

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

PATRICK PRIETO SOARES

**EXPLORAÇÃO DE AMBIENTES DESCONHECIDOS POR MEIO DE ROBÓTICA  
DE ENXAME**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2017

PATRICK PRIETO SOARES

**EXPLORAÇÃO DE AMBIENTES DESCONHECIDOS POR MEIO DE ROBÓTICA  
DE ENXAME**

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

Orientador: Prof. Dr. Márcio Mendonça.

CORNÉLIO PROCÓPIO  
2017



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



## **FOLHA DE APROVAÇÃO**

**Patrick Prieto Soares**

### **Exploração de ambientes desconhecidos por meio de robótica de enxame**

Trabalho de conclusão de curso apresentado às 17:00hs do dia 30/11/2017 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

---

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

---

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

---

Prof(a). Me(a). Marco Antonio Ferreira Finocchio - (Membro)

## **AGRADECIMENTOS**

Agradeço a minha família, Miriam e Wellington Jr., por todo o apoio e incentivo, a fé e o esforço que me fizeram o homem que sou hoje.

Ao meu orientador Prof. Dr. Márcio Mendonça, pelas oportunidades, experiências e conselhos.

Aos meus amigos e companheiros de república pelos momentos lúdicos e descontraídos.

Gostaria de deixar registrado também, o meu reconhecimento a todos que fizeram parte dessa fase da minha vida, pelos momentos, experiências e aprendizados.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

“Work until you no longer have to introduce yourself”  
-Harvey Specter

## RESUMO

SOARES, Patrick P. **Exploração de ambientes desconhecidos por meio de robótica de enxame**. 2017. 68 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Este trabalho tem como objetivo aplicar conceitos da robótica de enxame utilizando robôs móveis autônomos capazes de explorar ambientes desconhecidos ou semidesconhecidos. Técnicas computacionais inteligentes são aplicadas permitindo o robô interagir com o meio enquanto realiza suas funções. Apesar de ser um estudo em ambiente simulado, são realizados diversos experimentos em diferentes ambientes para verificar a autonomia do robô. Espera-se colaborar e poder servir de inspiração para futuros trabalhos na área e projetos capazes de utilizar essa tecnologia na resolução de diversos problemas como exploração de outros planetas, resgate de vítimas, entre outros.

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

## ABSTRACT

SOARES, Patrick P. **Unknown environment exploration by means of swarm robotics.** 2017. 68 f. Monograph – Control and Automation Engineering. Federal University of Technology - Paraná. Cornélio Procópio, 2017.

This paper's goal is to apply Swarm Robotics concepts using autonomous mobile robots capable of exploring semi-unknown or unknown environments. Intelligent computational techniques are applied allowing the robot to interact with the environment while performing its functions. Although, it is a simulated environment study, there will be run many tests on different scenarios in order to verify the robots autonomy. It is expected to collaborate and serve as inspiration for future works on this area and projects capable of using this technology on solving problems such as exploring other planets, rescuing victims, among others.

**Keywords:** Autonomous mobile robot. Swarm Robotics. Exploration of unknown environment. Robot navigation architecture.

## LISTA DE FIGURAS

Figura 1 - Robôs exploradores de Marte.....	14
Figura 2 - Aspirador Roomba980 e sua central de recarga.....	20
Figura 3 - <i>Landmarks</i> do robô Roomba980.....	21
Figura 4 – Estratégia defensiva de um cardume.....	23
Figura 5 – Metodologia de modelagem de comportamentos animais.....	25
Figura 6 – Lógica clássica (a) e Lógica <i>Fuzzy</i> (b) I.....	27
Figura 7 – Lógica clássica (a) e Lógica <i>Fuzzy</i> (b) II.....	27
Figura 8 – Arquitetura Reativa.....	28
Figura 9 – Sensor ultrassônico HC-SR04.....	31
Figura 10 – Ambiente de simulação I.....	33
Figura 11 – Ambiente de simulação II.....	33
Figura 12 – Ambiente de simulação III.....	34
Figura 13 - Diagrama de um robô móvel.....	35
Figura 14 - Movimento infinitesimal.....	35
Figura 15 – Síntese da alternância entre os modos de operação.....	39
Figura 16 – Funções de pertinência.....	41
Figura 17 – Estrutura do sistema <i>Fuzzy</i> .....	42
Figura 18 – Superfícies de controle do sistema <i>Fuzzy</i> .....	42
Figura 19 – Área de alcance dos sensores.....	43
Figura 20 – Planos auxiliares com robô em diferentes poses.....	44
Figura 21 – Distância entre dois pontos.....	45
Figura 22 – Quadrantes em função de senos e cossenos.....	46
Figura 23 – Simulação do ambiente I com um explorador.....	49
Figura 24 – Pulsos do explorador na simulação do ambiente I.....	50
Figura 25 – Simulação do ambiente I com múltiplos exploradores.....	51
Figura 26 – Pulsos do explorador 1 na simulação do ambiente I.....	51
Figura 27 – Pulsos do explorador 2 na simulação do ambiente I.....	52
Figura 28 – Pulsos do explorador 3 na simulação do ambiente I.....	52
Figura 29 – Pulsos do explorador 4 na simulação do ambiente I.....	53
Figura 30 – Simulação do ambiente II com um explorador.....	54
Figura 31 – Pulsos do explorador na simulação do ambiente II.....	54
Figura 32 – Simulação do ambiente II com múltiplos exploradores.....	55



Figura 33 – Pulsos do explorador 1 na simulação do ambiente II .....	56
Figura 34 – Pulsos do explorador 2 na simulação do ambiente II .....	56
Figura 35 – Pulsos do explorador 3 na simulação do ambiente II .....	57
Figura 36 – Pulsos do explorador 4 na simulação do ambiente II .....	57
Figura 37 – Simulação do ambiente III com um explorador .....	58
Figura 38 – Pulsos do explorador na simulação do ambiente III .....	59
Figura 39 – Simulação do ambiente III com múltiplos exploradores .....	60
Figura 40 – Pulsos do explorador 1 na simulação do ambiente III .....	60
Figura 41 – Pulsos do explorador 2 na simulação do ambiente III .....	61
Figura 42 – Pulsos do explorador 3 na simulação do ambiente III .....	61
Figura 43 – Pulsos do explorador 4 na simulação do ambiente III .....	62

## LISTA DE SIGLAS E ACRÔNIMOS

MatLab®	<i>Matrix Laboratory</i>
IA	Inteligência Artificial
UTFPR	Universidade Tecnológica Federal do Paraná
NASA	<i>National Aeronautics and Space Administration</i>
FCM	<i>Fuzzy Cognitive Maps</i>
AG	Algoritmos Genéticos

## LISTA DE SÍMBOLOS

$\Theta$	Ângulo formado entre o robô e o eixo x
$\alpha$	Ângulo formado entre o alvo e o eixo x'
$\omega$	Velocidade angular do robô

## SUMÁRIO

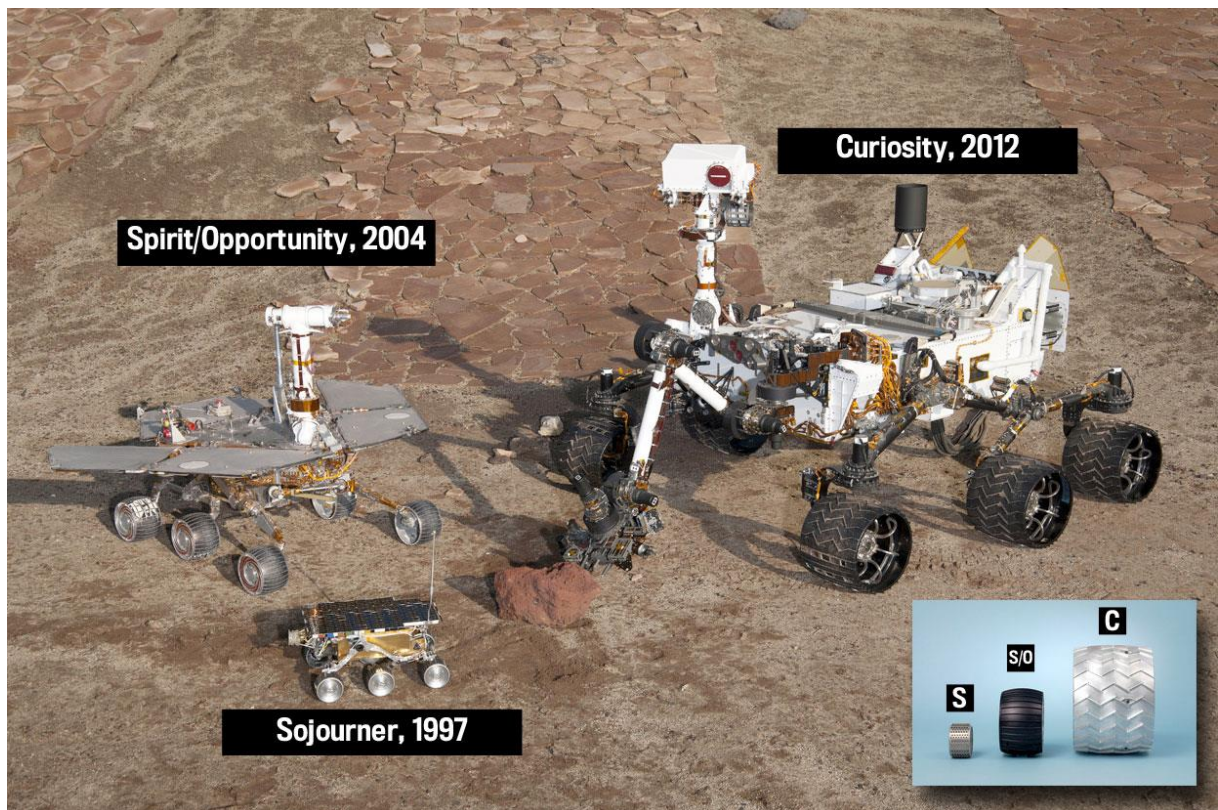
<b>1 INTRODUÇÃO</b> .....	<b>14</b>
1.1 Motivação.....	15
1.2 Trabalhos correlatos .....	16
1.3 Objetivos gerais .....	16
1.4 Objetivos específicos.....	17
1.5 Estrutura do trabalho .....	17
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>19</b>
2.1 Robótica móvel autônoma.....	19
2.2 Robótica de enxame .....	21
2.3 Sistemas <i>Fuzzy</i> .....	25
2.4 Arquitetura de subsunção .....	27
<b>3 METODOLOGIA</b> .....	<b>30</b>
3.1 Ambientes de teste .....	32
3.2 Modelo cinemático do robô.....	34
3.3 Modos de operação.....	36
3.3.1 Modo de operação zero: movimento livre.....	37
3.3.2 Modo de operação um: desvio de obstáculos .....	37
3.3.3 Modo de operação dois: iminência de colisão .....	37
3.3.4 Modo de operação três: captura de alvos .....	38
3.3.5 Síntese dos modos de operação .....	38
3.4 Sistema <i>Fuzzy</i> .....	39
3.5 Simulação dos sensores .....	43
3.6 Modelo robótico simulado .....	47
3.7 Parâmetros de teste .....	48
<b>4 RESULTADOS</b> .....	<b>49</b>
4.1 Resultados do ambiente de teste I .....	49

<b>4.2 Resultados do ambiente de teste II .....</b>	<b>53</b>
<b>4.3 Resultados do ambiente de teste III.....</b>	<b>58</b>
<b>4.4 Resultados gerais e discussões .....</b>	<b>62</b>
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>65</b>
<b>REFERÊNCIAS.....</b>	<b>67</b>

## 1 INTRODUÇÃO

A robótica consiste na aplicação de robôs na execução de tarefas, no qual podem ser capazes de realizar seus objetivos sozinhos ou em grupo. Sua utilização expande as capacidades motoras ou intelectuais, e contornam limitações físicas humanas, como por exemplo, a exploração de Marte realizada pelos robôs *Curiosity*, *Sojourner*, *Spirit* e *Opportunity* da NASA. Na Figura 1, pode-se observar os três modelos de robôs exploradores e no canto inferior direito o modelo da roda de cada um deles. O primeiro a pousar em solo marciano foi *Sojourner* pelo projeto *Mars Pathfinder*, seguido em 2004 pelos robôs geólogos gêmeos *Spirit* e *Opportunity* do projeto *Mars Exploration Rovers*, responsáveis por explorar a superfície do planeta, caracterizando seu clima e geologia. Por fim, o robô *Curiosity* do projeto *Mars Science Laboratory*, enviado em 2012 para estudar a possibilidade de Marte já ter abrigado alguma forma de vida.

Figura 1 - Robôs exploradores de Marte



Fonte: <<https://cdn.theatlantic.com/static/mt/assets/science/EvolutionofRovers.jpg>> Acesso em 13/05/2017.

Esse feito foi precedido de estudos massivos em robótica móvel, capacidade de aprendizagem, inteligência computacional e muitas outras áreas, que juntas permitiram ao homem alcançar novos horizontes. Essas pesquisas tem ganhado ainda mais força com o avanço tecnológico.

Ferramentas computacionais mais poderosas permitem simular, com certa simplicidade, como o cérebro humano processa as informações ao seu redor, adapta-se e gera conhecimento que pode ser utilizado na realização das tarefas necessárias. Um exemplo dessas ferramentas computacionais são os sistemas *Fuzzy*, capazes de lidar com informações imprecisas baseando-se em um conjunto de regras especialistas (ZADEH, 1965).

A exploração de ambientes desconhecidos como o caso do planeta Marte, é abordado em diversos estudos na área da robótica autônoma móvel. As principais dificuldades da área são: a percepção e modelagem do ambiente, localização e controle de tomada de decisões, entre outras (COSTA; GOUVÊA, 2010).

Essas são as bases para a exploração de novos ambientes, pois um robô precisa ser capaz de reagir de forma autônoma a obstáculos e responder a eventos imprevisíveis (RUSSELL; NORVIG, 1995).

Nesse contexto, uma área que tem chamado atenção nos estudos sobre robótica móvel autônoma é a robótica de enxame, que consiste na utilização de diversos robôs simples, que juntos simulam comportamentos coletivos observados na natureza, como formigas que procuram alimento em regiões desconhecidas deixando um rastro de feromônios no caminho mais curto para as demais seguirem.

Assim, ferramentas computacionais auxiliadas da inteligência computacional podem ser aplicadas na robótica para desenvolvimento de robôs autônomos, capazes de interagir e se adaptar em diversos cenários (MENDONÇA, 2011).

## **1.1 Motivação**

A robótica de enxame tem sido tema de pesquisas por encontrar soluções de forma simples e descentralizada para problemas complexos que eram intratáveis da forma como eram estudados. Apesar disso, ainda não foram muitas técnicas que encontraram uma aplicação real.

Nesse contexto, pretende-se utilizar conceitos da robótica de enxame para auxiliar robôs de busca em missões de resgate. Resgatistas que atuam em casos como o ataque as torres gêmeas de 11 de setembro, pessoas perdidas em florestas ou vítimas de terremotos, poderiam ser auxiliados por esses robôs autônomos capazes de operar em ambientes com obstáculos fixos e dinâmicos desconhecidos e quantidade incerta de alvos (vítimas), diminuindo o número de fatalidades pelo longo tempo que essas operações costumam durar.

## 1.2 Trabalhos correlatos

Essa subseção aborda trabalhos correlatos aos temas abordados nesta pesquisa, visando justificar a necessidade e atualidade dos temas em questão no meio científico.

Meléndez et al.(2009) utilizaram um robô NXT MINDSTORM da Lego™ controlado por Lógica *Fuzzy* Clássica para atravessar por um labirinto simulado. Porém, o controlador simulado era computacionalmente pesado para ser embarcado no micro controlador do robô real e não pode ser implementado totalmente.

Ghanea-Hercock e Barnes(1996) controlaram um robô utilizando a arquitetura de Subsunção de Brooks (1985) através de um base de regras *Fuzzy* dinamicamente atualizadas por algoritmos genéticos (AG). O robô estava em um ambiente no qual ele e outro robô autônomo deveriam alcançar o objetivo sem colisões com as paredes ou entre si.

Por fim, outro trabalho correlato interessante é o de Alwan (1996), no qual foi apresentada uma arquitetura híbrida de comando baseada em três níveis para lidar com obstáculos e seguir em direção a um local desejado. A simulação apresenta bons resultados quanto ao controle de velocidade, capacidade de manobra e, principalmente, reação a obstáculos desconhecidos.

## 1.3 Objetivos gerais

Nesse trabalho, tem-se como objetivo principal desenvolver e simular um explorador (robô móvel) autônomo, capaz de tomar decisões ao interagir com ambientes desconhecidos, ou semidesconhecidos, enquanto vasculha uma área por alvos de posição também desconhecida. Para isso, o robô deverá conter modos de



operações hierarquizados de forma que seja capaz de realizar todas as funções propostas: mapear o ambiente, desviar de obstáculos e capturar alvos.

Como sugerido por Russel e Norvig (1995), serão utilizados no mínimo três ambientes diferentes para validação do modelo e verificar se o robô apresenta características autônomas.

Além disso, serão aplicados conceitos da robótica de enxame utilizando múltiplos exploradores para validar a possível utilização de técnicas que tornem o problema solucionável.

Leva-se em consideração que o problema lida com diversos fatores dificultantes como o não conhecimento prévio do ambiente (dimensões e localização dos obstáculos fixos e/ou dinâmicos) pelo explorador, ou desconhecimento da quantidade e localização dos alvos.

#### **1.4 Objetivos específicos**

- Implementar o modelo cinemático do explorador autônomo;
- Implementar simulação de sensores ultrassônicos;
- Projetar um sistema *Fuzzy* para controle dos servo-motores do explorador;
- Implementar ao menos três possíveis cenários de atuação;
- Implementar funções e modos de operação necessários para interação com o ambiente;
- Implementar exploradores que caracterizem robótica de enxame;
- Testar e validar a viabilidade da utilização de técnicas da robótica de enxame para melhor desempenho.

#### **1.5 Estrutura do trabalho**

Este trabalho está estruturado da seguinte forma: a Seção 2 aborda o embasamento teórico necessário para a progressão do trabalho, abordando temas como robótica móvel autônoma, robótica de enxame, sistemas *Fuzzy* e arquitetura de subsunção. A Seção 3 aborda toda a metodologia, inclusive os aspectos algorítmicos do explorador, do sistema *Fuzzy*, entre outros. A Seção 4 contém todos

os resultados e considerações obtidos nos testes, enquanto que a Seção 5 conclui este Trabalho de Conclusão de Curso.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, é apresentada toda a fundamentação teórica que serviu de embasamento para realização deste projeto. Os temas abordados são robótica móvel autônoma, robótica de enxame, sistemas *Fuzzy* e arquitetura de subsunção.

### 2.1 Robótica móvel autônoma

A robótica móvel lida com todo e qualquer robô capaz de movimentar-se (mudar sua posição espacial), geralmente se tratando de robôs com alguma forma de locomoção (rodas ou pernas). Entretanto, braços robóticos também fazem parte dessa área da robótica, pois apesar de ter uma base fixa, eles têm a capacidade de manipular a posição final do seu efetor. Esta é uma área que recebe muitos investimentos em pesquisas pelo seu potencial de aplicabilidade em situações reais.

Atualmente, robôs manipuladores são utilizados em indústrias de grande porte, hospitais, armazéns e muitos outros lugares, no qual são responsáveis pelas mais diferentes e desafiadoras funções possíveis. Robôs já convivem no dia-a-dia das pessoas, existem robôs que estão dentro das casas, como por exemplo, o robô Roomba980 da Figura 2, um aspirador autônomo criado pela empresa iRobot capaz de navegar por diversos ambientes utilizando uma tecnologia de *landmarks* visuais (marcações nas imagens obtidas pela câmera do robô) como visto na Figura 3. Elas são utilizadas como pontos de referência para sua localização e planejamento de trajetória, e guiam o aspirador para lugares que ainda não foram limpos ou para a central de recarga quando a bateria está baixa.

Portanto, a critério da definição de Maes (1995), no qual um agente autônomo é um sistema computacional que habita um ambiente complexo e dinâmico, capaz de sensoriar e atuar autonomamente visando à execução de tarefas para as quais ele foi projetado, o Roomba980 é um agente autônomo.

Não existe um consenso sobre a taxonomia de robôs, porém existem algumas definições conceituadas como agentes reativos e deliberativos. O robô móvel autônomo proposto nesse Trabalho de Conclusão de Curso é classificado como reativo, pois seu processo de tomada de decisões utiliza apenas informações locais e atuais obtidas pelos sensores do robô, isto é, ele apenas reage aos

estímulos externos, não apresentando capacidade de aprendizado (MATARIC, 2007).

Isso não acontece nos robôs do tipo deliberativo, pois eles utilizam informações de tempos passados de seus sensores e realizam funções de planejamento prévio as ações.

**Figura 2 - Aspirador Roomba980 e sua central de recarga**



Fonte: <<https://irobot.conteudoespecial.com.br/roomba-980/images/mobile/recurso3.jpg>> Acesso em 13/05/2017.

**Figura 3 - Landmarks do robô Roomba980**



Fonte: <[http://techno-stream.net/wp-content/uploads/2015/09/wpid-980\\_vlsam-1442382478713.jpg](http://techno-stream.net/wp-content/uploads/2015/09/wpid-980_vlsam-1442382478713.jpg)> Acesso em 14/05/2017.

## 2.2 Robótica de enxame

A robótica de enxame é considerada uma área de pesquisa com grande popularidade na área da tecnologia robótica (ABIDIN; ARSHAD; NGAH, 2013), possuindo um grande potencial ainda não aproveitado.

Proposta em 1980, a partir dos estudos na área de inteligência artificial (IA) e biologia, consiste na utilização de inúmeros robôs simples programados de forma a apresentarem comportamentos coletivos, como a busca por comida de uma colônia de formigas no trabalho de Marco Dorigo (1999), estratégias defensivas de um cardume de peixes (Figura 4) ou migração de um bando de pássaros. Como pode-se notar, essas tarefas seriam de extrema dificuldade para um simples indivíduo, porém, a partir de um comportamento coletivo, emerge-se o que é chamado de inteligência coletiva ou inteligência de enxame (TAN; ZHENG, 2013).

Essa inteligência coletiva emerge da atuação de indivíduos simples que apresentam certas características ao trabalhar com outros indivíduos iguais ou

diferentes. A robótica de enxame é composta por cinco características base, sendo elas:

- **Simplicidade:** uma característica chave quando se trabalha nessa área, pois os robôs devem ser fisicamente simples, assim como sua complexidade algorítmica. Alavancando seu baixo custo;
- **Descentralização:** todos os indivíduos são autônomos na tomada de decisão, mesmo que as escolhas de um afetem os demais, cada robô deve ter a capacidade de agir por si só. Essa característica, na maioria das vezes, garante a escalabilidade da população;
- **Escalabilidade:** qualquer tamanho de população pode gerar a inteligência coletiva, assim se um ou mais robôs apresentarem falhas, não atrapalharão a função dos demais;
- **Comunicação local:** os indivíduos apenas se comunicam com indivíduos que estejam na sua proximidade, caracterizando um sistema de atuação e decisão descentralizado;
- **Paralelismo:** Por ser composto por múltiplos indivíduos em um único ambiente, cada robô deve ser capaz de trabalhar com diversos objetivos, como desvio de obstáculos, busca de alvos, comunicação com outros indivíduos, movimentação coordenada, entre outras funções possíveis.

Essas características da robótica de enxame são muito vantajosas, pois exigem um baixo custo computacional, uma programação simples, pouca memória e processamento individual (ABIDIN; ARSHAD; NGAH, 2013), diferente de outras áreas de inteligência artificial (IA).

É importante salientar a diferença entre robótica de enxame, nano robótica e sistemas multi-agentes, pois muitas vezes apresentam conceitos similares que causam confusão. As principais diferenças estão no nível de complexidade computacional e física de cada robô, seu porte e quantidade de indivíduos.

**Figura 4 – Estratégia defensiva de um cardume**



Fonte: <<http://likefotos.com/wp-content/uploads/2014/11/mergulho-cardume.jpg>>  
Acesso em 28/05/2017.

A nano robótica trabalha com robôs pequenos o suficiente para entrar na corrente sanguínea, porém cada robô pode, ou não, ter alta complexidade e ser capaz de realizar sozinho a tarefa ao qual foi construído. Muitas vezes, confunde-se a necessidade de utilizar diversos nano robôs com uma abordagem da robótica de enxame. O motivo de algumas aplicações exigirem utilizar um grande número de indivíduos é a necessidade de os robôs cumprirem uma tarefa que levaria muito tempo para um único indivíduo realizar devido ao seu tamanho e não sua incapacidade.

A robótica de enxame, por sua vez, faz uso da aplicação de robôs computacionalmente e fisicamente simples, indiferente do tamanho (pode ser nano ou do tamanho de *drones*), capazes de demonstrar certo grau de inteligência ao trabalharem paralelamente de forma descentralizada (inteligência coletiva).

Por fim, sistemas multi-agentes geralmente apresentam um controle centralizado, no qual as ações de um agente dependem das ações dos demais. Esses sistemas muitas vezes são complexos, como por exemplo, braços robóticos trabalhando em uma linha de produção. No Quadro 1, consta um resumo sobre a comparação entre essas três áreas da robótica móvel.

<b>Quadro 1 – Comparação entre multi-agentes, robótica de enxame e nano robótica.</b>			
<b>Área</b>	<b>Robótica de enxame</b>	<b>Nano robótica</b>	<b>Multi-agente</b>
<b>Quantidade de indivíduos</b>	Variável	Um ou mais	Pequena
<b>Complexidade</b>	Baixa	Média/Alta	Alta
<b>Controle</b>	Descentralizado e autônomo	Centralizado	Centralizado ou remoto
<b>Escalabilidade</b>	Alta	Baixa	Baixa
<b>Flexibilidade</b>	Alta	Baixa	Baixa
<b>Mobilidade</b>	Sim	Sim	Limitada
<b>Ambiente</b>	Ambos	Ambos	Conhecido

Fonte: Autoria própria.

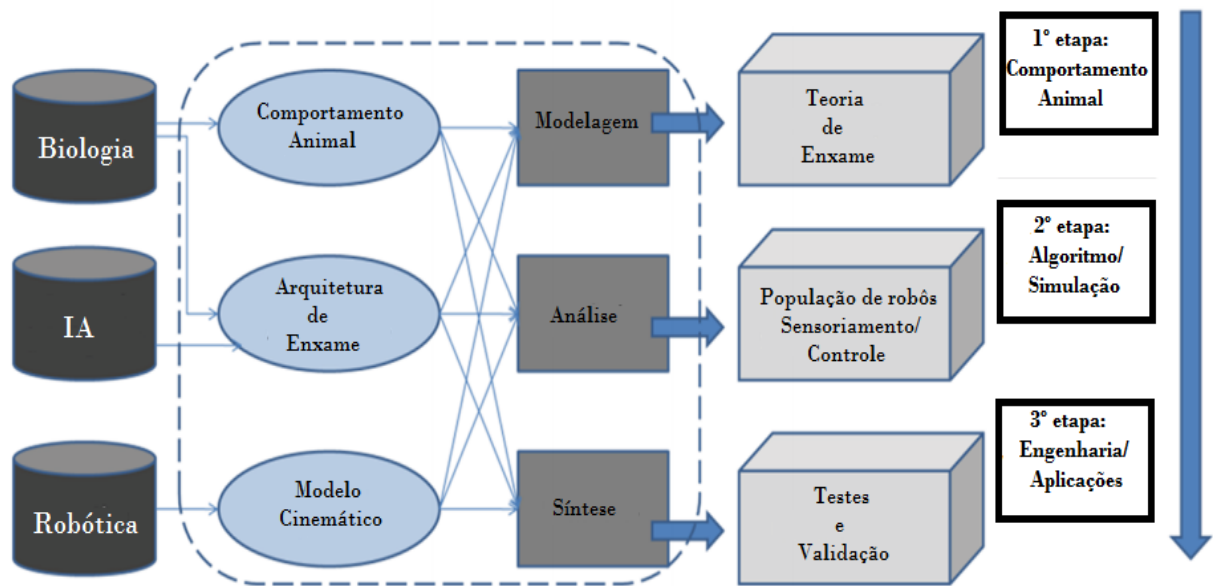
Em relação à pesquisa na área de robótica de enxame, existem diversas propostas de metodologias e soluções de problemas, porém poucas são aplicações práticas. Os estudos mais encontrados são na área de agregação (agrupamento de diversos indivíduos em uma mesma região, usualmente utilizado como ponto de início para funções mais complexas, como busca de alvos), formação de padrões (diversos indivíduos se organizam espacialmente de forma a preservar a comunicação local e superar dificuldades, como por exemplo, formar uma linha para atravessar uma passagem estreita), automontagem (uma conexão física é estabelecida entre os indivíduos para que possam superar dificuldades como um buraco no qual um único indivíduo cairia ou uma rampa muito íngreme), captura de alvos (indivíduos buscam alvos no ambiente e o retornam a um local específico), transporte coletivo (diversos indivíduos se unem para transportar um objeto, o qual seria impossível de ser realizado por apenas um robô), movimentação coordenada, mapeamento de ambiente, entre outros (LIEKNA; GRUNDSPENKIS, 2014).

A concepção de um comportamento coletivo através da robótica pode parecer simples por sua baixa complexidade, entretanto exige uma aplicação de conhecimento de diversas áreas (ABIDIN; ARSHAD; NGAH, 2013).

Em 2010, Vijay Kumar introduziu uma metodologia de como conceber um sistema com inteligência coletiva a partir do comportamento de animais, ela consiste em três etapas como pode ser visto na Figura 5.



Figura 5 – Metodologia de modelagem de comportamentos animais



Fonte: Adaptado de Abidin (2013).

A primeira etapa exige atenção, pois é uma observação cuidadosa do comportamento animal ao qual se pretende imitar, buscando compreender como a interação entre os indivíduos geram os resultados desejados ao coletivo. A segunda etapa é a análise do comportamento e sua implementação em forma de algoritmo, e por fim, a terceira etapa é a adequação do algoritmo obtido para a execução da finalidade desejada.

### 2.3 Sistemas Fuzzy

Proposto em 1965 por Zadeh, os sistemas *Fuzzy* foram uma evolução na abordagem de alguns problemas. Estes sistemas lidam com a informação qualitativamente, permitindo o trabalho com informações imprecisas. Assim como as demais ferramentas de inteligência computacional, os sistemas *Fuzzy* visam simular de forma simplificada como o cérebro humano lida com o ambiente ao seu redor e as informações que lhe são fornecidas.

Na lógica Booleana, o valor lógico de uma variável pode ter dois estados: “1” (verdadeiro) ou “0” (falso). Enquanto isso, a lógica *Fuzzy* assume que uma variável pode ter seu valor lógico representado por qualquer valor real entre [0,1], a partir de funções de pertinência (KASABOV, 1996).

Essa diferença de abordagem pode ser observada nas Figuras 6 e 7. Na Figura 6, pode-se notar que na lógica clássica Booleana qualquer velocidade abaixo de 100 km/h é considerada lenta (valor lógico “0”), enquanto os valores entre 100 e 140 km/h são considerados como rápido (valor lógico “1”). Por outro lado, na lógica *Fuzzy* os valores assumem graus de pertinência intermediários (0,35 e 0,76) indicando, por exemplo, uma velocidade “pouco rápida” e “meio rápida”, termos comumente utilizados no dia-a-dia das pessoas. O mesmo pode ser observado na Figura 7 quanto à temperatura, não existe uma divisão exata entre as definições de dia quente e dia não quente, sendo essa opinião relativa à perspectiva de cada pessoa (definição qualitativa).

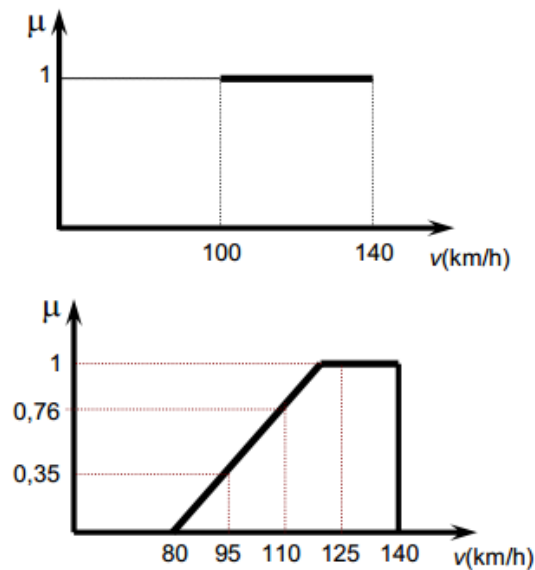
Os graus de pertinência são obtidos em relação a funções de pertinência que são definidas subjetivamente, heurísticamente ou por intuição (PASSINO; YURKOVICH, 1996). Por isso sistemas *Fuzzy* é uma ferramenta baseada em conhecimento especialista. Um funcionário que não saiba o modelo da planta de um processo pode construir uma base de regras que rege o comportamento do sistema baseado em sua experiência.

Alguns conceitos devem ser ressaltados na Figura 6, o trapézio é uma função de pertinência da variável “velocidade”, essa função é associada ao termo linguístico “rápido” e de acordo com o valor da variável pode apresentar diferentes graus de pertinência “ $\mu$ ” a essa função. No caso de um sistema de controle *Fuzzy*, existem mais funções de pertinência que compõem o universo de discurso de uma variável, assim uma variável pode assumir diversos graus de pertinência em relação a cada uma das funções.

A partir de uma base de regras linguísticas condicionais do tipo “SE – ENTÃO” são realizadas as operações *Fuzzy*. Essas regras devem ser caracterizadas de forma a representar o conhecimento especialista (MENDONÇA, 2011).

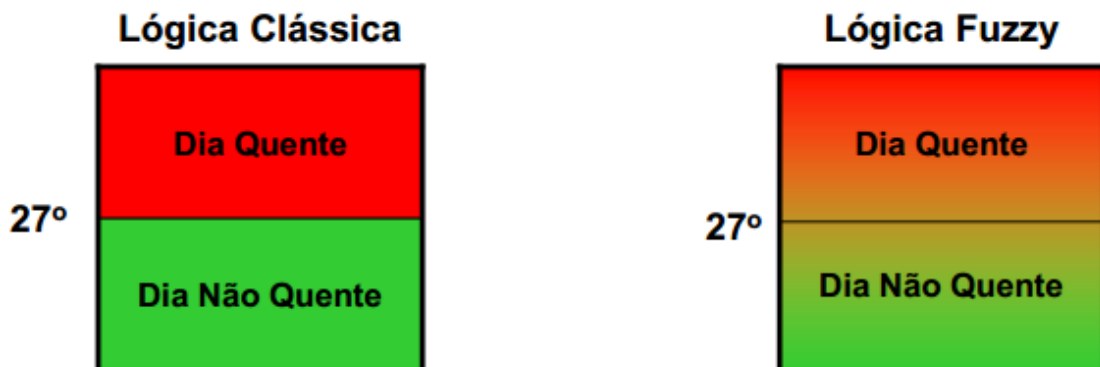
Assim um sistema *Fuzzy* pode representar um sistema utilizando apenas informações qualitativas, sem o modelo matemático do problema ou qualquer informação quantitativa (PASSINO; YURKOVICH, 1996).

Figura 6 – Lógica clássica (a) e Lógica *Fuzzy* (b) I



Fonte: Adaptado de Spatti (2016).

Figura 7 – Lógica clássica (a) e Lógica *Fuzzy* (b) II



Fonte: Adaptado de Spatti (2016).

## 2.4 Arquitetura de subsunção

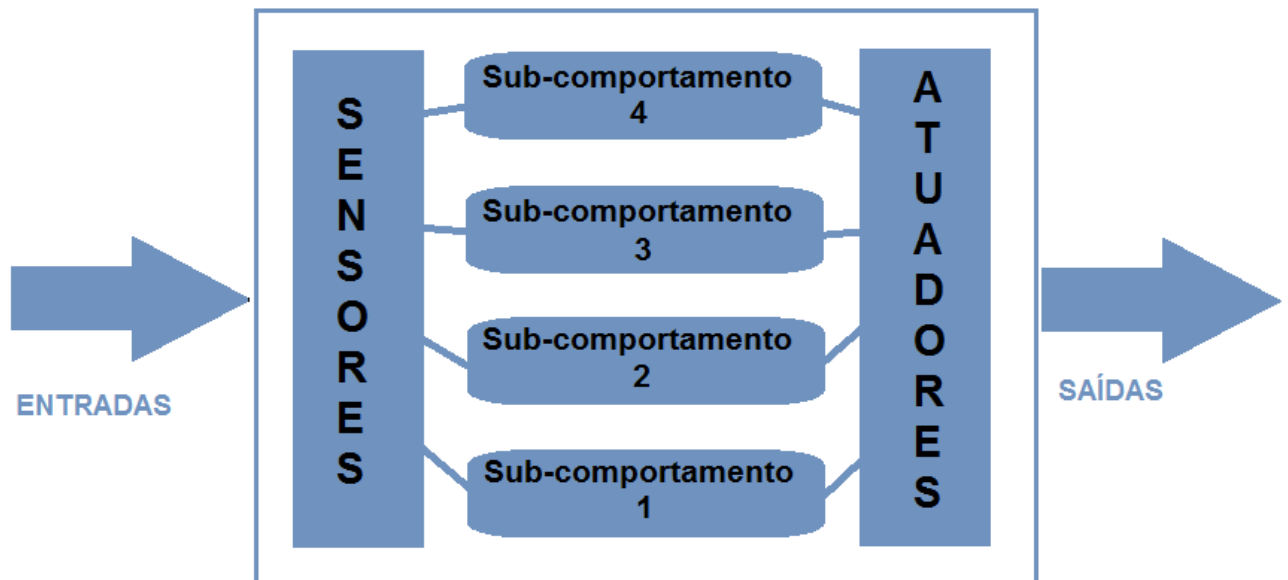
A arquitetura de subsunção proposta por Rodney Brooks foi uma ferramenta que trouxe uma nova perspectiva quanto à modelagem de robôs, influenciando diversas pesquisas na área da robótica autônoma e inteligência artificial (IA) em tempo real. Ela permite simular um comportamento complexo a partir da sua decomposição em sub-comportamentos, em oposição ao conceito tradicional de IA, que organiza o comportamento através de representações simbólicas do ambiente (BROOKS, 1991).

Essa arquitetura é justificável pelo princípio da incompatibilidade, o qual argumenta que quanto mais complexo for um sistema, mais difíceis e imprecisas serão as conjecturas sobre ele (ZADEH, 1976).

Esse princípio pode ser encontrado no exemplo dos veículos de Valentino Braitenberg, no qual quanto mais complexo o robô fica, mais difícil se torna prever como ele vai reagir ao ambiente (BRAITENBERG, 1986).

A arquitetura de subsunção é organizada de forma reativa a estímulos externos como mostra a Figura 8.

**Figura 8 – Arquitetura Reativa**



Fonte: Autoria própria.

Essa arquitetura decompõe o comportamento completo esperado do robô em sub-comportamentos, ou como chamados nesse trabalho, modos de operação. Esses modos de operação são organizados em camadas, a mais inferior sendo a mais essencial para a realização da função do robô no ambiente.

Cada camada é composta por uma máquina de estados finitos simples que rege seu funcionamento. Os modos de operação alternam entre si a partir das informações obtidas pelos sensores, sendo inibidos ou ativados. Todos os modos funcionam paralelamente, no qual os sub-comportamentos de nível mais baixo independem dos superiores (BROOKS, 1991).

Cada sub-comportamento tem seu próprio propósito e deve ser implementado progressivamente um a um. Suas capacidades de interação com o

ambiente (receber um estímulo externo e reagir de acordo com o seu objetivo) devem ser testadas extensivamente no cenário de atuação. Ao realizar isso, diminuem-se as chances de erros comportamentais posteriores. Além disso, a cada novo sub-comportamento adicionado, deve-se checar o sistema de alternância entre eles (BROOKS, 1991).

No fim, todos os sub-comportamentos gerando sinais de comando para os atuadores do robô simulam certo grau de inteligência. Essa inteligência é obtida através da interação com o ambiente, mesmo que esse seja dinâmico suas mudanças não causarão colapso na lógica do robô por ela ser descentralizada em diferentes camadas. Assim, caso o robô seja capaz de lidar em tempo real com os estímulos do ambiente, interagindo com aspectos dinâmicos e possuindo um ou mais objetivos de existir, ele poderá ser considerado autônomo (BROOKS, 1991).

### 3 METODOLOGIA

Para realização deste trabalho, inicialmente, foi empregada a arquitetura de subsunção, a qual consiste em começar o trabalho com um modelo simples e, passo a passo, implementar as funções desejadas para tratamento de um problema complexo. Organizar o trabalho dessa forma garante maiores chances de sucesso por exigir uma grande quantidade de testes a cada implementação, facilitando encontrar as causas dos erros no algoritmo. O *software* escolhido para realização desse trabalho foi o MatLab®.

Em seguida, obteve-se o modelo cinemático de um robô móvel, o qual a partir de pulsos enviados ao servo-motor de cada uma das rodas movimenta-se pelo ambiente.

Seguindo com a metodologia, implementou-se uma arena (ambiente simulado em duas dimensões) que delimita a área de operação do robô. Em um caso real, essa seria a área de busca dos resgatistas. O explorador deve ser capaz de interagir com esse ambiente, evitando possíveis colisões e manter-se vagando dentro dele até concluir seus objetivos.

Logo, a capacidade de desviar de obstáculos deve ser o primeiro modo de operação do robô (módulo essencial), no qual os pulsos de cada roda são as saídas geradas através de um sistema *Fuzzy*, cujas entradas são as informações obtidas pelos sensores do explorador. O sistema *Fuzzy* foi escolhido para controlar os pulsos da roda pela sua capacidade de lidar com informações incertas e/ou imprecisas, como por exemplo, a existência de ruído dos sensores.

O explorador é capaz de obter informações do ambiente através de um algoritmo que simula três sensores ultrassônicos posicionados de forma a permitir a interação do robô com o ambiente. Um exemplo de modelo real de sensor ultrassônico pode ser visto na Figura 9.

Além de reagir ao sinal de obstáculos em seus sensores, ao auxiliar em uma busca por sobreviventes, o robô móvel autônomo deve ser capaz de garantir sua segurança, isto é, não ser danificado durante sua operação. Para garantir que isso não ocorra, um modo de operação de emergência foi implementado. Ele é acionado em casos de iminência de colisão, garantindo assim maior robustez a ruídos dos sensores, capacidade de desvio de obstáculos dinâmicos, entre outros acontecimentos imprevisíveis.

**Figura 9 – Sensor ultrassônico HC-SR04**



Fonte: <<http://s3.amazonaws.com/img.iluria.com/product/6B8A2/2AE50F/450xN.jpg>> Acesso em 13/05/2017.

Em seguida, implementou-se a função *mapeamento*. Ao se deparar com obstáculos inicialmente desconhecidos pelo explorador, ele deve mapeá-los de forma a torná-los parte da arena (obstáculos ao qual se deve desviar). A detecção de novos obstáculos e alvos é visível na simulação pela mudança de cor dos mesmos ao adentrarem a região de alcance dos sensores simulados. Essa função muitas vezes pode não ser concluída totalmente (todos os obstáculos serem mapeados), pois a prioridade do robô é encontrar os alvos, e não mapear todos os obstáculos fixos. Ao encontrar a quantidade proposta de alvos, a simulação é encerrada, indiferente de quantos obstáculos desconhecidos ainda estiverem na arena.

A seguir, implementou-se um terceiro modo de operação, esse é responsável pela captura de alvos já encontrados pelo sensor. A função *capturar alvo* funciona alinhando o robô em direção ao alvo e gerando pulsos para que o explorador siga até ele, reduzindo a velocidade ao se aproximar (isto é, aproximar-se da vítima a uma baixa velocidade). A captura é completa após o explorador chegar a uma distância mínima definida pelo usuário, após isso, o alvo não é mais mostrado na simulação.

Este modo em particular só é interrompido pelo modo de operação dois: *iminência de colisão*, garantindo assim, que obstáculos próximos dos alvos não

acionem outros modos de operação e o impeçam de se aproximar e capturar o alvo. Em uma escala que classifica o quão reativo/deliberativo é um robô, esse modo é responsável por desclassificá-lo como puramente reativo já que existe um planejamento de trajetória por trás de seu algoritmo.

Com o modelo do explorador funcional, verificou-se métodos que possibilitassem a implementação de mais robôs exploradores autônomos, pois o custo computacional por adicionar diversas variáveis de tamanho indefinido pode ser altíssimo (a simulação pode demorar dias caso todos os alvos não sejam encontrados). Esse problema foi minimizado utilizando técnicas conhecidas e recomendadas pelo próprio *software* MatLab®, como por exemplo, utilizar alocação dinâmica, não utilizar variáveis como vetores desnecessariamente, entre outros. Após averiguar uma quantidade plausível de exploradores autônomos que podem ser simulados paralelamente, foram definidos os parâmetros dos testes.

### **3.1 Ambientes de teste**

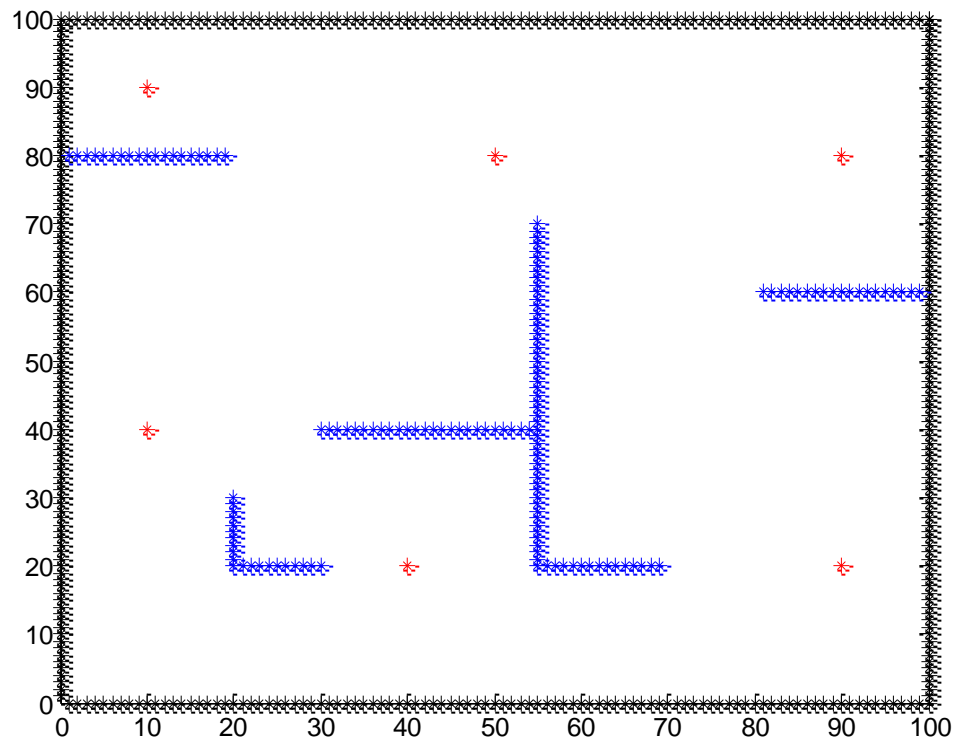
Neste trabalho, como proposto, foram utilizados três ambientes diferentes de teste, pois apenas alcançar os objetivos não é o suficiente para caracterizar um robô como autônomo. Ele deve ser capaz de movimentar-se, desviar de obstáculos e atuar sem interferências no seu funcionamento em todos os ambientes propostos.

Os ambientes são compostos de três elementos básicos: arena, obstáculos fixos e alvos. A arena inicialmente é a área de busca já reconhecida pelo robô denotada por pontos pretos, delimitando sua área de atuação. Ao longo da simulação os obstáculos desconhecidos também se tornam parte da arena após mapeamento.

A arena dos três ambientes tem dimensão 100x100 como pode ser observado nas Figuras 10, 11 e 12. Os seis alvos (pontos vermelhos) possuem a mesma posição em todos os ambientes, portanto a dificuldade de cada ambiente é consequência da posição dos obstáculos fixos. Eles são denotados por pontos azuis e são inicialmente desconhecidos pelo robô, tendo quantidade e posições diferentes em cada ambiente de teste.

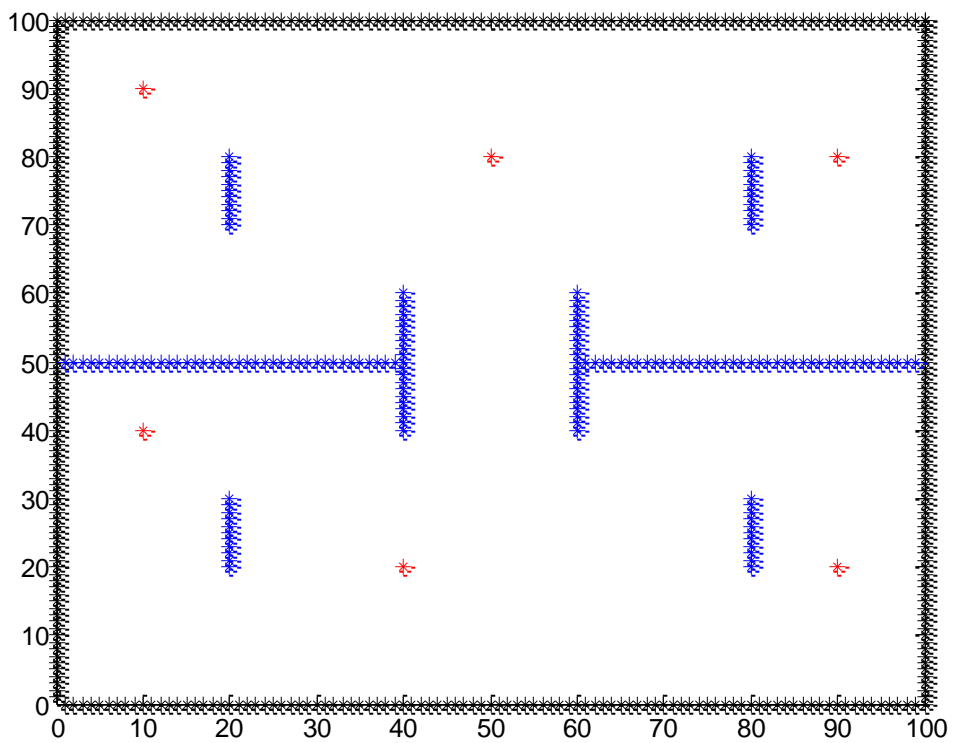


**Figura 10 – Ambiente de simulação I**



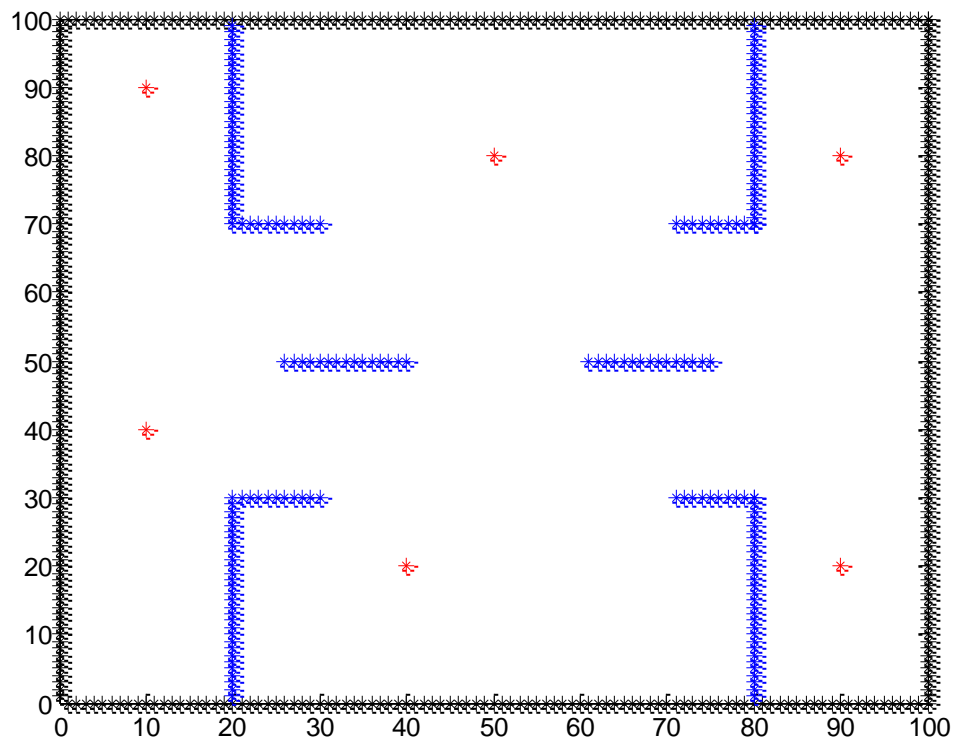
Fonte: Autoria própria.

**Figura 11 – Ambiente de simulação II**



Fonte: Autoria própria.

**Figura 12 – Ambiente de simulação III**



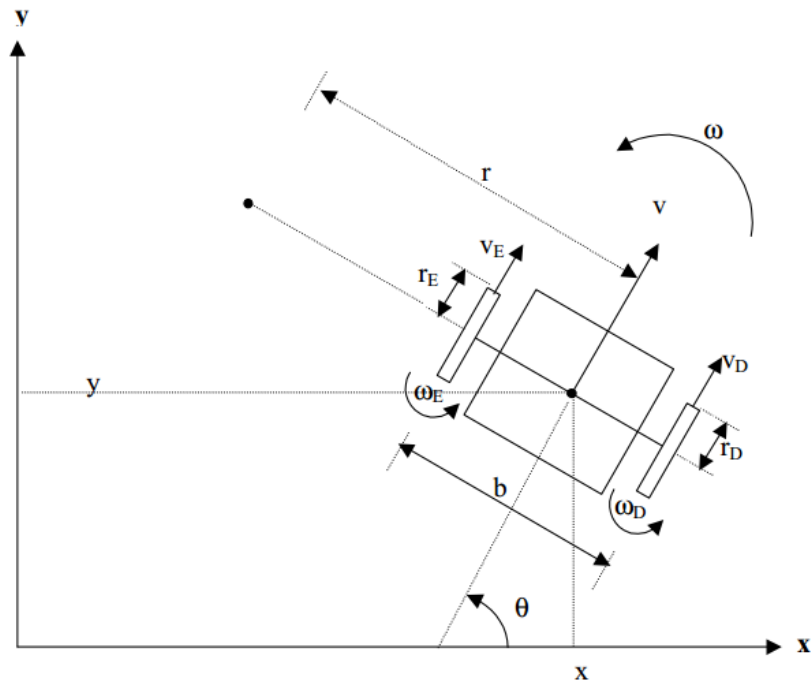
Fonte: Autoria própria.

### 3.2 Modelo cinemático do robô

Nesta subseção, é abordado especificamente como os pulsos enviados aos servo-motores influenciam na direção e posição do robô. Entender esta dinâmica é de suma importância para ser capaz de simular o movimento do robô autônomo o mais próximo possível da realidade.

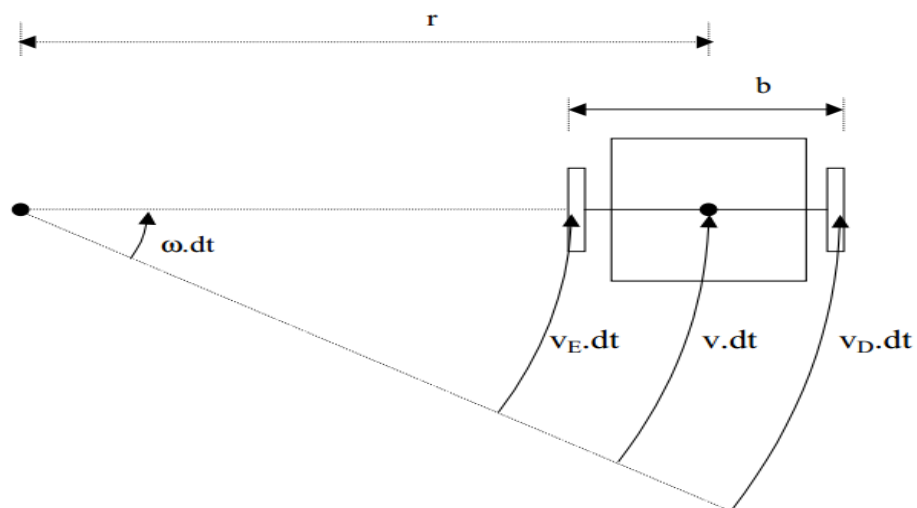
A Figura 13 mostra o diagrama de um robô móvel, no qual  $[x,y]$  é a posição do robô,  $\theta$  o ângulo de direção,  $b$  o comprimento do eixo,  $r$  o raio de giro,  $r_d$  e  $r_e$  os raios das rodas direita e esquerda respectivamente,  $\omega$  a velocidade angular do robô,  $\omega_d$  e  $\omega_e$  as velocidades angulares nas bordas das rodas direita e esquerda,  $v$  a velocidade linear do robô,  $v_d$  e  $v_e$  as velocidades lineares nas bordas das rodas direita e esquerda.

Figura 13 - Diagrama de um robô móvel



Fonte: Adaptado de Alsina (2016).

Figura 14 - Movimento infinitesimal



Fonte: Retirado de Alsina (2016).

Analisando o robô em relação a movimentos infinitesimais (Figura 14), pode-se demonstrar sua velocidade linear  $v$  e angular  $\omega$  em função dos pulsos  $\omega_d$  e  $\omega_e$  das rodas direita e esquerda, como visto nas equações a seguir:

$$v = \omega \cdot r \quad (1)$$

Portanto, as velocidades lineares de cada uma das bordas das rodas do robô são:

$$v_d \cdot dt = \omega \cdot \left(r + \frac{b}{2}\right) dt \quad (2)$$

$$v_e \cdot dt = \omega \cdot \left(r - \frac{b}{2}\right) dt \quad (3)$$

Utilizando as equações (2) e (3) chega-se a conclusão que:

$$v_d + v_e = \omega_d \cdot r_d + \omega_e \cdot r_e = 2 \cdot \omega \cdot r = 2 \cdot v \quad (4)$$

$$v_d - v_e = \omega_d \cdot r_d - \omega_e \cdot r_e = \omega \cdot b \quad (5)$$

Assim, isolando os termos  $v$  e  $\omega$  das equações (4) e (5) respectivamente, obtém-se a matriz que será utilizada para simular o modelo cinemático do robô explorador.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} (r_d/2) & (r_e/2) \\ (r_d/b) & -(r_e/b) \end{bmatrix} \cdot \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad (6)$$

Todos os robôs móveis autônomos simulados tem o mesmo modelo cinemático mostrado nas figuras desta subseção e, portanto, são descritos por ele. Os pulsos  $\omega_d$  e  $\omega_e$  são gerados pelo modo de operação ativo de acordo com o estado em que o robô se encontrar no momento, sendo que cada robô apresenta sua própria decisão. O conjunto de decisões de todos os indivíduos faz emergir a inteligência coletiva.

### 3.3 Modos de operação

Como citado na Seção 3, o explorador consiste de quatro modos de operação para ser capaz de realizar suas funções. Primeiramente, cada explorador é responsável por realizar as seguintes funções em ordem de prioridade:

- Localizar alvos;

- Capturar alvos;
- Mapear obstáculos desconhecidos.

Enquanto que os modos de operação são:

- **Zero:** Movimento livre;
- **Um:** Desvio de obstáculos;
- **Dois:** Iminência de colisão;
- **Três:** Captura de alvos.

### 3.3.1 Modo de operação zero: movimento livre

Ao início de uma nova iteração do algoritmo, o modo de operação de cada um dos robôs é atualizado para ele, por isso a definição de modo “zero”. Esse modo gerará pulsos de valor máximo (pulso unitário), mantendo o robô em velocidade máxima caso não haja presença de obstáculos. Caso existam obstáculos ou alvos sendo detectados pelos sensores, outros modos de operação de prioridade mais alta são ativados, inibindo este modo.

### 3.3.2 Modo de operação um: desvio de obstáculos

Este modo é ativado após a detecção de algum obstáculo através de seus sensores, sendo responsável por manobras que permitem o explorador percorrer o ambiente, como por exemplo, andar próximo a uma parede. Os valores de saída dos pulsos para cada roda são determinados por um sistema *Fuzzy*, variando de acordo com a leitura dos sensores quanto a distância entre o sensor e o obstáculo. O sistema *Fuzzy* tem o seus parâmetros demonstrados na Subseção 3.4.

### 3.3.3 Modo de operação dois: iminência de colisão

Este modo é ativado quando os três sensores ultrassônicos simulados detectam a presença de um obstáculo com uma distância menor do que a de segurança do explorador. Ele gera pulsos nos servo-motores de forma a girar rapidamente o explorador evitando a colisão. Além disso, para prevenir que o robô

fique preso circulando em certas regiões do ambiente, foi implementada uma lógica que muda a direção do movimento do explorador após certo número de vezes que esse modo foi acionado. Este modo também é responsável por impedir as colisões entre os diversos robôs utilizados. Ao utilizar mais de um robô em cada ambiente, eles são considerados como obstáculos dinâmicos aos demais indivíduos.

#### 3.3.4 Modo de operação três: captura de alvos

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

Esse modo, como já foi citado, só é interrompido pelo modo de operação dois para que alvos localizados próximos de obstáculos também possam ser capturados. Cada alvo só é capturado pelo explorador que o encontrou, assim, os demais indivíduos ficam responsáveis por desviar do explorador que estiver nesse modo.

Caso o explorador não seja capaz de capturar o alvo (por exemplo, ele esteja localizado depois de um obstáculo), após certo número de iterações o alvo volta a ser desconhecido para que outros robôs em melhores situações possam detectá-lo e capturá-lo.

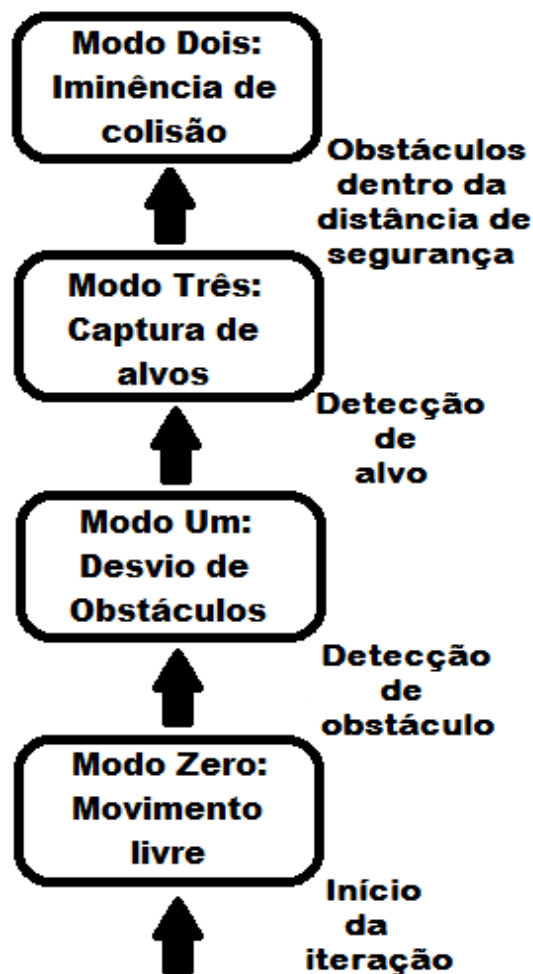
#### 3.3.5 Síntese dos modos de operação

Ao longo da simulação o explorador se encontra em diversos estados possíveis no qual deve reagir da forma mais apropriada possível, garantindo sua integridade e alcançando seus objetivos. Para isso seus modos de operação foram organizados de forma hierárquica como apresentado nas subseções anteriores ou como pode ser visto na Figura 15.

O modo *iminência de colisão* é o de maior prioridade seguido do modo *captura de alvos* e depois o de *desvio de obstáculos*. Essa sequência define quais são as prioridades do explorador ao longo da simulação. Ele começa vagando livremente, caso detecte algum obstáculo desviará. Porém, se ele detectar alguma vítima sua prioridade será resgatá-la, e seguirá adiante nessa função até concluir o

resgate ou encontrar-se em uma situação de iminência de colisão, assim ativando o modo de operação na qual sua prioridade é manter sua integridade. Isso acontece para que assim ele possa resgatar mais vítimas posteriormente ao invés de se danificar e encerrar sua operação. Caso ele evite a colisão imediata e ainda esteja responsável por resgatar determinada vítima, ele regressará para o modo de operação *captura de alvos*.

Figura 15 – Síntese da alternância entre os modos de operação



Fonte: Autoria própria.

### 3.4 Sistema *Fuzzy*

O sistema *Fuzzy* utilizado no modo de operação *desvio de obstáculos* determina o valor dos pulsos gerados para cada servo-motor (saídas do sistema,  $\omega_d$  e  $\omega_e$ ) a partir das entradas, que neste caso são as leituras dos sensores: esquerdo (SE), direito (SD) e frontal (SF). Assim, foi criada uma base de regras que

correlaciona as possíveis entradas e suas respectivas saídas desejadas. A base de regras escolhida contém 125 regras, trinta regras são apresentadas a seguir:

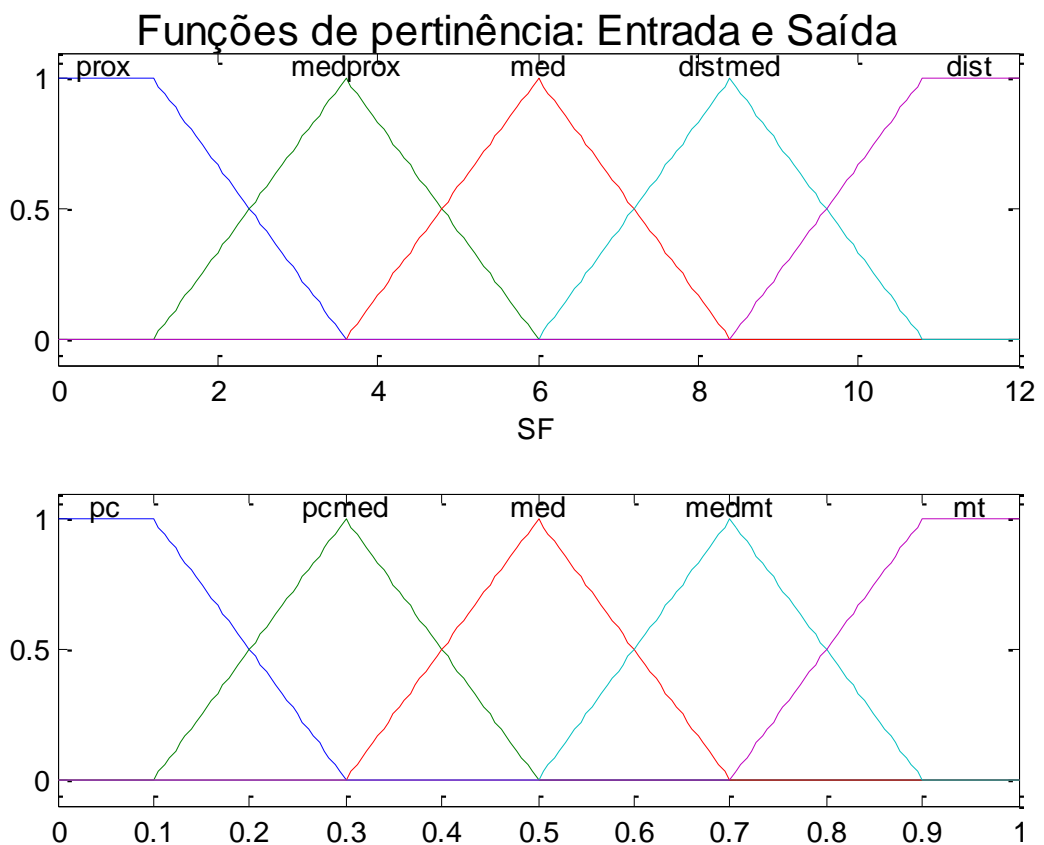
- (SE==prox)&(SF==prox)&(SD==prox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==prox)&(SD==medprox) => (we=mt)(wd=med);
- (SE==prox)&(SF==prox)&(SD==med) => (we=mt)(wd=pcmed);
- (SE==prox)&(SF==prox)&(SD==distmed) => (we=mt)(wd=pc);
- (SE==prox)&(SF==prox)&(SD==dist) => (we=mt)(wd=pc);
- (SE==prox)&(SF==medprox)&(SD==prox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==medprox)&(SD==medprox) => (we=mt)(wd=med);
- (SE==prox)&(SF==medprox)&(SD==med) => (we=mt)(wd=med);
- (SE==prox)&(SF==medprox)&(SD==distmed) => (we=mt)(wd=pcmed);
- (SE==prox)&(SF==medprox)&(SD==dist) => (we=mt)(wd=pc);
- (SE==prox)&(SF==med)&(SD==prox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==med)&(SD==medprox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==med)&(SD==med) => (we=mt)(wd=med);
- (SE==prox)&(SF==med)&(SD==distmed) => (we=mt)(wd=pcmed);
- (SE==prox)&(SF==med)&(SD==dist) => (we=mt)(wd=pc);
- (SE==prox)&(SF==distmed)&(SD==prox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==distmed)&(SD==medprox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==distmed)&(SD==med) => (we=mt)(wd=mt);
- (SE==prox)&(SF==distmed)&(SD==distmed) => (we=mt)(wd=mt);
- (SE==prox)&(SF==distmed)&(SD==dist) => (we=mt)(wd=mt);
- (SE==prox)&(SF==dist)&(SD==prox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==dist)&(SD==medprox) => (we=mt)(wd=mt);
- (SE==prox)&(SF==dist)&(SD==med) => (we=mt)(wd=mt);
- (SE==prox)&(SF==dist)&(SD==distmed) => (we=mt)(wd=mt);
- (SE==prox)&(SF==dist)&(SD==dist) => (we=mt)(wd=pc);
- (SE==medprox)&(SF==prox)&(SD==prox) => (we=medmt)(wd=mt);
- (SE==medprox)&(SF==prox)&(SD==medprox) => (we=mt)(wd=mt);
- (SE==medprox)&(SF==prox)&(SD==med) => (we=mt)(wd=med);
- (SE==medprox)&(SF==prox)&(SD==distmed) => (we=mt)(wd=pcmed);
- (SE==medprox)&(SF==prox)&(SD==dist) => (we=mt)(wd=pc).



Portanto, o sistema possui três entradas com universo de discurso de 0 até o alcance máximo do sensor (definido pelo usuário), nesse trabalho definido como 12. Cada entrada é dividida em duas funções de pertinência trapezoidais (relacionadas aos termos linguísticos *prox* de próximo e *dist* de distante) e três triangulares (relacionadas a *medprox*, *med* e *distmed*), que podem ser vistas na Figura 16.

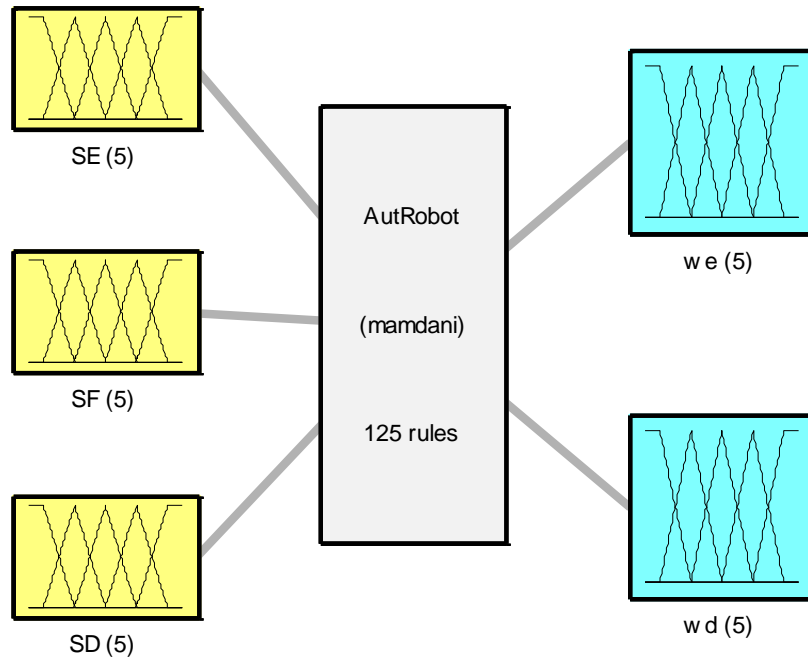
As saídas  $\omega_d$  e  $\omega_e$  seguem o mesmo padrão que as funções de pertinência, as trapezoidais estão relacionadas aos termos *pc* de pouco e *mt* de muito, enquanto as triangulares estão em relação a *pcmed*, *med* e *medmt*. Porém, seu universo de discurso é de [0,1].

Figura 16 – Funções de pertinência



Fonte: Autoria própria.

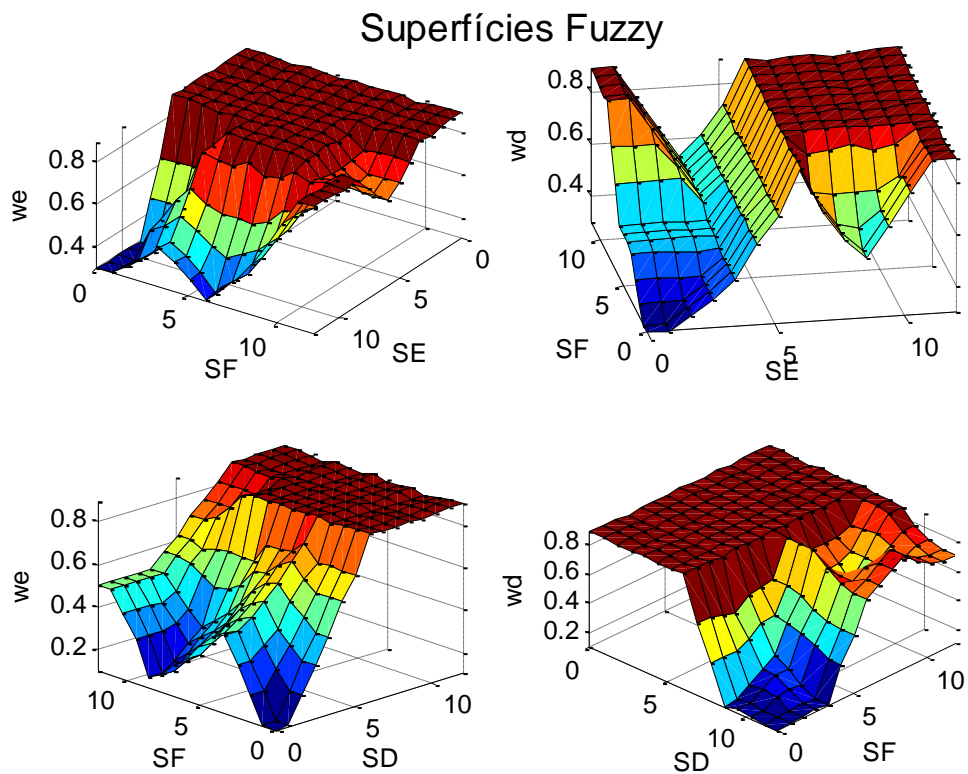
**Figura 17 – Estrutura do sistema *Fuzzy***



**Fonte: Autoria própria.**

A Figura 17 mostra a estrutura do sistema *Fuzzy*, enquanto a Figura 18 mostra a superfície de controle *Fuzzy* obtida através da base de regras.

**Figura 18 – Superfícies de controle do sistema *Fuzzy***

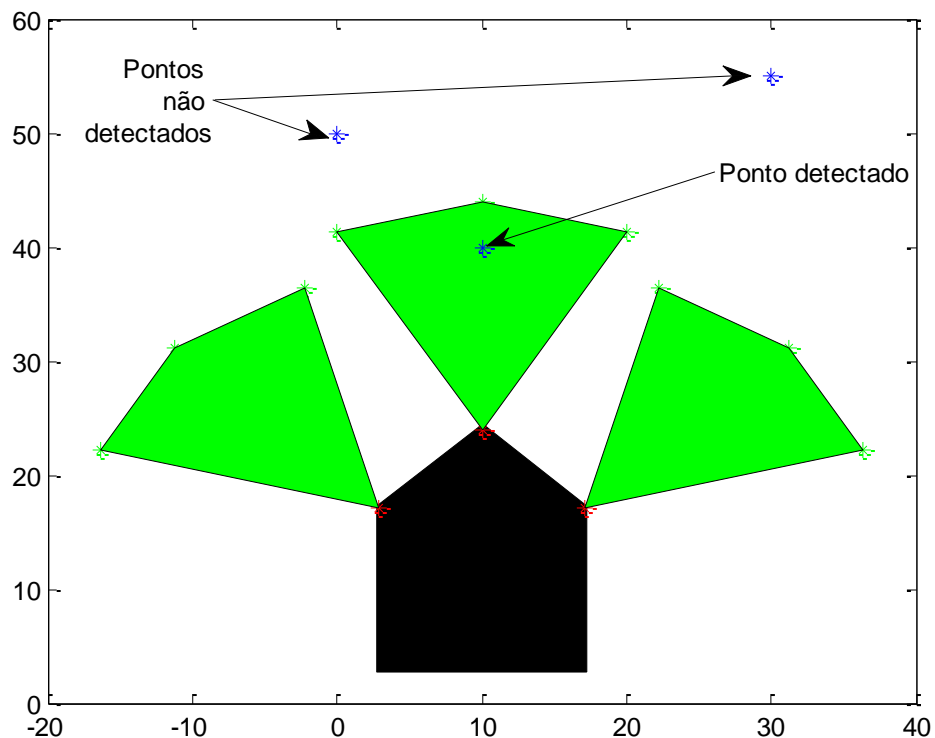


**Fonte: Autoria própria.**

### 3.5 Simulação dos sensores

O explorador tem ao todo três sensores que servem para interagir com o ambiente, detectando obstáculos e alvos, e mapeando o ambiente. As variáveis relacionadas aos sensores ultrassônicos, como por exemplo, alcance e ângulo de abertura, foram testados para garantir que mesmo havendo a presença de obstáculos próximos do alvo, o robô seja capaz de se aproximar e capturá-lo. Esse problema é comum e pode ser encontrado na literatura como no trabalho de Brooks (1985). Apesar disso, ainda existem pontos “cegos” na área de cobertura dos sensores, assim, ao longo das simulações alguns obstáculos podem não ser mapeados ou alvos encontrados em um primeiro momento.

**Figura 19 – Área de alcance dos sensores**



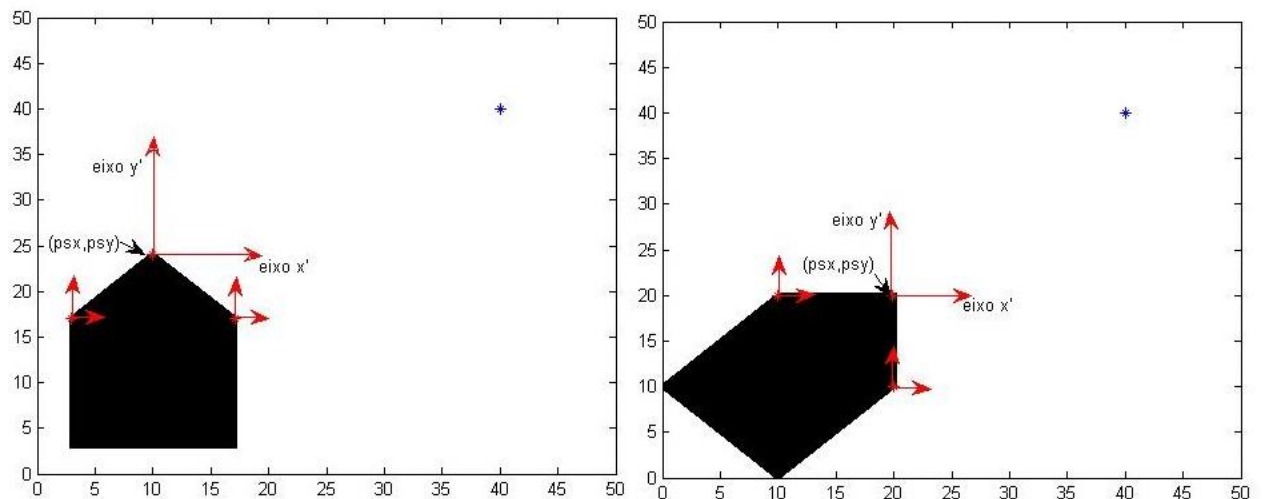
**Fonte: Autoria própria.**

Os alvos e obstáculos são detectados apenas quando adentram a área de atuação dos sensores, região verde vista na Figura 19. Essa região é definida pelo usuário utilizando as variáveis de alcance e ângulo de abertura de captação dos sensores.

Adotando-se um plano cartesiano  $[x,y]$ , é possível gerar informação suficiente para simular a detecção de pontos a partir dos sensores do robô. Para realizar essa função de detecção, a cada iteração é conferida a distância e angulação entre cada um dos pontos em relação a cada um dos sensores do explorador autônomo.

Para o cálculo de distância e ângulo, aplica-se o teorema de Pitágoras em um triângulo formado pelos eixos  $x, y$  e a distância do ponto ao sensor, como é visto na Figura 21. A origem do plano cartesiano  $(0,0)$  está fixa e cada sensor terá seu próprio plano cartesiano auxiliar  $[x',y']$  para cálculo de distância e ângulo.

**Figura 20 – Planos auxiliares com robô em diferentes poses**



Fonte: Autoria própria.

A partir do sensor ultrassônico localizado na frente do robô, obteve-se o plano auxiliar  $[x',y']$  e a posição do sensor  $[psx,psy]$  como visto na Figura 20. Nota-se que os eixos dos planos auxiliares sempre estão paralelos aos eixos do plano cartesiano. Essa simplificação é feita para facilitar os cálculos e diminuir a complexidade da simulação.

O sensor frontal é mostrado como exemplo, porém esses cálculos devem ser realizados para cada um dos sensores por iteração. Primeiro, calculam-se as distâncias  $\Delta x$  e  $\Delta y$ , pelas equações (7) e (8), respectivamente:

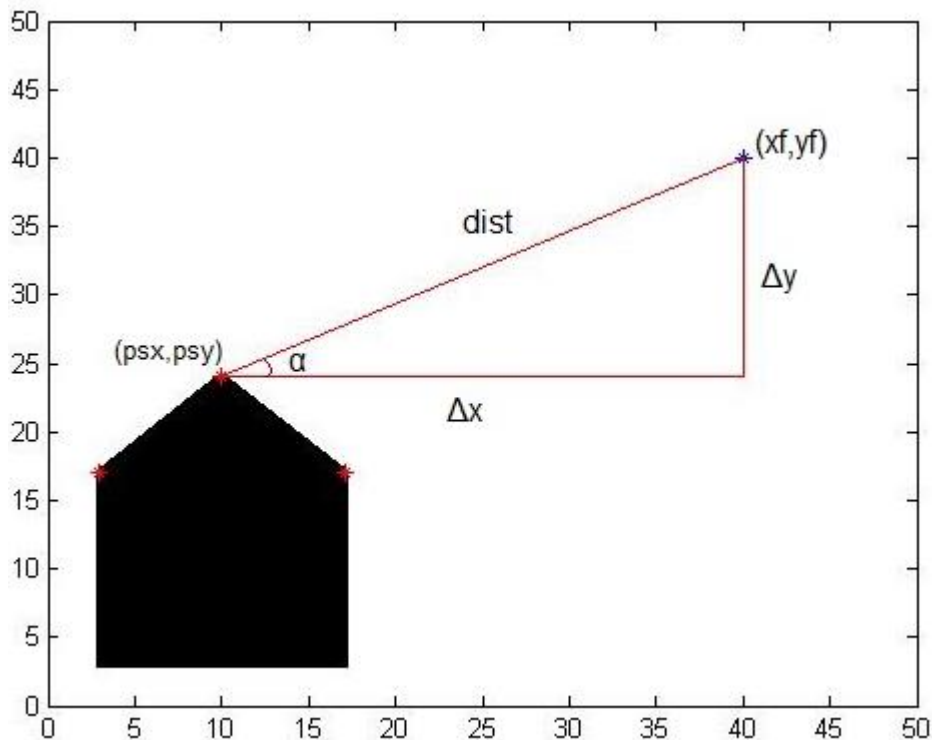
$$\Delta x = x_f - psx \quad (7)$$

$$\Delta y = y_f - psy \quad (8)$$

No qual,  $\Delta x$  e  $\Delta y$  são as distâncias relativas entre o ponto de referência  $(psx, psy)$  do sensor e o obstáculo  $(xf, yf)$ . Assim, calcula-se a distância entre a posição do sensor e o alvo com a equação (9).

$$dist = \sqrt{\Delta x^2 + \Delta y^2} \quad (9)$$

Figura 21 – Distância entre dois pontos



Fonte: Autoria própria.

É possível utilizar o resultado das equações (7), (8) e (9) para obtenção do ângulo  $\alpha$  (ângulo formado entre o alvo e o eixo auxiliar  $x'$ ) com *range* de  $[0^\circ, 360^\circ]$ , como visto na Figura 21. Isso só é possível simplificando o problema, usando o ponto  $(psx, psy)$  como referência ao invés da origem do plano cartesiano. As equações utilizadas para tal são mostradas a seguir:

$$\text{sen } \alpha = \Delta y / dist \quad (10)$$

$$\text{cos } \alpha = \Delta x / dist \quad (11)$$

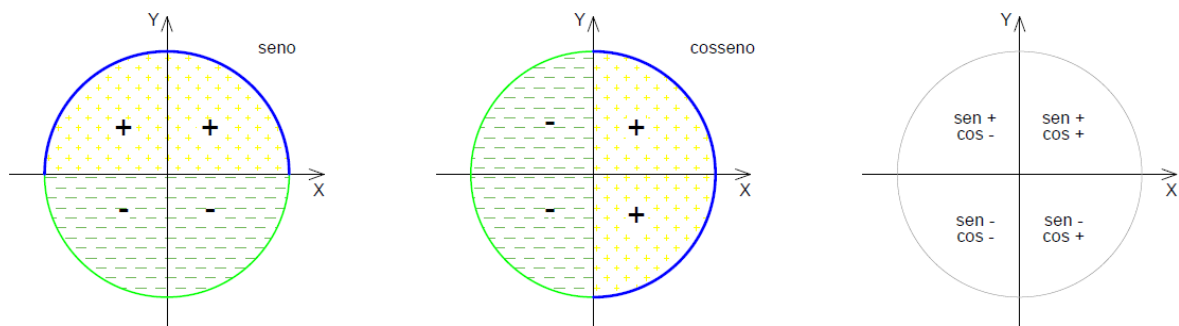
$$\alpha = \text{tg}^{-1}\left(\frac{\Delta y}{\Delta x}\right) \quad (12)$$

Apenas com o resultado da equação (12) não é possível definir o ângulo  $\alpha$  entre o alvo e o eixo  $x'$  com exatidão (ângulos em diferentes quadrantes podem ter o mesmo valor de tangente). Portanto, utiliza-se a informação obtida pelas equações (10) e (11) para definir em qual quadrante o ponto está em relação ao robô. Para isso, utiliza-se o seguinte conjunto de regras para corrigir seu valor:

- Se cosseno de  $\alpha$  é positivo, e o seno de  $\alpha$  é positivo, o alvo está no primeiro quadrante e, portanto não precisa de alterações, o valor do ângulo  $\alpha$  está correto;
- Se cosseno de  $\alpha$  é negativo, e o seno de  $\alpha$  é positivo; o alvo está no segundo quadrante e, portanto deve-se adicionar  $180^\circ$  ao valor do ângulo  $\alpha$ ;
- Se cosseno de  $\alpha$  é negativo, e o seno de  $\alpha$  é negativo; o alvo está no terceiro quadrante e, portanto deve-se adicionar  $180^\circ$  ao valor do ângulo  $\alpha$ ;
- Se cosseno de  $\alpha$  é positivo, e o seno de  $\alpha$  é negativo; o alvo está no quarto quadrante e, portanto deve-se adicionar  $360^\circ$  ao valor do ângulo  $\alpha$ .

Os fundamentos dessa base de regras podem ser vistos nos círculos trigonométricos da Figura 22.

**Figura 22 – Quadrantes em função de senos e cossenos**



Fonte: Retirado de Silva (2015).

Com as equações apresentadas nessa seção é possível obter as informações necessárias para simular os sensores assim como eles funcionam em

um ambiente real. Sua função de detecção pode ser observada na Figura 19, funcionando da seguinte maneira:

- Se um ponto tiver a distância calculada menor que o alcance do sensor e o ângulo  $\alpha$  estiver dentro do ângulo de captação do sensor, este ponto será mapeado ou será marcado para captura (no caso de alvos). Esse evento ativará o envio de informação do sensor para o controlador do robô, ativando algum modo de operação desejado;
- Se um ponto tiver a distância calculada maior que o alcance e/ou o ângulo  $\alpha$  estiver fora do ângulo de captação do sensor, nada acontecerá e o ponto continuará oculto.

Após o obstáculo ser detectado, ele gera um sinal referente à distância entre o ponto e o sensor que o detectou. As variáveis utilizadas para isso foram: SE (sensor esquerdo), SD (sensor direito) e SF (sensor frontal). Elas são necessárias para caso haja ativação do modo *desvio de obstáculo*, o explorador tenha as informações de entrada do sistema *Fuzzy*, como é visto na Seção 3.4.

### 3.6 Modelo robótico simulado

Como mencionado anteriormente, a simulação é realizada no ambiente do *software* MatLab®. O robô autônomo tem duas rodas fixas nas laterais com um servo-motor dedicado a cada uma e três sensores localizados estrategicamente. Diversas variáveis foram ajustadas no simulador para que houvessem resultados mais próximos da realidade. As definições das variáveis foram:

- **Posição cartesiana inicial dos robôs:** definida e adequada posteriormente em relação a cada um dos ambientes;
- **Direção inicial do robô:** quando utilizado apenas um robô, foi definido sempre como  $0^\circ$ , com a presença de mais indivíduos, foram inicializados em pares de frente um ao outro.
- **Distância de segurança do robô:** para garantir sua integridade frente a um obstáculo, foi definido como 5 unidades de distância da simulação;

- **Dimensões do robô:** raio da roda e comprimento da base, ambos de extrema importância para o modelo cinemático, foram definidos como 0,8333 e 4,8 respectivamente;
- **Alcance dos sensores e ângulo de captação:** necessário para saber o quão distante um alvo será detectado e se ele está dentro da área de alcance dos sensores, foram definidos como 12 unidades de distância e 45°;
- **Direção de cada sensor em relação à direção inicial do robô:** o sensor frontal (SF) será sempre igual a direção atual do robô, enquanto os sensores esquerdo (SE) e direito (SD) estarão defasados 45° positivamente e negativamente em relação ao frontal;
- **Erro desejado de captura:** o centro de massa do robô deve estar a 3 unidades de distância do alvo para ele ser considerado capturado.
- **Diferenciação:** caso exista mais de um indivíduo no ambiente de teste, eles podem ser diferenciados pelas cores de suas trilhas (se houver permissão para plotagem do rastro).

Uma das principais funções extras da simulação é a permissão do usuário de realizar a simulação com ruídos nos sensores, para uma melhor avaliação do modelo cinemático e das variáveis escolhidas.

### 3.7 Parâmetros de teste

Para cada ambiente foi realizado inicialmente um teste utilizando apenas um robô e, posteriormente com quatro robôs. Foram aferidos o tempo de processamento da simulação com o auxílio dos comandos *tic* e *toc*, e o tempo de vídeo, simulando um parâmetro de tempo real utilizando as funções *getframe* e *VideoWrite* com trinta quadros por segundo. Além disso, gerou-se um gráfico com a relação dos pulsos gerados ao longo das simulações.



## 4 RESULTADOS

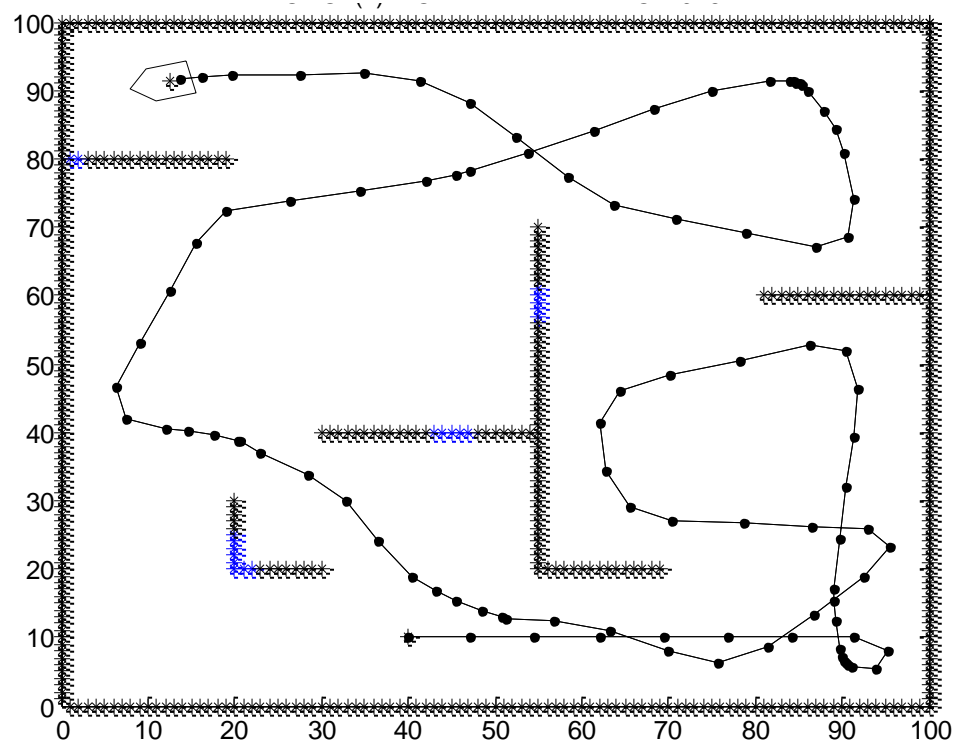
Os resultados de cada ambiente são apresentados separadamente em cada subseção para facilitar a visualização e, ao final, há uma tabela compilando todos os resultados para fins de comparação. Além disso, os pulsos resultantes dos exploradores nos testes realizados são comentados na Subseção 4.4.

### 4.1 Resultados do ambiente de teste I

Primeiramente, foi obtido o resultado referente a apenas um robô com posição inicial no ponto [40,10] e direção de  $0^\circ$ . Ele resgatou todas as vítimas ao longo de 976 iterações, com um tempo de processamento de 90,698207 segundos, ou seja, 1 minuto e 31 segundos aproximadamente.

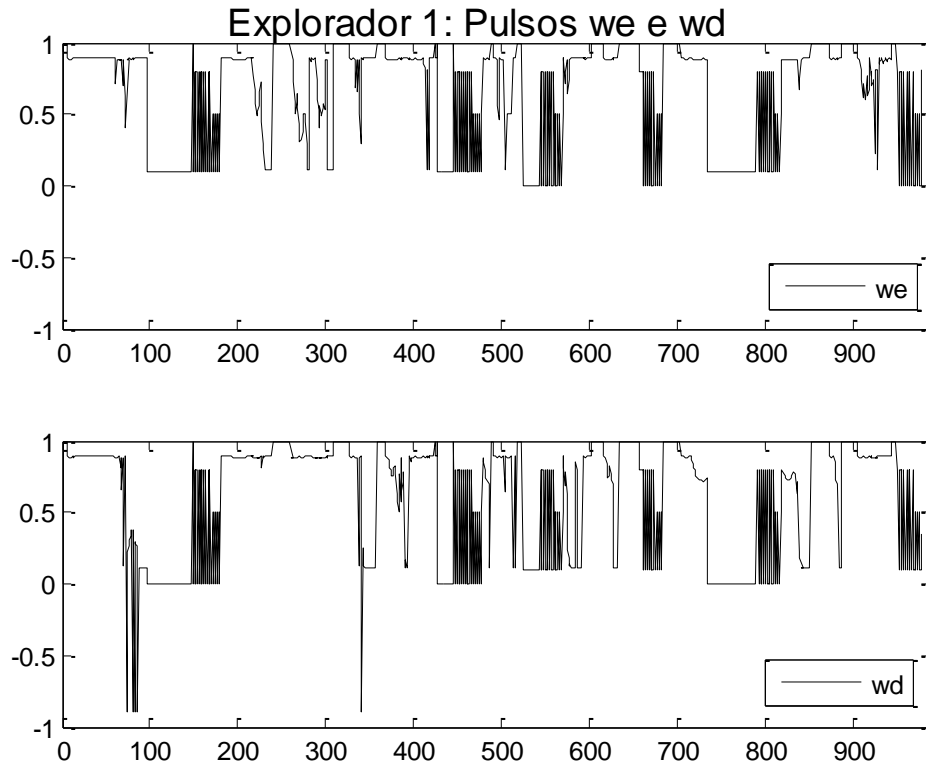
O vídeo gerado por essa simulação tem uma duração de 32 segundos. Além disso, pode-se observar na Figura 23 que o robô mapeou a maioria dos obstáculos ao longo da sua trajetória. O vídeo gerado pode ser acessado pelo link: <https://youtu.be/IOIFOFHOBd0>.

Figura 23 – Simulação do ambiente I com um explorador



Fonte: Autoria própria.

**Figura 24 – Pulsos do explorador na simulação do ambiente I**



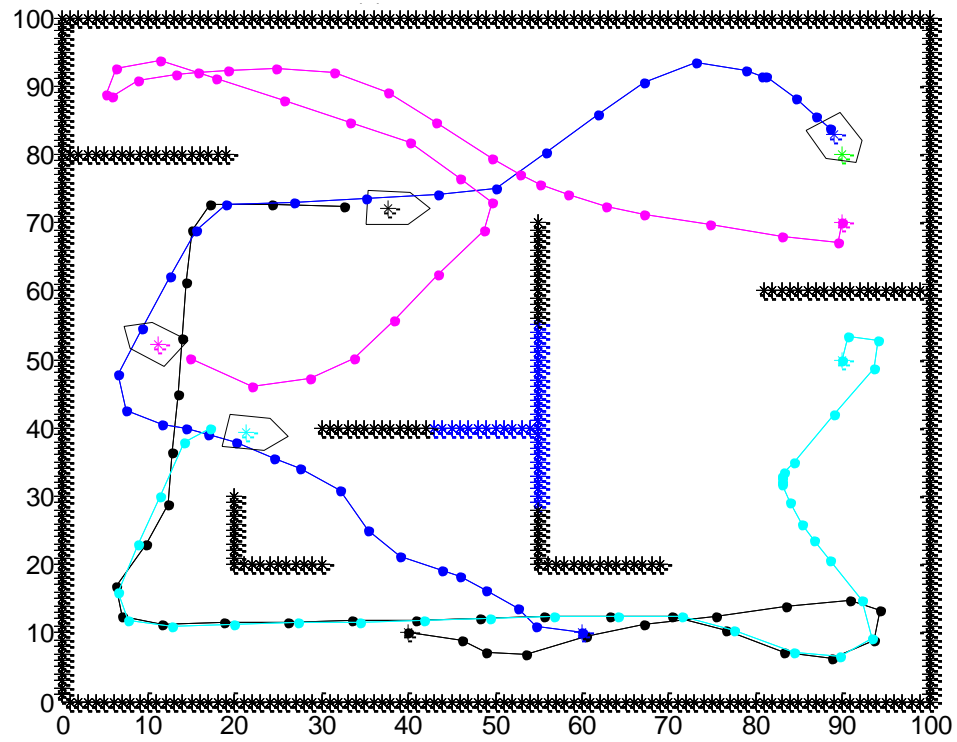
**Fonte: Autoria própria.**

Posteriormente, utilizaram-se quatro exploradores com suas posições iniciais sendo  $[40,10]$ ,  $[60,10]$ ,  $[90,50]$ ,  $[90,70]$  e direções  $0^\circ$ ,  $180^\circ$ ,  $90^\circ$ ,  $270^\circ$ , respectivamente. Assim, neste ambiente os exploradores 1 (trilha preta) e 2 (trilha azul) inicializaram de frente um para o outro, assim como o 3 (trilha azul claro) e o 4 (trilha magenta).

Esse teste teve ao todo 346 iterações com um tempo de processamento de 176,705741 segundos, ou seja, aproximadamente 2 minutos e 57 segundos, gerando um vídeo de 11 segundos até que todas as vítimas fossem resgatadas. Apenas os obstáculos da região central do ambiente não foram mapeados pelos robôs, pois nenhum esteve próximo dali, como pode ser observado na Figura 25 ou no vídeo pelo link: <https://youtu.be/nbTAUni8jso>.

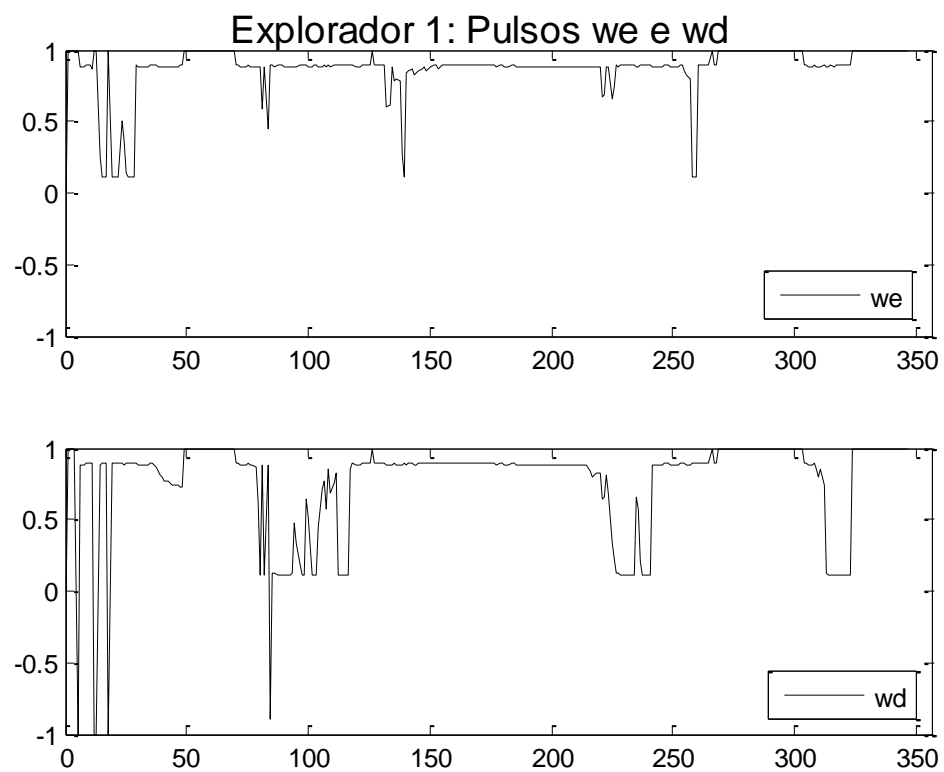
As Figuras 26, 27, 28 e 29 mostram os pulsos nas rodas esquerda e direita. As curvas são das mesmas cores utilizadas para diferenciar as trilhas dos exploradores para que seja possível acompanhar seus movimentos.

Figura 25 – Simulação do ambiente I com múltiplos exploradores



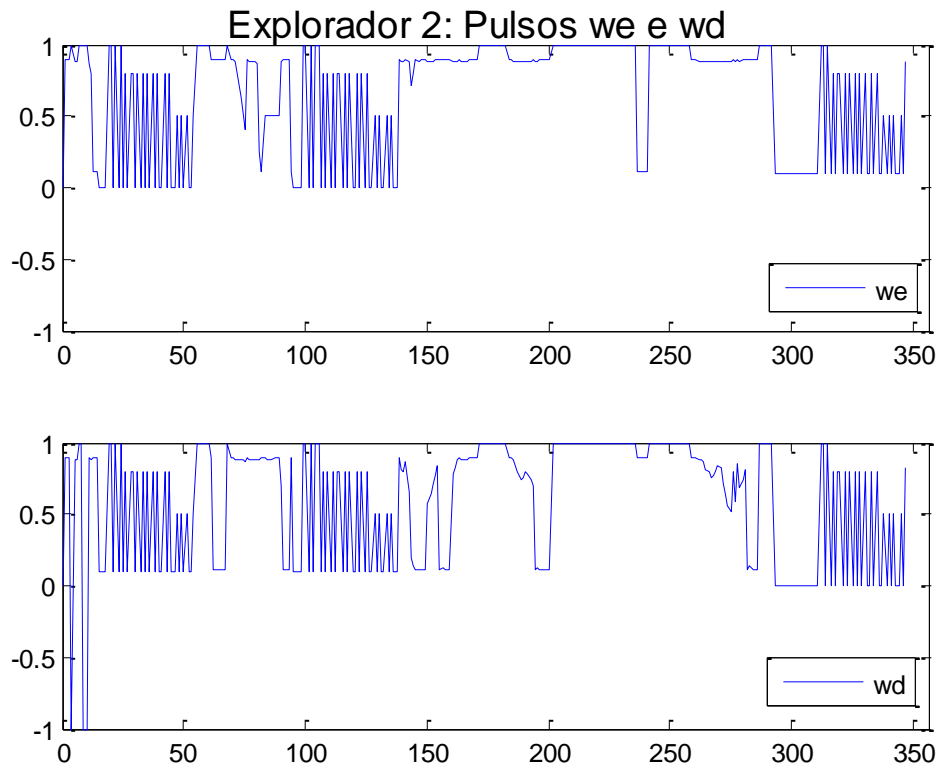
Fonte: Autoria própria.

Figura 26 – Pulsos do explorador 1 na simulação do ambiente I



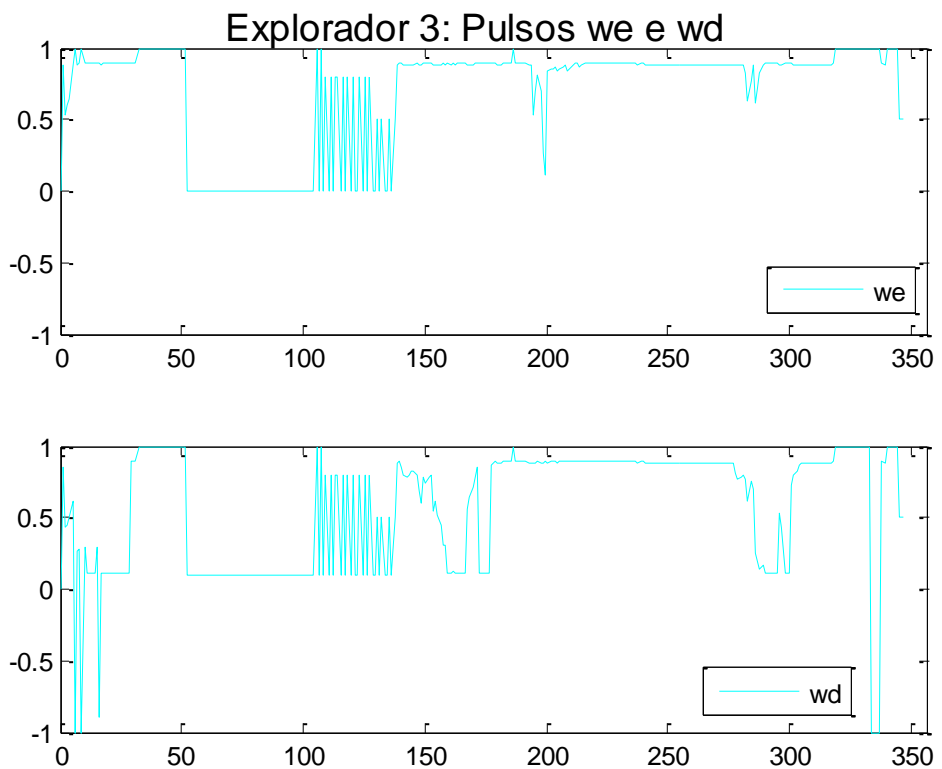
Fonte: Autoria própria.

**Figura 27 – Pulsos do explorador 2 na simulação do ambiente I**



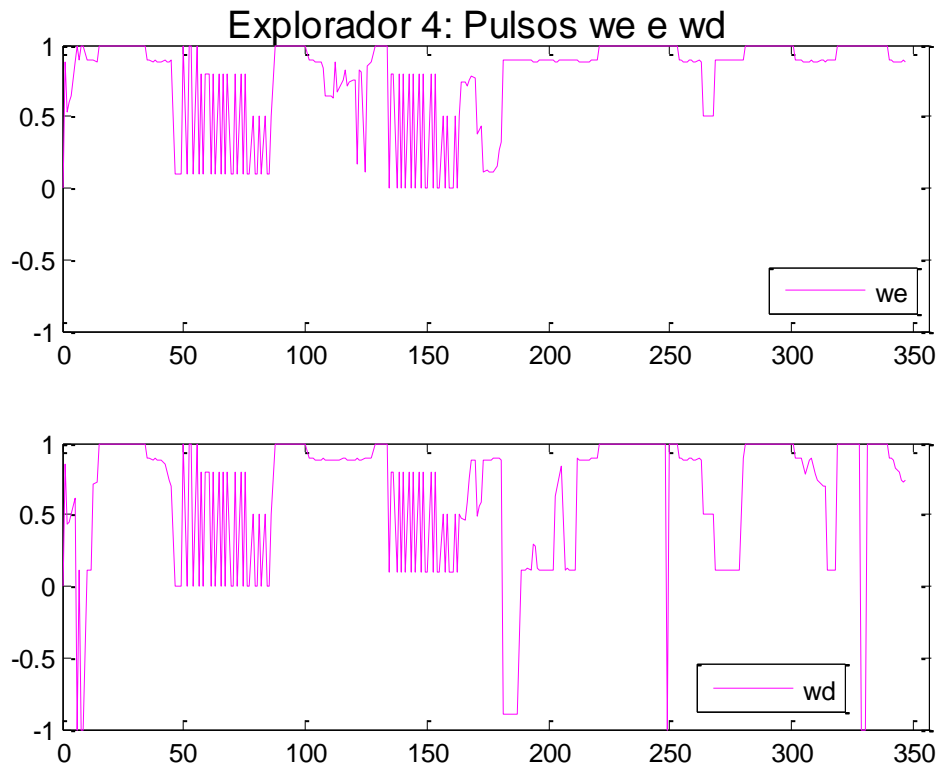
Fonte: Autoria própria.

**Figura 28 – Pulsos do explorador 3 na simulação do ambiente I**



Fonte: Autoria própria.

**Figura 29 – Pulsos do explorador 4 na simulação do ambiente I**



Fonte: Autoria própria.

## 4.2 Resultados do ambiente de teste II

Ao utilizar apenas um robô com posição inicial no ponto [40,10] e direção de  $0^\circ$ , foi obtido um tempo de processamento no resgate das vítimas de 874,987265 segundos, ou seja, aproximadamente 14 minutos e 35 segundos durante 6778 iterações. O vídeo, por sua vez, teve duração de 3 minutos e 45 segundos.

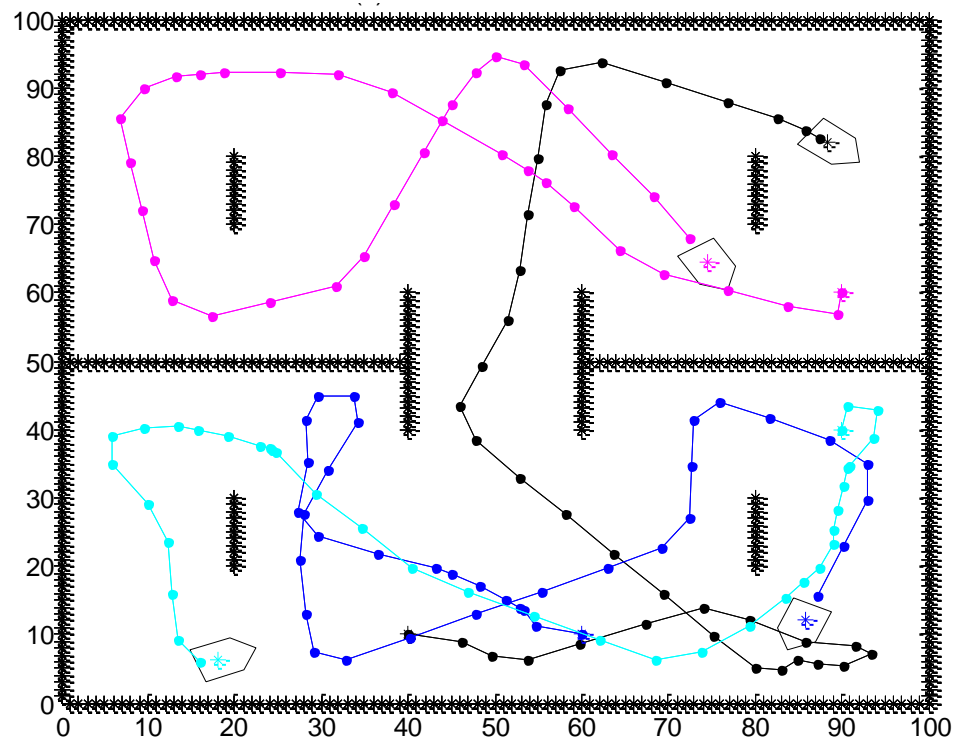
Quase todos os obstáculos foram mapeados como visto na Figura 30. Pode-se observar também que o explorador ficou vagando em busca de alvos quase todo o tempo na região inferior, até entrar em um estado que gerou os pulsos necessários para seguir em direção a passagem no centro do ambiente de teste. Os pulsos do explorador podem ser vistos na Figura 31. O vídeo do teste para melhor visualização pode ser acessado pelo link: <https://youtu.be/nr77Lyl0qRI>.



Utilizando-se quatro robôs com posições iniciais em [40,10], [60,10], [90,40], [90,60] e direções  $0^\circ$ ,  $180^\circ$ ,  $90^\circ$ ,  $270^\circ$ , o teste teve ao todo 365 iterações com um tempo de processamento de 167,111634 segundos, ou seja, aproximadamente 02 minutos e 47 segundos, gerando um vídeo de 12 segundos até que todas as vítimas fossem resgatadas.

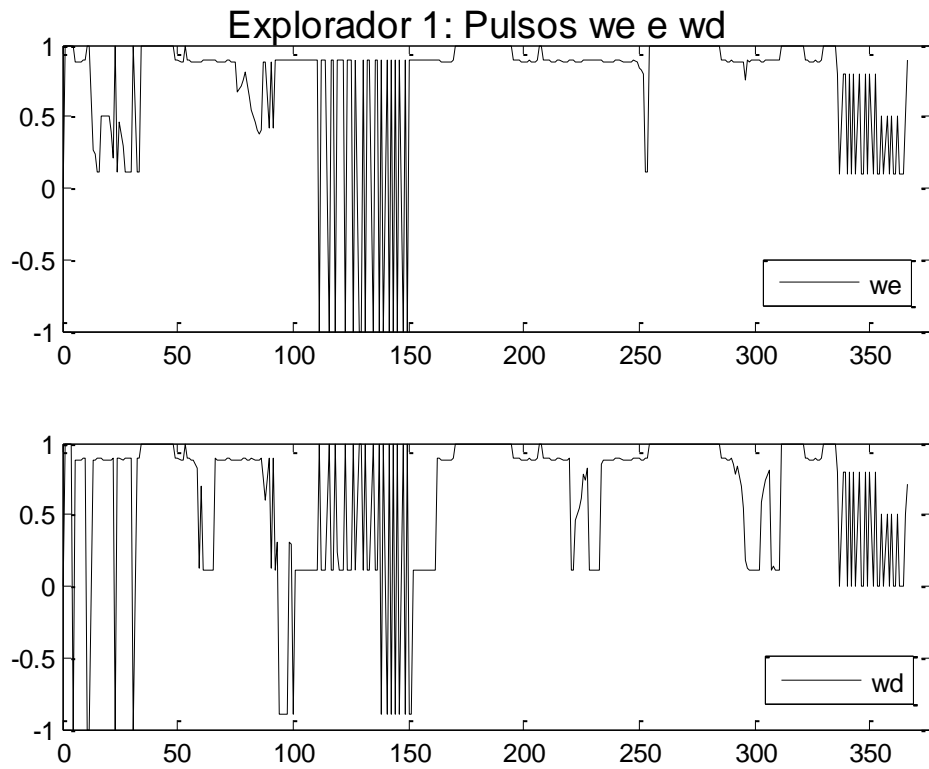
Na Figura 32, observa-se que todos os obstáculos desse ambiente foram mapeados, e os pulsos gerados por cada um dos exploradores estão nas Figuras 33, 34, 35 e 36. O vídeo do teste está disponível pelo link: <https://youtu.be/6Kz4BwN-rY0>.

**Figura 32 – Simulação do ambiente II com múltiplos exploradores**



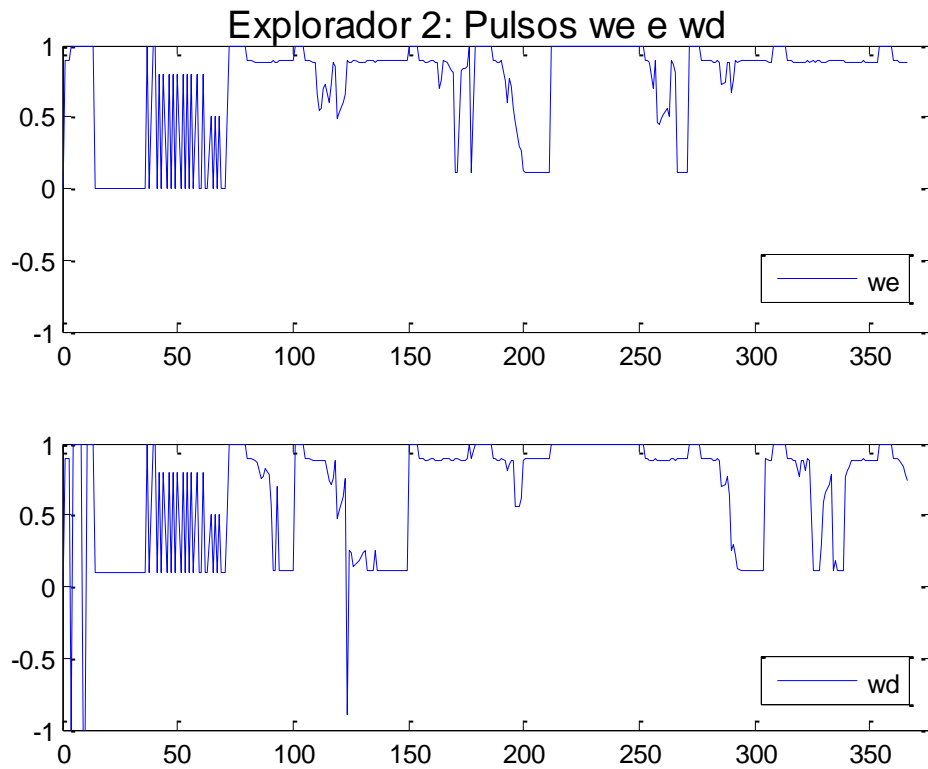
Fonte: Autoria própria.

**Figura 33 – Pulsos do explorador 1 na simulação do ambiente II**



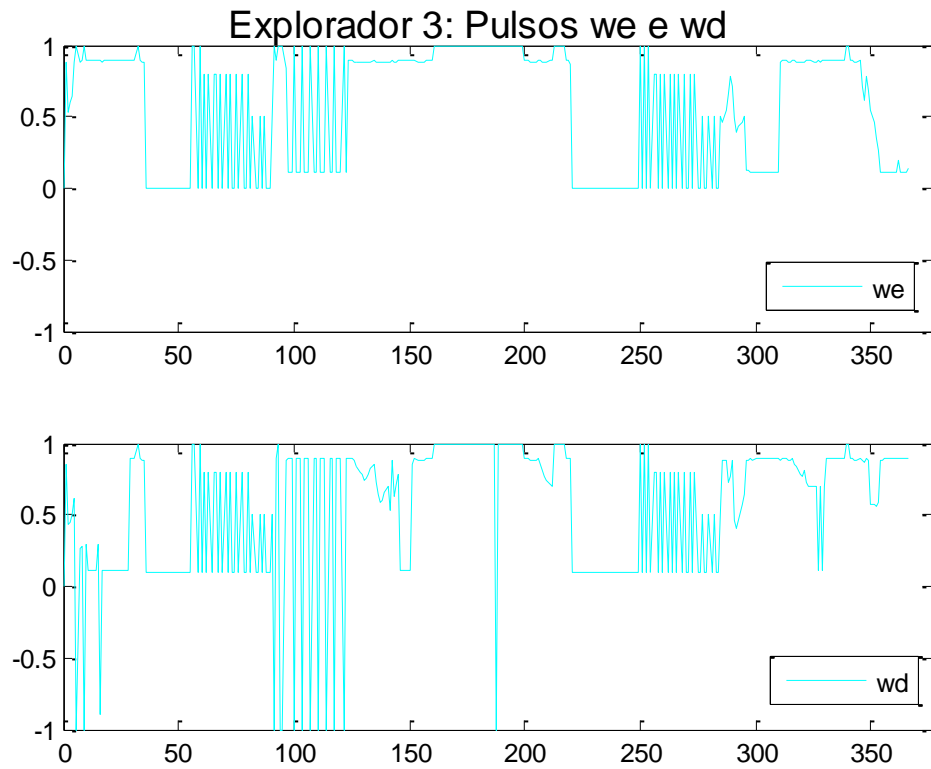
**Fonte: Autoria própria.**

**Figura 34 – Pulsos do explorador 2 na simulação do ambiente II**

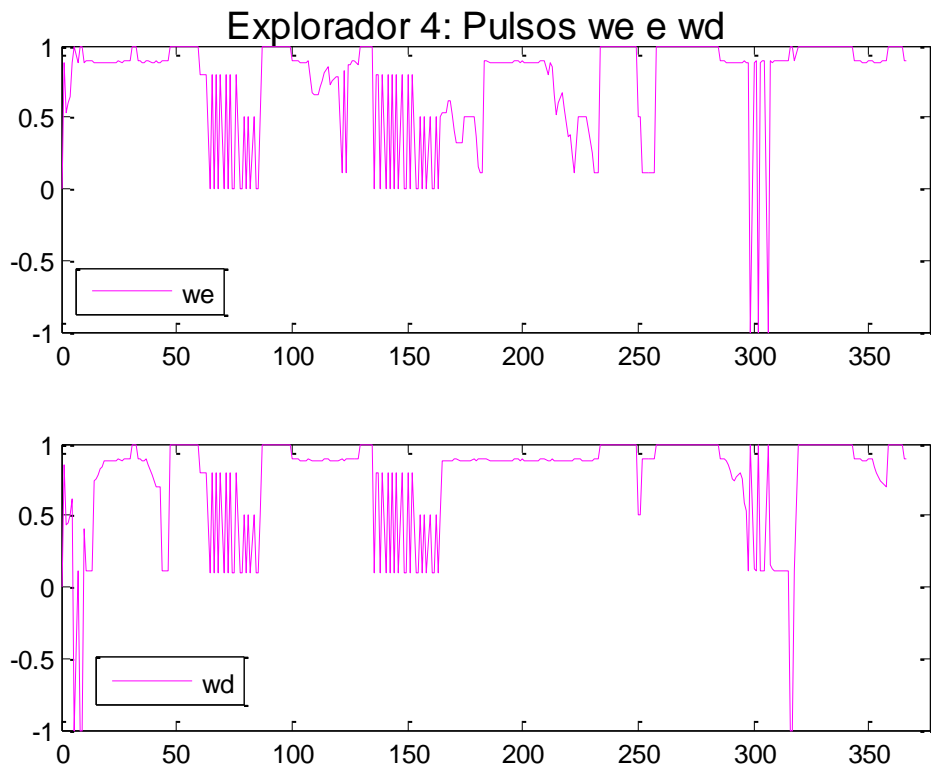


**Fonte: Autoria própria.**



**Figura 35 – Pulsos do explorador 3 na simulação do ambiente II**

Fonte: Autoria própria.

**Figura 36 – Pulsos do explorador 4 na simulação do ambiente II**

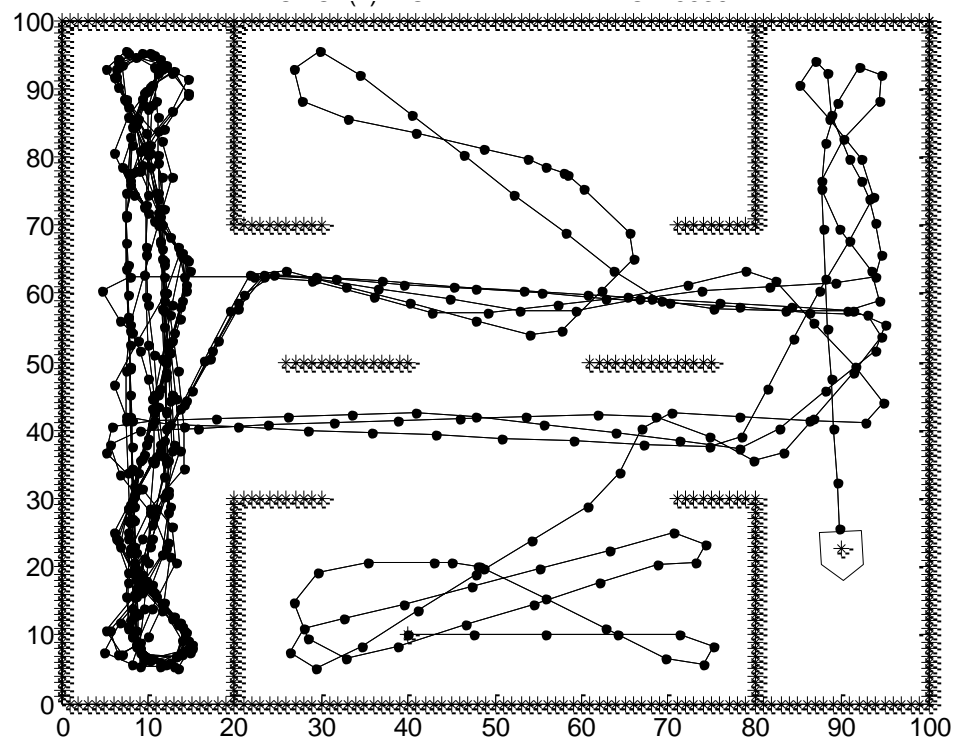
Fonte: Autoria própria.

### 4.3 Resultados do ambiente de teste III

Por fim, foi obtido o resultado com apenas um robô de posição inicial [40,10] e direção de  $0^\circ$  no terceiro ambiente. Ele resgatou todas as vítimas ao longo de 5536 iterações, com um tempo de processamento de 694,003522 segundos, ou seja, 11 minutos e 34 segundos aproximadamente.

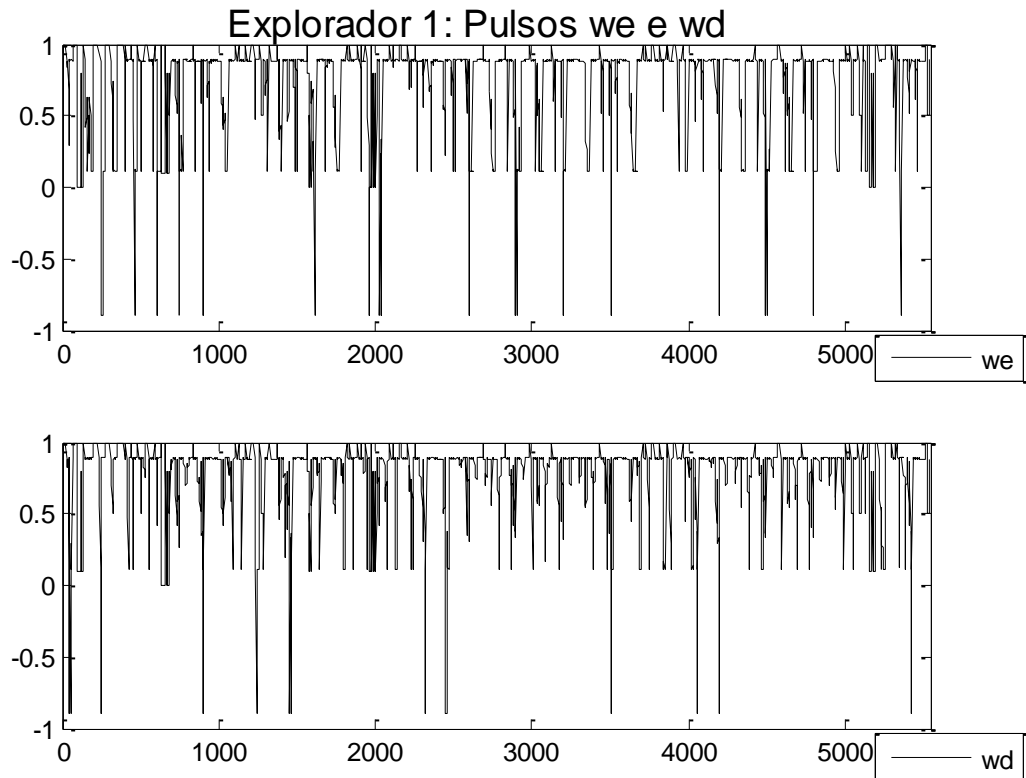
O vídeo gerado por essa simulação tem uma duração de 03 minutos e 04 segundos e pode ser acessado pelo link: [https://youtu.be/4\\_Vn9qcSnJk](https://youtu.be/4_Vn9qcSnJk). A Figura 37 mostra que o robô mapeou todos os obstáculos do ambiente ao longo da sua trajetória. Porém, nota-se que ele ficou mais tempo no corredor esquerdo do ambiente III do que nas demais. Seus pulsos são mostrados na Figura 38.

Figura 37 – Simulação do ambiente III com um explorador



Fonte: Autoria própria.

**Figura 38 – Pulsos do explorador na simulação do ambiente III**

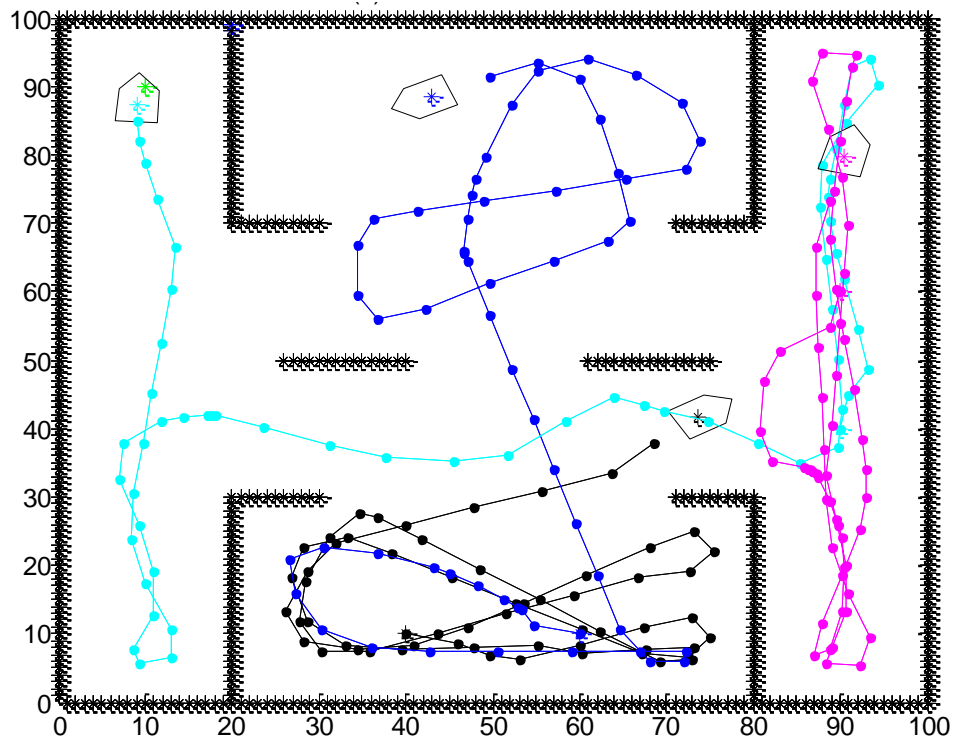


**Fonte: Autoria própria.**

Posteriormente, utilizaram-se quatro robôs com posições iniciais [40,10], [60,10], [90,40], [90,60] e direções  $0^\circ$ ,  $180^\circ$ ,  $90^\circ$ ,  $270^\circ$ , respectivamente. Foram 599 iterações com um tempo de processamento de 300,9427 segundos, ou seja, aproximadamente 5 minutos e 01 segundo, gerando um vídeo de 19 segundos até que todas as vítimas fossem resgatadas.

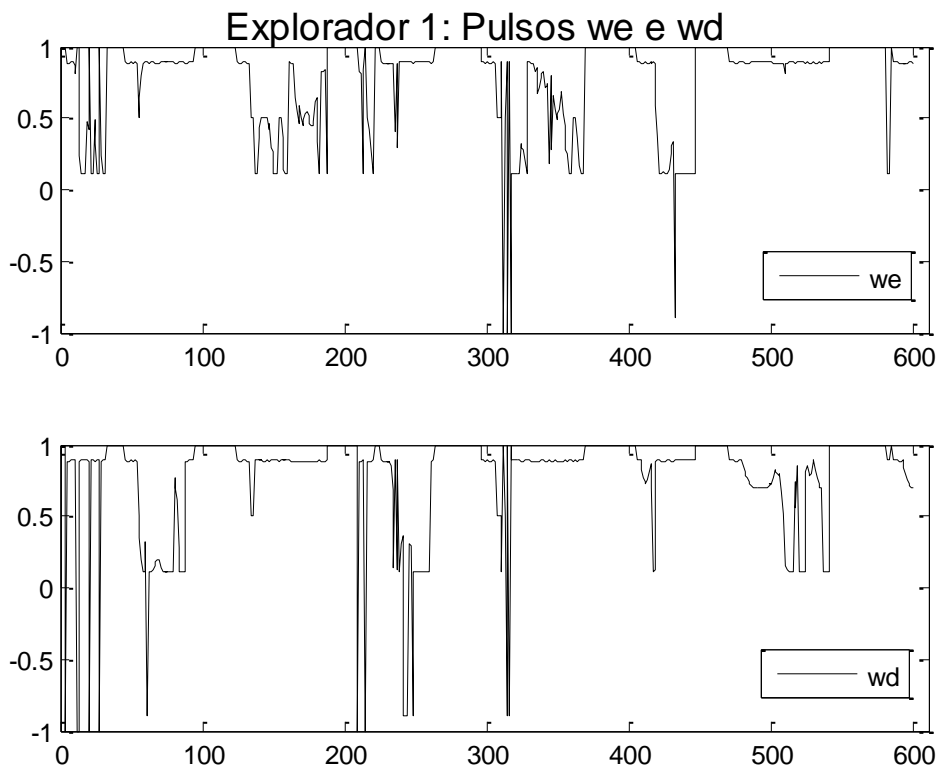
O vídeo está disponível no link: <https://youtu.be/BftkdVsNYTw>. Nesse ambiente, foram mapeados quase todos os obstáculos, faltando apenas um não identificado na posição [20,99] como pode ser observado na Figura 39. Nas Figuras 40, 41, 42 e 43 são mostrados os pulsos gerados pelos controladores de cada um dos exploradores ao longo da simulação.

**Figura 39 – Simulação do ambiente III com múltiplos exploradores**

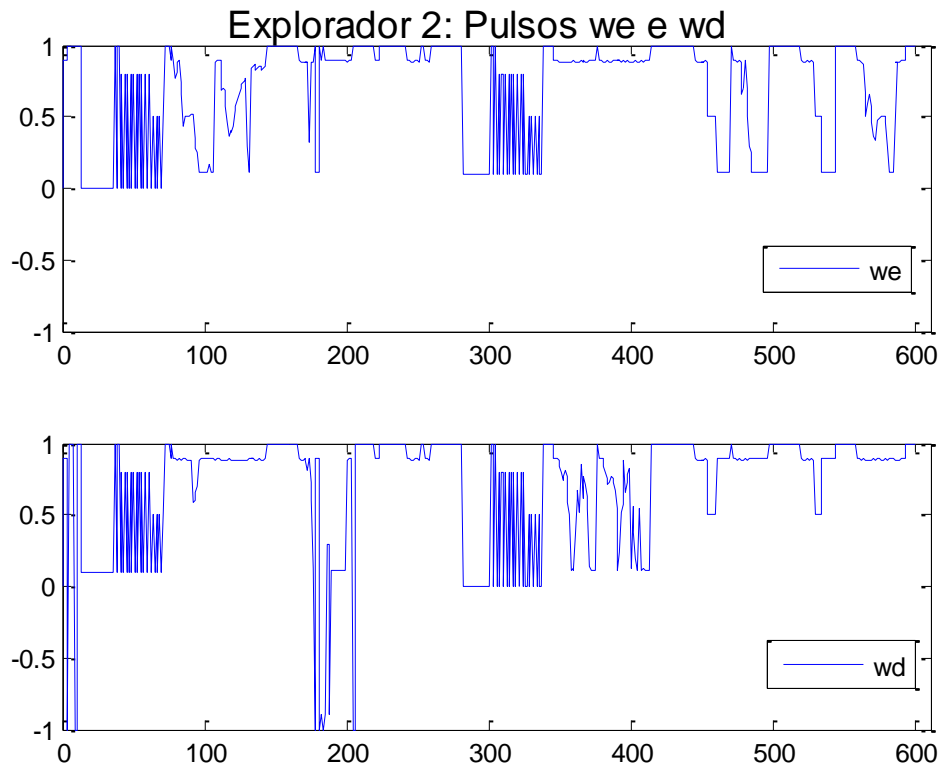


Fonte: Autoria própria.

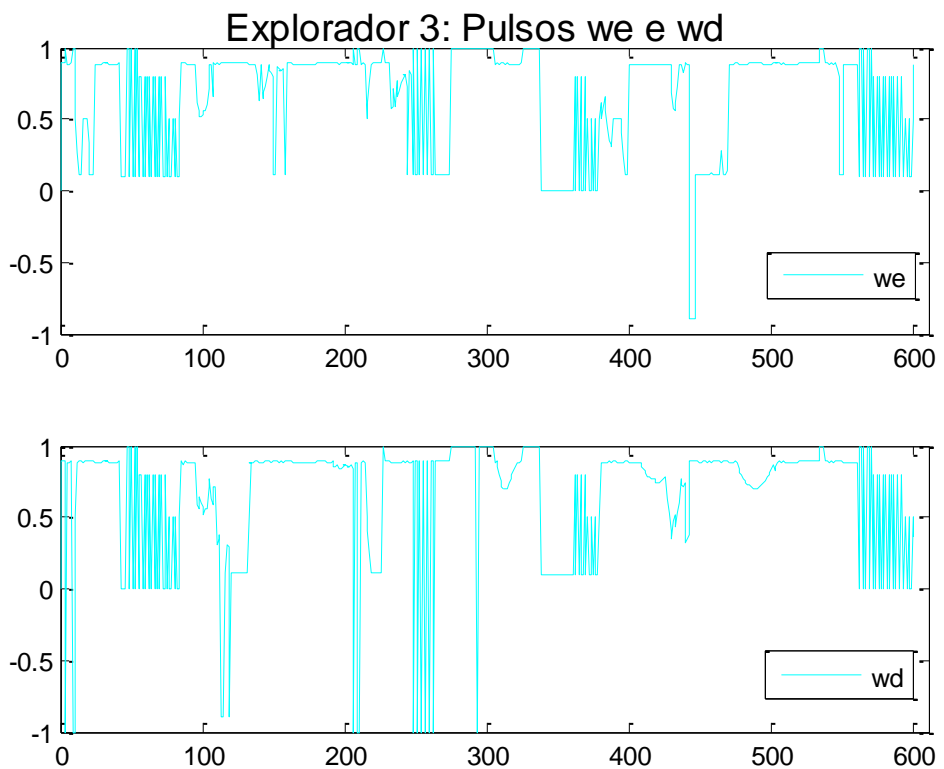
**Figura 40 – Pulsos do explorador 1 na simulação do ambiente III**



Fonte: Autoria própria.

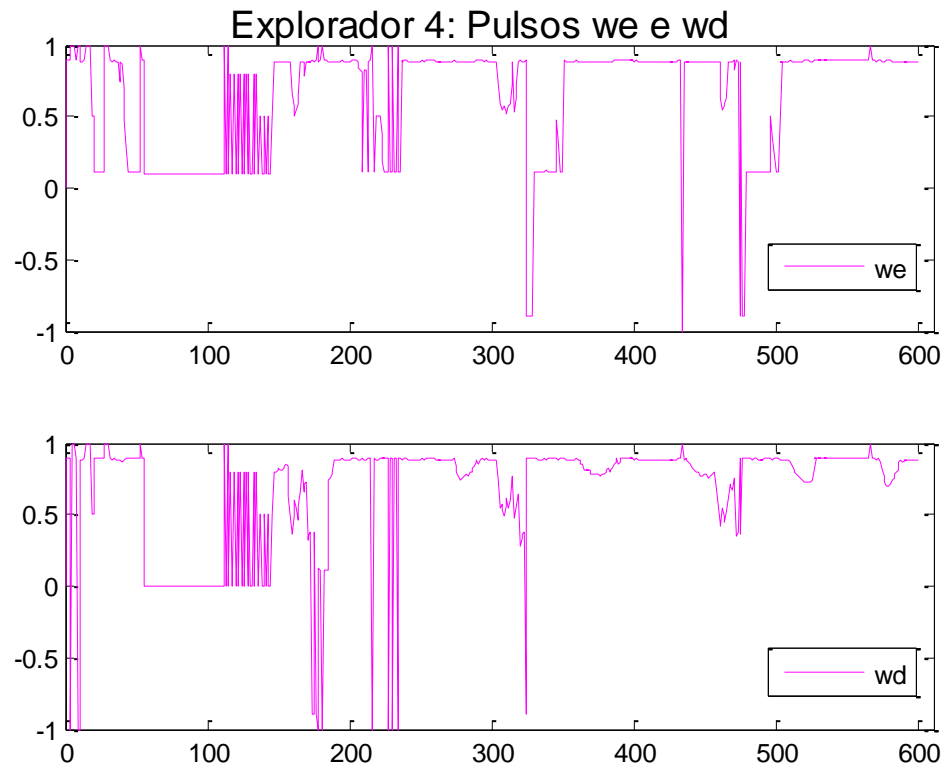
**Figura 41 – Pulsos do explorador 2 na simulação do ambiente III**

Fonte: Autoria própria.

**Figura 42 – Pulsos do explorador 3 na simulação do ambiente III**

Fonte: Autoria própria.

**Figura 43 – Pulsos do explorador 4 na simulação do ambiente III**



**Fonte: Autoria própria.**

#### 4.4 Resultados gerais e discussões

Como pôde ser observado nas subseções anteriores, todos os testes obtiveram sucesso, mesmo que alguns ambientes tenham levado mais tempo que outros. Isso se dá pela posição dos obstáculos, formando cantos e bloqueando o caminho para as vítimas, ou até mesmo o encontro com outros exploradores causando uma mudança de rota. Mas o que acontece ao longo da simulação ou em um caso real de resgate em uma tragédia não pode ser previsto, os robôs ou resgatistas devem estar preparados para reagir da melhor maneira possível frente às dificuldades.

Analisando os vídeos gerados pode-se observar diversas situações interessantes as quais os exploradores estão sujeitos, como na simulação do ambiente II com quatro exploradores, no qual os de trilha preta e trilha azul claro alternam entre desviar um do outro e desviar dos obstáculos próximos, até que um deles consiga sair daquele estado. Enquanto que, no vídeo da simulação do ambiente III com um único explorador, pode-se observar que mesmo o robô

passando perto o suficiente para capturar o alvo na posição [10,90] ele não o fez, isso porque o alvo passou pelo ponto “cego” entre os sensores frontal e direito do robô. O explorador foi ao outro lado do ambiente e só quando retornou ao corredor esquerdo que detectou a vítima e a resgatou.

A Tabela 1 contém todos os resultados compilados que foram apresentados nas subseções anteriores. Primeiramente analisando o tempo de processamento aferidos pelas funções *tic* e *toc*, nota-se que no ambiente I a simulação com um único explorador foi mais rápida, isso se dá pelo fato do algoritmo ter que rodar 4 vezes mais ao utilizar 4 exploradores. Mesmo com esse fator, os tempos de processamento dos ambientes II e III utilizando múltiplos exploradores foram muito mais rápidos. Entretanto, para analisar os resultados de desempenho do resgate em um caso real deve-se observar os resultados em relação a quantidade de iterações e tempo de vídeo.

É notável a redução de iterações e tempo de vídeo ao utilizar múltiplos exploradores, isso acontece por ter mais indivíduos realizando a mesma função, buscando vítimas em diferentes lugares do ambiente ao mesmo tempo.

Tabela 1 - Síntese dos resultados						
Ambiente	1 robô			4 robôs		
	Processamento tic/toc	Tempo de vídeo	Iterações	Processamento tic/toc	Tempo de vídeo	Iterações
I	01'31"	32"	976	02'57"	11"	346
II	14'35"	03'45"	6778	02'47"	12"	365
III	11'34"	03'04"	5536	05'01"	19"	599

Fonte: Aatoria própria.

Analisando as figuras dos pulsos dos exploradores em cada um dos testes, pode-se acompanhar a rota que eles realizaram sabendo que:

- Pulsos positivos de igual valor impulsionaram o robô para frente em linha reta;
- Quando o pulso  $\omega_d$  for maior que o  $\omega_e$ , o robô realiza um movimento de giro para a esquerda;
- Quando o pulso  $\omega_e$  for maior que o  $\omega_d$ , o robô realiza um movimento de giro para a direita;

- Um pulso positivo e outro negativo significa que o robô gira com uma alteração de direção muito maior (pulsos negativos foram utilizados apenas em estados de iminência de colisão).



## 5 CONSIDERAÇÕES FINAIS

Nesse trabalho, foram criados três ambientes de simulação de resgate de vítimas, nos quais robôs autônomos foram capazes de encontrá-las e resgatá-las em um ambiente com diversos obstáculos fixos e em alguns casos dinâmicos (outros robôs).

Averiguou-se que em testes no qual foram utilizados múltiplos robôs, eles realizaram a tarefa em um tempo consideravelmente menor. Esse melhor desempenho em casos reais pode ser a diferença entre a vida e a morte de vítimas acidentadas.

Em nenhum dos testes houve colisões com os obstáculos ou com os demais robôs, e todas as vítimas foram encontradas e resgatadas, assim pode-se sugerir que o modelo robótico utilizado neste trabalho apresenta características autônomas. Em nenhum dos testes houve interferência do usuário durante a simulação, os robôs eram responsáveis por tomar suas próprias decisões de qual direção seguir baseado nas informações obtidas localmente pelos seus sensores. Apesar do modo de operação *captura de alvos*, a arquitetura do robô é reativa, pois não utiliza informações passadas dos sensores, apenas informações locais.

Conclui-se também ao observar os vídeos dos testes que a hierarquia dos modos de operação e o sistema de alternância entre eles, assim como a lógica utilizada em cada um deles funcionaram como o esperado, garantindo maior robustez do robô a situações imprevistas e permitindo que o mesmo concluísse seus objetivos.

Nos testes com múltiplos indivíduos, pode-se concluir que apesar de serem fisicamente simples (dois servo-motores e três sensores), eles possuem alta escalabilidade. Isto é, mesmo quando um ou mais robôs estão presos em uma sequência de estados repetitivos (navegando na mesma rota), os demais continuam a operar normalmente e realizar suas funções.

Como futuros trabalhos, sugere-se aplicar *Fuzzy Cognitive Maps* (FCM) ou outras técnicas de sistemas inteligentes para gerar as saídas dos modos de operação com menor tempo de processamento, criação de ambientes maiores e mais complexos capazes de abrigar uma quantidade maior de indivíduos. Também pode-se aplicar diferentes técnicas de robótica de enxame como *Group Explosion*, ou movimentos coordenados entre os robôs próximos. Também se pode estabelecer

uma central no qual os robôs devem levar os alvos encontrados, ao invés de apenas aproximar-se e capturá-los.

## REFERÊNCIAS

- ABIDIN, Z. Z.; ARSHAD, M. R.; NGAH, U. K. An introduction to swarming robotics: application development trends. **Artificial Intelligence Review**, v. 43, n. 4, p. 501–514, 2013.
- ALSINA, P. J. **ELE 440 - Sistemas robóticos autônomos**. Notas de aula, apostila, cap. 2. UFRN, Natal, 2016.
- ALWAN, M. et al. **Combining Goal-Directed, Reactive and Reflexive Navigation in Autonomous Mobile Robots**. 1996 IEEE Australia and New Zealand Conference on Intelligent Information Systems, 1996.
- BRAITENBERG, V. **Vehicles: Experiments in Synthetic Psychology**. The MIT PRESS, 1986.
- BROOKS, R. A. **A Mobile Robot Project**. n. February, p. 31, 1985.
- BROOKS, R. A. **Intelligence without representation**. Artificial Intelligence, v. 47, n. 1–3, p. 139–159, 1991.
- COSTA, E. D. S.; GOUVEA, M. M. **Autonomous Navigation in Dynamic Environments with Reinforcement Learning and Heuristic**. Machine Learning and Applications (ICMLA), Ninth International Conference, 2010.
- DORIGO, M.; BONABEAU, E.; THERAULAZ, G. **Swarm Intelligence: From Natural to Artificial Systems**. New York, Oxford University Press, Santa Fe Institute Studies in the Science of Complexity, 1999.
- GHANEA-HERCOCK, R.; BARNES, D. **An evolved Fuzzy Reactive Control System For Co-operating Autonomous Robots**. 1 ed. MIT Press, 1996.
- KASABOV, N. K. **Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering**. 1 ed. MIT Press, Cambridge, MA, USA, 1996.
- KUMAR, V. **Scalable swarms of autonomous robots and mobile sensors**. University of Pennsylvania, 2010.
- LIEKNA, A.; GRUNDSPENKIS, J. **Towards practical application of swarm robotics: Overview of swarm tasks**. Engineering for Rural Development, v. 13, p. 271–277, 2014.
- MAES, P. **Artificial Life Meets Entertainment: Life like Autonomous Agents**, Communications of the ACM, vol. 38, no. 11, pp. 108-114, 1995.
- MATARIC, M. J. **The Robotics Primer**. The MIT press, 2007.
- MENDONÇA, M. **Uma contribuição ao desenvolvimento de sistemas inteligentes utilizando redes cognitivas dinâmicas**. 2011. 177 f. Tese (Doutorado

em Engenharia de Automação e Sistemas) - Universidade Tecnológica Federal do Paraná: Curitiba, 2011.

MELÉNDEZ, A. et al. **Fuzzy Logic Reactive Control of an Autonomous Mobile Robot in a Distributed Environment**. 2009 IEEE International Workshop on Hybrid Intelligent Models and Applications, 2009.

PASSINO, K. M.; YURKOVICH, S. **Fuzzy Control**. Addison Wesley Longman, Menlo Park, CA, 1996.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Englewood Cliffs: Prentice Hall, 1995.

SILVA, Esdras S. **Desenvolvimento de Arquiteturas Híbridas Através de Sistemas Computacionais Inteligentes Aplicados à Robótica Autônoma**. 2015. Dissertação de Mestrado – Programa de Pós-Graduação em Engenharia Mecânica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

SPATTI, D. H. **ET30G / ET69B Sistemas Inteligentes Aplicados a Engenharia**. Sistemas Fuzzy. Notas de aula, 19 p., slides. Cornélio Procópio, 2016.

TAN, Y.; ZHENG, Z. **Research Advance in Swarm Robotics**. Defence Technology, v. 9, n. 1, p. 18–39, 2013.

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**. Journal os Man-Machine Studies, Vol. 8:249-291, 1976.