MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

JÚLIO CÉSAR SANTANA DA ROSA FILHO

MÁQUINAS DE RECOMPENSA NO APRENDIZADO POR REFORÇO MULTIAGENTE: UM FRAMEWORK PARA CONTROLE ESCALÁVEL DE ENXAME DE VANTS

RIO DE JANEIRO
2025

RESUMO

Este trabalho propõe um novo framework que integra Máquinas de Recompensa (Reward Machines - RMs) com algoritmos de Aprendizado por Reforço Multiagente (Multi-Agent Reinforcement Learning - MARL) denomidado RM-MARL, visando ao desenvolvimento de agentes autônomos para enxames de Veículos Aéros Não Tripulados (VANTs). A utilização das Máquinas de Recompensa possibilita a exposição da estrutura interna da função recompensa durante o treinamento dos agentes, permitindo a convergência para políticas ótimas de maneira mais eficiente e reduzindo o tempo de treinamento. Ao permitir a especificação estruturada e hierárquica de uma função recompensa por meio das RMs, o framework proposto busca capacitar agentes descentralizados a aprenderem comportamentos complexos compostos por múltiplas subtarefas dentro de uma mesma missão. O resultado esperado é um framework reutilizável para o desenvolvimento de enxames autônomos voltados a missões complexas.

Palavras-chave: Enxame de VANTs. Aprendizado por Reforço Multiagente. Framework RM-MARL. Máquinas de Recompensa. Controle Descentralizado.

ABSTRACT

This work proposes a novel framework that integrates Reward Machines (RMs) with Multi-Agent Reinforcement Learning (MARL) algorithms, aiming to develop autonomous agents for UAV swarms. The use of Reward Machines allows the internal structure of the reward function to be explicitly represented during agent training, enabling more efficient convergence toward optimal policies and reducing training time. By allowing a structured and hierarchical specification of the reward function through RMs, the proposed framework seeks to enable decentralized agents to learn complex behaviors composed of multiple subtasks within a single mission. The expected outcome is a reusable framework for the development of autonomous swarms capable of handling complex missions.

Keywords: UAV Swarm. MARL. RM-MARL Framework. Reward Machines. Decentralized Control.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Veículos Aéreos Não Tripulados (VANTs) ou *Unmanned Aerial Vehicles* (UAVs), comumente conhecidos como drones, revolucionaram uma ampla gama de aplicações, desde monitoramento e missões de busca e resgate até agricultura e logística. Sua capacidade de operar autonomamente em ambientes diversos tornou-os indispensáveis em domínios civis e militares. Ao longo dos anos, os sistemas de VANTs evoluíram de dispositivos simples controlados remotamente para enxames altamente inteligentes e cooperativos, capazes de tomar decisões complexas. No setor militar, os enxames de VANTs possibilitam missões de reconhecimento, vigilância e ataque cooperativo, reduzindo riscos para tropas e aumentando a eficácia operacional. Contudo para habilitar o controle coordenado das ações dos agentes VANTs dentro do enxame, torna-se necessário o desenvolvimento de técnicas mais robustas capazes de lidar com ambientes dinâmicos e complexos que as missões exigem.

1.1 Contextualização

O controle de UAVs tem evoluído significativamente ao longo do tempo. No trabalho (**??**) fornece uma visão abrangente da evolução dos métodos de controle para sistemas de VANTs. O estudo destaca a transição de controladores PID clássicos para abordagens modernas baseadas em IA, incluindo aprendizado por reforço (RL). Com a crescente capacidade de processamento computacional, os UAVs passaram a incorporar técnicas de aprendizado de máquina e aprendizado por reforço profundo (**Deep Reinforcement Learning** - DRL), permitindo controle adaptativo e maior autonomia em operações de enxame.

Complementando essa evolução, estudos como (**????**), abordam técnicas de Aprendizado por Reforço Multiagente (*Multi-Agente Reinforcement Learning* - MARL) explorando sua capacidade de aprimorar a coordenação em enxames ao abordar desafios como: não estacionariedade e observabilidade parcial. Técnicas de MARL melhoram a escalabilidade e a autonomia em enxames de VANTs ao permitir tomada de decisão descentralizada. Contudo, o estudo ressalta uma lacuna persistente entre simulação e implantação prática, onde variabilidade ambiental e limitações de hardware introduzem desafios não abordados em ambientes simulados (**??**).

1.2 Motivação e Justificativa

Apesar dos avanços no poder computacional dos hardwares e no desenvolvimento dos algoritmos de IA, os frameworks existentes de MARL ainda enfrentam desafios significativos relacionados à não-estacionariedade e escalabilidade. Essas dificuldades tornam-se especialmente críticas à medida que o tamanho do enxame aumenta ou quando os agentes operam em espaços de estados de alta dimensão. Tais limitações são particularmente sensíveis em operações militares, onde o enxame deve executar múltiplas tarefas simultaneamente, como rastrear alvos móveis, evitar colisões com obstáculos, otimizar o consumo de energia e manter a formação da frota. Esses cenários apresentam um alto grau de complexidade, pois exigem um processo de tomada de decisão hierárquico e descentralizado por parte dos agentes, tornando essencial o desenvolvimento de estratégias de controle mais robustas e adaptativas.

Este trabalho propõe a integração de Máquinas de Recompensa (*Reward Machines* - RMs) com algoritmos de aprendizado multiagente (MARL), visando o projeto de funções recompensas estruturadas e hierárquicas para enxames de VANTs. Ao decompor missões complexas (por exemplo, rastreamento de alvos) em subtarefas modulares, o framework alinha agentes descentralizados aos objetivos globais, ao mesmo tempo em que enfrenta desafios como escalabilidade e observabilidade parcial. Essa abordagem se baseia em inovações recentes em aprendizado por reforço apresentados em (**??**), ao mesmo tempo em que aborda desafios reais enfrentados por enxames em aplicações militares e logísticas.

Esta proposta enquadra-se na área de Ciência da Computação em conformidade com a Necessidade de Conhecimento 01M2024 da Portaria Nº 007 do Departamento de Ciência e Tecnologia do Exército Brasileiro, em 27 de Janeiro de 2023.

1.3 Objetivos da Proposta

1.3.1 Objetivo Geral

Aprimorar o processo de treinamento em algoritmos de aprendizado por reforço multiagente (MARL), reduzindo o tempo de convergência para políticas ótimas por meio da exposição da estrutura interna da função recompensa, obtida a partir da modelagem da missão do enxame de VANTs com o uso de máquinas de recompensa.

1.3.2 Objetivos específicos

1. Integrar máquinas de recompensa nos algoritmos MARL de execução descentralizada.
2. Investigar como a utilização de RMs afeta a escalabilidade do enxame.

3. Desenvolver framework para o treinamento de agentes autônomos VANT utilizando ambientes de simulação 3D e bibliotecas de desenvolvimento consolidadas na área.

4. Validação dos agentes treinados por meio execução de missões de rastreamento de alvo em ambientes simulados e reais.

1.4 Contribuições Esperadas

- Um novo framework RM-MARL que permite a decomposição de missões complexas em subtarefas possibilitando o treinamento de agentes para convergência de políticas satisfatórias.
- Validação empírica mostrando que o framework é capaz de executar missões complexas.
- Criação de um ambiente de simulação open-source para desenvolvimento de agentes autônomos para VANTs.
- Validação empírica que o framework consegue escalar para um número maior de agentes.

1.5 Estrutura da Proposta

Os próximos capítulos deste trabalho, estão estruturados da seguinte forma:

- **Capítulo 2 - Fundamentação Teórica**: Apresentação dos conceitos fundamentais para o desenvolvimento da proposta.
- **Capítulo 3 - Trabalhos Relacionados**: Apresentação dos trabalhos relacionado encontrados durante a revisão sistemática da literatura.
- **Capítulo 4 - A Proposta**: Descrição detalhada da proposta propriamente dita, questões de pesquisa, objetivos e contribuições esperadas.
- **Capítulo 5 - Plano de Ação**: descrição da metodologia adotada e atividades, resultados parciais, viabilidade da proposta e cronograma.
- **Capítulo 6 - Conclusão**: Conclusão e considerações da proposta.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Sistemas de VANTs

Esta seção fornece uma visão geral dos conceitos fundamentais necessários para compreender os sistemas de VANT, seus mecanismos de controle e os princípios de coordenação em enxames. Começamos detalhando os principais componentes e classificações dos VANTs, seguido por uma discussão sobre estratégias de controle e, por fim, as arquiteturas de enxames.

2.1.1 Principais componentes de um VANT

Um VANT (Veículo Aéreo Não Tripulado) consiste em diversos componentes essenciais que possibilitam sua operação autônoma. Estes incluem:

- **Estrutura do Drone:** O quadro (ou frame) de um drone é a estrutura física que serve como base para todos os seus componentes. Ele é responsável por suportar os motores, hélices, controladores de voo, baterias, sensores e demais módulos eletrônicos. Em geral, o frame é projetado para ser leve, rígido e resistente, a fim de garantir estabilidade e eficiência durante o voo.
- **Sistema de Propulsão:** Motores elétricos e hélices para UAVs de múltiplos rotores ou motores a combustão para drones de asa fixa de maior porte.
- **Controladora de Voo:** É o componente central do drone responsável por interpretar os dados dos sensores (como giroscópios, acelerômetros, GPS e magnetômetros) e enviar comandos precisos aos motores para estabilizar e controlar o voo.
- **Computador embarcado:** Sistema de computação embarcada que realiza o processamento de dados oriundos dos sensores, do controlador de voô e da estação solo. Geralmente neste componente que ocorre a execução dos algoritmos de visão computacional essenciais para a navegação do drone. Neste componente que as decisões de alto nível são tomadas e repassadas a controladora de voo.
- **Sensores:** Unidades de Medição Inercial (IMU), GPS, LiDAR, câmeras e barômetros, que auxiliam na navegação, estabilidade e percepção do ambiente.
- **Sistema de Comunicação:** Links de dados para telemetria, controle e coordenação entre os agentes.

2.1.2 Dinâmica de Voô de um Quadricoptero

Neste trabalho a estrutura adotada para o drone será a de 4 motores, denominado quadricoptero ou *quadcopter*. Este tipo de drone possui 6 (seis) graus de liberdade (DOF), sendo 3 rotacionais e 3 translacionais. A sua dinâmica pode ser modelada da seguinte forma: $[\omega_1, \omega_2, \omega_3, \omega_4]$ indicam a velocidade angular cada um dos motores. Os ângulos de euler denominados roll, pitch e yaw são indicados como $[\phi, \theta, \psi]$ respectivamente. Portanto o efeito aerodinâmico u que cada velocidade w_i tem na força de empuxo f e nos ângulos de rotação é definido por:

$$u_f = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad \text{[Empuxo Total]} \tag{2.1}$$

$$u_\phi = b \cdot L(\omega_4^2 - \omega_2^2) \quad \text{[Momento de Rolagem]} \tag{2.2}$$

$$u_\theta = b \cdot L(\omega_3^2 - \omega_1^2) \quad \text{[Momento de Arfagem]} \tag{2.3}$$

$$u_\psi = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad \text{[Momento de Guinada]} \tag{2.4}$$

$$m\ddot{\mathbf{p}} = -m\mathbf{g} + \mathbf{R} \cdot u_f \quad \text{[Dinâmica Translacional]} \tag{2.5}$$

Onde b é o coeficiente de empuxo, d é o coeficiente de arrasto, L distância entre o centro de massa e o motor e $\mathbf{R}$ é matriz de rotação transformada de coordenadas. Para o drone realizar uma manobra, o seu sistema de controle de voô deve ser capaz coordenar as velocidades dos rotores com o intuito de atingir o movimento (rotação e translação) esperado.

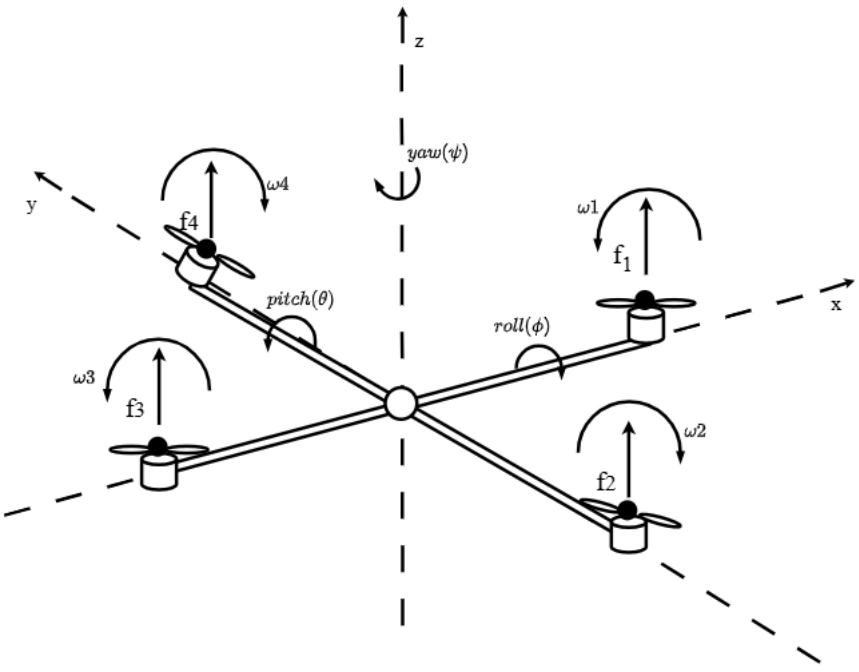

Figura 1 – Diagrama esquemático da dinâmica de um VANT com 4 motores

2.1.3 Sistema de Controle de Voo

Um sistema de controle de voo de UAV é o mecanismo central que mantém e ajusta a orientação de um drone durante o voo, gerenciando o yaw, pitch e roll. Esse sistema de controle depende do feedback dos sensores para comparar continuamente a atitude real com o estado desejado, efetuando correções rápidas para garantir a estabilidade. Dentre as principais estratégias existente para o controle de voo, abrangem técnicas clássicas até métodos avançados baseados em IA.

2.1.3.1 Controle PID

Os controladores PID nos sistemas de voo utilizam da teoria clássica de controle, especialmente na análise no domínio da frequência das funções de transferência. Ao representar o comportamento dinâmico do UAV por meio de uma função de transferência, é possível estudar como o sistema responde a diferentes frequências. O controlador PID ajusta os ganhos proporcional, integral e derivativo para modelar a resposta em frequência, aprimorando a estabilidade e o desempenho do sistema. Esse processo de ajuste garante que a atitude do drone — seu roll, pitch e yaw — permaneça responsiva aos comandos de controle, minimizando ultrapassagens e oscilações. Em essência, o projeto do controlador PID aproveita os insights matemáticos obtidos a partir da resposta em frequência do sistema, possibilitando uma estratégia de controle precisa e robusta para manter um desempenho de voo ideal. O sinal de controle de saída é matematicamente definido no domínio do tempo como:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t) \tag{2.6}$$

onde:

- $u(t)$ é o comando de controle enviado aos atuadores.
- $e(t)$ é o erro entre o estado desejado e o real.
- K_p, K_i e K_d são os ganhos proporcional, integral e derivativo, respectivamente.

Aplicando a transformada de Laplace o sinal de controle no domínio da frequência torna-se:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{2.7}$$

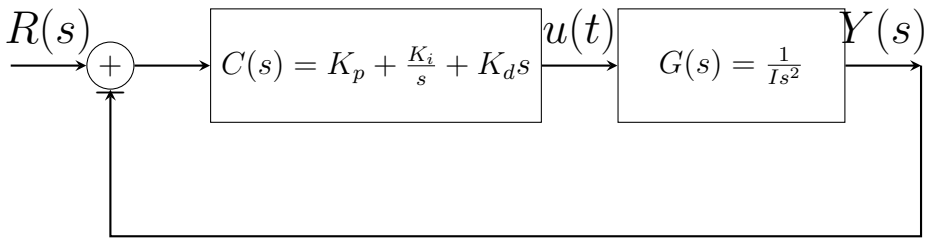

Figura 2 – Malha de controle fechada quadrotor

Fonte: Elaborado pelo autor.

Onde $G(s)$ representa a função transferência de malha aberta que modela a dinâmica do drone com 4 rotores, sendo I momento de inércia sob os respectivos eixos. $R(s)$ indica a entrada de referência, ou seja, os ângulos roll,pitch e yaw desejados. $Y(s)$ representa a saída obtida a partir dos sensores IMU do drone, que são retroalimentados para computação do erro, ajustando a entrada do controlador.

2.2 Enxame de VANTs

2.2.1 Definição

Um **enxame de VANTs (Veículos Aéreos Não Tripulados)** refere-se a um grupo coordenado de drones autônomos ou semiautônomos que colaboram de forma distribuída para atingir um objetivo comum. Esses sistemas inspiram-se frequentemente em comportamentos coletivos observados na natureza, como colônias de insetos sociais, e destacam-se por sua capacidade de operar de maneira cooperativa, robusta e eficiente (**????**).

Entre as propriedades fundamentais que caracterizam um enxame de VANTs, destacam-se:

- **Arquitetura**: Esta propriedade define o local no qual as decisões do exame são processadas. As principais arquiteturas são: centralizada, descentralizada e híbrida (**??**).
- **Escalabilidade**: Diz respeito à capacidade do sistema de manter sua eficiência à medida que o número de agentes no enxame aumenta. Um sistema escalável é capaz de coordenar ações com dezenas ou centenas de UAVs sem comprometer o desempenho geral (**??**).
- **Adaptabilidade**: Representa a habilidade do enxame de reagir a mudanças no ambiente, como a presença de obstáculos, a movimentação de alvos ou a introdução

de novas missões. Essa propriedade é essencial para operações em ambientes dinâmicos e não estruturados (**??**).

- **Redundância**: Refere-se à tolerância a falhas, possibilitada pela distribuição de funções entre múltiplos agentes. Caso um ou mais drones falhem, os demais são capazes de compensar a perda e manter a continuidade da missão.

2.2.2 Arquiteturas de Enxames

As arquiteturas de controle de enxames de VANTs podem ser classificadas em três categorias principais: centralizada, descentralizada e híbrida.

Na **arquitetura centralizada**, ilustrado na figura 3, um nó controlador central — comumente denominado estação solo — gerencia todos os VANTs, sendo responsável por enviar comandos e receber as informações coletadas pelos drones. Essa abordagem apresenta vantagens como a coordenação simplificada e a possibilidade de otimizações globais. No entanto, sofre com limitações importantes, como a existência de um ponto único de falha e baixa escalabilidade, tornando-se menos adequada para operações em larga escala ou em ambientes com conectividade instável.

A **arquitetura descentralizada**, ilustrado na figura 4, por sua vez, distribui a responsabilidade de tomada de decisão entre os próprios drones, que atuam com base em suas observações locais e comunicação entre pares. Para isso, cada drone deve possuir certo grau de autonomia, seja por meio de rotinas pré-programadas ou de sistemas inteligentes baseados em inteligência artificial. Essa abordagem é naturalmente mais robusta a falhas e altamente escalável. Em contrapartida, exige maior capacidade computacional embarcada nos drones e apresenta desafios adicionais de coordenação, o que pode levar a políticas subótimas em comparação com soluções centralizadas.

Por fim, a **arquitetura híbrida**, ilustrado pela figura 5, busca combinar o melhor dos dois mundos. Nela, um servidor central realiza o planejamento de alto nível, como a atribuição de tarefas, enquanto os drones executam essas tarefas de forma descentralizada, negociando trajetórias localmente e tomando decisões com base em suas próprias observações. Esse modelo permite maior flexibilidade e resiliência, ao mesmo tempo em que mantém um certo grau de controle global sobre o comportamento do enxame.

2.2.3 Aplicações e Comportamentos de Enxames de UAVs

Os enxames de UAVs têm ganhado destaque em diversas áreas devido à sua capacidade de realizar missões cooperativas de forma eficiente, robusta e escalável. Entre as principais aplicações, destacam-se as missões de vigilância e reconhecimento, como o rastreamento de alvos e patrulhamento de fronteiras, onde múltiplos drones podem cobrir grandes áreas simultaneamente, aumentando a efetividade da missão. Em situações de

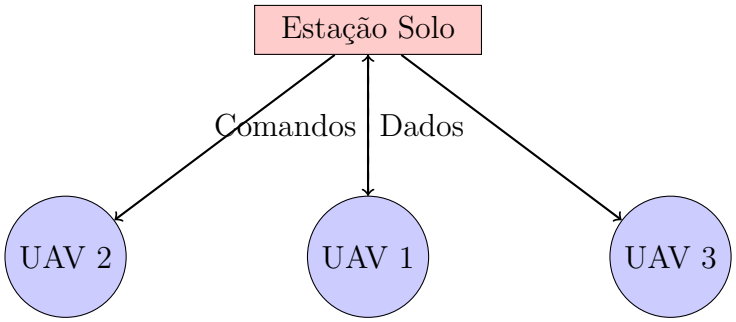

Figura 3 – Arquitetura Centralizada.

Fonte: Elaborado pelo autor.

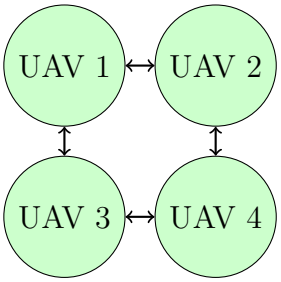

Figura 4 – Arquitetura Descentralizada.

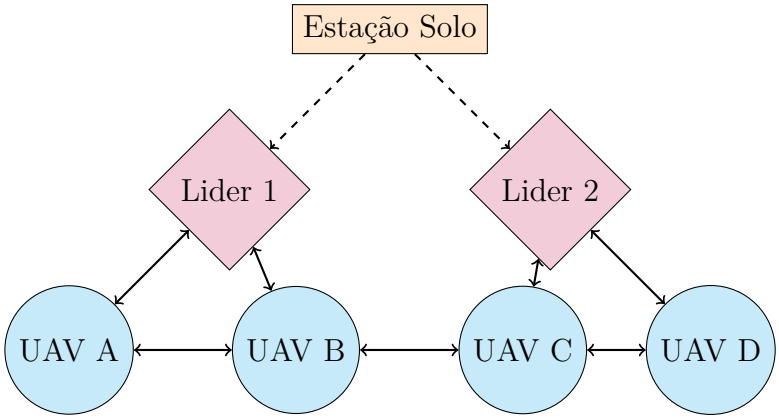

Figura 5 – Arquitetura Híbrida.

Fonte: Elaborado pelo autor.

resposta a desastres, os enxames são empregados em tarefas de busca e resgate e avaliação de danos, oferecendo uma solução ágil e segura para operar em ambientes perigosos ou de difícil acesso.

Na agricultura de precisão, esses sistemas são utilizados para o monitoramento em larga escala de plantações, identificação de áreas afetadas por pragas ou falta de irrigação, bem como pulverização seletiva de pesticidas. Enxames também podem ser empregados como redes de comunicação móveis, oferecendo cobertura temporária de redes 5G ou 6G

em áreas remotas ou durante eventos de emergência. Em operações militares, os UAVs em enxame são aplicados em ações coordenadas, como ataques simultâneos, guerra eletrônica, reconhecimento e bloqueio de sinais, com alto grau de adaptabilidade e redundância frente a ameaças.

A execução eficiente dessas aplicações depende de comportamentos cooperativos e distribuídos, que caracterizam os sistemas de enxame. Um dos comportamentos fundamentais é a formação de voo, onde os UAVs mantêm padrões geométricos estáveis — como formações em "V" ou em grade — permitindo maior organização e controle da missão. Outro comportamento essencial é a alocação de tarefas, que envolve a distribuição dinâmica de subtarefas entre os membros do enxame, frequentemente baseada em mecanismos de leilão, heurísticas distribuídas ou aprendizado por reforço.

O planejamento colaborativo de trajetórias e desvio de obstáculos garante que os UAVs naveguem de forma eficiente, evitando colisões e otimizando rotas em ambientes dinâmicos. A tomada de decisão coletiva é outro elemento-chave, permitindo que os drones cheguem a consensos em tempo real sobre alvos prioritários ou mudanças estratégicas na missão. Por fim, a autorreconfiguração é uma capacidade crítica de resiliência, na qual o enxame reorganiza sua estrutura e ajusta seu comportamento automaticamente em resposta à falha ou perda de um ou mais UAVs, assegurando a continuidade da operação.

Dessa forma, os enxames de UAVs oferecem um paradigma promissor para a automação de missões complexas, combinando aplicações práticas variadas com uma base de comportamentos coletivos sofisticados.

2.2.4 Principais Desafios no Controle de Enxames VANTs

O controle de enxames de UAVs envolve uma série de desafios técnicos e operacionais que precisam ser superados para garantir missões bem-sucedidas, especialmente em ambientes reais e dinâmicos. Um dos principais obstáculos está na complexidade da coordenação entre múltiplos agentes. Devido à observabilidade parcial, cada UAV frequentemente só tem acesso a informações limitadas sobre o ambiente e sobre os demais membros do enxame, o que restringe sua consciência situacional e dificulta a tomada de decisões cooperativas eficazes. Essa limitação é agravada pela não estacionariedade do ambiente, típica de cenários em que múltiplos agentes aprendem ou tomam decisões simultaneamente, interferindo uns nos outros.

Outro desafio crítico está relacionado à escalabilidade do sistema. À medida que o número de drones no enxame cresce, a comunicação entre os agentes tende a gerar uma sobrecarga significativa, cujo custo computacional e de largura de banda cresce quadraticamente com o tamanho do grupo. Além disso, a coordenação em espaços de ação conjuntos de alta dimensionalidade leva à chamada maldição da dimensionalidade,

dificultando o planejamento e o aprendizado de políticas eficazes.

A atuação em ambientes dinâmicos e incertos impõe a necessidade de replanejamento constante. Obstáculos móveis, alterações nas condições ambientais e movimentação de alvos exigem que os UAVs adaptem suas trajetórias e estratégias de forma reativa e em tempo real. Nessas situações, incertezas associadas a medições de sensores ou limitações na precisão dos atuadores podem comprometer a segurança e a eficácia da missão.

As restrições físicas dos drones também impõem limitações operacionais significativas. A autonomia de voo é geralmente limitada pela capacidade das baterias, o que restringe o tempo de operação contínua. Além disso, o poder computacional embarcado costuma ser reduzido, exigindo algoritmos eficientes e leves para processamento local. Isso gera um constante trade-off entre exploração (busca por novas informações) e exploração (execução de ações já conhecidas como eficazes).

Por fim, aspectos relacionados à segurança e resiliência não podem ser negligenciados. UAVs estão sujeitos a vulnerabilidades como interferência eletromagnética, ataques de spoofing e ciberataques que podem comprometer o controle, a navegação ou a integridade dos dados. Em ambientes operacionais compartilhados, a presença de agentes adversários ou aeronaves desconhecidas representa uma ameaça adicional que deve ser considerada nos mecanismos de controle e decisão coletiva.

Esses desafios exigem abordagens inovadoras em áreas como aprendizado por reforço multiagente, controle distribuído, comunicações seguras e design de arquiteturas robustas para garantir que os enxames de UAVs possam operar de forma autônoma, eficiente e segura em cenários cada vez mais complexos.

2.3 Fundamentos do Aprendizado por Reforço

O Aprendizado por Reforço (Reinforcement Learning - RL) é um paradigma de aprendizado de máquina inspirado pela psicologia comportamental, onde um agente aprende a tomar decisões interagindo com um ambiente para maximizar recompensas cumulativas. Diferentemente do aprendizado supervisionado, no qual os modelos aprendem com dados rotulados, o RL depende de interações por tentativa e erro para descobrir estratégias ótimas. Segundo (**??**), os conceitos fundamentais no RL são descritos como segue.

2.3.1 Agente

O agente é o tomador de decisões dentro do framework de RL. Ele interage com o ambiente executando ações e aprende uma política (*policy*)—um mapeamento entre estados e ações que determina a tomada de decisões do agente. A política é o núcleo

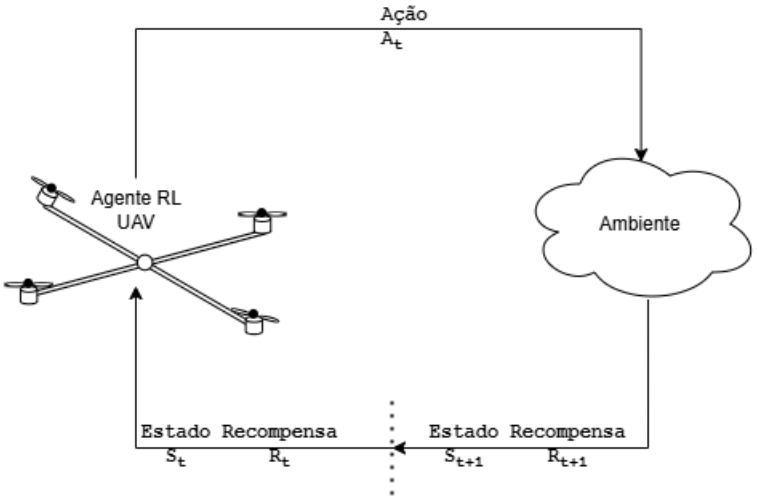

Figura 6 – Exemplo de diagrama esquemático sistema RL com agente VANT

Fonte: Autor.

central do agente, de modo esta determina o comportamento do agente. Portanto a fase de treinamento do agente, consiste em encontrar uma política que maximize a recompensa acumulada (retorno) esperada ao longo do período de atuação do agente. A figura 6 ilustra o fluxo de interação entre o agente e o ambiente.

2.3.2 Ambiente

O ambiente representa tudo fora do agente com o qual ele interage. Ele fornece ao agente um estado ou observação, que é uma representação da situação atual que o agente é capaz de perceber, e responde às ações do agente mudando para um novo estado e oferecendo feedback em forma de recompensa.

Definição 1 (Framework RL)**.** *Formalmente, o framwork de RL é modelado como um Processo de Decisão de Markov (MDP), definido pela tupla* $\langle S, A, P, R \rangle$ *onde:*

- *(S): Conjunto denominado espaço de estados. Neste trabalho o espaço de estados são o conjunto de observações que o drone obtém do ambiente. Informações como: pose do drone no espaço, pontos de nuvem das leituras do sensor LiDAR, imagens das câmeras do drone, GPS (latitude e longitude).*
- *(A): Conjunto denominado espaço de ações.*
- $P(s_{t+1}|s_t, a_t)$*: Uma função de transição , que define a probabilidade de transição para o estado* s_{t+1} *ao realizar a ação* a_t *no estado* s_t*.*
- $R(s, a)$*: Função de recompensa, que atribui um valor escalar às transições.*

2.3.3 Recompensa

A recompensa é um sinal de feedback escalar que quantifica o benefício imediato de uma ação tomada pelo agente. Este é o principal sinal que orienta o processo de aprendizado do agente. O objetivo do agente é maximizar a recompensa cumulativa (retorno) recebida ao longo do tempo. Isso envolve equilibrar recompensas imediatas e ganhos potenciais de longo prazo. O tipo de retorno mais utilizado é o **retorno esperado de horizonte infinito**, limitado o fator de desconto $\gamma \in (0, 1)$, definido como:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \tag{2.8}$$

onde $R(s_t, a_t) = r_t$ e $\tau = (s_0, a_0, s_1, a_1, ...)$ representa um histórico de sequência de estados e ações tomadas no ambiente.

2.3.4 Política

A política (π) define o comportamento do agente. É uma função ou distribuição de probabilidade que mapeia estados para ações. Políticas podem ser determinísticas ($a_t = \pi(s_t)$) ou estocásticas $\pi(a_t|s_t) \in [0, 1]$.

2.3.5 Problema central em RL

O objetivo principal do aprendizado por reforço, consiste então, em selecionar uma política π que maximize a função de retorno esperada. Para definir matematicamente esse problema é preciso estabelecer o conceito de trajetória, $\tau = (s_0, a_0, s_1, a_1, ...)$ que representa uma sequência de estados e ações tomadas no ambiente. Então a probabilidade de uma trajetória é definido como:

$$P(\tau|\pi) = \prod_{t=0}^{T} P(s_{t+1}|s_t, a_t)\pi(a_t, s_t) \tag{2.9}$$

O retorno esperado para uma política específica π, é denotado por $J(\pi)$ definido como:

$$J(\pi) = \int_{\tau} P(\tau|\pi)R(\tau)\, d\tau \tag{2.10}$$

O problema central de otimização no aprendizado por reforço é modelado matematicamente então como:

$$\pi^* = \arg\max_{\pi} J(\pi) \tag{2.11}$$

2.3.6 Funções de Valor

As funções de valor avaliam a qualidade de estados ou pares estado-ação sob uma política dada, sendo fundamentais para orientar o agente a estratégias melhores:

- Função de valor de estado $V^{\pi}(s) = E$: Retorno esperado ao começar no estado s e seguir a política π.
- Função de valor de ação ($Q^{\pi}(s, a)$): Retorno esperado ao tomar a ação a no estado s e seguir a política π.

2.3.7 Exploração versus Aprimoramento

Um desafio fundamental no RL é o equilíbrio entre exploração (testar novas ações para descobrir seus efeitos) e aprimoramento (Exploitation) (escolher ações já conhecidas que maximizam recompensas). Estratégias como ϵ-gananciosa (ϵ-greedy) e Bound de Confiança Superior (UCB, do inglês Upper Confidence Bound) tratam deste equilíbrio ao combinar a necessidade do agente de coletar informações e alcançar altas recompensas.

2.3.8 Métodos de Aprendizado

Os algoritmos de Aprendizado por Reforço (Reinforcement Learning, RL) podem ser classificados em três grandes categorias, cada uma com diferentes estratégias para lidar com a dinâmica do ambiente e o processo de tomada de decisão.

Os métodos **model-free** (sem modelo) não exigem conhecimento prévio das funções de transição e recompensa do ambiente. Em vez disso, eles aprendem diretamente a política ou a função de valor a partir das interações com o ambiente. Dentro desta categoria, destacam-se os métodos baseados em valores, como o *Q-Learning*, que busca estimar a função de valor de ação $Q(s, a)$ para selecionar ações que maximizem a recompensa acumulada. Uma extensão poderosa dessa abordagem é o uso de redes neurais profundas, resultando nos *Deep Q-Networks (DQNs)*, que permitem aplicar Q-Learning em ambientes com grandes espaços de estados, como aqueles representados por imagens ou sensores de alta dimensão.

Por outro lado, os métodos **model-based** (baseados em modelo) tentam construir uma representação explícita do ambiente, aprendendo suas dinâmicas — isto é, como os estados evoluem e quais recompensas são associadas às ações. Esses métodos permitem o uso de técnicas de planejamento, como simulações internas (*rollouts*), para prever consequências futuras antes de agir, o que pode acelerar o processo de aprendizado e melhorar a amostragem de dados. Contudo, são mais sensíveis a erros no modelo aprendido.

A terceira categoria compreende os **métodos de otimização de política**, que consistem em atualizar diretamente os parâmetros da política com base em gradientes estimados da função objetivo. Ao invés de depender da estimativa de funções de valor, esses métodos otimizam diretamente a probabilidade de selecionar boas ações. Entre os algoritmos mais representativos dessa abordagem estão o *Proximal Policy Optimization*

(PPO) e o *Trust Region Policy Optimization (TRPO)*, ambos amplamente utilizados por sua estabilidade e eficácia em tarefas com múltiplas etapas e ambientes contínuos.

Em sistemas multiagente, essas categorias se mantêm, mas a complexidade aumenta devido à não estacionariedade introduzida pela presença de múltiplos agentes aprendendo simultaneamente. A escolha do método adequado, portanto, deve considerar a natureza do ambiente, a escalabilidade e os objetivos do controle distribuído no sistema de enxame.

2.4 Aprendizado por Reforço Multiagente (MARL)

O Aprendizado por Reforço Multiagente (MARL) estende o aprendizado por reforço de agente único para ambientes onde múltiplos agentes aprendem a interagir e colaborar.

Definição 2 (Framework MARL)**.** *Formalmente, o MARL é modelado como um* ***Jogo de Markov*** *(??), definido pela tupla* $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}, \mathcal{P}, \{\mathcal{R}^i\}, \gamma \rangle$*, onde:*

- $\mathcal{N}$*: Conjunto de* n *agentes.*
- $\mathcal{S}$*: Espaço de estados compartilhado.*
- $\mathcal{A}^i$*: Espaço de ações do agente* i*.*
- $\mathcal{P}(s'|s, \mathbf{a})$*: Probabilidade de transição para o estado* s' *dada a ação conjunta* $\mathbf{a} = (a^1, \ldots, a^n)$*.*
- $\mathcal{R}^i(s, \mathbf{a})$*: Função de recompensa do agente* i*.*
- γ*: Fator de desconto.*

Diferentemente do RL de agente único, no MARL os agentes devem equilibrar recompensas individuais com objetivos coletivos, gerando desafios únicos:

- **Não Estacionariedade**: Políticas dos agentes mudam concorrentemente, violando a suposição de Markov.
- **Atribuição de Crédito**: Dificuldade em associar sucessos/falhas globais a ações individuais.
- **Observabilidade Parcial**: Agentes observam apenas estados locais $o^i \subset \mathcal{S}$.

2.4.1 Abordagens Algorítmicas em MARL

Os desafios de coordenação em enxames de UAVs exigem estratégias de *Multi-Agent Reinforcement Learning* (MARL) que equilibrem escalabilidade, eficiência e adaptabilidade. A literatura especializada propõe três paradigmas principais para lidar com essas demandas:

1. **Aprendizado Independente (IQL - Independent Q-Learning)** (**??**): Nesta abordagem, cada agente aprende uma função de valor $Q^i(o^i, a^i)$ de forma totalmente descentralizada, ignorando as ações e observações dos demais. A simplicidade computacional do IQL o torna escalável para grandes enxames, mas a falta de modelagem explícita das interdependências entre agentes frequentemente resulta em coordenação subótima, especialmente em tarefas que exigem sincronização ou divisão de recursos (**??**).

2. **Treinamento Centralizado com Execução Descentralizada (CTDE)** (**??**): Para superar as limitações do IQL, métodos como o QMIX utilizam informações globais durante o treinamento (e.g., estados agregados do enxame) enquanto mantêm políticas de execução baseadas em observações locais. O QMIX, por exemplo, impõe uma fatorização monotônica das funções-Q individuais ($\partial Q_{\text{total}}/\partial Q^i \geq 0$), garantindo que a maximização dos Q-valores locais corresponda à otimização do valor global do enxame. Essa estratégia é particularmente eficaz em missões de vigilância cooperativa, onde a coordenação tácita é crítica (**??**).

3. **Treinamento e Execução Totalmente Descentralizados (DTDE)** (**??**): Algoritmos como o IPPO (*Independent Proximal Policy Optimization*) priorizam a escalabilidade extrema, permitindo que cada agente treine e execute políticas baseadas apenas em observações locais. Embora adequado para enxames massivos em ambientes com restrições de comunicação, o DTDE enfrenta dificuldades em cenários que exigem sincronização fina entre agentes, como formação dinâmica em espaços congestionados (**??**).

Dentre os algoritmos de Aprendizado por Reforço Multiagente (MARL) mais consolidados para controle de enxames, destacam-se:

- **MADDPG** (**??**): Baseado no paradigma CTDE (Centralized Training with Decentralized Execution), combina redes *actor-critic* com críticos centralizados, sendo especialmente eficaz em espaços de ação contínuos — ideal para ajustes precisos de trajetória em UAVs.

- **VDN** (**??**): Decompõe o valor global do enxame em uma soma de Q-valores individuais, o que facilita a otimização distribuída em tarefas como a cobertura eficiente de áreas.

- **Mean-Field MARL** (**??**): Modela as interações entre agentes como a média do comportamento coletivo, reduzindo significativamente a complexidade computacional — uma abordagem eficaz em cenários envolvendo centenas de UAVs.

Apesar dos avanços recentes, diversos desafios práticos ainda persistem. Algoritmos CTDE, como o QMIX, demandam comunicação de alta frequência durante o treinamento, o que limita sua aplicabilidade em sistemas com restrições energéticas (**??**). Por outro lado, abordagens totalmente descentralizadas (DTDE) enfrentam a "maldição da dimensionalidade", especialmente em ambientes parcialmente observáveis (**??**). Além disso, a maioria das validações ocorre em simulações homogêneas, enquanto aplicações reais introduzem variabilidades como heterogeneidade de sensores, atrasos de comunicação e falhas de hardware não modeladas (**??**).

Nesse contexto, a integração de MARL com técnicas de *transfer learning* e arquiteturas neuro-simbólicas desponta como uma estratégia promissora para superar essas limitações e aproximar a pesquisa do uso prático em campo.

2.5 Máquinas de Recompensa

2.5.1 Conceitos sobre RMs

As *Reward Machines* (RMs) consistem em um formalismo baseado em autômatos finitos que visa estruturar e modularizar funções de recompensa em problemas de aprendizado por reforço (RL). Tradicionalmente, as funções de recompensa são tratadas como "caixas-pretas", sendo acessadas pelo agente apenas para consulta pontual de valores. As RMs propõem uma abordagem diferente, em que a estrutura interna da função de recompensa é explicitada ao agente, permitindo que ele utilize tal conhecimento para acelerar e modular o processo de aprendizado (**??**).

Definição 3 (Máquina de Recompensas)**.** *Uma* ***Reward Machine*** *(RM) é uma tupla* $M = \langle U, u_0, F, \delta_u, \delta_r, P \rangle$*, onde:*

- U *é um conjunto finito de estados da máquina de recompensa;*
- $u_0 \in U$ *é o estado inicial da RM;*
- $F \subseteq U$ *é o conjunto de estados terminais;*
- P *é um conjunto de proposições que descrevem eventos observáveis no ambiente;*
- $\delta_u : U \times 2^P \to U \cup F$ *é a função de transição da RM, que define a mudança de estados da RM com base nos eventos observados;*
- $\delta_r : U \times 2^P \to \mathbb{R}$ *é a função de recompensa que associa uma recompensa real a cada transição da RM.*

Formalmente, uma RM é composta por um conjunto de estados U, uma função de transição δ_u, e uma função de recompensa δ_r. O agente, ao interagir com o ambiente,

transita não apenas pelos estados do ambiente, mas também pelos estados da RM, de acordo com eventos de alto nível observados no ambiente e definidos via uma função de rotulagem L. Cada transição na RM pode especificar uma recompensa distinta, tornando possível descrever recompensas não-Markovianas ou recompensas que dependem de propriedades temporais do histórico do agente.

A principal vantagem das RMs está na capacidade de decompor missões complexas em *subtarefas* modulares, facilitando a especificação de propriedades temporais como sequências de eventos, loops e condicionais. Por exemplo, em um cenário de entrega de pacotes com UAVs, uma RM pode especificar que primeiro o agente deve "localizar o alvo" e, em seguida, "entregar o pacote" em outra localização, premiando adequadamente cada etapa.

Por fim, as RMs possuem o mesmo poder expressivo de linguagens regulares, sendo capazes de capturar propriedades temporais similares às especificadas por lógicas como LTL (*Linear Temporal Logic*), oferecendo uma alternativa prática e compacta para a especificação de tarefas em RL.

Definição 4 (MDP com Reward Machine (MDPRM))**.** *Um* ***MDP com Reward Machine*** *é uma tupla estendida* $T = \langle S, A, p, \gamma, P, L, M \rangle$*, onde:*

- S *é o conjunto finito de estados do ambiente;*
- A *é o conjunto finito de ações disponíveis ao agente;*
- $p : S \times A \times S \to [0, 1]$ *é a função de transição estocástica do ambiente;*
- $\gamma \in (0, 1]$ *é o fator de desconto;*
- P *é o conjunto de proposições de eventos (compartilhado com a RM);*
- $L : S \times A \times S \to 2^P$ *é a função de rotulagem, que associa a cada transição no ambiente um conjunto de proposições verdadeiras;*
- $M = \langle U, u_0, F, \delta_u, \delta_r, P \rangle$ *é a Reward Machine associada.*

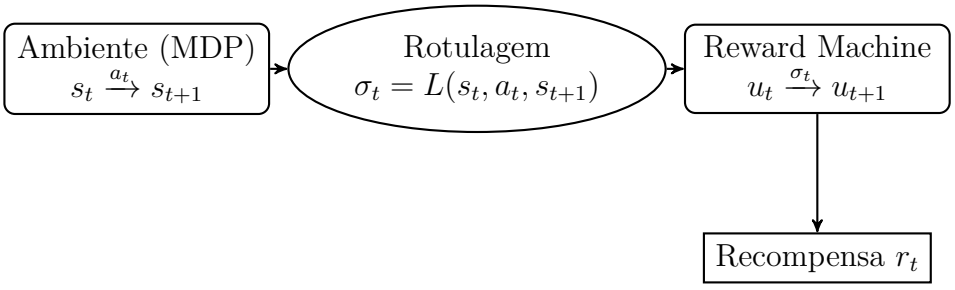

Figura 7 – Fluxo de interação entre o ambiente, a função de rotulagem e a Reward Machine.

Fonte: Adapatado de (**??**)

A figura 7 ilustra como a função de rotulagem L realiza a interface entre a transição dos estados do ambiente com a transição de estados da máquina de recompensa. O agente ao realizar a ação a_t muda o estado do ambiente de s_t para s_{t+1}. A função rotulagem L transforma a 3-tupla (s_t, a_t, s_{t+1}) em uma transição σ_t da máquina de recompensa alterando seu estado de u_t para u_{t+1}, fornecendo a recompensa r_t para o agente.

2.5.2 Exemplo RM: Grid World

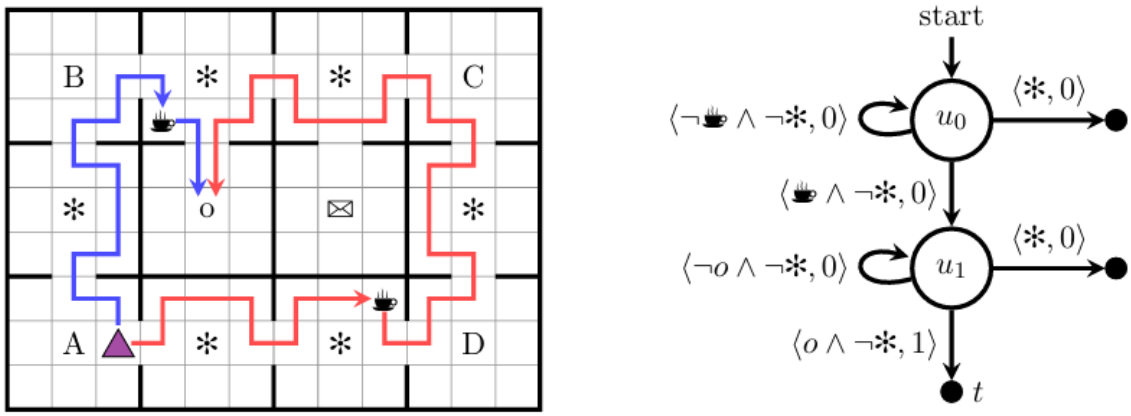

Figura 8 – Exemplo de RM para o ambiente GridWorld

Fonte: Extraído de (**??**)

A figura 8 mostra um exemplo de máquina de recompensas para o ambiente *GridWorld*. Neste ambiente o agente consegue se movimentar nas direções cardinais, seu objetivo consiste em obter o café e o jornal acessando as posições em que os itens se encontram e entrega-las até a posição do escritório o. Este é um exemplo simples no qual há um requisito temporal para as sequências das atividades que o agente deve completar antes de chegar até o seu destino final. Os rótulos sobre as setas de transições da máquina de recompensa na figura a direita indicam em forma de 2-tupla respectivamente a função de rotulagem e o retorno esperado pelo agente. Por exemplo o sob o estado u_1 o rótulo $< \neg o \wedge \neg *, 0 >$ indica que quando o agente movimenta-se para uma posição que não é o escritório (o) e também não possui um obstáculo $(*)$ ele recebe a recompensa escalar 0 e mantém a máquina de estados no estado u_1.

3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os trabalhos identificados a partir de uma revisão sistemática da literatura. É importante ressaltar que os trabalhos incluídos aqui representam apenas um extrato da literatura acadêmica, servindo como referencial para o desenvolvimento desta dissertação.

3.1 Revisão Sistemática da Literatura

Esta revisão foca especificamente em abordagens baseadas em IA, particularmente Aprendizado por Reforço (RL) e Aprendizado Profundo (DL), em vez de oferecer uma análise ampla dos métodos tradicionais. Esse foco é motivado pelo crescente consenso na literatura de que as técnicas de IA são essenciais para superar as limitações das abordagens convencionais na adaptação a ambientes complexos e dinâmicos. Os VANTs modernos agora são equipados com processadores poderosos e sensores avançados, permitindo tomada de decisão em tempo real e ajustes dinâmicos às mudanças ambientais. Os objetivos que esta revisão busca enfatizar são:

- **Validação em Ambientes Reais e Simulações de Alta Fidelidade**: Destacando métodos testados em enxames de VANTs físicos ou simuladores 3D avançados para fornecer insights práticos sobre sua viabilidade.
- **Contribuições Específicas para Tarefas**: Detalhando como os métodos de RL e DL abordam tarefas essenciais do enxame, como planejamento de trajetória, prevenção de colisões e controle de formação em ambientes reais ou simulados.
- **Métricas de Desempenho e Trade-Offs**: Avaliando escalabilidade, eficiência energética e robustez, além de identificar compromissos e desafios na implementação de sistemas de enxame baseados em IA.

3.2 Questões da RSL

Os questionamentos que guiaram a RSL foram:

1. **RQ1**: Como as abordagens de aprendizado por reforço (RL) e aprendizado profundo (DL) podem lidar com os desafios de escalabilidade, adaptabilidade e robustez em sistemas de enxame de VANTs?
2. **RQ2**: Quais são os algoritmos de RL e DL mais avançados utilizados em sistemas de enxame de VANTs?

3. **RQ3**: Quais são as contribuições específicas de RL e DL para tarefas de enxame de VANTs, como planejamento de trajetória, prevenção de obstáculos e controle de formação?

4. **RQ4**: Quais métricas de desempenho são comumente utilizadas para avaliar sistemas de enxame de VANTs?

5. **RQ5**: Como as abordagens de RL e DL melhoram a eficiência energética, a taxa de sucesso das missões e a adaptabilidade em sistemas de enxame de VANTs?

6. **RQ6**: Até que ponto os métodos de RL e DL são validados em cenários reais de enxames de VANTs e quais são os desafios para reduzir a lacuna entre simulação e implementação?

3.3 Metodologia

O processo de busca para esta revisão sistemática foi orientado pelo framework PICO (População, Intervenção, Comparação, Resultado), uma metodologia amplamente estabelecida e utilizada em revisões sistemáticas para desenvolver estratégias de busca abrangentes e focadas. O framework PICO facilitou a criação de uma string de busca projetada para capturar os estudos mais relevantes na área de sistemas de enxame de VANTs baseados em IA.

Tabela 1 – Construção da string de busca utilizando o framework PICO

Letra	**Componente**	**Termos buscados**
P	Population	“Multi UAV”, “autonomous drones”, “UAV Swarm”, “unmanned aerial vehicle swarm”, “autonomous UAV”
I	Intervention	“multi-agent reinforcement learning”, “Deep Reinforcement Learning”, “Reinforcement learning”, “Deep Learning”, “neural networks”, “MARL”, “DRL”
C	Comparison	“exploration”, “avoidance”, “planning”, “formation”, “coordination”
O	Outcome	“performance”, “efficiency”, “adaptability”, “robustness”, “scalability”

3.3.1 Base de Dados e Resultados

Para garantir uma coleta abrangente e focada de literatura relevante para esta revisão sistemática, foram utilizadas duas bases de dados acadêmicas renomadas, **Scopus** e **IEEE Xplore**. Essas bases foram selecionadas devido à sua ampla cobertura de publicações de alta qualidade nas áreas de robótica, inteligência artificial e sistemas de VANTs.

3.3.2 Definição dos Critérios

Para garantir a seleção dos estudos mais relevantes para as questões de pesquisa e os objetivos desta revisão, um conjunto de critérios de exclusão foi definido e aplicado sistematicamente durante o processo de triagem. Esses critérios foram elaborados para filtrar estudos que não apresentem relevância, rigor metodológico ou alinhamento com o foco principal da revisão. Abaixo estão as principais razões para cada critério:

- **Exclusões Baseadas no Escopo**: Estudos que não abordam enxames de VANTs (por exemplo, sistemas de um único VANT, robôs terrestres) foram excluídos para garantir que a revisão permaneça alinhada com o tema específico da inteligência de enxames de VANTs. Artigos não relacionados à inteligência artificial ou que não discutem técnicas de IA, como aprendizado por reforço (RL) ou aprendizado profundo (DL), foram excluídos, uma vez que a revisão enfatiza o papel da IA no controle de enxames de VANTs.
- **Exclusões Baseadas no PICO**: Artigos que não abordam os resultados pretendidos (por exemplo, adaptabilidade, robustez ou escalabilidade) foram excluídos para manter a relevância com os objetivos da pesquisa.
- **Exclusões Baseadas na Metodologia**: Estudos com metodologias vagas ou incompletas, como ausência de detalhes sobre algoritmos, simulações ou configurações experimentais, foram excluídos para garantir a inclusão de pesquisas reproduzíveis e confiáveis. Artigos não revisados por pares foram excluídos, a menos que fossem pré-prints relevantes de repositórios renomados, como o arXiv, garantindo o rigor científico dos estudos incluídos.
- **Exclusões Baseadas na Aplicação**: Artigos focados em aplicações ou tarefas fora do escopo do comportamento de enxames de VANTs (por exemplo, aplicações industriais, formações de robôs terrestres) foram excluídos. Estudos que não demonstram resultados de simulação em um ambiente 3D ou que não apresentam validação experimental foram excluídos, pois esses aspectos são essenciais para avaliar a aplicabilidade no mundo real e a escalabilidade das abordagens analisadas.

O diagrama ilustrado na figura 9 resume o processo de seleção dos trabalhos.

3.4 Resultados

Os artigos selecionados destacam coletivamente o papel transformador das técnicas de Aprendizado por Reforço (RL) e Aprendizado Profundo (DL) no avanço das capacidades dos sistemas de enxame de VANTs. Por fim a tabela 2 resume as informações dos trabalhos considerados.

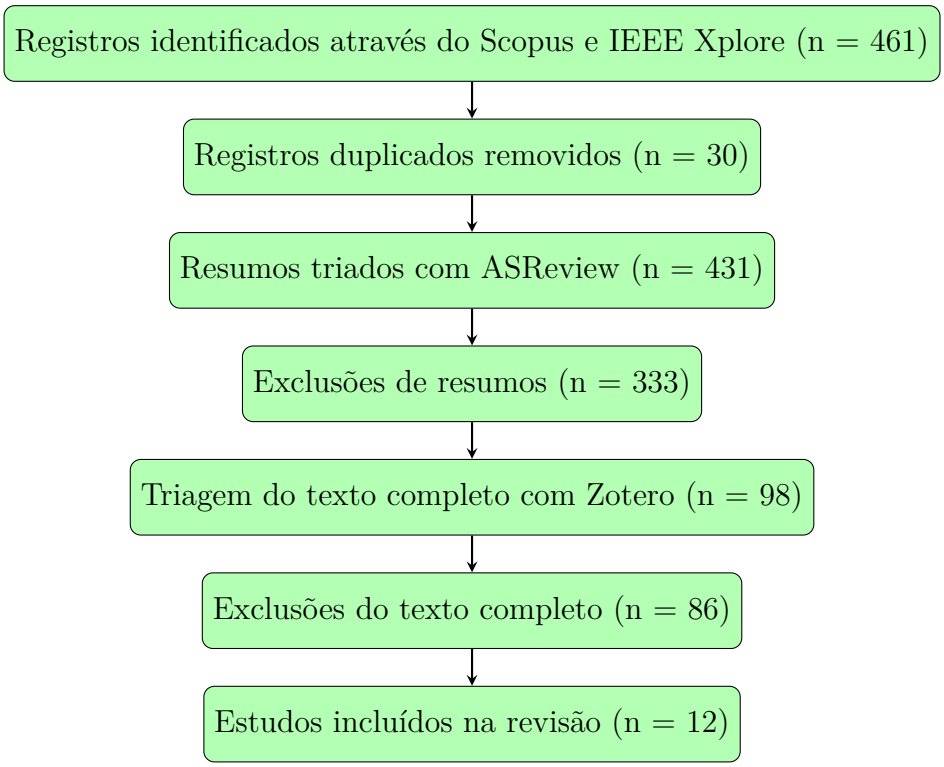

Figura 9 – Fluxograma PRISMA ilustrando o processo de revisão sistemática

Tabela 2 – Resumo dos trabalhos considerados pela revisão

Contribuição	**Algoritmos**	**Arquitetura**	**Tarefa**
Dynamic formation control using DRL under operational uncertainty (**??**).	Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG)	Centralized	Formation control, collision avoidance
Multi-agent PPO for combat UAV decision-making with reduced oscillation and enhanced stability (**??**).	Multi-Agent PPO	Centralized	Maneuvering decision-making
Distributed area coverage using FDSAC with centralized training and decentralized execution (**??**).	Fully Decentralized Soft Actor-Critic (FDSAC)	Decentralized	Dynamic area coverage
Multi-agent DRL for coverage-aware task allocation combining UAV and human collaboration (**??**).	Multi-Agent Actor-Critic (MAAC)	Decentralized	Task allocation, coverage optimization
Multi-drone shepherding for flock management using MTDDPG (**??**).	Multi-Task Deep Deterministic Policy Gradient (MTDDPG)	Decentralized	Shepherding, collision avoidance
MARL with adversarial randomization for robustness in dynamic environments (**??**).	Adversarial Domain Randomization with PPO	Decentralized	Dynamic task allocation, collision avoidance
Two-stage DRL for efficient target tracking with expert-guided exploration (**??**).	Double Deep Q-Network (DDQN), Expert-Guided Exploration	Decentralized	Target tracking
CBC-TP Net for UAV pursuit-evasion games in complex urban settings (**??**).	Cross-Boundary Cooperative Task-Planning Net (CBC-TP Net)	Decentralized	Pursuit-evasion
Vision-based formation control using YOLOv7 and DeepSORT in GNSS-denied environments (**??**).	YOLOv7, DeepSORT	Hybrid Hierarchical	Formation control
PPO with RNN layers for path planning in partially observable environments (**??**).	PPO with Recurrent Neural Network (RNN) layers	Decentralized	Path planning, collision avoidance
IPBO framework for multitarget tracking using island-based decentralized policy optimization (**??**).	Island Policy-Based Optimization (IPBO)	Decentralized	Multitarget tracking
Belief-policy interrelation framework using GAIL for coordination in heterogeneous swarms (**??**).	Generative Adversarial Imitation Learning (GAIL)	Decentralized	Coordination, formation control

3.4.1 Aprimoramento na Execução de Tarefas

3.4.1.1 Controle Dinâmico de Formação

Vários artigos, incluindo (**??**), demonstram a integração de algoritmos de RL e DL para permitir que enxames de VANTs mantenham formações sinérgicas em ambientes incertos. Por exemplo, o framework P-DRL em (**??**), utilizando algoritmos como A3C e DQN, aborda a incerteza operacional ao dividir o problema de controle em subproblemas mais simples, reduzindo significativamente a complexidade computacional, enquanto mantém a capacidade de resposta em tempo real. No entanto, limitações na prevenção de colisões indicam que melhorias adicionais na interação entre múltiplos agentes são necessárias.

3.4.1.2 Manobras e Rastreamento de Múltiplos Alvos

Os artigos (**??**) e (**??**) enfrentam os desafios da tomada de decisão cooperativa e do rastreamento de múltiplos alvos em ambientes 3D dinâmicos. Utilizando métodos avançados de MARL, como MP3O e frameworks otimizados por recompensa, esses trabalhos alcançam melhorias notáveis na eficiência da tomada de decisão e na versatilidade do enxame. Além disso, o conceito de enxame dinâmico em (**??**) oferece soluções inovadoras para particionamento e reagrupamento de VANTs com base na distribuição de alvos, aumentando a flexibilidade operacional.

3.4.1.3 Alocação de Tarefas e Cobertura de Área

A alocação de tarefas e a cobertura de área emergem como áreas de foco críticas em (**??**) e (**??**). Os algoritmos baseados em DRL propostos distribuem de forma eficiente as tarefas de sensoriamento e ajustam dinamicamente os pontos de cobertura. Por exemplo, o método de alocação de tarefas Pareto-ótimo em (**??**) combina a colaboração entre VANTs e humanos, enfatizando aplicações no mundo real, como cidades inteligentes e serviços públicos. Esses estudos ressaltam o potencial das técnicas de DL na otimização do desempenho das tarefas, ao mesmo tempo em que lidam com restrições como comunicação e poder computacional.

3.4.1.4 Prevenção de Colisões e Navegação

Técnicas como as descritas em (**??**) e (**??**) priorizam a navegação segura e a prevenção de obstáculos por meio de mecanismos avançados de recompensa e frameworks de treinamento com RL. A abordagem DRL em duas etapas apresentada em (**??**), incorporando experiência de especialistas, melhora a velocidade de convergência e a capacidade de generalização em ambientes dinâmicos. Isso reduz os custos de treinamento, garantindo uma navegação confiável para múltiplos agentes.

3.4.2 Melhorias de Desempenho e Trade-Offs

3.4.2.1 Escalabilidade

A escalabilidade, um desafio crítico em sistemas de enxame de VANTs, foi abordada de forma eficaz em diversos estudos. Por exemplo, o framework P-DRL em (**??**) demonstrou controle de formação em tempo real para enxames compostos por 10–20 VANTs, ampliando significativamente a escalabilidade da coordenação multiagente. Da mesma forma, (**??**) introduziu um conceito de enxame dinâmico que permitiu a partição e reagrupamento de grupos de VANTs de forma adaptativa, garantindo flexibilidade para atender às mudanças nos requisitos das tarefas. Essas abordagens ressaltam o potencial dos métodos de RL na expansão das operações do enxame sem comprometer o desempenho, embora frequentemente exijam recursos computacionais substanciais durante o treinamento.

3.4.2.2 Tempo de Execução

A redução do tempo de execução para tarefas como planejamento dinâmico de trajetória e controle de formação é essencial para aplicações em tempo real. Estudos como (**??**) utilizaram aprendizado por reforço profundo multitarefa (MT-DDPG) para acelerar a conclusão de operações de pastoreio. Da mesma forma, (**??**) empregou modelos leves de DL, como o YOLOv7, para realizar controle de formação baseado em visão em tempo real, demonstrando tempos de execução adequados para processamento embarcado. No entanto, essas melhorias aumentam a complexidade algorítmica e exigem treinamento intensivo em recursos computacionais.

3.4.2.3 Taxa de Sucesso

O aprimoramento da taxa de sucesso em tarefas do enxame, como prevenção de colisões e alocação dinâmica de tarefas, foi um foco central. Por exemplo, (**??**) relatou maior estabilidade e eficiência na tomada de decisão cooperativa utilizando o algoritmo MP3O, melhorando as taxas de sucesso em manobras de combate aéreo. Além disso, (**??**) demonstrou que o pré-treinamento de agentes com dados especializados levou a uma convergência mais rápida e a resultados de rastreamento mais confiáveis, especialmente em ambientes com obstáculos. No entanto, esses sucessos geralmente dependem de um design cuidadoso das funções de recompensa e otimizações específicas do domínio, o que pode limitar a generalização dos métodos.

3.4.2.4 Resiliência

A resiliência a incertezas ambientais e falhas do sistema foi destacada em estudos que empregam técnicas de randomização de domínio. (**??**) propôs a randomização adversarial de domínio para melhorar a robustez das políticas MARL na transição de simulação para

realidade, garantindo desempenho consistente em diversos cenários operacionais. Da mesma forma, (**??**) introduziu um framework de perseguição e evasão que manteve o sucesso da missão mesmo quando alguns VANTs do enxame foram comprometidos. Embora esses métodos aumentem a resiliência, geralmente exigem alto custo computacional e ambientes de simulação extensivos para um treinamento eficaz.

3.4.2.5 Trade-Offs

- **Demandas Computacionais**: Métodos como randomização adversarial de domínio (**??**) e frameworks hierárquicos profundos (**??**) exigem altos recursos computacionais, o que pode limitar sua implementação em sistemas de VANTs com restrições de hardware.
- **Complexidade do Treinamento**: Técnicas que enfatizam adaptabilidade e resiliência, como reconfiguração dinâmica de enxame (**??**) ou DRL em duas etapas (**??**), frequentemente envolvem pipelines de treinamento complexos, demandando expertise significativa e tempo de desenvolvimento.
- **Implantação no Mundo Real**: Embora os resultados em simulação sejam promissores, métodos como P-DRL (**??**) e MP3O (**??**) enfrentam desafios na transição para aplicações reais devido a fatores como imprecisões nos sensores e restrições de comunicação.

3.5 Discussão

3.5.1 Principais Descobertas

Esta revisão sistemática explorou as contribuições das técnicas de Aprendizado por Reforço (RL) e Aprendizado Profundo (DL) para o avanço dos sistemas de enxame de VANTs, destacando seu potencial transformador na solução de desafios críticos. Nos artigos revisados, foram alcançadas melhorias significativas em escalabilidade, tempo de execução, taxa de sucesso e resiliência. As principais contribuições incluem:

- **Controle Dinâmico de Formação**: Estudos como (**??**) demonstraram frameworks capazes de realizar controle de formação em tempo real para enxames de VANTs de médio porte, reduzindo a complexidade computacional por meio de algoritmos avançados de RL, como A3C e DQN.
- **Tomada de Decisão Multiagente e Rastreamento**: Abordagens inovadoras, como MP3O ((**??**)) e conceitos de enxame dinâmico ((**??**)), aprimoraram significativamente a tomada de decisão cooperativa e o rastreamento de múltiplos alvos, melhorando a eficiência operacional e a adaptabilidade.

- **Alocação de Tarefas e Cobertura de Área**: Métodos como alocação de tarefas Pareto-ótima ((**??**)) e cobertura baseada em FDSAC ((**??**)) apresentaram técnicas eficazes para equilibrar a distribuição da carga de trabalho e otimizar a eficiência do sensoriamento, mesmo em ambientes com restrições de comunicação.

- **Prevenção de Colisões e Navegação**: Frameworks como os descritos em (**??**) e (**??**) forneceram soluções robustas para navegação em ambientes complexos, com avanços em modelagem de recompensas e randomização de domínio, garantindo operações livres de colisões e melhores transições entre simulação e realidade.

- **Validações em Ambientes Reais**: Alguns estudos ((**??**), (**??**)) validaram seus métodos em cenários reais, demonstrando viabilidade prática e oferecendo uma base para melhorias futuras.

3.5.2 Lacunas e Desafios Não Resolvidos

Apesar desses avanços, diversas lacunas permanecem, apontando para oportunidades de pesquisa futura:

1. **Escalabilidade para Enxames Maiores**: Embora muitos métodos tenham melhorado a coordenação em enxames de médio porte (por exemplo, 10–20 VANTs em (**??**)), a escalabilidade para enxames maiores ainda é um desafio aberto. São necessários algoritmos eficientes que possam escalar sem um aumento proporcional na complexidade computacional.

2. **Redução da Lacuna entre Simulação e Realidade**: Embora métodos de randomização de domínio ((**??**)) tenham mostrado potencial, a tradução dos resultados simulados para operações no mundo real é limitada por ruído nos sensores, variabilidade ambiental e restrições do sistema. Melhorar a robustez em aplicações reais continua sendo essencial.

3. **Eficiência Energética e Otimização de Recursos**: Embora alguns estudos tenham abordado o consumo de energia indiretamente, estratégias dedicadas para otimizar o uso energético em todo o enxame ainda são pouco exploradas.

4. **Métricas Padronizadas e Benchmarking**: A ausência de métricas padronizadas entre os estudos ((**??**), (**??**)) dificulta comparações diretas. O estabelecimento de um framework unificado de benchmarking facilitaria avaliações consistentes do desempenho dos enxames.

5. **Integração de Sistemas Heterogêneos**: Estudos como (**??**) introduziram esforços iniciais para a coordenação de enxames heterogêneos, mas são necessárias investiga-

ções mais aprofundadas sobre o equilíbrio das disparidades de recursos e a alocação de tarefas entre diferentes tipos de VANTs.

3.6 Comparação com Estado da Arte

A tabela 3 ilustra como o trabalho proposto se posiciona em relação aos trabalhos da revisão de literatura. Os critérios adotados na comparação foram:

- Execução Descentralizada (DE): Verifica se a abordagem adotada pelo trabalho possibilita execução descentralizada pelos agentes, isto é, se o enxame não necessita de um nó central que envia os comandos para execução.
- Escalabilidade (Scalable): Verifica se a abordagem de controle do enxame permite a quantidade de agentes pode aumentar sem afetar o desempenho.
- Coordenação (Coord): Se há coordenação entre as ações dos agentes no ambiente que executam a missão.
- Espaço de Ação Contínuo (Cont): Verifica se o domínio do espaço de ações dos agente é contínuo ou discreto.
- Visão Computacional (Vision): Se a abordagem utiliza algum tipo de informação visual capturada pelos agentes em seu espaço de observações do algoritmo.
- Comunicação (Comm): Se há troca de informações entre os agentes através de algum meio de comunição.
- Missão Global (GM): Se a missão do enxame envolve tarefas complexas além das tarefas atividades comuns, como por exemplo: rastreamento de alvos, transporte de carga, cobertura de área e reconstrução 3D.
- Planejamento de Caminho (PP): Se o trabalho lida com a atividade de planejamento de trajetória.
- Prevenção de Colisões (CA): Se o trabalho aborda a atividade de prevenção de colisões.
- Controle de Formação (FC): Se o trabalho aborda a atividade do controle de formação do enxame.

Tabela 3 – Quadro comparativo da proposta com estado da arte.

Trabalho	DE	Scalable	Coord	Cont	Vision	Comm	GM	PP	CA	FC
(??)	✗	✓	✓	✓	✗	✓	✗	✓	✓	✓
(??)	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗
(??)	✓	-	✗	✗	✗	✓	✓	✗	✓	✗
(??)	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗
(??)	✓	✗	✓	✓	✗	✗	✓	✓	✗	✓
(??)	✓	✗	✓	✓	✓	-	✓	✓	✗	✓
(??)	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗
(??)	✓	✗	✓	✓	✗	✓	✓	✓	✓	✗
(??)	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓
(??)	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗
(??)	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
(??)	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓
Proposta	✓	✓	✓	-	✓	✓	✓	✓	✓	✓

4 DESENVOLVIMENTO DO TRABALHO

Este capítulo apresenta o desenvolvimento do trabalho proposto, abordando de forma detalhada a modelagem do problema de navegação cooperativa de enxames de veículos aéreos não tripulados. São descritos os ambientes de simulação utilizados, a especificação das tarefas atribuídas aos agentes, a definição dos espaços de estados e ações, a formulação das funções de recompensa, bem como os algoritmos de aprendizado por reforço multiagente e as estratégias de treinamento empregadas ao longo do estudo.

Inicialmente, o trabalho teve como objetivo utilizar exclusivamente o simulador AirSim como plataforma de desenvolvimento e experimentação. Essa escolha fundamentou-se na proposta de aprender uma política de controle de alto nível para coordenação de enxames, de modo a facilitar uma possível transferência de aprendizado para plataformas reais de VANTs. Nesse contexto, o aprendizado não teria como finalidade o controle de baixo nível, como a estabilização do voo, mas sim a coordenação estratégica dos agentes por meio do envio de comandos abstratos aos controladores de voo embarcados.

O simulador AirSim fornece uma camada de abstração compatível com controladores amplamente utilizados, como ArduPilot e PX4, permitindo que a política aprendida atue de forma semelhante ao que ocorreria em um cenário real de voo autônomo baseado em scripts de missão. Dessa forma, o ambiente de simulação possibilitou o estudo detalhado de aspectos relacionados à modelagem de tarefas cooperativas, percepção do ambiente e coordenação entre múltiplos agentes em cenários tridimensionais realistas.

Entretanto, conforme será discutido na seção de comparação entre os ambientes de simulação, tornou-se necessário adotar uma plataforma alternativa para a realização dos experimentos finais, em função da aproximação do cronograma de conclusão do trabalho e das demandas computacionais associadas ao treinamento de algoritmos de aprendizado por reforço multiagente. O simulador IsaacSim, aliado ao framework IsaacLab, apresentou desempenho computacional superior, maior escalabilidade e melhor suporte à paralelização de ambientes, possibilitando a obtenção de resultados mais consistentes e a prototipação rápida de diferentes configurações experimentais.

Dessa forma, o IsaacSim foi selecionado como a plataforma principal para o treinamento final dos modelos e a apresentação dos resultados deste trabalho. Ressalta-se, contudo, que a utilização prévia do AirSim desempenhou um papel fundamental no andamento da pesquisa, especialmente por sua especialização em aplicações envolvendo VANTs. O uso desse simulador contribuiu de maneira significativa para o entendimento do problema, para a validação conceitual das abordagens propostas e para a definição da metodologia adotada nos experimentos finais.

4.1 Modelagem do Problema

Esta seção apresenta a modelagem formal do problema de navegação cooperativa de enxames de VANTs, contemplando os ambientes de simulação adotados, a definição das tarefas, os espaços de estados e ações, e as funções de recompensa utilizadas no processo de aprendizado.

4.1.1 Ambientes de Simulação

4.1.2 Especificação das Tarefas dos Agentes

4.1.3 Espaço de Estados e Ações

4.1.4 Funções de Recompensa

4.2 Algoritmo de Aprendizado

Esta seção descreve o algoritmo de aprendizado por reforço multiagente utilizado, bem como sua implementação computacional nos diferentes ambientes de simulação.

4.2.1 MAPPO: Formulação e Pseudocódigo

4.2.2 Implementação Computacional

4.3 Estratégias de Treinamento

Esta seção apresenta as estratégias de treinamento adotadas, incluindo a comparação entre simuladores e as abordagens baseadas em aprendizado curricular com e sem o uso de Reward Machines.

4.3.1 Treinamento no AirSim

4.3.2 Treinamento no IsaacLab

4.3.3 Comparação de Performance entre Simuladores

4.3.4 Abordagem Baseline

4.3.5 Curriculum Learning com Reward Machines

5 PLANO DE AÇÃO

5.1 Metodologia

As principais atividades esperadas para o desenvolvimento da proposta são destacadas a seguir:

1. **Revisão de Literatura**: buscar exemplos de algoritmos MARL (IPPO, QMIX); analisar trabalhos que utilizam máquinas de recompensas (RM) em configurações simples e de múltiplos agentes RL; identificar limitações existentes em projeto de recompensas para múlitplas tarefas.

2. **Projetar Máquina de Recompensas**: decompor a missão de rastreamento de alvos em subtarefas gerando os estados da máquina como: desvio de obstáculos, controle de formação, indentificação de alvo.Especificar os eventos que disparam as transições entre os estados (e.g.,SE distancia alvo $<$ 10m ENTÃO mudar para estado de rastreamento).

3. **Adaptação dos Algoritmos MARL** integrar a máquina de recompensa nos algoritmos MARL para que durante o treinamento do agente seja considerado os estados da máquina q_{RM}.

    - Para o algoritmo IPPO: aumentar as observações do agente com o estado da máquina: $o'_i = [o_i, q_i]$
    - Para o algoritmo QMIX: além de extender o espaço de observações, gerar novas experiências sintéticas a partir da técnica CRM (**Counterfactual Experiences for RM**). Levando em consideração todos os estados da RM para expandir o replay buffer do agente durante o treinamento.

4. **Projeto do Ambiente de Simulação**: elaborar os cenários 3D de simulação, modelar os espaços de observação, ações e funções recompensas do ambiente.

    - Utilizar as bibliotecas AirSim e PX4 AutoPilot para modelagem da dinâmica física do drone seus recursos embarcados.
    - Integrar a modelagem do ambiente com a biblioteca Gymnasium para possibilitar o treinamento dos algoritmos MARL.
    - Implementar obstaculos e alvos dinâmicos dentro do cenário.

5. **Estabelecimento de Métricas**: definir métricas de desempenho para avaliação do treinamento, como: erro de rastreamento (RMSE), taxa de colisões, convergência de treinamento, eficiência energética.

6. **Treinamento e Validação**:
    - Treinar os agentes RM-MARL em cenários progressivos: 1-alvo, obstáculos estáticos (prova de conceito); múltiplos alvos, com obstáculos dinâmicos (teste de escalabilidade).
    - Conduzir estudos de hablação: desativar as componentes da máquina de recompensas para avaliar seu impacto na performance;
7. **Coleta de Dados e Análise**: realizar analises quantitativa e qualitativa visando melhor compreensão da convergência do treinamento.
8. **Iteração e Refinamento**:
    - Refinar o projeto da máquina de recompensas (RM) baseado nos resultados empíricos;
    - Otimizar os algoritmos para escalabilidade (e.g., testes com 5, 10 e 20 VANTs).
9. **Documentação**: disponibilizar a base de código e o ambiente de simulação de forma open-source, publicação de artigos e escrita da dissertação.

5.2 Resultados Parciais

5.2.1 Implementação

Realização de simulações e experimentos utilizando algoritmos de aprendizado por reforço profundo como DQN e PPO envolvendo apenas um único agente na execução da tarefa de desvio de obstáculos. Como ilustrado na figura 10.

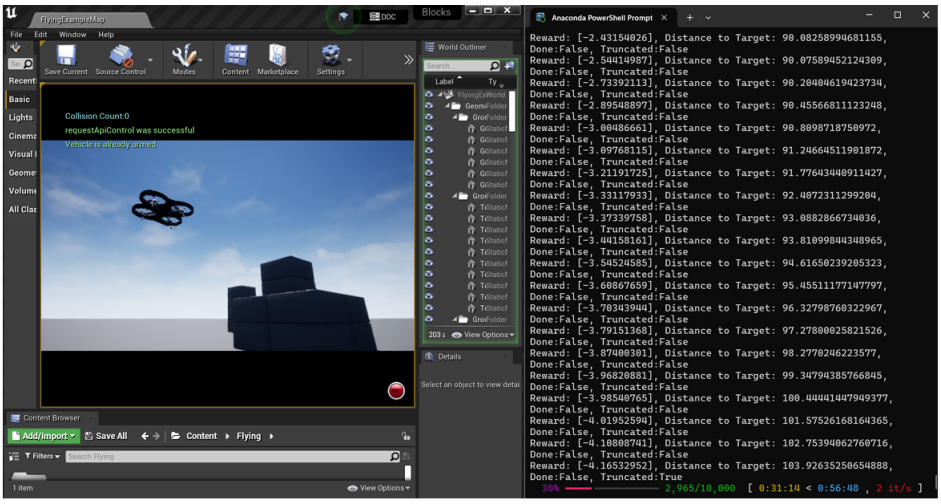

Figura 10 – Captura de tela simulação do treinamento de agente UAV.

Fonte: Autor.

A figura 11 ilustra os experimentos iniciais no controle de um enxame com 5 drones. Neste cenário o objetivo consiste em treinar o enxame para a navegação em ambiente com obstáculos estáticos mantendo a formação em V.

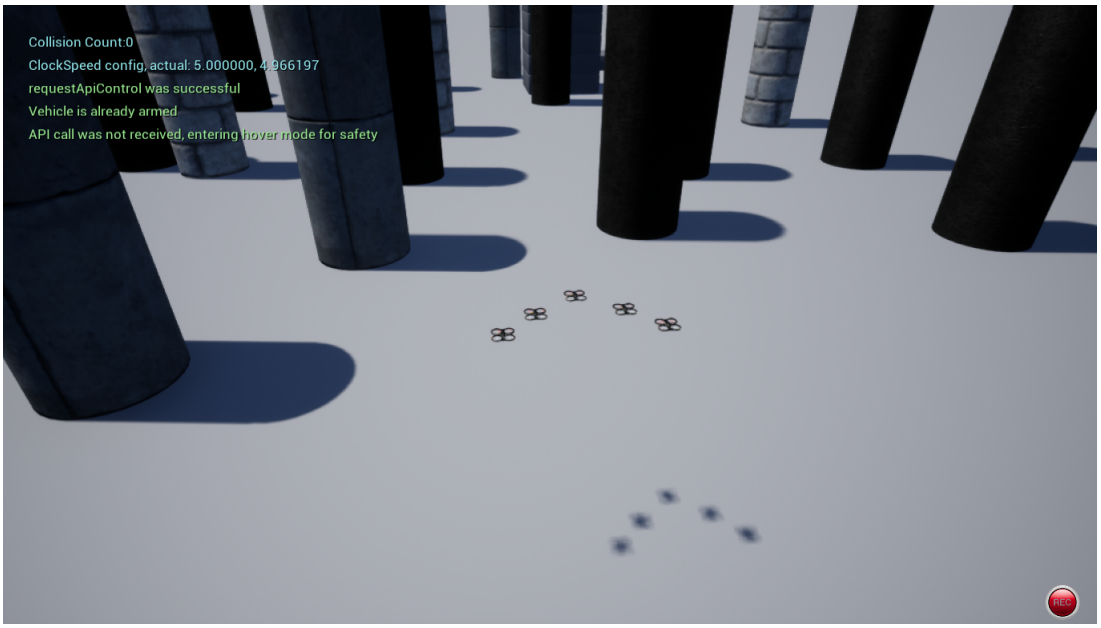

Figura 11 – Simulação Enxame de VANTs com AirSim.

Fonte: Autor.

5.2.2 Artigos Produzidos

- **Comparative Analysis of PPO and DQN for UAV Obstacle Avoidance in Simulated Environments**: artigo aceito na 11º Conferência Internacional IFAC que ocorrerá em Julho deste ano na Noruega.

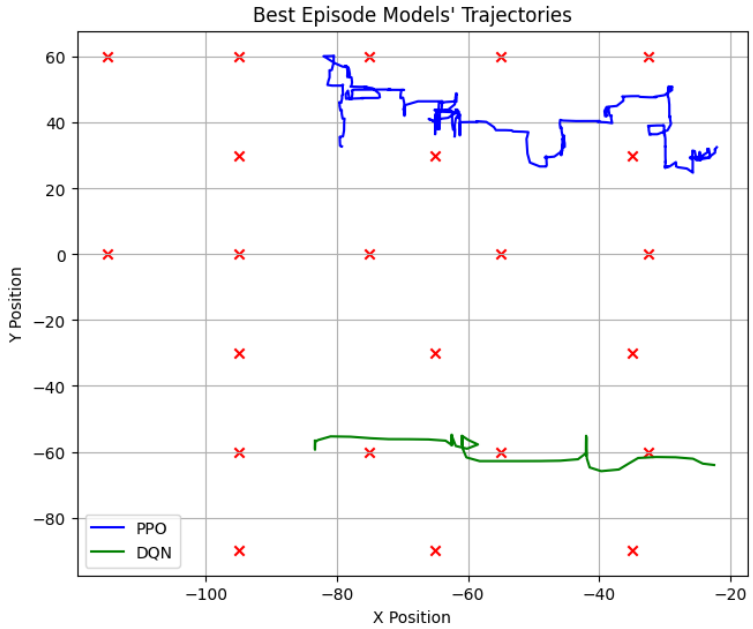

Figura 12 – Visualização 2D das trajetórias dos agentes na tarefa desvio de obstáculos.

5.3 Viabilidade

1. **Hardware Desktop para simulação**: Processador Intel Core i7 8 x4.2 Ghz 12th Gen, GPU Nvidia RTX 4080 12GB, Memória RAM 64GB. Atualmente o laboratório LIARC conta com computadores de alto desempenho capazes de suportar o volume computacional necessário das simulações.

2. **Hardware embarcado para drones**: Raspberry PI 5, Nvidia Jetson Nano. O LIARC já tem previsto recursos para aquisição no próximo ano dos componentes necessários para construção de 10 drones open hardware, com capacidade de processamento embarcado viabilizando a arquitetura de controle descentralizada.

5.4 Cronograma

O cronograma para o desenvolvimento das atividades relacionadas a esta proposta pode ser visto na figura abaixo.

ATIVIDADE	2025												LEG	STATUS
	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ		
Revisão de Literatura														
Projetar RM para Enxame VANTs														REALIZADO
Modificação dos Algoritmos														
Projeto dos cenários de Simulação														
Treinamento dos agentes														EM ANDAMENTO
Coleta e análise das métricas														
Iteração e Refinamento														
Publicação em periódicos														PREVISTO
Escrita Dissertação														
Defesa Dissertação														

Figura 13 – Cronograma da Proposta de Dissertação.

1° Ten JÚLIO CÉSAR SANTANA DA ROSA FILHO (SC 24101)
Aluno

PAULO FERNANDO FERREIRA ROSA, Ph.D.
Orientador

Maj GABRIELA MOUTINHO DE SOUZA DIAS, D.Sc.
Coordenador de Pós-graduação

Concordo com a presente Proposta de Dissertação e declaro que as necessidades para sua execução serão garantidas pela Seção.
IME, em 23 de Maio de 2025.

Cel LUÍS ANDRÉ GOMES DE ABREU
CHEFE da SE/9