



UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

FILIPPE SILVA DE JESUS

NAVEGAÇÃO AUTÔNOMA PARA VEÍCULOS AÉREOS NÃO TRIPULADOS
SIMULADOS EM AMBIENTES DESCONHECIDOS: PLANEJAMENTO DE
CAMINHOS ATRAVÉS DO USO DA APRENDIZAGEM POR REFORÇO

SALVADOR, BAHIA, BRASIL

2023

FILIPPE SILVA DE JESUS

NAVEGAÇÃO AUTÔNOMA PARA VEÍCULOS AÉREOS NÃO TRIPULADOS
SIMULADOS EM AMBIENTES DESCONHECIDOS: PLANEJAMENTO DE CAMINHOS
ATRAVÉS DO USO DA APRENDIZAGEM POR REFORÇO

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia - UNEB, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Orientador: Marco Antônio Costa Simões

Co-Orientador: Diego Gervásio Frias Suarez

SALVADOR, BAHIA, BRASIL

2023

FILIPPE SILVA DE JESUS

**NAVEGAÇÃO AUTONOMA PARA VEÍCULOS AEREOS NÃO TRIPULADOS
SIMULADO EM AMBIENTES DESCONHECIDOS: PLANEJAMENTO DE
CAMINHOS ATRAVÉS DO USO DA APREDIZAGEM POR REFORÇO**

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia - UNEB, como requisito à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Aprovado em: 14/12/2023

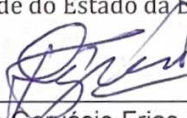
BANCA EXAMINADORA



Marco Antônio Costa Simões (Orientador)
Universidade do Estado da Bahia - UNEB



Ana Patrícia Fontes Magalhães Mascarenhas
Universidade do Estado da Bahia - UNEB



Diego Gervásio Frias Suarez
Universidade do Estado da Bahia - UNEB

Dedico este trabalho a minha mãe Anaci Sena,
meu porto seguro, maior inspiração de vida e
que todos os dias me incentiva a atingir meus
objetivos.

AGRADECIMENTOS

Alcançar o ponto final desta jornada, marcada por desafios superados, descobertas enriquecedoras e uma abundância de aprendizado, é verdadeiramente emocionante e gratificante sob diversos aspectos. A entrega deste Trabalho de Conclusão de Curso é um marco que eu não poderia ter alcançado sem o apoio de diversas pessoas e o esforço incansável de cada um de nós.

Primeiramente eu agradeço a Deus por ter me dado força e empenho para superar as dificuldades. Sua orientação silenciosa e divina foi uma luz constante ao longo deste percurso desafiador. Que sua sabedoria continue a me guiar nas futuras jornadas e que eu possa continuar a honrar sua bênção através do meu esforço e dedicação.

Agradeço também à minha família, em especial minha mãe Anaci Sena, ao meu irmão Lucas Silva, às minhas tias Rosani Sena e Lucineia Gonçalves, e aos meus primos Carolaine Gonçalves e Matheus Rosa por estarem juntos, me apoiando e motivando cada dia mais. Também não poderia esquecer da minha companheira Ludmila Brito e dos meus amigos Tailane Pereira, Marcos Vinicius e Natanel Alan, cujo apoio inabalável foi uma das minhas fontes de incentivo e motivação. Vocês estiveram ao meu lado nos momentos de ansiedade e compartilhando as alegrias das pequenas vitórias. Sem o amor e o suporte de vocês, essa conquista não teria sido a mesma.

Gostaria de expressar minha sincera gratidão ao meu orientador Marco Simões. Suas orientações valiosas ao longo de exatos três anos de convivência moldaram o meu trabalho e me guiaram ao longo deste caminho acadêmico. Suas experiências e conselhos foram essenciais, especialmente quanto as conquistas que tivemos juntos nos campeonatos em que participamos. Trabalhar ao seu lado e alcançar o sucesso em competições desafiadoras foi uma experiência enriquecedora que moldou profundamente minha jornada acadêmica. Além disso, gostaria de agradecer ao meu co-orientador, Diego Frias por suas propostas de melhorias ao longo deste projeto, que contribuíram significativamente para aprimorar a qualidade do trabalho. Sua ação incisiva e colaboração foram essenciais para o avanço deste projeto. Agradeço pelo apoio contínuo de vocês e pelas valiosas contribuições.

Não posso esquecer de mencionar meus colegas de turma Jéssica Rocha, Vitor Manoel e Reinilson Bispo. Juntos em todos os trabalhos em grupo desde o primeiro semestre, enfrentamos desafios acadêmicos, madrugamos programando, trocamos experiências e crescemos como profissionais.

O companheirismo e a colaboração entre nós foi fundamental para superar obstáculos e alcançar o sucesso. Equipe nota 10 em Amorim!

Obrigado a todos que tornaram este dia possível.

"N3o existe um caminho para a felicidade. A
felicidade 3 o caminho."

(Thich Nhat Hanh)

RESUMO

Com o avanço da tecnologia e o aumento do uso de Veículos Aéreos Não Tripulados (VANTs) em diversas aplicações, como inspeções de infraestruturas, monitoramento ambiental e entregas de mercadorias, torna-se fundamental o desenvolvimento de técnicas avançadas para o planejamento de caminhos dessas aeronaves. Trabalhos relacionados na área têm explorado abordagens de aprendizagem por reforço para o planejamento de caminhos em ambientes 2D, porém, ainda existem lacunas a serem preenchidas em relação a ambientes 3D e à inclusão de obstáculos complexos. Nessa perspectiva, este estudo propõe a integração do algoritmo de busca A* no módulo de planejamento de caminhos, comparando com técnicas de navegação que utilizam a aprendizagem por reforço. Além disso, para aprimorar a percepção do agente no ambiente tridimensional, é proposta uma modelagem que converte dados de um sensor 2D em uma representação abstrata 3D via software, permitindo ao VANT obter uma compreensão mais abrangente e precisa do cenário. Os experimentos revelaram um significativo aumento na taxa de sucesso da navegação autônoma, indo de 29% para 76% com a utilização da modelagem tridimensional dos obstáculos. Além disso, demonstramos que o VANT, treinado por meio de aprendizado por reforço, adquiriu habilidades fundamentais de navegação. Esses avanços visam preencher lacunas e, conseqüentemente, contribuir para a automação e eficiência das operações dessas aeronaves.

Palavras-chave: Planejamento de Caminhos. Veículos Aéreos Não Tripulados. Aprendizagem por Reforço. Multirotor

ABSTRACT

With the advancement of technology and the increasing use of Unmanned Aerial Vehicles (UAVs) in various applications, such as infrastructure inspections, environmental monitoring, and goods delivery, the development of advanced techniques for path planning of these aircraft becomes essential. Related works in the field have explored reinforcement learning approaches for path planning in 2D environments; however, there are still gaps to be filled regarding 3D environments and the inclusion of complex obstacles. In this perspective, this study proposes the integration of the A* search algorithm into the path planning module, comparing it with navigation techniques using reinforcement learning. Additionally, to enhance the agent's perception in the three-dimensional environment, a modeling approach is proposed that converts data from a 2D sensor into a 3D abstract representation via software, allowing the UAV to obtain a more comprehensive and accurate understanding of the scenario. The experiments revealed a significant increase in the success rate of autonomous navigation, rising from 29% to 76% with the utilization of 3D modeling of obstacles. Furthermore, we demonstrated that the UAV, trained through reinforcement learning, acquired fundamental navigation skills. These advancements aim to fill gaps and, consequently, contribute to the automation and efficiency of the operations of these aircraft.

Palavras-chave: Path Planning. Unmanned Aerial Vehicles. Reinforcement Learning. Multirotor

LISTA DE FIGURAS

Figura 1 – Arquitetura do <i>Robot Operating System</i> (ROS)	25
Figura 2 – O A-Estrela (A*) garante que os nós com menor custo total sejam priorizados na expansão, caso a função heurística seja admissível.	31
Figura 3 – A interação entre os componentes centrais no aprendizado por reforço . . .	32
Figura 4 – O algoritmo <i>Deep Q-Network</i> (DQN) é composto por três componentes principais: a <i>Q-Network</i> ($Q(s, a; \theta)$), que define a política de comportamento, a <i>Target Q-Network</i> ($Q(s, a; \theta^-)$), usada para gerar valores Q de destino para o termo de perda do DQN, e a memória de <i>replay</i> que o agente utiliza para amostrar transições aleatórias para treinar a <i>Q-Network</i>	36
Figura 5 – Modelo apresentado por Bouhamed et al.(1), com o caminho seguido pelo VANT com base no local de destino estático. Os pontos vermelhos referem-se ao VANT, pontos verdes referem-se ao alvo e caixas cinzas aos obstáculos. .	38
Figura 6 – Resultados da navegação autônoma dos Veículos Aéreos Não Tripulados (VANTs) no trabalho apresentado por Guerra et al.(2). Esquerda: alvo colocado no canto inferior direito. Direita: alvo colocado no canto inferior esquerdo	39
Figura 7 – Exemplos de trajetórias estimadas e mapas da abordagem realizada por Guerra et al.(3). Os marcadores azuis e vermelhos indicam p_0 e a posição de destino, respectivamente.	40
Figura 8 – Resultados obtidos por Qu et al.(4) comparando o <i>Reinforcement Learning based Grey Wolf Optimizer</i> (RLGWO) com técnicas similares de aprendizagem por reforço	41
Figura 9 – Árvore utilizada como obstáculo para navegação do VANT	45
Figura 10 – Cenário base	46
Figura 11 – Cenário de Teste 1	47
Figura 12 – Cenário de Teste 2	48
Figura 13 – Cenário de Teste 3	49
Figura 14 – Exemplo de parte do arquivo gerado pelo módulo de coleta das coordenadas do VANT	51

Figura 15 – DJI F450 simulado (MRS UAV System)	54
Figura 16 – Pontos de um semicírculo gerado pelos dados do RPLidar	59
Figura 17 – Ajuste do círculo a partir do semicírculo gerado pelos dados do RPLidar . .	60
Figura 18 – Pontos do círculo ajustado	63
Figura 19 – Projeção da esfera a partir do círculo ajustado	64
Figura 20 – Percepção 3D da árvore	65
Figura 21 – Percepção 3D em um ambiente contendo múltiplas árvores	66
Figura 22 – Arquitetura do sistema ROS do com sensores, módulos de iteração com o VANT e o módulo de navegação	73
Figura 23 – Máquina de Estados para o Planejamento de Caminhos Clássico	74
Figura 24 – Cenário simulado no Gazebo Simulator à esquerda e percepção do agente, bem como o caminho gerado (em vermelho) pelo Planejador de Caminhos. .	75
Figura 25 – Arquitetura do modelo proposto de aprendizagem por reforço.	76
Figura 26 – Taxa de Sucesso dos Experimentos da Abordagem Clássica	84
Figura 27 – Média das métricas nos 15 experimentos realizados	85
Figura 28 – Impacto das médias das métricas na distribuição dos 15 experimentos realizados	85
Figura 29 – Taxa de Sucesso nos Experimentos da Abordagem Clássica sem a Modelagem Tridimensional dos Obstáculos.	88
Figura 30 – Caminho encontrado colide com as árvores devido à percepção limitada que o agente tem sobre o ambiente, que neste caso, é composto apenas por uma visão planar e bidimensional dos obstáculos.	89
Figura 31 – Gráfico da evolução das métricas por steps durante o treinamento do agente de aprendizagem por reforço.	92
Figura 32 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 1. .	105
Figura 33 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 2. .	106
Figura 34 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 4. .	106
Figura 35 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 5. .	107
Figura 36 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 6. .	107
Figura 37 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 7. .	108
Figura 38 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 8. .	108
Figura 39 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 9. .	109
Figura 40 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 10.	109

Figura 41 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 11.	110
Figura 42 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 12.	110
Figura 43 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 13.	110
Figura 44 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 15.	111
Figura 45 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 15.	111
Figura 46 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 1.	112
Figura 47 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 2.	113
Figura 48 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 4.	113
Figura 49 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 6.	114
Figura 50 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 7.	114
Figura 51 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 8.	115

LISTA DE TABELAS

Tabela 1 – Metodologia Comparativa: Planejamento de caminhos com A* e aprendizagem por reforço	44
Tabela 2 – Experimentos de navegação	50
Tabela 3 – Parâmetros das características físicas do F450	55
Tabela 4 – Exemplos de conversão de pontos contínuos para discretos com diferentes resoluções de grade	70
Tabela 5 – Descrição do espaço de observação do ambiente	77
Tabela 6 – Descrição do espaço de observação do ambiente	80
Tabela 7 – Resultados Obtidos dos Experimentos Clássicos	82
Tabela 8 – Resultados Obtidos dos Experimentos Clássicos sem a Modelagem das Percepções em Três Dimensões	87
Tabela 9 – Comparativo da Taxa de Sucesso da Abordagem Clássica Com e Sem a Modelagem Tridimensional dos Obstáculos	90

LISTA DE ABREVIATURAS E SIGLAS

A*	A-Estrela
BFS	<i>Breadth First Search</i>
CDR	Taxa de Detecção Correta
CSV	<i>Comma Separated Values</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DBSCAN	<i>Density-Based Spatial Clustering of Applications with Noise</i>
DDPG	<i>Deep Deterministic Policy Gradient</i>
DDQN	<i>Double Q-Learning</i>
DQN	<i>Deep Q-Network</i>
DRL	<i>Deep Reinforcement Learning</i>
EIRP	Potência Efetiva Irradiada Isotropicamente
FAR	Taxa de Alarme Falso
GPS	<i>Global Positioning System</i>
GWO	<i>Grey Wolf Optimizer</i>
L-BFGS-B	<i>Limited-memory Broyden-Fletcher-Goldfarb-Shanno Bound</i>
MBZIRC	<i>Mohamed Bin Zayed International Robotics Challenge</i>
MDP	<i>Markov Decision Process</i>
MRS UAV	<i>Multi-robot Systems Group UAV System</i>
PPO	<i>Proximal Policy Optimization</i>
PRM	<i>Probabilistic Roadmaps</i>
RL	<i>Reinforcement Learning</i>
RLGWO	<i>Reinforcement Learning based Grey Wolf Optimizer</i>
ROS	<i>Robot Operating System</i>
SubT	<i>Subterranean Challenge</i>
VANTs	Veículos Aéreos Não Tripulados

LISTA DE SÍMBOLOS

T	Alvo
S	Posição Inicial
μ	Média
σ	Desvio Padrão

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Objetivos	20
1.2	Justificativas e Contribuições	21
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Sistema Operacional de Robôs - ROS	24
2.2	Gazebo	26
2.3	MRS UAV System	27
2.4	OpenAI Gym	27
2.5	Stable Baselines3	28
2.6	Métodos Clássicos	29
2.6.1	A* (A-Estrela)	29
2.7	Aprendizagem por Reforço	31
2.7.1	Agente	33
2.7.2	Ações	33
2.7.3	Recompensa	33
2.7.4	Ambiente	34
2.7.5	Observação	34
2.7.6	Algoritmos e Técnicas de Aprendizagem por Reforço	34
2.7.6.1	<i>Deep Q-Network</i> (DQN)	35
2.8	Trabalhos Relacionados	36
2.8.1	Bouhamed - Autonomous UAV Navigation: A DDPG-Based Deep Reinforcement Learning Approach	37
2.8.2	Guerra - Reinforcement Learning for UAV Autonomous Navigation, Mapping and Target Detection	38
2.8.3	Guerra - Real-Time Learning for THZ Radar Mapping and UAV Control	39
2.8.4	Chengzhi - A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning	41
3	METODOLOGIA	43

3.1	Plano de Testes	44
3.1.1	Cenários de Testes	44
3.1.2	Experimentos	49
3.1.3	Coleta da Posição do VANT	51
3.1.4	Métricas para avaliar os testes	51
3.2	Materiais de Desenvolvimento	52
3.3	F450	53
3.3.1	Percepções	55
4	FS PATH PLANNING	57
4.1	Modelagem de percepções 2D em 3D	57
4.1.1	Segmentação dos obstáculos	65
4.2	Abordagem Clássica - A*	68
4.2.1	Grade Discreta	69
4.2.2	A*, C++ e Python	70
4.2.3	Heurística	71
4.2.4	Custo de Movimento	72
4.2.5	Planejador de Caminhos	73
4.3	Abordagem Moderna - Aprendizagem por Reforço	75
4.3.1	Espaço de Ação	76
4.3.2	Espaço de Observação	77
4.3.3	Estado Inicial	77
4.3.4	Fim do Episódio	78
4.3.5	Função de Recompensa	78
4.3.6	Treinamento	79
5	EXPERIMENTOS E AVALIAÇÕES	81
5.1	Abordagem Clássica	81
5.2	Abordagem Clássica sem a Modelagem Tridimensional dos Obstáculos	86
5.3	Abordagem Moderna - Aprendizagem por Reforço	90
6	CONSIDERAÇÕES FINAIS	93
	REFERÊNCIAS	95
	APÊNDICES	100
	APÊNDICE A – Implementação da Grade Discreta	101

APÊNDICE B – Módulo para Análise das Estatísticas Apartir dos Caminhos	102
APÊNDICE C – Módulo para Plotagem 3D dos Caminhos	104
APÊNDICE D – Caminhos dos Experimentos da Abordagem Clássica . . .	105
APÊNDICE E – Caminhos dos Experimentos da Abordagem Clássica sem a Modelagem 3D dos Obstáculos	112

1 INTRODUÇÃO

Nos últimos anos, os Veículos Aéreos Não Tripulados (VANTs) tem passado por um aumento significativo em sua utilização, abrangendo diversas aplicações em diferentes setores. Especificamente, têm sido amplamente empregados em missões militares, onde desempenham papéis na coleta de informações, monitoramento de áreas de conflito e até mesmo no fornecimento de suporte tático em operações especiais (5, 6). Além disso, essas aeronaves não tripuladas estão sendo cada vez mais utilizadas no setor de logística, proporcionando serviços de entregas mais eficientes e rápidos (7). O uso de VANTs para entregas permite alcançar áreas remotas, evitar congestionamentos de tráfego e agilizar a distribuição de mercadorias em geral (7).

Outro campo de aplicação promissor é a detecção e combate a incêndios florestais (8), em que os VANTs equipados com sensores térmicos e câmeras de alta resolução podem sobrevoar extensas áreas florestais, identificar pontos de calor e alertar as equipes de combate terrestres para uma resposta rápida e precisa (9). Além disso, essas aeronaves são utilizadas em missões de busca e salvamento, sendo capazes de cobrir grandes áreas e acessar locais de difícil acesso com facilidade, auxiliando nas operações de resgate de pessoas em situações de emergência (9).

Um componente importante no processo de automação e sucesso das aplicações é o módulo de planejamento de caminhos, responsável por determinar a trajetória a ser seguida pela aeronave. Durante o planejamento, diversos desafios podem ser encontrados. Um deles é a presença de obstáculos móveis ou fixos no ambiente, como árvores, edifícios, veículos ou outras aeronaves, que devem ser evitados para garantir a segurança das operações (10). Além disso, existem restrições a serem consideradas, como o comprimento do caminho e o consumo de energia, que afetam diretamente a autonomia e a capacidade operacional dos VANTs.

Outro desafio no planejamento de caminhos para VANTs é a necessidade de otimizar a eficiência da missão. Isso envolve a definição de rotas que minimizem o tempo de voo. Para atingir esse objetivo, são empregadas técnicas de otimização, como algoritmos baseados em busca heurística. Essas abordagens levam em consideração diversos fatores, como a velocidade, as restrições operacionais do VANT e as metas específicas da missão.

Além disso, é importante mencionar a necessidade de integração com sistemas de sensoriamento e monitoramento em tempo real. Os VANTs são geralmente equipados com sensores, como câmeras e sensores de imagem, que fornecem informações para as aplicações em que são utilizados. O módulo de planejamento de caminhos deve ser capaz de processar essas informações em tempo real e ajustar a trajetória conforme necessário. Isso permite que os VANTs sejam utilizados de forma eficiente em diversas áreas, como vigilância, mapeamento e inspeções de infraestruturas.

Dentre as técnicas de planejamento de caminhos, destaca-se o uso da aprendizagem por reforço (11). Essa abordagem oferece a vantagem de permitir que os agentes, como os VANTs, aprendam comportamentos de navegação de forma autônoma, tomando decisões com base em interações com o ambiente. A aprendizagem por reforço envolve a atribuição de recompensas ou penalidades às ações do agente, incentivando-o a aprender ações que maximizem as recompensas ao longo do tempo (12).

A literatura atual apresenta soluções foram desenvolvidas e testadas apenas em ambientes fictícios internos, contendo obstáculos fixos, simulados em 2D (3, 2, 13, 14, 15). Devido a isso, é importante abordar a simulação de ambientes realísticos em três dimensões, bem como a necessidade de lidar com situações imprevistas, como mudanças repentinas no ambiente por meio de obstáculos móveis de diversos tipos e variações climáticas no cenário como a simulação de ventos (16). Após a conclusão da fase de levantamento teórico na área de planejamento de caminhos através da abordagem de aprendizagem por reforço, é possível notar, de maneira geral, a existência de lacunas a serem exploradas.

Além disso, os autores de Mannan et al.(17) citam também como limitações que os algoritmos de aprendizagem por reforço na literatura não consideram alguns fatores importantes, como os recursos limitados dos VANTs e a otimização do caminho. Deixam claro que o desempenho atual dos algoritmos de aprendizagem por reforço é adequado para demonstrar que, se esses algoritmos forem aprimorados em otimização, precisão, autonomia e prevenção de obstáculos, o uso de VANTs se tornará mais confiável e intensivo (17).

A navegação autônoma de VANTs em ambientes tridimensionais apresenta desafios significativos. É necessário lidar com a presença de obstáculos em diferentes alturas e a complexidade do espaço aéreo, o que demanda abordagens específicas para o planejamento de caminhos.

Ademais, os próximos capítulos estão organizados da seguinte maneira: O capítulo 2 abordará a fundamentação teórica, apresentando os conceitos essenciais para o entendimento do trabalho. No capítulo 3, será detalhada a metodologia empregada na realização dos experimentos e construção do trabalho. O capítulo 4 discutirá os métodos de planejamento de caminhos utilizados, a modelagem de percepções 2D em 3D proposta neste trabalho, bem como os métodos e base para a construção das abordagens de planejamento de caminhos clássica e moderna. O capítulo 5 apresentará os experimentos e avaliações, detalhando os resultados e análises a partir dos experimentos realizados. Por fim, o Capítulo 6 trará as considerações finais, concluindo o trabalho e destacando possíveis direções para pesquisas futuras.

1.1 OBJETIVOS

O objetivo principal desta pesquisa é abordar a problemática do planejamento de caminhos para VANTs em ambientes 3D, por meio do desenvolvimento de um módulo que utilize técnicas de aprendizagem por reforço. O foco está em superar os desafios do planejamento em espaços tridimensionais para as duas técnicas, considerando a presença de obstáculos e a necessidade de otimização de trajetórias. A proposta visa desenvolver um sistema capaz de aprender e adaptar suas estratégias de planejamento com base nas interações com o ambiente, levando em consideração fatores como a eficiência e otimização de caminhos. Além disso, será realizado um comparativo com a abordagem clássica de planejamento de caminhos, por meio do emprego do algoritmo de busca A*.

Os objetivos específicos desta pesquisa são:

- Produzir um referencial teórico abrangente sobre planejamento de caminhos com o uso de VANTs, explorando as principais abordagens, algoritmos e técnicas empregadas nessa área. Isso fornecerá um referencial teórico sólido para embasar o desenvolvimento da pesquisa;
- Configurar um VANT simulado com os sensores e equipamentos necessários para o planejamento de caminhos. Isso inclui a seleção adequada dos sensores, sua integração no sistema do VANT e a realização de testes para garantir seu correto funcionamento;
- Desenvolver cenários com complexidades distintas para a realização de testes e avaliação dos métodos propostos;
- Desenvolver um modelo de percepção em três dimensões a partir de dados de sensores 2D para detecção de obstáculos, fornecendo ao VANT informações tridimensionais sobre o

ambiente sem a necessidade de investimentos financeiros significativos em sensores de última geração;

- Realizar uma avaliação dos modelos propostos de planejamento de caminhos com o uso de VANTs. Isso envolverá testes em diferentes cenários e condições, a fim de verificar a eficácia e a robustez do método proposto. Os resultados serão analisados e discutidos, contribuindo para a validação e o aprimoramento do modelo proposto.

1.2 JUSTIFICATIVAS E CONTRIBUIÇÕES

A motivação para a realização deste trabalho surge da necessidade de aprimorar as técnicas de navegação autônoma, em especial no contexto da utilização de VANTs. Pesquisas recentes (3, 2, 13, 14, 15) têm demonstrado limitações nas abordagens de planejamento de caminhos que utilizam aprendizagem por reforço, demonstrando a existência de lacunas. As abordagens identificadas foram desenvolvidas e testadas apenas em ambientes estáticos, simulados em 2D, com obstáculos fixos. Além disso, os VANTs desempenham papéis em diversas aplicações de interesse social, como busca e salvamento (18), monitoramento ambiental, inspeção de infraestruturas e distribuição de suprimentos em áreas de difícil acesso.

O planejamento de caminhos eficiente e otimizado contribui significativamente para o sucesso dessas missões, ajudando a salvar vidas, proteger o meio ambiente e melhorar a qualidade de vida das pessoas (17). Portanto, é importante explorar e desenvolver abordagens de planejamento de caminhos que sejam capazes de lidar com ambientes dinâmicos, desconhecidos e complexos, a fim de impulsionar o progresso e a eficácia da navegação autônoma de VANTs em aplicações do mundo real (7).

A utilização de VANTs na indústria e em setores como logística, agricultura, construção civil e cinema está em crescimento (9). O planejamento de caminhos otimizados aumenta a eficiência operacional, reduzir custos e melhora a produtividade. Por exemplo, no meio agrícola, VANTs podem ser usados para mapear, irrigar e monitorar plantações (19), permitindo a aplicação precisa de insumos agrícolas, economizando recursos e aumentando os rendimentos.

A área de planejamento de caminhos com VANTs apresenta vastos desafios técnicos e científicos. Envolve o desenvolvimento de algoritmos de planejamento robustos, inteligentes e adaptativos, que considerem a mudança do ambiente, a presença de obstáculos e as restrições operacionais dos VANTs, como a limitação de energia. Essa área de pesquisa contribui para o

avanço da inteligência artificial, aprendizado de máquina, robótica e otimização, impulsionando o progresso do estado da arte.

2 FUNDAMENTAÇÃO TEÓRICA

Planejamento de caminhos é uma área fundamental no campo da robótica e dos veículos autônomos, que se dedica a determinar a trajetória ideal para um agente se mover e navegar de forma eficiente em ambientes complexos (20). O planejamento de caminhos envolve a concepção e implementação de algoritmos e estratégias que permitem aos robôs determinar trajetórias seguras e otimizadas para atingir seus objetivos (21). Esses algoritmos consideram uma variedade de fatores, como a topologia do ambiente, a presença de obstáculos, as restrições físicas do robô e os objetivos da tarefa em questão. É uma área que abrange uma ampla gama de desafios, desde a navegação básica em ambientes estáticos (22) até a exploração de espaços desconhecidos e a adaptação a mudanças dinâmicas (23). Para isso, são utilizados métodos de percepção por meio de sensores embarcados, modelagem do ambiente, planejamento de trajetórias e técnicas de controle, buscando otimizar o desempenho dos robôs em diferentes aplicações, como logística e indústria (17).

No contexto de VANTs, o planejamento de caminhos desempenha o papel de definir a rota a ser seguida para atingir um objetivo pré-determinado, levando em consideração fatores como obstáculos, restrições operacionais e eficiência do percurso (3). O objetivo é maximizar a eficácia e segurança da navegação do VANT, garantindo o cumprimento da missão de forma otimizada (4).

Para alcançar o objetivo de planejar caminhos eficazes para VANTs, é importante empregar técnicas e algoritmos de planejamento de caminhos adequados ao contexto. Estas abordagens podem ser categorizadas em métodos baseados em grade (24), aprendizado de máquina, e técnicas híbridas que combinam diferentes abordagens. A eficácia desses métodos de planejamento está diretamente ligada à capacidade de planejar trajetórias de voo de maneira eficiente e segura. Métodos baseados em grade, como o algoritmo A*, oferecem uma estrutura sólida para o planejamento de caminhos em ambientes discretizados, mas podem ser limitados em cenários complexos e dinâmicos (2). Por outro lado, técnicas de aprendizado de máquina, como redes neurais, e aprendizagem por reforço têm demonstrado promissoras capacidades de adaptação a ambientes desafiadores (17), aprendendo com experiências anteriores.

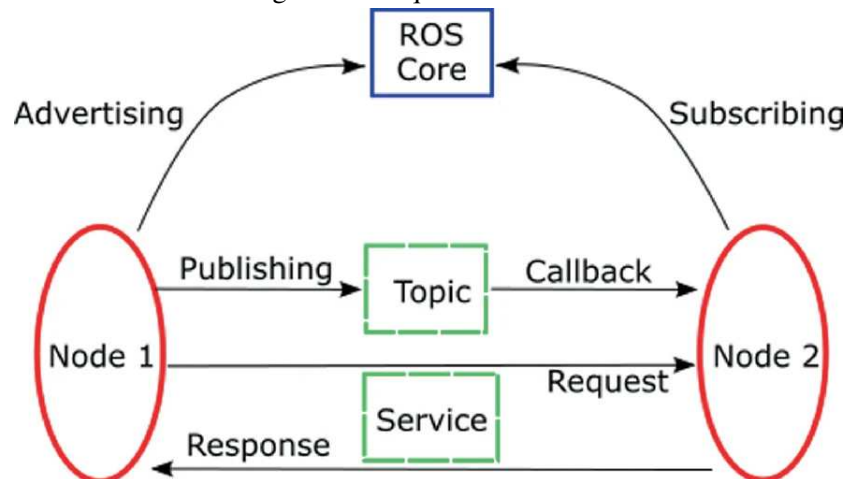
A seguir são apresentados mecanismos que podem ser explorados no planejamento de caminhos, como o *framework* de criação de ambientes para aprendizagem por reforço OpenAI Gym, a biblioteca de aprendizado por reforço Stable Baselines, e sistemas multi-robôs para VANTs MRS UAV System. Essas ferramentas e estruturas oferecem um ambiente customizado e flexível para o desenvolvimento e teste de algoritmos de planejamento de caminhos. Além disso, também serão abordados os métodos tradicionais utilizados em problemas de planejamento de caminhos. Em seguida, na seção Aprendizagem por Reforço, é introduzido o conceito central deste trabalho, abordando os fundamentos e princípios da aprendizagem por reforço, como a interação agente - ambiente, o processo de tentativa e erro, e a utilização de recompensas para guiar o aprendizado do agente. Por fim, na seção Trabalhos Relacionados, são apresentados estudos e pesquisas que exploram o uso de técnicas de aprendizagem por reforço na área de planejamento de caminhos, fornecendo um panorama do estado da arte e destacando contribuições relevantes no campo da aprendizagem por reforço.

2.1 SISTEMA OPERACIONAL DE ROBÔS - ROS

ROS (25) é uma plataforma flexível projetada para controlar diversos componentes robóticos a partir de um computador. O sistema ROS é construído com base em uma arquitetura distribuída, na qual vários nós independentes se comunicam entre si por meio de um modelo de mensagens de publicação/assinatura. Cada nó possui sua própria funcionalidade específica e pode ser facilmente desenvolvido e modificado para se adequar às necessidades do robô em questão (26). Essa abordagem modular permite que diferentes componentes, como *drivers* de sensores (uma câmera, por exemplo), sejam implementados como nós individuais que publicam dados em fluxos de mensagens. Dessa forma, um sensor pode ser criado como um nó separado, responsável por coletar os dados do sensor e transmiti-los para outros nós receberem essas informações.

A Figura 1 apresenta a arquitetura do ROS para o desenvolvimento de sistemas robóticos. No centro da figura, temos o núcleo do ROS, chamado *roscore*, que realiza a coordenação e comunicação entre os diferentes componentes (nós) do sistema. À esquerda e à direita do *roscore*, observamos dois nós individuais, denominados de *Node 1* e *Node 2*, que são partes autônomas do sistema ROS. Esses nós podem ser vistos como módulos de software que realizam tarefas específicas, como o nó para controlar um motor ou uma câmera, por exemplo.

Figura 1 – Arquitetura do ROS



Fonte: Retirado de St-Onge e Herath(27).

Os nós à esquerda e à direita estão ligados ao *roscore* e estão envolvidos na comunicação através de tópicos e serviços. Os tópicos são canais de comunicação de mão única, onde os nós podem publicar mensagens para que outros nós as consumam (25). Os serviços, por outro lado, são usados para comunicação de duas vias, permitindo que um nó solicite uma ação a outro nó e obtenha uma resposta (25).

A comunicação entre os nós é realizada por meio de *callbacks*. Quando um nó publica uma mensagem em um tópico, outros nós que estão inscritos nesse tópico têm funções de *callback* que são acionadas para processar a mensagem recebida (27). Isso permite que os nós interajam dinamicamente aos eventos e compartilhem informações de maneira assíncrona (27).

No contexto de VANTs, o ROS pode ser utilizado para diversas tarefas, como controle de voo, processamento de sensores, planejamento de trajetória e interação com outros sistemas. Os componentes de controle de voo de um VANT podem ser implementados como nós individuais no sistema ROS (28). Cada nó pode ser responsável por tarefas específicas, como a leitura de câmeras e sensores para detecção de obstáculos, ou controle e estimação de trajetória. Também pode ser utilizado para realizar a interface para controlar os motores do VANT ou outros tipos de atuadores como garras ou leds informativos.

A integração do ROS com o simulador Gazebo (29) oferece um ambiente simulado realista, permitindo a implementação e validação de algoritmos em um cenário virtual antes da implantação prática. A próxima seção explorará em detalhes como o ROS e o Gazebo são empregados nesse contexto para proporcionar uma plataforma eficiente de desenvolvimento e teste.

2.2 GAZEBO

Gazebo¹ (29) é um simulador 3D bastante utilizado nas pesquisas acadêmicas e mais recentemente na indústria de robótica. Possui código-fonte aberto e um motor físico robusto, podendo criar simulações com diversos parâmetros no mundo, possibilitando simulações em ambientes complexos externos e internos (30).

O Gazebo possui uma modelagem física altamente precisa, que permite simular o comportamento realista de robôs e VANTs em ambientes tridimensionais. Ele inclui suporte para dinâmica de corpo rígido, colisões, sensores e atuadores. O Gazebo permite que os usuários criem ambientes 3D personalizados, replicando cenários do mundo real, como áreas urbanas, florestas e ambientes industriais.

Embora o Gazebo possua um conjunto limitado de modelos prontos, a sua flexibilidade permite a integração de uma variedade de modelos de robôs provenientes de diversas plataformas de software. Existem fontes que disponibilizam modelos pré-construídos de VANTs (31, 32), por exemplo. A principal vantagem do Gazebo reside na sua capacidade de ser totalmente customizável, possibilitando a criação de modelos de robôs específicos para atender às necessidades de projetos específicos. Essa característica amplia significativamente as opções disponíveis para simulações realistas no ambiente Gazebo.

A integração do Gazebo com o ROS é um marco significativo que oferece vantagens para o desenvolvimento e teste de sistemas robóticos autônomos. A integração com o ROS facilita a comunicação e o controle dos robôs simulados, permitindo que os desenvolvedores testem algoritmos de controle, planejamento de caminho e percepção em um ambiente simulado e controlado. Além disso, permite que os desenvolvedores utilizem todas as bibliotecas e ferramentas disponíveis no ecossistema do ROS para melhorar a funcionalidade dos seus sistemas robóticos.

Para operacionalizar o VANT em nosso estudo, fazemos uso do MRS UAV System, que será descrito na próxima seção.

¹ <<https://gazebo.org>>

2.3 MRS UAV SYSTEM

O *Multi-robot Systems Group UAV System* (MRS UAV) (31) é um software de código aberto para VANTs desenvolvido por pesquisadores da Universidade Técnica Checa de Praga. O sistema foi projetado para ser modular, flexível e fácil de usar, tornando-o ideal para uma ampla gama de aplicações, incluindo:

- Pesquisa em áreas como enxames aéreos, manipulação aérea, planejamento de movimento, sensoriamento remoto;
- Implantação no mundo real em áreas como inspeção de infraestrutura, monitoramento ambiental, busca e salvamento, e entrega de pacotes;
- Educação em engenharia aeroespacial, robótica e inteligência artificial

O MRS UAV é baseado no ROS e pode ser executado inteiramente a bordo de um computador de bordo. O sistema inclui uma variedade de componentes de software para controle, estimativa de estado, mapeamento e planejamento (33). O sistema também é pré-configurado para uma variedade de plataformas e tipos de VANT, que podem ser adquiridas de parceiros do MRS UAV e, em simulações - possui VANTs já modelados para o simulador Gazebo. O sistema também inclui uma variedade de algoritmos de planejamento de alto nível, que permitem que ele execute tarefas complexas de forma autônoma.

Além disso o MRS UAV vem ganhando destaque com as primeiras colocações em competições prestigiadas envolvendo VANTs multi rotores nos desafios *Mohamed Bin Zayed International Robotics Challenge* (MBZIRC) 2017, 2020 e no *Subterranean Challenge* (SubT) do *Defense Advanced Research Projects Agency* (DARPA) (34). Oferecendo implementação em código-fonte aberto documentada, o sistema pode ser aplicado tanto em simulações por meio do *Gazebo Simulator* quanto na vida real. O software também inclui simulações realísticas com sensores (laser e câmera, por exemplo), sistemas de localização, planejamento de movimento e locomoção em ambientes dinâmicos e complexos (33).

2.4 OPENAI GYM

O OpenAI Gym Brockman et al.(35) é uma biblioteca amplamente utilizada no campo da aprendizagem por reforço. O Gym oferece uma ampla variedade de ambientes, desde problemas simples e clássicos, como jogos Atari (36), até tarefas complexas de controle de

robôs.

No contexto de VANTs, a biblioteca OpenAI Gym pode ser empregada para criar ambientes personalizados que permitem o treinamento e avaliação de agentes de aprendizado por reforço associados à navegação autônoma, por exemplo (37, 3, 38, 5).

Cada ambiente no Gym é projetado para fornecer uma representação clara do problema, permitindo que os agentes de aprendizagem por reforço interajam com ele de maneira intuitiva. Isso inclui a observação do estado atual do ambiente, a execução de ações e a recepção de recompensas com base em seu desempenho (35).

Além dos ambientes prontos para uso, o Gym também fornece uma API para criar novos ambientes personalizados (35). Isso permite a criação de ambientes personalizados, interagindo diretamente com os cenários implementados. Além disso, o Gym lida com as ações, percepções e controle dos agentes, possibilitando assim, uma fácil integração com os algoritmos de aprendizagem por reforço para realizar o treinamento do agente.

A fim de treinar o agente de aprendizagem por reforço, empregamos o Stable Baselines3, conforme será abordado na próxima seção.

2.5 STABLE BASELINES3

O Stable Baselines3 (39) é uma biblioteca de aprendizagem por reforço bastante conhecida e utilizada pela comunidade, que oferece implementações de algoritmos para treinar agentes em tarefas de aprendizagem por reforço.

A biblioteca fornece métodos simples para definir e configurar os hiperparâmetros dos algoritmos, além de acompanhar o progresso na fase de treinamento. Além disso, o Stable Baselines3 oferece recursos para avaliar o desempenho dos agentes, visualizar os resultados dos modelos treinados para uma análise posterior.

Uma das características notáveis do Stable Baselines3 é sua integração com o OpenAI Gym, facilitando a configuração de ambientes de treinamento e o desenvolvimento de modelos de aprendizado por reforço. Além disso, a biblioteca oferece suporte para treinamento em CPU e GPU, o que a torna adequada para uma ampla variedade de configurações de hardware.

Outro ponto do Stable Baselines3 é sua documentação detalhada e exemplos práticos

que auxiliam os desenvolvedores a começar com o aprendizado por reforço. A biblioteca é bastante utilizada na comunidade de pesquisa e desenvolvimento de aprendizado de máquina e robótica, sendo uma escolha popular para a implementação de agentes autônomos em diversos domínios, desde jogos até robótica.

2.6 MÉTODOS CLÁSSICOS

O planejamento de caminhos clássico é uma abordagem tradicionalmente utilizada para resolver problemas de determinação de trajetórias em diversos domínios, incluindo robótica e sistemas autônomos. Nesse contexto, o objetivo é encontrar uma sequência de ações que permita que um agente atinja um objetivo desejado de forma eficiente e segura.

Existem várias técnicas clássicas de planejamento de caminhos, cada uma com suas características e requisitos específicos. Algumas dessas técnicas incluem o algoritmo A* (40), o algoritmo de Dijkstra (41), os campos potenciais (42), algoritmo de *Breadth First Search* (BFS) (43) e o *Probabilistic Roadmaps* (PRM) (24). Esses algoritmos são projetados para trabalhar em espaços de estado discretos, onde o ambiente é representado por uma grade ou grafo, e as ações do agente são limitadas a movimentos específicos.

O planejamento de caminhos clássico geralmente envolve a construção de uma representação do ambiente, definindo os estados possíveis, as ações permitidas e as informações de custo ou heurística para guiar o processo de planejamento. Com base nessa representação, os algoritmos de planejamento exploram o espaço de estados para encontrar a melhor sequência de ações que leva ao objetivo.

2.6.1 A* (A-Estrela)

O A* (40) é um algoritmo de busca informada para encontrar o caminho ótimo em um grafo ou em um espaço de estado. O A* utiliza uma heurística para estimar o custo restante até o objetivo, juntamente com o custo atual acumulado, para selecionar os próximos nós a serem explorados. Essa heurística é chamada de função heurística e é uma parte fundamental do algoritmo.

A fórmula geral do A* para o custo total (f) de um nó é:

$$f(n) = g(n) + h(n) \quad (2.1)$$

Onde:

- $f(n)$ é o custo total estimado de passar pelo nó n até o objetivo.
- $g(n)$ é o custo atual do caminho do nó inicial até n .
- $h(n)$ é uma estimativa heurística do custo do caminho do nó n até o objetivo.

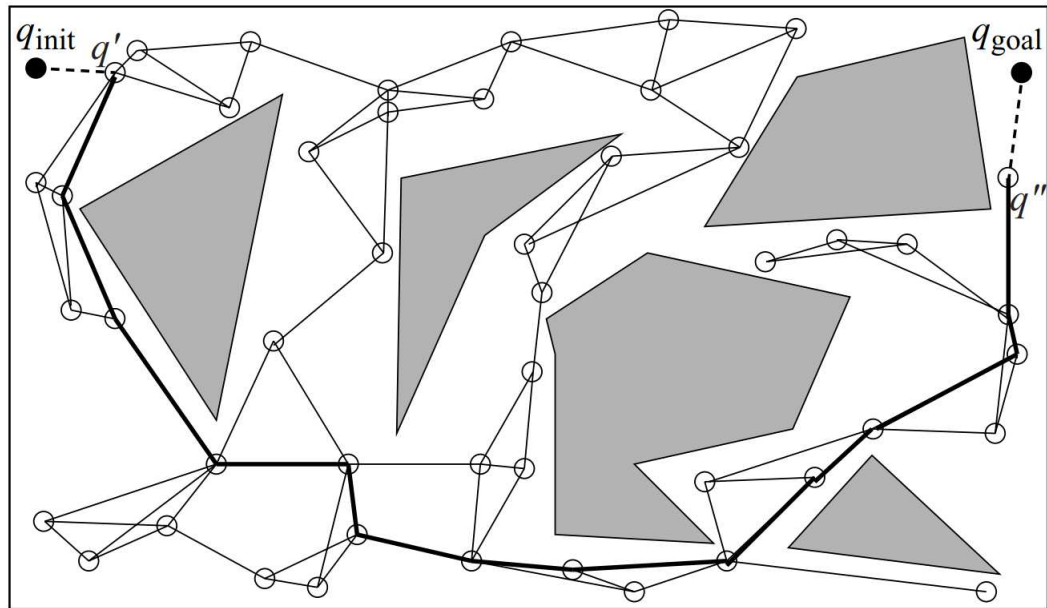
O A* mantém uma lista de nós a serem explorados, chamada de fronteira, e começa com o nó inicial. Ele expande os nós na fronteira com os menores valores de $f(n)$ primeiro. A cada expansão, o algoritmo verifica se o nó atual é o objetivo desejado. Se não for, ele gera os sucessores do nó atual e calcula os valores f , g e h para cada um deles, em seguida, os adiciona à fronteira.

Além disso, o A* possui o componente heurística, que é uma função que fornece uma estimativa do custo restante para atingir o objetivo a partir de um determinado nó no espaço de busca. Essa estimativa fornece ao algoritmo a capacidade de decidir qual caminho explorar primeiro, priorizando nós que têm uma estimativa de custo menor. Ela atua como um guia para o algoritmo, ajudando-o a tomar decisões mais informadas sobre qual caminho seguir.

A ideia central é que, ao usar uma heurística apropriada, o algoritmo pode priorizar nós promissores, ou seja, aqueles que têm uma estimativa menor de custo total para alcançar o objetivo. Isso permite que o A* explore primeiro os caminhos que provavelmente levarão à solução mais rapidamente, economizando recursos computacionais e acelerando a busca. Além disso, uma característica importante de uma heurística eficaz é que ela deve ser admissível, o que significa que ela nunca superestima o custo real para alcançar o objetivo. Se uma heurística é admissível, o A* é garantido para encontrar o caminho mais curto e ótimo em um espaço de busca discretizado (40). Em outras palavras, $h(n) \leq h^*(n)$, onde $h^*(n)$ é o custo real para atingir o objetivo a partir do nó n . A heurística admissível é importante no A* porque garante a otimalidade do algoritmo. Se a heurística for admissível, o A* sempre encontrará a solução de menor custo.

No contexto das heurísticas não admissíveis, a estimativa de custo para alcançar o objetivo a partir de certos nós superestima o custo real. Nesse caso, o A* ainda pode encontrar um caminho, mas não garante a otimalidade. A heurística não admissível pode levar o A* a explorar mais nós do que o necessário e, em alguns casos, encontrar um caminho que não é o mais curto.

Figura 2 – O A* garante que os nós com menor custo total sejam priorizados na expansão, caso a função heurística seja admissível.



Fonte: Retirado de Choset et al.(44).

Na robótica, o A* é comumente aplicado em sistemas de navegação autônoma, planejamento de trajetórias e robótica móvel. Ele permite que os robôs encontrem caminhos seguros e eficientes em ambientes complexos (37), levando em consideração obstáculos, restrições de movimento e outros critérios.

2.7 APRENDIZAGEM POR REFORÇO

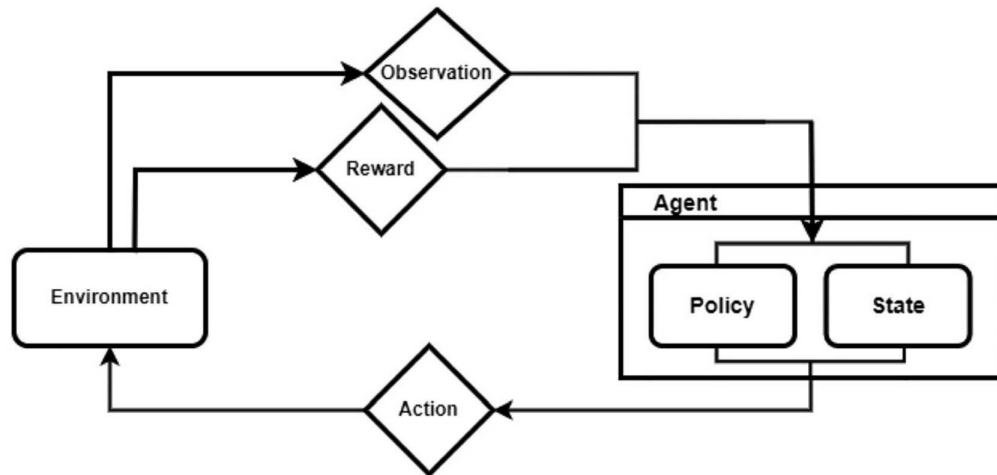
A aprendizagem por reforço é uma estrutura de aprendizado de máquina onde o agente toma decisões em seu ambiente, e as consequências dessas ações são refletidas em recompensas ou penalizações. Através de um grande número de tentativas, o agente ajusta suas estratégias de ação, otimizando seu comportamento para maximizar as recompensas e minimizar as penalizações (10).

A interação contínua entre o agente e o ambiente, representado pela Figura 3, é o que caracteriza o processo de aprendizagem. O agente busca aprender a tomar ações que maximizem as recompensas ao longo do tempo, adaptando sua política com base nas experiências adquiridas durante as interações com o ambiente. Esse ciclo de interação e aprendizado é fundamental para que o agente adquira conhecimento e desenvolva estratégias eficientes para resolver problemas complexos.

Um exemplo das iterações da aprendizagem por reforço seria o exemplo de um

agente aspirador de pó autônomo em um ambiente doméstico. nesse contexto, o agente atua como o aprendiz, tomando decisões no ambiente composto por diferentes quartos, podendo estar limpos ou sujos. Suas ações incluem movimentos para a esquerda, direita, cima e baixo, além da ação de aspirar quando posicionado sobre sujeira. Cada estado reflete uma configuração específica do ambiente, indicando quais quartos estão limpos ou sujos, assim como a posição atual do agente. As recompensas ou penalidades são atribuídas com base nas ações do agente, como receber uma recompensa positiva por aspirar sujeira ou uma penalidade por se movimentar para uma posição sem sujeira. Esse exemplo ilustra como o agente aprende a otimizar seu comportamento ao longo do tempo, buscando maximizar as recompensas obtidas nas interações com o ambiente.

Figura 3 – A interação entre os componentes centrais no aprendizado por reforço



Fonte: Retirado de Mannan et al.(17).

A seguir, na seção Agente, são apresentados a definição dos estados e o conceito de política, que determina as ações tomadas pelo agente com base no estado atual, e função de valor, que estima a recompensa total esperada ao longo do tempo. Em seguida, na seção Ações, são explorados os conceitos de ação, destacando sua importância no processo de aprendizagem por reforço e como eles permitem que o agente tome decisões e otimize seu desempenho. Na seção Ambiente abordaremos detalhes sobre o ambiente, que define as características do espaço onde o agente opera. Posteriormente na seção Recompensa será tratado a quantificação a eficácia das ações do agente por meio de uma recompensa. Na seção Observação, são discutidas a forma de representar as informações percebidas pelo agente durante sua interação com o ambiente. Por fim, na seção Algoritmos e Técnicas de Aprendizagem por Reforço, são apresentados alguns dos principais algoritmos e abordagens utilizadas na aprendizagem por reforço.

2.7.1 Agente

O agente é o responsável por tomar ações e aprender a partir das interações com o ambiente. Ele recebe informações sobre o estado atual do ambiente, como a posição atual do VANT, a distância até o objetivo e informações sobre os obstáculos presentes no ambiente. Com base nessas observações, o agente decide qual ação deve ser executada pelo VANT. escolhe uma ação com base nesse estado e recebe *feedback* na forma de recompensas, que indicam a qualidade de suas ações. O agente também possui uma política, que é uma estratégia para selecionar a melhor ação em cada estado, com o objetivo de maximizar a recompensa acumulada ao longo do tempo.

A política é a estratégia que um agente utiliza para buscar seus objetivos. Ela determina as ações que o agente executará com base em seu estado e no ambiente em que está inserido. Uma política é simplesmente uma função que retorna uma ação viável para um determinado problema. Por exemplo, uma política pode selecionar a primeira ação que vem à mente, escolher uma ação aleatória ou seguir uma heurística específica. Quanto melhor a política, mais precisos serão os resultados obtidos.

2.7.2 Ações

As ações representam as decisões que o agente toma em um determinado estado do ambiente para maximizar sua recompensa ao longo do tempo (45). Essas decisões são escolhidas a partir de um conjunto pré-definido de ações possíveis, formando o espaço de ação do agente.

Dependendo do problema, as ações podem ser discretas ou contínuas. Por exemplo, em um jogo de tabuleiro, as ações podem corresponder a movimentos específicos do tabuleiro, enquanto em um ambiente de controle de robô, as ações podem representar comandos de velocidade ou torque nos motores.

2.7.3 Recompensa

A recompensa é um valor de *feedback* fornecido ao agente após a execução de uma ação em um determinado estado. Ela representa uma medida de quão boa foi a ação tomada pelo agente. A recompensa pode ser positiva, negativa ou neutra, dependendo se a ação levou o agente em direção aos seus objetivos, afastou-se dos objetivos desejados ou não teve impacto

significativo, respectivamente.

O objetivo é que o agente aprenda a escolher ações que maximizem a recompensa acumulada ao longo do tempo. A formulação da função de recompensa é uma tarefa importante, pois ela define as prioridades e objetivos do agente durante o treinamento (2).

2.7.4 Ambiente

O ambiente é o contexto no qual o agente está inserido. Ele define as regras, as dinâmicas e as restrições do problema em que o agente deve aprender. O ambiente fornece ao agente informações sobre o estado atual, recebe as ações escolhidas pelo agente e retorna as recompensas correspondentes (3). Além disso, o ambiente também é responsável por atualizar seu estado interno de acordo com as ações do agente.

2.7.5 Observação

No contexto de aprendizado por reforço, a observação representa a percepção do agente sobre o ambiente após a execução de um passo (*step*) pelo agente no ambiente e fornece os dados necessários para que o agente tome decisões informadas.

As observações são obtidas por meio de sensores instalados no agente, como câmeras ou outros dispositivos de medição. No caso de um VANT navegando em um ambiente tridimensional, a observação pode incluir dados visuais, informações sobre a altitude, distâncias relativas a obstáculos, entre outros (46).

Essas observações podem variar significativamente com base na complexidade do ambiente e nos sensores disponíveis para o agente. As observações são projetadas para proporcionar ao agente uma visão do ambiente ao redor do agente.

2.7.6 Algoritmos e Técnicas de Aprendizagem por Reforço

Os algoritmos de aprendizagem por reforço têm várias aplicações, como gerenciamento de recursos, controle de tráfego, robótica, jogos, entre outros, onde um agente deve aprender a tomar decisões para maximizar uma recompensa em um ambiente dinâmico.

No campo da aprendizagem por reforço, diversos algoritmos são amplamente utilizados para resolver uma variedade de problemas. Entre eles, destacam-se o *Q-Learning* (47), que é

um dos algoritmos fundamentais no aprendizado por reforço, o DQN (48), que se destacou ao superar jogadores humanos em jogos de Atari, o *Proximal Policy Optimization* (PPO) (49), que foi projetado para lidar com políticas contínuas e discretas, e o *Deep Deterministic Policy Gradient* (DDPG) (50), especializado em problemas com ações contínuas. Esses algoritmos representam avanços significativos no campo, com aplicações bem-sucedidas em diversos domínios, contribuindo para o desenvolvimento de estratégias eficazes de aprendizado por reforço.

A seguir, abordaremos com mais detalhes o algoritmo DQN, técnica de aprendizagem por reforço utilizada neste trabalho.

2.7.6.1 *Deep Q-Network* (DQN)

DQN é uma variante do *Q-Learning* que usa uma rede neural profunda para aproximar a função de valor. O *Q-Learning* é um algoritmo de aprendizado por reforço que foi introduzido por Watkins e Dayan(47). O algoritmo é baseado no conceito de função de valor, que mapeia estados para valores esperados de recompensas futuras. O objetivo do *Q-Learning* é aprender a função de valor ótima para um determinado ambiente.

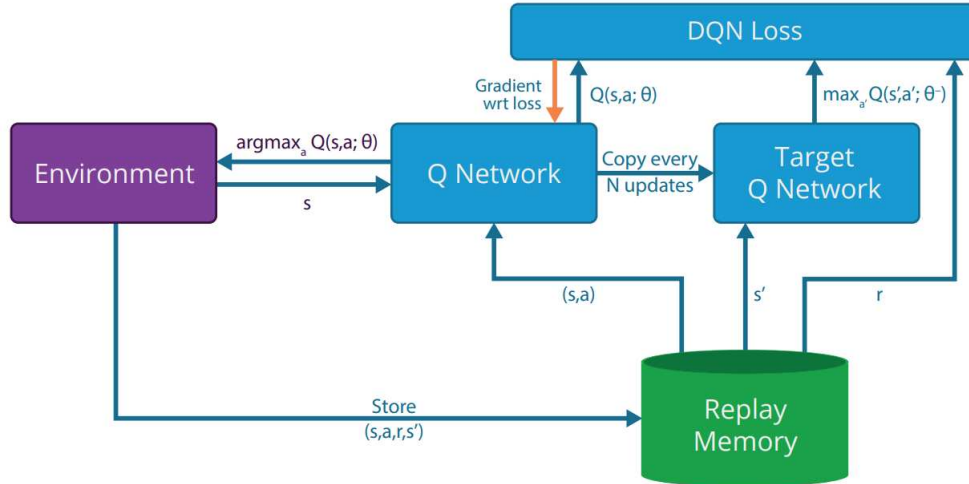
O *Q-Learning* funciona de forma iterativa, avaliando a função de valor para cada estado no ambiente. Para cada estado, o algoritmo calcula a diferença entre o valor esperado da ação atual e o valor esperado da melhor ação possível. Essa diferença é então utilizada para atualizar a função de valor para a ação atual. O algoritmo pode ser aplicado a uma ampla variedade de problemas de aprendizado por reforço.

O DQN foi apresentado por Mnih et al.(48) da DeepMind ². O DQN tem várias vantagens sobre o *Q-Learning* tradicional. Em primeiro lugar, o DQN pode representar funções de valor mais complexas do que o *Q-Learning* tradicional. Em segundo lugar, o DQN é mais eficiente do que o *Q-Learning*, pois pode aprender a partir de um conjunto de dados menor. O DQN funciona da mesma forma que o *Q-Learning*, com a diferença de que a função de valor é aproximada por uma rede neural profunda.

Para melhorar a eficiência do treinamento, o DQN utiliza uma técnica chamada *Experience Replay*. Isso envolve armazenar experiências passadas do agente em uma memória de *replay* e amostrar aleatoriamente essas experiências durante o treinamento (48). Isso ajuda a

² <<https://deepmind.google/>>

Figura 4 – O algoritmo DQN é composto por três componentes principais: a *Q-Network* ($Q(s, a; \theta)$), que define a política de comportamento, a *Target Q-Network* ($Q(s, a; \theta^-)$), usada para gerar valores Q de destino para o termo de perda do DQN, e a memória de *replay* que o agente utiliza para amostrar transições aleatórias para treinar a *Q-Network*.



Fonte: Retirado de Nair et al.(51).

quebrar a correlação temporal das transições de estado, melhorando a estabilidade do treinamento. Além disso, para evitar instabilidades no treinamento de redes neurais profundas, o DQN introduz uma segunda rede neural chamada *Target Q-Network*. Essa rede é usada para calcular a recompensa alvo durante o treinamento, ajudando a estabilizar o aprendizado.

2.8 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados alguns trabalhos relacionados que exploram o uso da aprendizagem por reforço no contexto do planejamento de caminhos para VANTs. O objetivo é analisar pesquisas que abordaram esse tema e identificar contribuições relevantes na área. Serão discutidos diferentes algoritmos e abordagens utilizados, bem como os resultados alcançados.

O autores Mannan et al.(17) realizaram uma revisão sistemática que compara algoritmos clássicos e técnicas de aprendizagem por reforço no planejamento de caminhos através do uso de VANT. As métricas identificadas nos trabalhos incluem distância percorrida, ambientes estáticos e dinâmicos, mobilidade 2D ou 3D, desvio de obstáculos e automação. Os resultados encontrados na pesquisa mostram que a aprendizagem por reforço proporciona maior autonomia, enquanto que os algoritmos clássicos são mais focados em soluções ótimas. O estudo deixa alguns *insights* sobre outras limitações existentes, como a utilização de ambientes conhecidos e desconhecidos e como os algoritmos aprendizagem por reforço lidam com obstáculos móveis

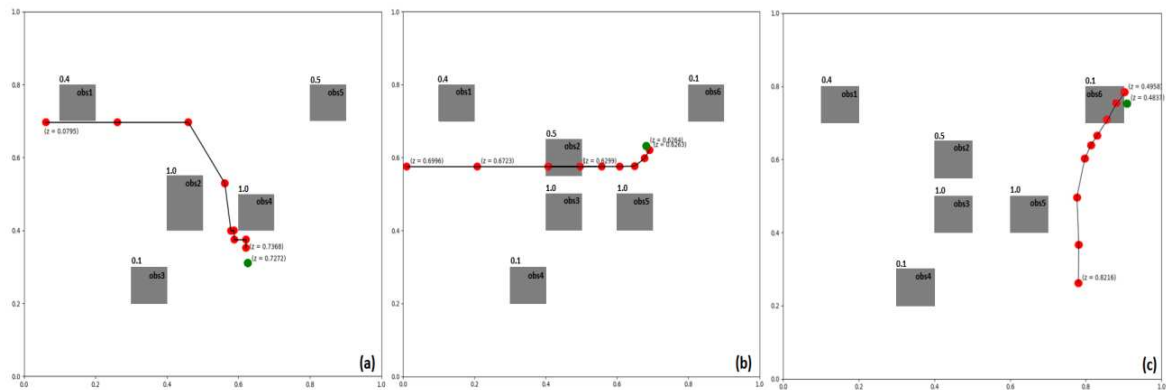
e fixos. As abordagens listadas no trabalho utilizaram ambientes de testes externos, florestais, internos e urbanos, e as técnicas de aprendizagem por reforço incluem várias variantes - *Deep Reinforcement Learning* (DRL), *Double Q-Learning* (DDQN) (52), *Reinforcement Learning* (RL) com *deep learning*, *Q-learning* e outros.

Ou et al.(53) propõe um modelo de desvio autônomo de obstáculos em ambientes semi conhecidos, utilizando a taxa de sucesso como métrica principal de avaliação. A abordagem apresentou uma taxa de sucesso média de 0,996 e comparou diferentes políticas de desvio de obstáculos, sendo que a D3RQN proposta obteve o melhor desempenho. O estudo foi testado em diferentes cenários de teste e tem como missão voar autonomamente evitando obstáculos e seguindo uma trajetória global planejada. As limitações incluem ambientes totalmente desconhecidos e a utilização de apenas um VANT. Seel et al.(54) e Xu et al.(55) apresentam abordagens que são utilizadas a navegação em ambientes complexos e totalmente desconhecidos através do uso de VANTs. Os resultados apresentados possuem altas taxas de sucessos nas missões, passando de 90% na maioria dos casos. Singh e Jha(56), Ma e Xu(57), Dong e Liu(13) e Chen et al.(58) propõem técnicas de planejamento de caminhos *Multi-VANTs*. As propostas foram capazes de aprender comportamentos seguros e cooperativos para a navegação em ambientes com restrições.

2.8.1 Bouhamed - Autonomous UAV Navigation: A DDPG-Based Deep Reinforcement Learning Approach

O trabalho realizado por Bouhamed et al.(1) avaliou a taxa de sucesso, distância percorrida e tempo necessário para alcançar o destino usando VANTs em ambientes com obstáculos. Embora não tenham sido testados em ambiente real, os resultados mostraram uma taxa de sucesso de 100% em ambiente livre de obstáculos e 84% e 82% em dois cenários diferentes com obstáculos. A abordagem utilizou o DDPG para realizar a aprendizagem por reforço.

Figura 5 – Modelo apresentado por Bouhamed et al.(1), com o caminho seguido pelo VANT com base no local de destino estático. Os pontos vermelhos referem-se ao VANT, pontos verdes referem-se ao alvo e caixas cinzas aos obstáculos.



Fonte: Retirado de Bouhamed et al.(1).

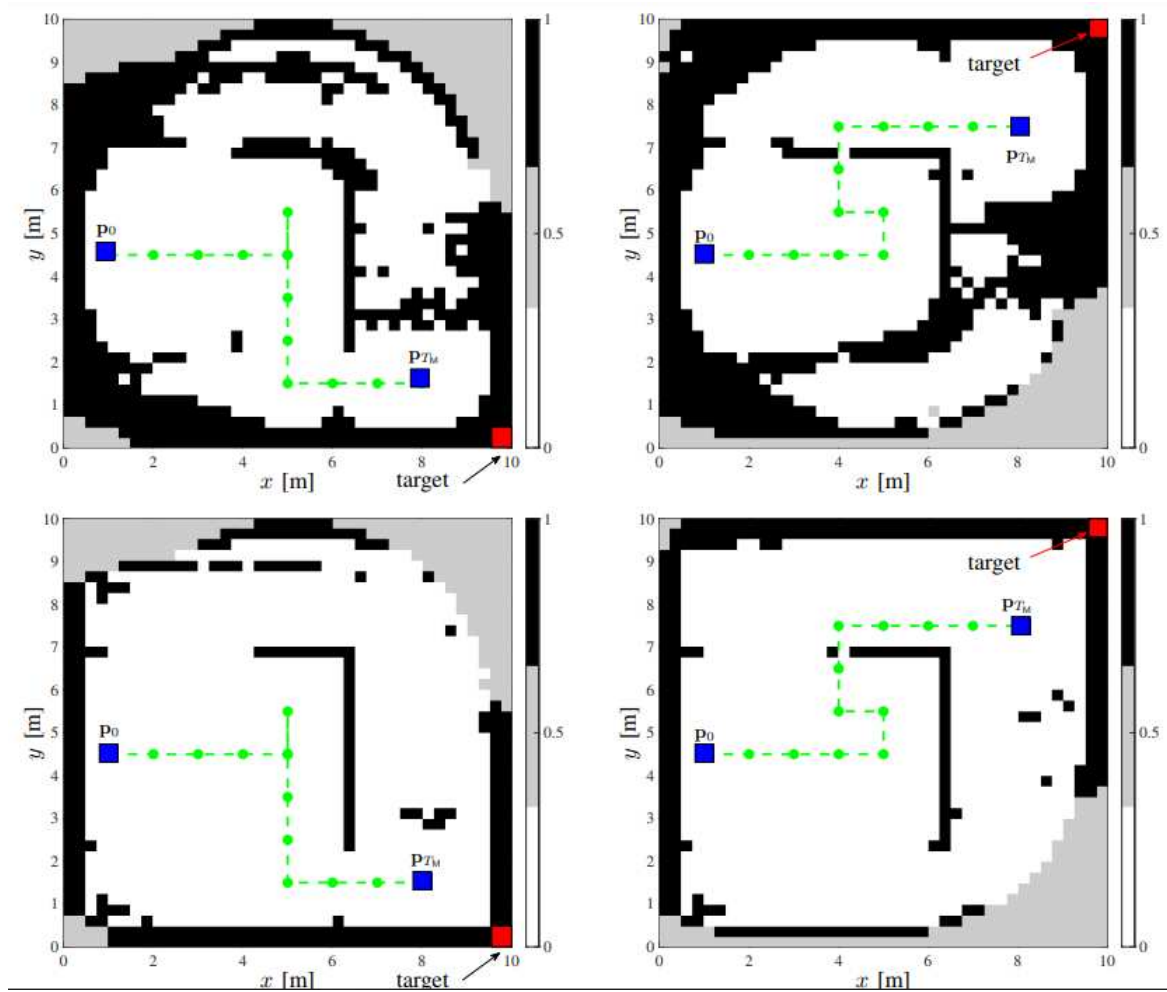
O objetivo era alcançar o destino evitando colisões com obstáculos no caminho. O objetivo era minimizar a distância percorrida pelo VANT e o tempo necessário para alcançar o alvo. Como limitações, não foram realizados testes em ambientes reais e os obstáculos eram fixos no mapa, além de ser uma abordagem em um ambiente 2D.

2.8.2 Guerra - Reinforcement Learning for UAV Autonomous Navigation, Mapping and Target Detection

O trabalho realizado por Guerra et al.(2) descreve uma abordagem baseada em aprendizado por reforço para a navegação autônoma de um VANT em um ambiente interno, com o objetivo de detectar alvos e reconstruir um mapa do ambiente. A abordagem é avaliada utilizando a taxa de detecção correta e a taxa de alarme falso e os resultados mostram que a abordagem proposta é capaz de otimizar a navegação do VANT e a detecção de alvos em um ambiente desconhecido, com resultados superiores em relação a outras abordagens existentes na literatura.

A abordagem proposta é capaz de otimizar a navegação do VANT e a detecção de alvos em um ambiente desconhecido. Os resultados mostraram que o desempenho da detecção de alvos e do mapeamento do ambiente depende da trajetória do VANT e do número de elementos de antena, e que a otimização da trajetória do VANT é essencial para melhorar o desempenho final. Além disso, a abordagem proposta foi comparada com outras abordagens existentes na literatura, mostrando resultados superiores em termos de Taxa de Detecção Correta (CDR) e

Figura 6 – Resultados da navegação autônoma dos VANTs no trabalho apresentado por Guerra et al.(2).
Esquerda: alvo colocado no canto inferior direito. Direita: alvo colocado no canto inferior esquerdo



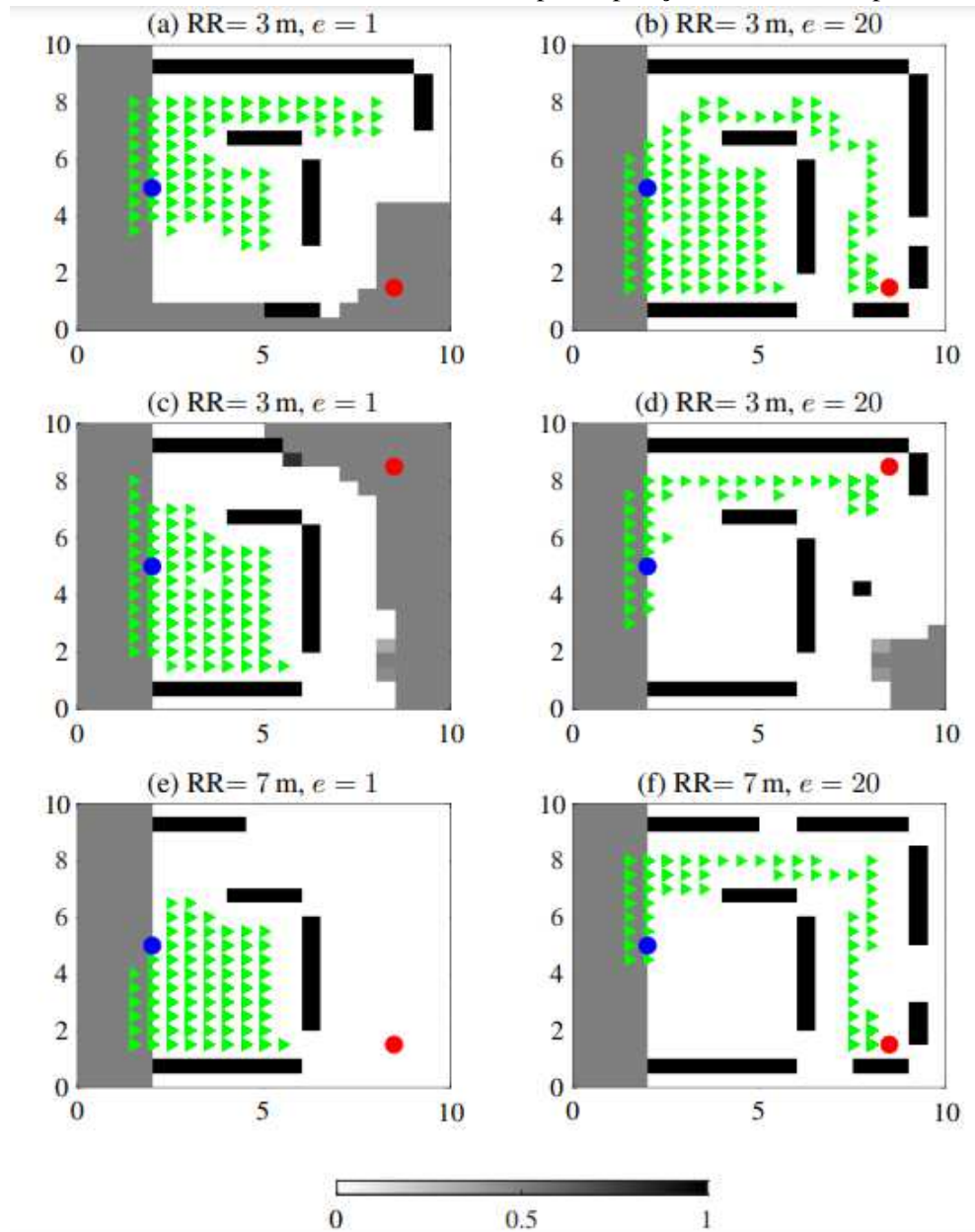
Fonte: Retirado de Guerra et al.(2).

a Taxa de Alarme Falso (FAR). No entanto, a pesquisa tem como limitação a utilização de ambientes fixos e 2D, além de não ter obstáculos no mapa, apenas paredes.

2.8.3 Guerra - Real-Time Learning for THZ Radar Mapping and UAV Control

Guerra et al.(3) descreve uma abordagem para navegação autônoma de um VANT em ambientes internos, utilizando aprendizado por reforço com o algoritmo *Q-learning*. O objetivo é detectar um alvo e reconstruir um mapa do ambiente. Os resultados mostram que o uso de um radar THz pode melhorar a capacidade do VANT de realizar a tarefa de detecção do alvo em ambientes internos. O artigo também descreve a formulação do problema como um *Markov Decision Process* (MDP) e apresenta uma análise dos resultados obtidos em diferentes cenários, considerando parâmetros do radar, como a Potência Efetiva Irradiada Isotropicamente (EIRP).

Figura 7 – Exemplos de trajetórias estimadas e mapas da abordagem realizada por Guerra et al.(3). Os marcadores azuis e vermelhos indicam p_0 e a posição de destino, respectivamente.



Fonte: Retirado de Guerra et al.(3).

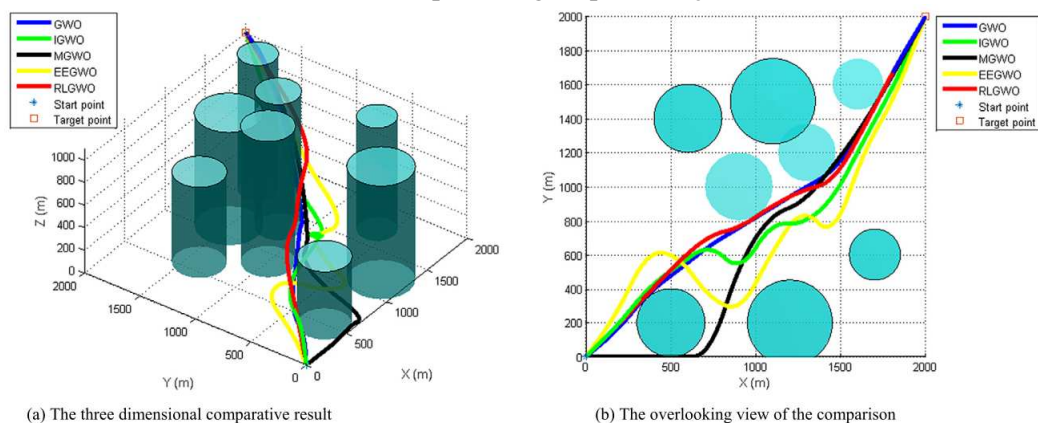
Os resultados obtidos mostram que a abordagem proposta é capaz de realizar a navegação autônoma de um VANT em um ambiente interno, detectando um alvo e reconstruindo um mapa do ambiente. Os resultados também mostram que o uso de um radar THz pode melhorar a capacidade do VANT de realizar a tarefa de detecção do alvo em ambientes internos. No entanto, a abordagem desenvolvida utiliza apenas um único algoritmo de aprendizado por reforço com um único tipo de radar, e foi testada em apenas um ambiente simulado e 2D.

2.8.4 Chengzhi - A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning

O trabalho realizado por Qu et al.(4) propõe um algoritmo inovador chamado RLGWO para o planejamento de trajetória de veículos aéreos não tripulados VANTs em ambientes tridimensionais complexos. O algoritmo combina o *Grey Wolf Optimizer* (GWO) com técnicas de aprendizagem por reforço para otimizar o problema de planejamento de trajetória. O RLGWO inclui quatro operações para cada indivíduo: alocação, exploração, ajuste geométrico e ajuste ótimo. Essas operações são controladas pela aprendizagem por reforço para alternar adaptativamente entre diferentes estratégias com base no desempenho acumulado. O algoritmo é comparado com outros algoritmos de otimização e os resultados mostram que o RLGWO supera os demais em termos de desempenho de busca e convergência.

Os resultados deste estudo revelaram que o algoritmo RLGWO proposto superou outros algoritmos em várias métricas, incluindo valores ótimos, piores resultados, média e desvio padrão. O desempenho do algoritmo RLGWO demonstrou notável convergência e estabilidade na busca do caminho ideal para VANTs em nos ambientes em que a abordagem foi avaliada. Os resultados experimentais destacaram a capacidade do algoritmo RLGWO em encontrar caminhos mais eficientes e viáveis para VANTs, enquanto outros algoritmos apresentaram limitações, como resultados insatisfatórios, otimização local ou comportamento oscilatório.

Figura 8 – Resultados obtidos por Qu et al.(4) comparando o RLGWO com técnicas similares de aprendizagem por reforço



Fonte: Retirado de Qu et al.(4).

Este estudo apresenta algumas limitações que merecem consideração. Em primeiro lugar, a abordagem proposta foi testada em um ambiente específico, o que pode limitar sua

representatividade em relação a cenários mais complexos para o planejamento de caminhos para VANTs. Além disso, o estudo se concentrou exclusivamente em um algoritmo específico, o RLGWO, sem a realização de comparações com outros algoritmos de planejamento de caminhos de última geração ou outras abordagens de Aprendizagem por Reforço. A inclusão de tal análise comparativa poderia proporcionar uma avaliação mais completa da eficácia da abordagem. Além disso, é importante observar que a pesquisa não abordou a presença de obstáculos complexos, limitando-se a utilizar cilindros. Isso, de certo modo, simplifica o ambiente, representando uma visão bidimensional e reduzindo a complexidade do mundo real.

3 METODOLOGIA

A metodologia utilizada neste trabalho será a análise comparativa, ou seja, o objetivo será projetar e executar experimentos controlados com VANTs simulados para coletar dados sobre o planejamento e a execução dos caminhos obtidos, para assim realizar uma comparação direta entre a abordagem clássica e o planejamento de caminhos utilizando a aprendizagem por reforço.

A metodologia é dividida em quatro fases principais: contextualização, planejamento, implementação e validação.

A primeira fase é contextualização, onde é realizada uma revisão da literatura sobre o tema, selecionando algoritmos de aprendizagem por reforço aplicáveis e estabelecendo critérios de avaliação e métricas de desempenho.

Em seguida, temos a fase de planejamento, em que diversas decisões estratégicas são tomadas para moldar o ambiente de simulação e os elementos-chave do sistema. A escolha do simulador, *framework* de desenvolvimento e modelo de VANT, sensores embarcados a serem utilizados e a definição dos cenários e quantidade de teste são determinantes para a eficácia e abrangência dos experimentos.

Na fase de implementação, na qual é realizada a implementação do ambiente de simulação, desenvolvimento do módulo de planejamento de caminhos com aprendizagem por reforço e A*, integração com o ambiente de simulação e treinamento do algoritmo.

Por fim, a fase de validação envolve a realização de experimentos nos cenários de teste definidos, coleta de dados de desempenho, análise comparativa com o algoritmo A*, interpretação dos resultados e gerar conclusões do estudo.

A Tabela 1 apresenta a metodologia comparativa para a análise do planejamento de caminhos com a abordagem clássica e aprendizagem por reforço em VANTs.

Tabela 1 – Metodologia Comparativa: Planejamento de caminhos com A* e aprendizagem por reforço

Fases	Atividades
Contextualização	1. Revisão da literatura sobre planejamento de caminhos com aprendizagem por reforço em VANTs 2. Definição dos critérios de avaliação e métricas de desempenho
Planejamento	1. Escolha do simulador 2. Modelo de VANT a ser utilizado nos experimentos 3. Sensoriamento do agente
Implementação	1. Implementação do ambiente de simulação 2. Desenvolvimento do módulo de planejamento de caminhos com o A* 3. Desenvolvimento do módulo de planejamento de caminhos com aprendizagem por reforço 4. Integração do módulo com o ambiente de simulação e o VANT 5. Ajuste dos parâmetros e treinamento do algoritmo de aprendizagem por reforço
Validação	1. Realização de experimentos nos cenários de teste definidos 2. Coleta de dados de desempenho e avaliação dos resultados 3. Análise comparativa com o algoritmo A* 4. Documentar os resultados e conclusões da análise

Fonte: O próprio autor.

3.1 PLANO DE TESTES

Nesta seção será apresentado o plano de testes para avaliar os métodos de planejamento de voo, A* (abordagem clássica) e aprendizagem por reforço (abordagem moderna), no cenário proposto. Abaixo será apresentado os cenários e o plano de testes que inclui pontos de saída e chegada, bem como métricas para avaliar o desempenho dos módulos de planejamento de voo.

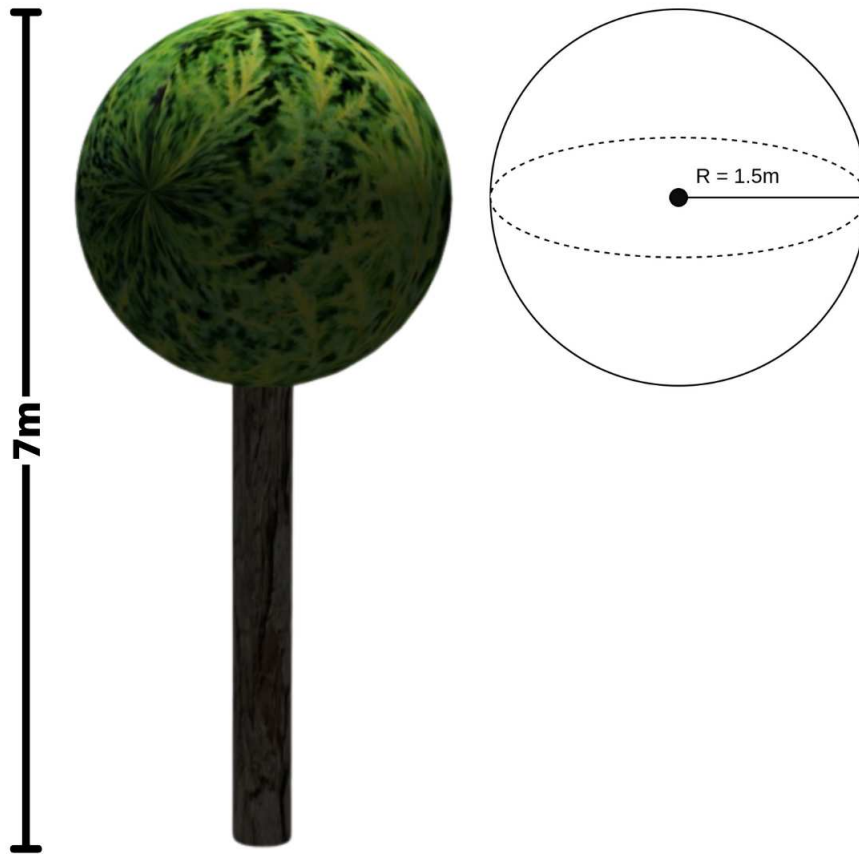
3.1.1 Cenários de Testes

Para testar a eficácia do planejamento de caminhos para VANTs em diferentes cenários, foram considerados três cenários distintos, todos com árvores idênticas, mas com distribuições de altura variadas. Esses cenários permitem uma avaliação abrangente da robustez e adaptabilidade do sistema de planejamento de caminhos em diversas condições de vegetação.

O obstáculo em questão é representado por uma árvore modelada no Gazebo, e possui uma altura de 7 metros. Além disso, sua copa forma uma esfera com um raio de 1.5 metros. Isso significa que, em sua parte mais larga, a copa da árvore se estende horizontalmente a 3 metros de distância. Essa extensão horizontal da copa da árvore impõe limitações ao espaço

disponível para a navegação do VANT, tornando essencial a detecção e evitação dessa estrutura durante o planejamento de caminhos. A Figura 9 exibe os detalhes do tamanho e raio da árvore utilizada como obstáculo nos cenários.

Figura 9 – Árvore utilizada como obstáculo para navegação do VANT

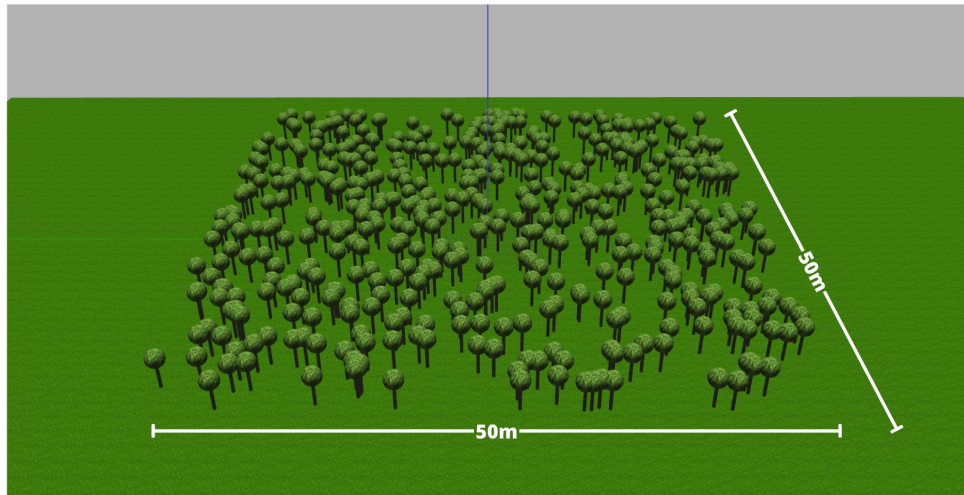


Fonte: O próprio autor.

A montagem dos cenários envolveu um processo de geração de ambientes simulados usando um código Python personalizado. Inicialmente, um arquivo base no formato *.world* do Gazebo - *forest.world*, que está disponível de forma padrão nos pacotes de mundo do MRS UAV System foi preparado, contendo árvores com altura uniforme de 7 metros. Esse cenário possui um total de 500 árvores distribuídas em uma área de 50m x 50m. O cenário base pode ser visualizado através da Figura 10. Em seguida, foi desenvolvido um código em Python que utilizou a biblioteca NumPy para aplicar a distribuição normal às alturas das árvores nesse ambiente base. Esse código gerou as alturas das árvores de acordo com os parâmetros desejados, como a média (μ) e o desvio padrão (σ) da distribuição normal, criando assim os diferentes cenários de teste.

O cenário base serviu como o ponto de partida para derivar os outros três cenários

Figura 10 – Cenário base



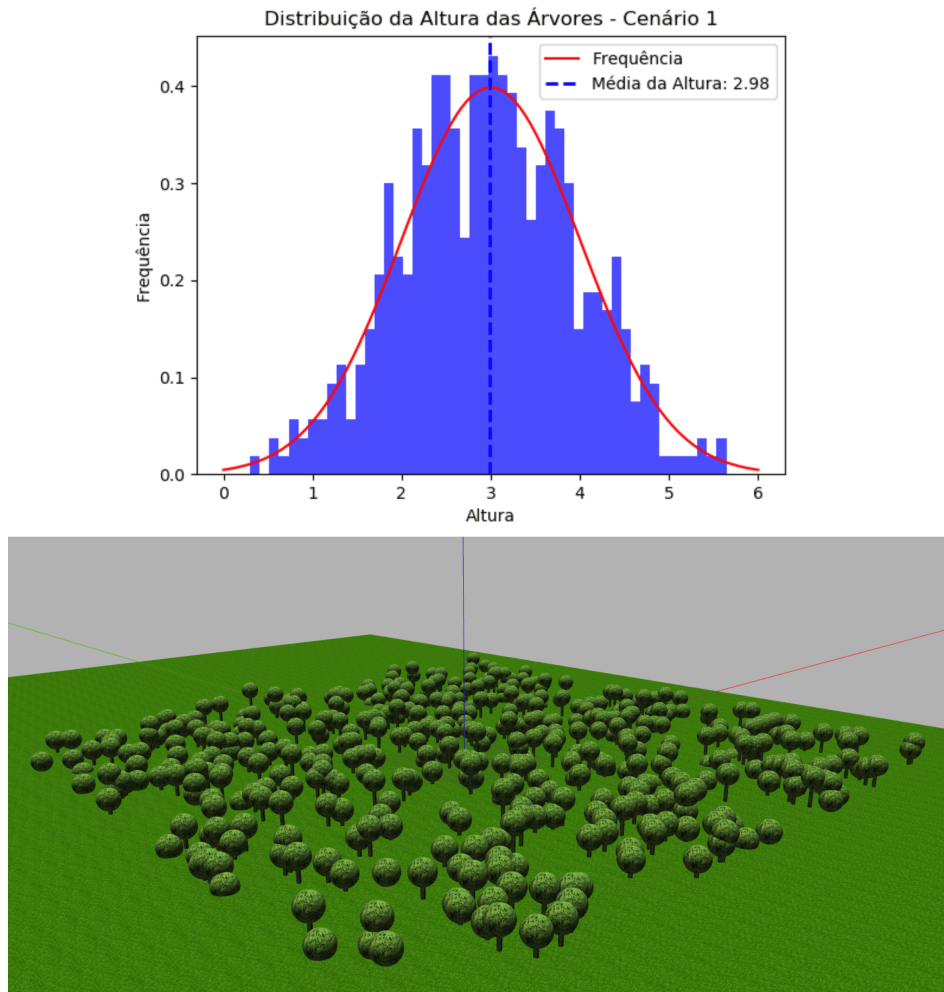
Fonte: O próprio autor.

que serão utilizados neste trabalho. A abordagem de utilizar a geração das alturas das árvores permitiu modelar a complexidade tridimensional do ambiente, fornecendo aos VANTs um desafio mais realista na navegação e na detecção de obstáculos. A escolha de montar os cenários derivados do cenário base é fundamentada na necessidade de proporcionar uma representação realista do ambiente, onde os obstáculos não se limitam a simples troncos de árvores, mas também incluem suas copas, resultando em um desafio mais completo de navegação. Esse cenário estabelece um contexto mais próximo do mundo real, permitindo avaliar o desempenho do VANT em condições mais representativas e, assim, contribuir para resultados mais fidedignos e relevantes na pesquisa.

No cenário 1, representado pela Figura 11, as alturas das árvores variam de 0.0 m (pequenos arbustos) a 6.0 m, seguindo uma distribuição normal com média (μ) igual à metade da soma da altura máxima ($\max z$) e da altura mínima ($\min z$), ou seja, $\mu = (\max z + \min z)/2 = 3$, e um desvio padrão (σ) de 1.0. Esse cenário permite avaliar a capacidade do sistema de planejamento de trajetória de lidar com obstruções mais altas e variações mais significativas na altura da vegetação, replicando situações realistas onde o VANT precisa navegar através de áreas mais densas e heterogêneas.

Já no cenário 2, representado através da Figura 12, as alturas das árvores variam de 2.0 m a 4.0 m, seguindo uma distribuição normal com média (μ) igual à metade da soma da altura máxima ($\max z$) e da altura mínima ($\min z$), ou seja, $\mu = (\max z + \min z)/2 = 3$, e um

Figura 11 – Cenário de Teste 1



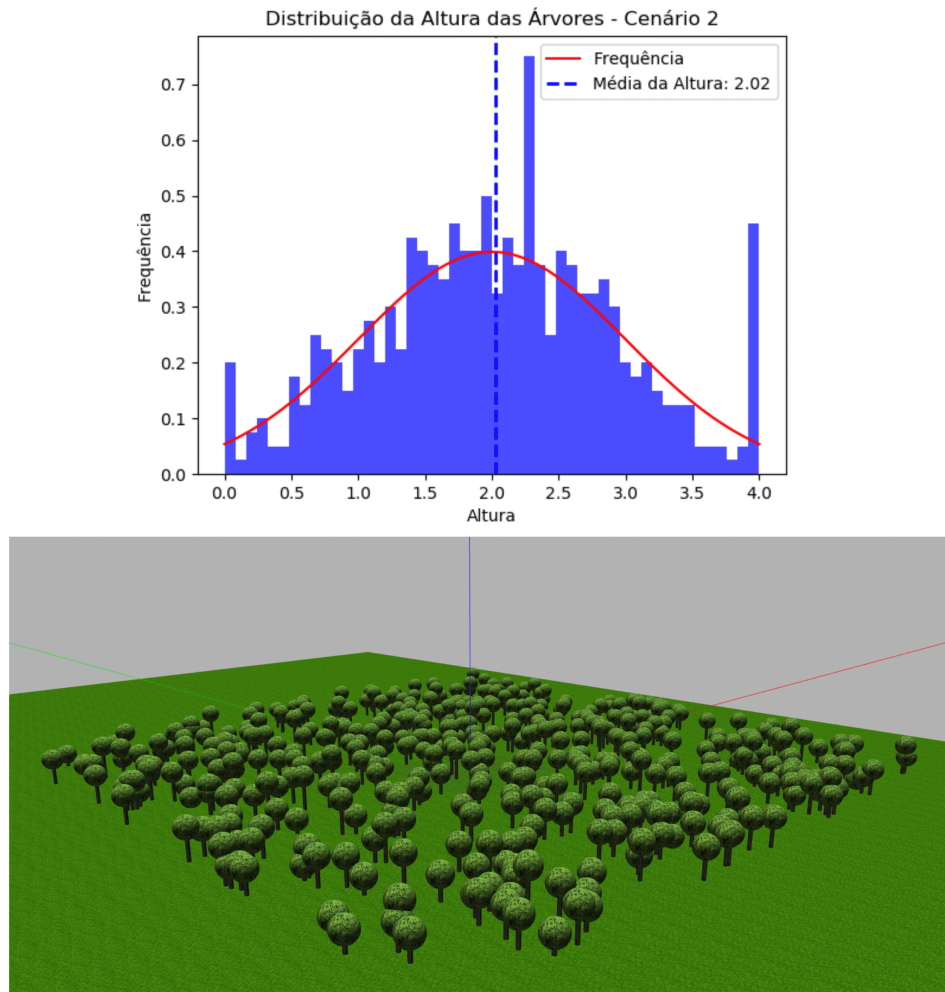
Fonte: O próprio autor.

desvio padrão (σ) de 1.0. Esse cenário é projetado para simular um ambiente de média densidade vegetal, onde as alturas das árvores variam de forma relativamente homogênea, permitindo avaliar como o sistema de planejamento de trajetória lida com obstruções moderadas e desafios de navegação em áreas com poucas variações na altura da vegetação.

Por fim, no cenário 3, representado pela Figura 13, as alturas das árvores variam de 3.5m a 6.0m, seguindo uma distribuição normal com média (μ) igual à metade da soma da altura máxima ($\max z$) e da altura mínima ($\min z$), ou seja, $\mu = (\max z + \min z)/2 = 3.25$, e um desvio padrão (σ) de 0.6. Este cenário é projetado para representar um ambiente de baixa densidade vegetal, onde as alturas das árvores são relativamente consistentes, mas com uma variação menor.

O cenário 3 é particularmente relevante para avaliar a capacidade do algoritmo de Aprendizado por Reforço em aprender a estratégia de navegar em alturas mais baixas para atingir o objetivo de forma eficaz. Com a predominância de troncos mais altos, o algoritmo deve ser

Figura 12 – Cenário de Teste 2

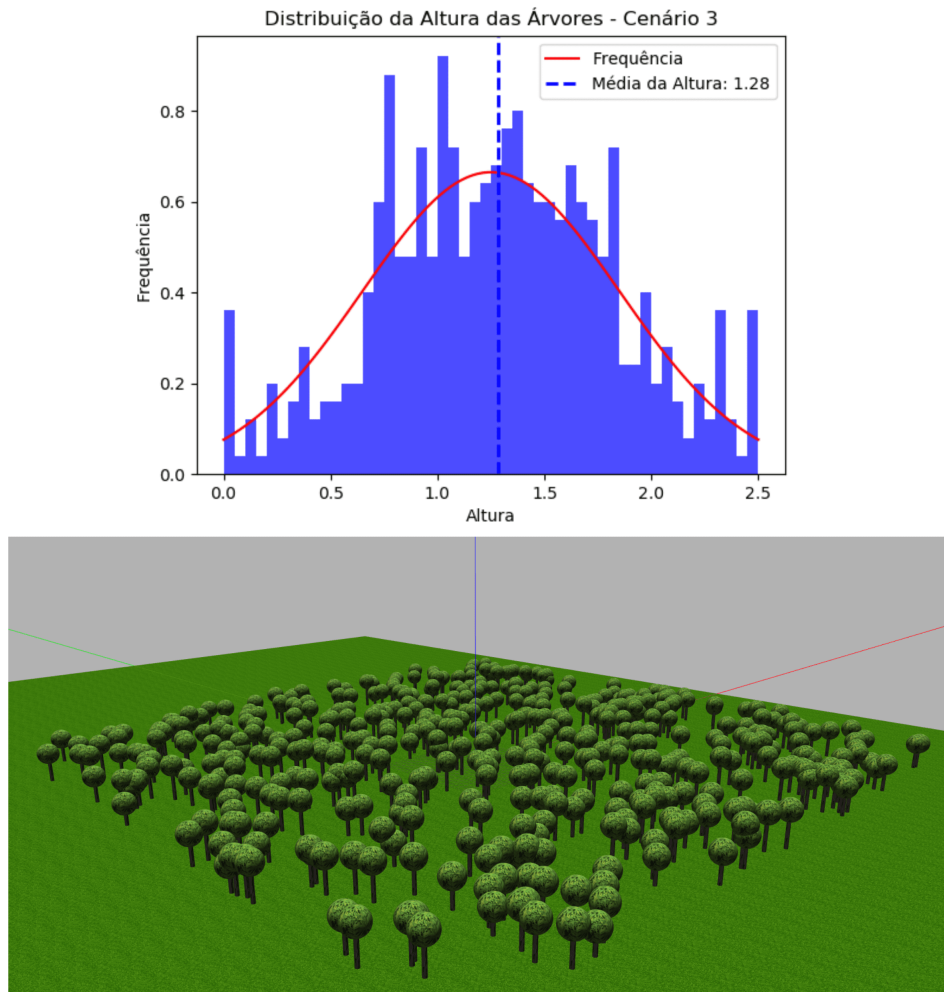


Fonte: O próprio autor.

incentivado a escolher trajetórias que minimizem a altitude do VANT, levando em consideração o perfil do ambiente. Isso proporciona uma oportunidades para verificar se o algoritmo é capaz de aprender a evitar obstáculos em alturas superiores, otimizando sua trajetória para maximizar a segurança e eficiência nas missões de navegação. Através da análise do desempenho do algoritmo de RL neste cenário, é possível verificar se ele é capaz de aprender e adaptar suas políticas de controle de forma eficiente para otimizar a altitude do VANT, atingindo assim os objetivos de navegação com sucesso, minimizando riscos e maximizando a eficácia.

A escolha destes cenários para os testes se deu por possuir objetos (modelos de árvores dispostas ao longo do mundo) que atrapalham o percurso do VANT e os parâmetros referentes ao motor físico do mundo estão em fidelidade com a realidade.

Figura 13 – Cenário de Teste 3



Fonte: O próprio autor.

3.1.2 Experimentos

Os experimentos têm como foco principal a avaliação do desempenho das abordagens de planejamento de caminhos, considerando a complexidade dos cenários propostos. Cada experimento envolve a definição de pontos de saída e chegada, localizados em regiões isentas de obstáculos. No entanto, é importante observar que o caminho entre esses pontos de referência possui uma série de obstáculos, representando árvores e suas copas, que irão testar a capacidade de navegação autônoma do VANT.

A tabela abaixo contém uma série de testes de navegação de VANTs em um ambiente simulado, onde são especificados pontos de saída (start), pontos intermediários e pontos de chegada (goal) para cada teste. Cada linha representa um teste individual e inclui coordenadas (x, y, z) em ponto flutuante para esses pontos. A inclusão de pontos intermediários entre o ponto

de saída e o ponto de chegada permite criar trajetórias mais complexas e desafiadoras para o VANT.

Tabela 2 – Experimentos de navegação

Teste	Ponto de Saída (Start)	Ponto de Chegada (Goal)	Distância em Linha Reta (m)
1	(0.0, 0.0, 1.2)	(-8.0, 7.0, 3.0)	10.78
2	(0.0, 0.0, 2.0)	(6.5, -3.0, 2.0)	7.16
3	(0.0, 0.0, 2.5)	(18.5, 10.0, 2.0)	21.04
4	(0.0, 0.0, 2.5)	(-23.5, 5.0, 1.5)	24.05
5	(0.0, 0.0, 2.5)	(17.5, 10.0, 1.0)	20.21
6	(6.5, -3.0, 2.5)	(17.5, 10.0, 1.0)	17.10
7	(6.5, -3.0, 2.0)	(18.5, 12.5, 1.5)	19.61
8	(6.5, -3.0, 1.5)	(-10.5, 5.0, 1.5)	18.79
9	(6.5, -3.0, 2.5)	(-13.0, 6.0, 3.0)	21.48
10	(6.5, -3.0, 1.5)	(-16.0, 15.0, 2.0)	28.82
11	(10.5, -2.0, 2.5)	(18.5, 12.5, 1.5)	16.59
12	(10.5, -2.0, 2.5)	(6.5, -3.0, 2.0)	4.15
13	(10.5, -2.0, 2.5)	(-8.0, 7.0, 3.0)	20.58
14	(-24.0, 5.0, 2.5)	(10.5, 20.0, 2.5)	37.62
15	(-24.0, 5.0, 2.5)	(10.5, -2.0, 2.0)	35.21

Fonte: O próprio autor.

Os testes foram planejados para cobrir diversas localidades nos cenários, englobando áreas distintas para navegação autônoma. Além disso, os pontos de saída e chegada foram selecionados para abranger um intervalo de distâncias entre aproximadamente 4 metros e 40 metros. Cada um dos 15 experimentos planejados será realizado com o uso de ambas as abordagens de planejamento de caminhos, ou seja, a abordagem clássica baseada no algoritmo A* e a abordagem moderna que faz uso do aprendizado por reforço. Esses experimentos serão realizados nos três cenários propostos, totalizando assim um conjunto abrangente de avaliações. Além disso, esses experimentos serão conduzidos nos três cenários propostos, totalizando 15 experimentos multiplicados por duas abordagens e pelos três cenários, tendo um total de 90 testes realizados.

O objetivo desses testes é avaliar o desempenho de métodos de planejamento de voo propostos neste trabalho - A* e aprendizagem por reforço, em cenários com múltiplos pontos de referência, com métricas definidas anteriormente para cada trajetória. Portanto, ao planejar a trajetória do VANT para alcançar esses pontos, é essencial que os sistemas de navegação a serem avaliados sejam adaptáveis e capazes de tomar decisões em tempo real, garantindo que o VANT possa navegar de forma segura e eficiente.

3.1.3 Coleta da Posição do VANT

Para obter os dados referentes às coordenadas do VANT a partir das trajetórias dos experimentos, foi necessário desenvolver um módulo de coleta de dados. Esse módulo foi responsável por registrar e armazenar as coordenadas espaciais do VANT em função do tempo ao longo de cada experimento. Esses dados tornaram-se essenciais para a avaliação das abordagens de planejamento de caminhos clássica e moderna, conforme métricas definidas na metodologia deste trabalho, contribuindo a validação dos modelos implementados.

O módulo de coleta de posição do VANT, denominado "PositionCollector," é responsável por obter e armazenar os dados de posição em tempo real, permitindo uma análise posterior das trajetórias e movimentos do VANT. O programa estabelece uma taxa de atualização de 10 Hz, (ou seja, o módulo de captura da posição e *timestamp* realiza cerca de 10 coletas em 0.1 segundos de simulação) para controlar a frequência da coleta de dados. Em seguida, os dados são registrados em um arquivo *Comma Separated Values* (CSV) separado por vírgulas, incluindo o tempo de simulação do ROS para rastrear a temporalidade das posições.

A Figura 14 exibe uma parte do arquivo gerado pelo módulo coletor, sendo que primeira coluna representa o tempo de simulação e as colunas 2, 3 e 4 correspondem as coordenadas x , y , e z respectivamente:

Figura 14 – Exemplo de parte do arquivo gerado pelo módulo de coleta das coordenadas do VANT

```

329 54.5,0.3626711551124864,0.36266247053918266,1.5001091136403122
330 54.508,0.3646868333032464,0.36467611735891375,1.5001013583005447
331 54.52,0.3677102943827292,0.36769615925172083,1.5000899429693786
332 54.528,0.3697257315521297,0.36970951809581043,1.5000828239668418
333 54.54,0.37274839287574063,0.3727294982457164,1.5000727705222527
334 54.548,0.37476326574240676,0.37474264976793326,1.5000660840243005
335 54.56,0.3777857332220175,0.3777622191437434,1.5000557916551287
336 54.568,0.3798014187562863,0.3797755267596352,1.500048774859104
337 54.58,0.38282615115320456,0.3827963100155472,1.5000384512505553
338 54.588,0.3848432782478029,0.38481099146555386,1.5000321646813755
339 54.6,0.38786962044681067,0.38783409624421095,1.5000236308634436
340 54.608,0.3898875241162446,0.3898499299606933,1.500018234154642
341 54.62,0.39291525727678783,0.3928742916614398,1.5000102224471537

```

Fonte: O próprio autor.

3.1.4 Métricas para avaliar os testes

A validação dos resultados será conduzida por meio da análise dos testes, apresentando os resultados por meio de gráficos e tabelas. Além disso, serão empregadas métricas

específicas para avaliar os módulos de planejamento de voo, incluindo:

- **Distância Percorrida:** A distância total percorrida pelo VANT para alcançar o destino em cada teste;
- **Tempo de Missão:** O tempo total que o VANT leva para completar uma missão, desde o momento da saída do primeiro ponto até a chegada ao destino;
- **Velocidade Média:** A velocidade média do VANT ao longo de toda a missão;
- **Altitude Média:** Média em metros da altitude do VANT na missão;
- **Variação de Altitude:** Diferença entre a altitude máxima e mínima na missão;
- **Taxa de sucesso do VANT em alcançar seu destino com segurança no ambiente:** A porcentagem de vezes que o VANT alcança seu destino com segurança, sem colidir com obstáculos. É dado por:

$$TaxadeSucesso(\%) = (NumerodeTestesBemSucedidos/TotaldeTestes) * 100 \quad (3.1)$$

3.2 MATERIAIS DE DESENVOLVIMENTO

Neste trabalho, a modelagem das abordagens propostas foi fundamentada no uso de ferramentas essenciais para o desenvolvimento do projeto. As principais áreas de enfoque foram a simulação e a codificação dos comportamentos a serem utilizados pelos VANTs.

As simulações são ferramentas importantes no desenvolvimento de sistemas autônomos como VANTs, pois evita o desgaste desnecessário e os custos associados aos testes em ambientes reais. Além disso, as simulações proporcionam um ambiente controlado para avaliar o desempenho de algoritmos, estratégias de planejamento de caminhos e sistemas de percepção, sem o risco de danificar o equipamento físico. Isso se traduz em economias de recursos, em especial financeiro, tempo e esforço, tornando as simulações uma etapa fundamental no desenvolvimento de VANTs e outras aplicações autônomas. Para este propósito, utilizou-se o MRS UAV System, uma plataforma que combina software e hardware para executar simulações realistas. O software é baseado na integração do ROS com o Gazebo Simulator, proporcionando um ambiente simulado para testes e experimentos.

O hardware utilizado para as simulações e a codificação das abordagens de planejamento de caminhos consistiu em um computador tipo *desktop* com as seguintes especificações:

- **Processador:** AMD Ryzen 5 5600

- Placa de vídeo: NVIDIA GTX 1060
- Memória RAM: 16GB DDR4 3200MHz
- Sistema Operacional: Ubuntu 22.04 LTS

Essas configurações forneceram um suporte para a execução das simulações no ambiente ROS e Gazebo Simulator. O sistema operacional Ubuntu 22.04 LTS foi escolhido devido à sua compatibilidade com o ambiente ROS 1 na versão Noetic Ninjemys, que é amplamente utilizado em aplicações robóticas. Essa combinação de hardware e software proporcionou um ambiente de desenvolvimento para o projeto, permitindo simulações que as simulações e experimentos fossem executados para desenvolver e avaliar as abordagens de planejamento de caminhos propostas.

Os códigos desenvolvidos durante a execução do projeto foram implementados nas linguagens de programação Python e C++. A escolha do Python como uma das linguagens se justificou devido à sua ampla e ativa comunidade de desenvolvedores, juntamente com uma vasta coleção de bibliotecas prontamente disponíveis. A flexibilidade do Python também foi um fator decisivo, pois essa linguagem pode ser utilizada tanto na construção de algoritmos simples como no desenvolvimento de sistemas altamente complexos. Isso se traduziu em uma redução no tempo necessário para aprender e implementar as tecnologias necessárias para o projeto.

Por outro lado, o C++ foi escolhido especificamente para o cálculo do algoritmo A* devido à sua alta *performance* e eficiência em tarefas computacionais intensivas. A utilização de C++ para essa finalidade permite que o algoritmo A* seja executado de forma otimizada, garantindo tempos de resposta rápidos e eficiência no planejamento de caminhos para os VANTs. A integração entre as partes desenvolvidas em Python e C++ foi realizada de forma a aproveitar as vantagens de ambas as linguagens, combinando a facilidade de desenvolvimento e recursos de Python com o desempenho robusto de C++. Essa abordagem híbrida permitiu alcançar um equilíbrio ideal entre eficiência e facilidade de desenvolvimento do projeto.

3.3 F450

Para a implementação e testes práticos das abordagens de planejamento de caminhos neste projeto, foi adotado o modelo VANT DJI F450, o qual foi equipado com o sensor RPLidar para detecção de obstáculos. O F450 é uma plataforma de VANT amplamente reconhecida por sua versatilidade e desempenho, o que o torna uma escolha sólida para aplicações de pesquisa

e desenvolvimento. A integração do sensor RPLidar proporcionou ao VANT a capacidade de perceber e reagir a obstáculos em seu ambiente, contribuindo para a segurança e eficácia do planejamento de caminhos. Esta configuração permitiu a realização de testes práticos realistas e a validação das abordagens propostas.

O modelo de VANT F450 é um quadricóptero popular de tamanho médio desenvolvido pela empresa DJI¹, líder no mercado de VANTs comerciais e de consumo. O F450 possui uma construção robusta, design modular (com peças que se encaixam para a montagem do VANT, como os motores, hélices e os braços de suporte) e capacidade de personalização com a utilização de sensores de diversos tipos como o GPS, RTK, câmeras e RPLidar. O modelo do F450 simulado no Gazebo Simulator pode ser visto na Figura 15

Figura 15 – DJI F450 simulado (MRS UAV System)



Fonte: O próprio autor.

Em geral, esse modelo de VANT possui utilização para aplicações comerciais e de pesquisa. Neste trabalho os módulos de planejamento de caminhos clássicos e através do uso da aprendizagem por reforço serão testados utilizando o VANT DJI F450, dispondo dos mesmos parâmetros relacionados a característica física do modelo, tais como a massa, tamanho do corpo e de peças como o braço e pernas, além da distância e massa dos motores. Estes parâmetros estão disponíveis na tabela 3

¹ <<https://www.dji.com/>>

Tabela 3 – Parâmetros das características físicas do F450

Parâmetro	Valor	Parâmetro	Valor
Massa	2.0 kg	Massa do rotor	0.005 kg
Altura dos rotores	0.0235 m	Comprimento do braço	0.23 m
Altura do corpo	0.1 m	Raio do rotor	0.118 m

Fonte: O próprio autor.

3.3.1 Percepções

Para realização da implementação do módulo de planejamento de caminhos, o agente precisa receber dados e informações sobre o ambiente em que ele está inserido, a fim de tomar decisões e realizar ações conforme o estado naquele instante. O VANT utilizado é equipado com três sensores distintos para receber percepções sobre o ambiente: RPLidar, Garmin Range e o GPS.

O RPLidar é um sensor bastante utilizado em aplicações de mapeamento e navegação autônoma. Baseado em tecnologias que usam o laser, o RPLidar é capaz de realizar varreduras em 360 graus ao seu redor, detectando obstáculos e coletando dados precisos sobre a distância e a localização desses objetos. Essas informações permitem que o VANT obtenha percepções sobre os obstáculos ao seu redor, a fim de evitar colisões com os obstáculos. Além disso, o RPLidar possui a vantagem de ser compacto e leve, permitindo a integração em VANTs como o F450, sem comprometer significativamente o peso ou a capacidade de voo da aeronave. O sensor possui capacidade de varredura em 360 graus, oferecendo uma visão abrangente do ambiente circundante, tornando-o eficaz na detecção de obstáculos em diferentes direções. No entanto, é importante mencionar que o alcance máximo do laser do RPLidar é limitado a 14 metros. Isso significa que a percepção do cenário pelo VANT é restrita a essa distância máxima, o que pode influenciar a sua capacidade de detecção de obstáculos em áreas mais distantes.

O Garmin Range é um sensor utilizado para estimar a altitude do VANT de forma precisa em cenários em que há um solo que não obstrua a luz. Esse dispositivo utiliza tecnologia de laser para medir a distância entre o VANT e o solo em tempo real. Com base nessas medições, o Garmin Range é capaz de fornecer uma estimativa precisa da altitude em que o VANT está voando. Uma alternativa ao Garmin Range é o uso do barômetro, que utiliza a pressão atmosférica para estimar a altura do VANT. Inclusive o barômetro pode ser utilizado em voos sob o mar (o laser se dispersa na água) e voos com alturas elevadas. Neste projeto foi utilizado o Garmin Range, pois o solo dos cenários são propícios à utilização do sensor, e o VANT não voará em

uma altura maior do que o limite em que o laser não alcance o solo.

Além do sensor RPLidar para detecção de obstáculos e o Garmin Range para estimativa de altura, também foi embarcado no F450 o uso de um sistema de posicionamento global para realizar a estimativa lateral para o VANT. A escolha foi na utilização do *Global Positioning System* (GPS), pois esse sensor fornece informações sobre a posição do VANT em relação às coordenadas geográficas, permitindo uma estimativa precisa da localização lateral da aeronave durante as simulações. Essa estimativa é fundamental no planejamento de caminhos, uma vez que ajuda a garantir que o VANT siga trajetórias precisas e atinja os pontos de destino com alta acurácia.

Essa combinação de tecnologia em sensoriamento, incluindo o RPLidar para detecção de obstáculos, o Garmin Range para estimativa da altura, e o sistema de GPS para precisão lateral, representa uma abordagem abrangente e eficaz para aplicações de planejamento de caminhos em VANTs. A interação entre esses sensores permite que o VANT obtenha a percepção do ambiente, enquanto mantém informações importantes sobre sua posição, altitude e sobre os obstáculos ao seu redor.

4 FS PATH PLANNING

Neste capítulo é explorado o desenvolvimento de soluções para o problema de planejamento de caminhos eficientes para VANTs. O objetivo deste capítulo é fornecer um detalhamento e análise de duas abordagens distintas para o planejamento de caminhos, abrangendo tanto a abordagem clássica, representada pelo algoritmo VANTs, quanto a abordagem moderna, baseada em Aprendizado por Reforço.

Para alcançar esse objetivo, são explorados detalhes de desenvolvimento e implementação de soluções. Este capítulo aborda as tecnologias, ferramentas e métodos empregados para a construção das abordagens de planejamento de caminhos. Isso inclui a descrição das linguagens de programação utilizadas, como Python e C++, e a integração com o ambiente do ROS e o Gazebo Simulator, que forneceram a base para as simulações e experimentos.

Além disso, serão abordados aspectos relacionados à modelagem das percepções na tomada de decisões autônomas dos VANTs. A integração de sensores como o RPLidar e o GPS, juntamente com a implementação de algoritmos de segmentação e conversão de percepções 2D em 3D, contribuiu para uma percepção tridimensional mais precisa do ambiente.

4.1 MODELAGEM DE PERCEPÇÕES 2D EM 3D

Neste trabalho, foi desenvolvida uma abordagem para a modelagem de percepções de obstáculos em duas dimensões e sua extensão para o espaço tridimensional. Essa modelagem visa proporcionar uma maneira eficiente e econômica de extrair informações tridimensionais sobre os obstáculos, a partir do sensor RPLidar - superando as limitações financeiras associadas à aquisição de sensores 3D mais custosos. Essa abordagem expandiu as possibilidades para o planejamento de caminhos e navegação em ambientes tridimensionais, oferecendo uma solução acessível para os módulos de planejamento de caminho. Este modelo visa aprimorar a capacidade de detecção e navegação de VANTs, tornando-os mais eficazes e seguros nos cenários propostos, ao mesmo tempo em que otimiza os recursos financeiros ao evitar a necessidade de sensores 3D mais caros.

O modelo desenvolvido neste projeto introduz uma abordagem modelagem dos

obstáculos, que são as árvores - com seus troncos representados como cilindros e copas na forma de esferas. O processo envolve a criação de uma esfera projetada que se ajusta aos dados de sensor obtidos do RPLidar, o componente que realiza a detecção de obstáculos.

A primeira etapa do processo consistiu em utilizar uma função de minimização de erro para otimizar o círculo gerado com base nos dados do sensor. A estimativa inicial do centro e raio do círculo é obtida através da média das coordenadas dos pontos do sensor. Em seguida, um processo de minimização é utilizado para realizar a estimativa do centro e raio do círculo, resultando em um círculo ajustado que representa com maior precisão a forma do obstáculo detectado. Essa metodologia visa melhorar a qualidade da percepção tridimensional dos obstáculos para o agente, proporcionando uma percepção aprimorada para o planejamento de caminhos.

Para realizar a modelagem da percepção 2D em 3D, foi essencial a utilização das percepções fornecidas pelo RPLidar. Essas percepções são representadas como vetores que descrevem a localização dos obstáculos em relação ao VANT. Na Figura 16, temos uma representação visual da percepção de obstáculos pelo VANT com base nos dados do RPLidar. O ponto destacado em vermelho, está a representação do VANT, que serve como ponto de referência para o RPLidar. Podemos observar uma série de pontos em preto que representam os vetores que descrevem a localização dos obstáculos detectados pelo RPLidar com relação ao VANT.

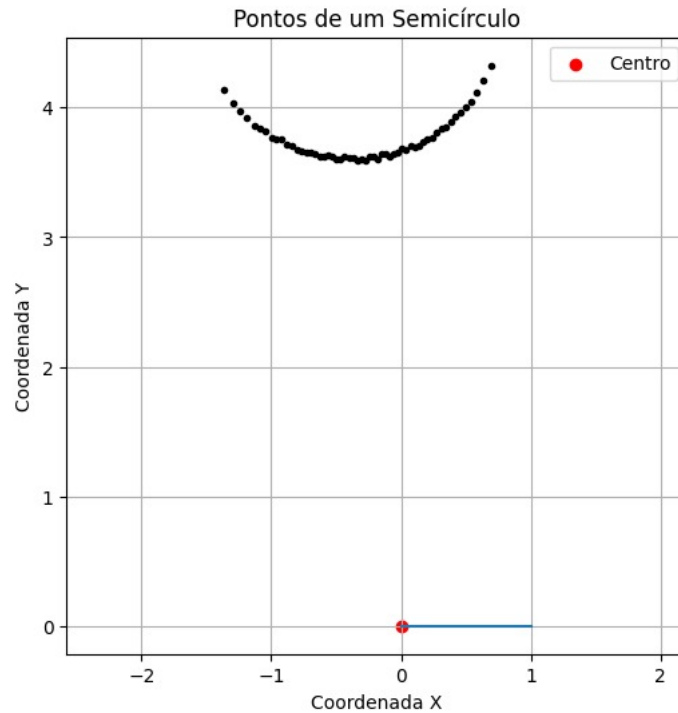
Para prosseguimento do modelo proposto, foi utilizado a função de erro, que é comumente usada em problemas de otimização, como ajuste de modelos matemáticos a dados experimentais. No contexto deste trabalho, a função apresentada é usada para ajustar um círculo aos pontos de dados coletados do sensor RPLidar. Neste caso, o erro é calculado como a soma dos quadrados das diferenças entre a distância de cada ponto de dados (x, y) ao centro estimado do círculo (cx, cy) e o raio estimado r menos o raio real ao quadrado r^2 . Isso é feito para cada ponto de dados no conjunto.

$$E = \sum_{i=1}^N ((x_i - cx)^2 + (y_i - cy)^2 - r^2)^2 \quad (4.1)$$

em que:

- E é o valor do erro,

Figura 16 – Pontos de um semicírculo gerado pelos dados do RPLidar



Fonte: O próprio autor.

- N é o número de pontos na amostra,
- x_i e y_i representam as coordenadas do i -ésimo ponto da amostra,
- cx e cy são as coordenadas do centro do círculo,
- r é o raio do círculo.

Neste caso, a função de erro mede a assimetria entre o círculo estimado e os pontos de dados, calculando a soma dos quadrados das diferenças entre as distâncias dos pontos ao centro estimado do círculo e o raio estimado, menos o raio real ao quadrado. A minimização dessa função foi utilizado por um algoritmo de otimização para encontrar os valores ótimos dos parâmetros do círculo, resultando em um círculo que melhor se ajusta aos dados. Os parâmetros do centro e do raio do círculo são otimizados de modo a reduzir o valor da função de erro, garantindo assim um ajuste mais preciso aos dados da amostra.

A primeira etapa do processo de projeção das esferas em 3 dimensões consiste no processo de ajuste do círculo aos dados, em que é realizada a estimativa inicial do centro (cx_{inicial} e cy_{inicial}) e do raio (r_{inicial}) do círculo com base nos pontos de dados. Isso fornece um ponto de partida para a otimização subsequente. As estimativas iniciais são calculadas da seguinte forma:

O centro inicial do círculo (cx_{inicial} e cy_{inicial}) é obtido calculando as médias das

coordenadas X e Y dos pontos da amostra:

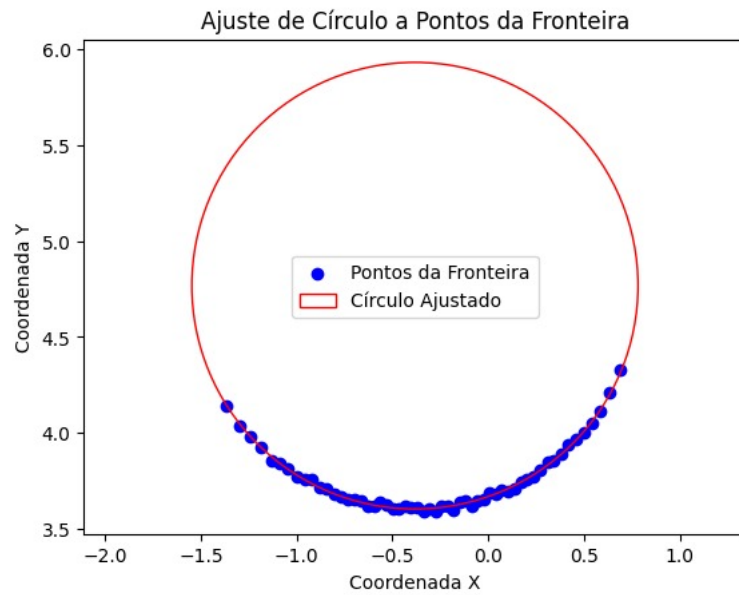
$$cx_{\text{inicial}} = \frac{1}{N} \sum_{i=1}^N x_i \quad \text{e} \quad cy_{\text{inicial}} = \frac{1}{N} \sum_{i=1}^N y_i \quad (4.2)$$

O raio inicial (r_{inicial}) é estimado como a média das distâncias dos pontos ao centro estimado:

$$r_{\text{inicial}} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - cx_{\text{inicial}})^2 + (y_i - cy_{\text{inicial}})^2} \quad (4.3)$$

Essas estimativas iniciais fornecem um ponto de partida para a otimização posterior da função de erro. A Figura 17 ilustra o ajuste - completção do círculo a partir dos dados de entrada do RPLidar.

Figura 17 – Ajuste do círculo a partir do semicírculo gerado pelos dados do RPLidar



Fonte: O próprio autor.

Na segunda etapa, temos a minimização da função de erro. O objetivo é encontrar os parâmetros otimizados do círculo que melhor se ajustam aos dados da amostra. Para realizar essa minimização, é utilizado o método de otimização *minimize*, disponível na biblioteca Scipy do Python, que busca os valores de parâmetros ($cx_{\text{otimizado}}$, $cy_{\text{otimizado}}$ e $r_{\text{otimizado}}$) que minimizam a função de erro E .

O problema de otimização é formulado da seguinte maneira:

$$(\hat{c}x, \hat{c}y, \hat{r}) = \arg \min_{cx, cy, r} E(cx, cy, r) \quad (4.4)$$

onde: - $(\hat{c}x, \hat{c}y, \hat{r})$ são os parâmetros otimizados do círculo, - cx, cy são as coordenadas do centro do círculo, - r é o raio do círculo, e - $E(cx, cy, r)$ é a função de erro definida anteriormente. O método *minimize* utiliza uma técnica de otimização, que no caso deste trabalho foi o método *Limited-memory Broyden-Fletcher-Goldfarb-Shanno Bound* (L-BFGS-B) (59), para encontrar os parâmetros que minimizam a função de erro. Os parâmetros otimizados resultantes, $(\hat{c}x, \hat{c}y, \hat{r})$, representam o centro e o raio do círculo que melhor se ajusta aos dados da amostra. Essa minimização foi utilizada para obter uma estimativa precisa do círculo a partir dos dados da amostra no processo de análise e modelagem das esferas 3D.

Após a obtenção dos parâmetros otimizados do círculo ($cx_{\text{otimizado}}, cy_{\text{otimizado}}, r_{\text{otimizado}}$) que representam a superfície aproximada, é realizada uma etapa de filtragem de obstáculos. Essa etapa tem o propósito de eliminar dados que podem ser considerados ruído e, assim, refinar a representação da superfície do círculo.

A estratégia para a filtragem de obstáculos foi verificar se o raio otimizado ($r_{\text{otimizado}}$) é maior que o valor de limite do raio da árvore (1.5 m), comparando neste caso, o valor 2.0. Essa verificação é realizada da seguinte forma:

$$\text{Se } r_{\text{otimizado}} > 2.0, \text{ então descarte o resultado } (\hat{c}x, \hat{c}y, \hat{r})$$

Essa condição de descarte foi realizada para evitar que pontos que se desviem muito do modelo de círculo sejam considerados parte da superfície. Dessa forma, ruídos ou dados inconsistentes são filtrados, melhorando a representação da superfície aproximada e que resultados indesejados sejam evitados.

Após obter os parâmetros otimizados do círculo ($\hat{c}x_{\text{otimizado}}, \hat{c}y_{\text{otimizado}}, \hat{r}_{\text{otimizado}}$), é possível completar o círculo gerando os pontos que representam a circunferência completa.

Para gerar os pontos do círculo completo, foi utilizado ângulos θ igualmente espaça-

dos ao longo da circunferência. No caso deste trabalho, a quantidade n de pontos igualmente espaçados ao redor do círculo foi igual a 10. Os ângulos θ para cada ponto é calculado da seguinte forma:

$$\theta_i = \frac{2\pi}{n} \cdot i, \quad i = 0, 1, 2, \dots, n-1 \quad (4.5)$$

Usando esses ângulos θ , as coordenadas x_{circulo} e y_{circulo} dos pontos do círculo completo podem ser geradas a partir dos parâmetros otimizados ($\hat{c}x_{\text{otimizado}}$, $\hat{c}y_{\text{otimizado}}$, $\hat{r}_{\text{otimizado}}$) como segue:

$$x_{\text{circulo}} = \hat{c}x_{\text{otimizado}} + \hat{r}_{\text{otimizado}} \cdot \cos(\theta) \quad (4.6)$$

$$y_{\text{circulo}} = \hat{c}y_{\text{otimizado}} + \hat{r}_{\text{otimizado}} \cdot \sin(\theta) \quad (4.7)$$

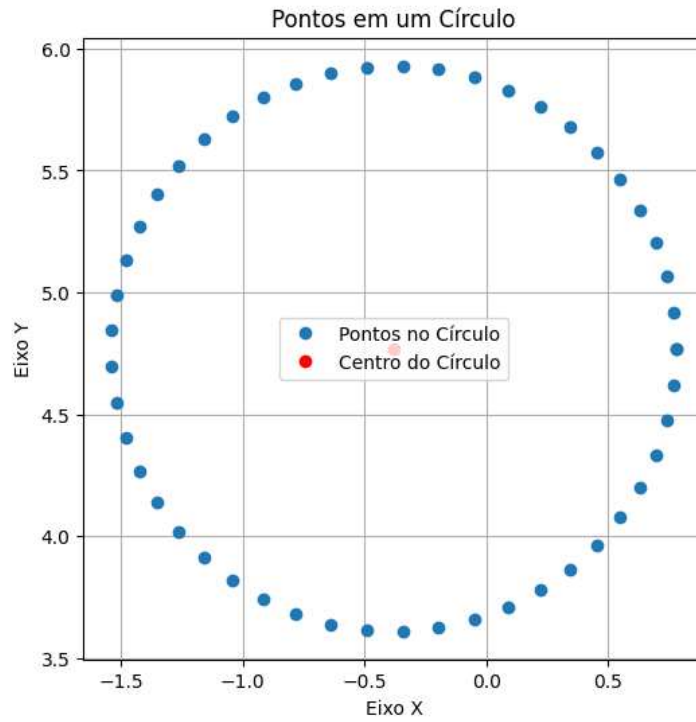
Essas fórmulas permitem a geração dos pontos do círculo completo a partir dos parâmetros otimizados. Os pontos completos representam a circunferência do círculo ajustado, fornecendo uma representação do círculo ajustado. A Figura 18 ilustra o círculo completo a partir dos centros e raio otimizados pelo algoritmo de minimização.

Após a otimização, filtragem de obstáculos e formação do círculo completo, o próximo passo é a geração das coordenadas tridimensionais (x_{pontos} , y_{pontos} , z_{pontos}) que representam a superfície aproximada em um espaço tridimensional. Essas coordenadas tridimensionais são geradas com base nos parâmetros otimizados do círculo ($\hat{c}x_{\text{otimizado}}$, $\hat{c}y_{\text{otimizado}}$, $\hat{r}_{\text{otimizado}}$).

Primeiramente, é utilizado o mesmo parâmetro n de número de pontos que será usado na geração das coordenadas tridimensionais. Em seguida, os ângulos θ e ϕ são gerados com base em n , representando a variação angular em torno do círculo e a inclinação em relação ao plano xy , respectivamente. Isso é feito da seguinte forma:

$$\text{angulos} = [\theta_1, \theta_2, \dots, \theta_n] = \left[0, \frac{2\pi}{n}, \frac{4\pi}{n}, \dots, 2\pi \right] \quad (4.8)$$

Figura 18 – Pontos do círculo ajustado



Fonte: O próprio autor.

$$\text{inclinação} = [\phi_1, \phi_2, \dots, \phi_n] = \left[0, \frac{\pi}{n}, \frac{2\pi}{n}, \dots, \pi \right] \quad (4.9)$$

As coordenadas x_{pontos} , y_{pontos} , e z_{pontos} são então geradas a partir desses ângulos e dos parâmetros otimizados ($cx_{\text{otimizado}}$, $cy_{\text{otimizado}}$, $r_{\text{otimizado}}$) como segue:

$$x_{\text{pontos}} = cx_{\text{otimizado}} + r_{\text{otimizado}} \cdot \sin(\phi) \cdot \cos(\theta) \quad (4.10)$$

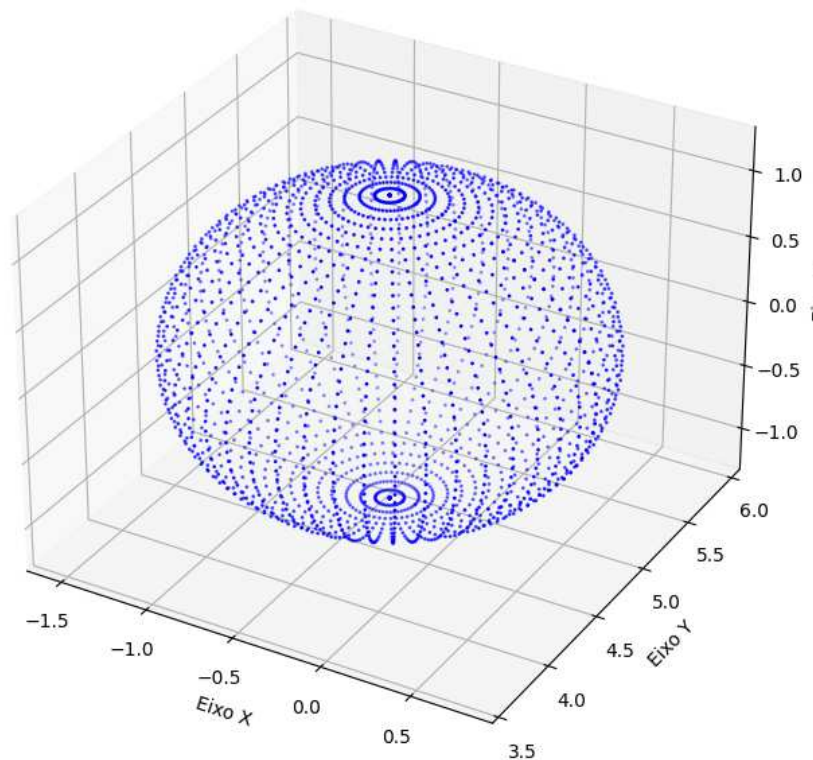
$$y_{\text{pontos}} = cy_{\text{otimizado}} + r_{\text{otimizado}} \cdot \sin(\phi) \cdot \sin(\theta) \quad (4.11)$$

$$z_{\text{pontos}} = r_{\text{otimizado}} \cdot \cos(\phi) \quad (4.12)$$

Essas coordenadas tridimensionais representam uma superfície aproximada por um círculo em um espaço tridimensional. A representação tridimensional pode ser observada na

Figura 19.

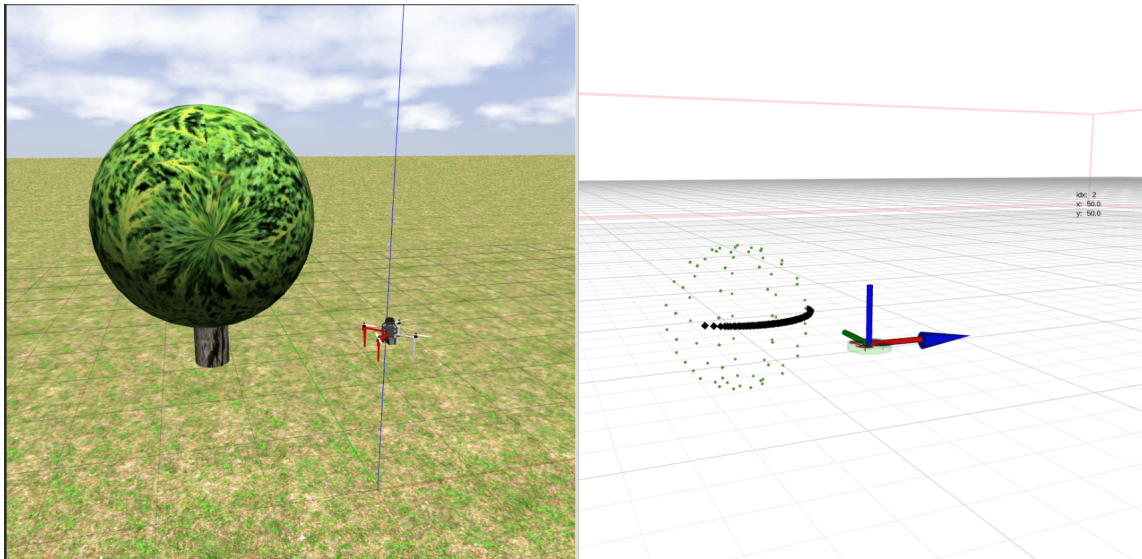
Figura 19 – Projeção da esfera a partir do círculo ajustado
Esfera com Raio Igual ao do Círculo



Fonte: O próprio autor.

A Figura 20 apresenta o cenário de simulação em duas partes. À esquerda, é mostrada a visão do simulador com o VANT sobrevoando uma área que contém uma árvore à frente dele. A representação gráfica do VANT e da árvore é visível, ilustrando o ambiente virtual em que a coleta de dados ocorre. À direita, a imagem exibe a visão do drone a partir do Rviz, com as percepções do sensor 2D RPLidar em preto. A esfera projetada, que representa a superfície aproximada por um círculo em 3D, é visível através dos pontos verdes. Esta esfera é gerada a partir dos dados coletados pelo sensor RPLidar e otimizados através do processo descrito anteriormente. Essa representação visual demonstra a capacidade do sistema em mapear uma superfície aproximada em 3D a partir de um sensor 2D, fornecendo assim, a relação entre o ambiente simulado e a visão capturada pelo drone em tempo real.

Figura 20 – Percepção 3D da árvore



Fonte: O próprio autor.

4.1.1 Segmentação dos obstáculos

Após o desenvolvimento do modelo de percepção 2D em 3D, que foi inicialmente implantado em apenas uma árvore, o próximo desafio consistiu em expandir esse modelo para o cenário mais complexo, que envolveu a presença dos cenários planejados para este projeto, que contém as 500 árvores dispostas de forma aleatória no ambiente. Nesse contexto, foi necessário realizar a segmentação dos obstáculos, ou seja, identificar e delimitar cada uma das árvores presentes no ambiente. Essa etapa representou um passo fundamental e envolveu a identificação e delimitação precisa das diversas árvores presentes no cenário, permitindo que o modelo estendesse sua capacidade de perceber e representar obstáculos tridimensionais em relação ao drone.

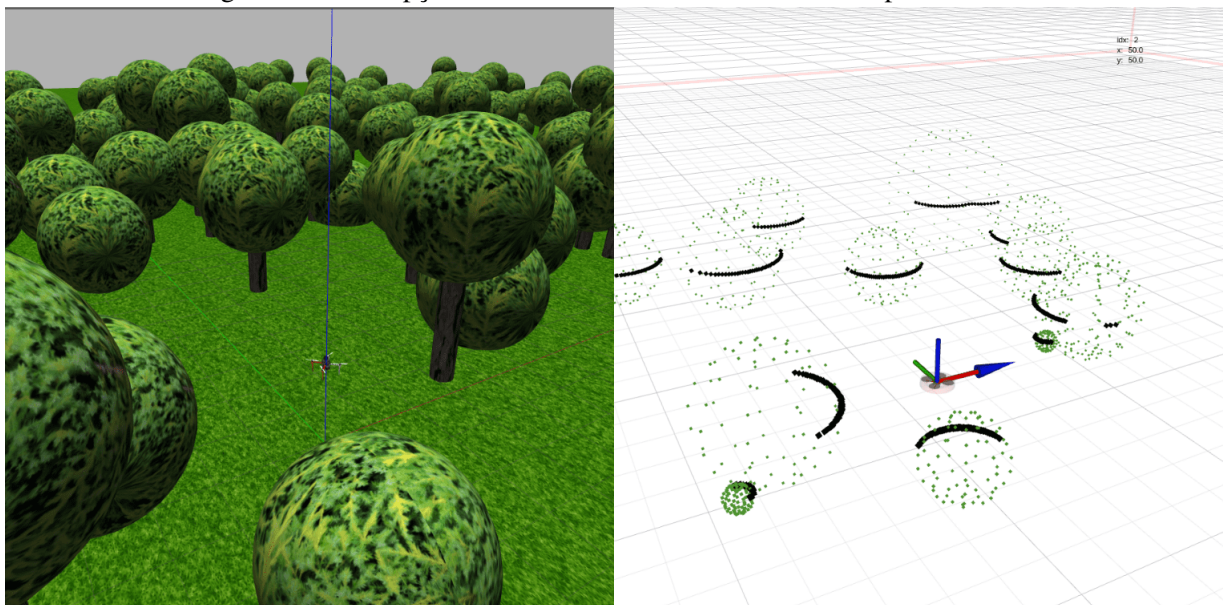
A segmentação das árvores no cenário foi realizada por meio do uso do algoritmo *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). Essa técnica foi aplicada ao conjunto de dados de percepção obtidos pelo RPLidar, onde cada ponto representava uma posição de um obstáculo. O algoritmo DBSCAN, com um valor específico de *epsilon* (eps) configurado em 0.6, foi utilizado para agrupar os pontos em *clusters* que representavam as árvores no ambiente.

O processo consistiu em ajustar o modelo DBSCAN aos pontos de dados, resultando na atribuição de rótulos (*labels*) a cada ponto, identificando a que *cluster* ele pertencia. Os *clusters* foram então organizados de forma apropriada em um dicionário, onde cada chave

correspondia a um rótulo de *cluster* e os valores associados eram os pontos pertencentes a esse *cluster*. Dessa forma, o algoritmo DBSCAN segmentou efetivamente as árvores no cenário, fornecendo agrupamentos de pontos que representavam cada árvore individualmente. Essa abordagem foi necessária para a expansão do modelo de percepção 2D em 3D para um ambiente mais complexo, permitindo uma representação precisa dos obstáculos no cenário com múltiplas árvores.

Na Figura 21 temos representação da simulação do VANT em um cenário cercado por árvores. Podemos observar a aplicação prática do modelo de percepção 2D convertido em percepção 3D. O aspecto mais notável da figura é a representação das percepções 3D, apresentadas como uma nuvem de pontos na cor verde. Essa figura é uma ilustração da capacidade do modelo de percepção desenvolvido em converter percepções 2D em um ambiente tridimensional complexo. Isso é fundamental para permitir que o VANT planeje trajetórias seguras e eficazes, evitando colisões com as árvores e garantindo sua operação sucedida nos cenários propostos. Essa representação tridimensional das percepções demonstra a precisão e a utilidade do modelo de conversão 2D para 3D no contexto de operações de VANT.

Figura 21 – Percepção 3D em um ambiente contendo múltiplas árvores



Fonte: O próprio autor.

A nuvem de pontos verde representa a modelagem tridimensional dos obstáculos presentes no ambiente, neste caso, as árvores. Cada ponto na nuvem de pontos verde representa uma posição específica de um obstáculo em relação ao VANT. Essa representação tridimensional permite que o VANT tenha uma compreensão mais completa do ambiente, pois não apenas

detecta a presença de obstáculos, mas também entende sua posição espacial, tamanho e orientação tridimensional. Ao utilizar as esferas projetadas em 3D em vez de depender apenas de informações bidimensionais (2D) dos obstáculos, o VANT adquire uma compreensão muito mais precisa do cenário, permitindo-lhe calcular trajetórias que não apenas evitem colisões com os obstáculos, mas também otimizem o espaço disponível para navegação. Isso é fundamental em ambientes complexos, onde a simples detecção da presença de obstáculos não é suficiente. Com a capacidade de perceber a forma e o tamanho dos obstáculos em 3D, o VANT pode contornar obstáculos de maneira mais eficiente, maximizando sua segurança e minimizando o desvio desnecessário.

A nuvem de pontos tridimensional descrita anteriormente é publicada em um tópico no ROS denominado `/fspp_classical/spheres_cloud`. Essa nuvem de pontos é composta por informações sobre esferas detectadas no ambiente, onde cada esfera é representada por um formato de mensagem personalizada. As mensagens desenvolvidas e utilizadas para representar as esferas são as seguintes:

- **Nuvem de Pontos do RPLidar - Tópico ROS:** A nuvem de pontos modelada de forma tridimensional, proveniente do sensor RPLidar, é publicada no tópico denominado `/fspp_classical/spheres_cloud`.
- **Formato de Mensagem - Sphere.msg:** A nuvem de pontos é composta por informações sobre as esferas detectadas no ambiente, e cada esfera é representada por uma mensagem personalizada com o seguinte formato:
 - `geometry_msgs/Point center`: Especifica o centro da esfera, permitindo a definição de sua posição tridimensional no espaço.
 - `float32 radius`: Indica o raio da esfera, descrevendo seu tamanho.
- **Formato de Mensagem - SphereCloud:** A estrutura mais ampla que engloba a nuvem de esferas detectadas no ambiente contém os seguintes elementos:
 - `std_msgs/Header header`: Fornecendo informações de cabeçalho, como carimbo de data/hora para a nuvem de esferas.
 - `fs_path_planning/Sphere[] spheres`: Representa uma lista de esferas, onde cada esfera é definida por uma mensagem do tipo `"Sphere.msg"`.

4.2 ABORDAGEM CLÁSSICA - A*

Na primeira etapa de implementação do projeto, o objetivo foi desenvolver um modelo clássico de planejamento de caminhos, utilizando o algoritmo A* para encontrar o caminho ótimo entre os pontos de partida S e destino T .

O algoritmo de busca A* foi escolhido para integrar o módulo de planejamento de caminhos clássico neste trabalho devido a várias vantagens que apresenta em contextos similares a este trabalho, especialmente em ambientes tridimensionais com obstáculos complexos. O A* utiliza informações heurísticas para guiar a busca, favorecendo a exploração de regiões mais promissoras do espaço de estados, além de ser conceitualmente simples de implementar. O algoritmo também pode ser implementado no cálculo de trajetórias para operar em tempo real, em que permite a geração de caminhos seguros para a navegação autônoma a medida que o agente se desloca pelo ambiente.

A estratégia do desenvolvimento do modelo clássico de planejamento de caminhos foi dividida em etapas, abrangendo a implementação de uma estrutura que envolve as fases de percepção e controle, essenciais para a navegação autônoma.

A primeira dessas etapas incluiu a aquisição e processamento de dados sensoriais dos sensores embarcados no VANT. Os dados do RPLidar, Garmin Range e GPS, foram coletadas e preparadas para fornecer ao VANT a percepção sobre o ambiente em que ele está inserido. A fase seguinte abordou a detecção e desvio de obstáculos.

A segunda fase abordou a detecção e desvio de obstáculos. Neste ponto, algoritmos de segmentação foram aplicados para identificar os obstáculos no campo de visão do VANT com base nos dados do RPLidar. Em seguida, estratégias de desvio foram implementadas para permitir que o VANT evitasse colisões, ajustando sua trajetória de acordo com as informações percebidas.

Na terceira fase, o cálculo do caminho foi realizado com o algoritmo A*. O VANT utiliza as percepções 3D dos obstáculos e as informações de localização do GPS para calcular o caminho ótimo até o destino, levando em consideração os obstáculos detectados. A combinação das percepções tridimensionais e com o algoritmo A* permite ao VANT planejar trajetórias que não apenas evitavam obstáculos, mas que também otimizem a navegação em relação aos objetivos da missão.

Por fim, o VANT utiliza uma máquina de estados para avaliar o plano de navegação, transitando entre os estados para recalcular a rota em casos de obstáculos próximos. A máquina de estados fornece ao VANT uma avaliação constante da sua situação, fornecendo a tomada de decisões de forma dinâmicas conforme o estado naquele instante.

Ademais, a seguir, serão apresentados em detalhes os aspectos de implementação da abordagem clássica de planejamento de caminhos. Este processo de implementação considerou diversos componentes essenciais, incluindo a discretização do ambiente, a heurística utilizada pelo algoritmo A*, a máquina de estados que gerencia o fluxo de execução, a comunicação e integração entre os C++ e Python por meio do ROS e os formatos de mensagem empregados para troca de informações.

4.2.1 Grade Discreta

Grade Discreta é uma representação utilizada para modelar o ambiente dividindo em células discretas, formando uma grade ou malha. Cada célula da grade possui uma resolução (tamanho) para realizar a representação de uma posição no espaço 3D. Na utilização no planejamento de caminhos, a grade discreta permite que o ambiente seja tratado como um problema de busca em um espaço discreto, onde o objetivo é encontrar um caminho válido entre um ponto inicial e um ponto final. A grade discreta facilita a aplicação de algoritmos de busca clássicos, como o A*, que exploram o espaço de forma eficiente em busca da solução ótima, já que o algoritmo tratará apenas de nós expandidos em formato discretizado, e não contínuo. No apêndice A temos a implementação da Grade Discreta utilizada neste trabalho.

A abordagem da Grade Discreta atua como um conversor de pontos contínuos para discretos, viabilizando a implementação do algoritmo A* de forma eficiente. Ela permite que o espaço de navegação seja representado em uma estrutura discreta composta por células, permitindo o planejamento de caminhos em um ambiente simplificado.

A tabela 4 fornece exemplos de conversão de pontos contínuos para discretos com as resoluções de grade 0.5m e 1m. Cada linha da tabela representa um ponto contínuo no espaço, especificado em metros, e mostra como esse ponto é mapeado para uma célula discreta em uma grade de navegação.

Tabela 4 – Exemplos de conversão de pontos contínuos para discretos com diferentes resoluções de grade

Ponto Contínuo	Resolução de 0.5 metros	Resolução de 1 metro
(2.7, 1.3)	Célula (5, 2)	Célula (2, 1)
(4.8, 2.2)	Célula (9, 4)	Célula (4, 2)
(3.2, 0.7)	Célula (6, 1)	Célula (3, 0)

Fonte: O próprio autor.

No entanto, é importante destacar que, neste contexto, o agente, no caso o VANT, não realiza mapeamento contínuo do ambiente nem mantém memória do mapa em suas operações. Em vez disso, o ambiente é discretizado em uma grade que define estados possíveis e acessíveis para o agente, e o planejamento de caminhos é executado com base nas informações sensoriais instantâneas e nas percepções 3D dos obstáculos obtidas naquele instante de processamento do caminho a ser seguido. Sendo assim, o planejamento de caminhos é realizado sem o mapeamento constante do ambiente, o que contribui para evitar o custo computacional associado à construção e manutenção de um mapa completo.

4.2.2 A*, C++ e Python

Neste trabalho, foi adotada a modelagem que utiliza um serviço do ROS para o cálculo do caminho com o algoritmo A*, fornecendo assim, a capacidade de combinar efetivamente a eficiência do C++ com a flexibilidade do Python. Nesse cenário, o Python chama o serviço implementado em C++, fornecendo os pontos iniciais e de destino. O serviço em C++, por sua vez, executa o algoritmo A* para calcular o caminho e retorna a lista de pontos que compõem a rota a ser seguida pelo VANT.

Essa abordagem híbrida é eficiente, pois otimiza o cálculo do caminho. O C++ é uma linguagem de programação eficiente em termos de desempenho computacional, tornando o A* otimizado para o cálculo do caminho. Ao mesmo tempo, o Python oferece facilidade de uso e foi preferido para tarefas de alto nível, como a implementação da máquina de estados que será descrita posteriormente.

A comunicação entre os componentes do sistema é realizada por meio de mensagens do ROS que foram implementadas e personalizadas para integrar o serviço de cálculo de caminho. As mensagens implementadas foram:

- **PointList.msg:**

- geometry_msgs/Point[] *points*: É uma lista de pontos, representando o caminho calculado.
- **Astar.srv:**
 - geometry_msgs/Point *start*: Especifica o ponto de início do caminho.
 - geometry_msgs/Point *goal*: Indica o ponto de destino para o qual o caminho deve ser calculado.
 - fs_path_planning/PointList *path*: É a resposta do serviço, onde o serviço fornece a lista de pontos que compõem o caminho calculado.

Por meio dessas mensagens personalizadas, os nós do sistema podem se comunicar, enviando e recebendo informações sobre os pontos de início, destino e o caminho resultante, compondo assim, a integração do Python com C++ através do ROS. Além disso, o algoritmo A* implementado em C++ faz uso das percepções do sensor RPLidar, que é modelado de forma tridimensional. Essa modelagem tridimensional é obtida por meio de um tópico desenvolvido especificamente para esse fim no sistema ROS.

4.2.3 Heurística

A heurística utilizada na abordagem clássica foi a distância euclidiana em três dimensões. O cálculo da distância é dada pela fórmula:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (4.13)$$

onde $\mathbf{p} = (p_1, p_2, \dots, p_n)$ e $\mathbf{q} = (q_1, q_2, \dots, q_n)$ são vetores representando os pontos em n dimensões.

A escolha da distância euclidiana como heurística se deu pela sua simplicidade e eficácia para estimar a distância entre dois pontos em um espaço contínuo. A distância euclidiana é uma medida direta que leva em consideração as coordenadas espaciais dos nós no plano cartesiano, proporcionando uma estimativa precisa do custo restante para alcançar o objetivo. Além disso, a distância euclidiana é admissível, ou seja, nunca superestima o custo real necessário para chegar ao objetivo. Isso garante a otimalidade do algoritmo A* quando usado em conjunto com a distância euclidiana como heurística. Outro ponto é a ampla utilização em

muitos problemas de planejamento de caminhos, oferecendo um bom equilíbrio entre a eficiência em termos computacionais e qualidade do caminho encontrado.

4.2.4 Custo de Movimento

O custo de movimento é utilizado para avaliar a desejabilidade de um determinado caminho, levando em consideração a distância e a geometria tridimensional do espaço. No algoritmo A*, o custo de movimento é utilizado para atribuir uma pontuação a cada nó candidato, permitindo a seleção do melhor caminho.

Neste trabalho foi realizada uma modificação no custo de movimento no algoritmo A*, para priorizar nós com o eixo z menores. A alteração é expressa matematicamente da seguinte forma:

$$custo(n) = |x_{atual} - x_{vizinho}| + |y_{atual} - y_{vizinho}| + 3 \cdot |z_{atual} - z_{vizinho}| \quad (4.14)$$

onde x_{atual} , y_{atual} , z_{atual} representam as coordenadas do nó atual, e $x_{vizinho}$, $y_{vizinho}$, $z_{vizinho}$ representam as coordenadas do nó vizinho. Nesse cálculo, o peso no eixo z é ampliado em 3 vezes em relação aos eixos x e y , o que influencia a decisão do algoritmo ao escolher entre os nós vizinhos. Sendo assim, a abordagem de aplicar um peso maior em nós com variação no eixo z indica que movimentos verticais têm um custo maior em relação aos movimentos horizontais, tratando assim a necessidade de evitar os obstáculos no eixo z , priorizando a navegação lateral do VANT.

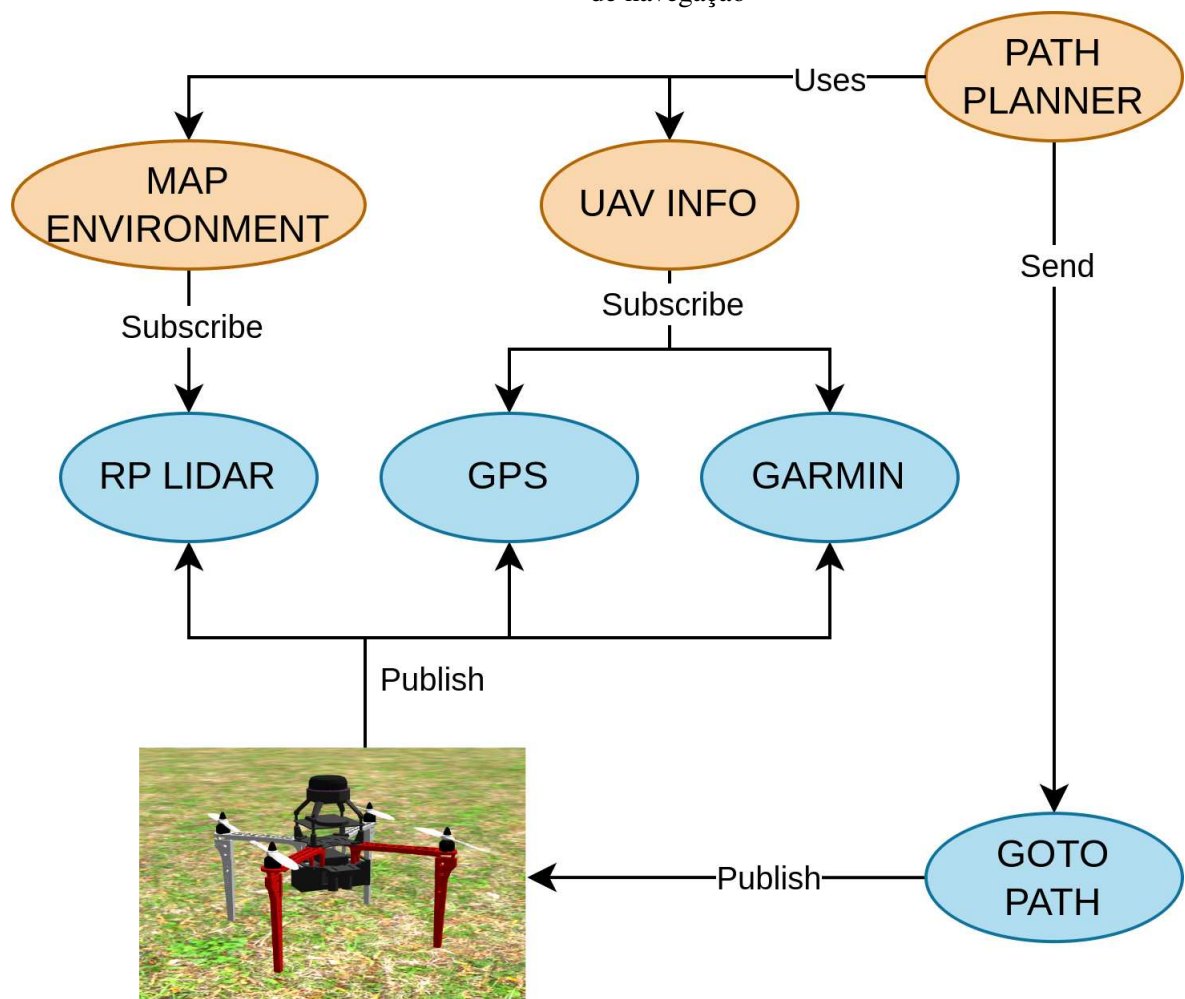
Além disso, essa modificação no cálculo do custo de movimento visa evitar que o algoritmo A* priorize caminhos que passem por cima das árvores nos ambientes. Em vez disso, o ajuste introduzido impõe um custo mais alto para movimentos verticais, forçando o sistema de navegação a tomar decisões laterais ao escolher trajetórias. Essa abordagem incentiva a contornar as árvores, ao invés de seguir diretamente por cima delas.

A expansão dos nós é realizada em todas as direções, incluindo as diagonais no eixo z . Isso significa que, durante a busca do caminho, o algoritmo considera movimentos não apenas ao longo dos eixos z e y , mas também ao longo do eixo x , permitindo a exploração de trajetórias tridimensionais.

4.2.5 Planejador de Caminhos

A arquitetura ilustrada na Figura 22 representa o sistema em ROS do planejador de caminhos. O sistema é composto por vários componentes para a navegação do VANT. Os sensores, incluindo o RPLidar, Gamin e GPS, são responsáveis por fornecer informações essenciais sobre o ambiente e o estado do VANT. O RPLidar alimenta os dados para o módulo de percepção - "Map Environment", que realiza o processamento dos dados para obter uma representação precisa do ambiente. O GPS e Gamin transmite os dados para o módulo "UAV Info", que interfaceia informações sobre o status do VANT. Esses módulos são então utilizados pelo módulo "Path Planner", que planeja e envia o caminho gerado diretamente para o módulo "Goto Path", que interfaceia o hardware e direciona os comandos para o módulo de controle.

Figura 22 – Arquitetura do sistema ROS do com sensores, módulos de interação com o VANT e o módulo de navegação



Fonte: O próprio autor.

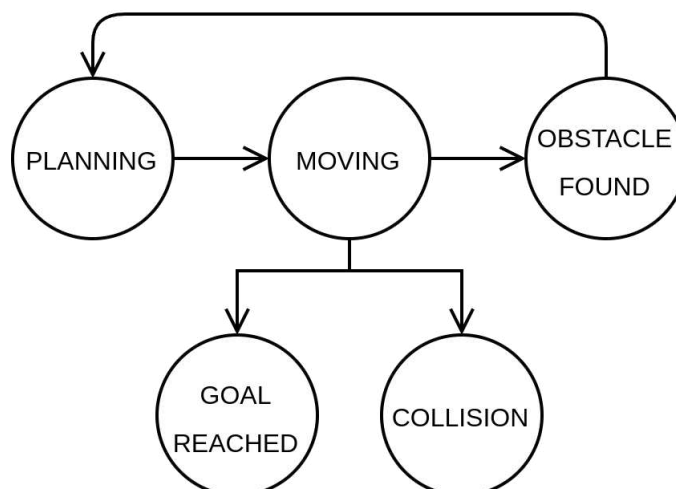
Na execução do plano de navegação, o VANT utiliza uma máquina de estados para

atualizar seus estados conforme as mudanças do ambiente e leitura dos sensores, tornando-se um modelo de navegação dinâmico em um ambiente simulado para concluir as missões atribuídas. A Figura 23 denota a máquina de estados implementada no módulo de planejamento de caminhos clássico.

A máquina de estados inicia no estado "planejando", onde o VANT analisa o ambiente a partir dos sensores e calcula a rota ótima a partir do A* para o destino. Em seguida, transita para o estado "movendo", onde o VANT segue o caminho planejado e se desloca em direção ao destino. Durante esse processo, caso um obstáculo seja encontrado, a máquina de estados passa para o estado "obstáculo encontrado". Nesse estado, o VANT recalcula uma nova rota para evitar a colisão com o obstáculo. Assim que o obstáculo é contornado, a máquina de estados retorna para o estado "movendo". Quando o VANT finalmente chega ao destino, a máquina de estados transita para o estado "chegou ao destino", indicando que a tarefa foi concluída com êxito.

Já o estado "colisão" representa o cenário em que o VANT colide com algum obstáculo no cenário. Essa colisão pode ocorrer em situações em que o planejamento de caminho inicial não conseguiu evitar adequadamente o obstáculo, o que pode ser influenciado pela percepção limitada do ambiente. Neste caso o VANT está equipado com apenas um sensor, o RPLidar, que mesmo com a modelagem das percepções 2D para 3D, pode não ser suficiente para detectar todos os obstáculos em seu caminho. Quando o VANT entra neste estado, a missão é abortada, resultando em falha da missão.

Figura 23 – Máquina de Estados para o Planejamento de Caminhos Clássico

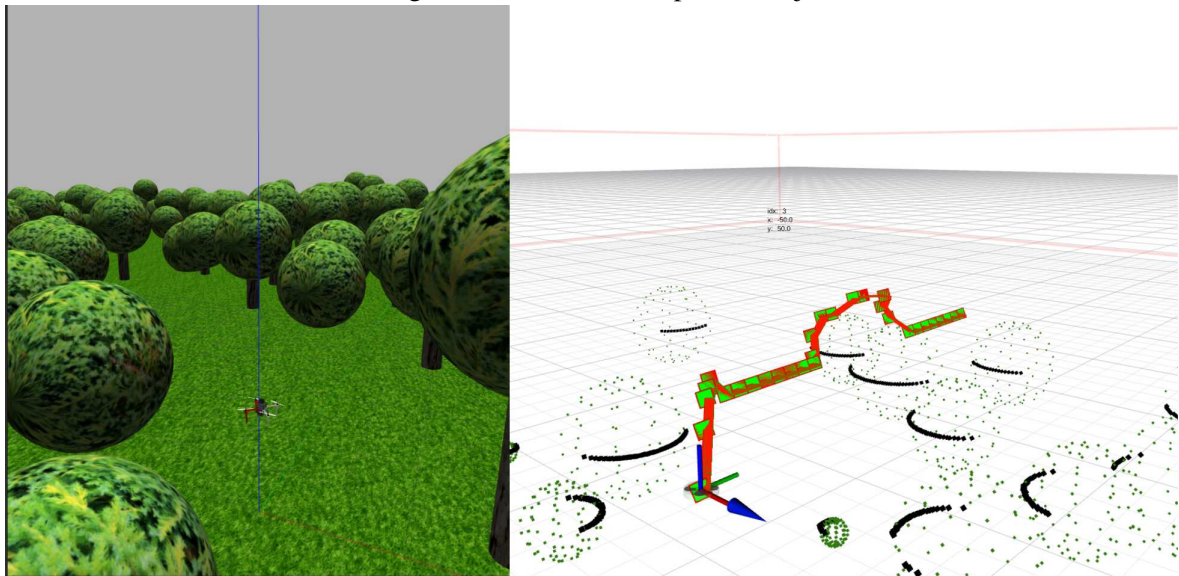


Fonte: O próprio autor.

A Figura 24 exhibe dois componentes-chave da navegação autônoma de um VANT em

um ambiente simulado. À esquerda, temos uma representação do cenário no simulador Gazebo, onde um VANT está inserido em um ambiente contendo árvores. Esse ambiente de exemplo requer que o VANT evite obstáculos (árvores) para alcançar seu objetivo com segurança. À direita, a imagem mostra a visão do agente capturada pelo RViz, uma ferramenta de visualização do ROS. Podemos observar a percepção do VANT, que inclui informações sensoriais, através do RPLidar modelado de forma tridimensional, permitindo ao VANT observar seu entorno e os obstáculos em tempo real. Além disso, podemos visualizar o caminho (em vermelho) gerado pelo módulo de planejamento de caminhos que utiliza o A* no RViz, que guiará o VANT através do ambiente, desviando dos obstáculos, otimizando sua trajetória para evitar colisões. Essa imagem ilustra a integração de percepção e planejamento de caminhos, componentes fundamentais da navegação autônoma, em um ambiente simulado.

Figura 24 – Cenário simulado no Gazebo Simulator à esquerda e percepção do agente, bem como o caminho gerado (em vermelho) pelo Planejador de Caminhos.



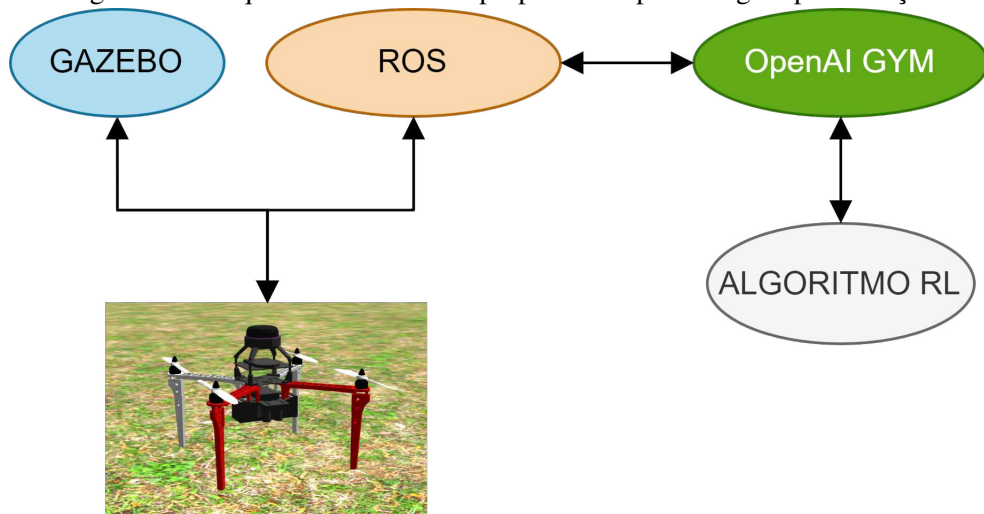
Fonte: O próprio autor.

4.3 ABORDAGEM MODERNA - APRENDIZAGEM POR REFORÇO

Na etapa do planejamento de caminhos utilizando a aprendizagem por reforço, foi necessário estender o projeto clássico descrito anteriormente, adicionando um agente de aprendizagem por reforço que possa interagir com o ambiente simulado. A Figura 25 ilustra a arquitetura da abordagem proposta, bem como a comunicação entre os módulos de simulação (Gazebo), controle e estado do VANT (ROS) e aprendizado (OpenAI Gym e Algoritmo de aprendizagem por reforço).

Durante o treinamento, o agente recebe recompensas positivas quando alcança com sucesso os pontos de chegada, e recompensas negativas quando colidir com obstáculos. Com base nessas recompensas, o agente ajusta suas ações e aprimora sua estratégia de navegação ao longo do tempo, buscando maximizar as recompensas acumuladas, selecionando ações que contribuem para o aprendizado da navegação. Dessa forma, a aprendizagem por reforço permite que o VANT desenvolva habilidades de navegação autônoma, adaptando-se às mudanças do ambiente e superando obstáculos de forma eficiente.

Figura 25 – Arquitetura do modelo proposto de aprendizagem por reforço.



Fonte: O próprio autor.

Para realizar a construção de um ambiente personalizado para o problema de planejamento de caminhos usando a biblioteca *Gym*, foi necessário a definição de um conjunto de estados, ações e recompensas que serão utilizados no processo de aprendizagem por reforço. Essas características serão exploradas com mais detalhes nas seções seguintes.

4.3.1 Espaço de Ação

O espaço de ação consiste em 6 ações discretas disponíveis. Cada uma dessas ações é mapeada para uma mudança específica na posição tridimensional do agente. Abaixo estão as ações e suas correspondentes alterações de posição:

- Ação 0 - Direita: A ação "direita" move o agente para a direita, adicionando uma velocidade positiva no eixo X (0.5,0.0,0.0).
- Ação 1 - Esquerda: A ação "esquerda" move o agente para a esquerda, adicionando uma velocidade negativa no eixo X (−0.5,0.0,0.0).

- Ação 2 - Frente: A ação "frente" move o agente para a frente, adicionando uma velocidade positiva no eixo Y (0.0, 0.5, 0.0).
- Ação 3 - Trás: A ação "trás" move o agente para trás, adicionando uma velocidade negativa no eixo Y (0.0, -0.5, 0.0).
- Ação 4 - Subir: A ação "subir" move o agente para cima, adicionando uma velocidade positiva no eixo Z (0.0, 0.0, 0.5).
- Ação 5 - Descer: A ação "descer" move o agente para baixo, adicionando uma velocidade negativa no eixo Z (0.0, 0.0, -0.5).

Essas ações permitem que o agente realize movimentos tridimensionais no ambiente. Através de um processo de aprendizado, o agente deve determinar quais ações tomar em diferentes estados do ambiente para maximizar sua recompensa acumulada ao longo do tempo.

4.3.2 Espaço de Observação

O espaço de observação é representado por um *Box* do *Gym*, que contém quatro elementos de observação distintos: distância para o alvo, posição atual do VANT, um *array* de obstáculos e o vetor para o alvo. Cada elemento possui parâmetros específicos que descrevem suas propriedades e limites, conforme detalhado na Tabela 5.

Esse espaço de observação permite ao agente receber informações relevantes sobre o ambiente, fornecendo dados sobre a distância entre o VANT e o objetivo, a localização atual do VANT no ambiente e a presença de obstáculos.

Tabela 5 – Descrição do espaço de observação do ambiente

Num	Observação	Tipo de Dado	Min.	Max.
0	Distância para o Alvo	float	0.0	Inf
1	Posição atual do VANT	Ponto 3D	-100.0	100.0
2	Obstáculos	Array 1D	0.0	14.0
3	Vetor para o Alvo	Ponto 3D	-100.0	100.0

Fonte: O próprio autor.

4.3.3 Estado Inicial

No contexto deste trabalho de planejamento de caminho, o estado inicial do agente é determinado pelo posicionamento inicial do VANT nas coordenadas [0.0, 0.0, 2.0] do ambiente em que ele está inserido. Essas coordenadas representam a localização tridimensional do

VANT, sendo os dois primeiros valores correspondentes às coordenadas x e y , e o terceiro valor representando a altitude do VANT em relação ao solo. O estado inicial fornece a base a partir da qual o agente começa sua jornada de navegação e tomada de decisões. A partir desse estado inicial, o agente deve explorar o ambiente, analisar informações sensoriais e tomar ações para alcançar com sucesso o objetivo desejado, como evitar obstáculos e chegar a um destino específico.

4.3.4 Fim do Episódio

O episódio termina se qualquer uma das seguintes situações ocorrer:

- Colidir com os obstáculos;
- Alcançar o destino;
- Se afastar mais de 5 metros do destino em relação à distância inicial.

Ao detectar uma das condições acima, o episódio é concluído e é iniciado um novo episódio, permitindo que o agente se adapte, aprenda com a experiência passada e melhore seu desempenho em futuros episódios.

4.3.5 Função de Recompensa

A função de recompensa é dada da seguinte forma:

$$\text{reward} = -0.1 * D + O + C \quad (4.15)$$

Onde:

D é a distância para o objetivo,

$O = -0.1$ se a distância para obstáculo mais próximo for inferior a 0.50 m; 0 caso contrário,

$$C : \begin{cases} 100 & \text{se o VANT chegou no alvo; 0 caso contrário} \\ -100 & \text{se o VANT colidiu com algum obstáculo; 0 caso contrário} \end{cases}$$

Essa função de recompensa fornece um valor numérico que reflete o desempenho do VANT em relação ao objetivo, incentivando-o a evitar colisões, alcançar o objetivo e se aproximar dele de forma eficiente.

4.3.6 Treinamento

O DQN (48) foi escolhido como a abordagem de aprendizagem por reforço para o problema de planejamento de caminhos desta pesquisa, devido a várias características que o tornam adequado para lidar com desafios específicos do domínio de VANTs. Uma das vantagens do DQN é sua capacidade de lidar com espaços de observação em forma de *Box*, o que é comum em ambientes complexos de robótica e simulação, incluindo o ambiente personalizado utilizado nesta pesquisa. Isso permite que o modelo proposto lide com informações heterogêneas, como a posição do VANT, distância até o alvo e os obstáculos naquele estado. As ações disponíveis para o VANT são discretas, o que torna o DQN eficaz em lidar com espaços de ação discretos. Além disso, o pacote Stable Baselines3 oferece uma implementação eficiente do DQN, disponibilizando hiperparâmetros para treinar e ajustar o modelo de aprendizagem por reforço. O agente foi treinado com os hiperparâmetros padrão do DQN.

Durante o treinamento, o agente é colocado para executar suas ações nos três cenários propostos, variando aleatoriamente entre eles a cada novo episódio. Essa estratégia teve como objetivo oferecer ao agente as diferentes configurações de altura dos obstáculos. A alternância entre os cenários promove uma aprendizagem mais abrangente, permitindo que o agente generalize suas habilidades de navegação para diferentes situações e condições de treinamento.

Além disso, foi abordado a estratégia de aprendizado do agente *Curriculum Learning* (60), visando otimizar o processo de treinamento. O *Curriculum Learning* busca a otimização o processo de treinamento de agentes de inteligência artificial, que ao contrário de métodos tradicionais que ensinam um agente diretamente em ambientes complexos desde o início, o *Curriculum Learning* adota uma abordagem mais gradual. Inicialmente, o agente é treinado em tarefas mais simples e, à medida que melhora seu desempenho, é gradualmente desafiado com tarefas mais complexas. Essa abordagem imita a forma como os seres humanos aprendem, começando com conceitos básicos antes de enfrentar desafios mais avançados.

No contexto desse trabalho, o *Curriculum Learning* foi aplicado para aprimorar o processo de aprendizagem do agente envolvido na navegação de destinos em níveis de dificuldade progressiva. Foram utilizados 19 destinos diferentes como alvos para o agente, iniciando com alvos sem obstáculos próximos. Os pontos selecionados para a progressão da missão pode ser observado na tabela 6. A progressão da dificuldade é realizada a cada 200 vezes que o agente

alcança um destino.

Tabela 6 – Descrição do espaço de observação do ambiente

Nível de Dificuldade	Destino
1	(0.0, -2.0, 1.2)
2	(2.0, 0.0, 1.2]
3	(-1.5, 1.4, 2.0)
4	(0.9, 2.9, 2.0)
5	(1.8, -2.6, 3.0)
6	(0.8, 3.5, 2.5)
7	(-4.0, 4.5, 2.5)
8	(4.4, -4.3, 3.0)
9	(1.2, -6.8, 2.0)
10	(10.1, -1.1, 2.5)
11	(-8.0, 8.0, 2.5)
12	(2.7, -13.2, 3.0)
13	(3.0, -13.3, 3.0)
14	(15.5, 8.8, 2.0)
15	(18.5, 13.1, 1.5)
16	(-24.2, 5.7, 2.5)
17	(-27.5, 8.7, 1.5)
18	(20.5, 22.2, 1.2)
19	(-45.1, 30.0, 2.0)

Fonte: O próprio autor.

Inicialmente, os pontos de treinamento foram projetados para oferecer ao VANT um ambiente de aprendizado progressivo. Os primeiros pontos são configurados como destinos próximos, sem a presença de obstáculos, em que o VANT aprende a navegar de maneira básica e segura. Essa fase inicial visa estabelecer as bases para o entendimento do VANT sobre os princípios fundamentais de navegação.

À medida que o VANT progride no treinamento, a estratégia de *Curriculum Learning* introduz gradualmente pontos mais distantes e desafia o VANT com destinos que possuem obstáculos. Ao expor o VANT a uma variedade de situações durante o treinamento, desde condições simples até cenários mais complexos, o *Curriculum Learning* busca a aprendizagem e adaptabilidade do VANT em melhorar sua capacidade de navegação autônoma nos cenários propostos, visando otimizar o aprendizado do VANT, proporcionando uma transição suave de tarefas mais simples para desafios mais complexos ao longo do treinamento.

5 EXPERIMENTOS E AVALIAÇÕES

Neste capítulo, serão descritos e analisados os experimentos realizados para avaliar a eficácia e o desempenho dos sistemas de navegação autônoma desenvolvidos. Os experimentos têm como objetivo principal verificar a capacidade dos sistemas em navegar de forma segura e eficiente nos ambientes tridimensionais propostos para este trabalho, contornando obstáculos de maneira adequada e atingindo seus objetivos de forma precisa.

Nas tabelas 7 e 8, que serão apresentadas com os resultados dos experimentos, a coluna "Teste" refere-se ao teste de navegação do VANT com os pontos de referência previamente estabelecidos na metodologia. Além disso, a coluna "Cenário" indica o ambiente em que os testes de navegação são realizados. Esse cenário descreve a distribuição da altura das árvores, conforme definido na metodologia do trabalho. As siglas utilizadas nos dados são as seguintes:

- **DP** (Distância Total Percorrida): Refere-se à quantidade total de distância percorrida pelo VANT durante a missão em metros.
- **VM** (Velocidade Média): Representa a velocidade média em metros por segundo (m/s) realizada pelo VANT ao longo da missão.
- **TV** (Tempo de Vôo): Corresponde ao tempo total que o VANT permaneceu no ar durante a missão, medido em segundos.
- **AM** (Altitude Média): Indica a altitude média em metros em que o VANT operou durante a missão.
- **VA** (Variação de Altitude): Representa a variação total na altitude, medida em metros, que o VANT experimentou ao longo da missão.

5.1 ABORDAGEM CLÁSSICA

A primeira etapa dos experimentos foi conduzida com a utilização da abordagem clássica do algoritmo A*. A tabela 7 apresenta os resultados da navegação clássica, com métricas relacionadas ao desempenho do sistema de navegação do VANT.

Os gráficos com os caminhos percorridos pelo VANT ao longo dos testes da abordagem clássica podem ser visualizados no Anexo D.

Tabela 7 – Resultados Obtidos dos Experimentos Clássicos

Teste	Cenário	DP (m)	VM (m/s)	TV (s)	AM (m)	VA (m)
1	1	11.91	0.76	15.49	2.31	2.03
	2	11.48	0.65	17.66	2.41	1.99
	3	11.53	0.73	15.74	2.59	2.03
2	1	8.86	0.74	11.92	1.74	1.04
	2	8.92	0.74	12.01	1.73	1.03
	3	9.00	0.74	12.04	1.75	1.04
3	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
4	1	31.67	0.53	59.10	1.41	0.96
	2	Colisão no Teste				
	3	29.82	0.57	58.26	1.63	1.02
5	1	27.08	0.49	55.14	1.42	3.01
	2	24.22	0.61	39.68	1.00	0.20
	3	23.70	0.54	43.64	0.95	0.68
6	1	21.04	0.54	38.93	1.01	1.93
	2	22.42	0.56	39.59	0.94	1.92
	3	22.36	0.53	41.73	0.97	1.92
7	1	23.10	0.54	42.74	1.24	1.42
	2	22.78	0.54	42.04	1.38	1.42
	3	Colisão no Teste				
8	1	21.37	0.53	39.88	1.46	0.48
	2	22.60	0.44	50.64	1.43	0.48
	3	21.58	0.52	41.36	1.43	0.48
9	1	25.90	0.65	39.40	3.51	3.48
	2	25.33	0.51	49.09	2.60	1.51
	3	25.97	0.51	50.67	2.99	2.68
10	1	31.80	0.51	61.19	1.91	0.98
	2	33.48	0.55	60.30	1.86	1.01
	3	33.33	0.48	68.26	1.83	1.00
11	1	23.00	0.37	60.82	3.14	4.48
	2	Colisão no Teste				
	3	18.93	0.48	39.08	1.45	0.49
12	1	Colisão no Teste				
	2	31.10	0.59	52.18	1.58	1.92
	3	30.54	0.50	60.48	1.49	1.93
13	1	25.67	0.51	49.40	2.59	2.98
	2	Colisão no Teste				
	3	Colisão no Teste				
14	1	24.27	0.56	42.93	3.18	3.48
	2	Colisão no Teste				
	3	Colisão no Teste				
15	2	45.98	0.52	100.72	1.09	1.81
	2	41.39	0.56	91.32	0.95	1.62
	3	42.24	0.50	84.04	1.00	1.55

Fonte: O próprio autor.

Nos 45 testes realizados, é observado que o VANT operado pela abordagem clássica colidiu em 11 deles. Esses resultados indicam que a taxa de sucesso da missão é de aproximadamente 76%, o que significa que o sistema foi capaz de completar a missão com sucesso em 34 dos 45 testes realizados. Embora seja um indicativo positivo da capacidade do sistema de navegação em contornar obstáculos e alcançar os destinos desejados na maioria dos casos, as colisões em 11 testes destacam a necessidade de aprimoramentos, em especial à percepção limitada do ambiente, especialmente em locais com muitas árvores próximas, onde o sistema teve dificuldade em identificar completamente todos os obstáculos.

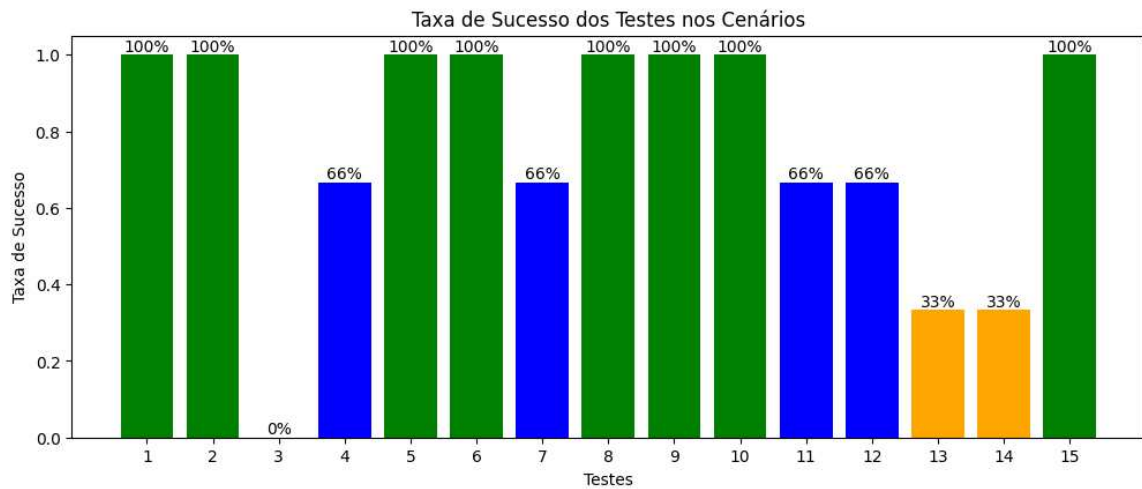
A complexidade do cenário 1 se deve principalmente à sua distribuição de altura das árvores, que é a mais variável entre os cenários de teste. Essa variabilidade na altura das árvores cria um ambiente altamente desafiador para a navegação do VANT. Como era previsto, as características desse cenário resultaram em missões que demandaram mais tempo, levando a uma maior distância percorrida e uma velocidade média mais baixa em comparação com outros cenários.

A distribuição variável da altura das árvores requer que o VANT execute desvios significativos ao planejar sua trajetória. A necessidade de navegar lateralmente ou na vertical para evitar árvores de diferentes alturas aumenta a distância total percorrida e, naturalmente, diminui a velocidade média do VANT, já que ele precisa manter um comportamento mais cauteloso e controlado. Também foi o cenário com maior variação na altura da missão, já que o VANT preferiu navegar por cima das árvores.

O gráfico 26 apresenta a taxa de sucesso nos 15 testes realizados na abordagem clássica nos diferentes cenários de navegação. Cada barra representa um teste individual, e as alturas das barras variam de acordo com a taxa de sucesso obtida. Alguns testes alcançaram uma taxa de sucesso de 100%, indicando o melhor desempenho nos três cenários, enquanto outros obtiveram taxas de sucesso variadas, como 66% (dois de três testes finalizados com sucesso), 33% (completou um de três testes), e um teste que resultou em uma taxa de sucesso de 0% - colisão em todos os cenários.

A taxa de sucesso de 0% no teste 3 representa o cenário em que a percepção limitada do agente se tornou um obstáculo para alcançar o objetivo. Nesse caso específico, a região próxima ao destino estava densamente composta por árvores agrupadas, criando o trajeto mais complexo dentre todos os testes. A percepção limitada do VANT não era capaz de identificar

Figura 26 – Taxa de Sucesso dos Experimentos da Abordagem Clássica



Fonte: O próprio autor.

adequadamente a presença de obstáculos nas áreas de agrupamento de árvores. Isso levou a tentativas de navegar por trajetórias que resultaram em colisões, resultado na falha da missão em todos os cenários.

Nos cenários 2 e 3, o VANT tende a escolher trajetórias que o levam a navegar entre os troncos das árvores, muitas vezes buscando contornar os obstáculos. No entanto, a presença de obstáculos como as próprias árvores e troncos cria um desafio adicional. A percepção limitada do ambiente, que se baseia no sensor RPLidar 2D modelado de forma 3D, permite ao VANT visualizar apenas uma porção do tronco das árvores, em vez de sua extensão completa.

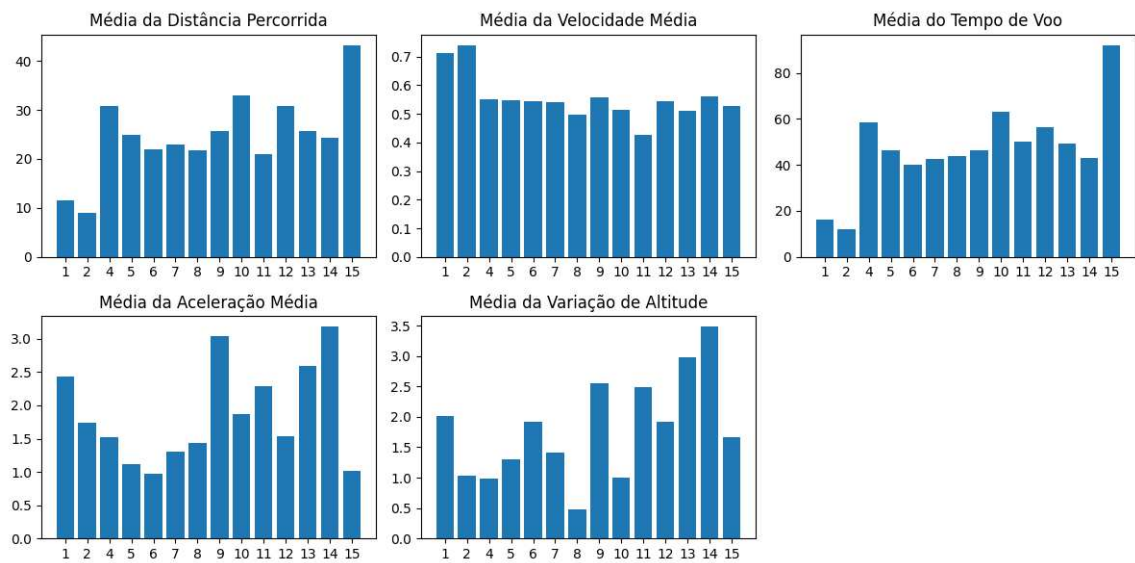
Isso significa que, ao tentar navegar por baixo ou por cima de uma esfera projetada no tronco, o VANT não é capaz de identificar completamente a presença de obstáculos ocultos, como as partes não visíveis do tronco ou a copa de outra árvore. Consequentemente, o VANT colide com essas áreas ocultas, representadas no modelo de forma tridimensional como esferas projetadas no tronco das árvores. Essa situação destaca o desafio associado à percepção limitada em ambientes tridimensionais. Embora os cenários 2 e 3 possam parecer menos complexos devido à uniformidade da altura das árvores, a navegação entre troncos requer uma percepção detalhada do ambiente, que não é totalmente possível com o sensor disponível.

As Figuras 27 e 28 apresentam as médias das métricas dos testes realizados, fornecendo uma visão geral do desempenho médio do VANT ao longo dos diferentes cenários de teste. Nesses gráficos, as métricas podem ser comparadas entre os cenários.

Um destaque notável nos gráficos é o teste 15, que se diferencia das demais médias.

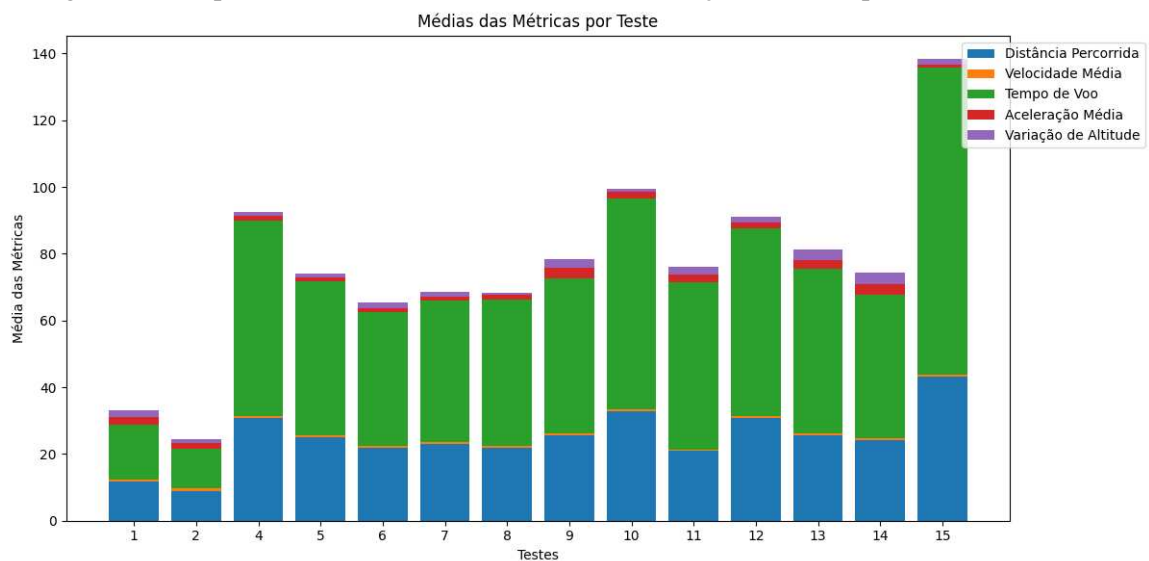
Especificamente, o teste 15 se destacou por apresentar a maior distância total percorrida em comparação com os outros cenários de teste. Isso indica que, ao navegar nesse teste, o VANT percorreu uma distância significativamente maior em comparação com os outros testes. Além disso, o teste 15 também registrou o maior tempo de voo entre os testes. Outro destaque do teste foi a observação de uma aceleração média mais baixa em comparação com os outros testes. Isso indica que o VANT manteve um comportamento de voo mais suave e controlado, devido a desvios e manobras para evitar colisões em uma trajetória maior.

Figura 27 – Média das métricas nos 15 experimentos realizados



Fonte: O próprio autor.

Figura 28 – Impacto das médias das métricas na distribuição dos 15 experimentos realizados



Fonte: O próprio autor.

5.2 ABORDAGEM CLÁSSICA SEM A MODELAGEM TRIDIMENSIONAL DOS OBSTÁCULOS

Os testes realizados sem a modelagem da percepção tridimensional dos obstáculos, obtidos por meio do sensor RPLidar, fornecem resultados dos desafios enfrentados pela navegação autônoma em ambientes tridimensionais complexos. A ausência da modelagem tridimensional implicou que o VANT depende principalmente de informações bidimensionais para perceber o ambiente ao seu redor. Nesse contexto, os resultados desses testes trazem as limitações decorrentes da percepção limitada do VANT. O RPLidar, sendo um sensor 2D, fornece dados sobre obstáculos apenas no plano horizontal, deixando uma lacuna significativa na capacidade do VANT de ter a percepção tridimensional do ambiente. Isso resultou em cenários nos quais o VANT não conseguiu identificar completamente a presença de obstáculos em diferentes alturas, levando a decisões de navegação inadequadas, e até mesmo a colisão na maioria dos testes.

A tabela 8 apresenta os resultados da navegação clássica sem a modelagem tridimensional dos obstáculos.

Tabela 8 – Resultados Obtidos dos Experimentos Clássicos sem a Modelagem das Percepções em Três Dimensões

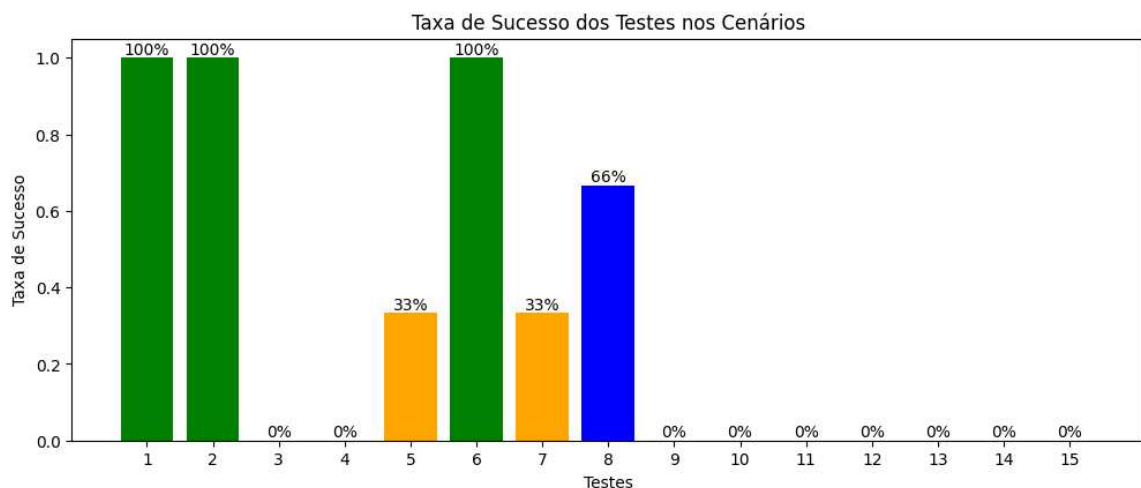
Teste	Cenário	DP (m)	VM (m/s)	TV (s)	AM (m)	VA (m)
1	1	11.96	0.80	15.49	2.57	2.18
	2	11.87	0.85	17.74	2.63	1.56
	3	11.96	0.89	14.21	2.64	1.85
2	1	8.00	0.79	10.06	1.97	0.47
	2	7.99	0.79	10.14	1.97	0.76
	3	7.98	0.79	10.04	1.98	0.45
3	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
4	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
5	1	Colisão no Teste				
	2	23.53	0.59	39.88	0.99	1.84
	3	Colisão no Teste				
6	1	20.36	0.59	34.36	0.98	1.81
	2	19.20	0.59	32.42	1.01	1.80
	3	18.43	0.73	25.14	1.09	1.51
7	1	Colisão no Teste				
	2	Colisão no Teste				
	3	20.08	0.75	27.08	1.51	0.43
8	1	20.28	0.75	26.88	1.43	0.54
	2	Colisão no Teste				
	3	20.11	0.76	20.32	1.40	0.52
9	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
10	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
11	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
12	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
13	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
14	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				
15	1	Colisão no Teste				
	2	Colisão no Teste				
	3	Colisão no Teste				

Fonte: O próprio autor.

Sem a modelagem tridimensional dos obstáculos que são obtidos através do RPLidar, nos 45 testes realizados, o VANT colidiu em 32 deles. Esses resultados indicam que a taxa de sucesso da missão é de aproximadamente 29%, o que significa que o sistema foi capaz de completar a missão com sucesso em 13 dos 45 testes realizados.

A comparação entre os testes com a modelagem tridimensional da percepção comparando sem ela destaca diferenças significativas no desempenho do VANT durante a navegação. Nos testes com a percepção tridimensional, a taxa de sucesso atingiu 76%, indicando maior capacidade do sistema de lidar com o ambiente. Em contrapartida, nos testes sem a modelagem tridimensional, a taxa de sucesso foi consideravelmente inferior, registrando 29%.

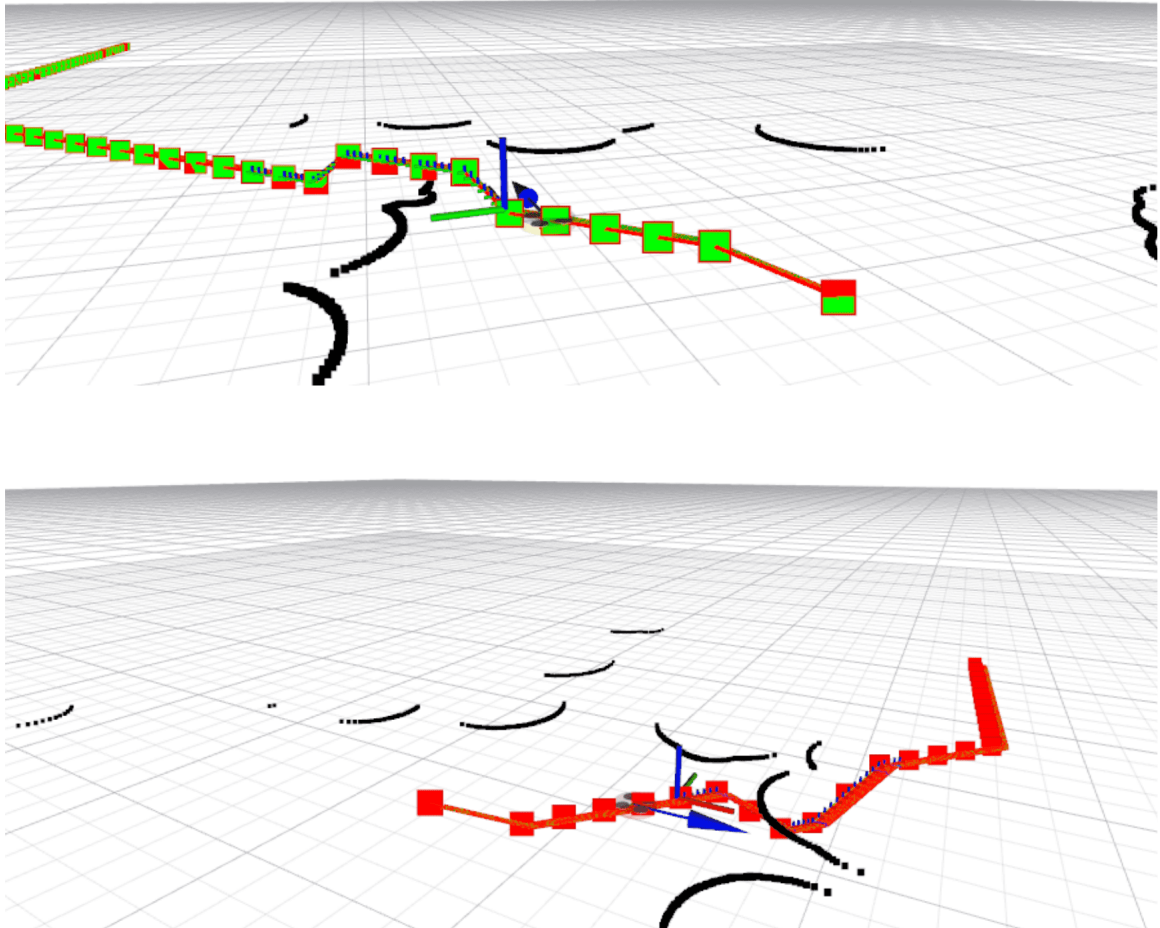
Figura 29 – Taxa de Sucesso nos Experimentos da Abordagem Clássica sem a Modelagem Tridimensional dos Obstáculos.



Fonte: O próprio autor.

A elevada taxa de colisões observada nos experimentos sem a modelagem tridimensional dos obstáculos foi resultante da representação 2D do ambiente pelo sensor RPLidar e a natureza tridimensional dos obstáculos reais. A escolha de caminhos baseada apenas na informação bidimensional do RPLidar levou o agente a desviar da linha de visão planar dos obstáculos, sem considerar adequadamente sua dimensão vertical. A Figura 30 ilustra o caminho gerado pelo agente, exibindo como o desvio apenas ao longo da projeção 2D do obstáculo não é suficiente para evitar colisões, uma vez que o obstáculo também possui uma dimensão vertical não percebida pelo sensor. As percepções tridimensionais aprimorou a capacidade do agente de tomar decisões mais seguras e adaptativas diante de obstáculos.

Figura 30 – Caminho encontrado colide com as árvores devido à percepção limitada que o agente tem sobre o ambiente, que neste caso, é composto apenas por uma visão planar e bidimensional dos obstáculos.



Fonte: O próprio autor.

A percepção tridimensional fornece ao VANT uma percepção mais completa do ambiente, identificando obstáculos em diversas altitudes e aprimorando suas decisões de navegação. Vale destacar que a solução implementada exclusivamente em software, sem custos financeiros adicionais ou de hardware mais avançado, contribuiu significativamente para o aumento da taxa de sucesso.

A Tabela 9 exhibe e compara a taxa de sucesso entre a abordagem clássica com e sem a modelagem das percepções tridimensionais dos obstáculos dos 15 experimentos nos três cenários de teste.

Tabela 9 – Comparativo da Taxa de Sucesso da Abordagem Clássica Com e Sem a Modelagem Tridimensional dos Obstáculos

Taxa de Sucesso (%)		
Experimento	Abordagem Clássica	Abordagem Clássica Sem Modelagem 3D
1	100	100
2	100	100
3	0	0
4	66	0
5	100	33
6	100	100
7	66	33
8	100	66
9	100	0
10	100	0
11	66	0
12	66	0
13	33	0
14	33	0
15	100	0

Fonte: O próprio autor.

5.3 ABORDAGEM MODERNA - APRENDIZAGEM POR REFORÇO

O agente de aprendizado por reforço foi treinado por um total de 171.184 segundos, equivalente a aproximadamente 47.5 horas. Durante esse período, foram realizados 2.824 episódios, totalizando 18.402.246 *steps* de treinamento. A recompensa média por episódio atingiu um máximo de -136.32 e um mínimo de -13.800, refletindo a variação na eficácia das políticas de decisão do agente ao longo do treinamento. O cumprimento médio do episódio variou de 56.091 *steps* no início do treinamento para 8.972 *steps* na última iteração do DQN, indicando que a quantidade de *steps* realizados pelo VANT durante o treinamento foi diminuindo ao longo do tempo.

Além disso, o número total de atualizações realizadas na rede neural durante o treinamento foi de 6.950.560, evidenciando a quantidade significativa de ajustes que o modelo passou para aprimorar suas estratégias ao longo do tempo.

A Figura 31 ilustra os dados obtidos a partir do treinamento do agente. Temos a Recompensa Média, Tamanho do Episódio e o *Loss* ao longo da fase de treinamento. A primeira sub-figura representa a Recompensa Média por episódio, indicando como o agente foi capaz de maximizar suas recompensas ao longo do treinamento. A segunda sub-figura refere-se ao

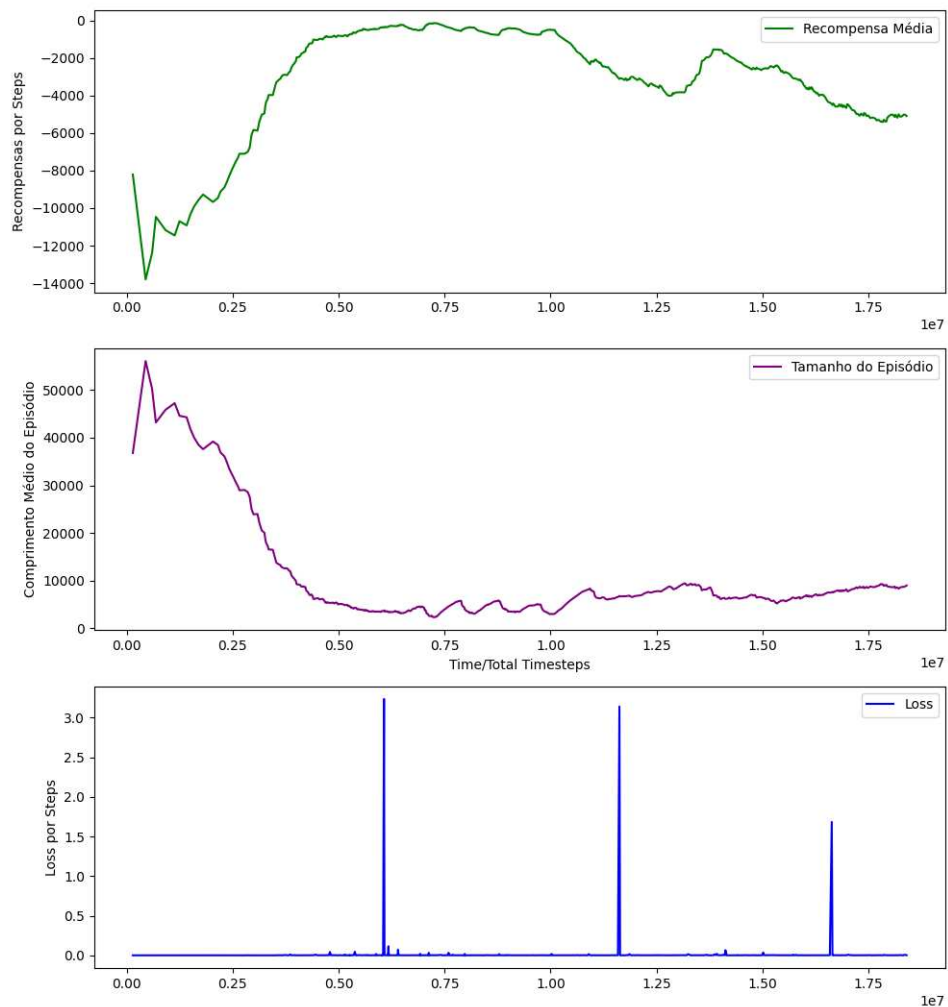
Tamanho do Episódio, revelando variações no número de passos que o agente executou em cada episódio durante o treinamento. Por fim, a terceira sub-figura exibe o *Loss*, que é uma métrica que reflete a discrepância entre as predições do agente e os alvos desejados, indicando o progresso do aprendizado.

Os picos na recompensa média refletem o progresso do agente em dominar uma determinada dificuldade do *Curriculum Learning*. Durante esses períodos, o agente foi capaz de acertar o objetivo várias vezes consecutivas, maximizando assim sua recompensa média. Por outro lado, observa-se uma leve queda na recompensa média quando o agente transita para uma nova dificuldade. Esse declínio é atribuível ao período de adaptação do agente às novas condições e desafios apresentados pela mudança de objetivo.

Próximo ao final do treinamento, há uma queda mais significativa na recompensa média, acompanhada por um aumento na quantidade de passos por episódio. Esse fenômeno acontece pois o agente enfrentou uma dificuldade considerável ao lidar com os desafios impostos por essa última etapa do *Curriculum Learning*. A queda na recompensa média indica que o agente teve dificuldades em atingir o objetivo desejado, enquanto o aumento no número de passos sugere que o agente estava explorando opções adicionais na tentativa de encontrar uma solução viável.

Figura 31 – Gráfico da evolução das métricas por steps durante o treinamento do agente de aprendizagem por reforço.

Dados Obtidos a Partir do Treinamento



Fonte: O próprio autor.

Ao longo do processo de treinamento, o agente foi capaz de superar apenas 9 das 19 dificuldades propostas no *Curriculum Learning*. A conclusão bem-sucedida das missões iniciais sinaliza a aquisição de habilidades fundamentais de navegação pelo agente. No entanto, surge claramente a limitação do desempenho do VANT na transição da dificuldade 9 para 10, caracterizado por uma densa presença de árvores ao longo do percurso. Essa dificuldade indica que, embora o agente tenha assimilado estratégias em destinos com menos obstáculos e pontos mais próximos, ele enfrenta desafios adicionais ao lidar com os objetivos mais complexos.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo o desenvolvimento de um módulo de planejamento de caminhos para VANTs, utilizando a técnica clássica A* e modernas com aprendizagem por reforço em ambientes 3D. Os resultados obtidos foram analisados em duas abordagens: a primeira, utilizando a abordagem clássica de planejamento de caminhos com a modelagem tridimensional para aprimorar a navegação do VANT e a segunda sem esse modelo.

Os experimentos revelaram que a abordagem clássica sem a modelagem tridimensional proporcionou um desempenho no sucesso das missões de 29%, mas a introdução da modelagem das percepções do agente para três dimensões proposta neste trabalho apresentou melhorias significativas na eficácia do planejamento de trajetória, subindo a taxa de sucesso para 76%. Essa modelagem tridimensional permitiu uma percepção mais abrangente do ambiente, resultando em decisões de navegação mais informadas e, conseqüentemente, em um desempenho global aprimorado.

Adicionalmente, foi explorado o treinamento do VANT utilizando técnicas de aprendizagem por reforço, visando o desenvolvimento de habilidades de navegação. Durante o treinamento, o agente do VANT superou 9 das 19 dificuldades propostas no *Curriculum Learning*, evidenciando sua capacidade de acertar consistentemente o objetivo em diferentes contextos. Essa conquista foi alcançada em pelo menos 1800 episódios, representando uma porção considerável dos 2.824 episódios completos no treinamento. Esses resultados destacam a habilidade do VANT treinado em aprender e generalizar suas capacidades de navegação para cenários variados, indicando que o agente foi capaz de aprender com sucesso essas habilidades fundamentais de navegação. No entanto, mesmo com esse sucesso, o agente ainda enfrentou desafios ao lidar com obstáculos, resultando em colisões durante os testes de navegação. Essa dinâmica sugere a necessidade contínua de aprimoramentos para lidar efetivamente com os desafios propostos no *Curriculum Learning*.

Considerando as limitações identificadas no desempenho do VANT com a abordagem moderna ao lidar com obstáculos, trabalhos futuros podem explorar o ajuste do ambiente personalizado criado neste trabalho. Ajustes na observação, na função de recompensa ou

mesmo no espaço de ação para lidar com esses cenários podem ser explorados para reforçar a adaptabilidade do VANT nas outras dificuldades do *Curriculum Learning*.

A escolha de um espaço de ação discreto limita a capacidade do agente de explorar comportamentos durante a navegação. A transição para um espaço de ação contínuo ofereceria ao agente uma flexibilidade maior para ajustar suas decisões de controle, permitindo respostas controladas e adaptativas a diferentes condições do ambiente.

A função de recompensa pode ser refinada para fornecer incentivos mais eficazes à aproximação do agente ao destino. Outro ajuste que pode ser feito na função de recompensa é implementar um mecanismo para ser sensível à distância entre o agente e os obstáculos, de modo que, à medida que a proximidade aumenta, a penalização se torna mais significativa.

Além disso, a pesquisa pode expandir para utilização de adaptação do modelo de planejamento de caminhos para ambientes dinâmicos. Isso envolveria o desenvolvimento de estratégias que permitam ao VANT tomar decisões em tempo real para evitar obstáculos móveis, como outros VANTs, ou objetos que se movem pelo cenário. Essa extensão poderá ser aplicada em ambientes reais, onde as limitações e complexidades do mundo real podem ser plenamente compreendidas e enfrentadas. A pesquisa também pode expandir no desenvolvimento de técnicas de sensoriamento mais avançadas, como o uso de sensores mais sofisticados, melhorando a capacidade do VANT de detectar e evitar obstáculos de maneira mais eficaz, aumentando a taxa de sucesso das missões.

REFERÊNCIAS

- 1 BOUHAMED, O.; GHAZZAI, H.; BESBES, H.; MASSOUD, Y. Autonomous UAV Navigation: A DDPG-Based Deep Reinforcement Learning Approach. In: **2020 IEEE International Symposium on Circuits and Systems (ISCAS)**. [S.l.: s.n.], 2020. p. 1–5. ISSN: 2158-1525.
- 2 GUERRA, A.; GUIDI, F.; DARDARI, D.; DJURIĆ, P. M. Reinforcement Learning for UAV Autonomous Navigation, Mapping and Target Detection. In: **2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)**. [S.l.: s.n.], 2020. p. 1004–1013. ISSN: 2153-3598.
- 3 GUERRA, A.; GUIDI, F.; DARDARI, D.; DJURIĆ, P. M. Real-Time Learning for THZ Radar Mapping and UAV Control. In: **2021 IEEE International Conference on Autonomous Systems (ICAS)**. [S.l.: s.n.], 2021. p. 1–5.
- 4 QU, C.; GAI, W.; ZHONG, M.; ZHANG, J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (uavs) path planning. **Applied soft computing**, Elsevier, v. 89, p. 106099, 2020.
- 5 DOUKHI, O.; LEE, D. J. Deep Reinforcement Learning for Autonomous Map-Less Navigation of a Flying Robot. **IEEE Access**, v. 10, p. 82964–82976, 2022. ISSN 2169-3536.
- 6 CHIKHAOUI, K.; GHAZZAI, H.; MASSOUD, Y. PPO-based Reinforcement Learning for UAV Navigation in Urban Environments. In: **2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)**. [S.l.: s.n.], 2022. p. 1–4. ISSN: 1558-3899.
- 7 HE, L.; AOUF, N.; SONG, B. Explainable Deep Reinforcement Learning for UAV autonomous path planning. **Aerospace Science and Technology**, v. 118, 2021. ISSN 12709638. Publisher: Elsevier Masson s.r.l. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85114628440&doi=10.1016/j.ast.2021.107052&partnerID=40&md5=51af21c3df05347233e33766a0f03331>>.
- 8 MA, S.; ZHANG, Y.; XIN, J.; YI, Y.; LIU, D.; LIU, H. An early forest fire detection method based on unmanned aerial vehicle vision. In: IEEE. **2018 Chinese Control And Decision Conference (CCDC)**. [S.l.], 2018. p. 6344–6349.
- 9 BIALAS, J.; DOLLER, M. Coverage Path Planning for Unmanned Aerial Vehicles in Complex 3D Environments with Deep Reinforcement Learning. In: **2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)**. [S.l.: s.n.], 2022. p. 1080–1085.
- 10 CARVALHO, K. B. de; OLIVEIRA, I. R. L. de; VILLA, D. K. D.; CALDEIRA, A. G.; SARCINELLI-FILHO, M.; BRANDÃO, A. S. Q-learning based Path Planning Method for UAVs using Priority Shifting. In: **2022 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.: s.n.], 2022. p. 421–426. ISSN: 2575-7296.
- 11 SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. [S.l.]: MIT press, 2018.

- 12 ABREU, M.; REIS, L. P.; LAU, N. Learning to run faster in a humanoid robot soccer environment through reinforcement learning. In: **Robot World Cup**. [S.l.]: Springer, 2019. p. 3–15.
- 13 DONG, Z.; LIU, C. Collaborative Coverage Path Planning of UAV Cluster based on Deep Reinforcement Learning. In: **2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)**. [S.l.: s.n.], 2021. p. 201–207.
- 14 THEILE, M.; BAYERLEIN, H.; NAI, R.; GESBERT, D.; CACCAMO, M. UAV Coverage Path Planning under Varying Power Constraints using Deep Reinforcement Learning. In: **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2020. p. 1444–1449. ISSN: 2153-0866.
- 15 THEILE, M.; BAYERLEIN, H.; NAI, R.; GESBERT, D.; CACCAMO, M. UAV Path Planning using Global and Local Map Information with Deep Reinforcement Learning. In: **2021 20th International Conference on Advanced Robotics (ICAR)**. [S.l.: s.n.], 2021. p. 539–546.
- 16 JAYAWEERA, H. M.; HANOUN, S. Path planning of unmanned aerial vehicles (uavs) in windy environments. **Drones**, MDPI, v. 6, n. 5, p. 101, 2022.
- 17 MANNAN, A.; OBAIDAT, M.; MAHMOOD, K.; AHMAD, A.; AHMAD, R. Classical versus reinforcement learning algorithms for unmanned aerial vehicle network communication and coverage path planning: A systematic literature review. **International Journal of Communication Systems**, v. 36, n. 5, 2023. ISSN 10745351. Publisher: John Wiley and Sons Ltd. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85145298912&doi=10.1002%2fdac.5423&partnerID=40&md5=550277cf227a3b19d6386349c5a764eb>>.
- 18 ZHANG, S.; LI, Y.; DONG, Q. Autonomous navigation of UAV in multi-obstacle environments based on a Deep Reinforcement Learning approach. **Applied Soft Computing**, v. 115, 2022. ISSN 15684946. Publisher: Elsevier Ltd. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85121133125&doi=10.1016%2fj.asoc.2021.108194&partnerID=40&md5=00def40fb36b5de3fb77df5b92ccf3dd>>.
- 19 TSOUROS, D. C.; BIBI, S.; SARIGIANNIDIS, P. G. A review on uav-based applications for precision agriculture. **Information**, MDPI, v. 10, n. 11, p. 349, 2019.
- 20 ZHANG, H.-y.; LIN, W.-m.; CHEN, A.-x. Path planning for the mobile robot: A review. **Symmetry**, MDPI, v. 10, n. 10, p. 450, 2018.
- 21 ALEXOPOULOS, C.; GRIFFIN, P. M. Path planning for a mobile robot. **IEEE Transactions on systems, man, and cybernetics**, IEEE, v. 22, n. 2, p. 318–322, 1992.
- 22 ISMAIL, A.; SHETA, A.; AL-WESHAH, M. A mobile robot path planning using genetic algorithm in static environment. **Journal of Computer Science**, v. 4, n. 4, p. 341–344, 2008.
- 23 BODHALE, D.; AFZULPURKAR, N.; THANH, N. T. Path planning for a mobile robot in a dynamic environment. In: IEEE. **2008 IEEE International Conference on Robotics and Biomimetics**. [S.l.], 2009. p. 2115–2120.
- 24 KAVRAKI, L. E.; SVESTKA, P.; LATOMBE, J.-C.; OVERMARS, M. H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. **IEEE transactions on Robotics and Automation**, IEEE, v. 12, n. 4, p. 566–580, 1996.

- 25 QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE, JAPAN. **ICRA Workshop on Open Source Software**. [S.l.], 2009. v. 3, n. 3.2, p. 5.
- 26 ARAÚJO, A.; PORTUGAL, D.; COUCEIRO, M. S.; ROCHA, R. P. Integrating arduino-based educational mobile robots in ros. **Journal of Intelligent & Robotic Systems**, Springer, v. 77, p. 281–298, 2015.
- 27 ST-ONGE, D.; HERATH, D. The robot operating system (ros1 &2): Programming paradigms and deployment. In: **Foundations of Robotics: A Multidisciplinary Approach with Python and ROS**. [S.l.]: Springer, 2022. p. 105–126.
- 28 HÖNIG, W.; AYANIAN, N. Flying multiple uavs using ros. **Robot Operating System (ROS) The Complete Reference (Volume 2)**, Springer, p. 83–118, 2017.
- 29 KOENIG, N.; HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE. **2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)**. [S.l.], 2004. v. 3, p. 2149–2154.
- 30 SCIORTINO, C.; FAGIOLINI, A. Ros/gazebo-based simulation of quadcopter aircrafts. In: IEEE. **2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)**. [S.l.], 2018. p. 1–6.
- 31 BACA, T.; PETRLIK, M.; VRBA, M.; SPURNY, V.; PENICKA, R.; HERT, D.; SASKA, M. The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. **Journal of Intelligent & Robotic Systems**, Springer, v. 102, n. 1, p. 26, 2021.
- 32 SANCHEZ-LOPEZ, J. L.; FERNÁNDEZ, R. A. S.; BAVLE, H.; SAMPEDRO, C.; MOLINA, M.; PESTANA, J.; CAMPOY, P. Aerostack: An architecture and open-source software framework for aerial robotics. In: IEEE. **2016 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.], 2016. p. 332–341.
- 33 HERT, D.; BACA, T.; PETRACEK, P.; KRATKY, V.; SPURNY, V.; PETRLIK, M.; VRBA, M.; ZAITLIK, D.; STOUDEK, P.; WALTER, V. et al. Mrs modular uav hardware platforms for supporting research in real-world outdoor and indoor environments. In: IEEE. **2022 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.], 2022. p. 1264–1273.
- 34 BACA, T.; PENICKA, R.; STEPAN, P.; PETRLIK, M.; SPURNY, V.; HERT, D.; SASKA, M. Autonomous cooperative wall building by a team of unmanned aerial vehicles in the mbzirc 2020 competition. **arXiv e-prints**, p. arXiv–2012, 12 2020.
- 35 BROCKMAN, G.; CHEUNG, V.; PETTERSSON, L.; SCHNEIDER, J.; SCHULMAN, J.; TANG, J.; ZAREMBA, W. Openai gym. **arXiv preprint arXiv:1606.01540**, 2016.
- 36 KAPLAN, R.; SAUER, C.; SOSA, A. Beating atari with natural language guided reinforcement learning. **arXiv preprint arXiv:1704.05539**, 2017.
- 37 JU, C.; LUO, Q.; YAN, X. Path planning using an improved a-star algorithm. In: IEEE. **2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)**. [S.l.], 2020. p. 23–26.

- 38 SONG, S.; ZHANG, Y.; QIN, X.; SAUNDERS, K.; LIU, J. Vision-guided Collision Avoidance Through Deep Reinforcement Learning. In: **NAECON 2021 - IEEE National Aerospace and Electronics Conference**. [S.l.: s.n.], 2021. p. 191–194. ISSN: 2379-2027.
- 39 RAFFIN, A.; HILL, A.; GLEAVE, A.; KANERVISTO, A.; ERNESTUS, M.; DORMANN, N. Stable-baselines3: Reliable reinforcement learning implementations. **Journal of Machine Learning Research**, v. 22, n. 268, p. 1–8, 2021. Disponível em: <<http://jmlr.org/papers/v22/20-1364.html>>.
- 40 HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE transactions on Systems Science and Cybernetics**, IEEE, v. 4, n. 2, p. 100–107, 1968.
- 41 DIJKSTRA, E. W. A note on two problems in connexion with graphs. In: **Edsger Wybe Dijkstra: His Life, Work, and Legacy**. [S.l.: s.n.], 2022. p. 287–290.
- 42 KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. **The international journal of robotics research**, Sage Publications Sage CA: Thousand Oaks, CA, v. 5, n. 1, p. 90–98, 1986.
- 43 HOLDSWORTH, J. The nature of breadth-first search. 02 1999.
- 44 CHOSET, H.; LYNCH, K. M.; HUTCHINSON, S.; KANTOR, G. A.; BURGARD, W. **Principles of robot motion: theory, algorithms, and implementations**. [S.l.]: MIT press, 2005.
- 45 CHANDAK, Y.; THEOCHAROUS, G.; KOSTAS, J.; JORDAN, S.; THOMAS, P. Learning action representations for reinforcement learning. In: PMLR. **International conference on machine learning**. [S.l.], 2019. p. 941–950.
- 46 LI, S.; CHEN, X.; ZHANG, M.; JIN, Q.; GUO, Y.; XING, S. A UAV Coverage Path Planning Algorithm Based on Double Deep Q-Network. In: **Journal of Physics: Conference Series**. IOP Publishing Ltd, 2022. v. 2216. ISSN: 17426588 Issue: 1. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85128449651&doi=10.1088%2F1742-6596%2F2216%2F1%2F012017&partnerID=40&md5=de323dab220328ab7eae27b25997df48>>.
- 47 WATKINS, C. J.; DAYAN, P. Q-learning. **Machine learning**, Springer, v. 8, p. 279–292, 1992.
- 48 MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; GRAVES, A.; ANTONOGLOU, I.; WIERSTRA, D.; RIEDMILLER, M. Playing atari with deep reinforcement learning. **arXiv preprint arXiv:1312.5602**, 2013.
- 49 SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O. Proximal policy optimization algorithms. **arXiv preprint arXiv:1707.06347**, 2017.
- 50 LILLICRAP, T. P.; HUNT, J. J.; PRITZEL, A.; HEES, N.; EREZ, T.; TASSA, Y.; SILVER, D.; WIERSTRA, D. Continuous control with deep reinforcement learning. **arXiv preprint arXiv:1509.02971**, 2015.
- 51 NAIR, A.; SRINIVASAN, P.; BLACKWELL, S.; ALCICEK, C.; FEARON, R.; MARIA, A. D.; PANNEERSHELVAM, V.; SULEYMAN, M.; BEATTIE, C.; PETERSEN, S. et al. Massively parallel methods for deep reinforcement learning. **arXiv preprint arXiv:1507.04296**, 2015.

- 52 HASSELT, H. V.; GUEZ, A.; SILVER, D. Deep reinforcement learning with double q-learning. In: **Proceedings of the AAAI conference on artificial intelligence**. [S.l.: s.n.], 2016. v. 30, n. 1.
- 53 OU, J.; GUO, X.; LOU, W.; ZHU, M. Quadrotor autonomous navigation in semi-known environments based on deep reinforcement learning. **Remote Sensing**, v. 13, n. 21, 2021. ISSN 20724292. Publisher: MDPI. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85118477341&doi=10.3390/rs13214330&partnerID=40&md5=1b1aaf57c2a0cbbbcba617952dd70f4>>.
- 54 SEEL, A.; KREUTZJANS, F.; KÜSTER, B.; STONIS, M.; OVERMEYER, L. Deep Reinforcement Learning Based UAV for Indoor Navigation and Exploration in Unknown Environments. In: **2022 8th International Conference on Control, Automation and Robotics (ICCAR)**. [S.l.: s.n.], 2022. p. 388–393. ISSN: 2251-2454.
- 55 XU, Y.; WEI, Y.; JIANG, K.; WANG, D.; DENG, H. Multiple UAVs Path Planning Based on Deep Reinforcement Learning in Communication Denial Environment. **Mathematics**, v. 11, n. 2, 2023. ISSN 22277390. Publisher: MDPI. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85146786977&doi=10.3390/math11020405&partnerID=40&md5=1c8f4b61f12ae23dcca2a2102046de0a>>.
- 56 SINGH, A.; JHA, S. S. Learning Safe Cooperative Policies in Autonomous Multi-UAV Navigation. In: **2021 IEEE 18th India Council International Conference (INDICON)**. [S.l.: s.n.], 2021. p. 1–6. ISSN: 2325-9418.
- 57 MA, Y.; XU, Y. A DDQN-Based Path Planning Method for Multi-UAVs in a 3D Indoor Environment. In: **2022 4th International Conference on Control and Robotics (ICCR)**. [S.l.: s.n.], 2022. p. 476–480.
- 58 CHEN, Y.; DONG, Q.; SHANG, X.; WU, Z.; WANG, J. Multi-UAV Autonomous Path Planning in Reconnaissance Missions Considering Incomplete Information: A Reinforcement Learning Method. **Drones**, v. 7, n. 1, 2023. ISSN 2504446X. Publisher: MDPI. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85146743581&doi=10.3390/drones7010010&partnerID=40&md5=32755470063f0e6c7fd5fb4cd2bca148>>.
- 59 BYRD, R. H.; LU, P.; NOCEDAL, J.; ZHU, C. A limited memory algorithm for bound constrained optimization. **SIAM Journal on scientific computing**, SIAM, v. 16, n. 5, p. 1190–1208, 1995.
- 60 BENGIO, Y.; LOURADO, J.; COLLOBERT, R.; WESTON, J. Curriculum learning. In: **Proceedings of the 26th annual international conference on machine learning**. [S.l.: s.n.], 2009. p. 41–48.

APÊNDICES

APÊNDICE A – Implementação da Grade Discreta

```
1  #include <geometry_msgs/Point.h>
2
3  class DiscreteGrid {
4  private:
5      float resolution;
6
7  public:
8      DiscreteGrid(const float &resolution) : resolution(
          resolution) { }
9
10     geometry_msgs::Point discrete_to_continuous(const
          geometry_msgs::Point &point) {
11         geometry_msgs::Point p;
12         p.x = point.x * resolution;
13         p.y = point.y * resolution;
14         p.z = point.z * resolution;
15         return p;
16     }
17     geometry_msgs::Point continuous_to_discrete(const
          geometry_msgs::Point &point) {
18         geometry_msgs::Point p;
19         p.x = std::floor(point.x / resolution);
20         p.y = std::floor(point.y / resolution);
21         p.z = std::floor(point.z / resolution);
22         return p;
23     }
24 };
```

APÊNDICE B – Módulo para Análise das Estatísticas Apartir dos Caminhos

```
1 import pandas as pd
2
3 df = pd.read_csv('Experimento.csv')
4 # distancia total...
5 df['delta_x'] = df['x'].diff()
6 df['delta_y'] = df['y'].diff()
7 df['delta_z'] = df['z'].diff()
8 df['distance'] = np.sqrt(df['delta_x']**2 + df['delta_y']
9     ]**2 + df['delta_z']**2)
10
11 # velocidade media...
12 mean_velocity = df['distance'].sum() / (df['timestamp'].
13     iloc[-1] - df['timestamp'].iloc[0])
14
15 # tempo de voo...
16 fly_time = df['timestamp'].iloc[-1] - df['timestamp'].iloc
17     [0]
18
19 # altitude media...
20 mean_height = df['z'].mean()
21
22 # variacao de altitude...
23 var_height = df['z'].max() - df['z'].min()
24
25 # show statistics...
26 print(f'Distancia total percorrida: {distancia_total} m')
27 print(f'Velocidade media: {mean_velocity} m/s')
28 print(f'Tempo de voo: {fly_time} s')
```

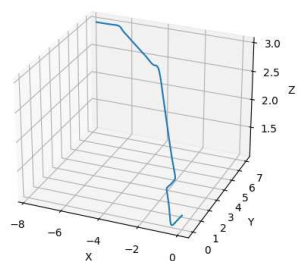
```
27 print(f'Altitude media: {mean_height} m')
28 print(f'Variacao de altitude: {var_height} m')
29 '''
```

APÊNDICE C – Módulo para Plotagem 3D dos Caminhos

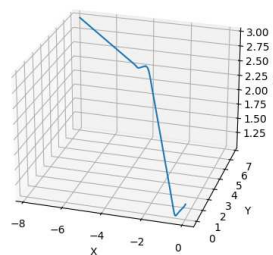
```
1 import matplotlib.pyplot as plt
2 from mpl_toolkits.mplot3d import Axes3D
3
4 df = pd.read_csv('Experimento.csv')
5
6 # visualizacao 3D do caminho...
7 fig = plt.figure()
8 ax = fig.add_subplot(111, projection='3d')
9 ax.plot(df['x'], df['y'], df['z'])
10 ax.set_xlabel('X')
11 ax.set_ylabel('Y')
12 ax.set_zlabel('Z')
13 plt.show()
```

APÊNDICE D – Caminhos dos Experimentos da Abordagem Clássica

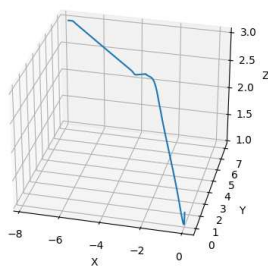
Figura 32 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 1.



(a) Cenário 1



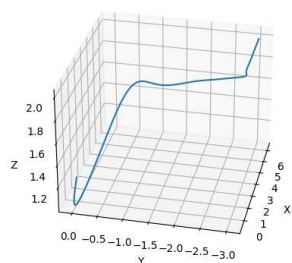
(b) Cenário 2



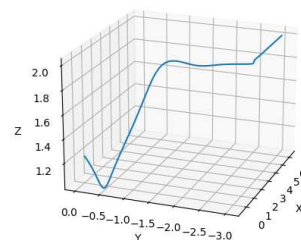
(c) Cenário 3

Fonte: O próprio autor.

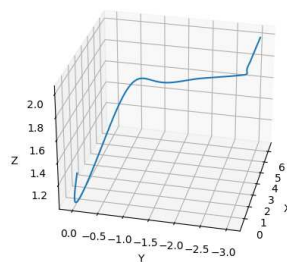
Figura 33 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 2.



(a) Cenário 1



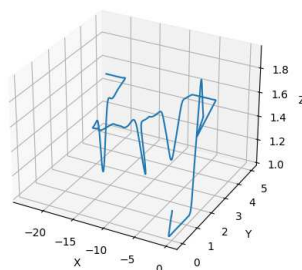
(b) Cenário 2



(c) Cenário 3

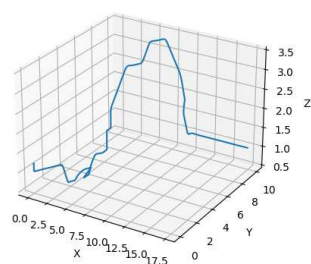
Fonte: O próprio autor.

Figura 34 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 4.

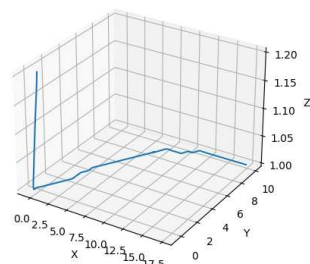


Fonte: O próprio autor.

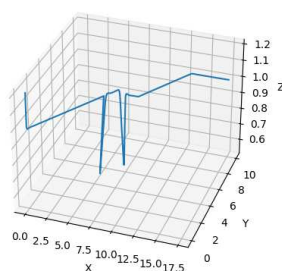
Figura 35 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 5.



(a) Cenário 1



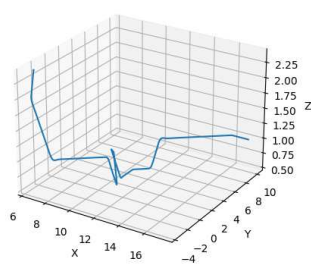
(b) Cenário 2



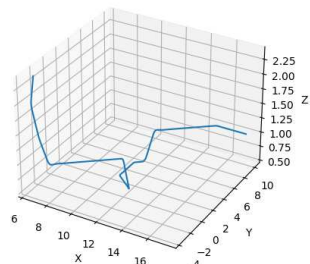
(c) Cenário 3

Fonte: O próprio autor.

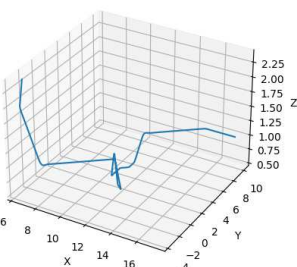
Figura 36 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 6.



(a) Cenário 1



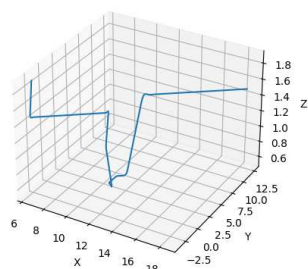
(b) Cenário 2



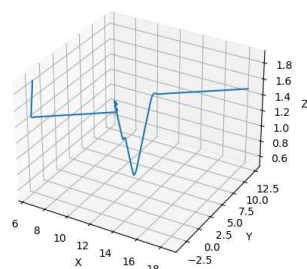
(c) Cenário 3

Fonte: O próprio autor.

Figura 37 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 7.



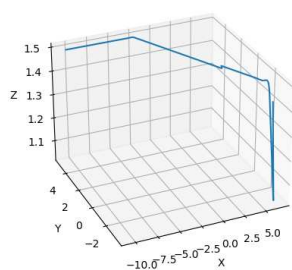
(a) Cenário 1



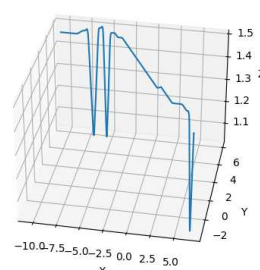
(b) Cenário 2

Fonte: O próprio autor.

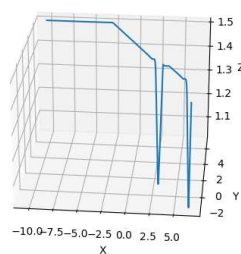
Figura 38 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 8.



(a) Cenário 1



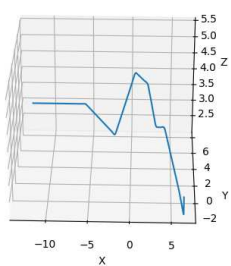
(b) Cenário 2



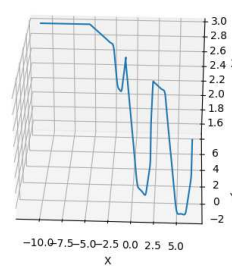
(c) Cenário 3

Fonte: O próprio autor.

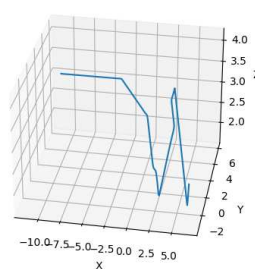
Figura 39 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 9.



(a) Cenário 1



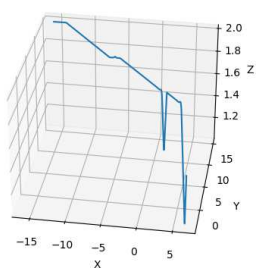
(b) Cenário 2



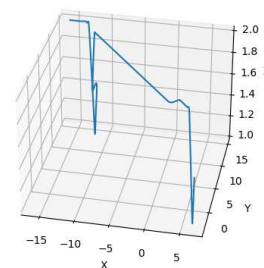
(c) Cenário 3

Fonte: O próprio autor.

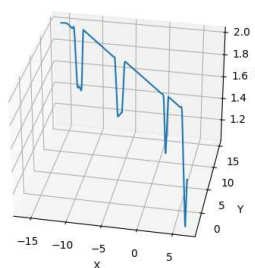
Figura 40 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 10.



(a) Cenário 1



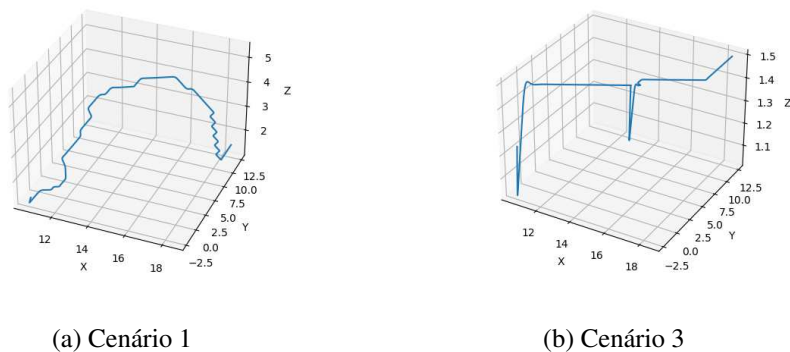
(b) Cenário 2



(c) Cenário 3

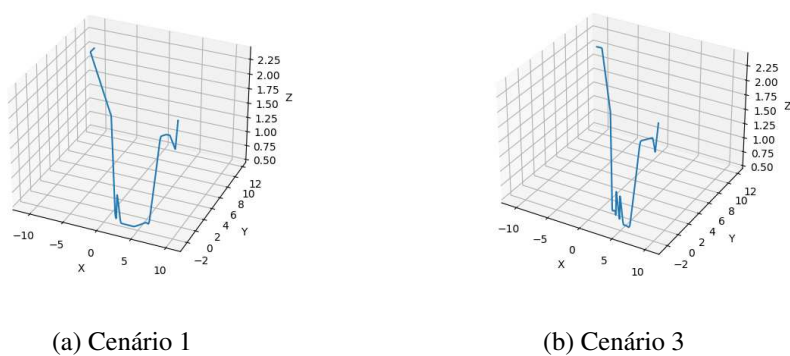
Fonte: O próprio autor.

Figura 41 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 11.



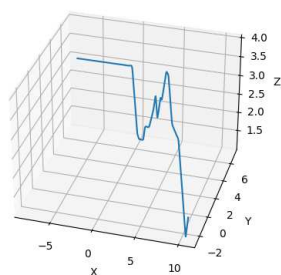
Fonte: O próprio autor.

Figura 42 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 12.



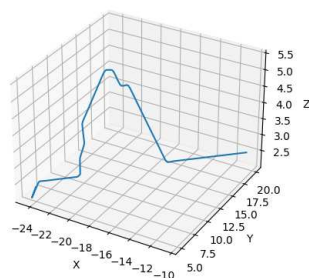
Fonte: O próprio autor.

Figura 43 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 13.



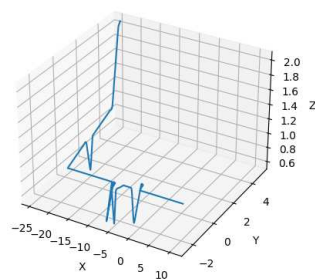
Fonte: O próprio autor.

Figura 44 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 15.



Fonte: O próprio autor.

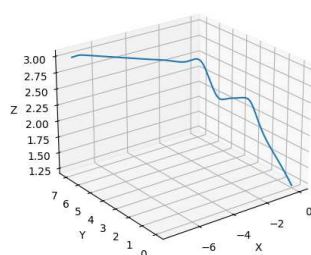
Figura 45 – Caminho realizado pelo VANT com a abordagem clássica. Experimento 15.



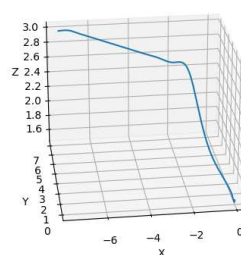
Fonte: O próprio autor.

APÊNDICE E – Caminhos dos Experimentos da Abordagem Clássica sem a Modelagem 3D dos Obstáculos

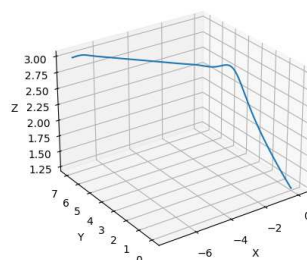
Figura 46 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 1.



(a) Cenário 1



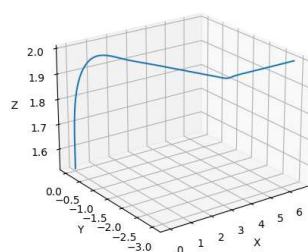
(b) Cenário 2



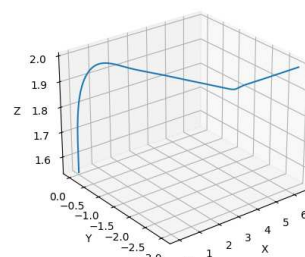
(c) Cenário 3

Fonte: O próprio autor.

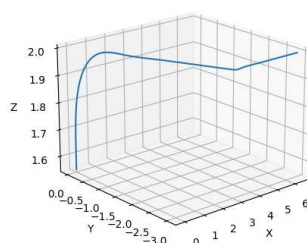
Figura 47 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 2.



(a) Cenário 1



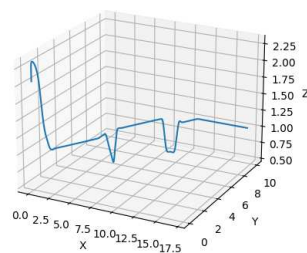
(b) Cenário 2



(c) Cenário 3

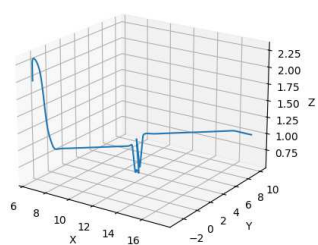
Fonte: O próprio autor.

Figura 48 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 4.

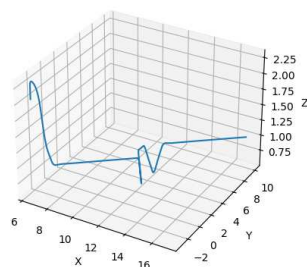


Fonte: O próprio autor.

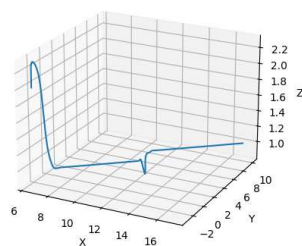
Figura 49 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 6.



(a) Cenário 1



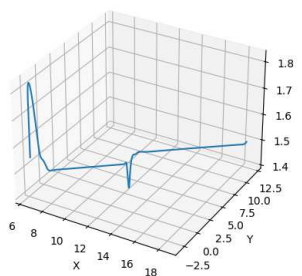
(b) Cenário 2



(c) Cenário 3

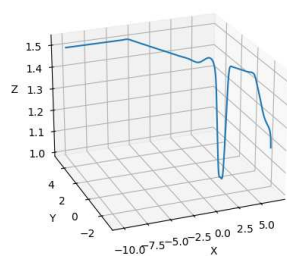
Fonte: O próprio autor.

Figura 50 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 7.

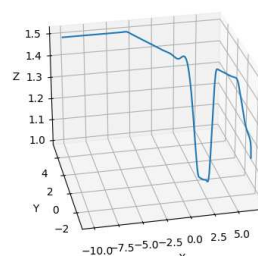


Fonte: O próprio autor.

Figura 51 – Caminho realizado pelo VANT com a abordagem clássica 2D. Experimento 8.



(a) Cenário 1



(b) Cenário 3

Fonte: O próprio autor.