Controle Seguro e Evasão de Colisões em Tráfego Denso de Drones via Aprendizado por Reforço

Henrique J. Felisardo dos Santos[1], Israel da S. Barros[1], Luiz F. Bittencourt[2], Carlos A. Kamienski[1], Fabíola M. C. de Oliveira[1]

[1]Centro de Matemática Computação e Cognição – Universidade Federal do ABC

[2]Instituto de Computação -– Universidade Estadual de Campinas

henrique.felisardo@aluno.ufabc.edu.br,

{israel.barros, carlos.kamienski, fabiola.oliveira}@ufabc.edu.br,

bit@ic.unicamp.br

Abstract. *The expansion of drones in metropolitan areas demands safe navigation in high-density environments, in which deterministic methods fail to avoid collisions. This paper proposes the LiDAR-Assisted Reinforcement Agent for Less Resolution and More Resolution (LiARA-LR and MR) models, using Proximal Policy Optimization, Curriculum Learning, and Simulator-to-Simulator validation. In critical traffic scenarios (12 drones/min), LiARA-MR reduced the collision rate to 0.43%, statistically significantly outperforming the 7.0% failure rate of the best geometric approach and the 1.77% failure rate of a baseline learning model, attesting that the proposed refined architecture with high-resolution perception enables scalable and safe autonomous aerial operations.*

Resumo. *A expansão de drones em metrópoles exige navegação segura em alta densidade, na qual métodos determinísticos falham em evitar colisões. Este artigo propõe os modelos LiDAR e Aprendizado por Reforço para Agentes de Leve Resolução e Maior Resolução (LiARA-LR e MR), usando Proximal Policy Optimization, Aprendizado Curricular e validação Simulador-para-Simulador. Em cenários críticos de tráfego (12 drones/min), o LiARA-MR reduziu a taxa de colisões para 0,43%, superando de forma estatisticamente significativa a taxa de falha de 7,0% da melhor abordagem geométrica e de 1,77% de um modelo base de aprendizado, atestando que a arquitetura refinada com percepção de alta resolução proposta viabiliza operações aéreas autônomas escaláveis e seguras.*

1. Introdução

Com o aumento no uso de Veículos Aéreos Não Tripulados (VANTs), ou drones, cresce também o interesse em suas aplicações em áreas como combate a incêndios, entregas, fiscalizações e transporte de equipamentos [Kong et al. 2023; Yasin et al. 2020]. Essa expansão acelera o adensamento do espaço aéreo em baixas altitudes, onde múltiplas aeronaves frequentemente compartilham rotas e áreas de operação, elevando significativamente o risco de colisões. Assim, garantir separação dinâmica e evitar acidentes torna-se um requisito crítico para a viabilidade e a segurança dessas operações [Jeon et al. 2021]. Antes da implementação de sistemas com múltiplos drones em uma região, é fundamental minimizar a probabilidade de colisões [Lu et al. 2023]. Para isso, a literatura propõe diferentes

estratégias de evasão, incluindo abordagens Geométricas [Oliveira et al. 2024; Seo et al. 2017; Park et al. 2008], Heurísticas e Meta-Heurísticas [Zhao et al. 2021; Wu et al. 2021], além de métodos baseados em Inteligência Artificial (IA) [Zhang et al. 2023; Jang et al. 2020]. Contudo, essas abordagens não são capazes de atingir uma taxa de colisão baixa em ambientes aéreos com alta densidade de drones.

Este artigo tem como objetivo elevar o nível de segurança em aplicações baseadas em drones por meio do desenvolvimento e avaliação de dois modelos de Aprendizado por Reforço (RL, do inglês *Reinforcement Learning*) voltados à minimização de colisões em um cenário de logística multidrones. A arquitetura proposta, denominada *LiDAR e Aprendizado por Reforço em Agentes* (LiARA), possui duas variantes definidas pela densidade do sensor simulado: Leve Resolução (LiARA-LR) e Maior Resolução (LiARA-MR). O cenário experimental é implementado em uma versão modificada do simulador UT-Sim [AlMousa et al. 2019], desenvolvido na plataforma Unity. A análise concentra-se em estratégias não colaborativas, nas quais os drones não se comunicam nem cooperam entre si, incluindo abordagens geométricas e um modelo de RL do estado-da-arte [Barros et al. 2026; Oliveira et al. 2024; Oliveira et al. 2022; Oliveira et al. 2021]. Nesses métodos, os dados são obtidos exclusivamente por sensores de detecção e alcance de luz (LiDAR, do inglês *Light Detection And Ranging*), condição necessária quando a comunicação não está disponível ou em situações de desvio emergencial.

Modelos recentes da literatura, como o Neural Autonomous UAV (NAV) [Barros et al. 2026] mostraram que técnicas como Proximal Policy Optimization (PPO), Aprendizado Curricular e sensoriamento LiDAR de baixa densidade podem gerar navegação estável, mas ainda apresentam ofuscamento espacial e degradação em tráfegos densos. Este trabalho avança além desse patamar ao adaptar o controle contínuo, o sistema de recompensas e os algoritmos de treinamento para explorar percepções espaciais de alta resolução. Com isso, o LiARA-MR reduz a taxa de colisões para 0,43% em cenários críticos, superando a melhor estratégia geométrica avaliada ($7,0\%$) e o modelo de RL de base ($1,77\%$), demonstrando maior robustez e segurança em ambientes urbanos congestionados.

Este artigo está estruturado da seguinte maneira: a Seção 2 discute os trabalhos relacionados, seguida pela Seção 3, que detalha a proposta de modelagem do agente autônomo baseado em Aprendizado por Reforço. A Seção 4 descreve a metodologia de simulação aplicada, cujos resultados são apresentados na Seção 5 e analisados na Seção 6. Por fim, a Seção 7 encerra o trabalho com as considerações finais e perspectivas futuras.

2. Trabalhos Relacionados

Os algoritmos anticolisão visam deixar as viagens de drones mais seguras, utilizando diferentes estratégias para atingir esse objetivo. Oliveira et al. [2024] propõem um conjunto de estratégias geométricas de prevenção de colisões, chamado GeoDrone (SingleDrone, MultiDrone e SpeedDrone), e o comparam com outras estratégias determinísticas da literatura, como a abordagem proposta por Seo et al. [2017] e a Vector Sharing Resolution (VSR) [Park et al. 2008]. Nessas simulações, os drones autônomos são apoiados por sensores e realizam rotas completas de logística, compreendendo a viagem desde a decolagem de um centro de distribuição, voo em cruzeiro, aterrissagem no local de entrega e retorno ao ponto inicial.

Em outras abordagens, os pesquisadores contrapõem os métodos estritamente geométricos e heurísticos [Zhao et al. 2021; Wu et al. 2021] com a utilização de IA para lidar com a imprevisibilidade do ambiente. Jang et al. [2020] exploram o RL para buscar escalabilidade na evasão de colisões na mobilidade aérea urbana. Da mesma forma, Zhang et al. [2023] propõem desvios adaptativos para múltiplos VANTs. Existem inúmeras variações e combinações de técnicas na literatura em busca de algoritmos eficientes que garantam a segurança operacional das aeronaves, variando desde arquiteturas de Aprendizado Profundo até modelos de otimização de trajetória [Kong et al. 2023].

Avançando nessa linha, Barros et al. [2026] propõem o conjunto de modelos de RL Neural Autonomous UAV (NAV), integrando Aprendizado Curricular, sensoriamento por LiDAR, PPO e Advantage Actor-Critic (A2C) e obtendo melhorias relevantes em relação às abordagens geométricas. O presente artigo diferencia-se deste trabalho ao reestruturar o processo de treinamento e refinar a engenharia de recompensas. O controle contínuo baseado em PPO segue a adaptação usada no NAV, com função Swish nas camadas ocultas e tangente hiperbólica na saída. Enquanto o NAV estabeleceu a viabilidade do PPO com sensores LiDAR de baixa densidade, os modelos LiARA propostos avançam ao mostrar empiricamente que o aumento da resolução do LiDAR reduz o ofuscamento espacial observado no NAV, permitindo políticas de evasão substancialmente mais precisas e seguras.

As estratégias propostas na literatura tendem a apresentar baixo desempenho em cenários de alta densidade ou dependem de comunicação entre os drones, o que as torna vulneráveis a falhas de rede [Oliveira et al. 2024]. Além disso, muitos modelos de RL não são testados em simuladores distintos para inferência, ficando suscetíveis ao sobreajuste ao ambiente de treinamento [Whiteson et al. 2011]. Este trabalho contrapõe essas limitações ao desenvolver uma estratégia não cooperativa guiada puramente pela percepção espacial do agente via LiDAR. De acordo com o nosso conhecimento, embora existam modelos que utilizem Aprendizado Curricular e validação Sim-to-Sim, ainda há pouca investigação sobre arquiteturas que explorem percepção espacial de alta resolução para maximizar o potencial dessas técnicas. Este artigo visa preencher essa lacuna ao focar na escalabilidade segura da frota, permitindo aumentar significativamente o número de drones em operação simultânea sem a degradação de desempenho típica das abordagens tradicionais em tráfegos densos.

3. Proposta

Esta seção detalha os modelos propostos para a navegação autônoma e evasão de colisões em cenários de alta densidade, chamados de *LiDAR e Aprendizado por Reforço em Agentes* (LiARA). O método baseia-se em RL, no qual agentes (drones) aprendem políticas de controle contínuo para operar de forma descentralizada e não-cooperativa, utilizando dados sensoriais locais simulados. A navegação autônoma descentralizada é modelada matematicamente como um Processo de Decisão de Markov Parcialmente Observável (POMDP).

Um POMDP está relacionado a cenários em que o agente deve tomar decisões em sequência sem ter acesso ao estado global do mundo. Diferente de um cenário no qual todos os obstáculos são globalmente visíveis, o drone opera com uma observabilidade parcial: os sensores detectam apenas o que está em seu entorno imediato, limitados por oclusões e pelo alcance. O processo de decisão de Markov implica que a decisão ótima

deve depender apenas da observação do estado atual do agente, sem depender de todo o histórico passado.

3.1. Modelo e Algoritmo de Aprendizado

Para resolver o problema de navegação autônoma definido pelo POMDP, adotou-se o paradigma de RL, que une a capacidade de extração de características de *Deep Learning* à otimização de decisão por tentativa e erro. A política de controle $\pi_\theta(a|o)$ é parametrizada por uma Rede Neural Artificial do tipo *Multi-Layer Perceptron* (MLP). Esta rede funciona como um "cérebro" aproximador de função, recebendo dados brutos dos sensores e aprendendo a mapear padrões complexos não-lineares para ações de controle de voo, como decolagem, cruzeiro, desvios e pousos.

O algoritmo de otimização escolhido foi o PPO [Schulman et al. 2017]. O PPO é essencial na estabilidade do voo por conta de seu mecanismo de Região de Confiança (*Trust Region*). Ao invés de permitir atualizações desproporcionais nos pesos da rede que alterem abruptamente as manobras do drone — o que frequentemente causaria o esquecimento de comportamentos seguros já aprendidos (colapso da política) —, o PPO limita matematicamente o quanto a nova política pode divergir da antiga por meio de um hiperparâmetro de *clipping*, garantindo um aprendizado seguro e progressivo.

3.2. Arquitetura da Rede e Hiperparâmetros

O modelo de controle contínuo adotado neste trabalho é uma adaptação das arquiteturas propostas por Barros et al. [2026]. A rede neural consiste em um MLP com duas camadas ocultas de 128 neurônios cada, dimensão empiricamente validada para extrair características espaciais da geometria do LiDAR (via *RayCast* na Unity) sem onerar excessivamente o custo computacional. A configuração de treinamento usando PPO — algoritmo de otimização que atualiza o comportamento do agente de forma gradual e restrita para evitar o esquecimento catastrófico de manobras seguras — foi ajustada para equilibrar a reatividade imediata e a estabilidade de voo. Os hiperparâmetros fundamentais são definidos na Figura 1 e os valores usados estão na Tabela 1.

Hiperparâmetro	Significado Teórico no Algoritmo PPO
Taxa de Aprendizado (`learning_rate`)	Define a magnitude do passo dado durante a atualização dos pesos da rede neural. Controla a velocidade geral com que o modelo assimila novas experiências.
Decaimento da Taxa (`learning_rate_schedule`)	Estratégia de redução progressiva da taxa de aprendizado. Permite ajustes amplos no início do treinamento e um refinamento delicado nas manobras finais do agente.
Regularização da Entropia (`beta`)	Grau de aleatoriedade induzido nas ações. Atua como um mecanismo de incentivo à exploração, impedindo a convergência prematura para comportamentos repetitivos.
Tamanho do Lote (`batch_size`)	Determina a quantidade de experiências processadas simultaneamente para calcular uma única atualização de gradiente. Impacta a estabilidade das otimizações.
Tamanho do *Buffer* (`buffer_size`)	Armazena temporariamente as experiências coletadas durante a simulação. É a quantidade total de dados coletados antes de pausar a simulação para atualizar a política de controle.
Horizonte de Tempo (`time_horizon`)	Define o alcance da visão preditiva do agente. Determina o equilíbrio de foco entre as recompensas imediatas (sobrevivência) e o planejamento de longo prazo.

Figura 1. Definição dos principais hiperparâmetros dos modelos propostos.

Em vez de adotar configurações padrão voltadas a longo prazo, as escolhas arquiteturais priorizaram a sobrevivência do agente em cenários de colisão iminente. O

Tabela 1. Principais hiperparâmetros do modelo PPO.

Parâmetro	**Valor**
Taxa de Aprendizado (`learning_rate`)	3.0×10^{-4}
Decaimento da Taxa (`learning_rate_schedule`)	Linear
Regularização da Entropia (`beta`)	1.0×10^{-5}
Tamanho do Lote (`batch_size`)	256
Tamanho do *Buffer* (`buffer_size`)	2048
Horizonte de Tempo (`time_horizon`)	16 passos

horizonte de tempo reduzido (16 passos) força o modelo a focar em reflexos rápidos de evasão, preterindo planejamentos incertos. Para garantir que eventos críticos de falha ou quase-colisão não fossem estatisticamente diluídos, adotou-se um *buffer* curto particionado em oito lotes rápidos, permitindo atualizações de gradiente altamente dinâmicas e baseadas em dados estritamente recentes.

A exploração do espaço de estados foi delegada majoritariamente ao Aprendizado Curricular — uma técnica de treinamento estruturada que expõe o agente a cenários de dificuldade progressiva —, o que permitiu o uso de um coeficiente de entropia bastante conservador, dispensando injeções elevadas de ruído estocástico que poderiam prejudicar a precisão da navegação. Por fim, o decaimento linear da taxa de aprendizado mostrou-se vital: viabiliza atualizações de pesos mais amplas nas lições curriculares iniciais (assimilação da dinâmica de voo) e restringe o modelo a ajustes finos e restritos nas fases avançadas, preservando as políticas de segurança já consolidadas contra a instabilidade de novos gradientes.

3.3. Espaço de Observação

A Figura 2 mostra a arquitetura da rede MLP usada para treinar a política de controle do agente, mapeando o espaço de observação para as ações contínuas de voo tridimensional. O vetor de estado fornecido à rede neural foi desenhado para mitigar a observabilidade parcial e garantir a propriedade de Markov. Ele compreende a percepção exteroceptiva, a propriocepção cinética e a navegação relativa. A percepção exteroceptiva utiliza a técnica de Ray Casting para simular um LiDAR 3D omnidirecional, emitindo feixes virtuais para medir distâncias. Para evitar a intratabilidade de fornecer nuvens de pontos brutas (dados não estruturados), as leituras sofrem um pré-processamento linear: cada raio capta a distância de impacto (limitada a 100 m) e o valor é normalizado entre $[0, 1]$. Por exemplo, um obstáculo a 20 m registra $0{,}2$, enquanto a ausência de detecção devolve $1{,}0$. Esse vetor estruturado compõe o tensor de entrada do MLP, permitindo que a rede construa uma representação volumétrica esparsa do entorno com baixo custo computacional.

Além de mapear o ambiente externo, o agente precisa de consciência sobre o próprio corpo físico, o que é fornecido pela propriocepção cinética. Esse componente insere a velocidade atual do drone como entrada na rede neural. A inclusão dessa variável é vital porque conhecer apenas a posição dos obstáculos não é suficiente para desviar deles se o agente desconhece a própria inércia. Ao processar a velocidade, o modelo torna-se capaz de aprender o conceito de distância de frenagem, o que preserva a propriedade de Markov e impede que o controle de voo se torne tardio e instável.

A navegação relativa consiste no vetor em direção ao objetivo, que é decomposto

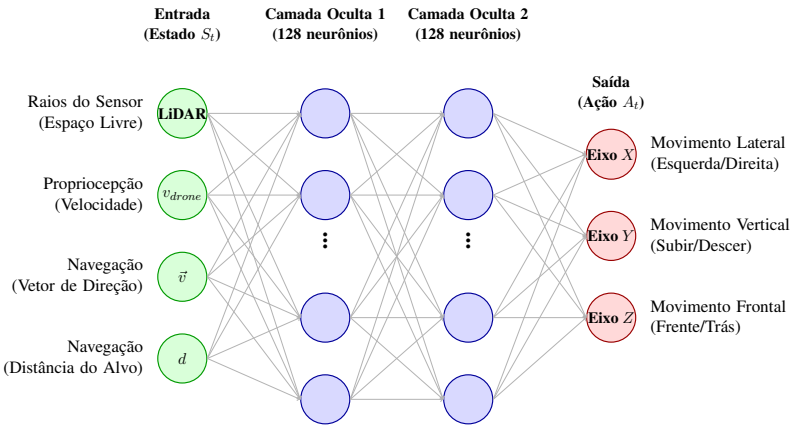


Figura 2. Arquitetura MLP para a política de controle do agente, mapeando o espaço de observação para as ações contínuas de voo tridimensional.

em dois sinais distintos: a direção normalizada, que define um vetor unitário servindo como "bússola", e a magnitude escalar, que define a distância absoluta. Essa separação evita a instabilidade numérica nos gradientes da rede neural, impedindo que valores de magnitude muito grandes (distâncias longas) ofusquem a informação direcional durante a retropropagação.

3.4. Função de Recompensa

A função de recompensa R_t guia o aprendizado do agente, definindo valores positivos para comportamentos desejados e penalidades (recompensas negativas) para falhas ou violações de restrições operacionais. A função composta é definida como:

$$R_t = R_{progresso} + R_{evasao} + R_{terminal} \tag{1}$$

em que $R_{progresso}$ fornece uma recompensa substancial e contínua proporcional à redução da distância ao alvo; R_{evasao} cria um campo de repulsão virtual que penaliza exponencialmente a aproximação de obstáculos, ensinando o agente a manter uma distância de segurança em relação a outros drones detectados; e $R_{terminal}$ representa o encerramento da tentativa de voo.

O campo de repulsão virtual de R_{evasao} é definido como o inverso da distância detectada pelo LiDAR ($\frac{k}{d_{obstaculo}}$). Em $R_{terminal}$, o agente recebe um bônus positivo significativo ao completar a missão, ponderado pelo tempo restante, o que incentiva a eficiência temporal. De maneira diametralmente oposta, o episódio é imediatamente encerrado com penalidades altas em casos de falhas críticas: colisão (com outros drones ou estruturas), violação de limites de altitude (ultrapassar o teto aéreo permitido ou voar abaixo do nível do chão) e a falha em completar a viagem no tempo estipulado (estouro do limite de tempo máximo definido para o voo).

3.5. Treinamento e Aprendizado Curricular

O treinamento foi realizado no ambiente Unity, utilizando aceleração temporal para simular dias em horas. O treinamento de agentes em cenários vastos sofre do problema da *Esparsidade de Recompensas*: se um drone iniciar a uma distância muito grande do

alvo sem possuir conhecimento prévio, a probabilidade de conseguir chegar ao destino aleatoriamente é quase nula, sendo um impedidor para o aprendizado.

A fim de superar essa limitação, adotou-se o Aprendizado Curricular [Bengio et al. 2009], que divide o treinamento em lições gradualmente mais desafiadoras. Alguns parâmetros são definidos como condições necessárias para o treinamento avançar para a próxima lição. Os drones precisam atingir uma certa quantidade de recompensa para conseguir prosseguir. Neste artigo, prosseguir de lição significa aumentar a distância entre os pontos de partida e entrega, dificultando o voo conforme o progresso dos drones. O modelo decide a velocidade e o plano de voo, que varia conforme a lição curricular. Assim, o agente vai descobrindo e aprendendo a traçar seu plano de voo em cada lição [Yin et al. 2024]. Esses planos de voo passam a se interceptar conforme os drones progridem nas lições e, dessa forma, o treinamento também evita colisões entre objetos dinâmicos, que são os drones. Esse processo garante que o agente esteja sempre em sua zona de desenvolvimento proximal, aprendendo e desenvolvendo políticas robustas gradualmente.

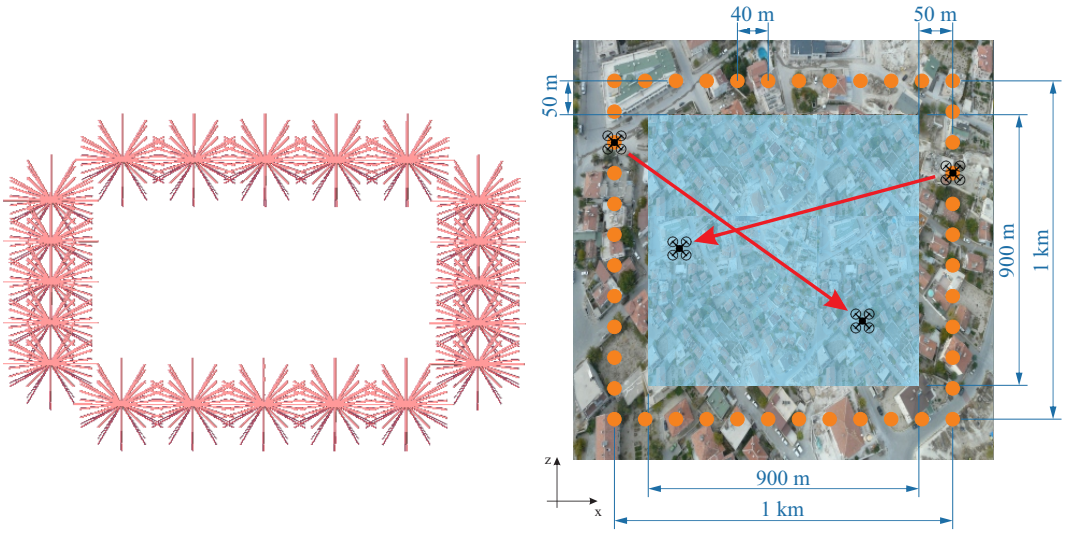
4. Metodologia

Nesta seção, são apresentados os materiais e métodos utilizados para avaliar a proposta, detalhando cada tecnologia empregada e as abordagens desenvolvidas. Para mitigar o sobreajuste (do inglês *overfitting*) e validar a capacidade de generalização do modelo, adota-se uma estratégia de validação Simulador-para-Simulador (*Sim-to-Sim*, do inglês *Simulator-to-Simulator*), uma abordagem que consiste em treinar o agente em um ambiente virtual isolado e, posteriormente, testá-lo em um simulador distinto. Essa separação permite que o agente aprenda a dinâmica de voo e as regras de evasão robustamente, evitando que ele se torne dependente das características específicas de um único simulador. Assim, dois simuladores de drones foram usados, desenvolvidos em linguagem C# usando a Unity, que é uma plataforma para desenvolvimento de aplicações 3D [Unity Technologies 2022]. Eles também empregam a biblioteca ML-Agents [Unity Technologies 2021], que consiste em um *framework* para aplicações que usam Aprendizado de Máquina. O treinamento usa um simulador dedicado totalmente baseado em ML-Agents, enquanto a inferência emprega uma modificação do simulador de drones UTSim [AlMousa et al. 2019] para simular o cenário de um serviço de entregas aéreas, coletar métricas e integrar o ML-Agents.

Neste estudo, tanto o cenário de treinamento quanto o cenário de inferência são livres de obstáculos estáticos, contendo apenas os drones com capacidade de colidirem entre si. Ambos os cenários contam com dronepontos, que são locais dos quais os drones decolam e pousam em centros de distribuição e locais de entrega. O cenário de treinamento é composto por 20 drones, que treinam o modelo simultaneamente e globalmente e seguem um plano de voo contendo decolagem, voo em cruzeiro, pouso da entrega, entrega, decolagem para o retorno, voo de volta em cruzeiro e pouso final. Cada episódio de treinamento, para cada drone, termina conforme o drone falhe ou seja bem-sucedido no objetivo. Falhar significa colidir, perder-se — não atingindo os destinos estabelecidos —, ultrapassar determinada altitude, voar abaixo do nível do chão ou exceder o tempo máximo para a missão, enquanto o sucesso é definido por completar o plano de voo estipulado. As falhas resultam em recompensas negativas para o modelo.

A estratégia pedagógica de Aprendizado Curricular foi definida de maneira a or-

ganizar o treinamento em 90 lições de dificuldade progressiva. Inicialmente, o agente resolve tarefas de navegação local (com distância < 50 m), na qual raramente drones poderiam voar em lugares próximos, o que aumentaria a possibilidade de colisão. Conforme atinge critérios de consistência de recompensa, o algoritmo aumenta linearmente a distância do objetivo e aumenta a densidade de tráfego nas regiões do cenário. O cenário inicial de treinamento com os drones e os pulsos do LiDAR no modelo representando mais pulsos (LiARA-MR) pode ser observado na Figura 3(a). Durante o treinamento, os drones podem realizar entregas em qualquer região dentro da distância definida pela lição atual.



(a) Perspectiva com drones (LiDAR) no simulador de treinamento

(b) Vista superior da inferência [Oliveira et al. 2021]

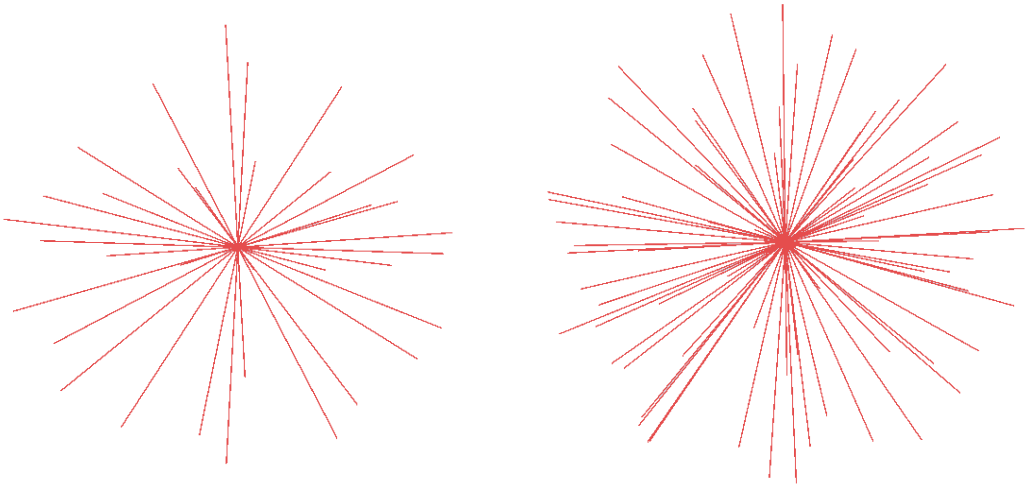
Figura 3. Cenários utilizados.

O cenário da inferência serve para testar o modelo e compará-lo com outras abordagens da literatura. A vista superior do cenário para a inferência pode ser vista na Figura 3(b). Cem dronepontos representam centros de distribuição (CD) de mercadorias e são dispostos ao longo de um quadrado de 1 km de lado a cada 40 m, com o objetivo de reduzir as colisões que ocorrem durante pousos e decolagens e estudar as colisões que ocorrem em cruzeiro, que são mais desafiadoras. As entregas ocorrem em pontos aleatórios dentro do quadrado azul, com uma área de 0,81 km^2, concêntrica ao quadrado com os dronepontos, e separada de cada droneponto dos CDs por pelo menos 50 m. Como o espaço aéreo é livre, a figura mostra que as rotas podem se entrelaçar e gerar colisões.

Os drones usados no treinamento e inferência medem 2 m × 2 m × 0,5 m, possuem uma velocidade máxima de 20 m/s e simulam um sensor LiDAR que capta obstáculos com precisão até 100 m de distância em todos os ângulos definidos para os raios do sensor. A Figura 4 apresenta as diferentes configurações de densidade sensorial exploradas, *i.e.*, o número de feixes de raios laser em ângulos distintos no LiDAR: os modelos de *Leve Resolução* – LiARA-LR – e de *Maior Resolução* – LiARA-MR, para investigar a correlação entre a resolução espacial da percepção e a eficácia na evasão de colisões.

O plano de voo no cenário de inferência ocorre em 10 etapas e acontece da

seguinte maneira e nesta ordem: 1) inicialização; 2) definição da velocidade; 3) decolagem a 30 m de altitude; 4) voo até o destino de entrega que foi definido aleatoriamente dentro do espaço no mapa; 5) pouso à altitude de 1 m; 6) espera de 10 segundos no ponto de entrega para realizá-la; 7) decolagem à altitude de 30 m; 8) voo até o ponto inicial sobre o droneponto; 9) pouso à altitude de 1 m; e 10) finalização. Da mesma forma que os métodos determinísticos podem definir diferentes velocidades para cada etapa e situação de voo, o modelo de RL decide a velocidade na inferência.



(a) *Leve Resolução* – LiARA-LR (b) *Maior Resolução* – LiARA-MR

Figura 4. Diferentes configurações de densidade sensorial para o LiDAR dos drones.

Na inferência, os drones são lançados a diferentes taxas de chegada, seguindo uma distribuição de Poisson, que varia de 2 a 12 drones/min. O tempo de simulação foi de 47 min e a aceleração, de três vezes. Cada simulação, baseada em um conjunto específico de características, é executada 30 vezes para considerar as variações estatísticas inerentes ao cenário e métodos de desvio. A taxa de colisões é reportada por sua média e intervalo de confiança de 95%. No treinamento, foram utilizadas algumas métricas de RL para indicar a qualidade do treinamento e mostrar que não há sobreajuste (*overfitting*), como a recompensa acumulada, a entropia, a perda de política e a perda de valor.

As estratégias comparadas com os modelos propostos neste artigo (LiARA-LR e LiARA-MR) são FazerNada, algoritmos geométricos da literatura e um modelo de RL de base, cujos resultados foram retirados diretamente da literatura [Oliveira et al. 2024]. Em FazerNada, o drone não desvia, mesmo que isso cause colisão. As abordagens geométricas são a proposta por Seo et al. [2017], aqui chamada *SeoKimKimTsourdos* (SKKT), Vector Sharing Resolution (VSR) [Park et al. 2008] e as abordagens GeoDrone (SingleDrone, MultiDrone e SpeedDrone) [Oliveira et al. 2024]. O modelo de RL é o Neural Autonomous UAV (NAV) [Barros et al. 2026].

5. Resultados

Esta seção apresenta os resultados obtidos no cenário do serviço de entregas no simulador de inferência e os resultados do treinamento no simulador de treinamento.

5.1. Resultados de Colisão no Simulador de Inferência

A Figura 5 compara diferentes estratégias de desvio, nas quais DoNothing significa Fazer-Nada, apresentando a média e intervalo de confiança de 95% das taxas de colisão obtidas em cada abordagem. Os melhores métodos de desvios usam RL e o LiARA-MR é o modelo mais bem-sucedido entre todos os métodos testados, com $(0,43 \pm 0,15)\%$ de colisões a uma taxa de 12 drones por minuto. O segundo melhor modelo é o LiARA-LR, também proposto neste artigo, com uma taxa de colisão de $(0,87 \pm 0,23)\%$, e o terceiro é o NAV, com $(1,77 \pm 0,21)\%$. SingleDrone é a melhor estratégia geométrica, apresentando $(7,08 \pm 0,48)\%$ e sendo quase 24 vezes pior que o melhor modelo de RL.

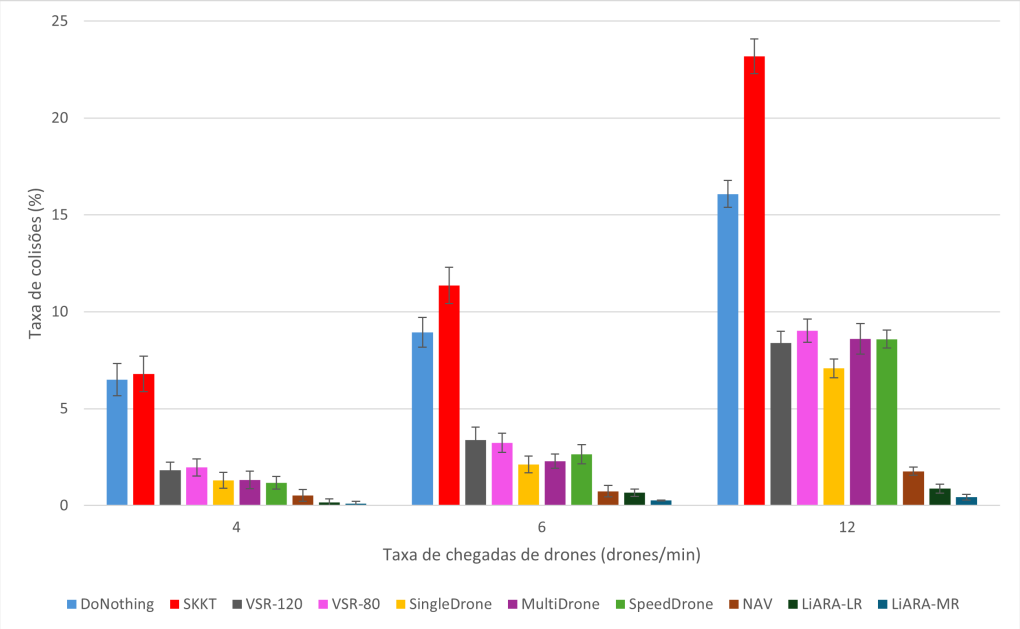


Figura 5. Taxa de colisões.

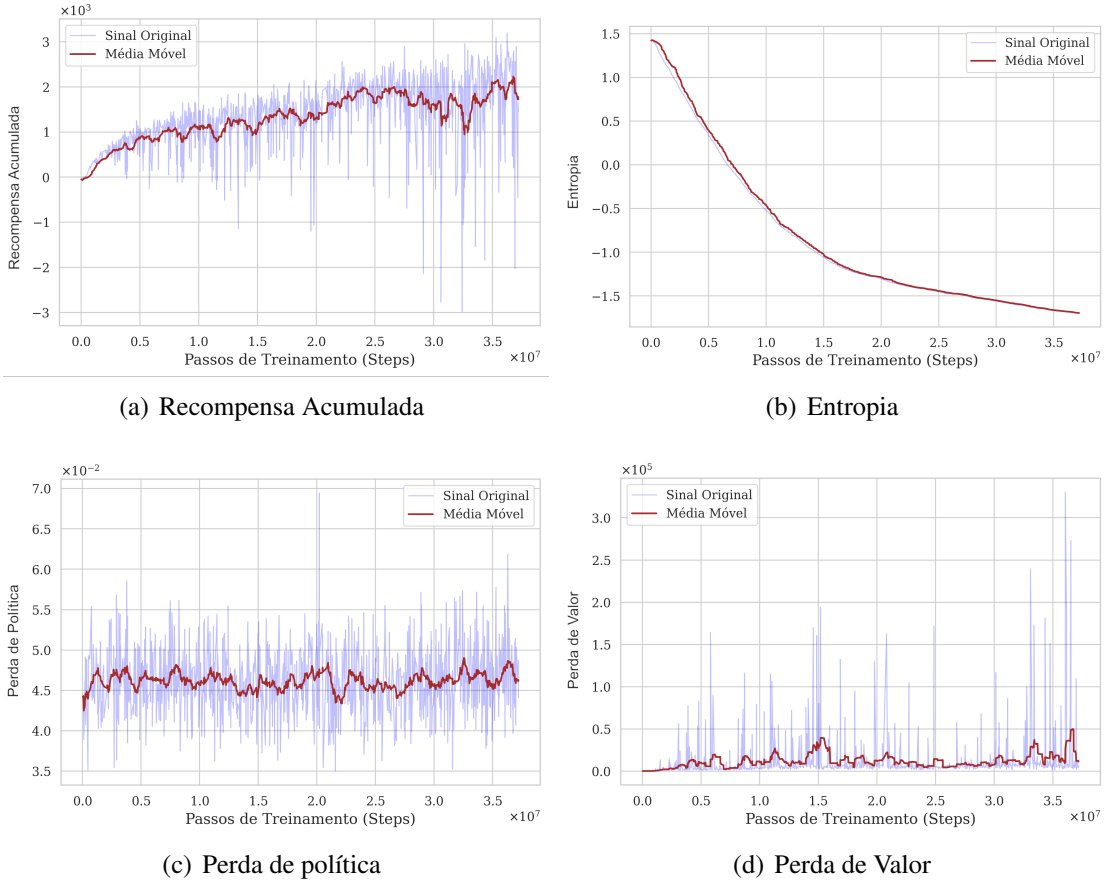
Estes resultados demonstram a limitação das técnicas tradicionais de desvio e apresentam uma melhora bastante significativa nos resultados em reduzir as colisões entre drones, o qual é o principal objetivo deste artigo. Eles indicam um caminho promissor em direção a uma operação segura com drones, assim como permitem seguir dificultando e tornando o cenário mais realista, como adicionar prédios e mais obstáculos presentes no mundo real.

5.2. Qualidade do treinamento

A avaliação da qualidade do treinamento é fundamental para atestar a estabilidade e a convergência do algoritmo de Aprendizado por Reforço. Os gráficos apresentados na Figura 6 detalham a evolução das métricas internas da rede neural ao longo de aproximadamente 37 milhões de passos de simulação para o LiARA-MR. Em todos os gráficos, o eixo horizontal representa o tempo global de treinamento, enquanto a linha mais escura ilustra a média móvel das métricas, facilitando a visualização da tendência geral do aprendizado do agente em meio à variação natural inerente a um ambiente estocástico.

A recompensa acumulada, ilustrada na Figura 6(a), quantifica a soma de todas as recompensas obtidas pelo agente durante cada passo de treinamento. Geralmente, ela é o

principal indicador de sucesso do modelo, pois reflete a capacidade da rede de maximizar a função de recompensa estipulada. No contexto desta pesquisa, o crescimento contínuo e a posterior estabilização da curva em valores elevados comprovam que o drone deixou de agir de forma irregular, minimizando colisões e aprendendo a alcançar os alvos com eficiência. A variação constante observada no sinal original é uma consequência direta do aprendizado curricular, uma vez que cada lição altera a distância do alvo e, consequentemente, gera rotas que se cruzam entre diferentes drones, aumentando a dificuldade e resultando em pontuações absolutas distintas dependendo de cada episódio.



(a) Recompensa Acumulada

(b) Entropia

(c) Perda de política

(d) Perda de Valor

Figura 6. Resultados do treinamento.

A entropia, apresentada na Figura 6(b), mede o nível de incerteza ou aleatoriedade nas ações tomadas pela rede neural a cada passo de treinamento. Valores altos no início refletem a fase de exploração, em que o agente testa diversos movimentos para compreender as dinâmicas de voo e as punições do ambiente. O gráfico demonstra um decaimento contínuo até atingir valores negativos, o que, na prática, indica uma transição bem-sucedida para a fase de explotação. Isso significa que, ao final do treinamento, o agente desenvolveu uma alta confiança em suas manobras evasivas baseadas nas leituras espaciais, não precisando mais recorrer a ações aleatórias para evitar obstáculos.

A perda de política, mostrada na Figura 6(c), avalia a intensidade das atualizações realizadas nos pesos da rede neural responsável por ditar as ações de controle do drone a cada passo de treinamento. O PPO busca manter essa métrica sob controle para evitar

mudanças de comportamento abruptas que poderiam arruinar o aprendizado prévio. A oscilação limitada da curva ao longo de todo o treinamento evidencia que as configurações de hiperparâmetros foram adequadas para o problema. O agente atualizou suas manobras de forma constante e segura, refinando a capacidade de desvio gradativamente sem sofrer episódios de esquecimento catastrófico das regras de navegação já consolidadas.

A perda de valor, mostrada na Figura 6(d), mensura o erro da rede neural ao prever a recompensa a partir de cada estado por passo de treinamento. Em um treinamento estático, valores cronicamente altos indicariam falha de convergência, mas, neste artigo, os picos agudos e periódicos representam a assinatura empírica do aprendizado curricular. Cada pico ocorre no exato momento em que o ambiente evolui para uma lição mais difícil, surpreendendo o agente com pontos de entrega mais distantes e maior tráfego, o que causa um erro temporário em sua previsão de sucesso. A rápida queda do erro imediatamente após cada pico sugere que o modelo de aprendizado por reforço conseguiu se adaptar velozmente a cada novo nível de complexidade introduzido.

6. Discussão

Os resultados obtidos no experimento indicam que algoritmos de Aprendizado por Reforço estruturados por meio de um currículo pedagógico e baseados em percepção espacial local superam significativamente as abordagens geométricas e o modelo de RL de base em cenários com alta densidade de obstáculos móveis. A expressiva redução da taxa de colisões do LiARA-MR para $(0,43 \pm 0,15)\%$, frente aos $(7,08 \pm 0,48)\%$ do melhor método determinístico na taxa de 12 drones por minuto, evidencia a limitação reativa das técnicas tradicionais, que frequentemente colapsam ao lidar com múltiplos obstáculos móveis. Em contraste, a rede treinada via PPO aprendeu uma representação clara de risco de colisão, antecipando manobras de forma fluida ao integrar posição do agente e distância ao ponto de entrega.

A superioridade do LiARA-MR sobre o LiARA-LR evidencia a mitigação do ofuscamento espacial: sensores mais densos reduzem pontos cegos e são decisivos para a navegação segura em ambientes congestionados. Além disso, a estratégia de Aprendizado Curricular foi essencial para a convergência do modelo, cujas lições progressivas suavizaram a grande variação de recompensas entre ações e permitiram que o agente aprendesse evasão mútua de forma gradual antes de enfrentar cenários críticos.

O custo computacional dos modelos propostos é baixo, na ordem de dezenas de milhares de operações de ponto-flutuante (FLOP) por inferência, o que significa que o impacto no tempo de resposta é mínimo e compatível com taxas de atualização como 40 Hz em um drone. Essa carga de processamento ocupa apenas uma fração pequena da capacidade de uma CPU embarcada típica, deixando margem para outras tarefas críticas, como fusão sensorial. Consequentemente, o consumo de energia adicional associado à execução contínua desse modelo tende a ser modesto, contribuindo pouco para o orçamento total de energia do sistema e permitindo seu uso em aplicações em tempo real sem comprometer significativamente a autonomia do drone.

A manutenção da robustez estatística do agente ao ser transferido do ambiente de treinamento para o simulador de inferência (versão modificada do UTSim) com dinâmicas de voo distintas atesta o sucesso da validação Sim-to-Sim, demonstrando que o modelo generalizou uma heurística de evasão em vez de sofrer sobreajuste. Con-

tudo, a transferência direta dessas políticas para o mundo físico (Sim-to-Real) esbarra em simplificações adotadas no simulador: o ambiente assume leituras perfeitas do LiDAR e ignora perturbações externas, como rajadas de vento, oclusões e oscilações de bateria. Para transpor essa lacuna de realidade, iterações futuras demandarão a aplicação de randomização de domínio, introduzindo ruído sensorial gaussiano, latência estocástica e obstáculos estáticos, a fim de forçar o modelo a operar sob as imperfeições intrínsecas dos cenários urbanos.

7. Conclusão

Este artigo apresenta uma abordagem não cooperativa de Aprendizado por Reforço (RL) para evasão robusta de colisões em drones. O problema é modelado como um Processo de Decisão de Markov Parcialmente Observável, cuja política é treinada com Proximal Policy Optimization e Aprendizado Curricular, utilizando observações cinéticas e varreduras de sensores LiDAR simulados. A capacidade de generalização do modelo é avaliada por meio de validação Simulador-para-Simulador, na qual agentes treinados em um simulador mantêm desempenho elevado em um ambiente logístico distinto.

Em cenários críticos de congestionamento, o modelo de maior resolução (LiARA-MR) reduziu a taxa de colisões para 0,43%, superando a abordagem geométrica do estado-da-arte ($7,0\%$) e o modelo de RL de base ($1,77\%$). Os resultados atestam que a maior fidelidade sensorial, combinada ao treinamento progressivo, é determinante para a segurança em espaços aéreos densos e reforça o potencial de escalabilidade da solução para operações logísticas reais.

Como trabalhos futuros, planeja-se a introdução de obstáculos urbanos 3D e altitudes dinâmicas. Ao ser inserida em cenários realistas onde drones possuam comunicação mútua (V2V), a política não-cooperativa aqui proposta funcionará como uma camada vital de redundância reativa de curto alcance, assumindo o controle para garantir evasões de emergência caso ocorram falhas de rede ou latência na coordenação primária.

8. Agradecimentos

Esta pesquisa é parte do INCT de Redes de Comunicação e Internet das Coisas Inteligentes (ICoNIoT), financiado por CNPq (proc. 405940/2022-0 e 197179/2025-8) e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 88887.954253/2024-00, e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo 2021/00199-8, CPE SMARTNESS.

Referências

AlMousa, A., B. H. Sababha, N. Al-Madi, A. Barghouthi, e R. Younisse (2019). UTSim: A framework and simulator for UAV air traffic integration, control, and communication. In: vol. 16. 5, pp. 1–19.

Barros, I. d. S., F. M. C. de Oliveira, L. F. Bittencourt, e C. Kamienski (2026). Collision-Aware UAV Delivery via Learning-Based Autonomous Control. *SSRN Electronic Journal*. *Em revisão.*

Bengio, Y., J. Louradour, R. Collobert, e J. Weston (2009). Curriculum learning. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, pp. 41–48.

Jang, K., Y. Pant, A. Rodionova, e R. Mangharam (Oct. 2020). Learning-to-Fly RL: Reinforcement Learning-based Collision Avoidance for Scalable Urban Air Mobility. In: pp. 1–10.

Jeon, A., J. Kang, B. Choi, N. Kim, J. Eun, e T. Cheong (2021). Unmanned Aerial Vehicle Last-Mile Delivery Considering Backhauls. In: vol. 9. IEEE, pp. 85017–85033.

Kong, F., J. Li, B. Jiang, H. Wang, e H. Song (2023). Trajectory Optimization for Drone Logistics Delivery via Attention-Based Pointer Network. In: vol. 24. 4, pp. 4519–4531.

Lu, L., G. Fasano, A. Carrio, M. Lei, H. Bavle, e P. Campoy (2023). A comprehensive survey on non-cooperative collision avoidance for micro aerial vehicles: Sensing and obstacle detection. In: n/a.

Oliveira, F. M. C. de, L. Bittencourt, R. Bianchi, e C. Kamienski (2022). Drones na Cidade Grande: Reduzindo Colisões em Entregas Aéreas. In: *Anais do VI Workshop de Computação Urbana*. Fortaleza: SBC, pp. 84–97.

Oliveira, F. M. C. de, L. Bittencourt, e C. Kamienski (2021). Prevenção de Colisões em Serviços de Entregas por Drones em Cidades Inteligentes. In: *Anais do V Workshop de Computação Urbana*. Uberlândia: SBC, pp. 182–195.

Oliveira, F. M. C. de, L. F. Bittencourt, R. A. C. Bianchi, e C. A. Kamienski (2024). Drones in the Big City: Autonomous Collision Avoidance for Aerial Delivery Services. *IEEE Trans. Intell. Transp. Syst.* 25:5, pp. 4657–4674.

Park, J.-W., H.-D. Oh, e M.-J. Tahk (2008). UAV collision avoidance based on geometric approach. In: *2008 SICE Annual Conference*, pp. 2122–2126.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, e O. Klimov (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Seo, J., Y. Kim, S. Kim, e A. Tsourdos (2017). Collision Avoidance Strategies for Unmanned Aerial Vehicles in Formation Flight. In: vol. 53. 6, pp. 2718–2734.

Unity Technologies (2021). *Unity Machine Learning Agents (ML-Agents)*. Version 2.0.1. Acessado em 19 de março de 2026. San Francisco, CA.

Unity Technologies (2022). *Unity*. Version 2022.3.22f1. Game development and 3D simulation platform. Acessado em 19 de março de 2026. San Francisco, CA.

Whiteson, S., B. Tanner, M. E. Taylor, e P. Stone (2011). Protecting against evaluation overfitting in empirical reinforcement learning. In: *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 120–127.

Wu, Y., K. H. Low, B. Pang, e Q. Tan (2021). Swarm-Based 4D Path Planning For Drone Operations in Urban Environments. In: vol. 70. 8, pp. 7464–7479.

Yasin, J., S. A.S. Mohamed, H. Haghbayan, J. Heikkonen, H. Tenhunen, e J. Plosila (June 2020). Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches. In: vol. 8, pp. 105139–105155.

Yin, H., S. Su, Y. Lin, et al. (June 17, 2024). Multi-AGV Path Planning Using Deep Reinforcement Learning with Internal Curiosity. In: Preprint.

Zhang, J., H. Zhang, J. Zhou, M. Hua, G. Zhong, e H. Liu (2023). Adaptive Collision Avoidance for Multiple UAVs in Urban Environments. In: vol. 7. 8.

Zhao, P., H. Erzberger, e Y. Liu (2021). Multiple-aircraft-conflict resolution under uncertainties. In: vol. 44. 11. American Institute of Aeronautics e Astronautics, pp. 2031–2049.