



**Universidade do Estado do Rio de Janeiro**  
Centro de Tecnologia e Ciências  
Faculdade de Engenharia

Karen da Silva Cardoso

**Transporte coletivo por enjaulamento em robótica de  
enxame**

Rio de Janeiro  
2023

Karen da Silva Cardoso

**Transporte coletivo por enjaulamento em robótica de enxame**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Nadia Nedjah

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Luiza de Macedo Mourelle

Rio de Janeiro  
2023

CATALOGAÇÃO NA FONTE  
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

C268 Cardoso, Karen da Silva.  
Transporte coletivo por enjaulamento em robótica de enxame / Karen da Silva Cardoso. – 2023.  
80 f.

Orientadora: Nadia Nedjah.  
Coorientadora: Luiza de Macedo Mourelle.  
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia eletrônica - Teses. 2. Robótica - Teses. 3. Inteligência coletiva - Teses. I. Nedjah, Nadia. II. Mourelle, Luiza de Macedo. III. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. IV. Título.

CDU 007.52

Bibliotecária: Júlia Vieira – CRB7/6022

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

---

Assinatura

---

Data

Karen da Silva Cardoso

## **Transporte coletivo por enjaulamento em robótica de exame**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em: 20/10/2023

Banca Examinadora:

---

Prof.<sup>a</sup> Dr.<sup>a</sup> Nadia Nedjah (Orientadora)  
Faculdade de Engenharia, UERJ

---

Prof.<sup>a</sup> Dr.<sup>a</sup> Luiza de Macedo Mourelle (Orientadora)  
Faculdade de Engenharia, UERJ

---

Prof. Dr. Amit Bhaya  
COPPE, UFRJ

---

Prof. Dr. Igor Machado Coelho  
IC, UFF

Rio de Janeiro  
2023

## **AGRADECIMENTOS**

Agradeço a Deus pela dádiva da vida, pela saúde e pelas oportunidades que me permitiram embarcar nesta jornada.

Expresso minha profunda gratidão às minhas orientadoras, as professoras Nadia Nedjah e Luiza de Macedo Mourelle, por seus preciosos ensinamentos e dedicação incansável ao longo deste processo de pesquisa.

Não posso deixar de reconhecer o apoio inabalável dos meus queridos pais, Alberto da Cruz Cardoso e Claudia Aparecida, que desde o início me incentivaram a estudar e seguir o desafiador caminho da engenharia.

Também desejo estender meus agradecimentos aos amigos que estiveram ao meu lado em todos os momentos desta jornada: Alberto Passos, Alessandra Mariano, Adriana Mariano, Tatiana Reis, Mariana Machado e Tais Salles. Sua amizade e apoio foram fundamentais para enfrentar os desafios.

Em especial, dedico minha gratidão a João Pacheco, que não apenas se tornou a voz da minha consciência, mas também meu maior incentivador em iniciar o curso de mestrado e a persistir ao longo dele.

A todos, a minha mais profunda gratidão.

O tempo não é uma linha reta, mas um círculo que nos traz de volta a velhas questões, mas nos permite vê-las com novos olhos a cada vez.

*Ian Stewart*

## RESUMO

CARDOSO, Karen da Silva. *Transporte coletivo por enjaulamento em robótica de enxame*. 2023. 80 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2023.

O transporte coletivo em robótica de enxame é baseado no transporte de objetos por enxames de robôs caracterizados por suas dimensões significativamente menores em comparação com o objeto a ser transportado. O conjunto de robôs do enxame geralmente possui a mesma arquitetura. Individualmente, os robôs são capazes de realizar funções simples, como detecção, locomoção e comunicação básica. No entanto, cooperativamente, eles podem executar tarefas complexas. O transporte coletivo tem diversas aplicações que abrangem o transporte de objetos em grande escala e objetos em nanoescala. Como resultado, inúmeros estudos foram conduzidos sobre esse tema, destacando três estratégias de transporte: impulsionamento, agarramento e enjaulamento. Neste trabalho, a estratégia de enjaulamento é adotada. Tal estratégia pode ser definida como o completo enclausuramento do objeto, permitindo-lhe um certo grau de liberdade, mas impedindo-o de escapar da formação de robôs ao seu redor. A principal vantagem dessa estratégia é a progressão coordenada e coesa do transporte, uma vez que as forças aplicadas ao objeto pelo enxame se complementam e evitam desvios significativos em sua trajetória em direção ao destino planejado. Este trabalho propõe um método para abordar o transporte coletivo por enjaulamento, que opera em quatro etapas: a busca do objeto pelo enxame, o recrutamento do enxame de robôs, o posicionamento inicial dos robôs ao redor do objeto, enjaulando-o, e a etapa de transporte. A implementação da estratégia proposta é realizada na plataforma CoppeliaSim, na qual o enxame é composto por robôs do tipo Khepera-III. Este robô é escolhido devido à disposição de seus sensores ao longo de seu perímetro, o que permite uma ampla visão do ambiente. A arena é o cenário onde as simulações são realizadas, a mesma é discretizada em diferentes configurações. Tal discretização tem impacto significativo no desempenho das etapas de busca e transporte. A avaliação de desempenho das etapas de busca e recrutamento é baseada no tempo de execução das mesmas, enquanto que para a etapa de transporte, além do tempo de execução, o erro normalizado da trajetória do objeto é avaliado para diferentes configurações de discretização da arena e comprimentos de trajeto. A avaliação da etapa de transporte também envolve a análise do impacto do número de robôs e das dimensões do objeto a ser transportado. O método proposto se mostra eficaz, uma vez que o enjaulamento é mantido ao longo de todo o percurso, garantindo assim o transporte uniforme do objeto.

Palavras-chave: Robótica de enxame; Transporte coletivo; Enjaulamento.

## ABSTRACT

CARDOSO, Karen da Silva. *Collective transport by caging in swarm robotics*. 2023. 80 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2023.

Collective transport in swarm robotics is based on the transportation of objects by swarms of robots characterized by having significantly smaller dimensions compared to the object to be transported. The set of robots of the swarm generally have the same architecture. Individually, the robots are able to perform simple functions, such as sensing, locomotion, and basic communication. However, cooperatively, they can perform complex task. Collective transport has diverse applications that encompass the transportation of both large-scale objects and nanoscale objects. Consequently, numerous studies have been driven by this topic, highlighting three transport strategies: pushing, grasping, and caging. In this work, the caging strategy is adopted. It can be defined as the complete enclosure of the object. Allowing it a certain degree of freedom but preventing it from escaping the formation of robots around it. The main advantage of this strategy is the coordinated and cohesive progression of the transport since the forces applied to the object by the swarm complement each other and prevent significant deviations in its trajectory towards the planned destination. This work proposes a method to approach collective transport by caging, which operates in four stages: the search for the object by the swarm, the recruitment of swarm of robots, the initial positioning of the robots around the object, enclose it, and the transportation stage. The implementation of the proposed strategy is carried out in the CoppeliaSim platform, wherein the swarm is composed of Khepera III-type robots. This robot is chosen due to the arrangement of its sensors along its perimeter, which allows for a wide view of the environment. The arena, which is the scenario where the simulations are conducted is discretized into different configurations. This space discretization has a significant impact on the performance of the search and transport stages. The performance evaluation of the search and recruitment stages are based on the execution time, while for the transport stage, in addition to the execution time, the normalized error of the object's trajectory is evaluated for different arena discretization configurations and path lengths. The evaluation of the transportation stage also involves analyzing the impact of the number of robots and the dimensions of the object to be transported. The proposed method proves to be effective, as the caging is maintained throughout the entire path, thus ensuring the uniform transport of the object.

Keywords: Swarm robotics; Collective Transport; Caging.



## LISTA DE FIGURAS

1	Impulsioneamento . . . . .	18
2	Agarramento . . . . .	19
3	Enjaulamento . . . . .	20
4	Representação gráfica das constantes $r_{\oplus}$ , $\varepsilon$ , $\epsilon$ , $w$ e $\theta_{min}$ . . . . .	36
5	Controlador PID . . . . .	37
6	Ângulo $\alpha_{\lambda}$ , angulo entre a orientação atual do robô $\beta_i$ e a orientação que o mesmo deve atingir na direção do ponto de referência $\lambda$ . . . . .	40
7	Exemplificação das distâncias $D(p_i^t, C_{\oplus})$ , $D(p_i^t, r)$ e $D(p_i^t, \ell)$ , e dos ângulos $\theta_r$ e $\theta_{\ell}$ no enjaulamento . . . . .	41
8	Máquinas de estados referentes à movimentação e orientação dos robôs . . . . .	43
9	Posicionamento dos sensores no Khepera-III . . . . .	51
10	Exemplificação do posicionamento inicial . . . . .	53
11	Etapas de busca e recrutamento . . . . .	63
12	Etapas de posicionamento inicial . . . . .	64
13	Etapas de transporte . . . . .	65
14	Análise do erro entre o percurso ideal $PI$ e o percurso real $PR$ . . . . .	67

## LISTA DE TABELAS

1	Constantes proporcionais, integrais e derivativas utilizadas nos controladores PID . . . . .	61
2	Constantes de tolerância empregadas nas máquinas de estados . . . . .	62
3	Dimensões dos robôs e constantes utilizadas nas equações: . . . . .	62
4	Resultados da etapa de busca . . . . .	63
5	Resultados das simulações para enxames compostos por diferentes números de robôs . . . . .	68
6	Resultados das simulações para enxame composto por oito robôs e objetos com distintas dimensões. . . . .	69
7	Resultados das simulações para arenas com passos de discretização de $0,5m$ e de $1,0m$ e enxames compostos por diferentes números de robôs. . . . .	70
8	Resultados das simulações para arenas com passos de discretização de $0,5m$ e de $1,0m$ , enxame composto por oito robôs e objetos com distintas dimensões. . . . .	71

## LISTA DE ALGORITMOS

1	PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, y$ ) . . . . .	39
2	Movimenta( $D_{p'}, \alpha_{p'}$ ) no robô $i$ . . . . .	44
3	Orienta( $\alpha_{p'}$ ) no robô $i$ . . . . .	45
4	Enjaula( $D_\lambda, D_r, D_\ell, D_\oplus, \alpha_\lambda, \theta_r, \theta_\ell$ ) no robô $i$ . . . . .	45
5	Inicializa enjaulamento no robô $i$ . . . . .	45
6	Controla transporte( $D_\lambda, D_r, D_\ell, D_\oplus, \alpha_\lambda, \theta_r, \theta_\ell$ ) no robô $i$ . . . . .	46
7	Aplica transições na máquina de estados ( $d, \epsilon$ ) no robô $i$ . . . . .	47
8	Aplica ações na máquina de estados( $u_{D_\lambda}, u_{D_r}, u_{D_\ell}, u_{D_\oplus}, u_{\alpha_\lambda}, u_{\theta_r}, u_{\theta_\ell}$ ) . . . . .	48
9	Transporte por enjaulamento . . . . .	49
10	Inicializa arena . . . . .	50
11	Procura no robô $i$ . . . . .	52
12	Recruta( $r^*, C_\oplus$ ) no robô $i$ . . . . .	53
13	Recrutador( $C_\oplus$ ) no robô $i$ . . . . .	54
14	Recrutado( $C_\oplus$ ) no robô $i$ . . . . .	55
15	Transporta no robô $i$ . . . . .	58

## LISTA DE SIMBOLOS

$\alpha_{p'}$	Ângulo entre a orientação atual do robô $i$ , $\beta_i^t$ , e a orientação a qual o robô deve atingir em direção à posição $p'$
$\beta_i^t$	Orientação atual do robô $i$
$\Delta$	Distância máxima de aproximação entre o enxame e o centro de massa do objeto no recrutamento
$\epsilon$	Distância de tolerância entre robôs adjacentes
$\eta(\lambda_i)$	Função de vizinhança entre pontos de referência
$\Lambda$	Número total de pontos de referência
$\lambda$	Ponto de referência
$\mu_i$	Matriz que contém a ordem de chamada para o posicionamento inicial referente ao robô $i$
$\phi$	Ângulo ideal entre robôs adjacentes no enjaulamento
$\rho$	Número total de robôs
$\rho_{min}$	Número mínimo de robôs
$\sigma$	Ângulo de tolerância das máquinas de estado
$\theta_r$	Ângulo entre o robô e seu vizinho à direita
$\theta_\ell$	Ângulo entre o robô e seu vizinho à esquerda
$\theta_{min}$	Ângulo mínimo entre robôs
$\varepsilon$	Distância ideal entre robôs adjacentes
$A$	Aceleração do objeto
$a$	Aceleração do robô
$c$	Contante adicionada ao posicionamento inicial
$C_\oplus$	Posição do centro de massa do objeto
$C_\oplus^{T_{r^*j}}$	Posição do centro de massa do objeto quando o enxame assume que o mesmo atingiu a posição $T_{r^*j}$

$d$	Distância de tolerância das máquinas de estado
$D(p, p')$	Distância entre os pontos $p$ e $p'$
$E$	Erro entre o percurso real $PR$ e o percurso ideal $PI$
$k_d$	Constante de ganho derivativo
$k_i$	Constante de ganho integral
$k_p$	Constante de ganho proporcional
$L$	Conjunto de pontos de referência
$M$	Massa do objeto
$m$	Massa do robô
$n$	Tamanho da trajetória $T_{r^*}$
$p_i^*$	Posição inicial de enjaulamento do robô $i$
$p_i^+$	Posição do robô $i$ quando o centro de massa do objeto atinge o próximo ponto de referência pertencente à $T_{r^*}$
$p_i^0$	Posição inicial do robô $i$ na arena
$p_i^t$	Posição atual do robô $i$
$PI$	Percurso ideal
$PR$	Percurso real percorrido pelo centro de massa do objeto
$R$	Conjunto de robôs
$r^*$	Robô recrutador
$r_{\oplus}$	Raio do objeto
$s$	Número de sensores do robô
$T_{r^*}$	Trajetoória percorrida pelo robô recrutador entre sua posição inicial na arena e a posição de detecção do objeto
$u$	Variável de controle
$w$	Largura do robô

# SUMÁRIO

INTRODUÇÃO .....	14
<b>1 TRANSPORTE COLETIVO POR ENJAULAMENTO.....</b>	<b>17</b>
1.1 Estratégias de Transporte Coletivo .....	17
1.1.1 Impulsionamento .....	18
1.1.2 Agarramento .....	19
1.1.3 Enjaulamento .....	19
1.2 Definição do problema .....	21
1.2.1 Inicialização da Arena .....	21
1.2.2 Recrutamento e Posicionamento Inicial .....	21
1.3 Desafios .....	22
1.4 Considerações Finais do Capítulo .....	24
<b>2 TRABALHOS RELACIONADOS.....</b>	<b>25</b>
2.1 Método de transporte coordenado distribuído .....	25
2.2 Método de transporte por aprendizado de máquina .....	29
2.3 Método de transporte baseado em liderança .....	31
2.4 Método de enjaulamento por garras .....	32
2.5 Considerações finais .....	33
<b>3 MODELOS PARA O ENJAULAMENTO DO OBJETO.....</b>	<b>35</b>
3.1 Modelo matemático para determinação dos números mínimo e máximo de robôs .....	35
3.2 Controlador PID .....	36
3.3 Movimentação dos robôs .....	41
3.4 Considerações Finais do Capítulo .....	47
<b>4 ALGORITMOS DO TRANSPORTE COLETIVO POR ENJAULAMENTO</b>	<b>49</b>
4.1 Inicialização da Arena .....	50
4.2 Busca .....	51
4.3 Recrutamento do enxame .....	53
4.4 Transporte .....	56
4.5 Considerações Finais do Capítulo .....	59
<b>5 AVALIAÇÃO DE DESEMPENHO .....</b>	<b>60</b>
5.1 Aspectos de Implementação .....	60
5.2 Parâmetros de Simulação .....	61
5.3 Busca e Recrutamento .....	62
5.4 Transporte .....	65

# SUMÁRIO

xiii

---

5.4.1	<u>Impacto do Número de Robôs</u> .....	66
5.4.2	<u>Impacto das Características do Objeto</u> .....	68
5.4.3	<u>Impacto da Discretização da Arena</u> .....	69
5.4.3.1	Impacto da Discretização da Arena Variando o Número de Robôs .....	69
5.4.3.2	Impacto da Discretização da Arena Variando as Características do Objeto	70
<b>5.5</b>	<b>Considerações Finais do Capítulo</b> .....	<b>71</b>
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> .....	<b>72</b>
<b>6.1</b>	<b>Conclusões</b> .....	<b>72</b>
<b>6.2</b>	<b>Trabalhos Futuros</b> .....	<b>76</b>
	<b>REFERÊNCIAS</b> .....	<b>78</b>

# INTRODUÇÃO

O TRANSPORTE coletivo, em robótica de enxame, é definido como um conjunto de robôs, capazes de transportar um determinado objeto, com dimensões significativamente maiores que as de um único robô, de um ponto a outro no plano. Enquanto que o enxame de robôs é caracterizado por um conjunto de robôs que, através de comportamentos similares ao de insetos ou animais que possuem comportamento de inteligência coletiva, realizam determinada tarefa (BERMAN et al., 2011; VARDHARAJAN; SOMA; BELTRAME, 2022). O conjunto de robôs que compõem o enxame, em geral, possui a mesma arquitetura, são robôs que, individualmente, realizam funções simples; como sensoriamento, locomoção e comunicação básica. Porém, cooperativamente, são capazes de realizar funções complexas como o transporte. O transporte coletivo, possui diversas aplicações e, conseqüentemente, diversos estudos são baseados no mesmo. Entre eles, destacam-se três estratégias de transporte: impulsionamento, agarramento e enjaulamento (WANG; KUMAR, 2002; CHEN et al., 2015a; SUDSANG; PONCE, 1998; VARDHARAJAN; SOMA; BELTRAME, 2022).

Neste trabalho, elegeu-se o enjaulamento como estratégia de transporte. O mesmo é caracterizado pela completa oclusão do perímetro de um objeto por um enxame de robôs para determinado objetivo. A principal vantagem do enjaulamento no transporte de um objeto é a progressão coordenada e coesa deste transporte. Pois, nesse método, as forças aplicadas ao objeto durante o transporte, em todas as direções, impossibilitam que haja desvio significativo de tal objeto para direções não programadas. Ao estar completamente ocluído, as forças aplicadas por cada robô ao objeto se complementam, permitindo que a movimentação seja retilínea. Ou seja, enquanto os robôs posicionados na parte posterior do objeto, aplicam forças que geram deslocamento, os robôs posicionados nas laterais e à frente do objeto exercem forças que o mantém enjaulado e na direção desejada.

São inúmeras as aplicações do transporte coletivo por enjaulamento. Entre elas, destacam-se: o transporte de objetos frágeis, a montagem de estruturas, a manipulação



de objetos em nanoescala e microescala, a otimização logística em espaços confinados e aplicações médicas que envolvem o transporte controlado de amostras, medicamentos e dispositivos médicos. Para o transporte de objetos frágeis, tal técnica se destaca, pois garante o transporte seguro dos mesmos, devido ao fato de mantê-los protegidos e estáveis durante toda a trajetória, prevenindo possíveis danos. No contexto da construção de estruturas complexas, o transporte coletivo por enjaulamento é empregado com eficácia para posicionar componentes com precisão e segurança, contribuindo significativamente para o processo de montagem. O transporte coletivo por enjaulamento também se destaca na manipulação de objetos em pequenas dimensões. Pois, em escalas extremamente pequenas, onde a manipulação direta é desafiadora, o enjaulamento se revela como uma solução precisa para controlar e movimentar objetos, como componentes microeletrônicos e materiais em nanoescala. Em ambientes com limitações de espaço, como armazéns e fábricas, o transporte coletivo por enjaulamento oferece uma solução eficiente para mover objetos de forma coordenada, otimizando a utilização do espaço disponível. Enquanto que no campo médico, o enjaulamento garante o transporte controlado de amostras, medicamentos e dispositivos médicos de forma precisa e segura.

Neste trabalho, é proposto um método para realizar o transporte coletivo de objetos cilíndricos por meio do enjaulamento. Tal método é baseado no conceito coordenado distribuído, o qual é caracterizado pela massiva comunicação entre os robôs, de forma que os mesmos são capazes de organizarem-se como enxame e realizarem todas as etapas referentes ao transporte coletivo por enjaulamento por meio de informações obtidas de maneira direta, por meio dos sensores individuais de cada robô, ou indireta, por meio da troca de mensagens entre os membros do enxame.

O problema de transporte coletivo abrange processos fundamentais, além do transporte propriamente dito. Estes processos incluem: a busca do objeto na arena pelo enxame de robôs; uma vez que o objeto é detectado, o recrutamento do enxame para a região de detecção; o posicionamento inicial dos robôs ao redor do perímetro do objeto, formando uma estrutura de enjaulamento; e, por fim, o próprio transporte. Portanto, este estudo foi dividido em quatro etapas distintas, sendo a primeira dedicada à busca do objeto pelo enxame na arena; a segunda destinada ao recrutamento do enxame para a região onde o objeto foi detectado; a terceira voltada ao posicionamento inicial dos robôs ao redor do objeto, criando o enjaulamento; e, por último, a quarta etapa dedicada ao transporte

propriamente dito. Esta dissertação está organizada em seis capítulos. O Capítulo 1 introduz o conceito de transporte coletivo e descreve suas principais estratégias, que incluem o impulsionamento, o agarramento e o enjaulamento. Além disso, neste capítulo, o foco é direcionado para a estratégia de transporte coletivo por enjaulamento, que é a abordagem adotada neste trabalho. Também é apresentada uma definição clara do problema relacionado ao transporte coletivo por enjaulamento.

O Capítulo 2 apresenta os trabalhos relacionados que motivaram essa pesquisa. Esses estudos são categorizados em quatro métodos de aplicação distintos: transporte coordenado distribuído, transporte por meio de aprendizado de máquina, transporte com base em liderança e o método de enjaulamento por garras. Além disso, neste capítulo, é estabelecido o método adotado nesta pesquisa, que é o método de transporte coordenado distribuído.

O Capítulo 3 introduz os modelos aplicados ao processo de enjaulamento do objeto. Isso inclui a apresentação do modelo matemático empregado para o enjaulamento do objeto, bem como as estratégias utilizadas para a movimentação dos robôs e para a manutenção do enjaulamento durante o transporte. Essa abordagem estratégica é fundamentada na utilização de máquinas de estados e controladores PID, que são detalhados e discutidos neste capítulo.

O Capítulo 4 apresenta os algoritmos desenvolvidos para essa implementação, apresentando-os de maneira estruturada, seguindo a organização das etapas do processo de implementação.

O Capítulo 5, por sua vez, concentra-se na apresentação dos resultados obtidos para o método de transporte coletivo por enjaulamento. Inicialmente, este capítulo discute os resultados alcançados nas etapas de busca e recrutamento. Em seguida, são apresentados os resultados relativos à etapa de transporte, os quais estão organizados em três estudos de caso. Esses estudos de caso analisam o impacto do número de robôs, das dimensões do objeto e da discretização da arena no que diz respeito ao erro entre a trajetória ideal e a trajetória efetivamente percorrida pelo centro de massa do objeto.

O Capítulo 6 conclui esta dissertação, fornecendo uma análise das principais conclusões decorrentes da implementação do método proposto para o transporte coletivo por enjaulamento. Tal capítulo discute os resultados obtidos e também sugere possíveis direções para trabalhos futuros.

# Capítulo 1

## TRANSPORTE COLETIVO POR ENJAULAMENTO

**E**STE capítulo apresenta o conceito de transporte coletivo e suas principais estratégias. Define a estratégia utilizada neste trabalho, explicitando suas vantagens, desvantagens e desafios. A Seção 1.1 exhibe as três estratégias de transporte coletivo: impulsionamento, agarramento e enjaulamento. A Seção 1.2 apresenta o problema de transporte coletivo. A Seção 1.3 exhibe os principais desafios relacionados ao transporte coletivo por enjaulamento. Na Seção 1.4 são apresentadas as considerações finais do capítulo

### 1.1 Estratégias de Transporte Coletivo

O transporte coletivo, em robótica de enxame, é definido como um conjunto de robôs, capazes de transportar um determinado objeto, com dimensões significativamente maiores que a de um único robô de um ponto a outro no plano. Enquanto que o enxame de robôs é caracterizado como um conjunto de robôs que através de comportamentos similares ao de insetos ou animais que possuem comportamento e inteligência coletiva, realizam determinada tarefa.

Os robôs que compõem o enxame, em geral, possuem a mesma arquitetura, são robôs que, individualmente, realizam funções simples, como sensoriamento, locomoção e comunicação básica. Porém, cooperativamente, são capazes de realizar funções complexas como o transporte.

O transporte coletivo, possui diversas aplicações e, conseqüentemente, diversos estudos são baseados no mesmo. Entre eles, destacam-se três estratégias de transporte: impulsionamento, agarramento e enjaulamento.

### 1.1.1 Impulsioneamento

A estratégia de impulsioneamento consiste no posicionamento de robôs na região posterior do objeto a ser transportado, de forma que a força aplicada por tais robôs gerem o deslocamento do mesmo na direção desejada. Nesta estratégia, a cada passo de simulação, os robôs se reposicionam objetivando garantir que objeto seja transportado na direção desejada, com o menor desvio possível. Pois, a cada impulsioneamento, o objeto pode se deslocar com certo grau de desvio da orientação pretendida. Portanto, após cada impulsioneamento, é necessário verificar se a orientação foi mantida, ou se é necessário que os robôs se reposicionem ao redor do objeto, de forma a garantir que a força resultante gerada pelo enxame no impulsioneamento seguinte, corrija a direção do transporte (FERREIRA et al., 2022).

A manutenção da direção e redução de desvios, são os principais desafios desta estratégia; além dos relacionados ao transporte por enxame, como a comunicação entre os robôs, por exemplo. A Figura 1 exemplifica essa estratégia.

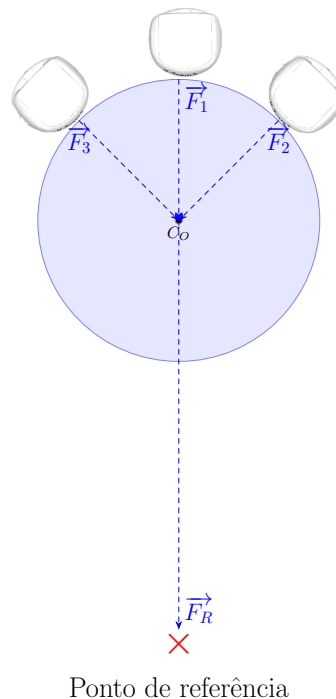


Figura 1: Impulsioneamento

### 1.1.2 Agarramento

A estratégia de agarramento, baseia-se em um enxame de robôs que contenham, em sua arquitetura, mecanismos físicos capazes de se acoplar ao objeto, de forma a segurá-lo e, conseqüentemente, transportá-lo seja empurrando-o, puxando-o, ou ainda levantando-o (SASAKI et al., 1995). Em comparação com a estratégia de impulsionamento definida na Seção 1.1.1; nesta, o impacto de desvio de trajetória, durante o transporte, é significativamente menor, uma vez que os robôs estão fisicamente conectados ao objeto, além de não ser necessário análise e cálculo de reposicionamento a cada passo de simulação.

Além da necessidade de possuir enxame de robôs com características físicas específicas para a implementação desse tipo de estratégia; outro desafio relevante desse método é a análise e o cálculo da disposição ideal do enxame ao longo do objeto, a fim de garantir maior eficiência ao transporte. A Figura 2 apresenta a disposição do enxame ao longo do objeto nessa estratégia.

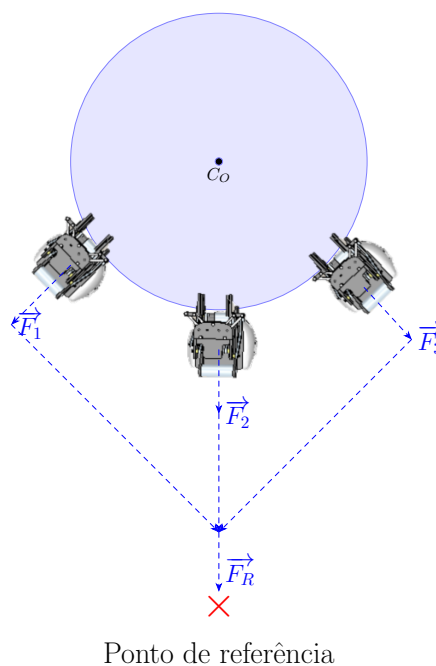


Figura 2: Agarramento

### 1.1.3 Enjaulamento

O enjaulamento pode ser definido como ato ou efeito de enjaular; aprisionamento em jaula. No estudo da robótica de enxame, o enjaulamento é a oclusão do perímetro de um objeto por um enxame de robôs para determinado objetivo, e o fim mais popular do enjau-

lamento por robôs é o transporte (VARDHARAJAN; SOMA; BELTRAME, 2022). A principal vantagem do enjaulamento no transporte de um objeto, em comparação com a estratégia de impulsionamento apresentada na Seção 1.1.1, é a progressão coordenada e coesa deste transporte. Pois, nesse método, as forças aplicadas ao objeto durante o transporte, em todas as direções, impossibilitam que haja desvio significativo de tal objeto para direções não programadas. Uma vez que, quando totalmente ocluído, as forças aplicadas por cada robô ao objeto se complementam, possibilitando um movimento retilíneo. Ou seja, enquanto os robôs posicionados na parte posterior do objeto aplicam forças que geram deslocamento, os robôs posicionados nas laterais e à frente do objeto exercem forças que o mantêm enjaulado e na direção desejada.

Embora, o transporte por enjaulamento seja eficiente e aplicável a diversos propósitos, a maior desvantagem desse método se deve ao desafio de sua implementação; uma vez que diversos parâmetros devem ser avaliados ao longo de cada passo de simulação. Pois, além dos parâmetros e variáveis relacionados ao simples deslocamento do robô e do objeto, é necessária a análise contínua de parâmetros e variáveis relacionados ao enjaulamento; o que torna tal aplicação significativamente complexa. A Figura 3 exemplifica a disposição dos robôs ao longo do objeto nessa estratégia.

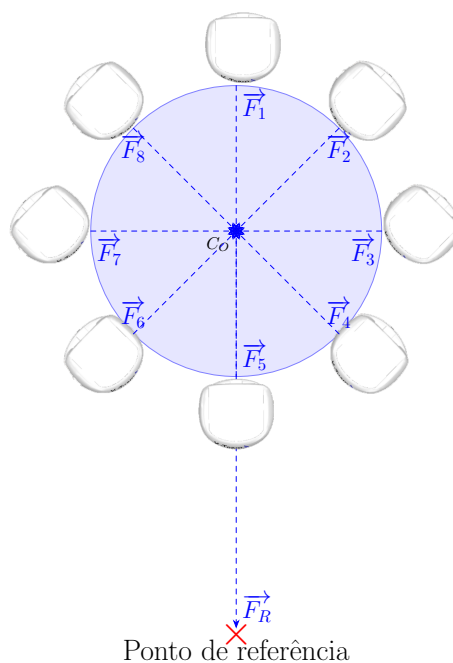


Figura 3: Enjaulamento

## 1.2 Definição do problema

O Transporte coletivo por enjaulamento baseia-se no transporte de um objeto de uma posição no plano para outra, através do trabalho coletivo de múltiplos robôs, a partir da estratégia de enjaulamento, definida na Seção 1.1.3.

O problema de transporte coletivo abrange outros processos que o complementam. Portanto, para o desenvolvimento coeso deste trabalho, o mesmo foi subdividido em 3 etapas: procura do objeto, recrutamento e transporte. Para cada uma dessas etapas, existem diversos parâmetros e variáveis que são formalmente definidos neste capítulo.

### 1.2.1 Inicialização da Arena

A arena é o cenário onde as simulações de todas as etapas do projeto são realizadas. Embora, a etapa de busca seja definida como a primeira etapa a ser implementada, a discretização da arena antecede tal etapa; uma vez que para a busca ser inicializada, deve ser, previamente, estabelecida a forma como os robôs percorrerão a arena, e quais elementos devem compor a mesma, de maneira que os mesmos sejam facilitadores no processo de procura do objeto.

Objetivando tornar a etapa de busca eficiente, foram inseridas marcações na arena, denominadas pontos de referência. Esses pontos de referência foram fixados de forma simétrica ao longo de toda a arena,  $L = \{\lambda_0, \lambda_1, \dots, \lambda_{\Lambda-1}\}$  é o conjunto de  $\Lambda$  pontos de referência. Para cada ponto de referência  $\lambda_i$  uma vizinhança é definida através da função  $\eta_{(\lambda_i)}$ . A principal função dos pontos de referência  $\lambda$ , durante a etapa de procura, é possibilitar o maior espalhamento dos agentes (robôs) ao longo da arena, além de permitir a identificação, pelo robô, de quais regiões já foram por ele examinadas.

### 1.2.2 Recrutamento e Posicionamento Inicial

$R = \{0, 1, \dots, \rho - 1\}$  é o conjunto de  $\rho$  robôs pertencentes ao enxame. Os robôs são caracterizados através da posição do seu centro de massa  $(x_i, y_i)$  e da orientação de seu eixo  $\beta_i$ . A variável  $p_i^t$  contém as informações de posicionamento e orientação do robô  $i$  no tempo  $t$ ,  $p_i^t = (x_i^t, y_i^t, \beta_i^t)$ , e a largura do robô é denominada  $w$ . O primeiro robô a detectar o objeto é denominado recrutador  $r^* \in R$  e a partir da matriz  $\mu_i$ , determina-se a ordem de recrutamento dos robôs, que será realizado por  $r^*$ . Cada robô  $i$  possui a quantidade total de  $s$

sensores, e tais sensores são estruturados da seguinte maneira:  $sensores\langle detect, id \rangle$ , onde  $detect$  contém um valor booleano e  $id$  possui o número identificador do item detectado.

O enjaulamento possui um importante fator que contribui de forma relevante para a complexidade do problema, trata-se do posicionamento inicial dos robôs ao redor do objeto que se deseja ocluir. Pois, diversos fatores influenciam na disposição eficiente dos robôs ao longo do perímetro do objeto, como, por exemplo, o formato do objeto, a quantidade de robôs disponíveis e também as características físicas dos robôs e do objeto. Foi estabelecido que o objeto a ser transportado possui uma forma cilíndrica, já que este constitui o primeiro estudo de transporte coletivo por enjaulamento conduzido por esta equipe. Essa escolha visa minimizar os fatores de complexidade associados a objetos com formas irregulares nesta fase inicial da pesquisa. É importante destacar que não há um modelo dinâmico para o objeto utilizado. A quantidade necessária de robôs  $\rho$  para o enjaulamento do objeto foi definida de acordo com a massa  $M$  e dimensões do objeto escolhido, onde  $r_{\oplus}$  é o raio do objeto e  $C_{\oplus} = (x_{\oplus}, y_{\oplus})$  é o centro de massa do objeto.  $T_{r^*}$  é o vetor de pontos de referência percorridos pelo robô recrutador  $r^*$  entre o ponto de partida do mesmo na arena até o ponto de detecção do objeto. Este vetor é percorrido, de forma inversa, pelo objeto durante o transporte.

A comunicação entre os robôs é realizada através de mensagens. As mensagens trocadas entre os robôs são estruturadas como um vetor de quatro elementos:  $msg\langle type, origin, destination, payload \rangle$ ; onde a primeira posição contém o tipo da mensagem; a segunda indica a origem da mensagem; a terceira aponta o destino para o qual tal mensagem deve ser enviada, e a quarta contém o conteúdo da mensagem, quando houver.

## 1.3 Desafios

Os desafios relacionados ao problema de transporte coletivo por enjaulamento, estão relacionados a cada etapa pertencente ao desenvolvimento do projeto.

O principal desafio da etapa de busca é assegurar a eficiência e a velocidade com a qual o enxame encontra o objeto. São diversas as estratégias de implementação que podem ser aplicadas nessa etapa, como, por exemplo, o uso de algoritmos genéticos que têm por aplicação principal a busca: ACO, PSO, etc. (KENNEDY; EBERHART, 1995). Neste trabalho optou-se por simplificar as etapas que antecedem o transporte do objeto propriamente dito, e por essa razão, a estratégia de busca implementada, baseia-se na



disposição dos pontos de referência  $\lambda$  ao longo da arena e a vizinhança estabelecida entre eles através da função  $\eta(\lambda_i)$ . Uma vez que a escolha de pontos de referência é estabelecida de forma randômica, porém limitada à vizinhança predeterminada, referente a cada ponto de referência  $\lambda_i$ .

Os desafios relacionados ao recrutamento, estão associados à comunicação entre os robôs, que se torna massiva nessa fase de desenvolvimento. Pois, o primeiro robô a identificar a posição do objeto, torna-se o recrutador do enxame e, portanto, tem por função, recrutar os demais robôs pertencentes ao enxame um a um. Esse processo é fundamentado na troca de sinais entre o recrutador e cada um dos robôs do enxame, o que demanda significativo custo computacional. Outro importante aspecto dessa etapa é a movimentação dos robôs no entorno do objeto; o ato de contorná-lo sem gerar possíveis colisões com o objeto ou com os demais robôs já posicionados. Para a implementação dessa função, foi empregado o conceito do veículo de Braitenberg (BRAITENBERG, 1986), (LAMBRINOS; SCHEIER, 1995) ao algoritmo de recrutamento. A etapa de recrutamento, também depende significativamente do posicionamento inicial fixado para cada um dos robôs.

Os desafios relacionados ao transporte tornam a implementação substancialmente complexa. Pois, baseiam-se na análise, cálculo e manutenção, a cada passo de simulação, dos parâmetros associados à movimentação dos robôs e enjaulamento do objeto, e são expressamente responsáveis pela preservação da formação dos robôs em configuração de enjaulamento ao longo do transporte. Tais parâmetros têm por finalidade, permitir que os robôs posicionados na parte posterior do objeto, realizem força suficiente para gerar deslocamento, enquanto que os robôs posicionados nas laterais e na região frontal do objeto, exercem forças tanto relacionadas ao deslocamento quanto à contenção, objetivando a manutenção do enjaulamento e da direção para qual deseja-se transportar o objeto. A comunicação também impacta essa implementação de forma significativa, uma vez que os robôs necessitam comunicar-se ao chegar e ao partir de cada ponto de referência durante a etapa de transporte. A comunicação garante que a etapa de transporte seja realizada de forma síncrona, e conseqüentemente que o enjaulamento do objeto seja mantido ao longo de toda a trajetória, viabilizando assim o transporte do objeto. Portanto, pode-se dizer que a comunicação entre os robôs é fundamental nesta implementação. Após finalizado o enjaulamento, a etapa de transporte é iniciada, orientando todos os robôs em direção ao

primeiro  $\lambda$  referente ao caminho  $T_{r^*}$  a ser percorrido. Portanto, o transporte é iniciado apenas quando os robôs possuem ciência de que todo o enxame está orientado. O mesmo ocorre ao atingir tal ponto de referência; pois, quando todos os robôs chegarem ao ponto de referência desejado, inicia-se o novo processo de orientação e transporte em direção ao próximo ponto de referência pertencente à  $T_{r^*}$ .

## 1.4 Considerações Finais do Capítulo

Neste capítulo, foram exemplificadas as três principais estratégias de transporte coletivo: impulsionamento, agarramento e enjaulamento. Determinou-se o transporte coletivo por enjaulamento como estratégia abordada neste trabalho e, são discutidos os principais desafios associados à implementação de tal estratégia. O capítulo a seguir, apresenta quatro diferentes métodos de aplicação da estratégia de transporte por enjaulamento: método de transporte coordenado distribuído; método de transporte por aprendizado de máquina; método de transporte baseado em liderança; e método de enjaulamento por garras, além de apresentar trabalhos relacionados à estratégia de transporte por enjaulamento, referente a cada um desses métodos.

## Capítulo 2

# TRABALHOS RELACIONADOS

**E**STE capítulo apresenta diferentes métodos de transporte coletivo por enjaulamento em robótica de enxame, a partir da análise de diversos trabalhos relacionados ao tema. Tais métodos foram categorizados em quatro seções: na Seção 2.1 são discutidos os trabalhos que utilizam o método de transporte coordenado distribuído; a Seção 2.2 apresenta trabalhos que aplicam o método de transporte por aprendizado de máquina; a Seção 2.3 aborda trabalhos que desenvolveram-se a partir do método de transporte baseado em liderança, e a Seção 2.4 discorre sobre trabalhos que abordam o método de enjaulamento por garras.

### 2.1 Método de transporte coordenado distribuído

O método de transporte coordenado distribuído caracteriza-se principalmente pela comunicação entre os robôs, que permite que os mesmos identifiquem individualmente a posição do objeto a ser transportado, e organizem-se ao redor do mesmo de forma a enjaulá-lo. O reconhecimento da posição do objeto pode se dar por contato direto (o robô reconhece a posição do objeto a partir de seus sensores) ou a partir de informações transmitidas por outros robôs que tenham realizado tal contato direto previamente e propagaram as coordenadas do objeto. O enjaulamento ocorre de forma semelhante, pois, os robôs organizam-se ao redor do objeto, aprisionando-o, a partir de informações sobre a posição do objeto e a posição dos demais robôs. Outra característica presente neste método de transporte é o consenso. Ou seja, após o enjaulamento, o transporte do objeto inicializa-se apenas após um consenso entre os robôs pertencentes à jaula sobre a direção para a qual o objeto deve ser transportado.

O trabalho proposto por (VARDHARAJAN; SOMA; BELTRAME, 2022) emprega do método de transporte coordenado distribuído a partir da definição de robô semente, um robô é denominado desta forma, uma vez que este seja o primeiro a realizar o contato direto com o objeto. Pois, através dele e de sua posição, os demais robôs são alocados ao redor do objeto com uma distância inter-robôs predefinida. O comportamento coletivo do enxame é realizado a partir de uma máquina de estados de alto nível, de forma que os robôs participam de rodadas sequenciais de alocação de tarefas, permitindo assim que os mesmos se posicionem ao redor do objeto a partir de localizações distribuídas sequencialmente após o contato do robô semente com o objeto, além de permitir que durante o posicionamento, os robôs mantenham a distância inter-robôs desejada. O formato do objeto não interfere no processo de enjaulamento, uma vez que não é necessário conhecimento prévio sobre a forma do objeto que se deseja transportar para que tal procedimento seja realizado de maneira eficaz. Enquanto que a quantidade de robôs que devem aplicar força no objeto, permitindo que o mesmo se desloque, é calculada a partir da diferença angular entre o objeto e posição alvo, posição para a qual deseja-se transportar o objeto. Concluído o processo de enjaulamento, o transporte é inicializado apenas após um consenso entre os robôs pertencentes à gaiola sobre o caminho a ser percorrido até a posição alvo. Alcançado o consenso, os robôs iniciam a aplicação de força sobre o objeto, o mesmo é empurrado e girado de forma a percorrer o trajeto desejado. A rota a ser seguida é representada como uma sequência de posições no espaço (pontos intermediários) as quais todos os robôs devem passar. Após percorrer todos os pontos intermediários, conclui-se o caminho e chega-se à posição alvo, executando assim o transporte eficiente do objeto.

Em (WANG; KUMAR, 2002) aborda-se o método de transporte coordenado distribuído objetivando-se a manipulação de objetos planos. A estratégia deste trabalho é desenvolvida a partir de duas etapas: na primeira etapa, todos os robôs aproximam-se do objeto de forma independente; enquanto que na segunda etapa, os robôs buscam uma configuração de jaula que não permita que o objeto escape. A etapa inicial que consiste na aproximação independente de cada robô do objeto é constituída a partir de um controle de movimentação baseado em campo potencial. Tal controle é aplicado ao sistema de cada um dos robôs, e esses robôs são movimentados a partir de um vetor de força bidirecional. O vetor de força é composto por: a força potencial do objeto e as forças de atração ou repulsão em relação aos demais robôs, que impedem que os mesmos colidam.

A segunda etapa traduz-se como a etapa do enjaulamento, nela os robôs identificam a melhor formação de jaula, ou seja, circundam o objeto de maneira que o mesmo não seja capaz de se libertar. Tal formação é obtida a partir da movimentação dos robôs por todo o perímetro do objeto até que a condição de enjaulamento seja satisfeita.

O trabalho apresentado por (PEREIRA; CAMPOS; KUMAR, 2004) aplica o método de transporte coordenado distribuído de forma semelhante os trabalhos anteriormente citados. Entretanto, nesta abordagem, todos os robôs possuem informações sobre o formato do objeto, embora tais robôs não contenham um modelo dinâmico do objeto em questão. Os robôs também são iniciados com posições aproximadas da localização do objeto, além de conterem a localização da posição alvo (posição para a qual o objeto deve ser transportado). Todos os robôs possuem um campo de visão limitado, porém dentro deste campo de visão eles possuem a capacidade de determinar a posição relativa e a orientação dos demais robôs. Essa característica é fundamental para que os robôs não colidam entre si. O sistema de controle implementado neste trabalho é dito descentralizado e opera por meio de um conjunto de controladores reativos, de forma que os robôs alternam entre os controladores. Os interruptores são governados pela ativação de restrições que dependem do posicionamento relativo de um robô em relação a seus vizinhos e a estimativa dos robôs da orientação do objeto.

Em (FINK; HSIEH; KUMAR, 2008) um modelo matemático de abordagem descentralizada para a manipulação de múltiplos robôs, que utiliza o método de transporte coordenado distribuído é proposto. Diferente dos trabalhos apresentados anteriormente nesta seção, tal abordagem prevê o transporte por enjaulamento em um ambiente composto por obstáculos. Outra característica que o diferencia dos demais trabalhos é a comunicação entre os robôs. Pois, neste projeto, não há comunicação entre os robôs, ou seja, os robôs não são capazes de transmitir ou receber informações uns dos outros. Toda a inteligência coletiva e o trabalho cooperativo é baseado nas informações locais de cada robô e no conhecimento global da tarefa com o qual os robôs são inicializados. O objetivo desse trabalho é alcançar uma coordenação descentralizada que utilize apenas informações locais dos robôs com mínima comunicação e mínimo sensoriamento o que garante que o projeto, como consequência, forneça robustez à implementação, simplifique o comando e o controle, e ainda minimize os custos de implementação. Esta abordagem é constituída de três etapas: a primeira etapa consiste na aproximação dos robôs ao objeto, de forma que

o robô evite colisões com os obstáculos e com os demais robôs da vizinhança; a segunda etapa compõe-se pelo processo de enjaulamento, no qual o robô, após aproximar-se do objeto, estabiliza-se em uma trajetória que orbita o objeto, tal comportamento permite o enjaulamento do objeto, uma vez que se aproximem do objeto robôs suficientes para o enclausuramento do mesmo. A terceira etapa é caracterizada pelo transporte em si do objeto, e tal transporte é realizado pelos robôs pertencentes à jaula, que empurram o objeto até a posição alvo. O trajeto da posição do robô até a posição alvo pode ser dado de suas maneiras: por uma função de navegação global ou por uma trajetória que é definida a partir da posição do objeto.

A abordagem proposta em (CHEN et al., 2015b) utiliza-se do método de transporte coordenado distribuído com o objetivo de transportar, por enjaulamento, objetos grandes e convexos através de um número significativo de robôs, robôs esses que possuam tamanho consideravelmente menor que o objeto que se deseja transportar. Semelhantemente ao trabalho exposto por (VARDHARAJAN; SOMA; BELTRAME, 2022) o trajeto a ser percorrido pelo objeto enjaulado traduz-se em uma sequência de pontos intermediários pelos quais os robôs devem seguir para atingirem a posição alvo final, posição para a qual o objeto deve ser transportado. Ou seja, durante o trajeto, diversas posições alvos intermediárias são atingidas sequencialmente até que a posição final desejada seja alcançada. Os robôs utilizados nessa implementação, são capazes de detectar a posição alvo para a qual devem seguir, além de serem capazes de detectar o objeto através de seus sensores. Tal proposta supõe que o objeto seja grande o suficiente para obstruir a percepção da posição alvo pelo robô, portanto, define-se que o robô é capaz de detectar a posição alvo, independente da sua localização do ambiente e que tal percepção só não é detectada caso a visão do robô seja obstruída pelo objeto. Os robôs realizam o processo de enjaulamento distribuindo-se uniformemente pelo perímetro do objeto. Para o caso em que a visão dos robôs é obstruída pelo objeto, considera-se que todos os robôs devem empurrar o objeto movendo-se em uma direção perpendicular à superfície do objeto em seus pontos de contato, e dessa forma o movimento do objeto será aproximadamente em direção à posição alvo. Admite-se que a superfície oclusa do objeto muda com o tempo, conforme o objeto é movido, portanto, a direção do movimento também sofre mutação ao longo do transporte. Porém, conclui-se que se os robôs persistirem em empurrar o objeto contra a superfície ocluída, a posição alvo será atingida. Semelhantemente à (VARDHARAJAN; SOMA; BELTRAME, 2022), uma

máquina de estado de alto nível é responsável pelo comportamento individual de cada robô. A movimentação dá-se da seguinte forma: Inicialmente, o robô busca pelo objeto a partir de um algoritmo baseado no ambiente, porém, caso o ambiente seja delimitado, o robô apenas inicia um passeio aleatório. Uma vez que o objeto é percebido, o robô se move em direção a ele. Após alcançar o objeto, o robô verifica se a posição alvo é perceptível de sua posição no espaço. Caso a posição alvo não seja perceptível, o robô empurra o objeto simplesmente movendo-se contra ele. Porém, se a posição alvo for visível da posição a qual o robô se encontra, o robô percorre o perímetro do objeto buscando por uma nova posição.

O trabalho apresentado por (STILWELL; BAY, 1993) propõe a implementação de um sistema de transporte de cargas paletizadas através do método de coordenação distribuída. Tal abordagem prevê que pequenos robôs autônomos que possuam mínima comunicação entre si e sem um controle central, aglomerem-se ao redor da carga paletizada e, semelhantemente a forma como as formigas transportam seus alimentos, tais robôs espremam-se sob as arestas do palete levantando-o. Uma vez embaixo do palete elevado, os robôs iniciam o transporte do objeto para a posição alvo e, em seguida, o recolocam no chão. Ou seja, os robôs realizam o enjaulamento, porém, ao invés de empurrar a carga paletizada, levantam-na e realizam o transporte. Esta estratégia define que os robôs tenham tamanho significativamente menor que o palete, e que o comportamento do transporte seja realmente semelhante ao de formigas, uma vez que cada robô, individualmente, seja capaz de levantar uma carga muito menor que a carga paletizada. Entretanto, muitos robôs, unidos, sejam capazes de distribuir o peso do palete entre si de forma que nenhum levante uma carga além de sua capacidade. Tal característica proporciona ao sistema de transporte a capacidade de se reconfigurar dinamicamente para cargas de tamanhos e formatos variados.

## **2.2 Método de transporte por aprendizado de máquina**

Entre os diversos métodos de transporte por enjaulamento, encontra-se o método de transporte por aprendizado de máquina. Tal método caracteriza-se por incorporar técnicas de aprendizado profundo ao controle dos robôs responsáveis pelo enjaulamento e transporte dos objetos.

O trabalho proposto por (DAI et al., 2016) tem por objetivo minimizar a quantidade de robôs necessárias ao enjaulamento, além de diminuir a rotação do objeto durante o transporte. Para reduzir o número de robôs, os pontos adequados, ao redor do objeto, para inserção de um robô na formação da jaula, são calculados e após definidos, são atribuídos aos robôs. Dessa forma, permite-se que os robôs formem, ao redor do objeto, uma jaula simétrica. Para garantir-se que o objeto seja incapaz de escapar da jaula simétrica criada, determina-se que a distância euclidiana entre dois robôs vizinhos seja menor que a largura mínima do objeto poligonal. Após o enjaulamento, os robôs e o objeto são vistos como um único corpo rígido, e para que não haja colisões entre tais robôs, o cálculo que determina a posição de cada um dos robôs na jaula simétrica, considera como parâmetro o raio do robô, visto que esse projeto implementa robôs cilíndricos. Nesta abordagem considera-se um ambiente com múltiplos obstáculos e determina-se que os robôs, após realizarem o processo de enjaulamento, são capazes de transportar o objeto até uma posição alvo, evitando colisões com os obstáculos. O método de controle utilizado para o rastreamento dos robôs ao longo do trajeto percorrido até a posição alvo é o método de controle por modo deslizante difuso.

Em (GROSS; DORIGO, 2009) aplica-se o método de transporte por aprendizado de máquina a partir da implementação de um algoritmo evolutivo com o objetivo de projetar controladores. Tais controladores, por sua vez, são treinados por técnicas de redes neurais artificiais, e são responsáveis pela movimentação dos robôs ao longo do trajeto entre a posição inicial do objeto e a posição alvo para a qual deseja-se transportá-lo. Para a execução desta proposta, foram desenvolvidas duas estratégias de organização dos robôs: a primeira permite que os robôs empurrem diretamente o objeto, sem que haja necessidade de uma automontagem prévia. Enquanto que a segunda pressupõe que os robôs efetuem a automontagem, e após atingir determinada formação, empurrem o objeto de forma direta e indireta. As duas estratégias propostas passaram por testes para objetos distintos, com diversos formatos e pesos. Além destes, também foram realizados testes para diferentes números de robôs responsáveis pelo transporte. Tal trabalho permite a análise do desempenho e do comportamento do sistema, além de possibilitar a quantificação da sensibilidade do mesmo em relação aos seguintes parâmetros: geometria do objeto, alterações na posição alvo, massa do objeto e número de robôs.



## 2.3 Método de transporte baseado em liderança

O método de enjaulamento baseado em liderança, consiste em um método de transporte por enjaulamento que é controlado, de forma global, por um líder. Tal líder pode ser um dos robôs pertencentes ao sistema, ou até mesmo um humano, que detenha o controle geral do sistema e forneça as instruções a serem seguidas pelos robôs à distância. Diferentemente do método de transporte coordenado distribuído, em que a comunicação entre todos os robôs e a percepção do objeto são características imprescindíveis a cada um dos robôs pertencentes ao sistema, no método baseado em líder a comunicação primordial deve ser entre cada robô e o líder, não sendo necessário que os robôs mantenham comunicação entre si, apenas que mantenham comunicação com o líder. Pois, o líder será o responsável por coordenar os robôs tanto no processo de enjaulamento como no processo de transporte do objeto até a posição alvo. Esse tipo de implementação baseada em líder permite uma gama de estratégias e aplicações nas mais diversas áreas. Pois, como o controle do sistema depende, primordialmente, do líder, e esse líder pode comunicar-se com o sistema até mesmo à distância, pode-se aplicar esse método em jogos online, procedimento médicos e cirúrgicos, entre outros.

O trabalho apresentado por (FARRUGIA; FABRI, 2018) propõe que o enjaulamento seja formado a partir da organização dos robôs nos dois lados do objeto, de forma que os robôs sejam capazes de aprisionar o objeto, porém permitindo uma área móvel limitada para o mesmo. As formações geradas pelo enjaulamento não são determinísticas, simplesmente objetiva-se circundar o objeto de forma a aprisioná-lo. Esta abordagem prevê que após a conclusão do enjaulamento, o objeto seja movido através da força perpendicular à superfície do objeto, aplicada pelos robôs pertencentes à jaula, de forma que para mudar a direção do movimento, os robôs precisem se reorganizar, alterando assim a formação da jaula. Esse sistema proposto, aplica o método de transporte baseado em líder, tal líder é responsável por controlar a movimentação dos robôs, bem como o trajeto realizado pelos mesmos ao longo do transporte, uma vez que os robôs se movimentam em direção ao robô líder que é operado a distância por um humano.

Em (BECKER; ERTEL; MCLURKIN, 2014) aplica-se o método de transporte baseado em liderança objetivando testar diversos cenários envolvendo iterações humano-enxame em grande escala, onde o humano é apresentado como líder do enxame. Neste trabalho, apresenta-se e compara-se três arquiteturas distintas de um sistema com  $n$ -robôs: A pri-

meira arquitetura é constituída por dois objetos que devem ser manipulados de forma que os mesmos se encaixem formando um único objeto; a segunda arquitetura possui os mesmos objetos da primeira arquitetura, porém possuem um passo a mais em sua implementação, pois, após realizar a tarefa da primeira arquitetura ( formarem um objeto único com os dois objetos apresentados), tal objeto deve ser transportado até uma posição alvo. A terceira arquitetura é semelhante à segunda, porém, com a inserção de obstáculos. O experimento proposto é realizado online<sup>1</sup> e os usuários manipulam os robôs através de comandos de teclado, objetivando cumprir a tarefa proposta por cada arquitetura. Ou seja, os usuários tornam-se o líder do enxame, e através de comandos de teclado, controlam todo o enxame, uma vez que todos os robôs se movem na mesma direção, a direção imposta pelo líder. Tais arquiteturas também são simuladas utilizando-se de hardware e através dessas duas formas de experimentação, são identificadas as semelhanças e discrepâncias dos resultados obtidos. Através desse trabalho, foi possível observar, que durante os experimentos, tanto online, como no experimento físico, muitos usuários implementaram métodos de enjaulamento para completar a tarefa proposta pelas três arquiteturas. Tal conclusão foi obtida, pois, as entradas do sistema proposto são apenas a velocidade e a taxa de giro, e durante o experimento (manipulação do usuário) registra-se a posição/orientação do objeto usando um sistema de rastreamento, além do tempo necessário para cada tarefa, de forma que é possível determinar o trajeto e estratégia implementada pelo usuário (líder).

## 2.4 Método de enjaulamento por garras

O método de transporte por garras foi o primeiro método de transporte por enjaulamento a ser discutido. Tal método prevê que o objeto que se deseja transportar seja aprisionado, enjaulado, por garras, de forma que tais garras funcionem como uma gaiola e o objeto não seja capaz de escapar. A partir do enclausuramento do objeto, realiza-se o transporte do mesmo.

O trabalho exposto em (RIMON; BLAKE, 1996) é considerado o primeiro trabalho a discutir o transporte por enjaulamento. Tal abordagem prevê o enjaulamento de um

---

<sup>1</sup>Becker, Aaron, Chris Ertel, and James McLurkin. *Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots*. IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014. Disponível em: <<http://www.swarmcontrol.net/games>, <http://youtu.be/HgNENj3hvEg>>. Acesso em 20 de junho de 2023.

objeto através de duas garras, que enclausuram o mesmo, de forma que o objeto aprisionado possua determinado grau de movimento, porém não o suficiente para escapar do enjaulamento. Ou seja, as garras agem como uma mão humana, que segura e aprisiona o objeto. Posteriormente ao enjaulamento, inicia-se o processo de transporte. O enjaulamento é realizado em duas etapas: na primeira etapa, o braço, que contém as garras, move-se com baixa precisão até que as garras alcancem a vizinhança do objeto; em seguida, como segunda etapa, o braço passa a mover-se com alta precisão e de forma lenta enquanto as garras enclausuram o objeto. Este trabalho desenvolve uma teoria capaz de calcular a maior gaiola possível em torno de um objeto, e considera a possibilidade de as garras persistirem em fechar-se em torno do objeto até que o mesmo seja imobilizado.

Em (DAVIDSON; BLAKE, 1998) estende-se a teoria de enjaulamento de 2 garras e um parâmetro apresentada em (RIMON; BLAKE, 1996) para 3 garras e 1 parâmetro. Semelhantemente (RIMON; BLAKE, 1996) o enjaulamento do objeto dá-se através de garras pertencentes a um braço. Tal braço, inicialmente, aproxima-se da vizinhança do objeto, movimentando-se com baixa precisão, e em seguida, começa a movimentar-se lentamente e com alta precisão até que as garras tenham enclausurado o objeto. Entretanto, neste trabalho, utilizam-se três garras e não duas como no trabalho anterior. Durante o processo de enjaulamento, as garras do robô são posicionadas ao redor do objeto de forma que o mesmo não possa escapar. Essa extensão da teoria possui grande importância uma vez que existem objetos que não podem ser enjaulados por apenas duas garras. Este trabalho também apresenta um método para calcular os pontos críticos dos parâmetros de abertura e o espaço de contato entre as garras, além de solucionar a associação das garras imobilizadoras com os furos do objeto a ser transportado.

## 2.5 Considerações finais

Neste capítulo foram apresentados trabalhos relacionados ao problema de transporte coletivo por enjaulamento em robótica de enxame. Baseado no método de transporte proposto, tais trabalhos foram categorizados em quatro seções. A seção 2.1 discute o método de transporte coordenado distribuído. Esse método caracteriza-se por possuir robôs autônomos que são capazes de manter uma comunicação, mesmo que simples, com seus vizinhos, e através de tal comunicação, identificam a posição do objeto e a posição alvo para a qual o objeto deve ser transportado. Nesse método, os robôs organizam-se, individualmente,

ao longo do perímetro do objeto enclausurando-o, e após um consenso sobre o trajeto e a direção que deve ser seguida para o alcance da posição alvo, os robôs, cooperativamente, iniciam o transporte do objeto. A seção 2.2 apresenta o método de transporte por aprendizado de máquina. Tal método consiste em utilizar técnicas de aprendizado profundo para treinar os controladores responsáveis pela movimentação dos robôs no sistema, ou seja, controladores que impõem as informações necessárias para que os robôs atinjam a formação de enjaulamento e em seguida, realizem o transporte do objeto. A seção 2.3 exhibe o método de transporte baseado em liderança, este método é caracterizado por possuir um líder capaz de comunicar-se com todos os robôs, além de transmitir informações de trajetória, localização e posicionamento a cada um dos robôs. O líder é assim denominado pois os demais robôs se movimentam em direção a ele, e este pode ser um dos robôs do sistema, ou até mesmo um humano, que de forma remota, opera o sistema. Em geral, os robôs pertencentes a aplicações que utilizam o método de liderança, não possuem comunicação entre si, a comunicação é bidirecional, porém, restrita entre cada um dos robôs e o líder. A seção 2.4 apresenta trabalhos onde o enjaulamento dos objetos é realizado por robôs que possuem garras imobilizadoras. Tais garras aproximam-se do objeto e em seguida o enclausuram, realizando movimento semelhante ao fechar de uma mão humana. Após o enjaulamento do objeto, realiza-se o transporte do mesmo. O método de transporte por garras é apresentado como primeiro método desenvolvido objetivando o transporte por enjaulamento. Apresentados os principais métodos de transporte por enjaulamento neste capítulo, foi definido o método que melhor se enquadra a proposta deste trabalho: o método de transporte coordenado distribuído. O próximo capítulo detalha o método proposto por esse trabalho, apresenta as vantagens e desvantagens do uso do mesmo e as razões pelas quais tal método foi selecionado para o desenvolvimento desse projeto.

## Capítulo 3

# MODELOS PARA O ENJAULAMENTO DO OBJETO

**E**STE capítulo apresenta a modelagem utilizada para a formação de enjaulamento dos robôs em torno do objeto a ser transportado. A Seção 3.1 apresenta o modelo desenvolvido para determinar o número máximo e o número mínimo de robôs necessários para o enjaulamento e transporte do objeto, além de determinar o ângulo mínimo entre os mesmos. A Seção 3.2 apresenta o controlador PID e explicita a aplicação do mesmo no desenvolvimento deste trabalho. A Seção 3.3 apresenta a modelagem desenvolvida para a movimentação dos robôs, a partir de máquinas de estado.

### 3.1 Modelo matemático para determinação dos números mínimo e máximo de robôs

O enjaulamento dos robôs em torno do objeto, foi calculado de acordo com as dimensões e características do objeto e do enxame. O primeiro passo dessa modelagem tem por objetivo identificar tais características e definir o número mínimo de robôs  $\rho$  necessários para transportar o objeto, além de determinar o ângulo mínimo que deve existir entre os robôs para que os mesmos não colidam. A definição da quantidade mínima de robôs necessária para o transporte do objeto é expressa na Equação 1:

$$\rho_{min} = \frac{M \times A}{m \times a}, \quad (1)$$

onde  $\rho_{min}$  representa o número mínimo de robôs necessários para o transporte do objeto,  $M$  é a massa do objeto e  $A$  representa a aceleração do objeto. As constantes  $m$  e  $a$  representam, respectivamente, a massa e aceleração do robô. O cálculo do ângulo mínimo

entre robôs, permite a determinação da quantidade máxima de robôs que devem pertencer ao enjaulamento. Tal cálculo é expresso na Equação 2:

$$\arctan\left(\frac{\theta_{min}}{2}\right) = \frac{(w/2 + \varepsilon/2)}{\sqrt{(w/2 + r_{\oplus})^2 - (w/2 + \varepsilon/2)^2}}, \quad (2)$$

onde,  $\theta_{min}$  representa a o ângulo mínimo entre robôs na formação de enjaulamento;  $w$  expressa a largura do robô; a constante  $r_{\oplus}$  representa o raio do objeto e  $\varepsilon$  é uma constante predefinida, e traduz o espaço mínimo a ser mantido entre robôs adjacentes.

A fixação das posições dos robôs ao redor do objeto, no enjaulamento, foram determinadas a partir do ângulo  $\phi$  obtido pela divisão dos trezentos e sessenta graus pertencentes ao perímetro circular do cilindro, pela quantidade total de robôs  $\rho$ , como define a Equação 3:

$$\phi = \frac{360^\circ}{\rho}. \quad (3)$$

Tais constantes são explicitadas na Figura 4, onde o ângulo  $\phi$  é apresentado como maior ou igual a  $\theta_{min}$ , ou seja, o ângulo entre robôs  $\phi$  deve ser maior ou igual ao ângulo mínimo  $\theta_{min}$  obtido através da Equação 2. A constante  $\varepsilon$  representa a distância entre robôs e a mesma deve ser maior ou igual à distância mínima entre robôs  $\varepsilon$ , uma vez que  $\varepsilon$  é definido para  $\theta_{min}$ .

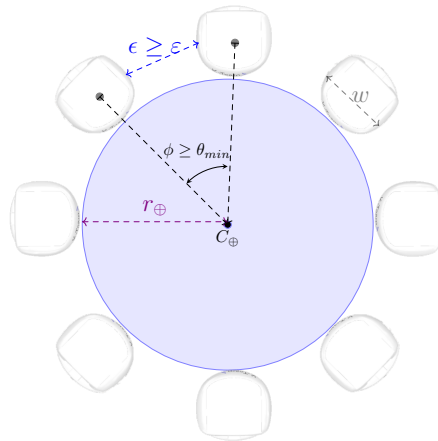


Figura 4: Representação gráfica das constantes  $r_{\oplus}$ ,  $\varepsilon$ ,  $\varepsilon$ ,  $w$  e  $\theta_{min}$ .

## 3.2 Controlador PID

O controle PID (Proporcional, Integral e Derivativo) foi selecionado como método de controle na movimentação dos robôs no ambiente de simulação. Tal escolha deu-se pela

grande credibilidade e eficiência desse método, além do amplo uso do mesmo nas mais diversas aplicações que dependem de controle contínuo.

O controlador PID ou controlador de “três termos” pode ser definido como um mecanismo de malha de controle com realimentação, que tem por finalidade calcular o erro  $e(t)$  obtido através do cálculo da diferença entre o valor desejado (*setpoint*)  $SP$  e a variável de processo  $PV$ . A variável de processo  $PV$  é medida continuamente com o objetivo de verificar se a mesma se aproxima de  $SP$ . Portanto, o controlador PID, realiza tais medições, calcula o erro  $e(t)$  e aplica as correções necessárias para que  $PV$  aproxime-se de  $SP$ . Tais correções são baseadas em termos proporcionais, integrais e derivativos. A função de transferência do controlador PID no domínio de Laplace (SPIEGEL, 1965) é dada pela Equação 4:

$$G(s) = k_p + k_i \frac{1}{s} + k_d s, \quad (4)$$

onde  $k_p$  é o ganho proporcional,  $k_i$  é o ganho integral,  $k_d$  é ao ganho derivativo e  $s$  corresponde à frequência complexa. As constantes  $k_d$ ,  $k_p$  e  $k_i$  impactam o comportamento do controlador individualmente e os valores atribuídos às mesmas são determinados de acordo com a observação de tais impactos na implementação (ANG; CHONG; LI, 2005), (ZHONG, 2006).

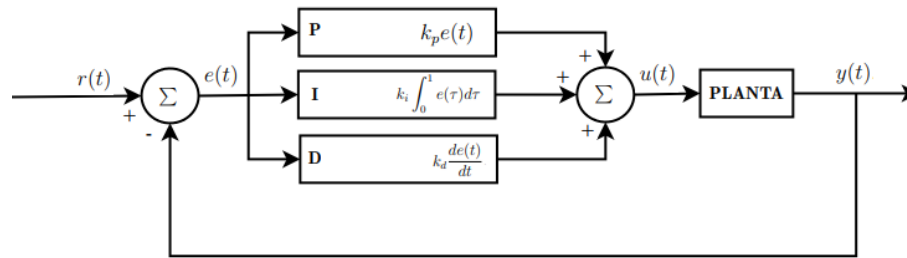


Figura 5: Controlador PID

A Figura 5 apresenta o diagrama de blocos que exemplifica o controlador PID. Nessa figura, o valor desejado (*setpoint*)  $SP$  é representada por  $r(t)$ , enquanto que a variável de processo  $PV$  é representada por  $y(t)$ . Observa-se que a partir das entradas dos valores  $r(t)$  e  $y(t)$ , o erro  $e(t)$  é calculado, como exhibe a Equação 5:

$$e(t) = r(t) - y(t). \quad (5)$$

A partir do erro obtido  $e(t)$ , o controlador, baseado em uma soma ponderada dos termos proporcional, integral e derivativo, aplica uma correção denominada variável de controle  $u(t)$ . A variável de controle  $u(t)$  pode ser expressa pela Equação 6:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}, \quad (6)$$

onde  $k_p$ ,  $k_i$  e  $k_d$  são, respectivamente, os ganhos proporcional, integral e derivativo.  $e(t)$  é o erro, diferença entre o valor desejado  $r(t)$  e a variável de processo  $y(t)$ ;  $t$  é o tempo instantâneo e  $\tau$  representa a variável de integração.

Após calculada a correção  $u(t)$ , a mesma é aplicada ao processo, e torna-se a medir a variável  $y(t)$ . A realimentação garante que tal execução seja contínua, permitindo assim que  $y(t)$  aproxime-se significativamente de  $r(t)$ ; o que torna o controlador PID significativamente preciso.

A variável de controle  $u(t)$ , possui três termos. O termo proporcional do controlador PID gera um valor proporcional ao erro atual  $e(t)$ . A resposta proporcional é dada pelo erro  $e(t)$  multiplicado pelo ganho proporcional  $k_p$ . O ganho proporcional tem impacto significativo na saída do sistema. Pois, quando o ganho proporcional possui valor elevado, gera mudança significativa na saída do sistema para uma pequena variação do erro. Portanto, altos valores de ganho proporcional podem tornar o sistema instável. Por outro lado, valores muito baixos de ganho proporcional tornam o sistema pouco sensível à pequenas perturbações.

O termo integral é dado pela multiplicação do ganho integral  $k_i$  pelo somatório do erro acumulado. Portanto, pode-se dizer que a resposta do termo integral é proporcional à magnitude e duração do erro. Tal termo permite que o processo se aproxime mais rapidamente do valor desejado (*setpoint*)  $r(t)$ ; além de eliminar o erro residual de estado estacionário, comum em controladores puramente proporcionais. O erro de estado estacionário pode ser definido como a diferença entre a saída final desejada e a saída real do sistema (LIPTAK, 2003). O ajuste do ganho integral tem impacto significativo. Pois, uma vez que o mesmo responde a erros acumulados, a resposta pode ultrapassar o *setpoint*  $r(t)$ . O termo derivativo é dado pela multiplicação do ganho derivativo  $k_d$  pela derivada do erro. A derivada do erro, representa a inclinação do mesmo ao longo do tempo. Tal termo permite a melhora do tempo de estabilização e a estabilidade do sistema.



O Algoritmo 1 descreve o controlador PID utilizado nessa implementação. A descrição desse algoritmo é a mesma apresenta nesta seção sobre do diagrama de blocos do controlador PID, exibido na Figura 5. Tal descrição é apresentada até a oitava linha desse algoritmo. Porém, a partir da nona linha implementou-se uma solução para possíveis problemas de *Wind-up* da ação integral (LEAO, 2020b).

---

**Algoritmo 1** PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, y$ )

---

**garante**  $u$ ;

$e_{antigo} := e$ ;

$e := (r - y)$ ;

$acum := acum + \frac{(e+e_{antigo})}{2 \times T_S}$ ;

$u := k_p \times e + k_i \times acum + k_d \frac{e - e_{antigo}}{T_S}$ ;

**se**  $u > u_{sat}$  **então**

$acum := acum - (u - u_{sat})k_i$

$u := u_{sat}$

**fim se**

**se**  $u < u_{sat} \wedge e < e_b$  **então**

$acum + \frac{(e+e_{antigo})}{2 \times T_S}$

$u = k_p \times e + k_i \times acum + k_d \frac{(e - e_{antigo})}{T_S}$

**fim se**

---

Devido às características do controlador PID apresentadas nessa seção, o mesmo foi escolhido para controlar a movimentação dos robôs e a formação de enjaulamento; uma vez que tal controlador é capaz de analisar e corrigir o erro a cada passo de simulação. Portanto, durante a movimentação utilizou-se dois controladores PID, um com variável de processo igual a  $D(p, p') = D(p_i^t, \lambda)$ , e outro com  $y(t)$  igual a  $\alpha_{p'} = \alpha_\lambda$ . De forma que tais controladores permitem a análise e correção do erro da movimentação dos robôs a partir do controle da distância entre o robô e o alvo, e da orientação do robô. A distância  $D(p_i^t, \lambda)$  é a distância entre a posição atual do centro de massa do robô e o ponto de referência escolhido  $\lambda$ . Neste trabalho, a distância entre dois pontos é dado pelo cálculo da distância euclidiana, apresentada na Equação 7:

$$D(p, p') = \sqrt{(x - x')^2 + (y - y')^2}. \quad (7)$$

O ângulo  $\alpha_{p'} = \alpha_\lambda$  é formado pela orientação do robô e pela orientação a qual deseja-se atingir em direção ao ponto de referência  $\lambda$ . Esse ângulo é calculado a partir da Equação 8:

$$\alpha_{p'} = |(\arctan(D(p, p')) - \beta_i)|. \quad (8)$$

o ângulo  $\alpha_\lambda$  é calculado como a diferença normalizada entre a orientação atual do robô  $\beta_i$  e o arco tangente da distância entre o centro de massa do robô e o ponto de referência. A normalização ocorre de forma que  $\alpha_\lambda$  assumam valores no intervalo de  $[-\pi, \pi]$ . Tal ângulo é representado na Figura 6.

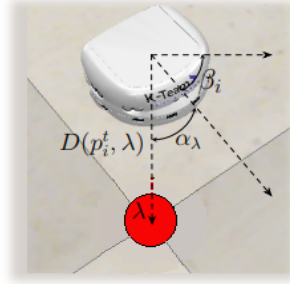


Figura 6: Ângulo  $\alpha_\lambda$ , ângulo entre a orientação atual do robô  $\beta_i$  e a orientação que o mesmo deve atingir na direção do ponto de referência  $\lambda$

Para a formação de enjaulamento, foram necessários, além destes, outros cinco controladores PID, responsáveis pelas seguintes variáveis de processo:  $D(p, p') = D(p_i^t, r)$ ,  $D(p, p') = D(p_i^t, \ell)$ ,  $D(p, p') = D(p_i^t, C_\oplus)$ ,  $\theta_r$  e  $\theta_\ell$ . As distâncias:  $D(p_i^t, C_\oplus)$ ,  $D(p_i^t, r)$  e  $D(p_i^t, \ell)$  também são expressas pela equação 7.  $D(p_i^t, C_\oplus)$  representa a distância entre  $p_i^t$  e a posição atual do centro de massa do objeto  $C_\oplus$ . As distâncias  $D(p_i^t, r)$  e  $D(p_i^t, \ell)$  representam a distância entre  $p_i^t$  e a posição atual do centro de massa de seus vizinhos da direita e esquerda, respectivamente. As variáveis  $\theta_r$  e  $\theta_\ell$  representam os ângulos entre a orientação do robô  $i$  e as orientações atuais de seus vizinhos da direita e da esquerda, respectivamente. Vale ressaltar que o valor ideal para  $\theta_r$  e  $\theta_\ell$  é  $\phi$ . A Figura 7 apresenta a disposição desses ângulos e distâncias no enjaulamento.

Portanto, o uso do controle PID tanto para a movimentação dos robôs como para a manutenção do enjaulamento, permite, que a cada passo da simulação, sejam avaliadas todas as variáveis relacionadas tanto à movimentação como ao enjaulamento; e independentemente da posição ao redor do objeto que o robô ocupar nesse passo, a velocidade atribuída aos seus motores e, conseqüentemente, a força empregada ao objeto, seja condizente com a função a qual a mesma deve desempenhar.

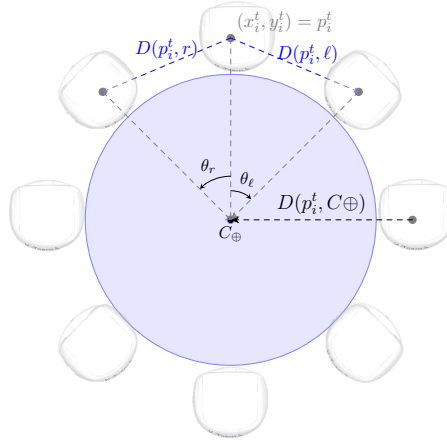


Figura 7: Exemplificação das distâncias  $D(p_i^t, C_{\oplus})$ ,  $D(p_i^t, r)$  e  $D(p_i^t, l)$ , e dos ângulos  $\theta_r$  e  $\theta_l$  no enjaulamento

### 3.3 Movimentação dos robôs

A movimentação dos robôs foi desenvolvida através de três máquinas de estados, e de controladores PID. A primeira máquina de estados é referente à movimentação dos robôs nas etapas de busca e recrutamento, a segunda refere-se à orientação dos robôs e a última, é referente à movimentação dos robôs durante a etapa de transporte. É importante destacar que a análise das máquinas de estados finitas desenvolvidas neste trabalho não revelou a presença de impasses ou ciclos infinitos.

A máquina de estados referente à movimentação dos robôs durante a busca e o recrutamento, é apresentada na Figura 8(a). A mesma é estruturada de maneira que a cada passo de simulação analisam-se as variáveis relacionadas à movimentação dos robôs ( $D(p, p') = D(p, \lambda)$  e  $\alpha_{p'} = \alpha_{\lambda}$ ) a partir de dois controladores PID, verificando assim, a distância e a orientação do robô em relação ao seu ponto de referência alvo. o mesmo ocorre nas máquinas de estados referente à orientação e ao transporte. Porém, na máquina de estados referente à orientação, apenas há apenas um controlador PID, que controla a variável  $\alpha_{p'} = \alpha_{\lambda}$ , enquanto que na máquina de estados referente ao transporte são necessários sete controladores PID, os quais analisam as variáveis:  $D(p, p') = D(p_i^t, \lambda)$ ,  $D(p, p') = D(p_i^t, r)$ ,  $D(p, p') = D(p_i^t, l)$ ,  $D(p, p') = D(p_i^t, C_{\oplus})$ ,  $\alpha_{p'} = \alpha_{\lambda}$ ,  $\theta_r$  e  $\theta_l$ . Ou seja, tais variáveis são entradas de controladores PID, os quais, verificam o erro das mesmas ao longo do percurso e determinam a correção. A máquina de estados, ao receber tais informações dos controladores, determina quais devem ser corrigidas, adicionando ou reduzindo velocidade nos motores de cada robô (LEAO, 2020a).

A máquina de estados implementada nas etapas de busca e recrutamento é explicitada na Figura 8(a), enquanto que a máquina de estados referente a orientação dos robôs é apresentada na Figura 8(b).

A Figura 8(a), apresenta os três estados nos quais consistem a movimentação do robô e podem ser interpretados da seguinte maneira: caso o robô  $i$  encontre-se no estado *parado*, analisa-se se a distância entre o robô  $i$  e o ponto de referência para o qual tal robô destina-se  $\lambda$ . Essa distância é denominada  $D(p, p') = D(p_i^t, \lambda)$ . Tal distância é comparada com o valor de distância de tolerância  $d$ , tal distância de tolerância é uma constante de valor predefinido.

Portanto, caso o robô  $i$  esteja no estado *parado* e  $D(p_i^t, \lambda)$  seja maior que  $d$ , significa que o mesmo ainda está distante do ponto de referência, portanto, o robô entra no estado *Alinhando*. Porém, caso o robô  $i$  esteja parado e a distância atual entre o mesmo e o ponto de referência seja menor ou igual à distância de tolerância; significa que o robô chegou ao seu destino, portanto, deve permanecer no estado *parado*.

Quando o robô  $i$  está no estado *alinhando*, caso a diferença entre a orientação do robô e o ângulo entre a orientação do robô e a orientação a qual o mesmo deve atingir para orientar-se em direção à  $\lambda$ , seja menor ou igual ao ângulo de tolerância predefinido,  $\alpha_{p'} \leq \sigma$ , significa que tal robô está na direção correta e deve avançar em sua movimentação. Portanto, o robô  $i$  entra no estado *seguindo*, no qual o robô mantém seu percurso de forma retilínea. Caso contrário,  $\alpha_{p'} > \sigma$ , o robô permanece no estado *alinhando* até que sua orientação aponte para a direção do ponto de referência.

Quando o robô  $i$  encontra-se no estado *seguindo*, e  $D(p_i^t, \lambda) \leq d$ , significa que o mesmo chegou ao ponto de referência desejado e, portanto, deve entrar no estado *parado*. Porém, caso o robô esteja no estado *seguindo* e a orientação do mesmo seja diferente da tolerada para que tal robô siga na direção de  $\lambda$ , então o robô  $i$  entra no estado *alinhando*, e permanece no mesmo até que a sua orientação coincida com a qual o mesmo deve se movimentar. Dessa maneira é descrita a movimentação dos robôs ao longo da arena.

A Figura 8(b) apresenta a máquina de estados referente à orientação dos robôs, onde enquanto  $\alpha_{p'} \leq \sigma$ , o robô  $i$  permanece no estado *parado*. Porém, caso contrário, o mesmo permanece no estado *alinhando*.

O Algoritmo 2 descreve a máquina de estados referente à movimentação dos robôs nas etapas de busca e recrutamento, apresentada na Figura 8(a). Nesse algoritmo, os

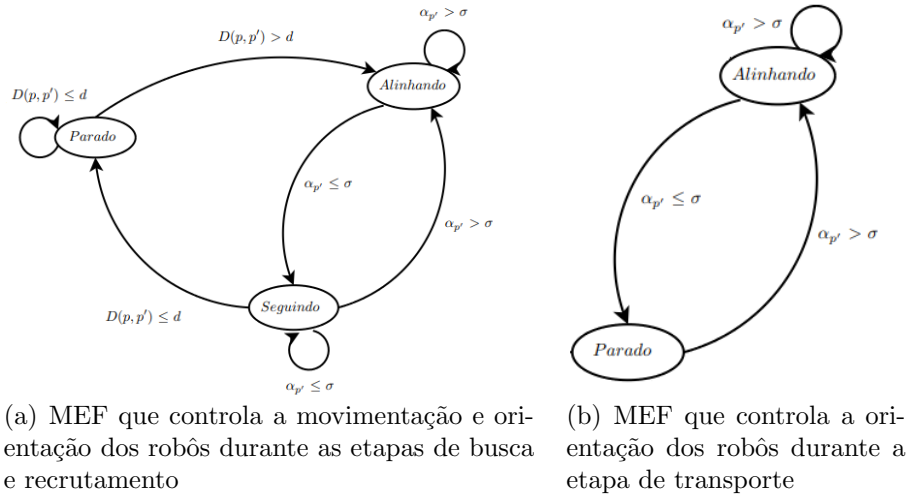


Figura 8: Máquinas de estados referentes à movimentação e orientação dos robôs

estados *alinhando* e *seguindo* tornam-se um único estado, denominado *movendo*. Nessa configuração, o único parâmetro analisado é a distância atual entre o robô e o ponto de referência. Quando o robô  $i$  encontra-se no estado *parado*, analisa-se se  $D(p_i^t, \lambda) > d$ , caso positivo, o robô  $i$  entra no estado *movendo* e permanece no mesmo até que  $D(p_i^t, \lambda) \leq d$ , ou seja, até que o robô  $i$  chegue ao ponto de referência desejado (LEAO, 2020c).

O Algoritmo 2 é descrito da seguinte maneira: inicialmente são lidas a posição  $p_i^t$  e a orientação  $\beta_i$  do robô  $i$ , em seguida, calcula-se a distância  $D(p, p')$  e o ângulo  $\alpha_{p'}$ , onde  $p'$  representa o ponto alvo que deseja-se atingir. Normaliza-se o ângulo  $\alpha_{p'}$  para garantir que o mesmo pertença ao intervalo entre  $-\pi$  e  $\pi$ , e conseqüentemente, garantir que ao orientar-se em relação a  $\alpha_{p'}$ , o robô será rotacionado no menor ângulo possível. A partir desse passo, utilizam-se controladores PID para análise das variáveis  $D(p, p')$  e  $\alpha_{p'}$ , após executados, os controladores PID retornam as variáveis de controle  $u_{D_{p'}}$  e  $u_{\alpha_{p'}}$  que são as correções calculadas pelo mesmo e que devem ser adicionadas aos motores como ações dessa máquina de estados. Em seguida, é executado o algoritmo Braitenberg, que tem por finalidade evitar a colisão entre os robôs. Nessa implementação, foi utilizado o algoritmo Braitenberg apresentado pela plataforma CoppeliaSim no código de exemplificação do modelo e-puck (BRAITENBERG, 1986). Tal algoritmo tem como saída as variáveis  $velr_{Brait}$  e  $vell_{Brait}$ , que representam as correções que devem ser aplicadas, respectivamente, aos motores direito e esquerdo do robô, afim de que as colisões sejam evitadas.

Em seguida, tem-se as transições da máquina estados, de forma que é verificado se o robô chegou ao seu ponto de referência alvo ou não, e tal análise determinará se o mesmo

---

**Algoritmo 2** Movimenta( $D_{p'}$ ,  $\alpha_{p'}$ ) no robô  $i$ 


---

**requer**  $p_i^t, \beta_i$ ;

**garante**  $Motor_r, Motor_\ell$ ;

**Calcula**  $D_{p'} = D(p, p')$  a partir da Equação 7;

**Calcula**  $\alpha_{p'}$  a partir da Equação 8;

**Normaliza**  $\alpha_{p'}$  para que  $\alpha_{p'} \in [-\pi, \pi]$ ;

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, D_{p'}$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, \alpha_{p'}$ );

Braitenberg(*sensores*);

**se**  $D_{p'} \leq d$  **então**

estado := *parado*;

**senão**
**se** estado = *parado*  $\wedge D_{p'} > d$  **então**

estado := *movendo*;

**senão**
**se** estado = *movendo*  $\wedge D_{p'} > d$  **então**

estado := *movendo*
**fim se**
**fim se**
**fim se**
**se** estado = *parado* **então**
 $Motor_r := 0.0$ ;

 $Motor_\ell := 0.0$ ;

**senão**
**se** estado = *movendo* **então**
 $Motor_r := (u_{D_{p'}} + u_{\alpha_{p'}}) + vel_r_{Brait}$ ;

 $Motor_\ell := (u_{D_{p'}} - u_{\alpha_{p'}}) + vel_\ell_{Brait}$ ;

**fim se**
**fim se**


---

encontra-se no estado *parado* ou no estado *movendo*. Posteriormente, implementam-se as ações da máquina de estado, onde, caso o estado seja *parado*, os motores do robô recebem velocidade igual a zero. Porém, caso o estado seja *movendo*, são adicionados aos motores tanto as correções obtidas pelos controladores PID, quanto as obtidas pelo algoritmo Braitenberg.

O Algoritmo 3 apresenta o controle da orientação dos robôs em relação à posição alvo desejada, descrito na Figura 8(b). Tal algoritmo é utilizado na etapa de transporte e é bastante similar ao Algoritmo 2, porém, o mesmo utiliza apenas um controlador PID, o qual controla a variável  $\alpha_{p'}$ . Portanto, a correção adicionada aos motores, por esse algoritmo, controla apenas a orientação dos robôs, diferentemente do Algoritmo 2 que

---

**Algoritmo 3** Orienta( $\alpha_{p'}$ ) no robô  $i$

---

**requer**  $p_i^t, \beta_i$ ;

**garante**  $Motor_r, Motor_\ell$ ;

**Calcula**  $\alpha_{p'}$  a partir da Equação 8;

**Normaliza**  $\alpha_{p'}$  para que  $\alpha_{p'} \in [-\pi, \pi]$ ;

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, \alpha_{p'}$ );

**se**  $\alpha_{p'} \leq \sigma$  **então**

estado := *parado*;

**senão**

**se** estado = *parado*  $\wedge$   $\alpha_{p'} > \sigma$  **então**

estado := *movendo*;

**senão**

**se** estado = *movendo*  $\wedge$   $\alpha_{p'} > \sigma$  **então**

estado := *movendo*

**fim se**

**fim se**

**fim se**

**se** estado = *parado* **então**

$Motor_r := 0.0$ ;

$Motor_\ell := 0.0$ ;

**senão**

**se** estado = *movendo* **então**

$Motor_r := (+u_{\alpha_{p'}})$ ;

$Motor_\ell := (-u_{\alpha_{p'}})$ ;

**fim se**

**fim se**

---

**Algoritmo 4** Enjaula( $D_\lambda, D_r, D_\ell, D_\oplus, \alpha_\lambda, \theta_r, \theta_\ell$ ) no robô  $i$

---

Inicializa enjaulamento;

Aplica transições na máquina de estados( $d, \epsilon$ );

Aplica ações na máquina de estados( $u_{D_\lambda}, u_{D_r}, u_{D_\ell}, u_{D_\oplus}, u_{\alpha_\lambda}, u_{\theta_r}, u_{\theta_\ell}$ );

---

controla tanto a orientação, como a distância do robô em relação ao ponto de referência alvo.

**Algoritmo 5** Inicializa enjaulamento no robô  $i$

---

**garante**  $u_{D_\lambda}, u_{D_r}, u_{D_\ell}, u_{D_\oplus}, u_{\alpha_\lambda}, u_{\theta_r}, u_{\theta_\ell}$ ;

**Calcula**  $D_\lambda = D(p_i^t, \lambda)$ ;  $D_r = D(p_i^t, r)$  a partir da Equação 7;

**Calcula**  $D_\ell = D(p_i^t, \ell)$ ;  $D_\oplus = D(p_i^t, C_\oplus)$  a partir da Equação 7;

**Calcula**  $\alpha_\lambda$  a partir da Equação 8;

**Calcula**  $\theta_r$  e  $\theta_\ell$ ;

**Normaliza**  $\alpha_\lambda, \theta_r$  e  $\theta_\ell$  para que  $\alpha_\lambda, \theta_r$  e  $\theta_\ell \in [-\pi, \pi]$ ;

Controla transporte( $D_\lambda, D_r, D_\ell, D_\oplus, \alpha_\lambda, \theta_r, \theta_\ell$ );

---

---

**Algoritmo 6** Controla transporte( $D_\lambda, D_r, D_\ell, D_\oplus, \alpha_\lambda, \theta_r, \theta_\ell$ ) no robô  $i$

---

**garante**  $u_{D_\lambda}, u_{D_r}, u_{D_\ell}, u_{D_\oplus}, u_{\alpha_\lambda}, u_{\theta_r}, u_{\theta_\ell}$ ;

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, D_\lambda$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, D_r$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, D_\ell$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, D_\oplus$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, \alpha_\lambda$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, \theta_r$ );

PID( $k_p, k_i, k_d, T_S, u_{sat}, e_b, r, \theta_\ell$ );

---

A máquina de estados referente à etapa de transporte é descrita através dos Algoritmos 4, 5, 6, 7 e 8. O Algoritmo 4 apresenta o algoritmo principal dessa máquina de estados, exibindo a correta ordem de chamada entre os demais. O Algoritmo 5 apresenta os passos de inicialização, o Algoritmo 6 exhibe os controladores PID implementados na movimentação durante o transporte; tal algoritmo é executado a partir do Algoritmo 5. Enquanto que os Algoritmos 7 e 8 apresentam, respectivamente, as transições e as ações da máquina de estados. O Algoritmo 5 tem por finalidade a execução dos controladores PID e obtenção das variáveis de controle  $u_{D_\lambda}, u_{D_r}, u_{D_\ell}, u_{D_\oplus}, u_{\alpha_\lambda}, u_{\theta_r}$  e  $u_{\theta_\ell}$ . Tal execução se dá a partir da chamada do Algoritmo 6, e as variáveis de controle obtidas são aplicadas como correções nas ações da máquina de estados.

O Algoritmo 7 apresenta as transições entre os nove estados pertencentes à tal máquina de estados finita. Tais estados são: *parado*, *movendo*,  $m_r$ ,  $m_\ell$ ,  $m_\oplus$ ,  $m_{r\ell}$ ,  $m_{r\oplus}$ ,  $m_{\ell\oplus}$  e  $m_{r\ell\oplus}$ . Os estados  $m_r$ ,  $m_\ell$ ,  $m_\oplus$ ,  $m_{r\ell}$ ,  $m_{r\oplus}$ ,  $m_{\ell\oplus}$  e  $m_{r\ell\oplus}$  representam estados em que a distância  $D(p_i^t, \lambda) > d$ , ou seja, estados em que o robô ainda não chegou no ponto de referência desejado, portanto, todos podem ser descritos como um estado de movimentação com aplicação de alguma correção. No estado  $m_r$ , a distância entre o centro de massa do robô e o centro de massa do vizinho do lado direito é maior que o valor de tolerância determinado para essa dimensão, ou seja,  $D(p_i^t, r) > \epsilon$ . No estado  $m_\ell$ ,  $D(p_i^t, \ell) > \epsilon$ ; no estado  $m_\oplus$ ,  $D(p_i^t, C_\oplus) > d$ . Enquanto que os estados  $m_{r\ell}$ ,  $m_{r\oplus}$ ,  $m_{\ell\oplus}$  e  $m_{r\ell\oplus}$  são combinações desses. No Algoritmo 8 são explicitadas as ações realizadas nos motores do robô a partir da determinação do estado em que o mesmo se encontra. Ou seja, aplicam-se as correções obtidas dos controladores PID.



---

**Algoritmo 7** Aplica transições na máquina de estados  $(d, \epsilon)$  no robô  $i$

---

**requer**  $D_\lambda, D_r, D_\ell, D_\oplus$ ;

**garante** estado;

**se**  $D_\lambda \leq d \wedge$  **então**

estado := *parado*;

**senão**

**se**  $(D_\lambda > d)$  **então**

**se**  $(D_r \leq \epsilon) \wedge (D_\ell \leq \epsilon) \wedge (D_\oplus \leq d)$  **então**

estado := *movendo*;

**senão**

**se**  $(D_r > \epsilon) \wedge (D_\ell \leq \epsilon) \wedge (D_\oplus \leq d)$  **então**

estado :=  $m_r$ ;

**senão**

**se**  $(D_r \leq \epsilon) \wedge (D_\ell > \epsilon) \wedge (D_\oplus \leq d)$  **então**

estado :=  $m_\ell$ ;

**senão**

**se**  $(D_r \leq \epsilon) \wedge (D_\ell \leq \epsilon) \wedge (D_\oplus > d)$  **então**

estado :=  $m_\oplus$ ;

**senão**

**se**  $(D_r > \epsilon) \wedge (D_\ell > \epsilon) \wedge (D_\oplus \leq d)$  **então**

estado :=  $m_{r\ell}$ ;

**senão**

**se**  $(D_r > \epsilon) \wedge (D_\ell \leq \epsilon) \wedge (D_\oplus > d)$  **então**

estado :=  $m_{r\oplus}$ ;

**senão**

**se**  $(D_r \leq \epsilon) \wedge (D_\ell > \epsilon) \wedge (D_\oplus > d)$  **então**

estado :=  $m_{\ell\oplus}$ ;

**senão**

**se**  $(D_r > \epsilon) \wedge (D_\ell > \epsilon) \wedge (D_\oplus > d)$  **então**

estado :=  $m_{r\ell\oplus}$ ;

---

## 3.4 Considerações Finais do Capítulo

Neste capítulo, determinou-se a modelagem matemática para definição das quantidades mínima e máxima de robôs necessárias para o enjaulamento e transporte do objeto. Definiu-se também o método desenvolvido para realizar a movimentação dos robôs ao longo da arena, baseado em máquina de estados e controladores PID, além de exemplificar e destacar o uso e importância desse tipo de controladores. Pois, os mesmos foram fundamentais, neste trabalho, tanto para a movimentação dos robôs ao longo da arena, como para a manutenção da formação de enjaulamento durante a etapa de transporte.

---

**Algoritmo 8** Aplica ações na máquina de estados  $(u_{D_\lambda}, u_{D_r}, u_{D_\ell}, u_{D_\oplus}, u_{\alpha_\lambda}, u_{\theta_r}, u_{\theta_\ell})$

---

**requer** estado;

**garante**  $MotorD, MotorE$ ;

**escolha** (estado)

**caso** parado:

$$MotorD := 0.0;$$

$$MotorE := 0.0;$$

**caso** movendo:

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda});$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda});$$

**caso**  $m_r$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r});$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r});$$

**caso**  $m_\ell$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + (u_{D_\ell} + u_{\theta_\ell});$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + (u_{D_\ell} + u_{\theta_\ell});$$

**caso**  $m_\oplus$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + u_{C_\oplus};$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + u_{D_\oplus};$$

**caso**  $m_{r\ell}$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r}) + (u_{D_\ell} + u_{\theta_\ell});$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r}) + (u_{D_\ell} + u_{\theta_\ell});$$

**caso**  $m_{r\oplus}$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r}) + u_{D_\oplus};$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r}) + u_{D_\oplus};$$

**caso**  $m_{\ell\oplus}$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + (u_{D_\ell} + u_{\theta_\ell}) + u_{D_\oplus};$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + (u_{D_\ell} + u_{\theta_\ell}) + u_{D_\oplus};$$

**caso**  $m_{r\ell\oplus}$ :

$$MotorD := (u_{D_\lambda} + u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r}) + (u_{D_\ell} + u_{\theta_\ell}) + u_{D_\oplus};$$

$$MotorE := (u_{D_\lambda} - u_{\alpha_\lambda}) + (u_{D_r} + u_{\theta_r}) + (u_{D_\ell} + u_{\theta_\ell}) + u_{D_\oplus};$$

**fim** escolha

---

No capítulo a seguir, é apresentado o algoritmo principal de transporte por enjaulamento e, em seguida, são explicitados os algoritmos de: inicialização da arena, busca, recrutamento e transporte.

## Capítulo 4

# ALGORITMOS DO TRANSPORTE COLETIVO POR ENJAULAMENTO

**E**STE capítulo apresenta os algoritmos de transporte coletivo por enjaulamento desenvolvidos nessa implementação. Na Seção 4.1 são descritos: o algoritmo principal e o algoritmo de inicialização da arena. Na Seção 4.2 explicita-se o algoritmo referente à etapa de busca. A Seção 4.3 apresenta os três algoritmos que compõem a etapa de recrutamento; são eles: o algoritmo *Recrutamento do Enxame*, que pode ser descrito como o algoritmo principal da etapa de recrutamento, e os algoritmos *Recrutador* e *Recrutado*, referentes às ações compreendidas pelo robô recrutador e o robô recrutado, respectivamente. A Seção 4.4 descreve o algoritmo implementado na etapa de transporte.

O algoritmo proposto de transporte coletivo por enjaulamento é estruturado em quatro etapas, como exhibe o Algoritmo 9: *Inicializa da arena*, *Procura*, *Recruta* e *Transporta*. O Algoritmo 9 sintetiza o transporte coletivo por enjaulamento em quatro algoritmos, os quais são descritos nesse capítulo.

---

**Algoritmo 9** Transporte por enjaulamento

---

```
Inicializa arena;
para paralelo  $i := 0 \rightarrow \rho - 1$  faça
  Procura; {Etapa de busca}
fim para
para paralelo  $i := 0 \rightarrow \rho - 1$  faça
  Recruta( $r^*$ ,  $C_{\oplus}$ ); {Etapa de recrutamento}
fim para
para paralelo  $i := 0 \rightarrow \rho - 1$  faça
  Transporta; {Etapa de transporte}
fim para
```

---

## 4.1 Inicialização da Arena

Os passos da etapa de inicialização da arena, descritos no Algoritmo 10, dependem do número total de pontos de referência pertencentes à arena  $\Lambda$ , da função de vizinhança entre os pontos de referência  $\eta(\lambda_i)$  e do número total de robôs  $\rho$ . A partir de tais dados esse algoritmo determina aspectos iniciais e fundamentais para o correto funcionamento dos algoritmos subsequentes.

A etapa de inicialização da arena é caracterizada por definir pontos de referência ao longo de toda a arena, determinar a vizinhança de cada ponto de referência a partir do cálculo da função  $\eta(\lambda_i)$ ; determinar, aleatoriamente, o ponto inicial do centro de massa do objeto  $C_{\oplus} = (x_{\oplus}, y_{\oplus})$  e estabelecer a posição inicial de partida dos robôs na arena  $p_i^0$ . Além disso, a etapa inicialização da arena permite definir a ordem de chamada dos robôs na etapa de recrutamento através da matriz  $\mu_i$ . A definição dos pontos de referência ao longo da arena (discretização da arena) e da vizinhança entre os mesmos, é de extrema importância para a etapa de busca, uma vez que tais aspectos determinarão a forma como os robôs percorrerão a arena. A determinação das posições iniciais dos robôs e do objeto tão possuem significativa importância, pois, tais definições impactam diretamente na procura pelo objeto. Outra importante definição realizada nesse algoritmo é a matriz  $\mu_i$ , pois, a mesma é responsável pela ordem de recrutamento e, conseqüentemente, pelo posicionamento dos robôs na formação de enjaulamento. O Algoritmo 10 descreve o processo de inicialização da arena.

---

### Algoritmo 10 Inicializa arena

---

garante  $C_{\oplus}, p_i^0, \mu_i$ ;  
**para**  $i := 0 \rightarrow \Lambda - 1$  **faça**  
    **Gera randomicamente**  $\lambda_i$  na arena;  
**fim para**  
**para**  $i := 0 \rightarrow \Lambda - 1$  **faça**  
    **Define**  $\eta(\lambda_i)$ ;  
**fim para**  
**Gera randomicamente**  $C_{\oplus} = (x_{\oplus}, y_{\oplus})$ ;  
**para**  $i := 0 \rightarrow \rho - 1$  **faça**  
    **Gera randomicamente**  $p_i^0$ ;  
    **para**  $j := 0 \rightarrow \rho - 2$  **faça**  
        **Define**  $\mu_{ij}$ ;  
    **fim para**  
**fim para**

---

## 4.2 Busca

A Busca tem por objetivo a procura pelo objeto a ser transportado. Nesta etapa, os robôs são inicializados em suas posições iniciais, definidas previamente pelo Algoritmo 10. A região onde tais robôs são inicialmente posicionados é denominada ninho. Após inicializados, cada robô seleciona um ponto de referência para o qual se deslocará. Tal seleção é aleatória, porém, só podem ser selecionados pontos de referência pertencentes à vizinhança da posição atual do robô, definida pela função de vizinhança entre pontos de referência  $\eta(\lambda_i)$ , determinada pelo algoritmo anterior. Ao chegar ao ponto de referência selecionado, o robô seleciona um novo ponto de referência, e tal processo se repete até que o objeto seja detectado.

O principal propósito do processo de busca é a identificação da posição cartesiana do objeto na arena. Portanto, a disposição dos sensores de proximidade ao longo do robô Khepera-III (K-TEAM, 2023) foi de grande importância para o rápido reconhecimento do objeto pelo robô ao aproxima-se do mesmo por qualquer uma de suas extremidades, como exhibe a Figura 9. Outro importante fator nessa etapa é a discretização da arena, pois a mesma permite o melhor espalhamento dos robôs ao longo do cenário, e conseqüentemente, maior eficiência na varredura do ambiente.

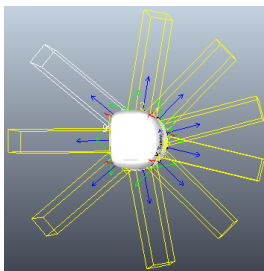


Figura 9: Posicionamento dos sensores no Khepera-III

O Algoritmo 11 descreve a procura do objeto por um robô  $i$ . Inicialmente, escolhe-se, de maneira randômica, o primeiro ponto de referência  $\lambda$  para o qual tal robô  $i$  se movimentará. A partir dessa configuração, o robô parte de sua posição atual  $p_i^t$  para o ponto de referência selecionado. A movimentação do robô, durante a procura, é baseada em uma máquina de estados e dois controladores PID que possuem como variáveis de processo:  $D(p, p') = D(p_i^t, \lambda)$  e  $\alpha_{p'} = \alpha_\lambda$ . Tal máquina de estados e controladores PID estão contidos na função de movimentação dos robôs, descrita no Algoritmo 2. A cada

passo de simulação, todos os sensores do robô  $i$  são analisados. Caso, algum desses detecte o objeto, a movimentação é interrompida, o robô  $i$  é promovido a recrutador  $r^*$  e inicializa-se a etapa de recrutamento. Isso ocorre por meio do envio de mensagens do robô recrutador para o enxame. Ao detectar o objeto, o recrutador envia mensagens do tipo-0, que contêm a posição do centro de massa do objeto e a trajetória do recrutador  $T_{r^*}$  como conteúdo da mensagem. No entanto, se o robô alcançar o ponto de referência  $\lambda$  sem detectar o objeto, o ponto de referência atual é adicionado à trajetória do robô  $T_i$  e um ponto de referência dentro da vizinhança atual é selecionado. Esse processo é repetido até que o robô  $i$  detecte o objeto ou receba uma mensagem anunciando que um recrutador foi identificado. Esse processo é executado até que o objeto seja detectado.

---

**Algoritmo 11** Procura no robô  $i$ 


---

```

found := false;  $r^* := \infty$ ; msg :=  $\infty$ ;  $n_i := 0$ ;  $t := 0$ ;
Seleciona aleatoriamente  $\lambda \in L$ ;
enquanto  $\neg$ found and  $r^* = \infty$  faça
  Calcula  $D_\lambda = D(p_i^t, \lambda)$  a partir da Equação (7);
  Calcula  $\alpha_\lambda$  a partir da Equação (8);
  Movimenta( $D_\lambda, \alpha_\lambda$ );
  para  $k := 0 \rightarrow s - 1$  faça
    se  $\text{sensores}[k].\text{detect} \wedge \text{sensores}[k].\text{id} = \text{id}_\oplus$  então
      found := true;
       $r^* := i$ ;
      para  $j := 0 \rightarrow \rho - 1$  faça
        se  $j \neq r^*$  então
          Envia  $\text{msg}(0, r^*, j, T_i, C_\oplus)$ ;
        fim se
      fim para
    fim se
  para  $\neg$ found e  $r^* = \infty$  então
    Recebe msg;
    se  $\text{msg} \neq \infty$  então
       $r^* := \text{msg}.\text{origin}$ ;
       $C_\oplus := \text{msg}.\text{payload}$ ;
    senão
      se  $p_i^t \approx \lambda$  então
        Adiciona  $\lambda$  em  $T_i$ ;  $n_i := n_i + 1$ ;
        Seleciona aleatoriamente  $\lambda \in \eta(\lambda)$ ;
         $p_i^t := \lambda$ ;
      fim se
    fim se
  fim para
   $t := t + 1$ ;
fim enquanto

```

---

### 4.3 Recrutamento do enxame

O Algoritmo 12 descreve o algoritmo principal da etapa de recrutamento. Neste algoritmo, é determinado se o robô  $i$  atua como o recrutador, ou seja, o primeiro robô a identificar o objeto, ou não. Se o robô  $i$  for designado como o recrutador, ele executa a sequência de instruções exemplificada no Algoritmo 13. Por outro lado, se o robô  $i$  não for o recrutador, ele inicia a sequência de instruções definida no Algoritmo 14.

---

**Algoritmo 12**  $\text{Recruta}(r^*, C_{\oplus})$  no robô  $i$

---

**se**  $i = r^*$  **então**  
 Recrutador( $C_{\oplus}$ );  
**senão**  
 Recrutado( $C_{\oplus}$ );  
**fim se**

---

O recrutamento é iniciado assim que a etapa de busca é finalizada, ou seja, no momento em que um dos robôs encontra o objeto a ser transportado. Ao encontrar o objeto, o recrutador  $r^*$ , além de enviar mensagens a todo o enxame indicando a posição atual do centro de massa do objeto  $C_{\oplus}$  e a sua trajetória  $T_{r^*}$ , também calcula a própria posição ao redor do objeto. Esse cálculo é realizado a partir da Equação 9:

$$\begin{aligned} x_i^* &= x_{\oplus} + ((r_{\oplus} + w) + c) \times \cos(i \times \phi); \\ y_i^* &= y_{\oplus} + ((r_{\oplus} + w) + c) \times \sin(i \times \phi), \end{aligned} \quad (9)$$

onde  $c$  é uma constante adicionada para garantir que o robô não toque no objeto quando posicionado inicialmente e  $w$  denota a largura do robô. A Figura 10 exemplifica o posicionamento inicial dos robôs ao redor do objeto.

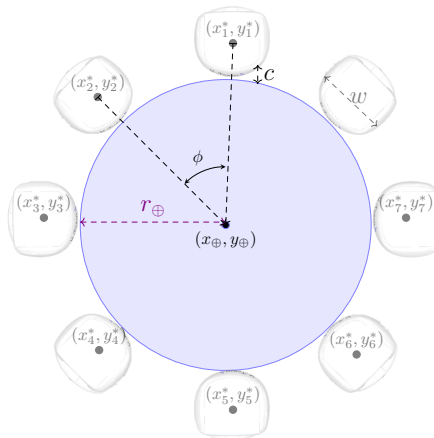


Figura 10: Exemplificação do posicionamento inicial

O Algoritmo 13 descreve as ações executadas pelo robô recrutador. O robô recrutador se move em direção a  $p_i^*$  com base na máquina de estados apresentada no Algoritmo 2, com  $D(p, p') = D(p_i^t, p_i^*)$  que representa a distância entre  $p_i^t$  e a posição inicial de enjaulamento do robô  $i$ ,  $p_i^*$ , e  $\alpha_{p'} = \alpha_{\oplus}$ . Essa distância é calculada a partir da Equação 7, e o ângulo é calculado conforme a Equação 8. Enquanto o recrutador circunda o objeto a fim de encontrar a sua posição inicial de enjaulamento, o enxame aproxima-se do centro de massa  $C_{\oplus}$  do objeto, movimentando-se em tal direção até chegar à distância de aproximação  $\Delta$ , essa distância é calculada de acordo com a Equação 10:

$$\Delta = (r_{\oplus} + w/2 + 0, 6). \quad (10)$$

Após concluir seu posicionamento, o robô  $r^*$  envia uma mensagem do tipo-2 ao enxame para indicar sua conclusão e, em seguida, convoca o primeiro robô conforme prescrito em  $\mu_{r^*}$  e aguarda que tal robô alcance a própria posição inicial de enjaulamento. Esse processo é repetido com os robôs subsequentes, de acordo com a ordem de recrutamento preestabelecida em  $\mu_{r^*}$ , até que todo o enxame esteja posicionado, completando assim o enjaulamento do objeto. O recrutamento é concluído quando o processo de enjaulamento do objeto é finalizado.

---

**Algoritmo 13** Recrutador( $C_{\oplus}$ ) no robô  $i$ 


---

**Calcula**  $p_i^*$  a partir da Equação (9);  
**Calcula**  $\alpha_{\oplus}$  a partir da Equação (8);  
**enquanto**  $\neg(p_i^t \approx p_i^*)$  **faça**  
    **Calcula**  $D_* = D(p_i^t, p_i^*)$  a partir da Equação (7);  
    Movimenta( $D_*$ ,  $\alpha_{\oplus}$ );  
    **Calcula**  $p_i^*$  a partir da Equação (9);  
    **Calcula**  $\alpha_{\oplus}$  a partir da Equação (8);  
**fim enquanto**  
**para**  $j := 0 \rightarrow \rho - 2$  **faça**  
     $r := \mu_{ij}$ ;  
    **Envia** ( $msg(1, i, r, \emptyset)$ );  
    **repita**  
        **Recebe**  $msg$ ;  
        **até** ( $msg.type = 2$ )  $\wedge$  ( $msg.origin = r$ );  
**fim para**

---

O Algoritmo 14 descreve as ações dos robôs recrutados. Inicialmente, o robô  $i$  aproxima-se da posição do centro de massa do objeto  $C_{\oplus}$  até que a distância entre a posição atual do robô  $p_i^t$  e a posição do centro de massa do objeto  $C_{\oplus}$  (distância  $D(p_i^t, C_{\oplus})$ ) seja menor ou igual à distância de aproximação  $\Delta$  definida na Equação 10, ou até que



tal robô receba uma mensagem do tipo-1, indicando que o mesmo deve seguir para sua posição inicial de enjaulamento. Caso o robô  $i$  chegue à distância  $\Delta$  antes de receber tal mensagem, o mesmo aguarda. Ao receber a mensagem do tipo-1, o robô  $i$  calcula a sua posição inicial ao longo do objeto  $p_i^*$  a partir da Equação 9. Uma vez definida  $p_i^*$ , tal robô movimenta-se até que seu centro de massa atinja essa posição, em seguida, tal robô envia uma mensagem de tipo-2 ao recrutador, ou seja, informa que seu posicionamento inicial foi completado.

---

**Algoritmo 14** Recrutado( $C_\oplus$ ) no robô  $i$

---

**Calcula**  $D_\oplus = D(p_i^t, C_\oplus)$  a partir da Equação (7);  
**Calcula**  $\alpha_\oplus$  a partir da Equação (8);  
**enquanto**  $D_\oplus > \Delta \vee \neg msg$  **faça**  
    Movimenta( $D_\oplus, \alpha_\oplus$ );  
    **Calcula**  $D_\oplus = D(p_i^t, C_\oplus)$  a partir da Equação (7);  
    **Calcula**  $\alpha_\oplus$  a partir da Equação (8);  
    **Recebe**  $msg$ ;  
**fim enquanto**  
**se**  $D_\oplus \leq \Delta$  **então**  
    **Recebe**  $msg$ ;  
**fim se**  
**se**  $(msg.type = 1) \wedge (msg.origin = r^*)$  **então**  
    **Calcula**  $p_i^*$  a partir da Equação (9);  
    **Calcula**  $\alpha_\oplus$  a partir da Equação (8);  
    **enquanto**  $\neg(p_i^t \approx p_i^*)$  **faça**  
        **Calcula**  $D_* = D(p_i^t, p_i^*)$  a partir da Equação (7);  
        Movimenta( $D_*, \alpha_\oplus$ );  
        **Calcula**  $p_i^*$  a partir da Equação (9);  
        **Calcula**  $\alpha_\oplus$  a partir da Equação (8);  
    **fim enquanto**  
    **para**  $j := 0 \rightarrow \rho - 1$  **faça**  
        **se**  $j \neq i$  **então**  
            **Envia**  $(msg(2, i, r^*, \emptyset))$ ;  
        **fim se**  
    **fim para**  
**fim se**

---

O processo de recrutamento é fortemente baseado na troca de mensagens entre os robôs. Portanto, a comunicação massiva, por troca de sinais é parte relevante e fundamental no desenvolvimento dessa etapa. Outro ponto fundamental para o correto desenvolvimento da etapa de recrutamento, é a capacidade que os robôs devem possuir de contornar o objeto e os demais robôs ao aproximarem-se deles. Pois, tal capacidade garante que não existam colisões entre robôs durante a movimentação dos mesmos, inclusive na etapa de busca; e que durante a alocação de robôs, um a um, no entorno do objeto, erros prove-

nientes de possíveis colisões sejam minimizados. Garantindo assim, o correto e eficiente enjaulamento do objeto. A implementação dessa capacidade foi baseada no Veículo de Braitenberg apresentado pela plataforma CoppeliaSim em algumas de suas simulações de exemplificação. Utilizou-se o algoritmo apresentado no código de exemplificação do robô e-puck, dado pela plataforma (COPPELIA ROBOTICS, 2023; BRAITENBERG, 1986).

## 4.4 Transporte

Uma vez que o enjaulamento é concluído, inicia-se a etapa de transporte. Tal etapa, assim como a anterior, é caracterizada pela massiva comunicação entre os robôs. Pois, para que a mesma seja inicializada, é preciso que todos os robôs identifiquem que estão em suas posições iniciais corretas e transmitam tal informação aos demais. Apenas, após todos os robôs terem ciência que ele próprio e os demais robôs do enxame estão prontos para darem início ao transporte, ou seja, tenham ciência de que o enjaulamento está corretamente finalizado, as etapas subsequentes do transporte são iniciadas.

A primeira etapa do transporte é a identificação do caminho que deve ser percorrido pelo enxame, transportando o objeto da posição a qual foi encontrado para o ninho. Esse caminho é definido pelo recrutador. Ou seja, trata-se do trajeto percorrido pelo recrutador durante a etapa de busca, porém, ao contrário; de forma que o enxame passe por cada  $\lambda_i$  percorrido pelo recrutador até a chegada ao ninho. O armazenamento do trajeto (pontos de referência percorridos) é realizado na etapa de busca, e a escolha do trajeto trilhado pelo robô recrutador, se dá pelo fato desse ser o primeiro robô a chegar até o objeto, portanto, admite-se que tal percurso é o menor entre a posição inicial do objeto e o ninho.

Após o reconhecimento do caminho, os robôs identificam o  $\lambda_i$  para o qual o objeto deve ser transportado, em seguida, orientam-se, individualmente, na direção que deve ser seguida, e após a comunicação do enxame e identificação coletiva de que todos os robôs estão corretamente orientados, dão início à aplicação de forças ao objeto, transportando-o até que o mesmo atinja tal  $\lambda_i$ . Após chegarem, verificam se a localização atual é a posição final do caminho, caso seja, o transporte é finalizado; do contrário, seguindo os mesmos processos de comunicação, orientação e movimentação, realizam o transporte,  $\lambda_i$  a  $\lambda_i$ , percorrendo todo o caminho definido até o ninho.

O principal desafio da etapa de transporte é a manutenção do enjaulamento ao longo do trajeto; além da correta aplicação de forças ao objeto, por cada um dos robôs de

acordo com a posição que ocupam ao redor do objeto em cada passo da simulação. Pois, ao longo do caminho, um mesmo robô pode estar posicionado tanto na parte posterior do objeto, e a força por ele empregada tenha função de deslocamento; como em outro momento da mesma simulação, devido a orientação do  $\lambda_i$  para o qual o transporte está orientado, o mesmo robô seja pertencente à lateral ou até mesmo à região frontal do objeto, e o emprego de sua força terá por objetivo a manutenção do enjaulamento, manutenção da orientação do objeto e resistência, agindo como força contrária a de deslocamento e possibilitando a movimentação coordenada do objeto e minimizando possíveis desvios do curso retilíneo da trajetória.

Visando realizar tal implementação sem definir função específica para cada robô; o que acarretaria, a cada passo da simulação, na necessidade de movimentar os robôs ao redor do objeto. Empregou-se uma máquina de estados e controladores PID também para manutenção do enjaulamento como descrito no Algoritmo 4.

O Algoritmo 15 apresenta a rotina de transporte no robô  $i$ . Inicialmente, o robô  $i$  aguarda o recebimento de mensagens do tipo-2 de todos os  $\rho$  robôs. Pois, o recebimento dessas mensagens indica que o enjaulamento foi finalizado e o transporte pode ser iniciado. A trajetória  $T_{r^*}$  a ser usada para transportar o objeto é a mesma que a trajetória seguida pelo robô recrutador  $r^*$  mas na ordem inversa. Nesse caso, a orientação do robô é realizada a partir da máquina de estados apresentada no Algoritmo 3 e um controlador PID com  $\alpha_{p'} = \alpha_\lambda$ . Após orientar-se na direção de  $\lambda$ , o robô  $i$  informa ao enxame por meio de uma mensagem do tipo-3 sobre a conclusão do seu processo de orientação. Em seguida, aguarda a confirmação do enxame por meio do mesmo tipo de mensagem, indicando a conclusão desse mesmo processo por todo o enxame. Quando todos os robôs estão orientados na direção do ponto de referência, inicializa-se a movimentação simultânea do enxame, gerando o transporte do objeto.

A movimentação do robô  $i$  durante o transporte se dá a partir do Algoritmo 4, e tal movimentação é tida como concluída, quando a posição atual do robô  $p_i^t$  coincide com a posição que o mesmo deve atingir quando o objeto chegar ao ponto de referência  $p_i^+$ . A posição  $p_i^+$  é calculada de maneira semelhante a  $p_i^*$  conforme apresentado na Equação 9. No momento em que  $p_i^t$  atinge  $p_i^+$ , o robô  $i$  envia mensagens do tipo-4 a todo o enxame, informando que o ponto de referência foi por ele alcançado; em seguida, aguarda a confirmação de que todo o enxame também alcançou  $\lambda$  através desse mesmo

tipo de mensagem. Tal processo se repete até que toda a trajetória  $T_{r^*}$  seja percorrida e consequentemente, o transporte do objeto, concluído.

---

**Algorithm 15** Transporta no robô  $i$

---

```

 $j := 0;$ 
enquanto  $j < \rho$  faça
  Recebe  $msg;$ 
  se  $(msg.type = 2) \wedge (msg.origin = j) \vee j \neq i$  então
     $j := j + 1;$ 
  fim se
fim enquanto
para  $j := n - 1 \rightarrow 0$  faça
   $\lambda := T_{r^*j};$ 
  repita
    Calcula  $\alpha_\lambda$  a partir da Equação (8);
    Orienta $(\alpha_\lambda);$ 
  até  $\beta_i^t \approx \alpha_\lambda$ 
  para  $j := 0 \rightarrow \rho - 1$  faça
    se  $j \neq i$  então
      Envia  $msg\langle 3, i, j, \emptyset \rangle;$ 
    fim se
  fim para
  enquanto  $k < \rho - 1$  faça
    Recebe  $msg;$ 
    se  $(msg.type = 3) \wedge (msg.origin = k) \vee k \neq i$  então
       $k := k + 1;$ 
    fim se
  fim enquanto
  repita
    Calcula  $D_\lambda = D(p_i^t, \lambda), D_r = D(p_i^t, r)$  a partir da Equação (7);
    Calcula  $D_\ell = D(p_i^t, \ell)$  e  $D_\oplus = D(p_i^t, C_\oplus)$  a partir da Equação (7);
    Calcula  $\alpha_\lambda, \theta_r$  e  $\theta_\ell;$ 
    Enjaula $(D_\lambda, D_r, D_\ell, D_\oplus, \alpha_\lambda, \theta_r, \theta_\ell);$ 
  até  $p_i^t \approx p_i^+$ 
  para  $j := 0 \rightarrow \rho - 1$  faça
    Envia  $msg\langle 4, i, j, \emptyset \rangle;$ 
  fim para
   $k := 0;$ 
  enquanto  $k < \rho - 1$  faça
    Recebe  $msg;$ 
    se  $(msg.type = 4) \wedge (msg.origin = k) \vee (k \neq i)$  então
       $k := k + 1;$ 
    fim se
  fim enquanto
fim para

```

---

## 4.5 Considerações Finais do Capítulo

Neste capítulo são apresentados os algoritmos que compõem o transporte por enjaulamento. Inicialmente, exibe-se um algoritmo principal que descreve a estrutura da implementação; em seguida, são descritos os algoritmos de inicialização da arena, busca, recrutamento e transporte. A Seção 4.3 apresenta também um algoritmo principal referente à etapa de recrutamento e, em seguida os algoritmos que o compõem, referentes ao algoritmo do robô recrutador e do robô recrutado. Enquanto que a Seção 4.4 explicita o algoritmo de transporte.

O capítulo a seguir, apresenta os experimentos realizados e os resultados obtidos. São apresentados estudos de caso para enjaulamentos com diferentes números de robôs e objetos com distintas dimensões.

# Capítulo 5

## AVALIAÇÃO DE DESEMPENHO

**E**STE capítulo tem por objetivo apresentar o desempenho desse trabalho nas etapas de busca, recrutamento e transporte. A Seção 5.1 exhibe os aspectos da implementação. A Seção 5.2 apresenta os parâmetros utilizados, a Seção 5.3 descreve os impactos dessa implementação nas etapas de busca e recrutamento. A Seção 5.4 apresenta a avaliação de desempenho da etapa de transporte através de três estudos de caso, e a Seção 5.5 apresenta as considerações finais do capítulo.

### 5.1 Aspectos de Implementação

Adotou-se a plataforma CoppeliaSim (COPPELIA ROBOTICS, 2023) como ambiente de simulação para o desenvolvimento desse projeto. Essa escolha foi motivada pelo fato de a plataforma possuir modelos de robôs físicos em seu acervo, possibilitando a futura implementação das simulações desenvolvidas em modelos físicos. Porém, para que a implementação física desse projeto possa ser viabilizada, alguns fatores devem ser observados. Pois, devido à necessidade de comunicação contínua entre os robôs, a implementação física deste projeto exigiria a delimitação da arena de tal maneira que os sinais entre os robôs possam ser transmitidos e recebidos de forma confiável. Além disso, no cenário real, os tempos de comunicação e computação dos robôs apresentam notável disparidade em relação aos valores obtidos em ambientes simulados. Dessa forma, a análise metódica desses parâmetros é crucial para a transição efetiva desta implementação para um contexto físico, pois, é de extrema importância nesse projeto que os robôs realizem alguns processos de forma simultânea.

A plataforma CoppeliaSim tem como vantagem a flexibilidade quanto a linguagem de programação, uma vez que o CoppeliaSim permite não apenas o uso da linguagem Lua

(API regular- funções disponibilizadas pelo próprio CoppeliaSim), como também o uso de linguagens como C, Python e MatLab, através de API remota (funções disponibilizadas por plugins específicos). A plataforma CoppeliaSim também possui fórum e tutorias, ferramentas que facilitam a integração do projetista com o ambiente de desenvolvimento, e que foram fundamentais na decisão desta como o ambiente de desenvolvimento deste projeto.

A linguagem Lua foi escolhida por permitir, dentro da plataforma, a implementação do paralelismo. Uma vez que cada robô possui seu próprio código e o executa de forma independente dos demais. Quanto ao robô, escolheu-se o robô Khepera-III (K-TEAM, 2023), sem adição de garras, devido ao seu formato frontal arredondado, o que permite bom contato entre a parte frontal do robô e o contorno do objeto, detalhe fundamental para as etapas de enjaulamento e transporte do objeto.

A arena é o cenário onde as simulações são realizadas na plataforma CoppeliaSim. Nessa implementação, utilizou-se arenas com dimensões  $5m \times 5m$  e  $10m \times 10m$  e as mesmas foram discretizadas com passo de  $0,5m$  e  $1m$ . As arenas utilizadas são planas. O objeto a ser transportado possui forma cilíndrica e as dimensões do mesmo variam de acordo com o estudo de caso apresentado na Seção 5.4.2.

## 5.2 Parâmetros de Simulação

A movimentação dos robôs foi implementada a partir de máquinas de estado e controladores PID. A Tabela 1 apresenta os valores das componentes proporcionais, integrais e derivativas utilizadas nos controladores PID. Os valores empregados às constantes proporcionais, integrais e derivativas foram ajustados, através de várias simulações, para atingir o controle desejado. A Tabela 2 exibe as constantes de tolerância empregadas nas máquinas de estados. A Tabela 3 apresenta as características físicas dos robôs como a largura  $w$  e a massa  $m$  dos mesmos, além das constantes utilizadas nas equações.

Tabela 1: Constantes proporcionais, integrais e derivativas utilizadas nos controladores PID

PIDs	$\alpha_{p'}$	$D(p_i^t, \lambda)$	$D(p_i^t, \ell), D(p_i^t, r)$	$\theta_\ell, \theta_r$	$D(p_i^t, C_\oplus)$
$k_p$	1	1	0,3	0,3	1
$k_i$	0,01	0,01	0,01	0,01	0,01
$k_d$	0,01	0,01	0,1	0,1	0,1

Tabela 2: Constantes de tolerância empregadas nas máquinas de estados

$d$ (m)	$\sigma$ (°)
0,05	3

Tabela 3: Dimensões dos robôs e contantes utilizadas nas equações:

$w$ (m)	$m$ (kg)	$\epsilon$ (m)	$c$ (m)	$\Delta$ (m)
0,122	0,65	$w/2$	0,01	$(r_{\oplus} + w/2 + 0,6)$

### 5.3 Busca e Recrutamento

A etapa de busca foi implementada a partir da definição de uma vizinhança entre os pontos de referência na arena. De forma que, a partir do ponto de referência atual, o robô escolhe, aleatoriamente, dentro de uma vizinhança pré-definida, um novo ponto de referência para onde deve se deslocar. A função de vizinhança  $\eta(\lambda_i)$  foi implementada a partir da Equação 11:

$$\eta(\lambda_i) = \lambda_{i,j+1}, \lambda_{i+1,j+1}, \lambda_{i-1,j+1}, \quad (11)$$

onde  $i$  representa a linha a qual pertence o ponto de referência e  $j$  representa a coluna. Uma vez que os pontos de referência foram dispostos de forma matricial na arena. Ou seja, dessa forma, os vizinhos de cada ponto de referência são pertencentes à coluna de pontos de referência seguinte, o que permite que o robô avance continuamente em sua procura pela arena. Enquanto que a escolha aleatória entre pontos de referência dentro da vizinhança preestabelecida, garante maior espalhamento dos robôs ao longo da arena durante a etapa de busca.

A Figura 11 apresenta as etapas de busca e recrutamento para um enxame composto por seis robôs ( $\rho = 6$ ) e uma arena de dimensões  $5m \times 5m$  com passo de discretização de  $1m$ .

A análise da etapa de busca foi realizada a partir de dois estudos de caso, quatro experimentos e cinco simulações para cada experimento, totalizando vinte simulações. A Tabela 4 apresenta os resultados obtidos para etapa de busca. Nesta tabela são apresentadas duas arenas, uma com dimensões  $5m \times 5m$  e outra com dimensões  $10m \times 10m$ . Discretizou-se cada uma dessas arenas em dois diferentes passos:  $0,5m$  e  $1m$ . Nesta tabela são apresentados os tempos médios obtidos para a execução da etapa de busca em cada experimento.



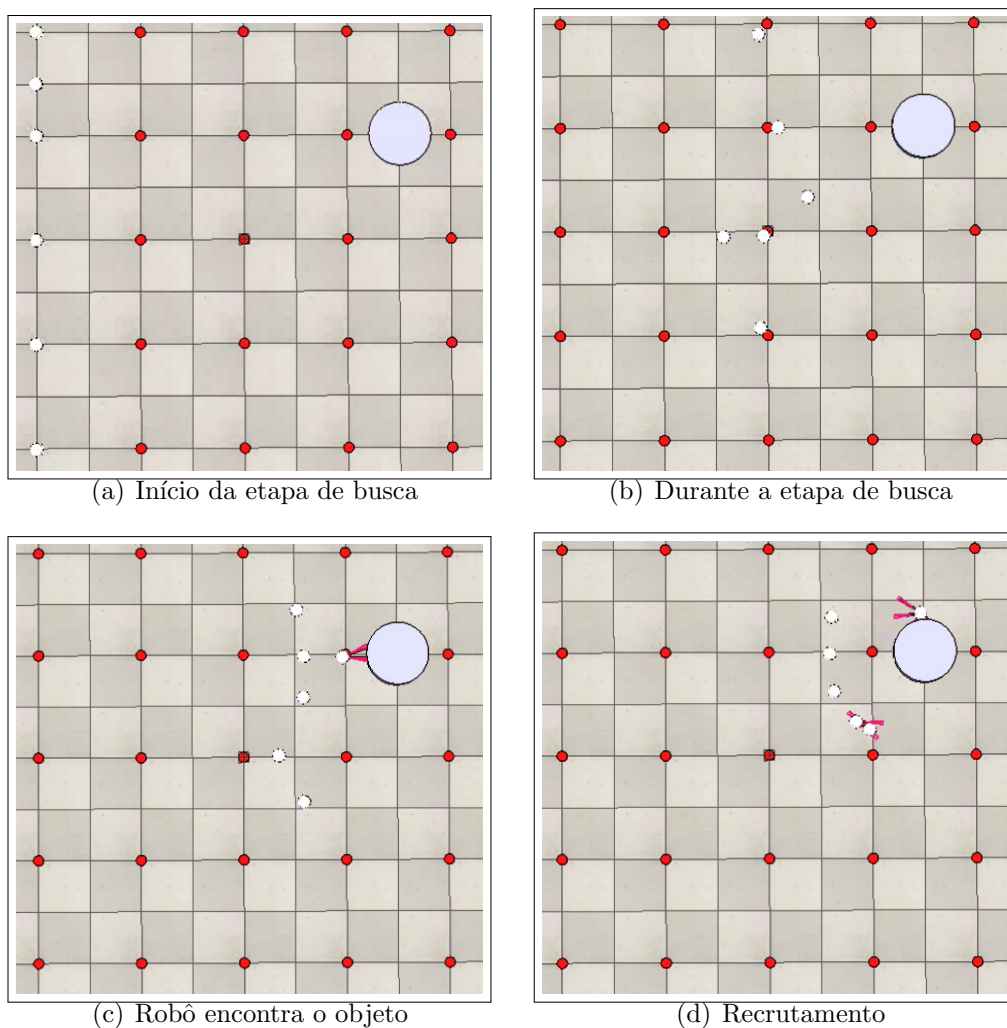


Figura 11: Etapas de busca e recrutamento

Tabela 4: Resultados da etapa de busca

Arena ( $m$ )	Passo ( $m$ )	Tempo médio ( $s$ )
$5 \times 5$	0,5	70,56
	1,0	60,12
$10 \times 10$	0,5	222,165
	1,0	197,535

A análise de desempenho das etapas de busca e recrutamento foi realizada a partir da análise do tempo médio necessário para o desenvolvimento dessas etapas para diferentes passos de discretizações da arena. Observou-se que na etapa de busca, o passo de discretização da arena tem impacto direto sobre tal grandeza. Isso ocorre, pois, quanto menor a distância entre o robô e o ponto de referência, menor é a velocidade do robô, a fim

de que haja melhor ajuste em sua rota. Portanto, quanto menor o passo de discretização, menor a velocidade atingida pelos robôs ao longo do percurso da etapa de busca.

Enquanto que o tempo médio necessário para a execução da etapa de recrutamento independe da discretização da arena, o que impacta essa etapa é a distância entre o enxame e o objeto no momento em que o mesmo é detectado. O tempo de execução da etapa de recrutamento é também impactado pela ordem de posicionamento inicial preestabelecida ( $\mu_i$ ).

Finalizada a etapa de recrutamento, inicia-se o posicionamento inicial do enxame em torno do objeto. Nessa etapa adota-se uma ordem de recrutamento preestabelecida  $\mu_r^*$ . Em outras palavras, para cada robô  $i$  que se torna o recrutador  $r^*$ , há uma sequência preestabelecida de chamadas para o posicionamento inicial. O robô recrutador, inicialmente, convoca seus vizinhos à esquerda e à direita. Após o posicionamento desses vizinhos, os robôs que ocupam as posições imediatamente ao lado dos vizinhos do recrutador também são alocados. Esse processo é repetido até que todo o enxame seja posicionado, concluindo assim o enjaulamento do objeto. A implementação de uma ordem de posicionamento inicial predefinida, representada por  $\mu_i$ , objetiva prevenir possíveis colisões entre robôs, bem como entre os robôs e o objeto durante a etapa de posicionamento inicial. A Figura 12 apresenta a etapa de posicionamento inicial. Finalizado o posicionamento inicial, é também concluído o enjaulamento do objeto, como apresenta a Figura 12(b).

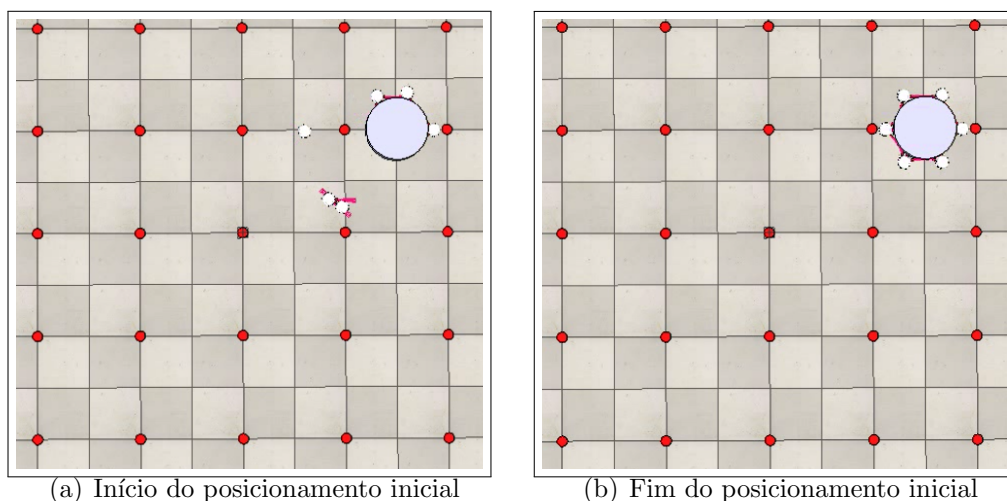


Figura 12: Etapa de posicionamento inicial

## 5.4 Transporte

A análise de desempenho da etapa de transporte foi baseada em três estudos de caso, nos quais são avaliados os impactos nos resultados obtidos para diferentes números de robôs, objetos com diferentes dimensões e arenas com distintos passos de discretização. A Seção 5.4.1 apresenta os resultados para enxames com diferentes números de robôs. A Seção 5.4.2 exibe o impacto das simulações para objetos cilíndricos com diferentes dimensões e a Seção 5.4.3 discute os impactos dos resultados obtidos para arenas com diferentes passos de discretização. A Figura 13 ilustra a etapa de transporte, para um enxame composto por oito robôs ( $\rho = 8$ ) e uma arena com dimensões  $5m \times 5m$  com passo de discretização de  $0,5m$ .

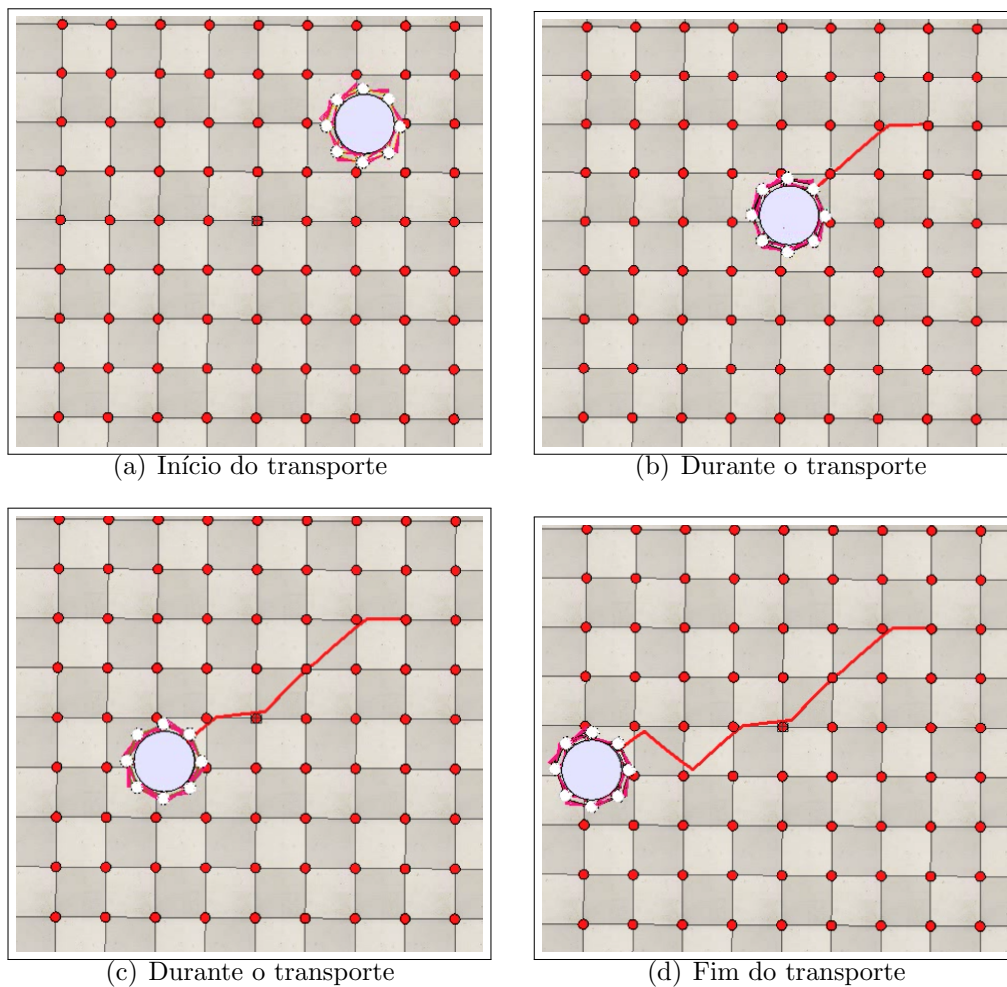


Figura 13: Etapa de transporte

### 5.4.1 Impacto do Número de Robôs

Objetivando analisar o impacto do número de robôs no transporte do objeto, definiu-se três enxames com tamanhos distintos, ou seja, cada um dos enxames possui uma quantidade de robôs preestabelecida. Foram definidos enxames com as seguintes quantidades de robôs:  $\rho = 6$ ,  $\rho = 7$  e  $\rho = 8$ . Para essa implementação, utilizou-se uma arena com dimensões  $5m \times 5m$  e passo de discretização igual a  $1m$ . O objeto possui forma cilíndrica com raio  $r_{\oplus} = 0,3m$  e massa  $M = 0,3kg$ . As simulações completas para este estudo de caso estão disponíveis em: [linktr.ee/numeroderobos](http://linktr.ee/numeroderobos).

Analisou-se o erro do trajeto percorrido pelo objeto durante o transporte para os três diferentes números de robôs e o tempo necessário para a conclusão de todas as etapas que constituem o transporte coletivo por enjaulamento, ou seja: a busca, o recrutamento, o posicionamento inicial e o transporte. O erro é calculado a partir da diferença absoluta entre o trajeto que deve ser percorrido pelo objeto durante o transporte  $T_{r^*}$  e o caminho percorrido pelo centro de massa do objeto. Ou seja, calculou-se a distância mínima a ser percorrida a partir do trajeto contido em  $T_{r^*}$ . Tal distância foi obtida através do somatório das distâncias entre os pontos de referência pertencentes à  $T_{r^*}$ . Em seguida, calculou-se a distância percorrida pelo centro de massa do objeto durante o transporte, esse cálculo foi obtido a partir do somatório das distâncias entre as posições que o centro de massa do objeto atingiu, durante o percurso, quando o enxame considerava ter chegado ao ponto de referência desejado. Portanto, a partir da diferença absoluta entre tais percursos, foi possível definir o erro do trajeto percorrido pelo centro de massa do objeto. O cálculo do percurso ideal é obtido a partir da Equação 12:

$$PI = D(C_{\oplus}, T_{r^*0}) + \sum_{j=0}^{n-1} D(T_{r^*j}, T_{r^*j+1}), \quad (12)$$

onde  $n$  representa o tamanho do vetor de trajetória  $T_{r^*}$ , a distância  $D(C_{\oplus}, T_{r^*0})$  é a distância entre o centro de massa do objeto, na posição em que o mesmo foi encontrado, e a posição do primeiro ponto de referência para o qual o objeto deve ser transportado. A distância  $D(T_{r^*j}, T_{r^*j+1})$  representa a distância entre pontos de referência subsequentes pertencentes à trajetória a ser seguida durante o transporte  $T_{r^*}$ . O cálculo do percurso real percorrido pelo centro de massa do objeto é calculado a partir da Equação 13:

$$PR = D(C_{\oplus}, C_{\oplus}^{T_{r^*0}}) + \sum_{j=0}^{n-1} D(C_{\oplus}^{T_{r^*j}}, C_{\oplus}^{T_{r^*}(j+1)}), \quad (13)$$

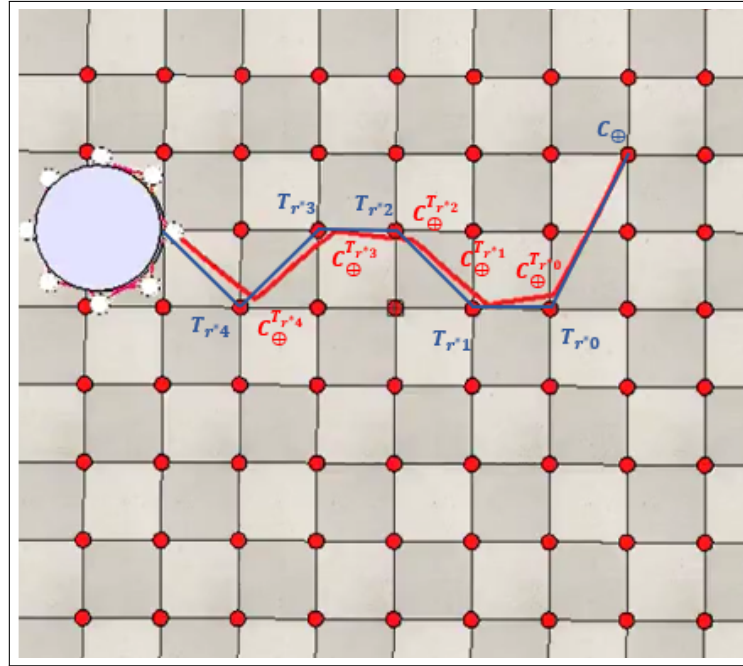


Figura 14: Análise do erro entre o percurso ideal  $PI$  e o percurso real  $PR$

onde a distância  $D(C_{\oplus}^{T_{r^*j}}, C_{\oplus}^{T_{r^*(j+1)}})$  representa a distância entre as posições subsequentes atingidas pelo centro de massa do objeto, quando, durante o transporte, o enxame assume que o centro de massa do objeto atingiu o ponto de referência desejado. O cálculo do erro é obtido a partir da diferença absoluta entre o percurso ideal e o percurso real percorrido pelo centro de massa do objeto, como apresenta a Equação 14:

$$E = |PR - PI|. \quad (14)$$

A Figura 14 apresenta o percurso ideal  $PI$ , em azul, e o percurso real  $PR$  percorrido pelo centro de massa do objeto, em vermelho, e denota o erro  $E$  entre tais percursos.

Nesse estudo de caso, onde analisa-se o impacto do número de robôs no erro de percurso, foram realizados três experimentos e uma simulação por experimento, totalizando três simulações. Cada experimento contém uma quantidade específica de robôs, foram considerados: seis, sete e oito robôs. A arena possui dimensões  $5m \times 5m$  e passo de discretização igual a  $1,0m$ . O objeto a ser transportado possui massa igual a  $0,3kg$  e raio igual a  $0,3m$ .

A partir da análise da Tabela 5, nota-se que o erro médio percentual para os três diferentes números de robôs é de aproximadamente 4%, enquanto que o erro médio em metros é de aproximadamente  $0,17m$ . Pode-se dizer que o erro de percurso se mantém aproximadamente constante nessa implementação independentemente do número de robôs

pertencentes ao enxame. Quanto ao tempo de execução, observa-se que quanto mais robôs no enxame, maior o tempo necessário para a completa execução do transporte por enjaulamento. Esse fato deve-se ao posicionamento inicial dos robôs em torno do objeto. Pois, quanto mais robôs, mais tempo é necessário para que os mesmos sejam corretamente alocados ao redor do objeto, concluindo assim o enjaulamento do objeto. Pois, nesta etapa, os robôs devem contornar o objeto e evitar colisões tanto com o objeto como com os demais robôs, o que gera expressiva demanda de tempo para que os robôs se aloquem corretamente em suas posições iniciais de enjaulamento. Observa-se que o aumento na quantidade de robôs não resulta em aceleração no tempo de execução da etapa de transporte, uma vez que esses robôs operam de forma constante em suas velocidades máximas.

Tabela 5: Resultados das simulações para enxames compostos por diferentes números de robôs

$\rho$	$PI (m)$	$PR (m)$	$E (m)$	$E (\%)$	Tempo de Transporte (s)	Tempo Total (s)
6	4,3284	4,1505	0,1778	4,1097	70,8	215,07
7	4,5322	4,3427	0,1894	4,1789	71,4	239,35
8	3,9142	3,7615	0,1526	3,9003	72,6	244,79

### 5.4.2 Impacto das Características do Objeto

Esse estudo de caso tem o objetivo de examinar o impacto das dimensões físicas do objeto no transporte, de forma a verificar se tais dimensões impactam de forma expressiva no erro da trajetória percorrida pelo centro de massa do objeto. Foram realizados três experimentos, uma simulação por experimento, totalizando três simulações. Para análise de tal estudo de caso, manteve-se a arena com dimensões  $5m \times 5m$  e passo de discretização igual a  $1m$ . Determinou-se que o enxame seria composto por oito robôs,  $\rho = 8$ , e que o objeto possuiria três distintas dimensões. As simulações completas para este estudo de caso estão disponíveis em: [linktr.ee/objetodimensoes](http://linktr.ee/objetodimensoes). A Tabela 6 apresenta os resultados obtidos para tais simulações. O cálculo do percurso ideal  $PI$  é obtido pela Equação 12, o cálculo do percurso real é obtido pela Equação 13 e o erro é calculado a partir da Equação 14. Através da análise da Tabela 6, nota-se que o erro médio percentual na trajetória, nesse caso, é de aproximadamente 3,6% enquanto que o erro médio em metros é de aproximadamente 0,16m. Pode-se dizer que independentemente das dimensões do

objeto, o erro de percurso se mantém aproximadamente constante, ressaltando a robustez da metodologia proposta. Observa-se também que o aumento da massa do objeto não resulta em maior tempo necessário para a execução da etapa de transporte, uma vez que esses robôs operam de forma constante em suas velocidades máximas. Ou seja, os robôs mantêm uma velocidade constante, independentemente da massa do objeto.

Tabela 6: Resultados das simulações para enxame composto por oito robôs e objetos com distintas dimensões

$r_{\oplus}$ (m)	$M$ (kg)	$PI$ (m)	$PR$ (m)	$E$ (m)	$E$ (%)	Tempo de Transporte (s)	Tempo Total (s)
0,30	0,3	3,9142	3,7615	0,1526	3,9003	72,6	244,79
0,40	0,7	4,3284	4,1598	0,1686	3,8954	75,0	282,47
0,45	1,0	4,3284	4,1809	0,1474	3,4073	73,2	258,62

### 5.4.3 Impacto da Discretização da Arena

Objetivando analisar o impacto da discretização da arena no transporte coletivo por enjaulamento, foram implementadas simulações semelhantes às descritas nas seções anteriores, mantendo a mesma arena de dimensões  $5m \times 5m$ , porém utilizando-se dois diferentes passos de discretização:  $0,5m$  e  $1,0m$ . A arena com passo de discretização de  $0,5m$  é composta por 25 pontos de referência, enquanto que a arena com passo de discretização de  $1,0m$  é composta por 81 pontos de referência. As simulações completas estão disponíveis em: [linktr.ee/discretizacaodaarena](http://linktr.ee/discretizacaodaarena).

#### 5.4.3.1 Impacto da Discretização da Arena Variando o Número de Robôs

Analisa-se se há impacto no erro de percurso obtido para arenas com diferentes passos de discretização, variando o número de robôs. Foram realizados seis experimentos e uma simulação para cada experimento, totalizando seis simulações.

A Tabela 7 apresenta os resultados obtidos para enxames compostos por diferentes números de robôs. A partir da análise de tal tabela, observa-se que quanto menor o passo de discretização da arena, maior o erro de percurso. Semelhantemente ao analisado no estudo de caso que avalia o impacto do número de robôs para a arena com passo de discretização de  $1,0m$ , quanto maior a quantidade de robôs, maior o tempo de execução do transporte coletivo por enjaulamento, devido ao impactado da etapa de posicionamento

inicial. Ou seja, quanto mais robôs, mais tempo é necessário para que os mesmos se posicionem corretamente em torno do objeto, enjaulando-o.

O erro médio percentual para a arena com passo de discretização de  $0,5$  é de aproximadamente  $6,2\%$ , enquanto que o erro médio em metros é de aproximadamente  $0,28m$ . Nota-se que há um relevante aumento em relação à arena com discretização de  $1,0m$ , onde o erro médio percentual é de aproximadamente  $4\%$ . Tal fato se dá devido a maior quantidade de pontos de referência entre a posição inicial do objeto na arena a e posição alvo para a qual o mesmo é transportado. Portanto, quanto menor o passo de discretização da arena, mais pontos de referência pertencem ao trajeto a ser percorrido. Logo, mais erros locais são produzidos e conseqüentemente, maior é o erro acumulado durante o percurso.

Tabela 7: Resultados das simulações para arenas com passos de discretização de  $0,5m$  e de  $1,0m$  e enxames compostos por diferentes números de robôs

$\rho$	Passo(m)	PI (m)	PR (m)	E (m)	E (%)	Tempo de Transporte (s)	Tempo Total (s)
6	0,5	4,5322	4,2901	0,2420	5,3396	121,2	280,11
	1,0	4,3284	4,1505	0,1778	4,1097	70,8	215,07
7	0,5	4,3284	4,0468	0,2815	6,5035	120,6	318,33
	1,0	4,5322	4,3427	0,1894	4,1789	71,4	239,35
8	0,5	4,7393	4,4172	0,3220	6,7942	127,8	330,00
	1,0	3,9142	3,7615	0,1526	3,9003	72,6	244,79

#### 5.4.3.2 Impacto da Discretização da Arena Variando as Características do Objeto

Analisa-se se há impacto no erro de percurso obtido para arenas com diferentes passos de discretização, variando as características do objeto. Foram realizados seis experimentos e uma simulação para cada experimento, totalizando seis simulações.

A Tabela 8 apresenta os resultados obtidos para enxames compostos por oito robôs e objetos com distintas dimensões. Nota-se que, para arena com passo de discretização de  $0,5m$ , o erro médio percentual do percurso é de aproximadamente  $6,3\%$ , enquanto que o erro médio em metros é de aproximadamente  $0,29m$ . Comparando tais erros com os obtidos para os mesmos casos com passo de discretização de  $1,0m$ , observa-se aumento. portanto, através da análise de tal tabela, observa-se que, quanto menor o passo de discretização da arena, maior o erro acumulado no percurso. Dessa forma, pode-se concluir que quanto menor o passo de discretização da arena, mais pontos de referência pertencem



ao trajeto a ser percorrido. Logo, mais erros locais são produzidos e conseqüentemente, maior é o erro acumulado durante o percurso.

Tabela 8: Resultados das simulações para arenas com passos de discretização de  $0,5m$  e de  $1,0m$ , enxame composto por oito robôs e objetos com distintas dimensões

$r_{\oplus}$ (m)	$M$ (kg)	Passo (m)	PI (m)	PR (m)	$E$ (m)	$E$ (%)	Tempo de Transporte (s)	Tempo Total (s)
0,30	0,3	0,5	4,7393	4,4172	0,3220	6,7942	127,8	330,00
		1,0	3,9142	3,7615	0,1526	3,9003	72,6	244,79
0,40	0,7	0,5	4,7393	4,4011	0,3381	7,1339	124,8	350,13
		1,0	4,3284	4,1598	0,1686	3,8954	75,0	282,47
0,45	1,0	0,5	4,3284	4,1128	0,2156	4,9810	121,2	312,09
		1,0	4,3284	4,1809	0,1474	3,4073	73,2	258,62

A partir dessas análises, pode-se concluir que, para arenas com mesma configuração, o erro percentual da trajetória do centro de massa do objeto é aproximadamente contante independentemente do número de robôs ou das dimensões do objeto. O erro da trajetória do centro de massa do objeto é impactado apenas pelo passo de discretização da arena.

## 5.5 Considerações Finais do Capítulo

Neste capítulo, é apresentada a avaliação de desempenho do algoritmo proposto para o transporte coletivo por enjaulamento. São apresentados os aspectos da implementação como a plataforma de simulações, a linguagem de programação e os robôs utilizados. São também apresentados os parâmetros de simulação que foram empregados, incluindo as constantes proporcionais, integrais e derivativas utilizadas em cada controlador PID presente na implementação, além das constantes de tolerância utilizadas nas máquinas de estados e equações. A avaliação de desempenho das etapas de busca e recrutamento é realizada a partir da análise do tempo de execução das mesmas para diferentes passos de discretização da arena. Enquanto que a avaliação de desempenho do transporte é obtida através da análise de estudos de caso, onde são avaliados os impactos do número de robôs, das dimensões do objeto e da discretização da arena no erro da trajetória percorrida pelo centro de massa do objeto.

O capítulo a seguir conclui esta dissertação, apresentando as principais conclusões derivadas deste estudo e destacando os principais pontos abordados. Além disso, tal capítulo também delinea possíveis direções para trabalhos futuros.

## Capítulo 6

# CONCLUSÕES E TRABALHOS FUTUROS

**N**ESTE capítulo, apresentam-se as conclusões acerca do modelo de transporte coletivo por enjaulamento proposto neste trabalho. A Seção 6.1 destaca as principais conclusões resultantes do desenvolvimento desta dissertação. Enquanto que a Seção 6.2 explora as possíveis direções para trabalhos futuros, objetivando aprimorar o desempenho do sistema de transporte coletivo por enjaulamento proposto.

### 6.1 Conclusões

Nesta dissertação foi abordado o problema de transporte coletivo por enjaulamento. Foi proposto um algoritmo que permite o transporte coletivo por enjaulamento de um objeto cilíndrico. O método desenvolvido nessa pesquisa baseia-se na utilização de máquinas de estados e controladores PID, a fim de controlar e manter o enjaulamento do objeto ao longo de toda a trajetória de transporte. Os resultados obtidos demonstram que o método proposto é capaz de manter o enjaulamento do objeto por toda a trajetória pela qual o mesmo deve ser transportado.

O algoritmo desenvolvido é aplicável a objetos com diversos formatos, desde que seja construído um modelo matemático, semelhante ao realizado neste projeto, para o formato específico desejado. Em outras palavras, é necessário estabelecer os parâmetros necessários para o enjaulamento do objeto desejado (ângulos e distâncias) de forma a permitir o equacionamento do posicionamento inicial dos robôs em torno de tal objeto, enjaulando-o. Identificando esses parâmetros críticos (variáveis de processo), que devem ser controlados, durante o transporte, para garantir o enjaulamento, juntamente com a equação de posicionamento inicial dos robôs, torna-se possível, mediante pequenas adap-

tações no algoritmo, utilizá-lo para o transporte coletivo por enjaulamento de objetos não cilíndricos. O algoritmo possui etapas, além do transporte em si, que o complementam, são elas: a procura do objeto na arena pelo enxame de robôs, o recrutamento do enxame para a região de detecção do objeto, o posicionamento inicial dos robôs em torno do objeto, enjaulando-o e o transporte propriamente dito.

Primeiramente, realiza-se a inicialização da arena, a qual precede a etapa de busca. Durante tal inicialização, pontos de referência são introduzidos ao longo da arena. Esses pontos de referência desempenham um papel crucial na fase de busca, uma vez que a locomoção dos robôs nessa fase é altamente dependente deles e da função de vizinhança que os relaciona. Além de incorporar os pontos de referência na arena, tal inicialização também estabelece aspectos cruciais às etapas subsequentes. Isso inclui a definição da função de vizinhança entre os pontos de referência, representada por  $\eta(\lambda_i)$ , bem como a determinação das posições iniciais tanto dos robôs quanto do objeto na arena. Adicionalmente, essa etapa estabelece a ordem de posicionamento inicial dos robôs em torno do objeto por meio da matriz  $\mu_i$ . Isso significa que, para cada robô  $i$ , uma ordem específica de posicionamento inicial é definida, e caso o robô  $i$  torne-se o robô recrutador, ao fim da etapa de busca, tal ordem será utilizada para a definição do posicionamento inicial dos robôs em torno do objeto.

A etapa de busca ocorre de maneira randômica. Inicialmente, os robôs escolhem de forma aleatória um ponto de referência na arena e se movem em direção a ele. Após alcançar esse ponto, eles escolhem outro ponto de referência na vizinhança do ponto atual, com base na função de vizinhança predefinida entre os pontos de referência, denotada por  $\eta(\lambda_i)$ . Este ciclo se repete até que o objeto seja identificado. Quando o objeto é detectado, o primeiro robô a fazê-lo é designado como o robô recrutador  $r^*$ . Nesse ponto, a fase de busca é encerrada, dando lugar à etapa de recrutamento.

A etapa de recrutamento depende substancialmente da comunicação entre os robôs. Inicialmente, o robô designado como recrutador envia mensagens para todo o enxame, fornecendo informações sobre a localização do centro de massa do objeto na arena, bem como a trajetória que percorreu desde sua posição inicial na arena até o ponto de detecção do objeto. Essa trajetória é referida como  $T_r^*$  e é posteriormente utilizada na etapa de transporte, como o caminho a ser percorrido pelo centro de massa do objeto. No entanto, vale ressaltar que essa trajetória é seguida de forma inversa, ou seja, transportando o

objeto desde o ponto de detecção até a posição inicial do robô recrutador na arena. A escolha dessa trajetória se baseia no fato de que o robô recrutador é o primeiro a detectar o objeto, tornando essa rota a mais curta entre a posição inicial dos robôs na arena e a posição inicial do objeto na arena. Quando os demais robôs do enxame recebem as mensagens do robô recrutador, eles se aproximam do centro de massa do objeto até alcançarem uma distância de aproximação  $\Delta$ . Alternativamente, eles aguardam serem chamados pelo robô recrutador para ocuparem suas posições iniciais de enjaulamento. O robô recrutador convoca os demais robôs de forma sequencial, seguindo a ordem preestabelecida na matriz  $\mu_{r^*}$ . Ao término dessa etapa, todos os robôs estão dispostos ao redor do objeto, enjaulando-o. Dessa forma, ao final dessa etapa, o processo de enjaulamento do objeto é concluído.

A etapa de transporte é inicializada assim que o posicionamento inicial dos robôs em torno do objeto, enjaulamento, é concluído. Inicialmente, o enxame aguarda a confirmação de que todos os robôs estão corretamente posicionados, em seguida, tais robôs orientam-se em direção ao primeiro ponto de referência  $\lambda$  para o qual devem transportar o objeto, onde  $\lambda \in T_{r^*}$ . Definido o ponto de referência, o enxame orienta-se em direção ao mesmo, e em seguida, os robôs movimentam-se, simultaneamente, na direção de tal ponto de referência, transportando o objeto. Tal movimentação ocorre até que o enxame assuma que o centro de massa do objeto atingiu o ponto de referência desejado, dentro de uma pequena faixa de tolerância preestabelecida. Ao atingir o ponto de referência desejado, o enxame verifica se chegou ao fim do trajeto a ser percorrido, se sim, o transporte é finalizado, do contrário, tais robôs orientam-se em direção ao próximo ponto de referência pertencente ao trajeto e movimentam-se, simultaneamente, em direção ao mesmo. Tal processo se repete até que todo o trajeto  $T_{r^*}$  seja percorrido, concluindo assim o transporte do objeto.

Escolheu-se a plataforma de simulações CoppeliaSim (COPPELIA ROBOTICS, 2023) e utilizou-se robôs do tipo Khepera-III (K-TEAM, 2023). A escolha desses robôs se deve à disposição de seus sensores, o que lhes permite identificar o objeto de maneira eficaz, independentemente da direção pela qual se aproximam do mesmo.

A avaliação do desempenho das etapas de busca e recrutamento foi conduzida mediante a análise do tempo de execução dessas etapas em arenas com diferentes passos de discretização. Verificou-se que à medida que o passo de discretização diminuiu, a ve-

localidade dos robôs entre os pontos de referência também diminui. Isso ocorre devido à proximidade do robô em relação ao ponto de referência, o que resulta em uma redução na sua velocidade, a fim de ajustar sua trajetória em direção ao ponto desejado. Esse fenômeno afeta principalmente a etapa de busca, uma vez que a diminuição do passo de discretização da arena está diretamente relacionada ao aumento do tempo de execução dessa etapa. No entanto, a etapa de recrutamento não é afetada por esse fator, pois o tempo de execução dessa etapa depende da distância entre o enxame de robôs e o objeto no momento em que o mesmo é detectado.

O tempo de execução da fase de recrutamento é influenciado tanto pela ordem de posicionamento inicial preestabelecida ( $\mu_i$ ) quanto pelo número de robôs do enxame. A presença de mais robôs implica em um tempo maior para a alocação adequada de todos nas suas posições iniciais de enjaulamento. Além disso, a ordem preestabelecida para o posicionamento inicial também afeta o tempo médio desta etapa. Isso se deve ao fato de que, quando um robô é acionado para assumir sua posição inicial de enjaulamento, o tempo necessário depende da sua localização atual, da posição final a ser alcançada e da quantidade de robôs entre essas posições, os quais o robô precisará contornar, podendo resultar em uma redução ou aumento do tempo necessário.

A avaliação de desempenho da etapa de transporte foi realizada a partir de estudos de caso. No primeiro estudo de caso, avalia-se o impacto do número de robôs no desempenho da etapa de transporte. Observou-se que, independentemente da quantidade de robôs envolvida, em uma arena com a mesma configuração e nível de discretização, o erro percentual entre a trajetória efetivamente percorrida pelo objeto e a trajetória ideal é aproximadamente contante. Entretanto, quanto maior o número de robôs, maior o tempo de execução do transporte coletivo por enjaulamento. Isso se deve à influência direta do número de robôs na etapa de posicionamento inicial, uma vez que um maior contingente de robôs demanda mais tempo para que o enxame se posicione adequadamente ao redor do objeto, enjaulando-o. Observou-se também que o aumento na quantidade de robôs pertencentes ao enxame não resulta em redução do tempo de execução da etapa de transporte. Tal fato ocorre pois os robôs operam de forma constante em suas velocidades máximas.

No segundo estudo de caso, foi analisado o efeito das dimensões do objeto no transporte coletivo por enjaulamento. Neste cenário, empregou-se um enxame composto

por oito robôs e objetos de três distintas dimensões. Assim como no estudo anterior, constatou-se que a o erro percentual da trajetória percorrida pelo centro de massa do objeto, em relação à trajetória ideal, permaneceu aproximadamente contante, independentemente das dimensões do objeto. Além disso, observou-se que o aumento das dimensões do objeto não resultou em aumento no tempo de execução da etapa de transporte. Pois, assim como constatado no estudo de caso anterior, os robôs atingem rapidamente suas velocidades máximas e as mantêm durante todo o transporte. Ou seja, os robôs mantêm uma velocidade constante, independentemente da massa do objeto.

No terceiro estudo de caso, investigou-se o impacto da discretização da arena. Nesse cenário, a arena possuía as mesmas dimensões dos casos anteriores, ou seja,  $5m \times 5m$ , no entanto, utilizou-se dois passos de discretização:  $0,5m$  e  $1,0m$ . Ao analisar os resultados, constatou-se que, da mesma forma que nos casos anteriores, o erro percentual entre a trajetória ideal e a trajetória efetivamente percorrida pelo centro de massa do objeto, é aproximadamente o mesmo independentemente do número de robôs no enxame ou das dimensões do objeto transportado. No entanto, esse estudo revelou que quanto menor o passo de discretização da arena, mais pontos de referência compõem o trajeto ao qual o objeto precisa percorrer, resultando em mais erros locais e conseqüentemente, pelo fato do erro de percurso ser acumulativo, um aumento no erro da trajetória. Em outras palavras, pode-se concluir que quanto menor o passo de discretização da arena, maior será o erro entre a trajetória ideal e a trajetória percorrida pelo objeto.

## 6.2 Trabalhos Futuros

A análise dos resultados demonstra que a abordagem proposta para a etapa de busca não assegura que o primeiro robô a detectar o objeto tenha seguido o caminho mais curto desde sua posição inicial e a localização de detecção do objeto. Como esse caminho é percorrido em sentido inverso durante a etapa de transporte, é necessário otimizá-lo. Portanto, a exploração de algoritmos de otimização de trajetória é necessária para assegurar um trajeto otimizado.

A implementação da estratégia proposta para objetos com variados formatos é parte do desenvolvimento contínuo desse projeto, viabilizando a utilização dessa estratégia no transporte de objetos de formatos diversos.

A adição de obstáculos na arena também faz parte do desenvolvimento natural desse projeto. Isso permitirá avaliar como o sistema lida com interferências e obstáculos no ambiente de transporte.

A estratégia proposta também pode ser aplicada em arenas não planares, como por exemplo em ambientes fluídos ou aéreos. Porém, nesses casos, será necessária a substituição dos robôs utilizados por robôs capazes de se locomover no meio específico.

Além disso, uma direção promissora para futuras pesquisas envolve a busca, recrutamento e transporte de diversos objetos por um ou múltiplos enxames. Essa proposta exploraria a capacidade do sistema em lidar com tarefas mais complexas e diversificadas, o que poderia ter implicações importantes em cenários do mundo real.

# REFERÊNCIAS

- ANG, K. H.; CHONG, G.; LI, Y. Pid control system analysis, design, and technology. *IEEE transactions on control systems technology*, IEEE, v. 13, n. 4, p. 559–576, 2005.
- BECKER, A.; ERTEL, C.; MCLURKIN, J. Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots. In: *IEEE. 2014 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2014. p. 2825–2830.
- BERMAN, S. et al. Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems. *Proceedings of the IEEE*, v. 99, n. 9, p. 1470–1481, 2011.
- BRAITENBERG, V. *Vehicles: Experiments in synthetic psychology*. [S.l.]: MIT press, 1986.
- CHEN, J. et al. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, IEEE, v. 31, p. 307–321, 2015.
- CHEN, J. et al. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, IEEE, v. 31, n. 2, p. 307–321, 2015.
- COPPELIA ROBOTICS. *CoppeliaSim: Robotics Simulation Platform*. 2023. <https://www.coppeliarobotics.com>. Acesso em: 18 de agosto de 2023.
- DAI, Y. et al. Symmetric caging formation for convex polygonal object transportation by multiple mobile robots based on fuzzy sliding mode control. *ISA transactions*, Elsevier, v. 60, p. 321–332, 2016.
- DAVIDSON, C.; BLAKE, A. Caging planar objects with a three-finger one-parameter gripper. v. 3, p. 2722–2727, 1998.



- FARRUGIA, J. L.; FABRI, S. G. Swarm robotics for object transportation. p. 353–358, 2018.
- FERREIRA, G. B. et al. Transporte cooperativo de objeto por multi-robôs. Universidade do Estado do Rio de Janeiro, 2022.
- FINK, J.; HSIEH, M. A.; KUMAR, V. *Multi-robot manipulation via caging in environments with obstacles*. 2008. 1471–1476 p.
- GROSS, R.; DORIGO, M. Towards group transport by swarms of robots. *International Journal of Bio-Inspired Computation*, Inderscience Publishers, v. 1, n. 1-2, p. 1–13, 2009.
- K-TEAM. *Khepera-III: User Manual*. 2023. <https://ftp.k-team.com/KheperaIII/UserManual/Kh3.Robot.UserManual.pdf>. Acesso em: 18 de agosto de 2023.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Proceedings of ICNN'95-international conference on neural networks*. [S.l.], 1995. v. 4, p. 1942–1948.
- LAMBRINOS, D.; SCHEIER, C. Extended braitenberg architectures. *Technical Report, AI Lab, No. IFI-AI-95.10*, Citeseer, 1995.
- LEAO, W. *Aplicando Controle On-off baseado em uma Maquina de Estados Finita*. 2020. YouTube, 06 de Julho de 2020. Disponível em: <[https://www.youtube.com/watch?v=eyTORTjPQcA&list=PL1WrY7PmiW\\_iwesX41-rS6ddHlvbrpe00&index=22](https://www.youtube.com/watch?v=eyTORTjPQcA&list=PL1WrY7PmiW_iwesX41-rS6ddHlvbrpe00&index=22)>. Acesso em: 20 de junho de 2023.
- LEAO, W. *Controle PID aplicado em robo movel de rodas (parte 2)*. 2020. YouTube, 06 de Julho de 2020. Disponível em: <[https://www.youtube.com/watch?v=PSt96KnCNxc&list=PL1WrY7PmiW\\_iwesX41-rS6ddHlvbrpe00&index=34](https://www.youtube.com/watch?v=PSt96KnCNxc&list=PL1WrY7PmiW_iwesX41-rS6ddHlvbrpe00&index=34)>. Acesso em: 20 de junho de 2023.
- LEAO, W. *Controle PID duplo aplicado em robo movel de rodas*. 2020. YouTube, 07 de Julho de 2020. Disponível em: <[https://www.youtube.com/watch?v=IlhhvFpWJmE&list=PL1WrY7PmiW\\_iwesX41-rS6ddHlvbrpe00&index=35](https://www.youtube.com/watch?v=IlhhvFpWJmE&list=PL1WrY7PmiW_iwesX41-rS6ddHlvbrpe00&index=35)>. Acesso em: 20 de junho de 2023.
- LIPTAK, B. G. *Instrument Engineers' Handbook, Volume One: Process Measurement and Analysis*. [S.l.]: CRC press, 2003.

- PEREIRA, G. A.; CAMPOS, M. F.; KUMAR, V. Decentralized algorithms for multi-robot manipulation via caging. *The International Journal of Robotics Research*, SAGE Publications, v. 23, n. 7-8, p. 783–795, 2004.
- RIMON, E.; BLAKE, A. Caging 2d bodies by 1-parameter two-fingered gripping systems. v. 2, p. 1458–1464, 1996.
- SASAKI, J. et al. Cooperating grasping of a large object by multiple mobile robots. In: IEEE. *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. Nagoya, Japan, 1995. p. 1205–1210.
- SPIEGEL, M. R. *Laplace transforms*. [S.l.]: McGraw-Hill New York, 1965.
- STILWELL, D. J.; BAY, J. S. Toward the development of a material transport system using swarms of ant-like robots. p. 766–771, 1993.
- SUDSANG, A.; PONCE, J. On grasping and manipulating polygonal objects with disc-shaped robots in the plane. v. 3, p. 2740–2746, 1998.
- VARDHARAJAN, V. S.; SOMA, K.; BELTRAME, G. Collective transport via sequential caging. In: MATSUNO, F.; AZUMA, S.-i.; YAMAMOTO, M. (Ed.). *Distributed Autonomous Robotic Systems*. Cham: Springer International Publishing, 2022. p. 349–362. ISBN 978-3-030-92790-5.
- WANG, Z.; KUMAR, V. Object closure and manipulation by multiple cooperating mobile robots. v. 1, p. 394–399, 2002.
- ZHONG, J. Pid controller tuning: A short tutorial. *Mechanical Engineering, Purdue University*, p. 1–10, 2006.