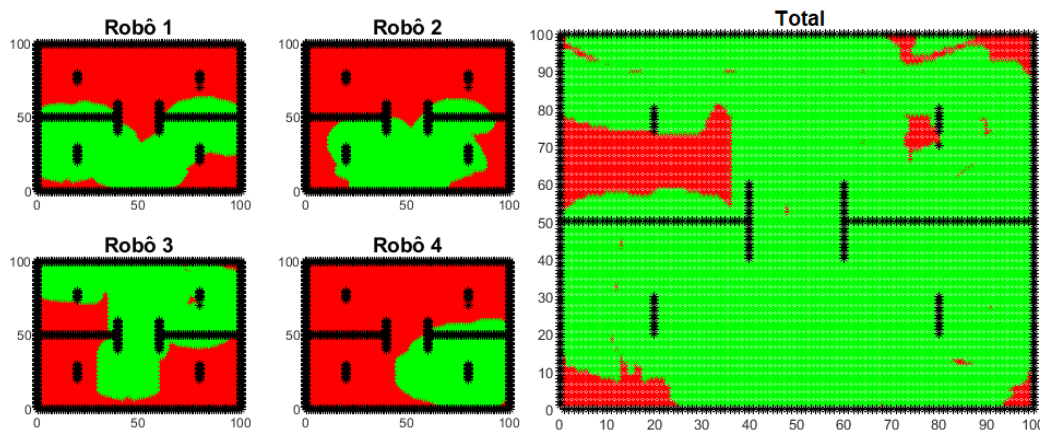
Fonte: Autoria própria.

Figura 4.24 – Área explorada pelo SMR com FLC: ambiente II



Fonte: Autoria própria.

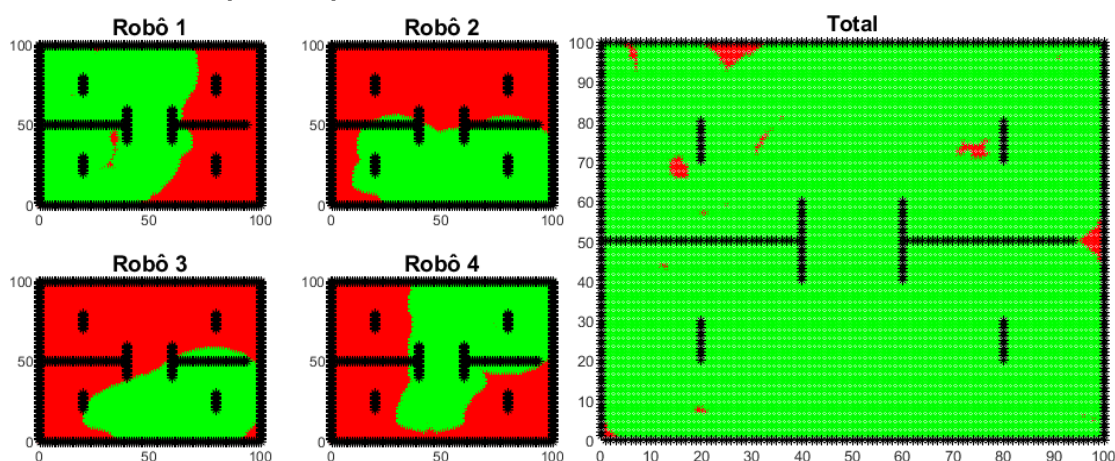
Figura 4.25 – Área explorada pelo SMR com DFCM: ambiente II



Fonte: Autoria própria.



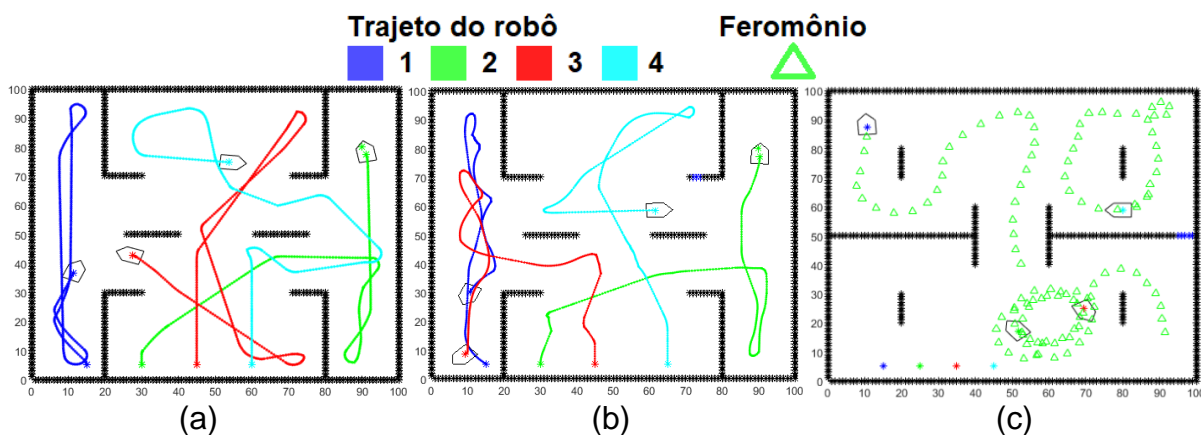
Figura 4.26 – Área explorada pelo SMR com DFCM-ACO: ambiente II



Fonte: Autoria própria.

De acordo com as Figuras 4.24, 4.25 e 4.26, pode-se verificar que a dispersão dos robôs para o DFCM foi menor que a dos demais controladores, uma vez que nela somente o robô 4 explorou o quadrante superior (Figura 4.25). Nos outros casos, dois robôs conseguiram o feito: 3 e 4 para o FLC e 1 e 4 para o DFCM-ACO.

Figura 4.27 – Trajeto de quatro robôs do ambiente III: (a) FLC, (b) DFCM e (c) DFCM-ACO

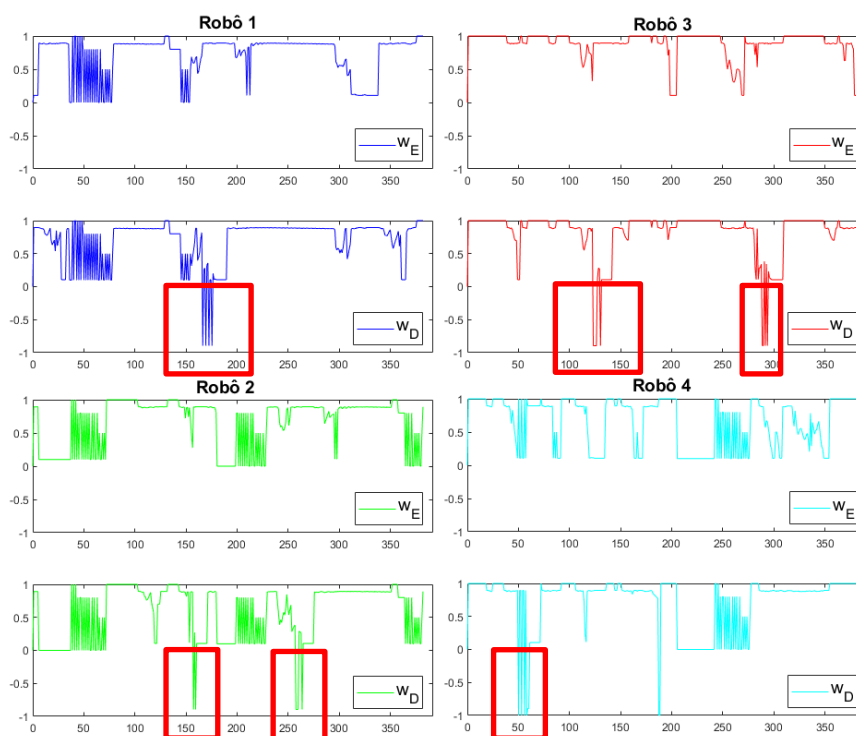


Fonte: Autoria própria.

No terceiro ambiente, os robôs encontraram dificuldades semelhantes às vistas nos experimentos com um robô. No entanto, neste caso (Figura 4.27 (a) para o FLC, (b) DFCM e (c) para o DFCM-ACO), o subcomportamento de iminência de colisão entre robôs funcionou como um mecanismo para alterar sua orientação (Figuras 4.28, 4.29 e 4.30). No DFCM, esse comportamento é visto nos robôs 3 e 4 próximos das posições (30-45, 15), entre as iterações 25 e 50 (Figura 4.29): ambos os robôs mudaram suas orientações para evitar o contato, e esse fato levou o robô 3

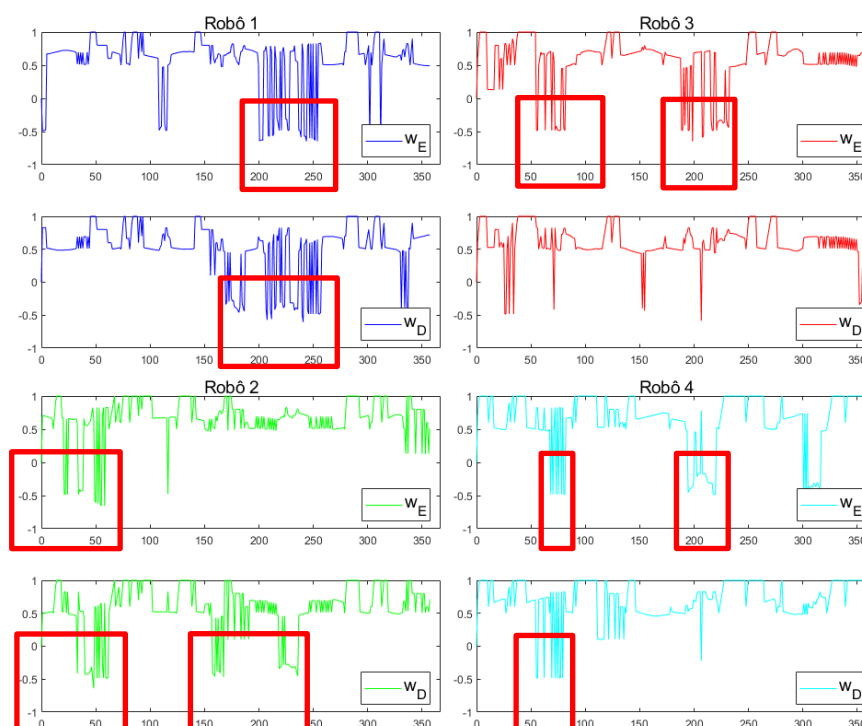
a procurar no lado direito do terceiro ambiente. Já na Figura 4.30, pode-se observar que esse comportamento é acionado brevemente apenas nos robôs 2 e 3.

**Figura 4.28 – Pulsos de quatro robôs com FLC: ambiente III**



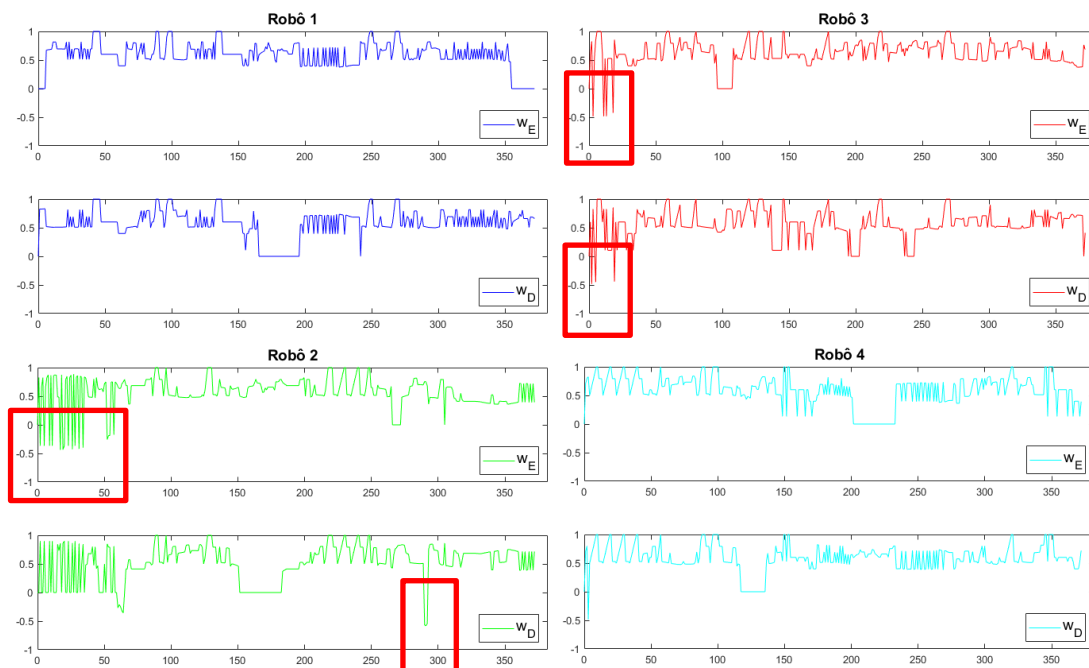
Fonte: Autoria própria.

**Figura 4.29 – Pulsos de quatro robôs com DFCM: ambiente III**



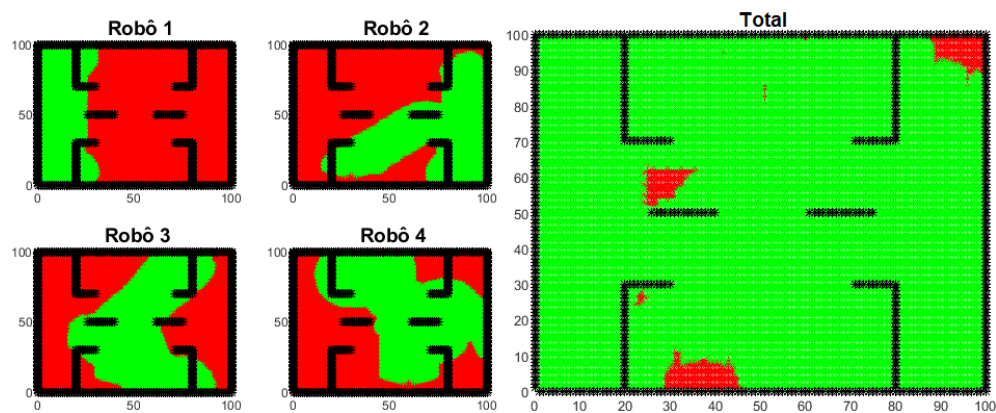
Fonte: Autoria própria.

Figura 4.30 – Pulsos de quatro robôs com DFCM-ACO: ambiente III



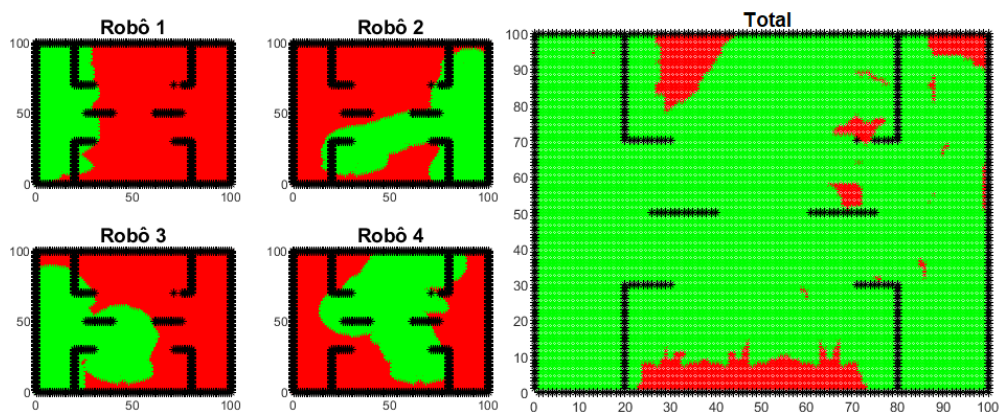
Fonte: Autoria própria.

Figura 4.31 – Área explorada pelo SMR com FLC: ambiente III, 4 robôs



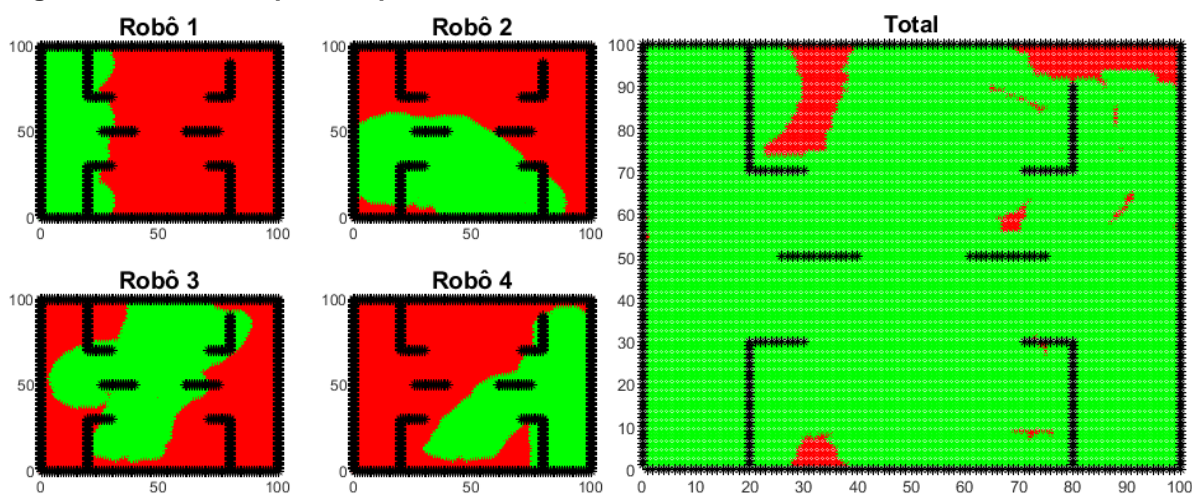
Fonte: Autoria própria.

Figura 4.32 – Área explorada pelo SMR com DFCM: ambiente III, 4 robôs



Fonte: Autoria própria.

Figura 4.33 – Área explorada pelo SMR com DFCM-ACO: ambiente III, 4 robôs



Fonte: Autoria própria.

Para o cenário de quatro robôs, a abordagem utilizando DFCM obteve resultados menores em relação ao número de iterações (NI) e tempo de processamento (TP), como observado na Tabela 4.2. Nela, o único resultado divergente é o tempo de processamento do DFCM no terceiro ambiente. Essa diferença pode ser devida ao fato de que os robôs 1, 2 e 4 ficaram “presos” na seção esquerda do ambiente, como pode ser visto na Figura 4.20 (b), enquanto que para o FLC (Figura 4.20 (a)) e DFCM-ACO (Figura 4.20 (c)), dois robôs conseguiram se dispersar para a parte superior do ambiente.

Tabela 4.2 – Número de iterações e tempo de processamento para 4 robôs

4 Robôs	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO
NI	717,00	279,00	800,00	727,00	409,00	570,00	381,00	356,00	372,00
TP (s)	604,67	239,30	510,85	713,29	323,47	366,31	228,11	541,76	255,07

Fonte: Autoria própria.

Os robôs com a abordagem DFCM exploraram 90,12% do ambiente III (Tabela 4.4), como mostram as Figuras 4.31 a 4.33 para FLC, DFCM e DFCM-ACO, respectivamente. Com o DFCM, a região ao redor de (35, 90) não foi explorada, novamente, devido ao critério de parada, ou seja, todas as vítimas foram encontradas antes dessa região ser explorada. Nesse cenário, os robôs 1 e 2 cobriram as laterais do ambiente, enquanto que os robôs 3 e 4 (principalmente) cobriram a área central.

Tabela 4.3 – Área explorada para 4 robôs

Área expl. (cm <sup>2</sup> )	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM- ACO	FLC	DFCM	DFCM- ACO	FLC	DFCM	DFCM- ACO
<b>Robô 1</b>	4178,00	3812,00	5429,00	3778,00	4082,00	5999,00	2545,00	2593,00	2657,00
<b>Robô 2</b>	5239,00	3399,00	4411,00	4391,00	3345,00	4156,00	3614,00	3680,00	3712,00
<b>Robô 3</b>	3724,00	3327,00	6423,00	4471,00	5058,00	3259,00	4023,00	3564,00	4060,00
<b>Robô 4</b>	3176,00	3415,00	5191,00	6047,00	2582,00	4401,00	4498,00	3753,00	3445,00
<b>Total</b>	9277,00	7849,00	8916,00	9015,00	8739,00	9809,00	9622,00	9012,00	9343,00

Fonte: Autoria própria.

Tabela 4.4 – Distância percorrida para 4 robôs

Distância percorrida (cm)	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM- ACO	FLC	DFCM	DFCM- ACO	FLC	DFCM	DFCM- ACO
<b>Robô 1</b>	427,39	166,63	389,09	471,06	162,12	283,77	224,54	158,83	181,93
<b>Robô 2</b>	450,03	143,96	349,03	429,58	168,96	262,38	192,53	182,71	168,34
<b>Robô 3</b>	472,49	136,09	374,29	456,00	238,21	258,87	268,83	174,10	182,98
<b>Robô 4</b>	412,62	142,85	340,74	430,02	162,58	273,60	216,30	197,27	182,37
<b>Total</b>	1762,53	589,53	1453,15	1786,66	731,87	1078,62	902,20	712,91	715,62

Fonte: Autoria própria.

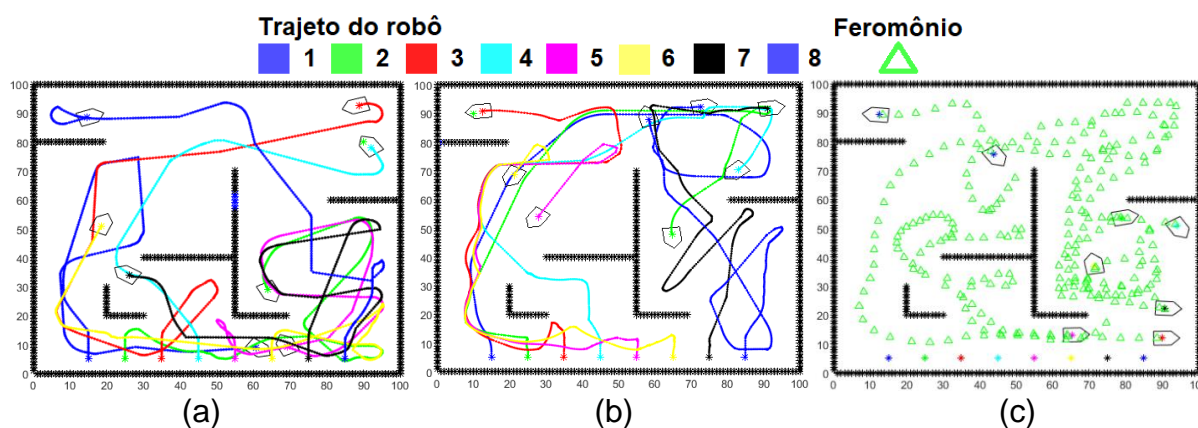
A partir dos resultados das Tabelas 4.3 e 4.4 pode-se observar que o SMR com a abordagem DFCM percorreu distâncias significativamente menores sem prejuízos para a área explorada total. Em outras palavras, em aplicações reais, essa característica concede aos robôs uma vida útil prolongada da bateria. Assim, a razão entre distância percorrida e área explorada indica que a abordagem DFCM cobre mais área por unidade de distância.

### 4.3 RESULTADOS PARA OITO ROBÔS

O último cenário proposto possui um SMR com oito robôs. Diferentemente dos outros, um e quatro robôs, a abordagem DFCM concluiu a tarefa com menos

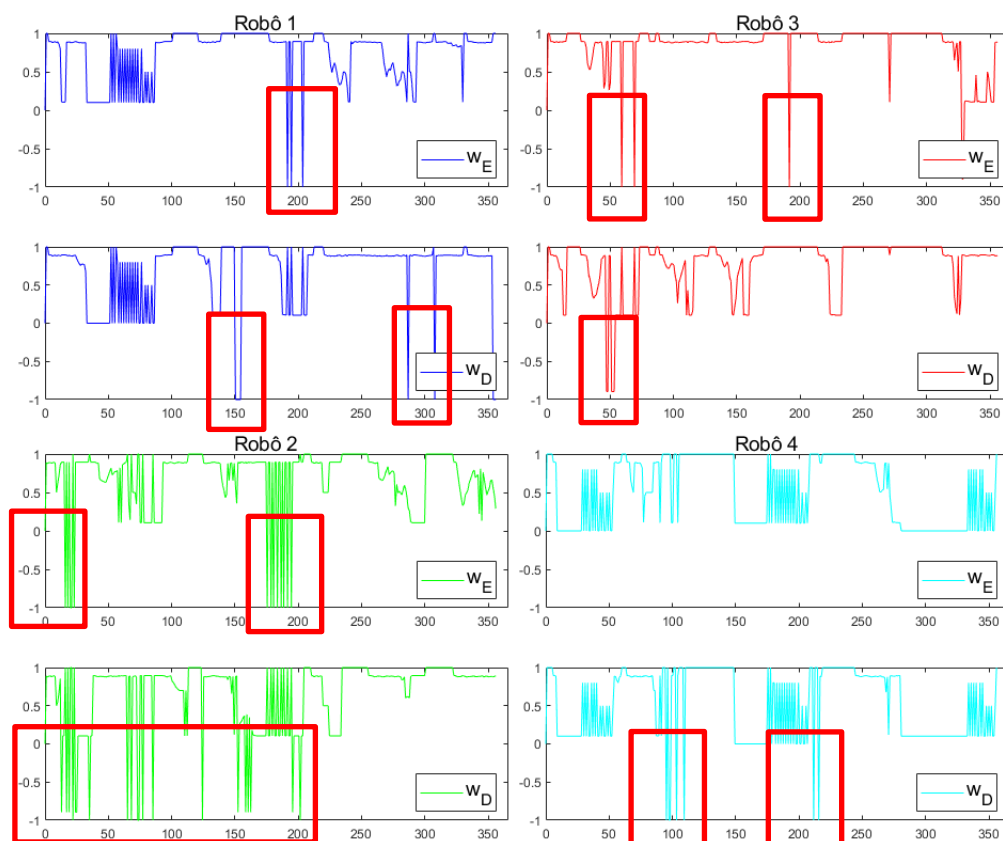
iterações apenas no segundo ambiente. Assim, os resultados das simulações do primeiro ambiente são mostrados na Figura 4.34 para as trajetórias dos robôs, nas Figuras 4.35 e 4.36 (FLC); 4.37, 4.38 (DFCM); 4.39 e 4.40 (DFCM-ACO) para os pulsos das rodas, enquanto que as áreas exploradas podem ser vistas nas Figuras 4.41, 4.42 e 4.43. Pode-se notar que, ao comparar os três cenários (um, quatro e oito robôs), o número de iterações cai significativamente.

Figura 4.34 – Trajeto de oito robôs do ambiente I: (a) FLC, (b) DFCM e (c) DFCM-ACO



Fonte: Autoria própria.

Figura 4.35 – Pulsos de oito robôs com FLC: ambiente I, robôs 1 a 4



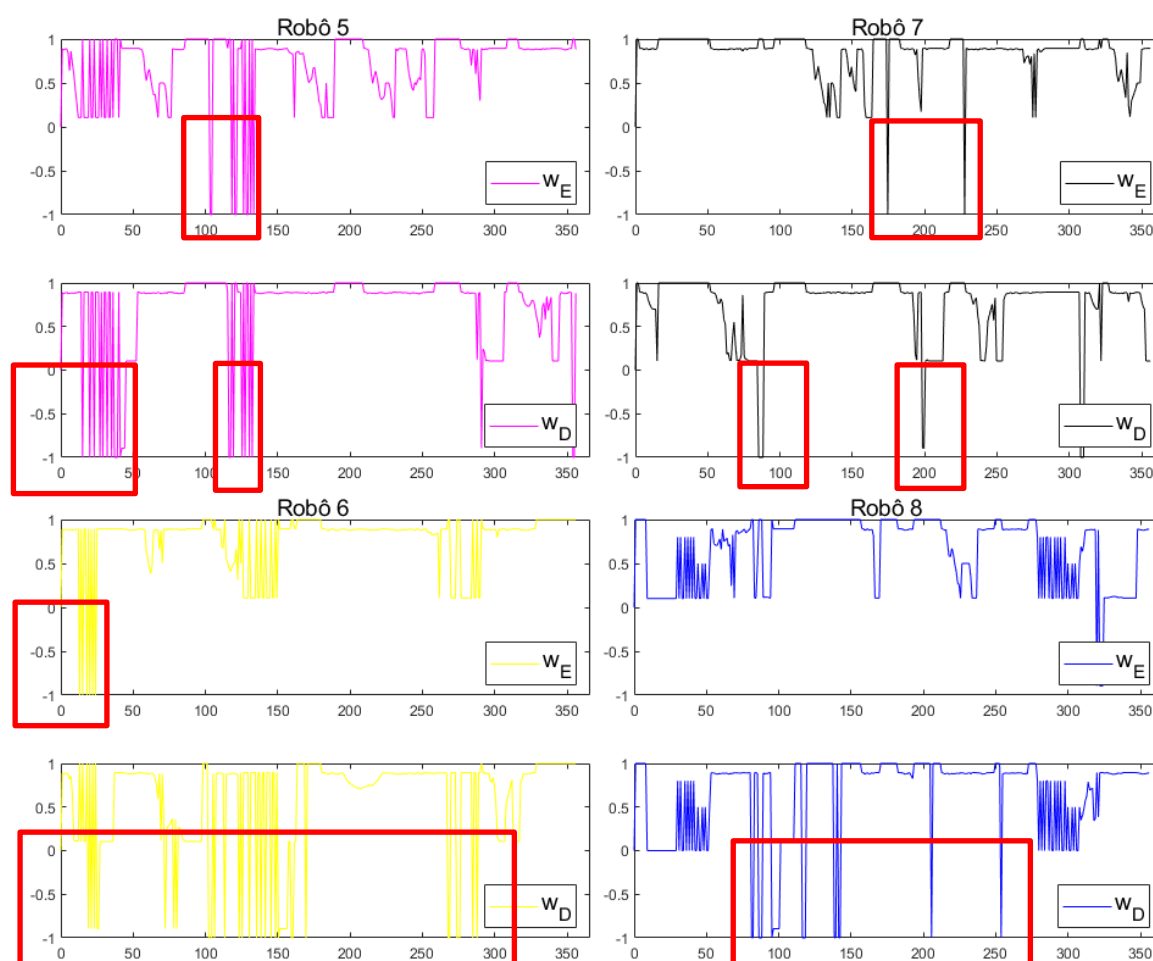
Fonte: Autoria própria.



Em particular, a abordagem DFCM no primeiro ambiente foi realizada em 1212 iterações com um robô e concluiu a tarefa em 400 iterações com oito robôs. No terceiro ambiente, utilizando o FLC, a diferença é mais nítida: com um robô não foi possível encontrar as vítimas dentro dos parâmetros desejados, enquanto que com um SMR de 8 robôs os objetivos foram concluídos em 155 iterações.

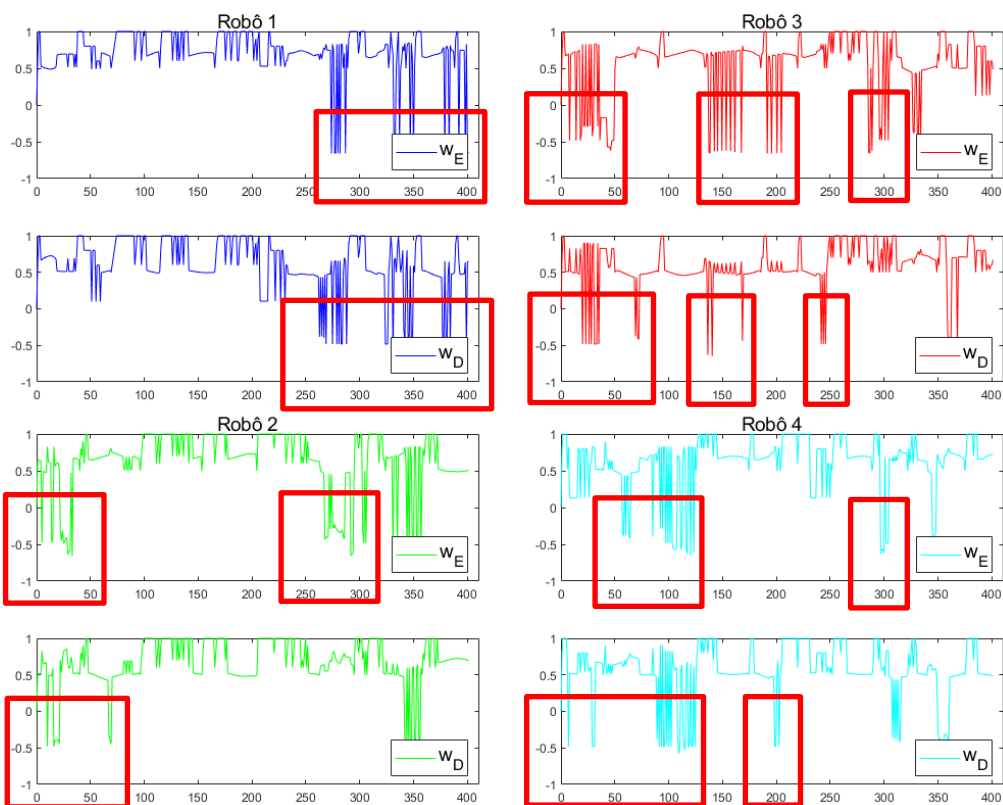
Com a utilização de mais robôs, a proximidade entre eles, causada pelo tamanho fixo dos ambientes e pelo aumento de seu número no SMR, desencadeou o subcomportamento de colisão iminente mais vezes em comparação com os outros cenários, como visto nas Figuras 4.37 e 4.48 através dos robôs 5 e 8 da abordagem DFCM, por exemplo. As zonas com o comportamento de iminência de colisão mais intenso estão destacadas em vermelho conforme visto nas Figuras 4.35 a 4.40. Com isso, principalmente a partir da comparação entre DFCM com FLC e DFCM-ACO, tem-se que, neste caso, a primeira abordagem acionou expressivamente mais vezes a iminência de colisão entre os robôs (Figuras 4.37 e 4.38).

**Figura 4.36 – Pulsos de oito robôs com FLC: ambiente I, robôs 5 a 8**



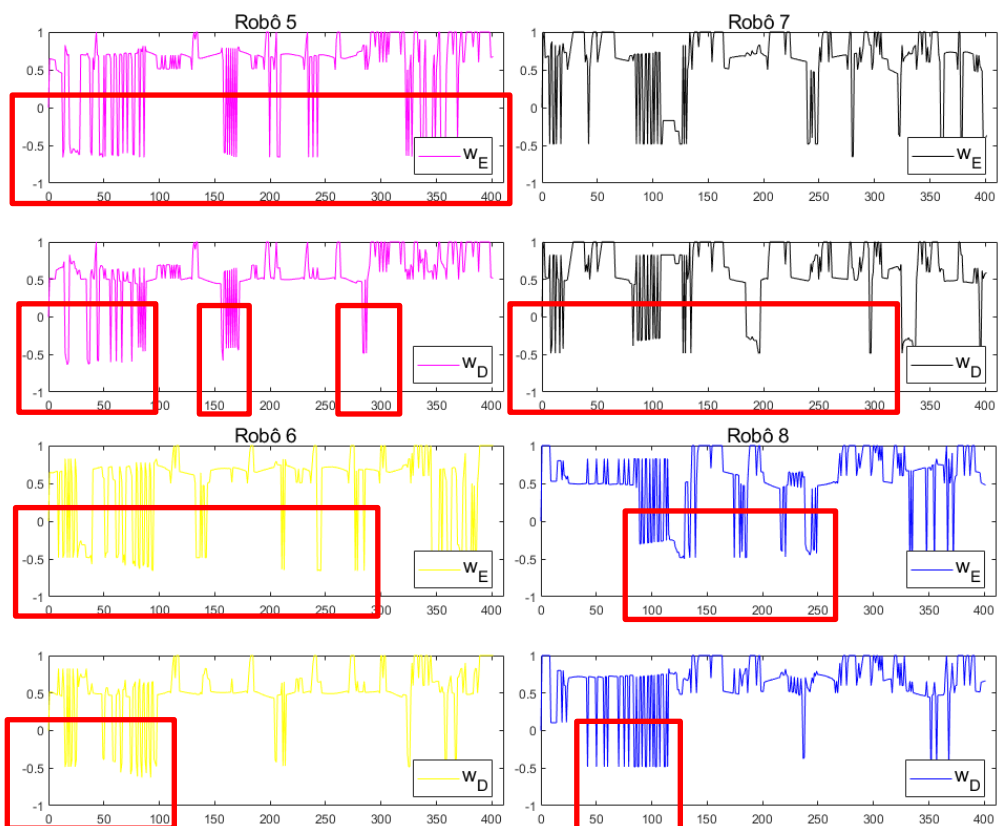
Fonte: Autoria própria.

Figura 4.37 – Pulsos de oito robôs com DFCM: ambiente I, robôs 1 a 4



Fonte: Autoria própria.

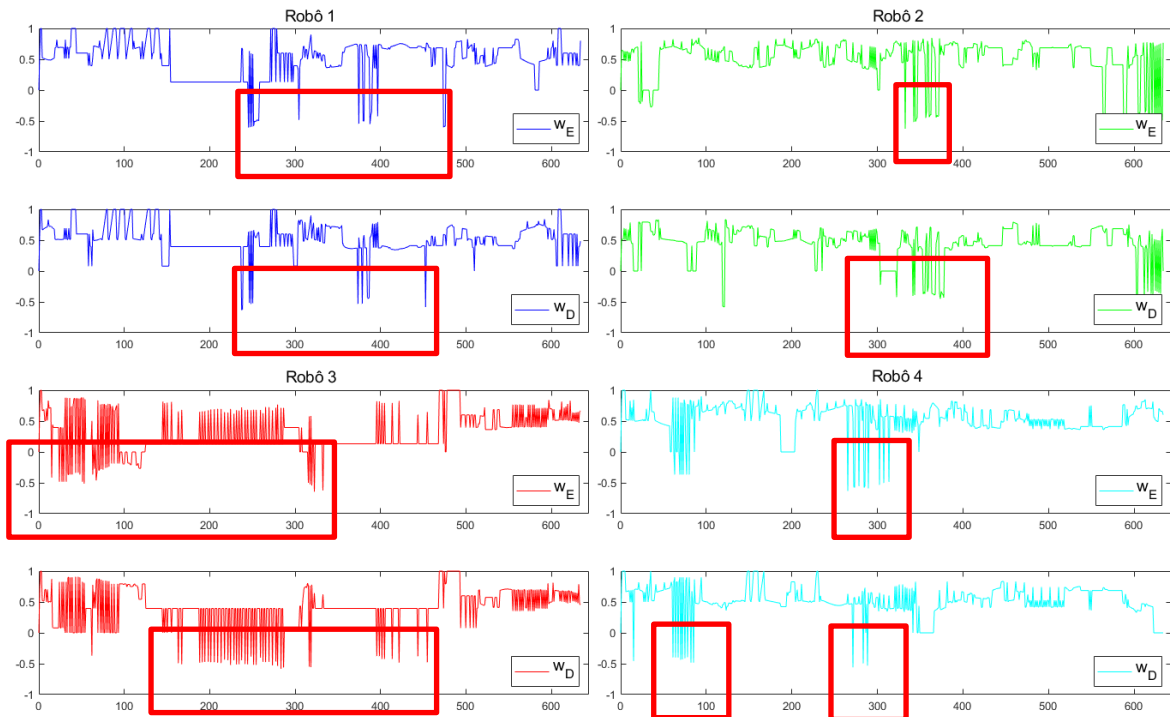
Figura 4.38 – Pulsos de oito robôs com DFCM: ambiente I, robôs 5 a 8



Fonte: Autoria própria.

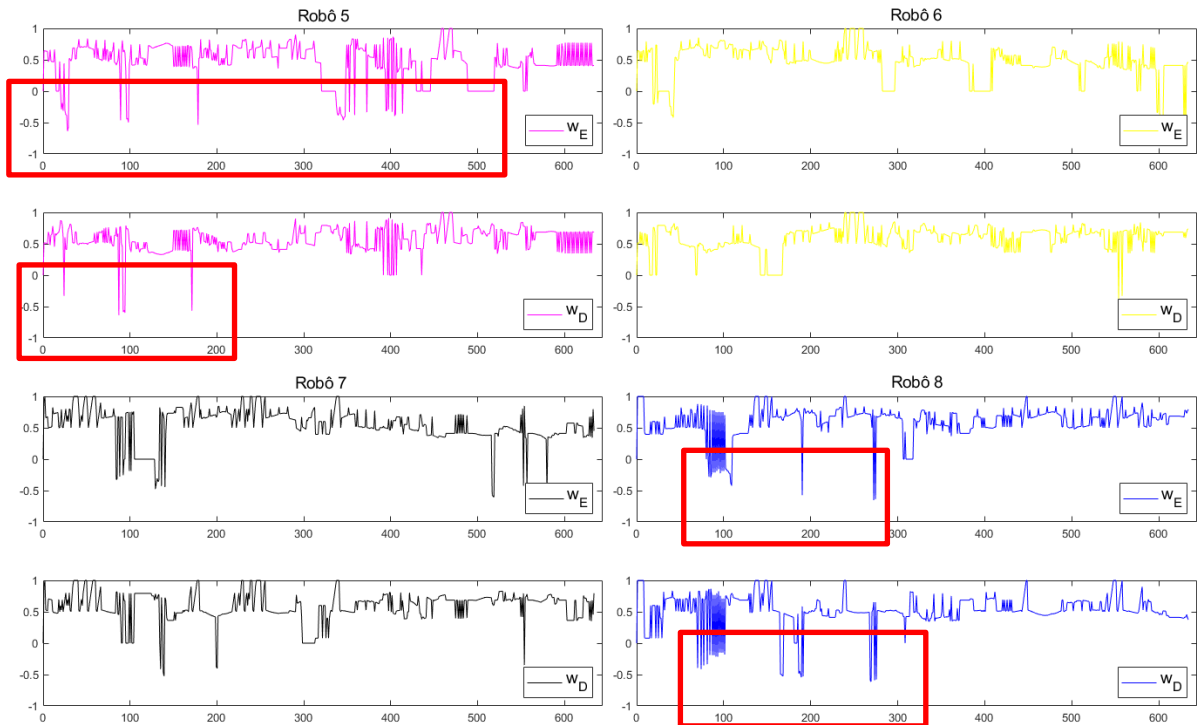


Figura 4.39 – Pulsos de oito robôs com DFCM-ACO: ambiente I, robôs 1 a 4



Fonte: Autoria própria.

Figura 4.40 – Pulsos de oito robôs com DFCM-ACO: ambiente I, robôs 5 a 8

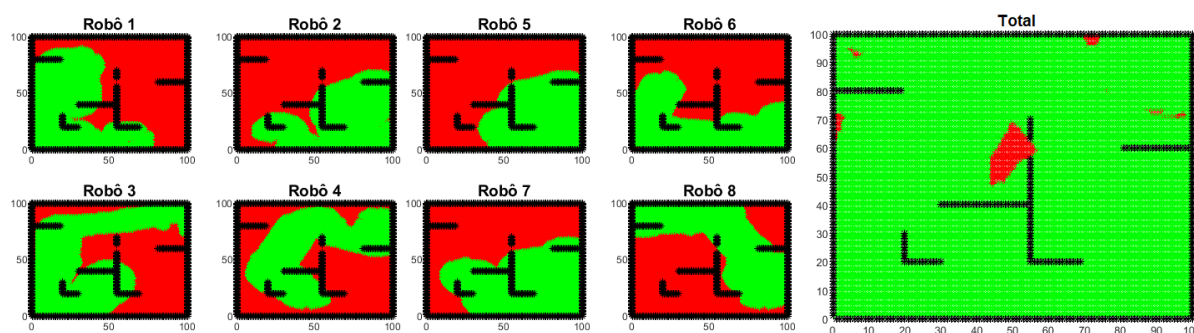


Fonte: Autoria própria.

Como os robôs concluíram a tarefa de resgate em menos tempo do que os outros cenários, menores áreas foram exploradas nos ambientes II e III devido ao

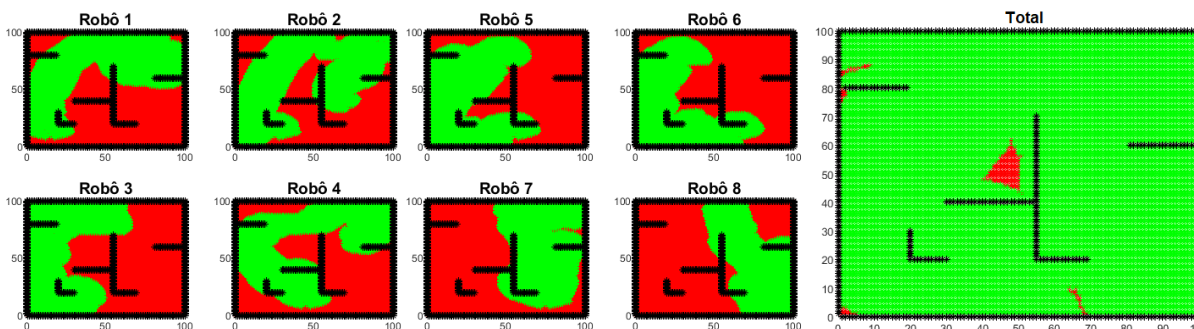
critério de parada de resgate das vítimas. No entanto, no primeiro ambiente, a abordagem DFCM explorou 98,40% da área, como pode ser visto nas Figuras 4.41 (FLC, que explorou 97,57% da área), 4.42 (DFCM) e 4.43 (DFCM-ACO, que explorou 99,73% da área). Pela Figura 4.42, tem-se que os robôs 5 e 6 cobriram praticamente a mesma parcela de área, embora não tenham viajado lado a lado em nenhum momento nas simulações.

**Figura 4.41 – Áreas individuais e total exploradas pelo SMR com FLC: ambiente I, 8 robôs**



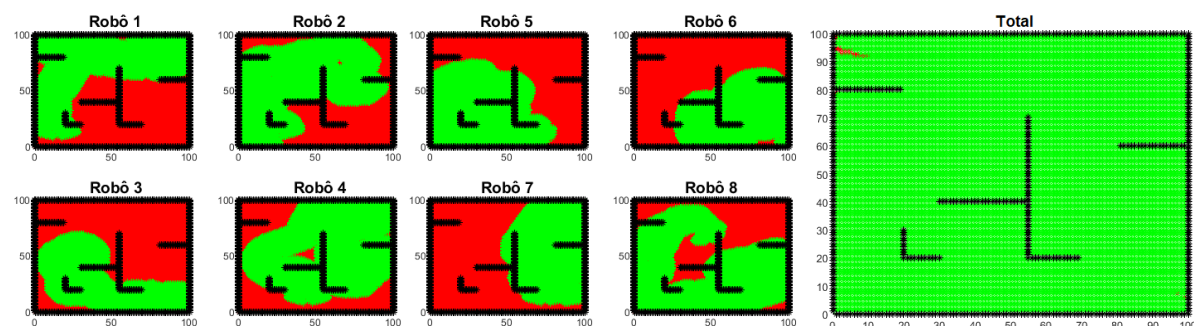
Fonte: Autoria própria.

**Figura 4.42 – Áreas individuais e total exploradas pelo SMR com DFCM: ambiente I, 8 robôs**



Fonte: Autoria própria.

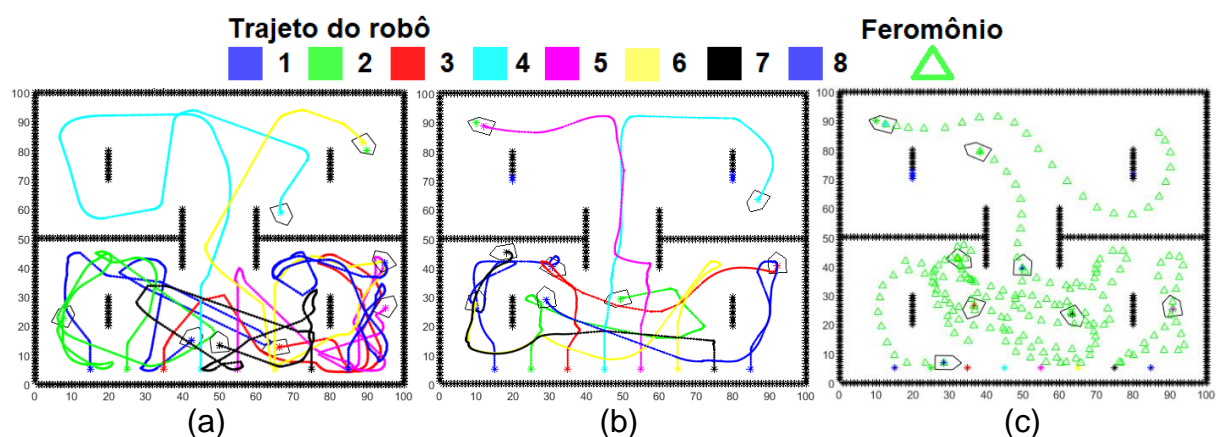
**Figura 4.43 – Áreas individuais e total exploradas pelo SMR (DFCM-ACO): ambiente I, 8 robôs**



Fonte: Autoria própria.

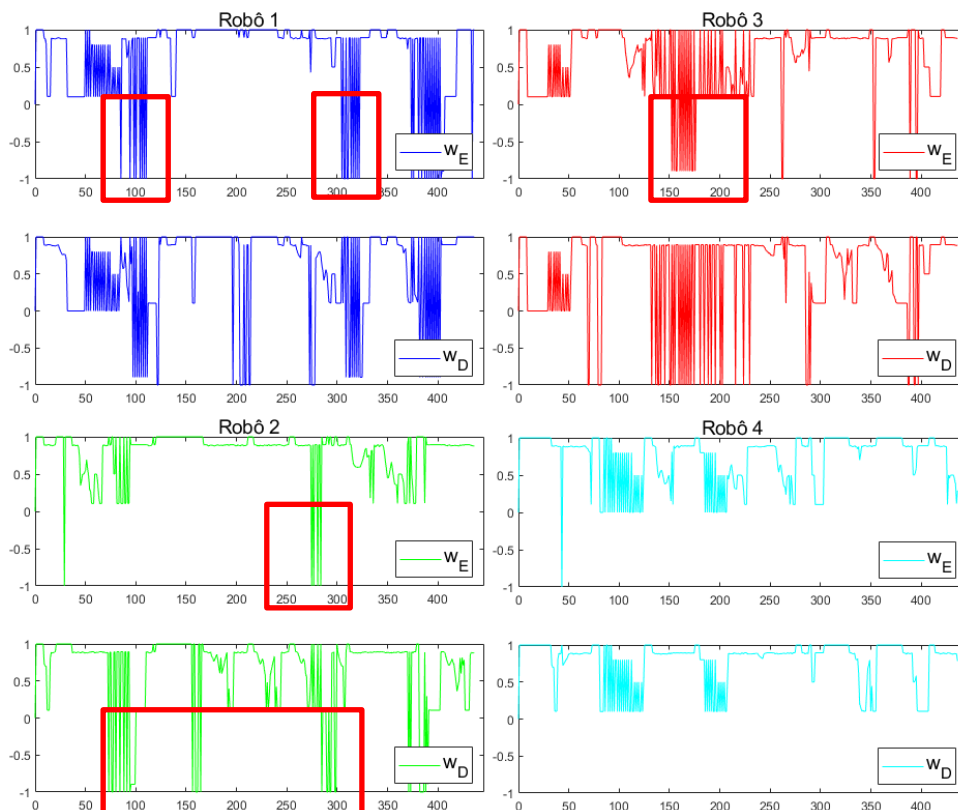
Para o segundo ambiente, o SMR – com exceção os robôs 4 e 5 – enfrentou dificuldades para explorar a seção superior, como visto na Figuras 4.44 a 4.50, respectivamente para FLC, DFCM e DFCM-ACO. Esse fato pode ser explicado pela concentração massiva de robôs nesta seção alinhada com o subcomportamento de evitar obstáculos, que considera uma distância de segurança de 5 cm, limitando a movimentação como visto principalmente na Figura 4.44 (a).

Figura 4.44 – Trajeto de oito robôs do ambiente II: (a) FLC, (b) DFCM e (c) DFCM-ACO



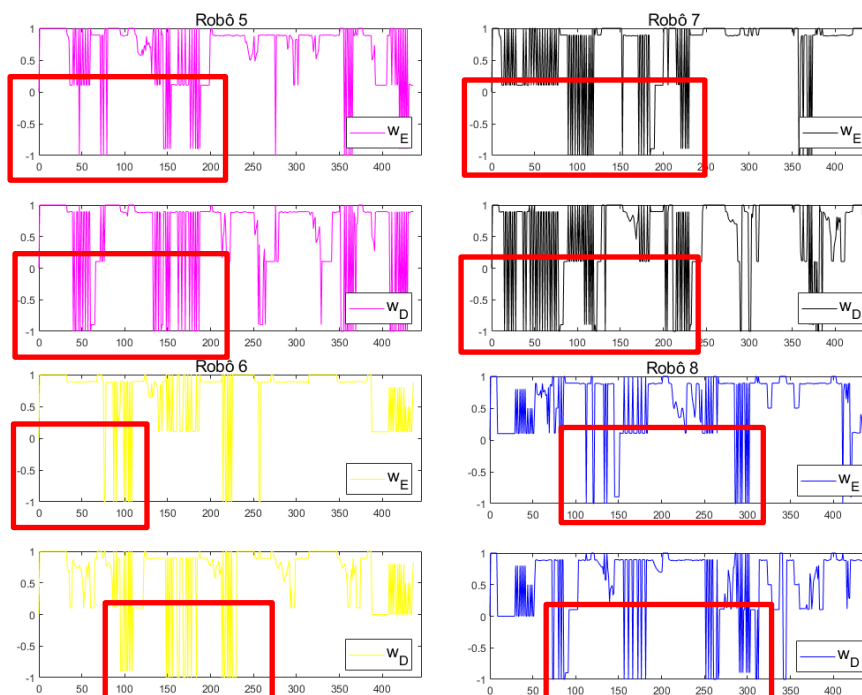
Fonte: Autoria própria.

Figura 4.45 – Pulsos de oito robôs com FLC: ambiente II, robôs 1 a 4



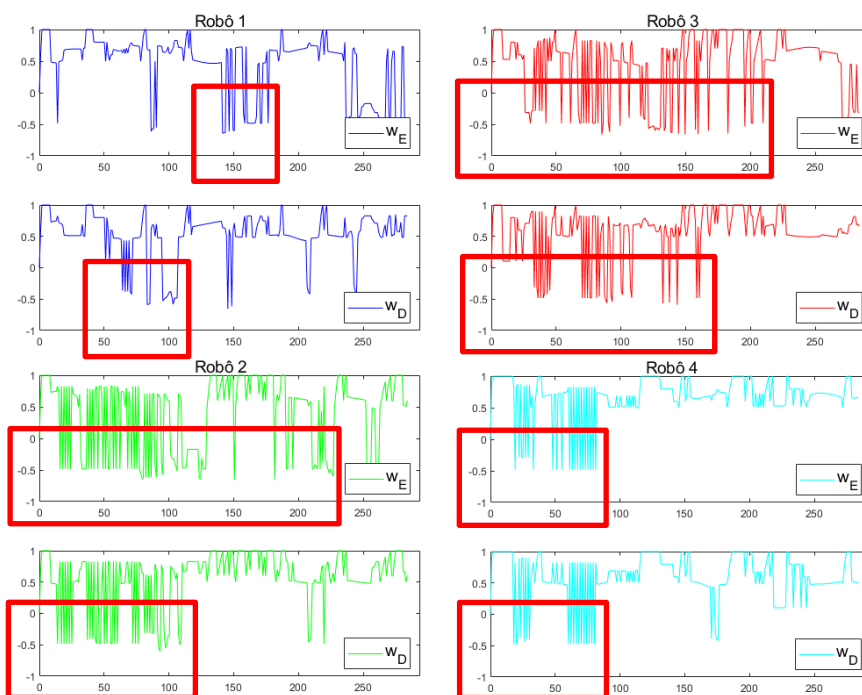
Fonte: Autoria própria.

Figura 4.46 – Pulsos de oito robôs com FLC: ambiente II, robôs 5 a 8



Fonte: Autoria própria.

Figura 4.47 – Pulsos de oito robôs com DFCM: ambiente II, robôs 1 a 4

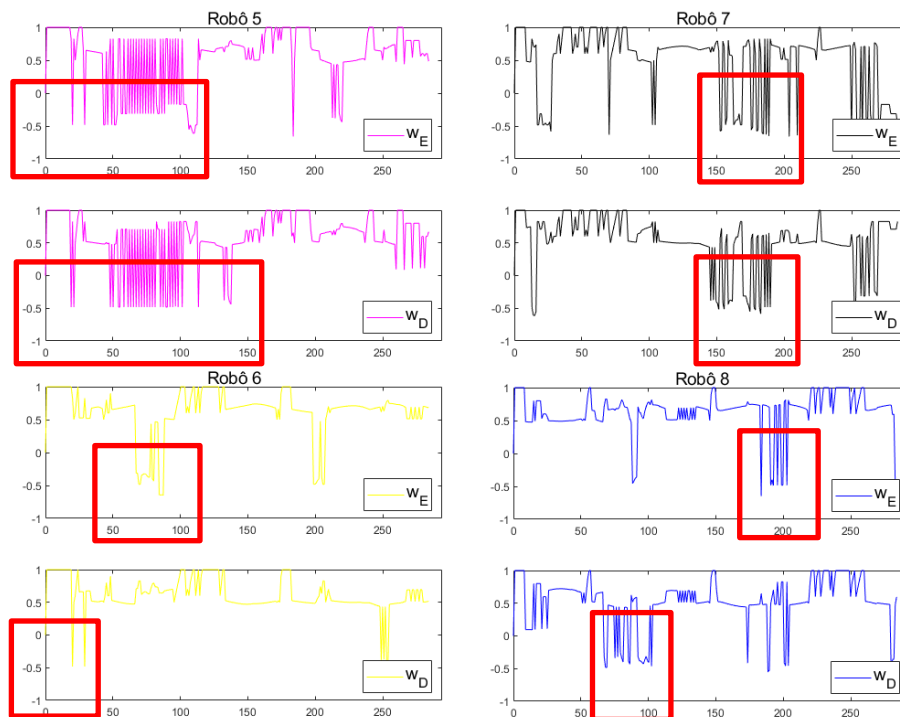


Fonte: Autoria própria.

O comportamento do SMR descrito acima ilustra um cenário de estresse, no qual o número de robôs é maior que o número de vítimas (alvos). Isso ocorre pois, neste trabalho, provou-se que um número menor de robôs é capaz de resgatar as

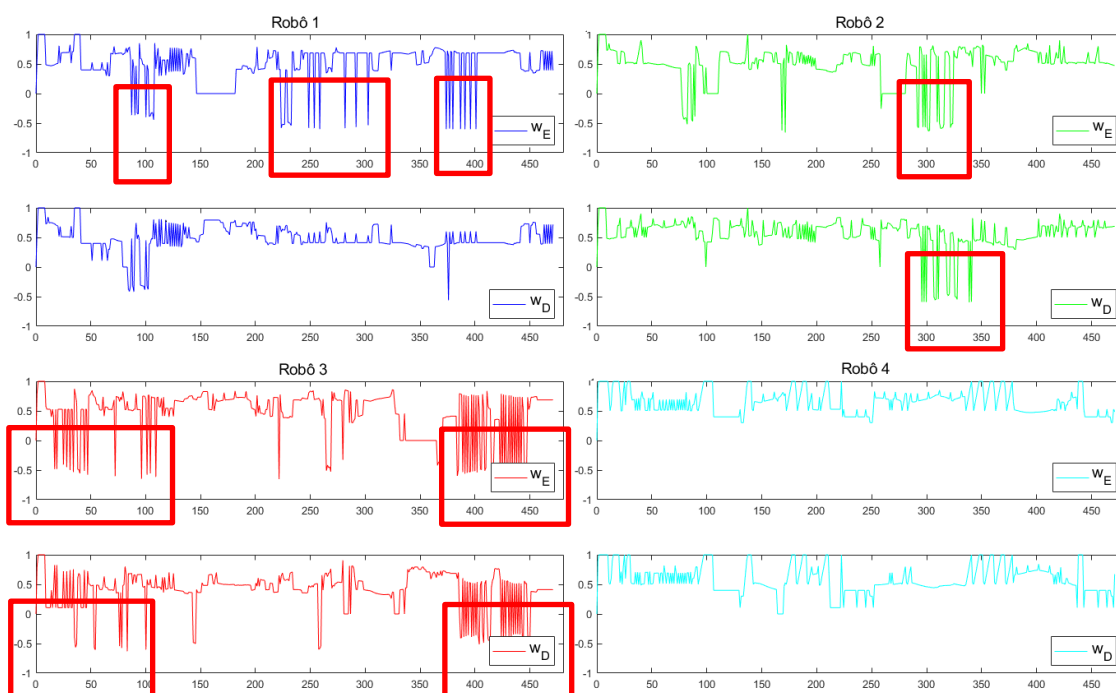
vítimas. No entanto, no caso particular de resgate de vítimas, um número maior de robôs é justificado devido à necessidade de cumprir o objetivo o mais rápido possível, pois cada segundo é importante para preservar a vida das vítimas.

**Figura 4.48 – Pulsos de oito robôs com DFCM: ambiente II, robôs 5 a 8**



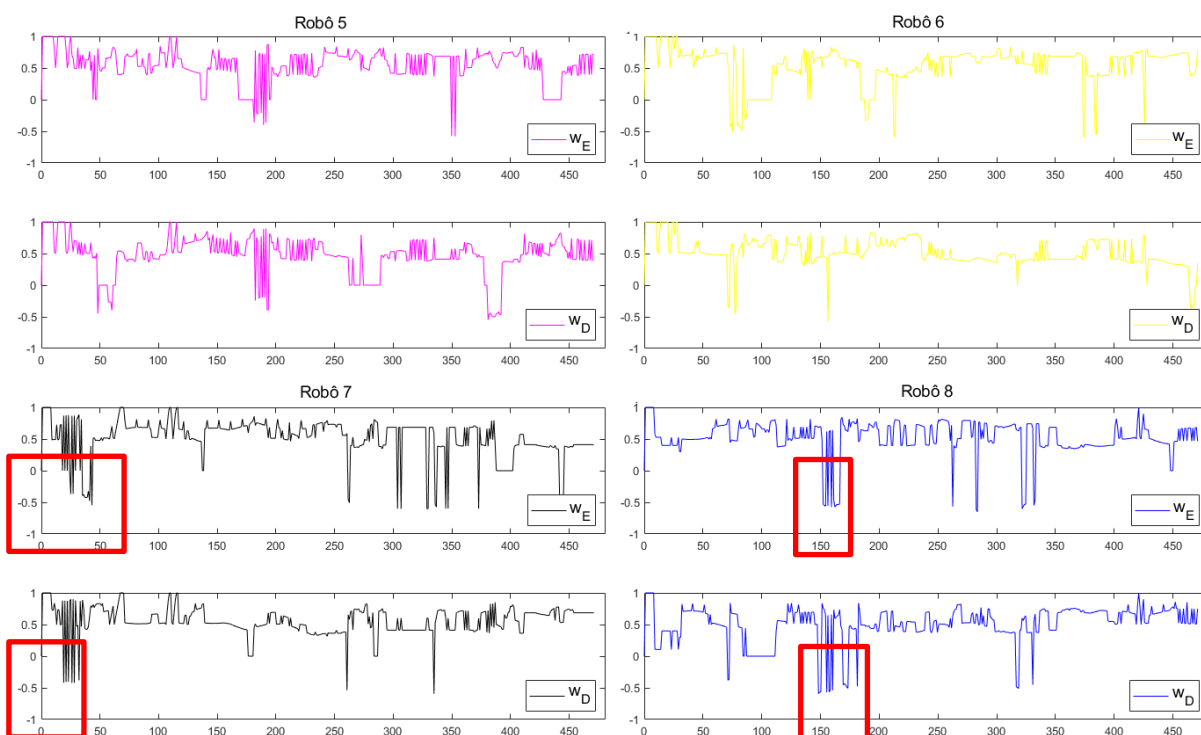
Fonte: Autoria própria.

**Figura 4.49 – Pulsos de oito robôs com DFCM-ACO: ambiente II, robôs 1 a 4**



Fonte: Autoria própria.

**Figura 4.50 – Pulsos de oito robôs com DFCM-ACO: ambiente II, robôs 5 a 8**

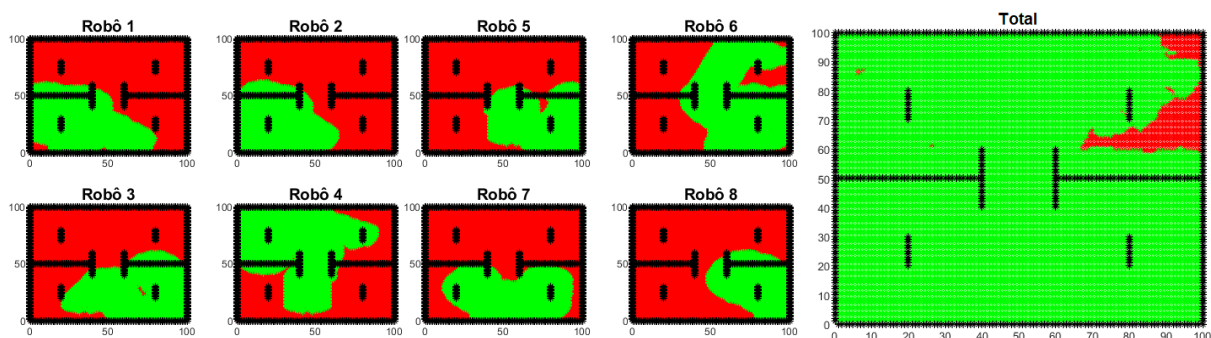


Fonte: Autoria própria.

Ao comparar as Figuras 4.45 e 4.46 (relacionadas ao FLC) com as Figuras 4.47 e 4.48 (DFCM) e, posteriormente, 4.49 e 4.50 (DFCM-ACO), tem-se que as duas primeiras apresentam consideravelmente mais pulsos negativos (subcomportamento de iminência de colisão do que as demais, diminuindo esse número nas mudanças de abordagem).

Como pode ser visto nas Figuras 4.51, 4.52 e 4.53, respectivamente para FLC, DFCM e DFCM-ACO, apenas dois robôs exploraram a seção superior do segundo ambiente. Neste caso, os robôs 4 e 6 para o FLC, 4 e 5 para DFCM e 2 e 4 para o DFCM-ACO).

**Figura 4.51 – Áreas individuais e total exploradas pelo SMR com FLC: ambiente II, 8 robôs**

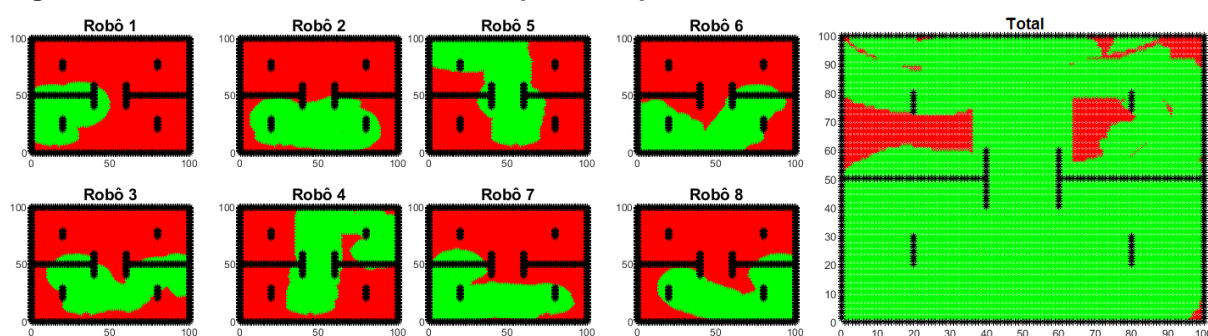


Fonte: Autoria própria.



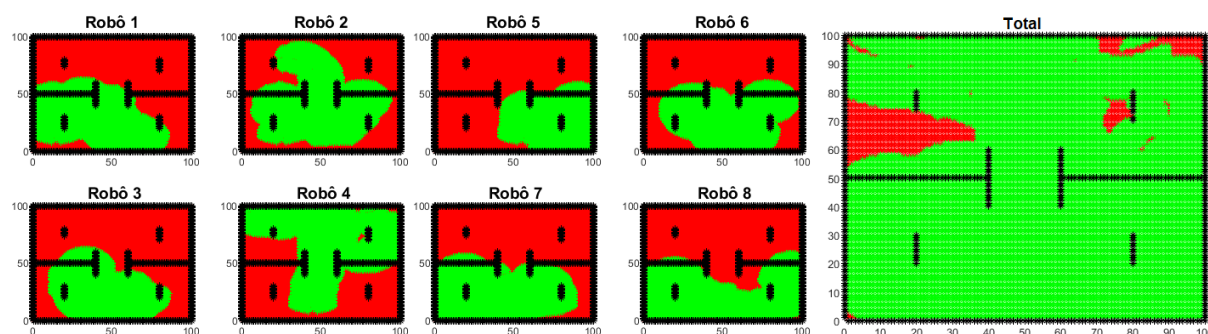
Ademais, a detecção dos sensores através das paredes do ambiente – robôs 1, 3, 6, 7 e 8 – foi causada pelo ruído branco inserido nos sensores ultrassônicos e não demonstrou nenhum comportamento emergente. Pela área explorada total da Figura 4.52, nota-se mais uma vez que as regiões inexploradas são resultado do critério de parada de resgate das vítimas, e são similares para DFCM e DFCM-ACO nesse cenário.

**Figura 4.52 – Áreas individuais e total exploradas pelo SMR com DFCM: ambiente II, 8 robôs**



Fonte: Autoria própria.

**Figura 4.53 – Áreas individuais e total exploradas pelo SMR com DFCM-ACO: ambiente II, 8 robôs**



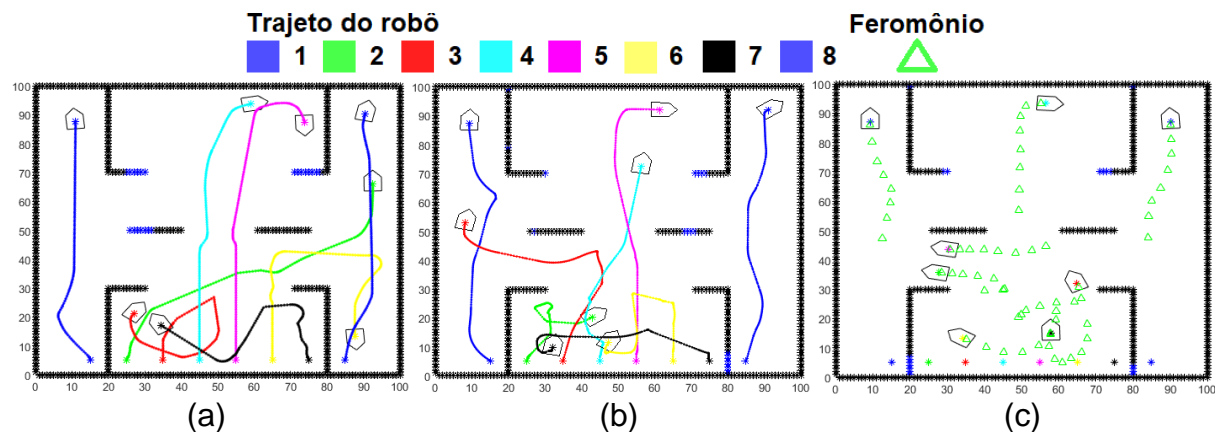
Fonte: Autoria própria.

A configuração de oito robôs no SMR foi a mais bem-sucedida no terceiro ambiente em termos das trilhas deixadas pelos robôs. Nesse cenário, as áreas laterais não ofereceram dificuldades devido ao número de robôs no ambiente, como mostra a Figura 4.54 (a), (b) e (c), respectivamente para FLC, DFCM e DFCM-ACO.

Ao comparar os pulsos enviados as rodas das Figuras 4.55 a 4.60, cabe ressaltar que os robôs 2, 3, 4, 5, 6 e 7 apresentaram o comportamento de colisão iminente durante todo o tempo de simulação para o terceiro ambiente na abordagem FLC (Figuras 4.55 e 4.56). Do mesmo modo, os robôs 2, 3, 6 e 7 apresentaram este comportamento tanto para o DFCM quanto para o DFCM-ACO. No entanto, conforme

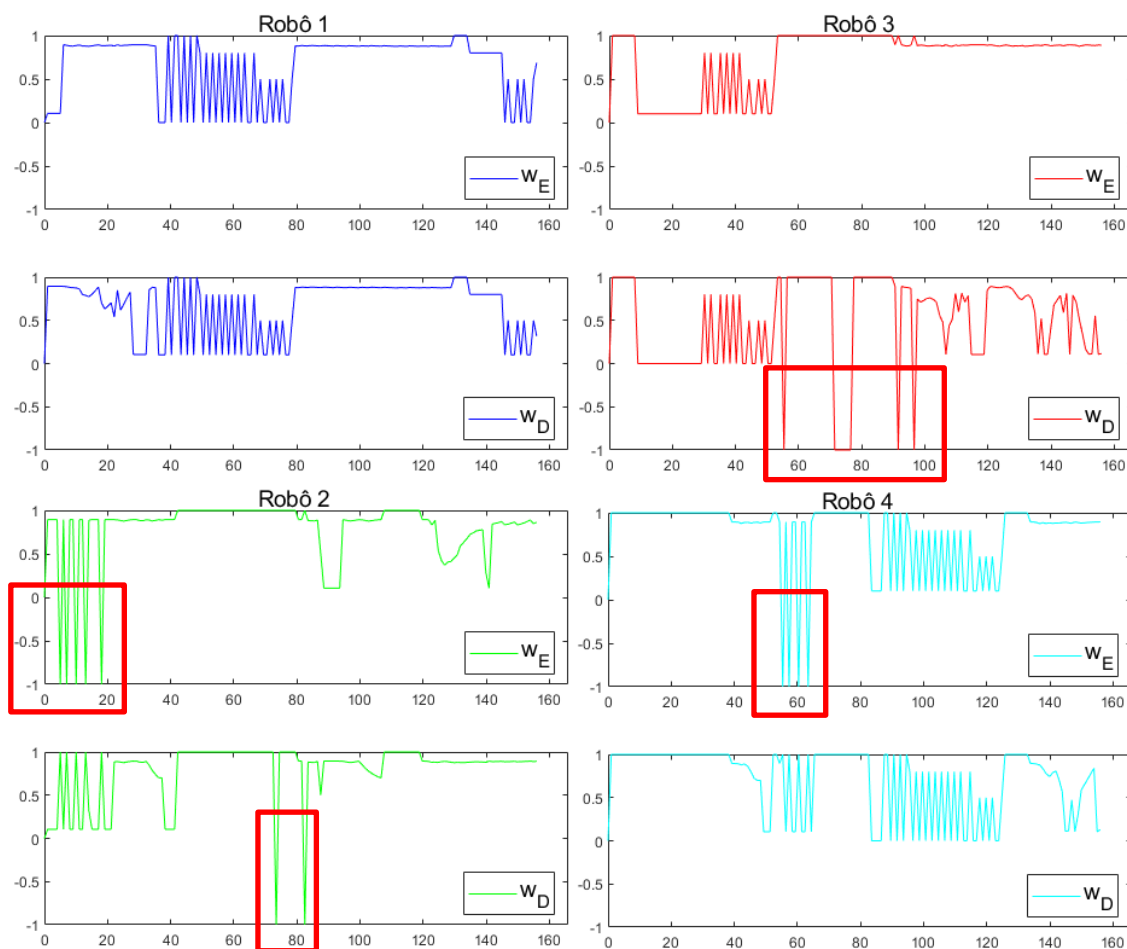
visto na Figura 4.55, esse subcomportamento é menos ativado quando o FLC é utilizado no SMR. Novamente, nas Figuras 4.55 a 4.60 foram destacados pontos críticos nos quais o subcomportamento de iminência de colisão é acionado.

Figura 4.54 – Trajeto de oito robôs do ambiente III: (a) FLC, (b) DFCM e (c) DFCM-ACO



Fonte: Autoria própria.

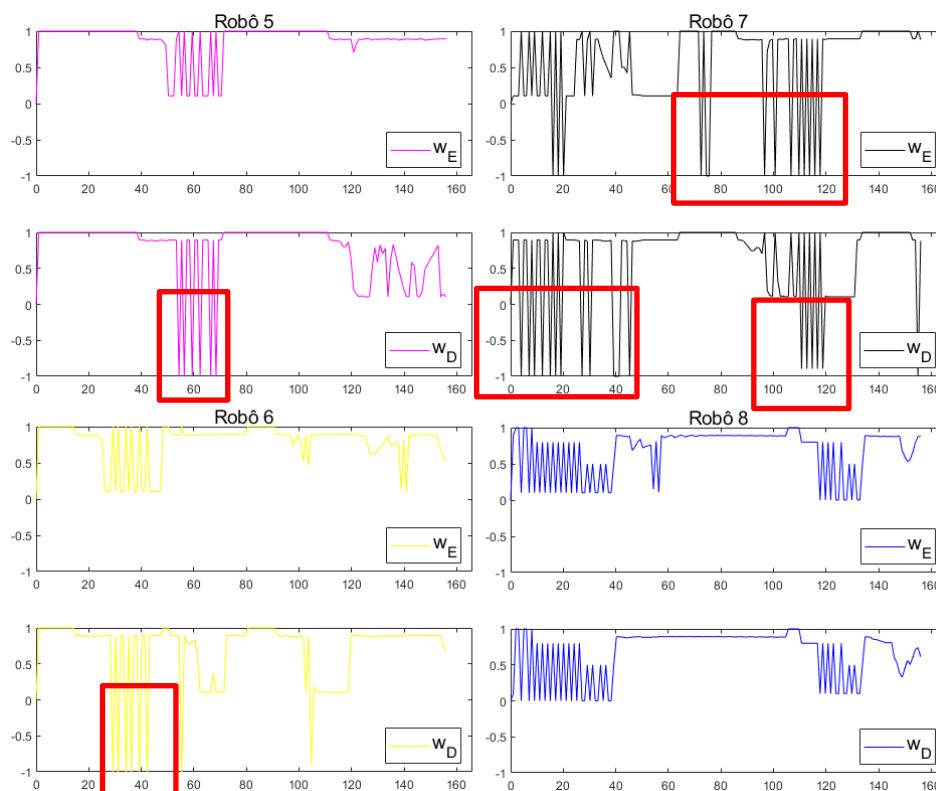
Figura 4.55 – Pulsos de oito robôs com FLC: ambiente III, robôs 1 a 4



Fonte: Autoria própria.

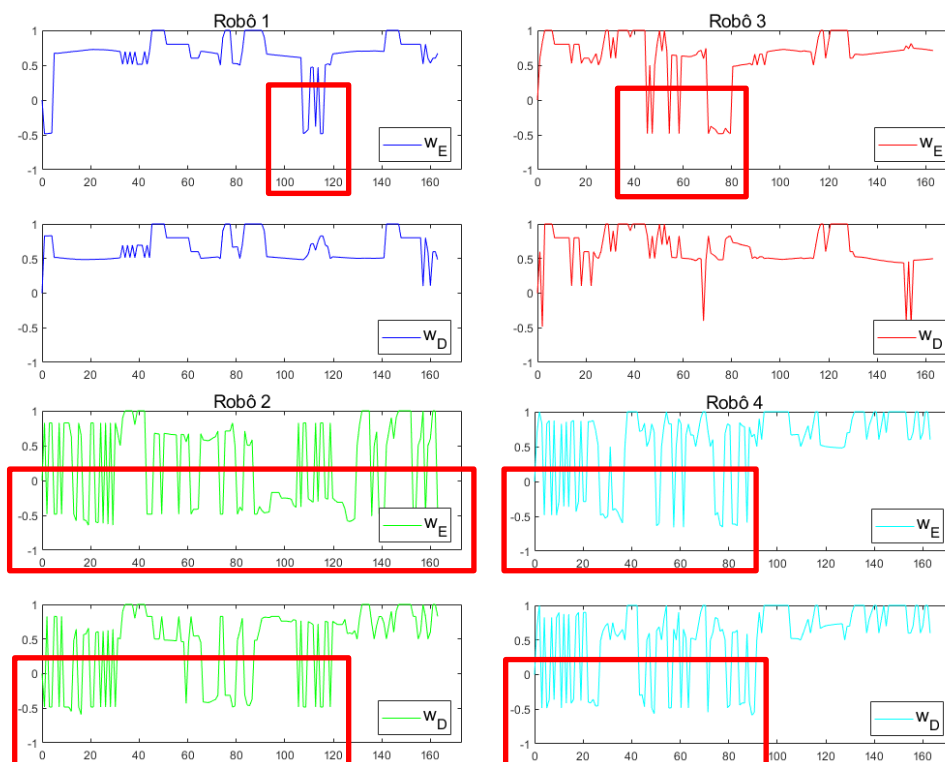


Figura 4.56 – Pulsos de oito robôs com FLC: ambiente III, robôs 5 a 8



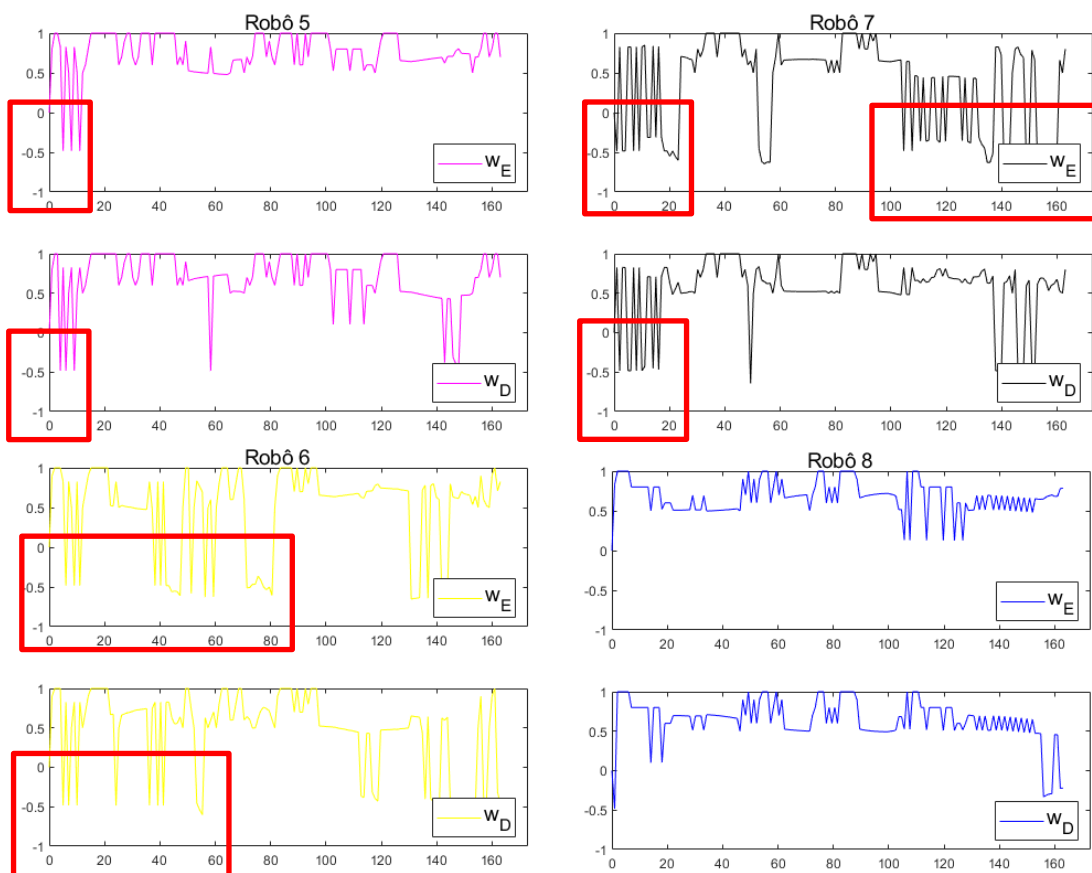
Fonte: Autoria própria.

Figura 4.57 – Pulsos de oito robôs com DFCM: ambiente III, robôs 1 a 4



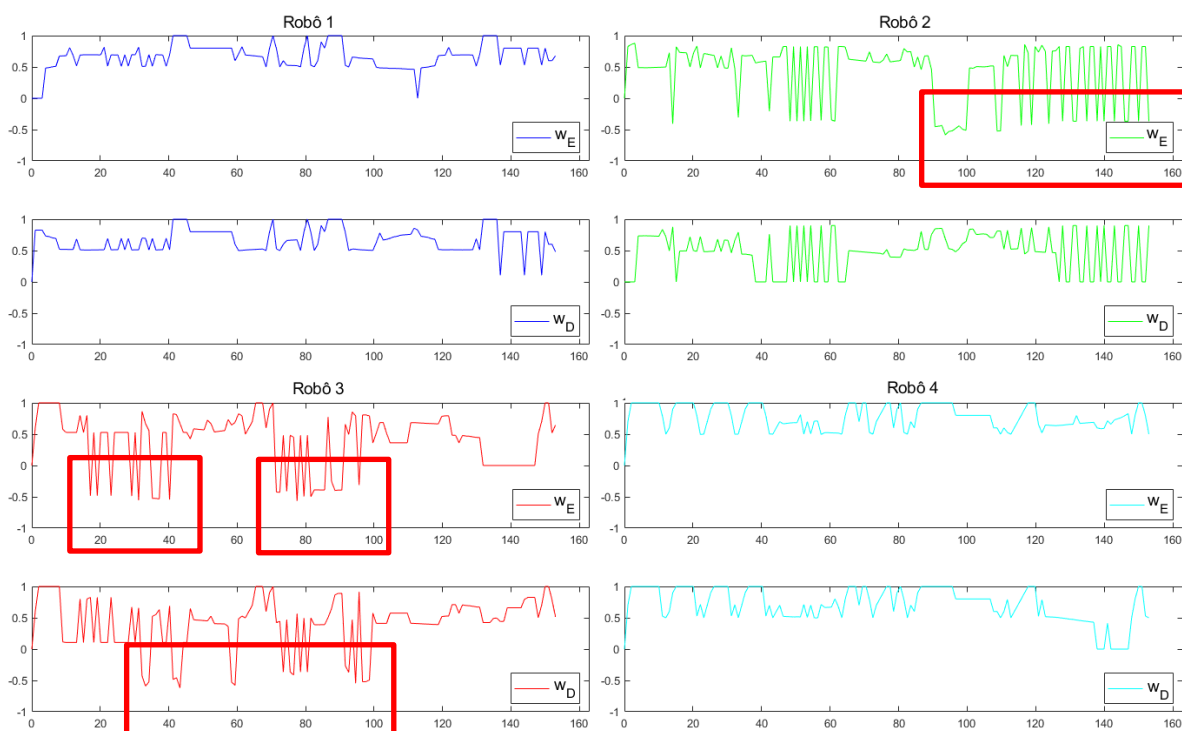
Fonte: Autoria própria.

Figura 4.58 – Pulsos de oito robôs com DFCM: ambiente III, robôs 5 a 8



Fonte: Autoria própria.

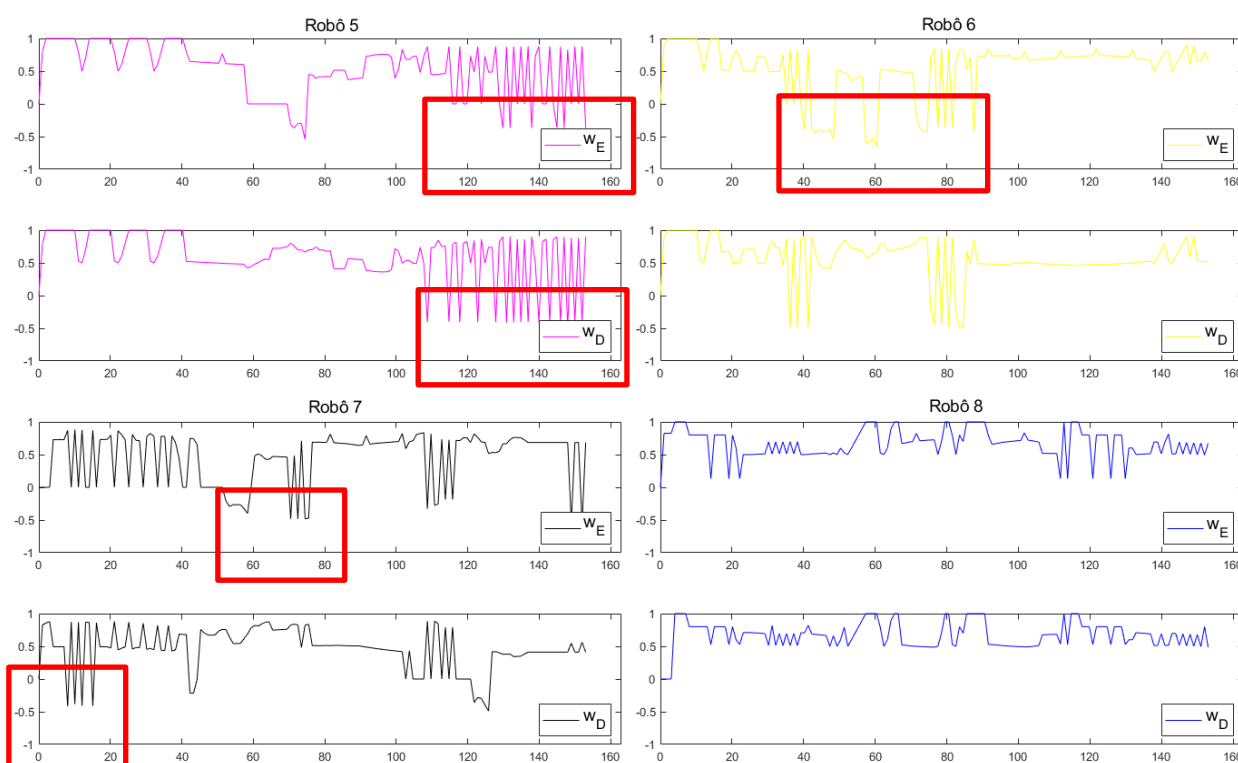
Figura 4.59 – Pulsos de oito robôs com DFCM-ACO: ambiente III, robôs 1 a 4



Fonte: Autoria própria.

Neste terceiro ambiente, nota-se pelas Figuras 4.57 e 4.58 que, em comparação com FLC e DFCM-ACO, o controlador DFCM enfrentou consideravelmente mais situações de iminência de colisão entre os robôs (principalmente entre os quatro primeiros). Esse fato vai na contramão do esperado de acordo com os experimentos simulados anteriores, nos quais, em geral, o FLC apresentava este comportamento mais vezes. Entretanto, quando o DFCM-ACO é implementado a ocorrência dessa situação é amenizada, conforme destacado nas Figuras 4.59 e 4.60.

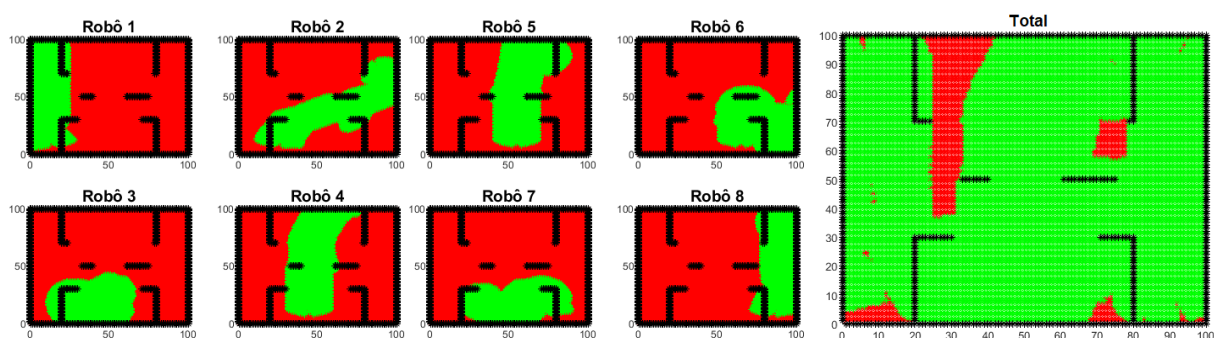
**Figura 4.60 – Pulsos de oito robôs com DFCM-ACO: ambiente III, robôs 5 a 8**



Fonte: Autoria própria.

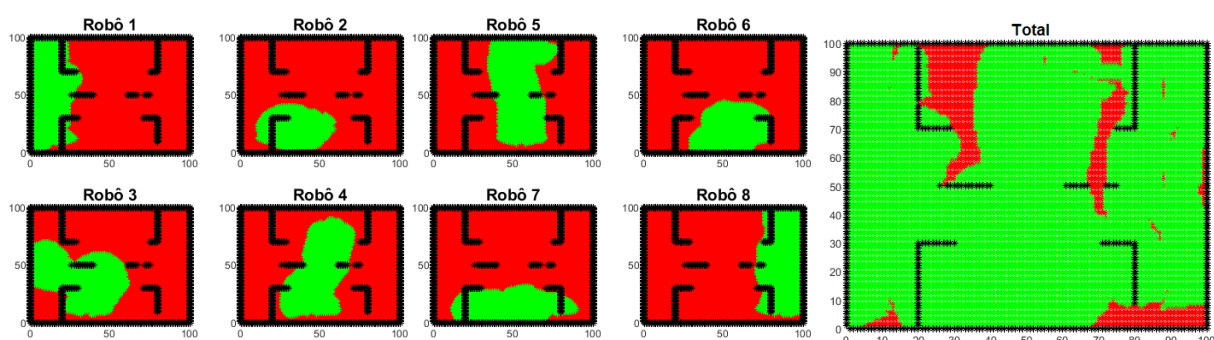
A área explorada por cada robô é mostrada nas Figuras 4.61 (FLC), 4.62 (DFCM) e 4.63 (DFCM-ACO). Os robôs controlados por DFCM exploraram 87,87% do ambiente III, em comparação com 89,94% da abordagem FLC e 85,57% do DFCM-ACO, percorrendo aproximadamente 115 cm a menos que o FLC e aproximadamente 20 cm a mais do que o DFCM-ACO. Ademais, as regiões inexploradas poderiam ser cobertas, ou seja, não foram identificadas regiões inacessíveis em nenhum dos ambientes.

Figura 4.61 – Áreas individuais e total exploradas pelo SMR com FLC: ambiente III, 8 robôs



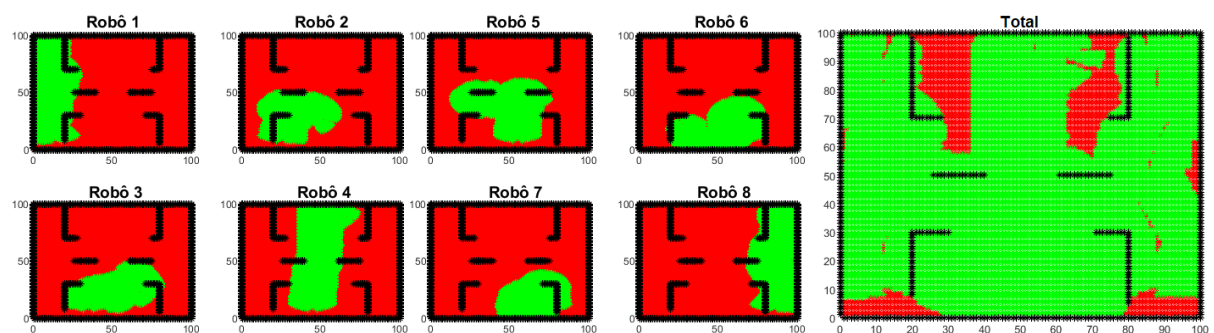
Fonte: Autoria própria.

Figura 4.62 – Áreas individuais e total exploradas pelo SMR com DFCM: ambiente III, 8 robôs



Fonte: Autoria própria.

Figura 4.63 – Áreas individuais e total exploradas pelo SMR com DFCM-ACO: ambiente III, 8 robôs



Fonte: Autoria própria.

Para o cenário de oito robôs, como visto na Tabela 4.5, em todos os ambientes a estratégia DFCM consumiu menos tempo de processamento. Essa característica sugere que, com o aumento do número de robôs, a implementação da abordagem DFCM continuará consumindo menos poder de processamento (TP) sem prejudicar a vida útil da bateria, conforme observado pelo número de iterações.

Apesar de consumirem mais poder de processamento computacional, tanto FLC quanto DFCM-ACO exploram mais área do que o DFCM no geral, com o FLC perdendo no ambiente I e o DFCM-ACO apenas no terceiro ambiente.

**Tabela 4.5 – Número de iterações e tempo de processamento para 8 robôs**

8 Robôs	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO
<b>Iterações</b>	355,00	400,00	634,00	435,00	283,00	471,00	155,00	162,00	153,00
<b>TP (s)</b>	1204,04	647,99	739,94	976,84	493,59	621,39	487,63	297,41	153,00

Fonte: Autoria própria.

**Tabela 4.6 – Área explorada para 8 robôs**

Área expl. (cm <sup>2</sup> )	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO
<b>Robô 1</b>	3937,00	4574,00	4841,00	3352,00	1877,00	3745,00	2157,00	2280,00	2189,00
<b>Robô 2</b>	3737,00	5146,00	5878,00	2990,00	2774,00	4656,00	2541,00	1310,00	1573,00
<b>Robô 3</b>	4830,00	4083,00	3668,00	3282,00	2793,00	3183,00	1860,00	2348,00	1579,00
<b>Robô 4</b>	3884,00	4556,00	6507,00	4896,00	3739,00	5055,00	2518,00	2240,00	2510,00
<b>Robô 5</b>	3535,00	4249,00	4740,00	2850,00	3520,00	2793,00	2824,00	2622,00	2087,00
<b>Robô 6</b>	3765,00	3855,00	3898,00	4886,00	3152,00	3777,00	2224,00	1641,00	1850,00
<b>Robô 7</b>	4460,00	4059,00	4169,00	3046,00	3135,00	3970,00	2013,00	1746,00	1421,00
<b>Robô 8</b>	4109,00	3517,00	6093,00	2287,00	3059,00	4210,00	2069,00	2293,00	2273,00
<b>Total</b>	9757,00	9840,00	9973,00	9475,00	8896,00	9174,00	8994,00	8787,00	8557,00

Fonte: Autoria própria.

As três abordagens do SMR apresentaram resultados semelhantes em relação à área explorada, como pode ser visto na Tabela 4.6. Assim, a razão entre a área explorada o tempo de processamento do DFCM é significativamente maior do que a obtida pelo FLC. No ambiente I, por exemplo, essa proporção é de aproximadamente o dobro.

A partir da Tabela 4.7, observa-se que os robôs do SMR por DFCM percorreram menores distâncias em todos os ambientes, mesmo apresentando área total explorada similar à do SMR por FLC. Para o DFCM-ACO a distância percorrida

foi maior que a do DFCM e, no geral, menor que a do FLC. Esse fato sugere novamente uma autonomia estendida da bateria para as estratégias baseadas em DFCM e DFCM-ACO.

**Tabela 4.7 – Distância percorrida para 8 robôs**

Distância percorrida (cm)	Ambiente I			Ambiente II			Ambiente III		
	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO	FLC	DFCM	DFCM-ACO
<b>Robô 1</b>	211,53	218,72	269,81	222,87	113,71	189,07	85,36	86,90	84,55
<b>Robô 2</b>	198,22	214,95	255,35	257,74	111,95	206,44	100,90	46,07	56,88
<b>Robô 3</b>	237,75	190,93	188,91	210,65	118,89	166,61	76,81	83,89	51,28
<b>Robô 4</b>	150,93	192,45	282,97	272,00	148,19	242,76	93,02	72,06	93,87
<b>Robô 5</b>	208,54	183,22	271,36	212,78	129,99	200,50	104,08	96,85	70,88
<b>Robô 6</b>	198,37	161,22	281,13	237,39	145,18	206,72	94,07	62,69	69,74
<b>Robô 7</b>	231,70	197,37	299,89	212,61	116,56	210,14	73,01	61,47	58,12
<b>Robô 8</b>	191,13	197,48	298,65	200,31	137,76	202,51	86,88	89,38	85,21
<b>Total</b>	1628,16	1556,34	1849,42	1826,35	1022,23	1624,75	714,13	599,31	570,53

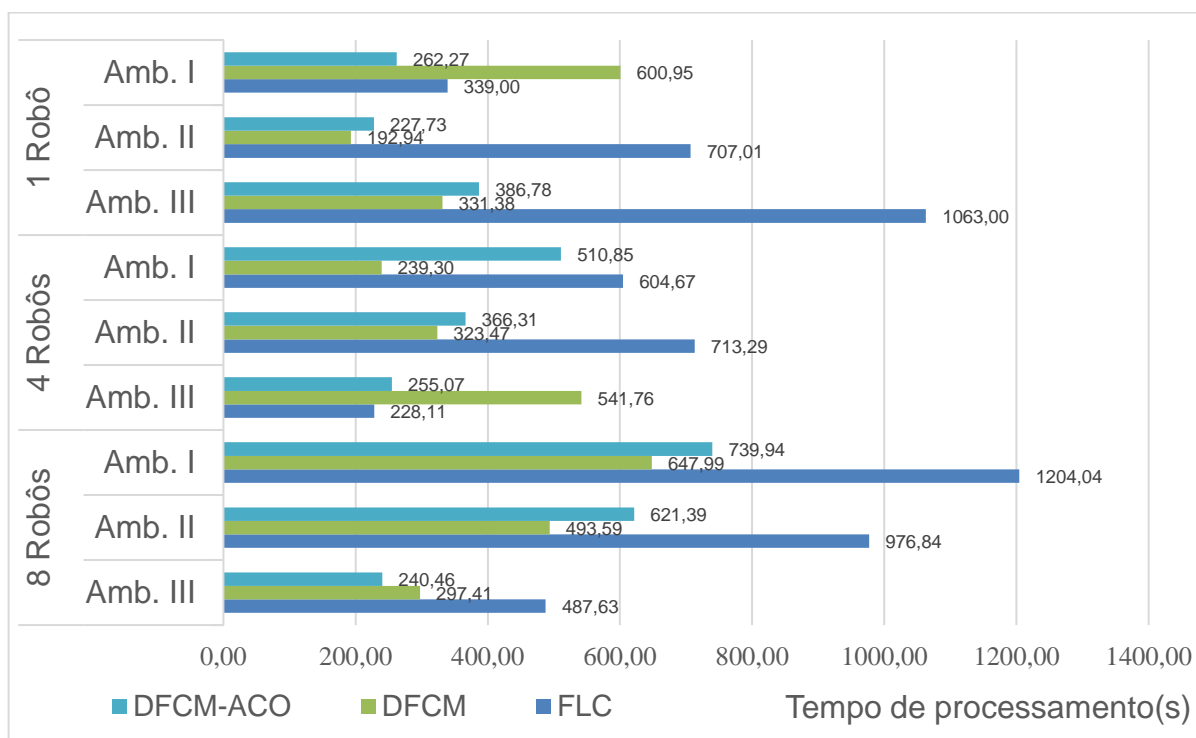
Fonte: Autoria própria.

Uma visão geral dos resultados dos tempos de processamento dos controladores FLC e DFCM propostos para o SMR é apresentada na Figura 4.64. Os resultados do terceiro ambiente para o um robô com FLC justificam a proposta do trabalho, uma vez que nesse cenário não foi possível encontrar todas as vítimas.

Considerando os cenários testados, a estratégia DFCM consumiu apenas mais tempo de processamento que o FLC em duas ocasiões de nove possíveis, enquanto que o DFCM-ACO apresentou resultados de tempo localizados entre as demais estratégias, com exceção do ambiente I com um robô e no ambiente III com 4 e oito robôs. No restante, supera o FLC principalmente ao usar oito robôs. Esse aspecto sugere que a abordagem DFCM é melhor escalável do que o FLC. Em outras palavras, sua implementação é simplificada em SMRs maiores ou mesmo em enxames

de robôs. Cabe ressaltar que para a configuração de um robô, o FLC não foi capaz de resgatar todas as vítimas no terceiro ambiente.

**Figura 4.64 – Comparativo gráfico entre os tempos de processamento**



Fonte: Autoria própria.

Por fim, no Apêndice B são mostrados os aspectos do protótipo inicial deste trabalho, ainda em construção, conforme observado na Figura B.1, além de sua utilização em alguns experimentos práticos. A plataforma (chassis) do robô possui quatro rodas, com o mesmo modelo cinemático do modelo simulado. O microcontrolador utilizado é um Arduino® Mega 2560. Ademais, são utilizados três sensores HC-SR04 ultrassônicos, um sensor de cor para identificação de alvos no chão (TCS3200), um módulo de GPS GY-NEO6MV2 para enviar a localização dos alvos localizados, dois motores DC de 6V (redução de 1:48) com interface de controle baseada em duas Pontes H (driver VNH2SP30), e uma bateria de 24V.

Ainda se tem a utilização de dois sensores laser de tempo de voo (do inglês *time-of-flight*) VL53L0X para auxiliar o cálculo da distância a objetos juntamente com os sensores ultrassônicos. A vantagem de sua utilização se dá pela maior precisão desse tipo de sensor em distâncias menores, o que pode auxiliar no comportamento de desvio de obstáculos.

## 5 CONSIDERAÇÕES FINAIS

Neste trabalho são investigados problemas de exploração autônoma em diferentes cenários para um sistema multirrobôs, utilizando uma abordagem baseada na arquitetura de subsunção e na robótica de enxame. Para isso, foram testadas duas estratégias baseadas em lógica *Fuzzy* para controle dos robôs, a primeira por um FLC e a segunda utilizando um DFCM, que posteriormente foi ampliado para a estratégia DFCM-ACO.

Nos três cenários simulados, o grupo de robôs deve realizar a tarefa cooperativa de resgatar todas as vítimas, devendo ser capazes de desviar de obstáculos estáticos e dinâmicos. Nesses cenários não há informação dos obstáculos de cada cenário. Dessa forma, os robôs são guiados exclusivamente por três sensores ultrassônicos localizados na parte frontal de sua estrutura.

Após a implementação dos controladores propostos para sistemas de um, quatro e oito robôs em condições equivalentes (quantidade de vítimas e cenários), realizou-se a comparação das estratégias levando em consideração fatores como tempo de processamento dos algoritmos, número de iterações, distância percorrida e área explorada pelos robôs. Dessa forma, foram explanadas as características de cada método, suas vantagens e desvantagens.

Os resultados observados mostraram flexibilidade na comunicação entre os robôs e a autonomia na exploração e resgate das vítimas nos cenários. Esses aspectos sugerem que o controle DFCM pode ser utilizado com sucesso em robôs autônomos, uma vez que esse controlador apresentou resultados otimizados em comparação ao FLC. Em outras palavras, no geral, os SMRs baseados em DFCM e DFCM-ACO resgataram todas as vítimas enquanto percorriam menores distâncias, cobrindo mais área. Contudo, o fato de não ter obtido os melhores resultados para todas as situações simuladas não descarta outros benefícios dessa estratégia.

Com isso, o desempenho computacional da abordagem DFCM, composto pelo número de iterações e tempo de processamento, tem clara vantagem nos cenários com sistema de oito robôs, o que se deve à sua escalabilidade aprimorada em comparação com o FLC. Portanto, conclui-se que os resultados obtidos neste trabalho vão ao encontro dos objetivos traçados inicialmente, uma vez os robôs são capazes de atuar autonomamente em ambientes distintos, capturar seus alvos e



desviar de obstáculos. Assim, a abordagem DFCM-ACO se caracteriza por um balanço entre as métricas de processamento computacional e as de exploração do ambiente, possuindo resultados satisfatórios que estão entre aqueles encontrados no FLC (exploração) e DFCM (processamento computacional).

Ainda, ressalta-se que a utilização de estratégias baseadas em DFCMs pode ser benéfica para aplicações de baixo custo financeiro, como microcontroladores PIC e Arduino, que possuem baixo poder de processamento computacional, além de microcomputadores como o Raspberry® Pi.

## 5.1 TRABALHOS GERADOS

A partir dos resultados obtidos preliminarmente nesta pesquisa foram gerados quatro artigos, dos quais dois apresentados no congresso internacional FUZZ-IEEE em 2018 (WCCI) e 2019: *“Group of Robots Inspired by Swarm Robotics Exploring Unknown Environments”* e *“Semi-Unknown Environments Exploration Inspired by Swarm Robotics using Fuzzy Cognitive Maps”*. Desses, o mais recente foi convidado para publicação na nova edição do livro referência em Mapas Cognitivos Fuzzy: *“Fuzzy Cognitive Maps for Applied Sciences and Engineering”*, da editora Elsevier.

Artigos foram submetidos recentemente ao periódico *IEEE Transactions on Fuzzy Systems* e nos congressos internacionais FUZZ-IEEE 2020 (WCCI) e *IEEE Conference on Evolving and Adaptive Intelligent Systems 2020 (IEEE EAIS)*.

## 5.2 TRABALHOS FUTUROS

Apesar das abordagens DFCM e DFCM-ACO terem cumprido satisfatoriamente todos os objetivos, trabalha-se para a implementação de um algoritmo de planejamento de trajetória online inspirado no método de Otimização por Colônia de Formigas (ACO). Essa estratégia utilizará a trilha deixada pelos robôs como um feromônio repulsivo para os outros durante o tempo desejado. Assim,

espera-se que essa estratégia expanda a área explorada, resultando, por exemplo, em menor consumo de bateria em aplicações reais e que os robôs não fiquem presos a um mesmo caminho por muito tempo.

Outro objetivo para trabalhos futuros é a implementação de um grupo maior de robôs e cenários diferentes, além da validação por meio de mais experiências com o protótipo modificado. Além disso, simulações buscando comportamentos emergentes nos robôs e testando a possibilidade de falha em um ou mais robôs para verificar a robustez do grupo. Por fim, será estudada a adição de comportamentos colaborativos e a utilização do conceito de líder para o grupo de robôs nos experimentos simulados, assim como funções para gerenciamento de bateria e a investigação de diferentes estratégias para o enxame, como a utilização de um líder teleguiado ou robôs com pernas (para transposição de obstáculos mais fácil em ambientes irregulares).

Para os experimentos práticos, uma das necessidades é a realização de otimizações tanto de hardware quando de software no projeto, visando a minimização de falhas e mal funcionamentos. Além disso, deve-se buscar outra solução para a localização por meio de GPS, uma vez que o módulo utilizado não foi capaz de funcionar em ambientes cobertos, como destacado no Apêndice B.

## REFERÊNCIAS

- ACAMPORA, G.; LOIA, V.; VITIELLO, A. Distributing emotional services in Ambient Intelligence through cognitive agents. **Service Oriented Computing and Applications**, v. 5, n. 1, p. 17–35, 22 mar. 2011.
- ACAMPORA, G.; PEDRYCZ, W.; VITIELLO, A. A Competent Memetic Algorithm for Learning Fuzzy Cognitive Maps. **IEEE Transactions on Fuzzy Systems**, v. 23, n. 6, p. 2397–2411, 2015.
- AHMED, M. T.; HEMAYED, E. E.; FARAG, A. A. **Neurocalibration: a neural network that can tell camera calibration parameters**. Proceedings of the Seventh IEEE International Conference on Computer Vision. **Anais...Kerkyra, Greece: IEEE, 1999.**
- ALMEIDA, J. P. L. S. DE et al. Bio-inspired on-line path planner for cooperative exploration of unknown environment by a Multi-Robot System. **Robotics and Autonomous Systems**, v. 112, p. 32–48, 2019.
- AMIRKHANI, A. et al. **Fuzzy cognitive map for visual servoing of flying robot**. 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016. **Anais...2016.**
- AMIRKHANI, A. et al. A Review of Fuzzy Cognitive Maps in Medicine: Taxonomy, Methods, and Applications. **Computer Methods and Programs in Biomedicine**, v. 142, p. 129–145, 2017.
- ANEZAKIS, V. et al. **Fuzzy Cognitive Maps for Long-Term Prognosis of the Evolution of Atmospheric Pollution, Based on Climate Change Scenarios: The Case of Athens**. (N. T. Nguyen et al., Eds.) International Conference on Computational Collective Intelligence ICCCI 2016. **Anais...Halkidiki, Greece: Springer, Cham, 2016.**
- ARRUDA, L. V. R. et al. Artificial Life Environment Modeled by Dynamic Fuzzy Cognitive Maps. **IEEE Transactions on Cognitive and Developmental Systems**, v. 10, n. 1, p. 88–101, 2018.
- AXELROD, R. **Structure of decisions: the cognitive maps of political elites**. 1. ed. Princeton, NJ, USA: Princeton University Press, 1976.
- BAKHSIPOUR, M.; GHADI, M. J.; NAMDARI, F. Swarm robotics search & rescue: a novel artificial intelligence-inspired optimization approach. **Applied Soft Computing**, v. 57, p. 708–726, 2017.
- BARCA, J. C.; SEKERCIOGLU, Y. A. Swarm robotics reviewed. **Robotica**, v. 31, n. 3, p. 345–359, 2013.
- BAYINDIR, L. A review of swarm robotics tasks. **Neurocomputing**, v. 172, p. 292–321, 2016.
- BAZAN, G. H. et al. Stator Short-Circuit Diagnosis in Induction Motors Using Mutual Information and Intelligent Systems. **IEEE Transactions on Industrial Electronics**, v. 66, n. 4, p. 3237–3246, 2019.
- BEHROOZ, F. et al. Review of control techniques for HVAC systems-nonlinearity

approaches based on fuzzy cognitive maps. **Energies**, v. 11, n. 3, p. 1–41, 2018.

BEN-ARI, M.; MONDADA, F. **Elements of Robotics**. 1. ed. Cham, Switzerland: Springer International Publishing, 2018.

BENI, G. **From Swarm Intelligence to Swarm Robotics**. (E. Şahin, W. M. Spears, Eds.) Swarm Robotics. SR 2004. Lecture Notes in Computer Science. **Anais...** Heidelberg: Springer Berlin Heidelberg, 2004.

BEVILACQUA, M.; CIARAPICA, F. E.; MAZZUTO, G. Fuzzy cognitive maps for adverse drug event risk management. **Safety Science**, v. 102, n. February 2018, p. 194–210, 2018.

BONABEAU, E.; DORIGO, M.; THERAULAZ, G. **Swarm Intelligence: From Natural to Artificial Systems**. 1. ed. New York, USA: Oxford University Press, 1999.

BOUTALIS, Y.; KOTTAS, T. L.; CHRISTODOULOU, M. Adaptive estimation of fuzzy cognitive maps with proven stability and parameter convergence. **IEEE Transactions on Fuzzy Systems**, v. 17, n. 4, p. 874–889, 2009.

BRAITENBERG, V. **Vehicles: Experiments in Synthetic Psychology**. 1. ed. Cambridge, USA: MIT Press, 1986.

BROOKS, R. A. A Robust Layered Control System For A Mobile Robot. **IEEE Journal on Robotics and Automation**, v. 2, n. 1, p. 14–23, 1986.

CALVO, R. et al. A Bioinspired Coordination Strategy for Controlling of Multiple Robots in Surveillance Tasks. **International Journal on Advances in Software**, v. 5, n. 3–4, p. 146–165, 2012.

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. **Anais...** Melbourne, Victoria, Australia: IEEE, 2001.

CBR. **LARC/CBR**. Disponível em: <<http://www.cbrobotica.org/>>. Acesso em: 1 nov. 2019.

COLARES, R. G.; CHAIMOWICZ, L. **A Novel Distance Cost Approach for Multi-robot Integrated Exploration**. 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). **Anais...** Uberlandia, Brazil: IEEE, 2015.

COLARES, R. G.; CHAIMOWICZ, L. **The Next Frontier: Combining Information Gain and Distance Cost for Decentralized Multi-Robot Exploration**. SAC '16 Proceedings of the 31st Annual ACM Symposium on Applied Computing. **Anais...** Pisa, Italy: ACM, 2016.

CORKE, P. **Robotics, Vision and Control. Fundamental Algorithms in MATLAB**. 1. ed. Heidelberg: Springer-Verlag Berlin Heidelberg, 2011. v. 73.

COUCEIRO, M. S. et al. Benchmark of swarm robotics distributed techniques in a search task. **Robotics and Autonomous Systems**, v. 62, n. 2, p. 200–213, 2014.

DA SILVA, I. N. et al. **Artificial neural networks: a practical course**. 1. ed. Cham,

Switzerland: Springer International Publishing, 2017.

DARPA. **DARPA Robotics Challenge (DRC)**. Disponível em: <<https://www.darpa.mil/program/darpa-robotics-challenge>>. Acesso em: 1 nov. 2019.

DARPA. **The Grand Challenge**. Disponível em: <<https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles>>. Acesso em: 1 nov. 2019.

DE RANGO, F. et al. Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks. **Soft Computing**, v. 22, n. 13, p. 4251–4266, 2018.

DIN, A. et al. Behavior-based swarm robotic search and rescue using fuzzy controller. **Computers & Electrical Engineering**, v. 70, p. 53–65, 2018.

DORIGO, M.; BIRATTARI, M.; STÜTZLE, T. Ant Colony Optimization. **IEEE Computational Intelligence Magazine**, v. 1, n. 4, p. 28–39, 2006.

DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. **Theoretical Computer Science**, v. 344, n. 2–3, p. 243–278, 2005.

DORIGO, M.; BONABEAU, E. **Swarm Intelligence: From Natural to Artificial Systems**. 1. ed. Oxford, United Kingdom: Oxford University Press, 1999.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: Optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 26, n. 1, p. 29–41, 1996.

DUNDON, E.; COURIER, H. **MoDOT Grapples With Issues Related to Vehicle Automation**. Disponível em: <<https://www.ttnews.com/articles/modot-grapples-issues-related-vehicle-automation>>. Acesso em: 3 out. 2019.

ELENI, V.; PETROS, G. **New Concerns on Fuzzy Cognitive Maps Equation and Sigmoid Function**. 2017 25th Mediterranean Conference on Control and Automation (MED). **Anais...**Valletta, Malta: 2017.

FIRST. **FIRST Robotics Competition Game & Season Info**. Disponível em: <<https://www.firstinspires.org/robotics/frc>>. Acesso em: 1 nov. 2019.

GAZI, V.; PASSINO, K. M. **Swarm stability and optimization**. 1. ed. Heidelberg: Springer Berlin Heidelberg, 2011.

HAMMAN, H. **Space-Time Continuous Models of Swarm Robotic Systems**. 1. ed. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2010.

HARMATI, I. Á.; KÓCZY, L. T. **Notes on the Dynamics of Hyperbolic Tangent Fuzzy Cognitive Maps**. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). **Anais...**New Orleans, USA: IEEE, 2019a.

HARMATI, I. Á.; KÓCZY, L. T. On the convergence of sigmoidal fuzzy grey cognitive maps. In: **International Journal of Applied Mathematics and Computer Science**. 1. ed. Zielona Góra, Poland: AMCS, 2019b. v. 29p. 453–466.

HATWÁGNER, M. F. et al. Banking Applications of FCM Models. In: CORNEJO, M. E. et al. (Eds.). **Trends in Mathematics and Computational Intelligence - Studies in**

**Computational Intelligence 796**. 1. ed. Cham, Switzerland: Springer International Publishing, 2019. p. 61–72.

HAYKIN, S. S. **Neural networks and learning machines**. 3. ed. Upper Saddle River: Pearson Prentice Hall, 2009.

HE, W. et al. Adaptive Neural Network Control for a Robotic Manipulator with Unknown Deadzone. **IEEE Transactions on Cybernetics**, v. PP, n. 99, p. 1–13, 2017.

HEXMOOR, H. **Essential Principles for Autonomous Robotics: Synthesis Lectures on Artificial Intelligence and Machine Learning**. 1. ed. San Rafael, CA, USA: Morgan & Claypool, 2013.

HSIEH, M. A.; CHIRIKJIAN, G. (EDS.). **Distributed Autonomous Robotic Systems**. 104. ed. London, England: Springer Heidelberg New York Dordrecht London, 2014.

IROBOT. **Roomba série 900**. Disponível em: <<https://www.irobot.com.br/roomba/900-series>>. Acesso em: 12 out. 2019.

KALIŃSKI, K. J.; MAZUR, M. Optimal control of 2-wheeled mobile robot at energy performance index. **Mechanical Systems and Signal Processing**, v. 70–71, n. 1, p. 373–386, 2016.

KARAGIANNIS, I.; GROUMPOS, P. Input-Sensitive Fuzzy Cognitive Maps. **International Journal of Computer Science**, v. 10, n. 3, p. 143–151, 2013.

KHAMIS, A.; HUSSEIN, A.; ELMOGY, A. Multi-robot Task Allocation: A Review of the State-of-the-Art. In: KOUBÂA, A.; MARTÍNEZ-DE DIOS, J. (Eds.). . **Cooperative Robots and Sensor Networks**. 2015. ed. Cham, Switzerland: Springer, 2014. v. 554p. 31–51.

KLIR, G. J. **Fuzzy Logic and Probability Applications: Bridging the Gap**. 1. ed. Alexandria, Virginia, USA: Society for Industrial and Applied Mathematics, Philadelphia, and American Statistical Association, 2007.

KORKMAZ, B. et al. **Fuzzy Logic Based Self-Driving Racing Car Control System**. 2018 6th International Conference on Control Engineering & Information Technology (CEIT). **Anais...**Istanbul, Turkey: IEEE, 2018.

KOSKO, B. Fuzzy cognitive maps. **International Journal of Man-Machine Studies**, v. 24, n. 1, p. 65–75, 1986.

KOULOURIOTIS, D. E.; DIAKOULAKIS, I. E.; EMIRIS, D. M. **Anamorphosis of fuzzy cognitive maps for operation in ambiguous and multi-stimulus real world environments**. 10th IEEE International Conference on Fuzzy Systems. **Anais...**Melbourne, Victoria, Australia: IEEE, 2001.

KUMOVA, B. Í.; HEYE, S. B. **A survey of robotic agent architectures**. 2017 International Artificial Intelligence and Data Processing Symposium (IDAP). **Anais...**Malatya, Turkey: IEEE, 2017.

LOPES, H. S.; RODRIGUES, L. C. DE A.; STEINER, M. T. A. (EDS.). **Meta-Heurísticas em Pesquisa Operacional**. 1. ed. Curitiba, PR, Brazil: Ominpax Editora, 2013.

MACIEL, R. **A Indústria 4.0 e a evolução dos negócios**. Disponível em: <<https://russellbedford.com.br/noticias/industria-4-0-evolucao-negocios/>>. Acesso em: 15 out. 2019.

MAES, P. Artificial life meets entertainment: lifelike autonomous agents. **Communications of the ACM**, v. 38, n. 11, p. 108–114, 1995.

MAGO, V. K. et al. Fuzzy cognitive maps and cellular automata: An evolutionary approach for social systems modelling. **Applied Soft Computing**, v. 12, n. 12, p. 3771–3784, 2012.

MAMDANI, E. H. Application of fuzzy algorithms for control of simple dynamic plant. **Proceedings of the Institution of Electrical Engineers**, v. 121, n. 12, p. 1585, 1974.

MARIANI, M.; BORGHI, M. Industry 4.0: A bibliometric review of its managerial intellectual structure and potential evolution in the service industries. **Technological Forecasting and Social Change**, v. 149, n. December 2019, p. 1–24, 2019.

MATARIC, M. J. **The Robotics Primer**. 1. ed. Cambridge, MA, USA: The MIT Press, 2007.

MCLURKIN, J. **Multi-Robot Systems Engineering**. Disponível em: <<https://people.csail.mit.edu/jamesm/images/swarm/images/theswarm.jpg>>. Acesso em: 2 out. 2019.

MENDONÇA, M. **A Contribution to Intelligent System Modeling based on Dynamic Cognitive Networks**. [s.l.] Federal University of Technology - Paraná, Curitiba, 2011.

MENDONÇA, M. et al. A dynamic fuzzy cognitive map applied to chemical process supervision. **Engineering Applications of Artificial Intelligence**, v. 26, n. 4, p. 1199–1210, 2013a.

MENDONÇA, M. et al. **Autonomous Navigation Applying Dynamic-Fuzzy Cognitive Maps and Fuzzy Logic**. (H. Papadopoulos et al., Eds.) Artificial Intelligence Applications and Innovations. AIAI 2013. **Anais...Berlin, Heidelberg**: Springer Berlin Heidelberg, 2013b.

MENDONÇA, M. et al. **Hybrid Dynamic Fuzzy Cognitive Maps Evolution for Autonomous Navigation System**. 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). **Anais...Istanbul, Turkey**: IEEE, 2015.

MENDONÇA, M. et al. **Hybrid Dynamic Fuzzy Cognitive Maps and Hierarchical Fuzzy Logic controllers for Autonomous Mobile Navigation**. 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016. **Anais...Vancouver, BC, Canada**: IEEE, 2016.

MENDONÇA, M. et al. A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps. **Engineering Applications of Artificial Intelligence**, v. 59, n. March, p. 122–132, 2017.

MENDONÇA, M. et al. **Semi-Unknown Environments Exploration Inspired by Swarm Robotics using Fuzzy Cognitive Maps**. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). **Anais...New Orleans, USA**: IEEE, 2019.

MENDONÇA, M.; ARRUDA, L. V. R.; NEVES-JR, F. Autonomous navigation system using Event Driven-Fuzzy Cognitive Maps. **Applied Intelligence**, v. 37, n. 2, p. 175–188, 2011.

MENDONÇA, M.; CHUN, I. R.; ROCHA, M. E. C. Dynamic fuzzy cognitive maps applied in realibility centered maintainance of electric motors. **IEEE Latin America Transactions**, v. 15, n. 5, p. 827–834, 2017.

MEYER, J. A. **Evolutionary approaches to neural control in mobile robots**. Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on. **Anais...**San Diego, CA, USA: 1998.

MOURHIR, A. et al. Exploring Precision Farming Scenarios Using Fuzzy Cognitive Maps. **Sustainability**, v. 9, n. 7, p. 1241, 2017.

MPELOGIANNI, V.; GROUMPOS, P. P. **A revised approach in modeling fuzzy cognitive maps**. 2016 24th Mediterranean Conference on Control and Automation (MED). **Anais...**Athens, Greece: IEEE, 2016.

MPELOGIANNI, V.; GROUMPOS, P. P. Increasing the Energy Efficiency of Buildings using Human Cognition; via Fuzzy Cognitive Maps. **IFAC-PapersOnLine**, v. 51, n. 30, p. 727–732, 2018a.

MPELOGIANNI, V.; GROUMPOS, P. P. Re-approaching fuzzy cognitive maps to increase the knowledge of a system. **AI & Society**, v. 33, n. 2, p. 175–188, 2018b.

NÁPOLES, G. et al. On the convergence of sigmoid Fuzzy Cognitive Maps. **Information Sciences**, v. 349–350, p. 154–171, 1 jul. 2016.

NÁPOLES, G.; BELLO, R.; VANHOOF, K. **Learning Stability Features on Sigmoid Fuzzy Cognitive Maps through a Swarm Intelligence Approach**. (J. Ruiz-Shulcloper, G. Sanniti di Baja, Eds.)CIARP 2013: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Lecture Notes in Computer Science, vol 8258. **Anais...**Berlin: Springer Berlin Heidelberg, 2013.

NATARAJAN, R.; SUBRAMANIAN, J.; PAPAGEORGIU, E. I. Hybrid learning of fuzzy cognitive maps for sugarcane yield classification. **Computers and Electronics in Agriculture**, v. 127, p. 147–157, 2016.

NEWEGG. **GY-NEO6MV2 Flight Controller NEO-6M APM 2.5 GPS Module For Arduino EEPROM**. Disponível em: <<https://www.newegg.com/p/2A7-00D0-000D2>>. Acesso em: 12 jan. 2020.

NIKU, S. B. **Introduction to Robotics: Analysis, Control, Applications**. 2. ed. Hoboken, New Jersey, USA: John Wiley & Sons, 2011. v. 53.

PALÁCIOS, R. H. C. et al. Fault Identification in the Stator Winding of Induction Motors Using PCA with Artificial Neural Networks. **Journal of Control, Automation and Electrical Systems**, v. 27, n. 4, p. 406–418, 2016.

PALMIERI, N. et al. Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team. **Neurocomputing**, v. 306, p. 17–36, 2018.



PAPAGEORGIU, E. I. (ED.). **Fuzzy Cognitive Maps for Applied Sciences and Engineering**. Heidelberg: Springer-Verlag Berlin Heidelberg, 2014.

PAPAGEORGIU, E. I.; FROELICH, W. Multi-step prediction of pulmonary infection with the use of evolutionary fuzzy cognitive maps. **Neurocomputing**, v. 92, n. 1, p. 28–35, 2012.

PAPAGEORGIU, E. I.; SALMERON, J. L. A review of fuzzy cognitive maps research during the last decade. **IEEE Transactions on Fuzzy Systems**, v. 21, n. 1, p. 66–79, 2013.

PARKER, L. E. Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. **Journal of Physical Agents**, v. 2, n. 1, p. 5–14, 2008.

PASSINO, K. M.; YURKOVICH, S. **Fuzzy Control**. 1. ed. Menlo Park, CA, USA: Addison Wesley, 1998.

PEDRYCZ, W.; GOMIDE, F. **Fuzzy Systems Engineering: Toward Human-Centric Computing**. 1. ed. Hoboken, New Jersey, USA: John Wiley & Sons, 2007.

POCZETA, K.; PAPAGEORGIU, E. I.; YASTREBOV, A. **Application of Fuzzy Cognitive Maps to Multi-Step Ahead Prediction of Electricity Consumption**. 2018 Conference on Electrotechnology: Processes, Models, Control and Computer Science (EPMCCS). **Anais...**Kielce, Poland: IEEE, 2018.

RAJ, A. et al. Barriers to the Adoption of Industry 4.0 Technologies in the Manufacturing Sector: An Inter-Country Comparative Perspective. **International Journal of Production Economics**, v. 1, n. 1, p. 1–49, 2019.

RBR. **Honeywell, Fetch Robotics Team Up to Bring Autonomous Mobile Robots to Distribution Centers**. Disponível em: <<https://www.roboticsbusinessreview.com/content-from-our-sponsor/honeywell-fetch-robotics-team-up-for-distribution-centers/>>. Acesso em: 5 out. 2019.

RIZK, Y.; AWAD, M.; TUNSTEL, E. W. Decision Making in Multiagent Systems: A Survey. **IEEE Transactions on Cognitive and Developmental Systems**, v. 10, n. 3, p. 514–529, 2018.

ROBLA-GOMEZ, S. et al. Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. **IEEE Access**, v. 5, p. 26754–26773, 2017.

ROBOCUP. **A Brief History of RoboCup**. Disponível em: <[https://www.robotcup.org/a\\_brief\\_history\\_of\\_robotcup](https://www.robotcup.org/a_brief_history_of_robotcup)>. Acesso em: 1 nov. 2019.

ROMERO, R. A. F. et al. (EDS.). **Robótica Móvel**. 1. ed. Rio de Janeiro, Brasil: LTC Editora, 2014.

ROSS, T. J. **Fuzzy logic with engineering applications**. 3. ed. New Mexico, USA: John Wiley & Sons, 2010.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River, USA: Prentice Hall, 2009.

SALMERON, J. L. Modelling grey uncertainty with fuzzy grey cognitive maps. **Expert**

**Systems with Applications**, v. 37, n. 12, p. 7581–7588, 2010.

SALMERON, J. L.; PAPAGEORGIOU, E. I. Fuzzy grey cognitive maps and nonlinear Hebbian learning in process control. **Applied Intelligence**, v. 41, n. 1, p. 223–234, 2014.

SBA. **ROSI Challenge**. Disponível em: <<https://www.sbai2019.com.br/rosi-challenge>>. Acesso em: 1 nov. 2019.

SCHMICKL, T.; CRAILSHEIM, K. **A Navigation Algorithm for Swarm Robotics Inspired by Slime Mold Aggregation**. (E. Şahin, W. M. Spears, A. F. T. Winfield, Eds.) International Workshop on Swarm Robotics - SR 2006. **Anais...Rome, Italy: Springer**, 2006.

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

SIEGWART, R.; NOURBAKHSH, I. R. **Introduction to Autonomous Mobile Robots**. 1. ed. Cambridge, MA, USA: MIT Press, 2014.

SOARES, P. P. et al. **Group of Robots Inspired by Swarm Robotics Exploring Unknown Environments**. 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). **Anais...Rio de Janeiro, Brazil: IEEE**, 2018.

ŠTULA, M.; MARAS, J.; MLADENOVIĆ, S. Continuously self-adjusting fuzzy cognitive map with semi-autonomous concepts. **Neurocomputing**, v. 232, p. 34–51, 2017.

SUTTER, B. et al. A semi-autonomous mobile robot for bridge inspection. **Automation in Construction**, v. 91, p. 111–119, 1 jul. 2018.

TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 15, n. 1, p. 116–132, 1985.

THOMSEN, A. **Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino**. Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acesso em: 6 nov. 2019.

TOLMAN, E. C. Cognitive maps in rats and men. **Psychological Review**, v. 55, n. 4, p. 189–208, 1948.

VAŠČÁK, J. **Approaches in adaptation of fuzzy cognitive maps for navigation purposes**. SAMI 2010 - 8th International Symposium on Applied Machine Intelligence and Informatics, Proceedings. **Anais...2010**.

VENAYAGAMOORTHY, G. K.; GRANT, L. L.; DOCTOR, S. Collective robotic search using hybrid techniques: Fuzzy logic and swarm intelligence inspired by nature. **Engineering Applications of Artificial Intelligence**, v. 22, n. 3, p. 431–441, 2009.

WAHDE, M. **Introduction to Autonomous Robots**. 1. ed. Goteborg, Sweden: Chalmers University of Technology, 2016.

XIA, C. **Intelligent Mobile Robot Learning in Autonomous Navigation**. [s.l.] Ecole Centrale de Lille, 2016.

ZADEH, L. A. Fuzzy Sets. **Information and control**, v. 353, p. 338–353, 1965.

ZADEH, L. A. A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. **International Journal of Man-Machine Studies**, v. 8, n. 3, p. 249–291, 1976.

ZHANG, D.; WEI, B. (EDS.). **Adaptive Control for Robotic Manipulators**. 1. ed. Oshawa, ON: CRC Press, 2017.

ZHANG, P.; JETTER, A. **A framework for building integrative scenarios of autonomous vehicle technology application and impacts, using fuzzy cognitive maps (FCM)**. 2018 Portland International Conference on Management of Engineering and Technology (PICMET). **Anais...**Honolulu, HI, USA: IEEE, 2018.

## APÊNDICE A – REGRAS DO SISTEMA FUZZY

Todas as regras para o FLC utilizado neste trabalho são mostradas a seguir, conforme vistas no Matlab®.

1. Se (SE é próximo) e (SF é próximo) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)
2. Se (SE é próximo) e (SF é próximo) e (SD é próximomédio) então (w\_E é alto)(w\_D é médio)
3. Se (SE é próximo) e (SF é próximo) e (SD é médio) então (w\_E é alto)(w\_D é baixomédio)
4. Se (SE é próximo) e (SF é próximo) e (SD é médioidistante) então (w\_E é alto)(w\_D é baixo)
5. Se (SE é próximo) e (SF é próximo) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
6. Se (SE é próximo) e (SF é próximomédio) e (SD é próximo) então (w\_E é alto)(w\_D é alto)
7. Se (SE é próximo) e (SF é próximomédio) e (SD é próximomédio) então (w\_E é alto)(w\_D é médio)
8. Se (SE é próximo) e (SF é próximomédio) e (SD é médio) então (w\_E é alto)(w\_D é médio)
9. Se (SE é próximo) e (SF é próximomédio) e (SD é médioidistante) então (w\_E é alto)(w\_D é baixomédio)
10. Se (SE é próximo) e (SF é próximomédio) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
11. Se (SE é próximo) e (SF é médio) e (SD é próximo) então (w\_E é alto)(w\_D é alto)
12. Se (SE é próximo) e (SF é médio) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
13. Se (SE é próximo) e (SF é médio) e (SD é médio) então (w\_E é alto)(w\_D é médio)
14. Se (SE é próximo) e (SF é médio) e (SD é médioidistante) então (w\_E é alto)(w\_D é baixomédio)
15. Se (SE é próximo) e (SF é médio) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
16. Se (SE é próximo) e (SF é médioidistante) e (SD é próximo) então (w\_E é alto)(w\_D é alto)
17. Se (SE é próximo) e (SF é médioidistante) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
18. Se (SE é próximo) e (SF é médioidistante) e (SD é médio) então (w\_E é alto)(w\_D é alto)
19. Se (SE é próximo) e (SF é médioidistante) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto)
20. Se (SE é próximo) e (SF é médioidistante) e (SD é distante) então (w\_E é alto)(w\_D é alto)
21. Se (SE é próximo) e (SF é distante) e (SD é próximo) então (w\_E é alto)(w\_D é alto)
22. Se (SE é próximo) e (SF é distante) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
23. Se (SE é próximo) e (SF é distante) e (SD é médio) então (w\_E é alto)(w\_D é alto)
24. Se (SE é próximo) e (SF é distante) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto)
25. Se (SE é próximo) e (SF é distante) e (SD é distante) então (w\_E é alto)(w\_D é baixo)

26. Se (SE é próximomédio) e (SF é próximo) e (SD é próximo) então (w\_E é médioalto)(w\_D é alto)
27. Se (SE é próximomédio) e (SF é próximo) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
28. Se (SE é próximomédio) e (SF é próximo) e (SD é médio) então (w\_E é alto)(w\_D é médio)
29. Se (SE é próximomédio) e (SF é próximo) e (SD é médiodistante) então (w\_E é alto)(w\_D é baixomédio)
30. Se (SE é próximomédio) e (SF é próximo) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
31. Se (SE é próximomédio) e (SF é próximomédio) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
32. Se (SE é próximomédio) e (SF é próximomédio) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
33. Se (SE é próximomédio) e (SF é próximomédio) e (SD é médio) então (w\_E é alto)(w\_D é médio)
34. Se (SE é próximomédio) e (SF é próximomédio) e (SD é médiodistante) então (w\_E é alto)(w\_D é baixomédio) (1)'
35. Se (SE é próximomédio) e (SF é próximomédio) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
36. Se (SE é próximomédio) e (SF é médio) e (SD é próximo) então (w\_E é médioalto)(w\_D é alto)
37. Se (SE é próximomédio) e (SF é médio) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
38. Se (SE é próximomédio) e (SF é médio) e (SD é médio) então (w\_E é alto)(w\_D é médio)
39. Se (SE é próximomédio) e (SF é médio) e (SD é médiodistante) então (w\_E é alto)(w\_D é baixomédio)
40. Se (SE é próximomédio) e (SF é médio) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
41. Se (SE é próximomédio) e (SF é médiodistante) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
42. Se (SE é próximomédio) e (SF é médiodistante) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
43. Se (SE é próximomédio) e (SF é médiodistante) e (SD é médio) então (w\_E é alto)(w\_D é médio)
44. Se (SE é próximomédio) e (SF é médiodistante) e (SD é médiodistante) então (w\_E é alto)(w\_D é baixo)
45. Se (SE é próximomédio) e (SF é médiodistante) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
46. Se (SE é próximomédio) e (SF é distante) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
47. Se (SE é próximomédio) e (SF é distante) e (SD é próximomédio) então (w\_E é alto)(w\_D é alto)
48. Se (SE é próximomédio) e (SF é distante) e (SD é médio) então (w\_E é alto)(w\_D é médio)
49. Se (SE é próximomédio) e (SF é distante) e (SD é médiodistante) então (w\_E é alto)(w\_D é baixomédio)

50. Se (SE é próximomédio) e (SF é distante) e (SD é distante) então (w\_E é alto)(w\_D é médioalto)
51. Se (SE é médio) e (SF é próximo) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)
52. Se (SE é médio) e (SF é próximo) e (SD é próximomédio) então (w\_E é médio)(w\_D é alto)
53. Se (SE é médio) e (SF é próximo) e (SD é médio) então (w\_E é alto)(w\_D é alto)
54. Se (SE é médio) e (SF é próximo) e (SD é médioidistante) então (w\_E é alto)(w\_D é baixo)
55. Se (SE é médio) e (SF é próximo) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
56. Se (SE é médio) e (SF é próximomédio) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
57. Se (SE é médio) e (SF é próximomédio) e (SD é próximomédio) então (w\_E é médioalto)(w\_D é alto)
58. Se (SE é médio) e (SF é próximomédio) e (SD é médio) então (w\_E é alto)(w\_D é alto)
59. Se (SE é médio) e (SF é próximomédio) e (SD é médioidistante) então (w\_E é alto)(w\_D é baixomédio)
60. Se (SE é médio) e (SF é próximomédio) e (SD é distante) então (w\_E é alto)(w\_D é baixo)
61. Se (SE é médio) e (SF é médio) e (SD é próximo) então (w\_E é baixomédio)(w\_D é alto)
62. Se (SE é médio) e (SF é médio) e (SD é próximomédio) então (w\_E é médio)(w\_D é alto)
63. Se (SE é médio) e (SF é médio) e (SD é médio) então (w\_E é alto)(w\_D é alto)
64. Se (SE é médio) e (SF é médio) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto)
65. Se (SE é médio) e (SF é médio) e (SD é distante) então (w\_E é alto)(w\_D é médioalto)
66. Se (SE é médio) e (SF é médioidistante) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)
67. Se (SE é médio) e (SF é médioidistante) e (SD é próximomédio) então (w\_E é baixomédio)(w\_D é alto)
68. Se (SE é médio) e (SF é médioidistante) e (SD é médio) então (w\_E é alto)(w\_D é alto)
69. Se (SE é médio) e (SF é médioidistante) e (SD é médioidistante) então (w\_E é alto)(w\_D é médio)
70. Se (SE é médio) e (SF é médioidistante) e (SD é distante) então (w\_E é alto)(w\_D é alto)
71. Se (SE é médio) e (SF é distante) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
72. Se (SE é médio) e (SF é distante) e (SD é próximomédio) então (w\_E é médio)(w\_D é alto)
73. Se (SE é médio) e (SF é distante) e (SD é médio) então (w\_E é médioalto)(w\_D é alto)
74. Se (SE é médio) e (SF é distante) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto)
75. Se (SE é médio) e (SF é distante) e (SD é distante) então (w\_E é alto)(w\_D é médioalto)
76. Se (SE é médioidistante) e (SF é próximo) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)
77. Se (SE é médioidistante) e (SF é próximo) e (SD é próximomédio) então (w\_E é baixo)(w\_D é alto)
78. Se (SE é médioidistante) e (SF é próximo) e (SD é médio) então (w\_E é baixomédio)(w\_D é médio)
79. Se (SE é médioidistante) e (SF é próximo) e (SD é médioidistante) então (w\_E é médio)(w\_D é baixomédio)
80. Se (SE é médioidistante) e (SF é próximo) e (SD é distante) então (w\_E é alto)(w\_D é baixo)

81. Se (SE é méiodistante) e (SF é próximomédio) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
82. Se (SE é méiodistante) e (SF é próximomédio) e (SD é próximomédio) então (w\_E é médioalto)(w\_D é alto)
83. Se (SE é méiodistante) e (SF é próximomédio) e (SD é médio) então (w\_E é alto)(w\_D é alto)
84. Se (SE é méiodistante) e (SF é próximomédio) e (SD é méiodistante) então (w\_E é alto)(w\_D é alto)
85. Se (SE é méiodistante) e (SF é próximomédio) e (SD é distante) então (w\_E é alto)(w\_D é baixomédio)
86. Se (SE é méiodistante) e (SF é médio) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
87. Se (SE é méiodistante) e (SF é médio) e (SD é próximomédio) então (w\_E é médio)(w\_D é alto)
88. Se (SE é méiodistante) e (SF é médio) e (SD é médio) então (w\_E é alto)(w\_D é alto)
89. Se (SE é méiodistante) e (SF é médio) e (SD é méiodistante) então (w\_E é alto)(w\_D é alto)
90. Se (SE é méiodistante) e (SF é médio) e (SD é distante) então (w\_E é alto)(w\_D é alto)
91. Se (SE é méiodistante) e (SF é méiodistante) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
92. Se (SE é méiodistante) e (SF é méiodistante) e (SD é próximomédio) então (w\_E é médio)(w\_D é alto)
93. Se (SE é méiodistante) e (SF é méiodistante) e (SD é médio) então (w\_E é alto)(w\_D é alto)
94. Se (SE é méiodistante) e (SF é méiodistante) e (SD é méiodistante) então (w\_E é alto)(w\_D é baixo)
95. Se (SE é méiodistante) e (SF é méiodistante) e (SD é distante) então (w\_E é alto)(w\_D é médioalto)
96. Se (SE é méiodistante) e (SF é distante) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
97. Se (SE é méiodistante) e (SF é distante) e (SD é próximomédio) então (w\_E é médioalto)(w\_D é alto)
98. Se (SE é méiodistante) e (SF é distante) e (SD é médio) então (w\_E é alto)(w\_D é alto)
99. Se (SE é méiodistante) e (SF é distante) e (SD é méiodistante) então (w\_E é alto)(w\_D é alto)
100. Se (SE é méiodistante) e (SF é distante) e (SD é distante) então (w\_E é alto)(w\_D é alto)
101. Se (SE é distante) e (SF é próximo) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)
102. Se (SE é distante) e (SF é próximo) e (SD é próximomédio) então (w\_E é baixo)(w\_D é alto)
103. Se (SE é distante) e (SF é próximo) e (SD é médio) então (w\_E é baixomédio)(w\_D é alto)
104. Se (SE é distante) e (SF é próximo) e (SD é méiodistante) então (w\_E é alto)(w\_D é alto)
105. Se (SE é distante) e (SF é próximo) e (SD é distante) então (w\_E é médio)(w\_D é médio)
106. Se (SE é distante) e (SF é próximomédio) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)

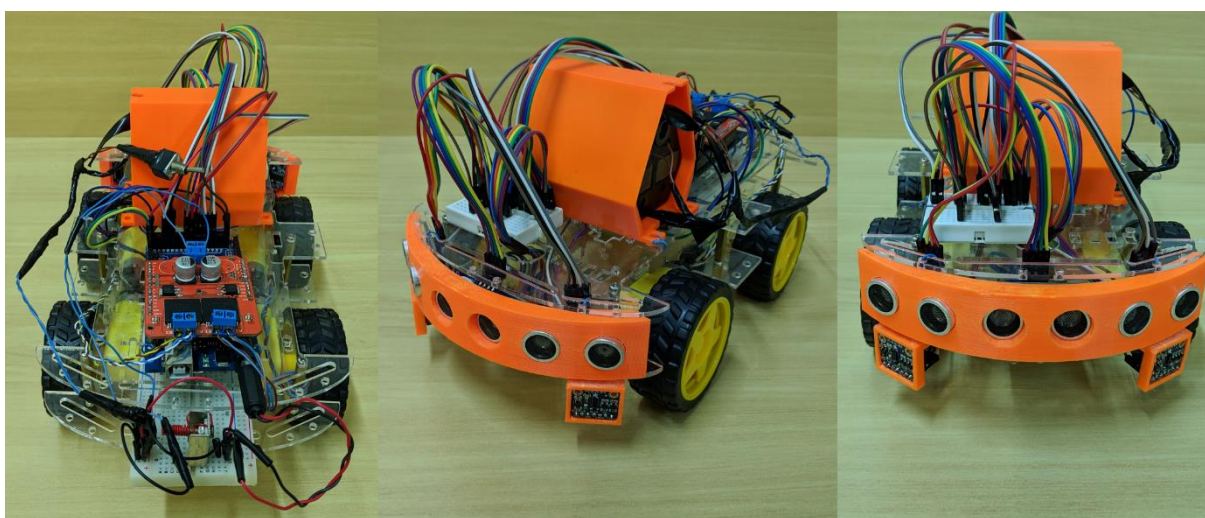
107. Se (SE é distante) e (SF é próximomédio) e (SD é próximomédio) então (w\_E é baixomédio)(w\_D é alto)
108. Se (SE é distante) e (SF é próximomédio) e (SD é médio) então (w\_E é médio)(w\_D é alto)
109. Se (SE é distante) e (SF é próximomédio) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto)
110. Se (SE é distante) e (SF é próximomédio) e (SD é distante) então (w\_E é alto)(w\_D é alto)
111. Se (SE é distante) e (SF é médio) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
112. Se (SE é distante) e (SF é médio) e (SD é próximomédio) então (w\_E é baixomédio)(w\_D é alto)
113. Se (SE é distante) e (SF é médio) e (SD é médio) então (w\_E é baixomédio)(w\_D é alto)
114. Se (SE é distante) e (SF é médio) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto)
115. Se (SE é distante) e (SF é médio) e (SD é distante) então (w\_E é alto)(w\_D é alto)
116. Se (SE é distante) e (SF é médioidistante) e (SD é próximo) então (w\_E é médio)(w\_D é alto)
117. Se (SE é distante) e (SF é médioidistante) e (SD é próximomédio) então (w\_E é médio)(w\_D é alto)
118. Se (SE é distante) e (SF é médioidistante) e (SD é médio) então (w\_E é médio)(w\_D é alto)
119. Se (SE é distante) e (SF é médioidistante) e (SD é médioidistante) então (w\_E é baixomédio)(w\_D é alto)
120. Se (SE é distante) e (SF é médioidistante) e (SD é distante) então (w\_E é alto)(w\_D é alto)
121. Se (SE é distante) e (SF é distante) e (SD é próximo) então (w\_E é baixo)(w\_D é alto)
122. Se (SE é distante) e (SF é distante) e (SD é próximomédio) então (w\_E é médioalto)(w\_D é alto)
123. Se (SE é distante) e (SF é distante) e (SD é médio) então (w\_E é alto)(w\_D é alto)
124. Se (SE é distante) e (SF é distante) e (SD é médioidistante) então (w\_E é alto)(w\_D é alto);
125. Se (SE é distante) e (SF é distante) e (SD é distante) então (w\_E é alto)(w\_D é baixo).



## APÊNDICE B – RESULTADOS INICIAIS DO PROTÓTIPO

Um pequeno circuito de testes real foi feito *a priori* para verificar o comportamento do DFCM embarcado. É importante salientar que para as experimentações práticas, o subcomportamento de resgate de vítimas é modificado: ao invés de alterar a matriz de pesos do DFCM, ele agora apenas envia a posição da vítima encontrada quando o robô passa sobre ela. O protótipo utilizado pode ser visto na Figura B.1

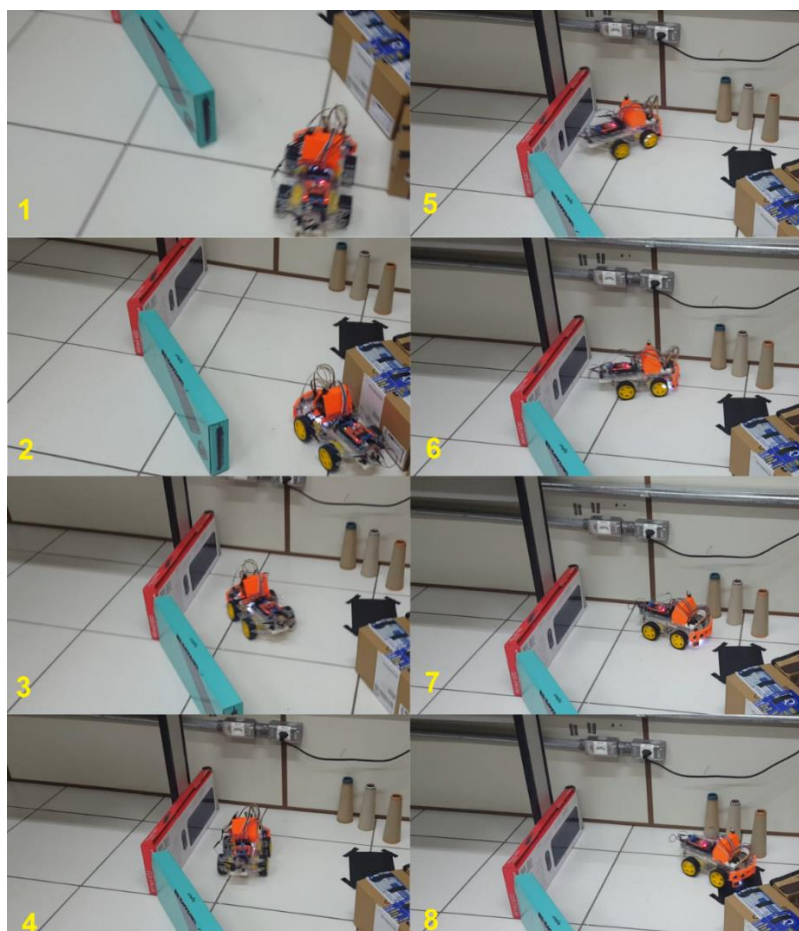
**Figura B.1 – Vistas do robô protótipo**



Fonte: Autoria própria.

Para simular uma vítima, foi utilizado um quadrado de fita isolante e sua percepção foi feita através de um sensor de cor posicionado na parte inferior do robô. Os resultados iniciais são mostrados na Figura 4.66. Nessas condições, os números descrevem a sequência de eventos do robô. Ao final (número oito), observa-se que o robô encontrou a vítima e parou sem colidir com as paredes.

A partir da verificação do funcionamento global do sistema pelo teste inicial mostrado na Figura B.2, a próxima etapa consistiu na fabricação do primeiro dos ambientes simulados em escala 2:1 para o teste final devido ao tamanho do robô móvel utilizado, que possui comprimento maior que o empregado nas simulações. Assim, foi gerado um mural das operações propostas – identificação e envio da localização dos alvos – para o protótipo apresentado. A sequência de eventos é mostrada na Figura B.3.

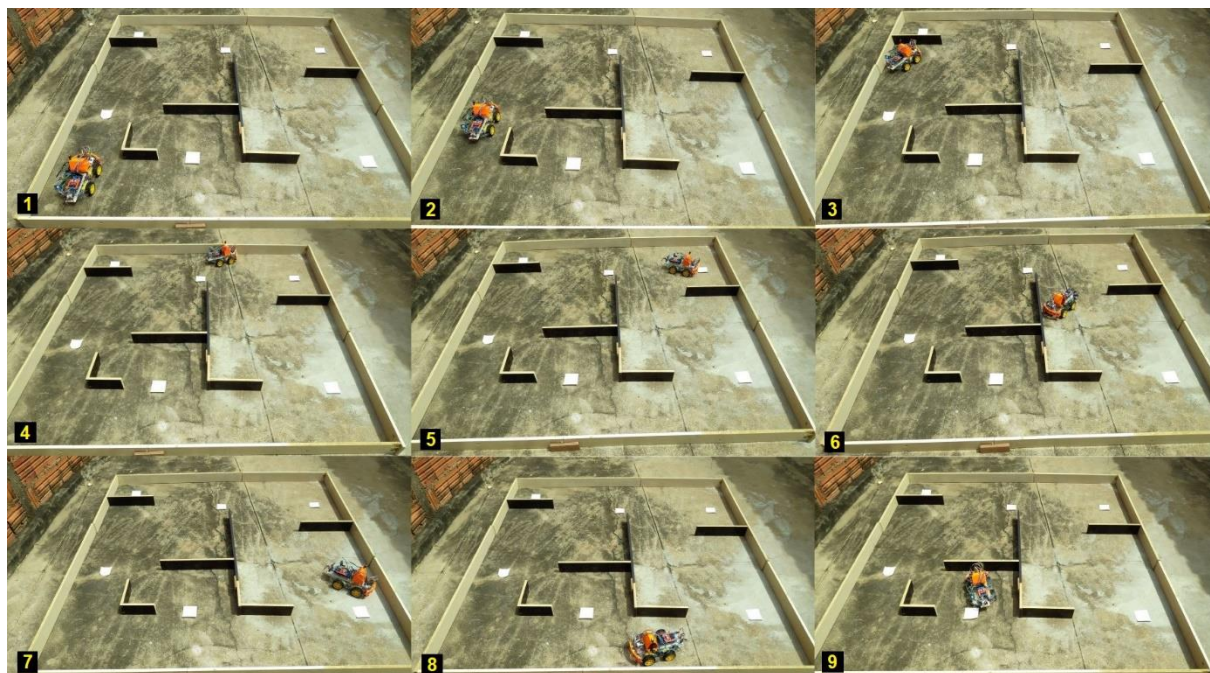
**Figura B.2 – Sequência de eventos do teste inicial**

Fonte: Autoria própria.

Pela Figura B.3 tem-se que o robô móvel encontrou três objetivos, como visto nos quadros 2, 5 e 9. Entretanto, nos quadros 4 e 7, o robô não foi capaz de identificar os objetivos dados pelos quadrados brancos na imagem. Neste experimento, algumas das possíveis fontes de erros são a luminosidade, possíveis mal contatos/falha de fios e componentes, além da reflexão das placas de madeira utilizadas como paredes e obstáculos.

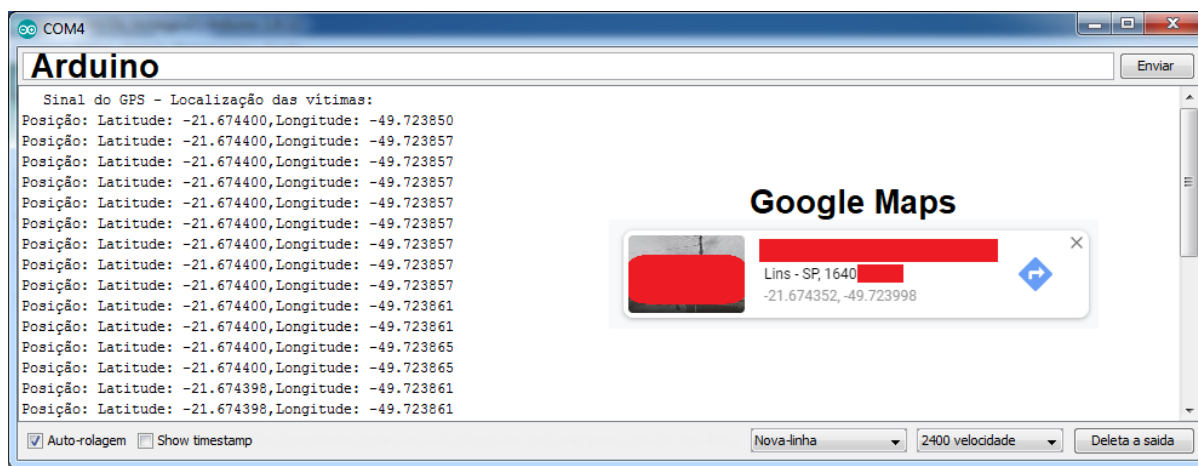
Na operação proposta, quando uma vítima é identificada o robô grava sua localização do GPS para possíveis socorristas realizarem o resgate. Contudo, neste caso não estava disponível a tempo um aparato para o envio das coordenadas obtidas para um smartphone ou computador, como módulos Wi-Fi ou Bluetooth, não foi possível a observação das mesmas, o que pode ser contornado futuramente pela utilização de um microcontrolador ESP8266, por exemplo. Ao ligar o sistema em um laptop, as coordenadas obtidas pelo monitor serial do Arduino® foram condizentes com as esperadas, como mostra a Figura B.4, com informações pessoais ocultadas.

Figura B.3 – Sequência de eventos do teste final



Fonte: Autoria própria.

Figura B.4 – Comparação entre coordenadas obtidas entre Google Maps e monitor serial



Fonte: Autoria própria.