

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**DESVIO DE OBSTÁCULOS PARA UAVS EM
AMBIENTES DINÂMICOS**

PEDRO HENRIQUE CORRÊA KIM

**ORIENTADORA: PROFA. DRA. KELEN CRISTIANE TEIXEIRA
VIVALDINI**

Sorocaba – SP
13 de junho de 2024

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**DESVIO DE OBSTÁCULOS PARA UAVS EM
AMBIENTES DINÂMICOS**

PEDRO HENRIQUE CORRÊA KIM

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial

Orientadora: Profa. Dra. Kelen Cristiane Teixeira Vivaldini

Sorocaba – SP

13 de junho de 2024

ABSTRACT

Kim, P. H. C. (2023). Desvio de obstáculos para UAVs em ambientes Dinâmicos. 62p
Dissertação – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2023.

O crescente uso e acessibilidade dos UAVs (*Unmanned Aerial Vehicles*) proporcionaram uma oportunidade para a automação em diversas aplicações. Uma área de grande interesse é o desenvolvimento de UAVs autônomos para realizar tarefas como inspeção e exploração de ambientes inacessíveis ou perigosos. Independente da aplicação, a capacidade do UAVs de planejar sua própria trajetória em diferentes condições é crucial. Nesse contexto, existe um esforço significativo para desenvolver algoritmos que assegurem a autonomia dos UAVs, permitindo interações diversificadas com o ambiente sem a necessidade de intervenção humana. Embora haja vários algoritmos capazes de realizar o planejamento de trajetória em ambientes desconhecidos e não estruturados, a presença de obstáculos dinâmicos representa um desafio substancial. A latência dos sensores e a natureza dos algoritmos existentes limitam sua eficácia em ambientes com obstáculos em movimento. Diante desse cenário, esta dissertação de mestrado propõe o desenvolvimento de um algoritmo inovador capaz de lidar com esses desafios, utilizando técnicas adaptativas provenientes de algoritmos mais rápidos de esquiva de obstáculos e detecção de obstáculos. O objetivo é proporcionar uma solução robusta e confiável para o planejamento autônomo de trajetória em ambientes dinâmicos, onde métodos convencionais enfrentam dificuldades e podem apresentar riscos à segurança.

Palavras-chave: UAV, Drone, Robôs autônomos, Evitar colisões, Detecção de obstáculos, Obstáculos dinâmicos

ABSTRACT

Kim, P. H. C. (2023). UAV Collision Avoidance for Dynamic Environments 62p Masters Thesis – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2023.

The growing use and accessibility of Unmanned Aerial Vehicles (UAVs) have provided an opportunity for automation in various applications. One area of great interest is the development of autonomous UAVs to perform tasks such as inspecting and exploring inaccessible or dangerous environments. Regardless of the application, the ability of UAVs to plan their own trajectory under different conditions is crucial. In this context, there is a significant effort to develop algorithms that ensure the autonomy of UAVs, allowing diverse interactions with the environment without the need for human intervention. Although there are various algorithms capable of performing trajectory planning in unknown and unstructured environments, the presence of dynamic obstacles poses a substantial challenge. The latency of sensors and the nature of existing algorithms limit their effectiveness in environments with moving obstacles. Given this scenario, this master's dissertation proposes the development of an innovative algorithm capable of addressing these challenges, using adaptive techniques from faster obstacle avoidance and obstacle detection algorithms. The goal is to provide a robust and reliable solution for autonomous trajectory planning in dynamic environments, where conventional methods face difficulties and may pose safety risks.

Keywords: UAV, Drone, Collision avoidance, Autonomous Vehicles, Obstacle detection, Dynamic obstacles.

LISTA DE FIGURAS

3.1	<i>Diagrama da Detecção de Obstáculos</i>	26
3.2	Diagrama da Detecção de Obstáculos	26
3.3	Exemplo do resultado da medição de U-map e V-map em simulação	30
3.4	<i>Diagrama de funcionamento do MPC</i>	35
3.5	UAV F450 - Ambiente Simulado Gazebo.	36
4.1	Comparação do mesmo mapa de profundidade sob o algoritmo U-map e UV-map	39
4.2	<i>Configuração no início da simulação da simulação</i>	40
4.3	<i>Configuração no fim da simulação da simulação</i>	41
4.4	<i>Trajetória de voo bem sucedido a 2 m/s.</i>	43
4.5	Exemplo de voo bem sucedido dos testes preliminares a 2 m/s. Link para o video das imagens: https://youtu.be/uJgZj4NwSmY	43
4.6	<i>Ambiente de testes para comparação de métodos de desvio</i>	45
4.7	Exemplo de voo que falhou enquanto o obstáculo atingia na parte traseira do UAV a 1 m/s. Link para o video das imagens: https://youtu.be/r1BZpuWfIg0	48
4.8	<i>Trajetória de voo que falhou enquanto o obstáculo atingia na parte traseira do UAV a 1 m/s.</i>	49
4.10	<i>Trajetória de voo que foi bem sucedido no voo a 1 m/s.</i>	49
4.9	Exemplo de voo que foi bem sucedido no voo a 1 m/s. Link para o video das imagens: https://youtu.be/LcdCtfYK-ko	50
4.11	Exemplo de voo a 2 m/s que foi bem sucedido no voo. Link para o video das imagens: https://youtu.be/WDECSvbOP3o	51

4.12 <i>Trajétoria de voo que falhou enquanto o obstáculo atingia na parte traseira do UAV a 1 m/s.</i>	51
---	----

LISTA DE TABELAS

4.1	<i>Resultados de AGV por simulação dos testes preliminares.</i>	42
4.2	<i>Resultados de tempo para recalcular a rota e para detectar os obstáculos com o UV-map</i>	46
4.3	<i>Resultados de tempo para recalcular a rota e para detectar os obstáculos U-map</i>	46
4.4	Resultados de AGV por simulação utilizando o U-map	52
4.5	<i>Resultados de AGV por simulação utilizando o UV-map</i>	53
4.6	<i>Resultados de AGV por simulação utilizando o UV-map e o Plannie</i>	55
4.7	Resultados de SR para cada velocidade e algoritmo	56
4.8	Resultados de AGV para cada velocidade e algoritmo	56

GLOSSÁRIO

ABT – *Argumented Belief Trees*

AGV – *Average Goal Velocity*

APF – *Artificial Potential Field*

CCNMPC – *Chance-Constrained Nonlinear Model Predictive Control*

CNN – *Convolutional Neural Network*

EKF – *Extended Kalman Filter*

IA – *Inteligência Artificial*

IRP – *Improved Rolling Plan*

LARIS – *Laboratório de Robôs Autônomos e Sistemas Inteligentes*

LASI – *Laboratório de Sistemas Inteligentes*

LiDAR – *Light Detection and Ranging*

MPC – *Model Predictive Control*

MRS – *Multi Robot Systems*

PPOMCP – *Partially Observable Monte Carlo Planning*

PT – *Processing Time*

ROS – *Robot Operating System*

RRT-C – *Rapid Exploring Random Tree Connect*

RRT – *Rapid Exploring Random Tree*

SLAM – *Simultaneous Localization and Mapping*

SR – *Success Rate*

UAV – *Unmanned Aerial Vehicle*

UFSCar – *Universidade Federal de São Carlos*

SUMÁRIO

GLOSSÁRIO

CAPÍTULO 1 – INTRODUÇÃO	11
1.1 Objetivo	14
1.2 Proposta	15
1.3 Estrutura do Trabalho	16
CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA	17
2.1 Métodos de IA	17
2.2 Algoritmos de amostragem e desvios de obstáculos	19
2.2.1 Métodos de Desvio de Obstáculos	19
2.2.2 Métodos de Amostragem do Ambiente	22
2.3 Considerações finais	24
CAPÍTULO 3 – MÉTODO PROPOSTO	25
3.1 Detecção de Obstáculos	25
3.2 Transformação das Coordenadas de Referências	30
3.3 Estimativa de Velocidade	32
3.4 Correção de trajetórias e MPC	34
3.5 Materiais	35
3.5.1 Recursos simulados	35

3.6	Considerações finais	36
CAPÍTULO 4 – RESULTADOS		37
4.1	Métricas	38
4.2	Diferenças entre o U-map e o UV-map	39
4.3	Ambiente de testes preliminares do método proposto	39
4.4	Testes sem o UV-Map	41
4.5	MPC Auxiliado do U-map e V-map	42
4.6	Análise comparativa	43
4.6.1	Ambiente para análise comparativa	44
4.6.2	Testes de PT	45
4.6.3	Testes de SR e AGV	47
4.7	Considerações finais	55
CAPÍTULO 5 – CONCLUSÕES E TRABALHOS FUTUROS		58
5.1	Considerações Finais	58
5.2	Trabalhos Futuros	59
REFERÊNCIAS		60

Capítulo 1

INTRODUÇÃO

Explorando o campo da robótica móvel, encontramos uma subárea dedicada ao desenvolvimento de métodos para garantir a autonomia dos Veículos Aéreos Não Tripulados (*Unmanned Aerial Vehicles* - UAVs), também chamados de drones. A aplicação de UAVs autônomos tem demonstrado benefícios significativos em diversas áreas, como evidenciado por estudos recentes (FREIMUTH; KÖNIG, 2018; LEE; HO; ZHOU, 2021; LOQUERCIO et al., 2019). Essa autonomia é essencial para permitir que os UAVs tomem decisões cruciais, assegurando a integridade da aeronave e o cumprimento eficaz das missões atribuídas.

Um dos elementos fundamentais para a autonomia dos UAVs é a capacidade de planejar trajetórias de forma autônoma, que possa atualizar a trajetória dinamicamente e que possa identificar obstáculos que eventualmente sejam dinâmicos e desviar deles. Esse processo permite que um UAV crie um caminho que conecte um ponto inicial a um ponto final sem a intervenção direta de um operador (ROCHA, 2021). Para alcançar um planejamento eficiente, é necessário que o UAV obtenha dados do ambiente, proporcionando a criação de trajetórias seguras e eficazes que a conduza até o seu objetivo.

No contexto dos UAVs autônomos, certas características são essenciais para garantir não apenas a geração de rotas seguras, mas também a plena utilização dos graus de liberdade dessas aeronaves. Estas incluem:

- Navegar em três dimensões: Um UAV pode movimentar-se em três dimensões, e um algoritmo de geração de trajetórias ideal deve ser capaz de utilizar todas essas dimensões. Isso é crucial, especialmente quando o UAV precisa contornar obstáculos por cima ou por baixo em determinadas circunstâncias (ROCHA, 2021);
- Atuar em ambientes desconhecidos: Muitas missões expõem os UAVs a ambientes completamente desconhecidos. Portanto, os UAVs devem ser capazes de realizar reconheci-

mento do espaço, tomar decisões com base em informações limitadas e replanejar suas trajetórias conforme mais dados sobre o ambiente são descobertos;

- Atuar em ambientes não-estruturados: Ambientes não-estruturados apresentam obstáculos dispostos de maneira aleatória, sem um padrão previsível. Os UAVs autônomos devem ser capazes de lidar com essa imprevisibilidade na disposição dos obstáculos;
- Atuar em ambientes dinâmicos: Ambientes dinâmicos são caracterizados por mudanças constantes. Em situações reais, os UAVs precisam realizar mapeamento constante para redefinir trajetórias, desviando de elementos dinâmicos que possam interferir ou remapeando obstáculos deslocados (LIN; ZHU; ALONSO-MORA, 2020).

Embora tenham sido desenvolvidos diversos algoritmos de planejamento de trajetória capazes de lidar com obstáculos dinâmicos em ambientes simulados, a reprodutibilidade desses algoritmos em ambientes reais é desafiada devido à necessidade de mapeamento do ambiente em tempo real, bem como as limitações de processamento embarcado (YAO; ZHAO, 2015; QIAN et al., 2015; GONZÁLEZ-DESANTOS et al., 2021). Essa discrepância entre simulações controladas e situações reais destaca a complexidade e os desafios enfrentados na implementação prática desses sistemas autônomos, acrescentando além da questão que antecede à pergunta de como se deslocar dentro do ambiente, que é definir quais são as restrições para esse deslocamento.

Paralelamente aos avanços em algoritmos de planejamento de trajetórias, também tem sido uma prioridade o desenvolvimento de métodos específicos para o desvio de obstáculos dinâmicos (LIN; ZHU; ALONSO-MORA, 2020; LOQUERCIO et al., 2019; WANG et al., 2020; DAI et al., 2020). Embora esse problema seja semelhante ao planejamento de trajetórias, as preocupações aqui são distintas, pois esses algoritmos buscam desviar de obstáculos móveis, demandando respostas rápidas para corrigir a trajetória do veículo autônomo e evitar colisões (LIN; ZHU; ALONSO-MORA, 2020).

No entanto, a complexidade do desvio de obstáculos dinâmicos vai além da simples reação do UAV a obstáculos em seu caminho. O primeiro desafio consiste na identificação eficaz desses obstáculos dinâmicos (SAHA et al., 2022; LIN; ZHU; ALONSO-MORA, 2020; ZHONG et al., 2024). Somente após essa identificação, é possível planejar trajetórias que contornem esses obstáculos dinâmicos com sucesso (ZHU; ALONSO-MORA, 2019). Dada a natureza crítica das respostas rápidas necessárias para evitar colisões, os algoritmos de desvio de obstáculos não apenas requerem velocidade para recalcular trajetórias, mas também uma taxa de amostragem do ambiente ofereça os dados dos obstáculos com precisão e tão rápida quanto possível (YUAN

et al., 2023).

Existem diversos algoritmos que podem fazer esse tipo de identificação e adaptar de diversas maneiras, por exemplo Lu et al. (2021) adapta o *Artificial Potential Field* para utilizar como um detector de obstáculos, enquanto Oleynikova, Honegger e Pollefeys (2015) criou e testou o U-map em um contexto de desvio de obstáculos estáticos que continuou sendo reproduzido em situações de obstáculos dinâmicos Lin, Zhu e Alonso-Mora (2020) e Saha et al. (2022) combinou o U-map com o V-map para melhorar a precisão para delimitar os obstáculos.

Estudos mais antigos, como o conduzido por Qian et al. (2015), já exploravam a possibilidade de lidar com obstáculos dinâmicos. Nesse estudo, o algoritmo de *Artificial Potential Field* (APF) foi testado em conjunto com o *Improved Rolling Plan* (IRP) para otimizar o uso dos campos potenciais, utilizando o IRP em situações em que o campo potencial do APF é reduzido e alternando para o APF quando a trajetória é submetida a um certo valor limítrofe dos campos do APF. No entanto, é importante observar que, embora o algoritmo tenha obtido sucesso em evitar colisões, esses testes foram conduzidos em simulações que não consideravam as limitações de movimento de um UAV, como a aceleração, assumindo que os obstáculos são conhecidos e restringindo-se a movimentos em 2D, mesmo em ambientes tridimensionais (ROCHA, 2021).

Alinhando-se a essa eficácia demonstrada pelos métodos clássicos, Lu et al. (2021) realizou experimentos com um UAV real, utilizando a estratégia que combina APF para evitar colisões e o algoritmo de *Rapidly-Exploring Random Trees* (RRT) para planejar trajetórias, nesse trabalho, os experimentos com obstáculos estáticos foram realizados em ambientes reais e os experimentos com obstáculos dinâmicos foram obtidos em ambientes simulados no software Gazebo.

No trabalho em questão, os resultados do número de sucessos em realizar as missões propostas e velocidade média obtidos destacam que o desvio de obstáculos dinâmicos não depende apenas de algoritmos capazes de gerar trajetórias livres de colisões para o UAV. É igualmente crucial ter algoritmos que tirem proveito dos dados dos sensores, amostrando o ambiente com precisão e velocidade suficientes para permitir ao UAV a capacidade de ajustar sua trajetória em tempo real.

É notável que, apesar de o desvio de obstáculos dinâmicos ter sido uma área substancialmente explorada, a contribuição significativa não reside apenas na melhoria dos métodos de desvio, mas também na integração eficaz desses métodos com estratégias de planejamento de trajetórias. Essa sinergia é fundamental para garantir a navegabilidade segura dos UAVs em ambientes dinâmicos e imprevisíveis.

Essa necessidade de não só ser capaz de reconhecer o ambiente para poder desviar de

obstáculos também se demonstra tão importante quanto a técnica de desvio em si, dado que em uma grande parte dos casos, identificar os obstáculos é necessário para o próprio funcionamento dos algoritmos de desvio de obstáculos. Se por um lado alguns métodos já integram os sensores com o método de desvio (LOQUERCIO et al., 2019; DAI et al., 2020), outros algoritmos precisam de algoritmos que geram essas informações para possibilitar o ajuste da trajetória (LU et al., 2021; LIN; ZHU; ALONSO-MORA, 2020; GONZÁLEZ-DESANTOS et al., 2021). Evidentemente que, tratando-se de campos distintos de pesquisa existem exemplos de métodos de detecção de obstáculos que ainda não foram explorados, mas que potencialmente úteis para a realização de desvios de obstáculos (SAHA et al., 2022; ZHONG et al., 2024), mais especificamente Saha et al. (2022) propôs uma melhoria do método U-map, um algoritmo testado em outros trabalhos de desvio de obstáculos dinâmicos (LIN; ZHU; ALONSO-MORA, 2020) e portanto é potencialmente um método ainda potencialmente útil para esse fim, mas que foi pouco explorado para esse fim.

1.1 Objetivo

O objetivo deste trabalho é desenvolver um algoritmo de desvio de obstáculos dinâmicos que seja adaptável a ambientes desconhecidos. Reconhecendo que existem algoritmos eficazes para tarefas semelhantes quando o ambiente é conhecido (LU et al., 2021; QIAN et al., 2015), optamos por uma abordagem concentrada na identificação de obstáculos dinâmicos em ambientes desconhecidos. A proposta final para atingir o objetivo proposto é integrar esses dados a métodos comprovadamente eficientes para evitar colisões com os obstáculos dinâmicos.

Para tal, foi necessário buscar um algoritmo que possa identificar os obstáculos dinâmicos com precisão e latência o suficiente para que os dados gerados possam alimentar um algoritmo que seja capaz de ajustar a trajetória do UAV a fim de que este seja capaz de ajustar o percurso para evitar as colisões com os obstáculos dinâmicos. Outra determinação importante foram quais métricas deveriam ser utilizadas para que fosse possível alguma comparação, dado que a de forma geral os trabalhos ainda não estão alinhados com um padrão para comparação entre os resultados experimentais dos algoritmos (YU; CROON; WAGTER, 2023). No entanto, Yu, Croon e Wagter (2023), propuseram uma série de métricas para avaliar o desempenho de veículos autônomos a fim de possibilitar comparações de resultados.

Para atender aos requisitos do problema, a integração de algoritmos de identificação e de ajuste de trajetória realizará as seguintes operações:

- Identificar obstáculos dinâmicos, o vetor velocidade e estimar sua posições futuras base-

ado na projeção da mesma;

- Identificar riscos de colisão;
- Ajustar uma trajetória inicial do UAV, caso haja um potencial risco de colisão com os obstáculos identificados.

Essa abordagem visa não apenas aprimorar a capacidade de desvio de obstáculos dinâmicos, mas também a adaptabilidade do UAV a ambientes desconhecidos, estendendo o escopo de sua autonomia operacional.

1.2 Proposta

Com base no objetivo delineado, e definida a necessidade de um algoritmo para desvio de obstáculos integrado a um algoritmo de detecção de obstáculos, este estudo propõe a integração dos métodos de detecção de obstáculos UV-map (SAHA et al., 2022), por ser um método de detecção como sendo uma melhoria do método de U-map (OLEYNIKOVA; HONEGGER; POLLEFEYS, 2015) que por sua vez já foi testado para o propósito de desvio de obstáculos dinâmicos, mas em contra partida, o UV-map oferece mais precisão na determinação dos limites no eixo Y da câmera e que, com base na informação de odometria e no filtro de Kalman, estima as trajetórias dos obstáculos detectados para a criação de trajetórias compatíveis para o consumo do *trajectory tracker* do MRS (*Multi-robot Systems*) (BACA et al., 2018, 2021), esse por sua vez foi escolhido pela versatilidade na integração de métodos de controle e de controle de colisões sendo assim uma ferramenta extremamente versátil para contornar obstáculos dinâmicos. Além disso, busca-se avaliar a eficácia da incorporação de um algoritmo de planejamento de caminho, especificamente o *Rapidly-exploring Random Tree Connect* (RRT-C) implementado por (ROCHA, 2021), como um planejador de missões. A avaliação será realizada utilizando algumas das métricas propostas por (YU; CROON; WAGTER, 2023), nomeadamente a *Success Rate* (SR), o *Average Goal Velocity* (AGV) e o Tempo de Processamento (PT). Este estudo representa uma contribuição científica ao explorar a sinergia entre detecção de obstáculos, estratégias de desvio e planejamento de trajetória em ambientes dinâmicos, visando aprimorar a eficiência e robustez de sistemas de UAV autônomos.

1.3 Estrutura do Trabalho

Este trabalho está dividido em 5 capítulos contando com este capítulo de Introdução. No Capítulo 2 apresenta-se a Revisão Bibliográfica referente aos trabalhos que obtiveram sucesso em criar ou alterar trajetórias para desviar de obstáculos dinâmicos, e com o embasamento destes trabalhos justifica-se o motivo da escolha da metodologia utilizada.

No Capítulo 3 explicam-se os algoritmos desenvolvidos e quais foram os recursos utilizados para desviar de obstáculos e para testar esse algoritmo.

No Capítulo 4 explora-se os resultados obtidos pelos testes através de métricas. E no Capítulo 5 apresenta-se as conclusões e os trabalhos futuros que podem ramificar a partir dos resultados obtidos e recursos desenvolvido.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

Com o intuito de explorar de forma abrangente de solução do problema de desvio de obstáculos dinâmicos para UAVs autônomos, realizamos uma revisão extensiva dos métodos que tratam do desvio de obstáculos dinâmicos. A bibliografia relacionada possui duas formas de abordagens distintas do problema. Uma delas emprega técnicas de Inteligência Artificial (IA) para realizações de diversos tipos de missões tendo que lidar com ambientes com obstáculos dinâmicos e portanto tendo que evitar colisões (WANG et al., 2020; LOQUERCIO et al., 2019; WANG et al., 2020; DAI et al., 2020), enquanto a outra adota a combinação de métodos de amostragem do ambiente em conjunto com algoritmos de planejamento de trajetórias ou métodos de desvio de obstáculos (LIN; ZHU; ALONSO-MORA, 2020; LU et al., 2021; SANDINO et al., 2020).

O motivo pelo qual está se estabelecendo essa divisão se dá pelo fato de serem metodologias muito distintas que são dificilmente combináveis entre elas, isso porque os métodos que usam IA são mais rígidos na utilização dos dados de sensores, tipicamente esses dados são inseridos em redes neurais e suas saídas são diretamente instruções para os atuadores com pouca ou nenhuma margem para modificações ou combinações com outras técnicas ou mesmo utilização de outros sensores. Dada essa diferença, esses dois tipos de abordagem serão discutidos em duas seções diferentes seguidas por uma conclusão de qual método se mostra mais adequado ao problema.

2.1 Métodos de IA

Dentre os métodos de IA avaliados, os que mais se destacaram na quantidade de sucessos em realizar suas missões são métodos que usam dados de voo simulados, reais ou uma combinação entre os dois para o treinamento de redes neurais, mais especificamente são métodos

de *Reinforcement Learning* (Aprendizado por Reforço) (WANG et al., 2020; LOQUERCIO et al., 2019; DAI et al., 2020; SADEGHI; LEVINE, 2016).

Um dos estudos que proporciona uma perspectiva relevante é o trabalho apresentado por Gandhi, Pinto e Gupta (2017), que empregou uma abordagem baseada em *Reinforcement Learning*. O estudo utilizou um ambiente simulado para treinar uma rede neural profunda com camadas convolucionais mas que deu destaque na produção de dados de erros de voo, com o objetivo de criar reforços negativos para ajudar a desenvolver um UAV capaz de voar autonomamente. Apesar de os resultados não terem sido expressivos se comparados com outros trabalhos posteriores, o autor menciona um conceito crucial que continuou sendo adotado posteriormente por outros trabalhos: a proposta de utilizar dados simulados e reais para treinar inteligências artificiais destinadas a operar em ambientes reais.

Esse conceito é corroborado pelo trabalho de Loquercio et al. (2019), que treinou uma *Convolutional Neural Network* (CNN) empregando uma combinação de dados reais e simulados de voos de drones. A CNN foi treinada para atravessar arcos de corrida de UAVs, fornecendo o vetor velocidade e tempo de execução como saídas. Os testes foram conduzidos em diversas circunstâncias adversas, e o UAV demonstrou não apenas a capacidade de atravessar os arcos em situações ideais, mas também obteve uma taxa de acerto de 100% em obstáculos que ocupavam até 40% da passagem. Mesmo diante de variações nos parâmetros, como iluminação e adição de movimentação nos arcos, o algoritmo conseguiu completar um percentual muito elevado das tentativas de percorrer o circuito de testes.

Os resultados obtidos no trabalho em questão não só demonstram a eficácia das CNNs para lidar com obstáculos dinâmicos mas também reforçam as conclusões obtidas por Sadeghi e Levine (2016), que demonstraram a possibilidade de treinar CNNs para navegação de UAVs utilizando somente informações obtidas de ambientes virtuais.

Um outro exemplo da versatilidade desse método foi evidenciada por Dai et al. (2020). Nesse cenário, dados provenientes de imagens capturadas de carros inicialmente coletadas para o treinamento de redes neurais para utilização em carros autônomos foram usados para o treinamento de uma rede neural destinada a um UAV. Essa adaptação essencialmente restringiu os graus de liberdade do movimento do UAV aos de um automóvel, permitindo apenas o eixo *yaw* e a velocidade como saída da rede neural. Importante mencionar que o UAV em questão utilizava exclusivamente uma câmera integrada, sendo todo o processamento conduzido por um computador externo. Muito embora o estudo não tenha registrado a taxa de sucesso nem tenha sido executado em um ambiente controlado com desafios planejados e reproduzíveis, ele utilizou a métrica de média de metros percorridos em cada cenário e demonstrou autonomia elevada dado

que a média dos testes de aproximação à distância do trajeto evidenciando uma notável eficiência das CNNs no desvio de obstáculos dinâmicos.

Nesses estudos, evidenciou-se que os métodos de IA apresentam duas limitações para aplicação. Em primeiro lugar, eles acarretam custos computacionais significativos e demandam um computador externo para processar os algoritmos na maioria dos casos (WANG et al., 2020; DAI et al., 2020). Embora essa constatação não seja universal, os trabalhos analisados indicam uma tendência de alto custo computacional associado a essas redes (LOQUERCIO et al., 2019). Essa exigência de processamento não é condizente com o poder computacional embarcado, o que é inadequado para o escopo deste trabalho, o qual visa proporcionar a maior autonomia possível ao UAV. O segundo motivo, embora menos relevante, é a necessidade de dados para o treinamento dos modelos, um recurso não prontamente disponível.

Outros trabalhos que não fizeram uso desse recurso combinaram uma gama de sensores para o mesmo fim e obtiveram sucesso nessa tarefa com um custo computacional factível para computadores embarcados processarem em tempo real.

2.2 Algoritmos de amostragem e desvios de obstáculos

Esta seção será dividida em duas partes, primeiramente será apresentada a bibliografia dos trabalhos relacionados com métodos de desvio de obstáculos dinâmicos em seguida serão apresentados os métodos de amostragem do ambiente.

2.2.1 Métodos de Desvio de Obstáculos

Entre os métodos de desvio de obstáculos, foram identificados vários estudos que adaptaram técnicas de *path planning* para esse propósito (QIAN et al., 2015; YAO; ZHAO, 2015; CHEN; ZHAO; YIN, 2020; LU et al., 2021; ZAMMIT; KAMPEN, 2018). Um dos métodos de *path planning* com uma quantidade significativa de trabalhos e resultados positivos foi o RRT (*Rapidly-exploring Random Tree*) (LAVALLE et al., 2001; ZAMMIT; KAMPEN, 2018). Em particular, Yao e Zhao (2015) evidenciaram que o RRT representa uma solução eficaz para lidar com obstáculos móveis em ambientes 2D, com essa conclusão corroborada por Lu et al. (2017), que realizaram um teste da técnica em ambientes 3D. Em ambos os casos, os experimentos foram conduzidos utilizando UAVs de asa fixa e o resultado observado dessa única simulação demonstrou a possibilidade dessa hipótese.

Ainda com o mesmo objetivo de tratar ambientes dinâmicos usando métodos de *path plan-*

ning, os experimentos envolvendo a adaptação do APF com o IRP (XI; ZHANG, 2002) *Improved Rolling Plan*, realizados por Qian et al. (2015), demonstraram também resultados positivos, muito embora pouco testado para se ter um resultado conclusivo da robustez do algoritmo. Nesse trabalho o IRP é utilizado para atingir o objetivo do voo, que consiste em usar o controle preditivo que no caso específico consiste em prever o cenário futuro, escolher o algoritmo de otimização adequado para os objetivos intermediários e por último atualizar o ambiente para repetir o processo, mas toda vez que os campos do APF atingem um certo valor absoluto o UAV passa a gerar a trajetória usando o APF para atingir o objetivo, voltando ao IRP se o valor absoluto dos campos APF reduzirem até a um valor abaixo do estabelecido. O algoritmo alterna entre os dois geradores de trajetórias durante o percurso até finalizar a missão, uma abordagem muito prática para combinar dois algoritmos em um, se aproveitando das vantagens de ambos nas circunstâncias mais favoráveis.

Outra abordagem que usa *path planning* foi abordada por Chen, Zhao e Yin (2020), que optaram por recriar rotas imediatamente depois da detecção de obstáculos móveis. No caso, o UAV utiliza o algoritmo A* (YAO et al., 2010) que faz uma busca heurística para avaliar a melhor trajetória. Essa ponderação é feita entre as funções de colisão R , consistência e custo de suavidade. Este padrão é alterado caso seja detectado uma ameaça móvel, a qual ocasiona na adição de uma função de custo de segurança dinâmica, que impõe mais exigências à trajetória baseado na posição e vetor velocidade do obstáculo. Logo, destaca-se o tratamento diferenciado de obstáculos dinâmicos e obstáculos estáticos que este trabalho propõe, reduzindo a dependência do rastreamento constante dos obstáculos, uma vez que o algoritmo usa a projeção do movimento esperado para prever rotas com risco de colisão, algo que os trabalhos anteriores citados não consideravam.

Até o presente momento, os exemplos elucidados empregaram simuladores de baixa fidelidade como ambiente de teste para avaliação de métodos de planejamento de trajetória. Esta escolha levanta inquietações quanto à transferibilidade e efetividade desses métodos para aplicações em ambientes reais. Contudo, um exemplar de aplicação prática desses métodos é destacado no trabalho de Lu et al. (2021), que adota técnicas clássicas de planejamento de trajetória para um UAV.

Neste estudo específico, uma abordagem integrativa de APF e RRT é implementada, utilizando um mapa do ambiente construído a partir de dados provenientes de um sensor LiDAR e uma câmera de profundidade. Essa estratégia foi testada em ambientes simulados e reais, revelando notável velocidade na identificação de obstáculos estáticos e dinâmicos, sucessos em desviar dos mesmos. Os resultados obtidos evidenciam a robustez e aplicabilidade desses

métodos em cenários práticos, corroborando sua eficácia em contextos do reais dado que o algoritmo demonstrou capacidade de reação efetiva a obstáculos dinâmicos dado o elevado valor do SR observado no ambiente testado, que batia o valor de 95%

Em uma derivação das soluções provenientes dos algoritmos de planejamento de trajetória propostos por Zhu e Alonso-Mora (2019), foi conduzido um estudo empregando o método de *Chance-Constrained Nonlinear Model Predictive Nontrol* (CCNMPC), conforme descrito por Kamel et al. (2017). Este método foi aplicado para o planejamento e execução de trajetórias, utilizando informações derivadas de obstáculos dinâmicos como fatores restritivos. O *Extended Kalman Filter* (EKF) foi empregado para criar uma sequência de pontos, delineando regiões de risco que o UAV deve respeitar, com o objetivo de evitar colisões com os obstáculos dinâmicos.

No trabalho em questão, propôs-se adotar um APF modificado para a geração da trajetória. A camada mais interna do APF foi definida como uma "restrição rígida", onde qualquer trajetória calculada deve estar em conformidade, sendo imposta uma trajetória abrupta, se necessário, para respeitar esses limites. Em contraste, uma camada mais externa incorporou a regra de campos potenciais típica do APF, visando uma alteração mais suavizada da trajetória, tornando-a mais compatível com a utilização eficaz do CCNMPC.

Novamente, a concepção de projetar as posições futuras dos obstáculos dinâmicos foi empregada, destacando-se pela contribuição essencial na solidez das soluções de geração de trajetória. Esta previsão conferiu robustez ao processo, permitindo antecipar as futuras posições dos obstáculos.

O estudo conduzido por Lin, Zhu e Alonso-Mora (2020) alinhou-se à mesma abordagem, contudo, em acréscimo ao algoritmo proposto por Zhu e Alonso-Mora (2019), incorporou o U-map para detectar a posição dos obstáculos em cada frame. Utilizando o mesmo EKF, o trabalho determinou a velocidade do UAV, tornando-se, assim, independente de qualquer informação externa para a determinação das posições dos obstáculos e suas respectivas velocidades. E o UAV usava esses dados de posição e velocidade no CCNMPC para definir rotas livres de colisão

A evidência definitiva da eficácia do método foi fornecida pelo seu teste em ambiente real, em uma abordagem diferente de outros trabalhos (KAMEL et al., 2017; ZHU; ALONSO-MORA, 2019). No cenário em questão, o UAV foi restrito a utilizar apenas a alteração do ângulo yaw para realizar desvios em torno dos obstáculos. Essa restrição decorre da dependência da detecção de obstáculos em relação ao campo de visão para identificar obstáculos e suas trajetórias. O método foi submetido a testes em ambientes dinâmicos, demonstrando a capacidade de evitar obstáculos móveis com tempos de resposta na ordem de dezenas de milissegundos.

Outra contribuição significativa no contexto de desvio de obstáculos dinâmicos foi apresentada por Baca et al. (2018), que desenvolveu um framework para controle multi-UAV. Os recursos para desvio de obstáculos nesse trabalho guardaram notável semelhança com os delineados por Zhu e Alonso-Mora (2019). Embora as propostas de ambos os trabalhos fossem distintas, compartilhavam o mesmo objetivo de assegurar trajetórias de voo seguras para os UAVs. Porém, o enfoque de Baca et al. (2018) era facilitar a convergência de trajetórias seguras através da hierarquização das rotas de voo de múltiplos UAVs.

De maneira específica, foi estabelecida uma hierarquia arbitrária entre os UAVs, os quais, por sua vez, geravam trajetórias acompanhadas por regiões cilíndricas de segurança. No caso de dois ou mais UAVs gerarem trajetórias inicialmente lineares que infringem uma região de segurança, os UAVs com menor hierarquia aumentam a altura dos pontos de violação até que estes estejam fora da região. Esse procedimento difere do método proposto por Zhu e Alonso-Mora (2019), que visa corrigir os pontos infringentes simultaneamente até a convergência de rotas seguras, conforme explicado anteriormente nesta subseção.

2.2.2 Métodos de Amostragem do Ambiente

Durante a revisão bibliográfica sobre o tema, observou-se que o método de amostragem do ambiente e identificação de obstáculos é tão crucial para o desvio de obstáculos quanto o próprio método de desvio. Isso se deve ao fato de que todos os métodos de desvio de obstáculos dependem de uma representação precisa do ambiente para poder criar trajetórias realmente seguras. A relevância do método de amostragem do ambiente é destacada por Falanga, Kim e Scaramuzza (2019), que conduziu um benchmark avaliando as limitações de velocidade de um UAV considerando diversos tipos de câmeras.

Um exemplo ilustrativo da importância da escolha apropriada do método de amostragem foi apresentado no estudo conduzido por Sandino et al. (2020). Neste trabalho, o *Simultaneous Localization and Mapping* (SLAM) foi empregado para mapear o ambiente, fornecendo dados ao método POMCP (*Partially Observable Monte Carlo Planning*), combinado com o ABT (*Argumented Belief Trees*) para o planejamento de trajetórias. No entanto, o autor reconhece as limitações inerentes ao sistema como um todo em lidar eficazmente com obstáculos dinâmicos. Embora existam propostas emergentes de SLAM, como aquelas que visam tratar ambientes dinâmicos (YUAN et al., 2023), ainda não representam ferramentas completamente adequadas para solucionar o problema específico de desvio de obstáculos dinâmicos.

O exemplo referenciado na Seção 2.2.1, apresentado por Lu et al. (2021), exemplifica a integração do método de amostragem do ambiente com um algoritmo de planejamento de tra-

jetória. Neste cenário específico, a abordagem une o APF com o RRT. Essa integração envolve a utilização do APF para verificar se o estado subsequente resultará em uma colisão. No caso mencionado, caso seja identificada uma colisão, o algoritmo recalcula a trajetória a partir do ponto afetado. Essa combinação visa aprimorar a eficácia do planejamento de trajetória, integrando a capacidade de detecção de obstáculos do APF com a exploração dinâmica de caminhos proporcionada pelo RRT.

Outra abordagem previamente discutida na Seção 2.2.1, que ressalta a relevância da amostragem do ambiente e que apresentou resultados positivos, foi proposta por Lin, Zhu e Alonso-Mora (2020). Neste contexto, o U-map (OLEYNIKOVA; HONEGGER; POLLEFEYS, 2015) foi empregado para detectar obstáculos, utilizando esses dados para traçar trajetórias que seriam posteriormente utilizadas para alimentar um CCNMPC (ZHU; ALONSO-MORA, 2019).

O U-map em questão realiza um histograma com o mapa de profundidade obtido por uma câmera de profundidade, identificando os elementos cujos valores ultrapassam um limiar pre-determinado. Posteriormente, aglutina esses elementos detectados para determinar as bordas horizontais do obstáculo. No caso específico desse trabalho, a estimativa da altura do obstáculo é realizada pelo número de colunas no histograma. Essa abordagem ilustra como a escolha do método de amostragem do ambiente pode contribuir significativamente para a eficácia do planejamento de trajetória em ambientes dinâmicos.

Apesar dos resultados de velocidade de processamento e distância mantida dos obstáculos da solução proposta por Lin, Zhu e Alonso-Mora (2020), uma limitação observada era a presença de uma estimativa de altura imprecisa. Essa inconsistência foi notada durante os testes do método, embora não tenha acarretado problemas significativos para a solução, dado que o UAV criava trajetórias utilizando apenas alterações no ângulo yaw, realizando seu deslocamento exclusivamente na horizontal. Para abordar essa imprecisão de medição, Saha et al. (2022) introduziu uma melhoria utilizando o U-map em conjunto com o V-map, que é um procedimento similar ao U-map mas enquanto o U-map é executado no eixo X da imagem o V-map é executado no eixo Y da imagem, o que melhora a definição do obstáculos uma vez que a definição no eixo X não se dá por meio de uma estimativa do valor do U-map. Esse aprimoramento assegura uma medição mais precisa da altura do obstáculo, permitindo que o UAV execute desvios verticais dos obstáculos.

O método do U-map e V-map também foi empregado por Yuan et al. (2023), que combinou a detecção de obstáculos com um método de *residual flow estimation* para identificar obstáculos dinâmicos. Esse recurso tornou-se necessário para filtrar o deslocamento da câmera, pois todos os testes foram realizados com a câmera em movimento. Sendo assim, era imperativo distin-

guir entre o movimento causado pelo deslocamento da câmera e o deslocamento dos próprios obstáculos. Os resultados positivos obtidos nesse trabalho corroboram a eficácia da abordagem U-map e V-map na definição precisa de obstáculos.

2.3 Considerações finais

Considerando os métodos empregados para lidar com obstáculos dinâmicos, é viável categorizá-los em duas abordagens distintas. Uma delas se utiliza de inteligência artificial (WANG et al., 2020; LOQUERCIO et al., 2019; DAI et al., 2020; SADEGHI; LEVINE, 2016), enquanto a outra faz uso de câmeras de profundidade e outros sensores para a amostragem e desvios de obstáculos (QIAN et al., 2015; YAO; ZHAO, 2015; CHEN; ZHAO; YIN, 2020; LU et al., 2021; ZAMMIT; KAMPEN, 2018; SANDINO et al., 2020; LU et al., 2021; LIN; ZHU; ALONSO-MORA, 2020).

Embora os métodos baseados em inteligência artificial demonstrem capacidade de desviar de obstáculos, é crucial salientar que essas abordagens apresentam custos computacionais elevados e demandam conjuntos de dados substanciais para o treinamento de redes. Diante desse cenário, optou-se por combinar o método proposto por Saha et al. (2022) para a identificação de obstáculos dinâmicos. Essa combinação envolve, ainda, a implementação do método proposto por Lin, Zhu e Alonso-Mora (2020), que utiliza o EKF para a identificação e projeção do deslocamento do obstáculo dinâmico. Posteriormente, emprega-se o recurso de controle de trajetórias do MRS (BACA et al., 2018), responsável por gerar trajetórias evasivas em relação aos obstáculos. Essas trajetórias são então comunicadas a um *Model Predictive Controller* (MPC).

Essa estratégia integrada proporciona uma abordagem robusta para lidar com obstáculos dinâmicos, superando as limitações computacionais inerentes aos métodos puramente baseados em inteligência artificial.

Capítulo 3

MÉTODO PROPOSTO

Neste capítulo, serão apresentados os algoritmos desenvolvidos para atender os objetivos propostos no projeto. O capítulo está dividido em quatro seções, nas quais a Seção 3.1 demonstra a identificação dos obstáculos dinâmicos. Na Seção 3.2, são abordadas as operações geométricas utilizadas para determinar a posição espacial dos obstáculos em relação a um eixo de coordenadas de referência em cada *frame*. A Seção 3.3 explora como os obstáculos definidos anteriormente têm seu estado estabelecido, incluindo o vetor de velocidade, que é crucial para projetar suas trajetórias. Na Seção 3.4, será detalhado como as trajetórias dos obstáculos dinâmicos são consideradas para a geração de uma trajetória para o UAV e como essas trajetórias são alimentadas no MPC do MRS e na Seção 3.5 os recursos utilizados.

3.1 Detecção de Obstáculos

Primeiramente, o algoritmo de planejamento de trajetória necessita de dados do ambiente, no caso se há obstáculos dinâmicos. Neste contexto, optou-se por empregar um algoritmo de U-map e V-map, conforme descrito por Saha et al. (2022) dada a eficácia que o U-map e a melhoria na definição dos obstáculos como explicado na 2.2.2, um método similar, observado em trabalho de Lin, Zhu e Alonso-Mora (2020), e que foram bem integrados para o uso de UAVs autônomos. Este método envolve a execução do processo denominado U-map em um mapa de profundidade para segmentar a largura dos objetos. Em seguida, a imagem é segmentada para de acordo com a largura dos objetos para determinar a altura dos obstáculos com o V-map. O funcionamento do algoritmo de detecção de obstáculos é ilustrado no diagrama apresentado na Figura 3.1. Essas informações são fundamentais para a proposta de desvio de obstáculos dinâmicos, delineada por Saha et al. (2022).

Tanto o U-Map quanto o V-map envolvem o cálculo do histograma normalizado do mapa

FIGURA 3.1 – Diagrama da Detecção de Obstáculos

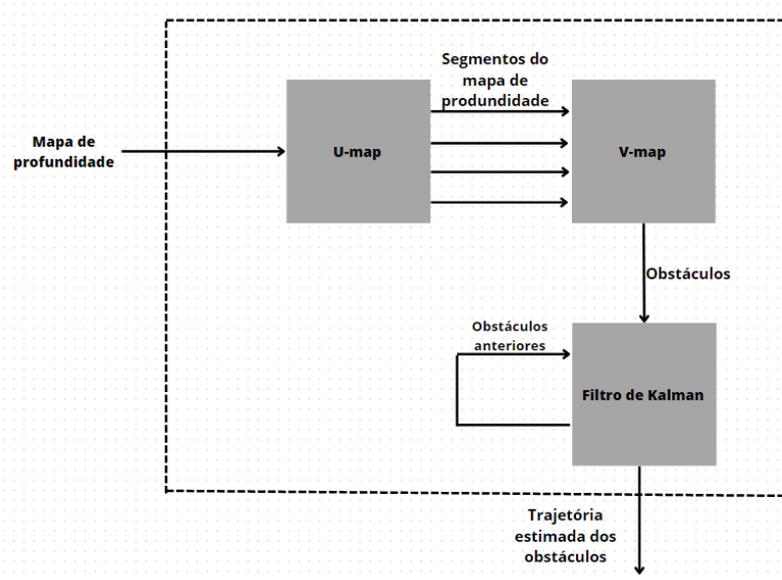


FIGURA 3.2 – Diagrama da Detecção de Obstáculos

de profundidade (OLEYNIKOVA; HONEGGER; POLLEFEYS, 2015), diferindo apenas na orientação do histograma. Enquanto o U-map cria um histograma no eixo vertical, o V-map o faz na horizontal. O algoritmo inicia reduzindo a escala da imagem, um procedimento que diminui o número de operações necessárias para processar um único frame sem comprometer substancialmente a precisão global do processo. Em seguida, é gerado o histograma para os elementos que passam do threshold de cada coluna do U-map que, em seguida as colunas são nomeadas de acordo com seus valores e em seguida as colunas que são adjacentes e com valores próximos são aglutinadas, definindo assim os limites do obstáculo no eixo X, e após esse procedimento, a imagem é cortada para conter apenas os segmentos com obstáculos detectados. Posteriormente, esse segmento da imagem é processado pelo V-map, permitindo assim a determinação da altura do obstáculo. Os algoritmos específicos para U-map e V-map são descritos nos Algoritmos 3.1 e 3.2.

Os obstáculos são detectados como demonstrado no Algoritmo 3.4 do U-map, são usados para definir os obstáculos como demonstrado no Algoritmo 3.3 em seguida um algoritmo análogo é feito com o V-map para determinar a altura do obstáculo.

Algoritmo 3.1: *U-map*

```

1 depth-map-rescaled = depth-map em escala reduzida
2 N-cols = comprimento do depth-map-rescaled
3 N-rows = altura do depth-map-rescaled
4 histSize = depth-map.altura/downsample - scale
5 bin_width = ceil((Dist_Max - Dist_Min)/histSize)
6 for col = 0, 1, 2, ..., N - cols do
7   for row = 0, 1, 2, ..., N - rows do
8     if depth_map[row][col] > Dist_Min & depth_map[row][col] < Dist_Max then
9       bin_index = (depth_map[row][col] - Dist_Min)/bin_width
10      if U_map[bin_index, col] < 255 then
11        Aumenta U_map[bin_index, col] em 1
12      end
13    end
14  end
15 end

```

Algoritmo 3.2: *V-map*

```

1 depth-map-section = segmento da imagem com obstaculo detectado pelo U-map
2 depth-map-rescaled = depth-map-section em escala reduzida
3 N-cols = comprimento do depth-map-rescaled
4 N-rows = altura do depth-map-rescaled
5 histSize = depth_map.size/downsample - scale
6 bin_width = ceil((Dist_Max - Dist_Min)/histSize)
7 for row = 0, 1, 2, ..., N - rows do
8   for col = 0, 1, 2, ..., N - cols do
9     if depth_map[row][col] > Dist_Min & depth_map[row][col] < Dist_Max then
10      bin_index = (depth_map[row][col] - Dist_Min)/bin_width
11      if V_map[row, bin_index] < 255 then
12        Aumenta V_map[row, bin_index] em 1
13      end
14    end
15  end
16 end

```

Algoritmo 3.3: *Definição dos obstáculos*

```

1 sum_line = 0
2 max_line = 0
3 length_line = 0
4 for row = 0, 1, 2, ..., U_map.rows do
5     for col = 0, 1, 2, ..., U_map.cols do
6         if U_map[row, col] >= Threshold then
7             length_line = length_line + 1
8             sum_line = sum_line + U_map[row, col]
9             if U_map[row, col] > max_line then
10                max_line = U_map[row, col]
11            end
12        end
13        if U_map[row, col] < Threshold || col == (U_map.cols - 1) then
14            if length_line > len_threshold & sum_line > len_threshold * max_line
15                then
16                    Dá Labels às colunas de cada obstáculo do histograma e lista em
17                    UV-boxes
18                end
19            end
20        end
21    end
22    for b = 0, 1, 2, ... N de elementos de UV_boxes do
23        if UV_boxes[b] é o maior id das labels then
24            Identifica se as UV_boxes[b+n] tem a mesma Label
25            Anexa as UV_boxes[b+n] à UV_boxes[b] se sim
26        end
27        if Tamanho de UV_boxes[b] > Threshold then
28            Lista a UV_boxes.b na lista de Obstáculos
29        end
30    end

```

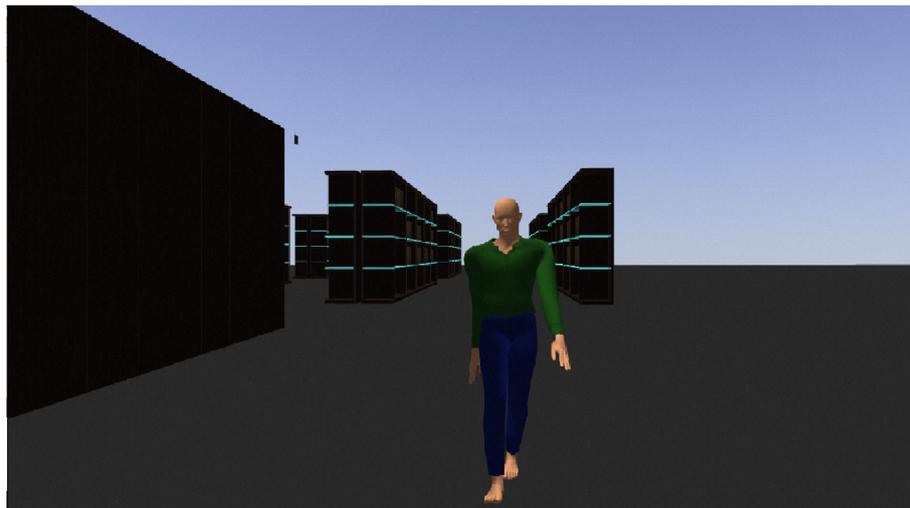
Algoritmo 3.4: *Detecção de obstáculos*

```
1 Executa Algoritmo 3.3
2 Definição dos obstáculos no Eixo X e lista em UV_boxes
3 for  $b = 0, 1, 2, \dots$  Comprimento de UV_boxes do
4   |   Calcula o algoritmo 3.2
5   |   Definição dos obstáculos no Eixo Y
6 end
```

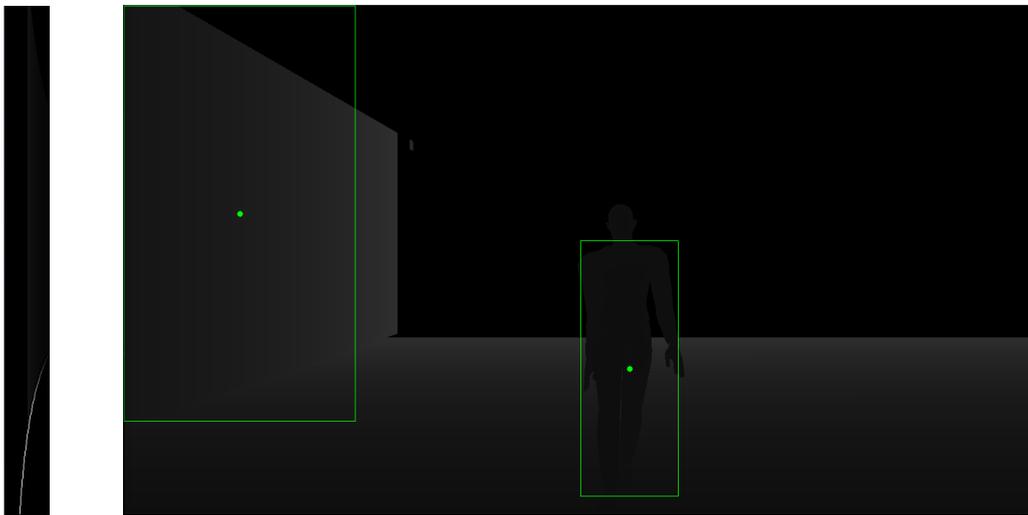
Os histogramas resultantes são representados em uma matriz que, uma vez definida, requer a consolidação das colunas que indicam o mesmo obstáculo. Esse processo consiste em percorrer coluna por coluna, comparando os valores das colunas adjacentes para determinar se são semelhantes, ou seja, se a diferença entre eles é menor que um valor limite. Caso sejam similares, as colunas são agrupadas e identificadas como parte de um obstáculo. Uma vez que todas as colunas destacadas são agrupadas, elas precisam ser reescaladas para indicar quais valores das coordenadas da imagem elas representam.

Por exemplo, ao executar o U-map, o histograma da imagem é gerado ao longo do eixo X da imagem. Nesse caso, um mapa de profundidade com dimensões 1280x720 pixels, quando reescalado pela metade, resulta em um mapa de dimensões 1280x360 pixels. Esse redimensionamento é essencial para que as informações das colunas agrupadas possam ser correlacionadas corretamente com as coordenadas da imagem original. Esse processo de agrupamento e reescalonamento é fundamental para obter uma representação precisa dos obstáculos detectados.

O resultado desse algoritmo é exemplificado na Figura 3.3, que demonstra uma montagem com a imagem RGB recebida em Fig. 3.3 (a), abaixo dela existe o mapa de profundidade da mesma imagem com os obstáculos já contornados em verde em, Fig. 3.3 (b), abaixo desse mapa é demonstrado o U-map que determina a dimensão dos obstáculos no eixo X em Fig. 3.3 (d) e na esquerda do mapa de profundidade é representado o V-map que corresponde ao segmento da imagem que contém a parede em Fig. 3.3 (c).



(a) Imagem em RGB



(c) V-map

(b) Mapa de profundidade



(d) U-map

FIGURA 3.3 – Exemplo do resultado da medição de U-map e V-map em simulação

3.2 Transformação das Coordenadas de Referências

Após realizar essa operação com o mapa de profundidade, torna-se possível estimar a posição do objeto no espaço. Neste caso, optou-se por utilizar o centróide da imagem para determinar a posição relativa do obstáculo em relação ao UAV. Para realizar essa estimativa, foram empregadas as equações de projeção da câmera, representadas pelas Equações 3.1, 3.2 e 3.3.

$$Dx = \frac{(x - W_x)D}{K_x} \quad (3.1)$$

$$Dy = D \quad (3.2)$$

$$Dz = \frac{(y - H_y)D}{K_y} \quad (3.3)$$

onde x e y são as coordenadas do centroide do obstáculo, W_x e H_y representam, respectivamente, metade da largura e altura do mapa de profundidade. D é a medição da profundidade, e K_x e K_y são os parâmetros intrínsecos da câmera.

Essas equações permitem estimar as coordenadas tridimensionais do obstáculo no espaço em relação ao UAV, proporcionando informações cruciais para o planejamento de trajetória e desvio de obstáculos.

No entanto, os resultados que se obtém até o ponto onde se calcula a distância real obtidos até agora são incompletos para a detecção de colisões, pois não proporcionam informações conclusivas sobre os obstáculos, exceto suas dimensões e a posição relativa ao UAV. Além disso, não permitem determinar o movimento do obstáculo, uma vez que qualquer movimento do UAV altera a posição relativa do obstáculo. Esse problema foi solucionado por meio da transformação linear dessa posição relativa em relação ao UAV para a posição relativa a um eixo de coordenadas de referência estático.

Essa operação envolve a rotação da posição relativa dos obstáculos de modo que os eixos de coordenadas da posição relativa se alinhem paralelamente ao eixo de coordenadas de referência. Além da operação de rotação em relação ao eixo de coordenadas, também foi necessário fixar as posições em relação ao eixo de coordenadas de referência. As equações que descrevem essas operações estão representadas na Equação 3.4, onde $[x, y, z]_{real}$ é a coordenada do obstáculo em relação ao eixo de coordenadas de referência, R_{UAV} é a matriz de rotação com os ângulos de *pitch*, *yaw* e *roll* da direção do UAV, $[x, y, z]_{UAV}$ são as coordenadas de um obstáculo detectado pelo UAV, e $[x, y, z]_{UAV}$ são as coordenadas x , y e z do UAV.

$$[x, y, z]_{real} = R_{UAV} * [x, y, z]_{camera} + [x, y, z]_{UAV} \quad (3.4)$$

Essas operações de transformação proporcionam informações mais objetivas sobre a posição e movimento dos obstáculos, pois padroniza um eixo de referência estático, facilitando a tomada de decisões no planejamento de trajetória e desvio de obstáculos.

3.3 Estimativa de Velocidade

Após a determinação das posições dos obstáculos, torna-se essencial distinguir entre os obstáculos móveis e estáticos. Para essa tarefa, utilizou-se o *Extended Kalman Filter* (EKF), o mesmo empregado no algoritmo proposto e disponibilizado por Oleynikova, Honegger e Pollefeys (2015) e Jafari, Mitzel e Leibe (2014). O EKF é uma extensão do Filtro de Kalman, sendo um filtro recursivo não linear Bayesiano (ARULAMPALAM et al., 2002), projetado para lidar com funções não lineares. Isso o torna ideal para a aplicação em questão, uma vez que, como ilustrado na Figura 3.3, mesmo ao considerar um obstáculo em movimento retilíneo, o método de medição pode sofrer variações na detecção do obstáculo devido a outros movimentos. Nesse exemplo, os próprios movimentos dos braços e pernas do obstáculo causam oscilações na detecção, e no caso o EKF lineariza o problema localmente para estimar a velocidade.

O algoritmo como um todo realiza uma operação que compara os obstáculos detectados entre dois frames consecutivos e optou-se por aplicar o EKF somente a duas coordenadas, ignorando a coordenada vertical. Essa decisão foi tomada porque a câmera, em sua maioria, está direcionada para as coordenadas paralelas ao solo e embora haja uma pequena oscilação para cima ou para baixo durante o deslocamento do UAV, de maneira geral, pode-se assumir que o UAV coleta dados principalmente sobre os eixos x e y . Portanto, a coordenada z foi ignorada na projeção futura, dessa maneira a detecção dos obstáculos é realizado nos eixos x , y e z , mas somente se considera o deslocamento nos eixos x e y , isso porque seria pouco útil e uma potencial fonte de erros na detecção de movimento.

O EKF utiliza duas medições do mesmo obstáculo para estimar o estado do mesmo. Logo, para realizar essa operação, foi necessário determinar quais são os dois pontos a serem comparados dado que cada frame processado pelo U-map e V-map resulta na detecção avulsa dos obstáculos, sem nenhuma especificação objetiva quais desses obstáculos correspondem com os obstáculos detectados no frame anterior. A solução para esse problema consistiu em comparar cada obstáculo do detectado no U-map do frame atual com todos os obstáculos do frame anterior. Os obstáculos que apresentam a maior área de interseção nos eixos X e Y são definidos como o mesmo obstáculo.

Essa estratégia parte da premissa de que o número de frames por segundo da câmera será elevado, e os obstáculos não terão velocidade suficiente entre um frame e outro para se deslocarem significativamente de posição. Essa abordagem visa garantir a correspondência adequada entre os obstáculos nos frames consecutivos, possibilitando a aplicação eficaz do EKF para a estimativa do estado dos obstáculos móveis.

O EKF utiliza uma função de probabilidade para estimar o estado do obstáculo (ESS et al., 2010). Os parâmetros empregados no EKF são apresentados na Equação 3.5, onde A é a matriz de estado, ts é a variação de tempo, B é a matriz de injeção, H é a matriz de observação, P é a matriz de incerteza, Q representa o ruído do processo, e R representa o ruído de observação. Os valores dos parâmetros nos elementos descritos estão dispostos na Equação 3.5.

$$\begin{aligned}
 A &= \begin{bmatrix} 1 & 0 & ts & 0 & 0 & 0 \\ 0 & 1 & 0 & ts & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 H &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & P &= \begin{bmatrix} 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.99 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 \end{bmatrix} & (3.5) \\
 Q &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & R &= \begin{bmatrix} 0.3 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix}
 \end{aligned}$$

Dada as que a estimativa de uma obstáculo é dado em z como um vetor representado como mostrado na equação 3.6

$$z = \begin{bmatrix} \text{coordenada X do centro} \\ \text{coordenada Y do centro} \\ \text{Largura do Obstáculo} \\ \text{Altura do Obstáculo} \end{bmatrix} u = \begin{bmatrix} 1 \end{bmatrix} \quad (3.6)$$

Cada vez que o algoritmo de detecção de obstáculos realiza sua operação e detecta o mesmo obstáculo, o EKF realiza as equações 3.7 e 3.8 para estimar a velocidade do obstáculo:

$$\begin{aligned} State &= A.State + B.u \\ P &= A.P.A^T + Q.u \end{aligned} \quad (3.7)$$

$$\begin{aligned} S &= R + H.P.H^T \\ K &= P.H^T.S^{-1} \\ State &= State + K.(z - H.State) \end{aligned} \quad (3.8)$$

Uma vez determinada a velocidade do obstáculo, realiza-se a projeção das posições por onde ele deverá passar. Para isso, a velocidade vetorial é normalizada, e a partir do centro do obstáculo, são gerados pontos ao longo dessa direção. Dessa forma, é definida uma linha reta de pontos que é considerada como a trajetória do obstáculo. Esse resultado é então alimentado para o MPC, contribuindo para a tomada de decisões do controle da trajetória do UAV. Essa abordagem visa antecipar e reagir adequadamente ao movimento dos obstáculos, permitindo que o UAV ajuste sua trajetória de maneira preditiva.

3.4 Correção de trajetórias e MPC

O controlador adotado no projeto foi o MPC, conforme proposto por Baca et al. (2021) entre alguns métodos de controles disponibilizados, mais especificamente o controlador SE(3) (*Special Euclidean Group*) (LEE; LEOK; MCCLAMROCH, 2010) e o MPC (PETRLÍK et al., 2020). No caso o MRS integra as etapas de um observador de estados, identificação de colisão, ajuste de trajetória e o controlador selecionado. Neste trabalho, foi incorporada uma entrada adicional a esse mecanismo, conforme ilustrado na Figura 3.4.

Esse processo complexo permite a coordenação e controle eficazes do UAV, garantindo que a trajetória seja adaptada em tempo real para evitar colisões com obstáculos dinâmicos.

No estudo original conduzido por Baca et al. (2018), foi desenvolvido um sistema de rastreamento e desvio de obstáculos integrado a um MPC para múltiplos UAVs, com foco em ambientes externos. O algoritmo proposto inicialmente presumia a comunicação entre os UAVs para troca de coordenadas e trajetórias, a fim de identificar possíveis conflitos de trajetória. Neste estudo, essa abordagem é substituída pelo uso das trajetórias detectadas pelo U-map e V-map, combinadas com o EKF para fornecer informações sobre os obstáculos, em vez das trajetórias de outros UAVs.

Com base em um conjunto de trajetórias, é necessário resolver os conflitos entre elas, tarefa gerenciada pelo *MPC Tracker*. Neste caso, o método utilizado consiste em verificar a distância

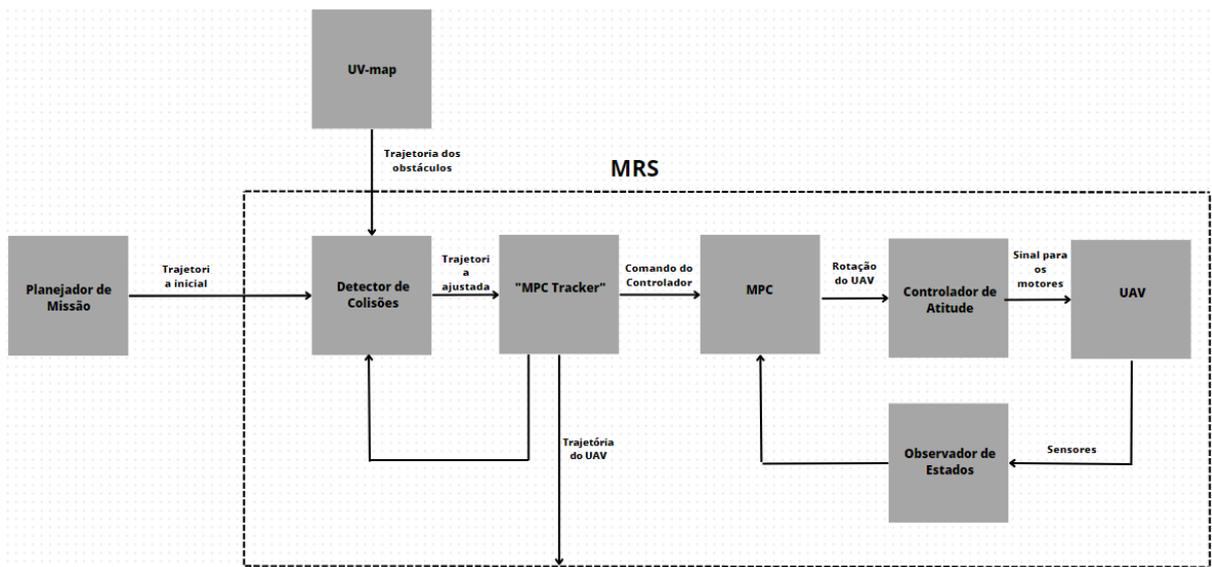


FIGURA 3.4 – Diagrama de funcionamento do MPC

entre todos os pontos das trajetórias. Se pontos a uma distância inferior a 1 metro forem encontrados, no caso a distância de 1 metro foi determinada de maneira arbitrária, mas considerando os limites que a câmera de profundidade, que consegue detectar até 10 metros de profundidade, essa foi considerado como uma margem segura e cabível dentro do espaço detectado, considerando o quanto de o algoritmo analisa a hierarquia das rotas em colisão e determina quais delas podem ser modificadas. Em seguida, é incrementado um valor de *offset* até que o UAV desvie do obstáculo, sendo que neste estudo específico o *offset* é de 2 metros. Embora o método de desvio não seja inovador, este trabalho visa demonstrar um algoritmo eficaz para resolução de conflitos de trajetória, capaz de lidar com múltiplos obstáculos.

Apenas um UAV foi utilizado neste estudo em particular, portanto, era a única trajetória sujeita a ajustes. Após a resolução de todos os conflitos identificados entre as trajetórias envolvendo o UAV, a nova trajetória ajustada é enviada ao MPC.

3.5 Materiais

3.5.1 Recursos simulados

Nesta seção, são apresentados os materiais e recursos utilizados nos testes, abrangendo desde frameworks e bibliotecas até ambientes simulados e bibliotecas específicas para o domínio de UAVs.

- Framework *Robot Operating System* (ROS), que oferece recursos de baixo nível para o

desenvolvimento de aplicações em robôs. Link: <https://github.com/ros/ros>

- Pacote MRS em conjunto com ROS para o desenvolvimento de aplicações em UAVs.
Link: https://github.com/ctu-mrs/mrs_uav_system
- Biblioteca Plannie, adotada para algoritmos de planejamento de trajetória.
Link: <https://github.com/lidiexp/plannie>
- Ambientes para testes no simulador Gazebo. Link: <https://gazebosim.org/home>
- Biblioteca do U-map (JAFARI; MITZEL; LEIBE, 2014).
Link: https://github.com/0Jiahao/uv_detector



FIGURA 3.5 – UAV F450 - Ambiente Simulado Gazebo.

3.6 Considerações finais

Neste capítulo, foram apresentados todos os recursos desenvolvidos e utilizados para atingir os objetivos propostos neste trabalho. Esses recursos foram integrados aos já existentes, operando de maneira modular e sendo compatíveis com diversos tipos de simulações. Eles estão prontos para serem submetidos a testes que visam validar sua eficácia e desempenho em ambiente real. Essa abordagem modular e a prontidão para testes facilitam a implementação prática do sistema proposto, proporcionando flexibilidade e adaptabilidade em diferentes cenários e ambientes de simulação.

Capítulo 4

RESULTADOS

Neste capítulo, serão apresentadas as métricas e os recursos para a verificação da proposta realizada, seguindo a seguinte ordem: primeiramente, serão apresentadas as métricas que serão empregadas para a medição e comparação de resultados, bem como as definições e diferenças entre o U-map e o U-map com o V-map (que definiremos como UV-map por conveniência). Em seguida, será apresentado um ambiente que simula condições mais próximas da realidade de operação para uma avaliação inicial do método UV-map integrado ao sistema de verificação de trajetórias do MRS. Posteriormente, serão apresentados os resultados preliminares obtidos nessas condições de testes. Uma vez apresentados os testes preliminares, será realizado um teste comparativo do método, inicialmente justificando e descrevendo um novo ambiente para testes, seguido de uma série de testes com o UAV em diversas velocidades, com as seguintes configurações:

- Testes do U-map integrado ao sistema de ajuste de trajetórias do MRS;
- Testes do UV-map integrado ao sistema de ajuste de trajetórias do MRS;
- Testes que inclui o Plannie (ROCHA, 2021) atuando como sistema de planejador de missões como indicado na Figura 3.4 em conjunto do UV-map em conjunto com o MRS.

Para avaliar os resultados dos testes preliminares e comparativos, serão mensurados o SR e o AGV em testes isolados para medir o PT usando dois métodos de detecção de obstáculos: o U-map e o UV-map. Esses procedimentos visam avaliar o desempenho, a eficácia e a viabilidade do método proposto, além de sua integração em outros sistemas. É importante ressaltar, no entanto, que as expectativas em relação aos resultados devem ser consideradas com cautela, uma vez que os critérios para a avaliação de UAVs ainda não possuem um método padronizado, conforme detalhado na Seção 4.1.

4.1 Métricas

Para avaliar os resultados dos testes, foi necessário adotar métricas específicas para este trabalho. De acordo com Yu, Croon e Wagter (2023), o campo da robótica ainda não convergiu para uma definição de métricas para avaliação da performance de desvio de obstáculos em UAVs. No entanto, o autor sugere algumas métricas para a avaliação dos resultados obtidos. Essas métricas foram escolhidas com o objetivo de fornecer uma análise quantitativa e qualitativa do desempenho do método proposto, considerando as características específicas do problema abordado neste trabalho.

Dentre as métricas sugeridas, muitas delas demandariam algum tipo de condicionamento dos testes para extrair informações conclusivas. Isso se deve ao fato do ambiente ser dinâmico, tornando a avaliação altamente dependente das circunstâncias específicas de cada simulação, eventualidades como iniciar a trajetória quando o obstáculo está em momentos diferentes do percurso programado pode acarretar em circunstâncias completamente diferentes como será visto na Seção 4.6. Diante desse contexto, foram selecionadas as seguintes métricas, que têm o potencial de fornecer informações conclusivas sobre o desempenho do algoritmo. Essas métricas foram escolhidas com o intuito de oferecer uma avaliação robusta e relevante, considerando a dinâmica variável do ambiente durante as simulações.

No presente caso, foram adotadas três métricas para a avaliação, incluindo AGV, SR e PT. Essas métricas foram escolhidas para realizar uma análise comparativa entre os métodos, uma vez que representam elementos objetivos fundamentais para a discussão em um ambiente dinâmico. Outras métricas podem ter relevância em testes sob condições idênticas, no entanto, a natureza dinâmica do ambiente pode ocasionar variações nas condições da missão, gerando um resultado que não se dá necessariamente por mérito da qualidade do algoritmo, mas devido a fatores ambientais que favorecem ou desfavorecem a execução da missão e mesmo com essas considerações, ainda são feitas diversas ressalvas durante os testes como vai ser explicado futuramente na Seção 4.6. As métricas adotadas são calculadas das seguintes maneiras:

- SR: Mede o percentual de missões bem sucedidas, onde $N_{sucessos}$ é o número de missões bem sucedidas e $N_{missoes}$ é o número de missões realizadas;

$$SR = \frac{N_{sucessos}}{N_{missoes}} \quad (4.1)$$

- AGV: Mede a velocidade média para concluir a missão, onde d_{min} é a distância mínima para concluir a missão e t_{trial} é o tempo necessário para concluir a missão;

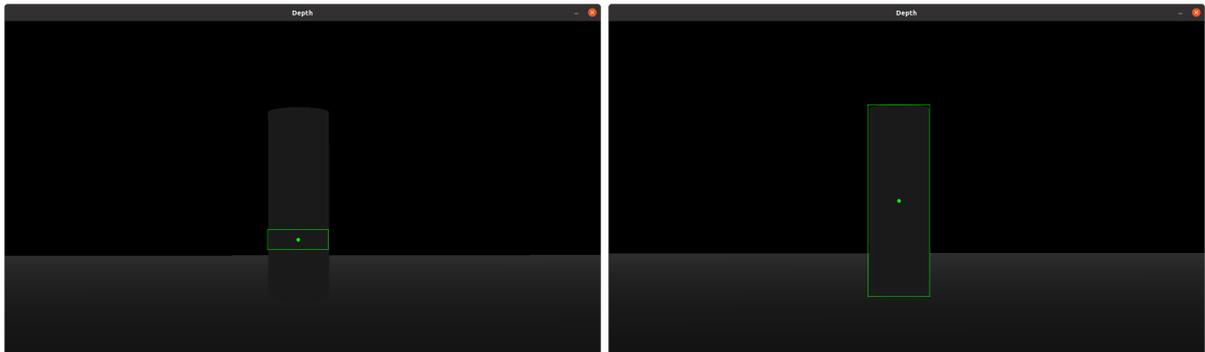
$$AGV = \frac{d_{min}}{t_{trial}} \quad (4.2)$$

- PT: Mede o tempo de processamento de uma única iteração de processamento para replanejar uma trajetória.

4.2 Diferenças entre o U-map e o UV-map

Primeiramente, dada a proposta de comparar o tempo de processamento do algoritmo, é interessante definir a diferença entre o método que usa a detecção do obstáculo usando o U-map e o que usa o U-map e V-map, apresentados na Seção 4.6. No caso do primeiro, esse método foi inicialmente usado por Oleynikova, Honegger e Pollefeys (2015), que fazia a clara distinção entre o U-map e o V-map. No entanto, para a resolução do problema proposto, ela utilizava apenas o U-map, dado que não se esperava desviar dos obstáculos verticalmente, limitando os graus de liberdade que o UAV poderia recorrer para desviar do obstáculo. Ainda utilizando as mesmas limitações de graus de liberdade de desvio, o trabalho apresentado por Zhu e Alonso-Mora (2019) também utilizou somente o U-map para identificar os obstáculos.

FIGURA 4.1 – Comparação do mesmo mapa de profundidade sob o algoritmo U-map e UV-map



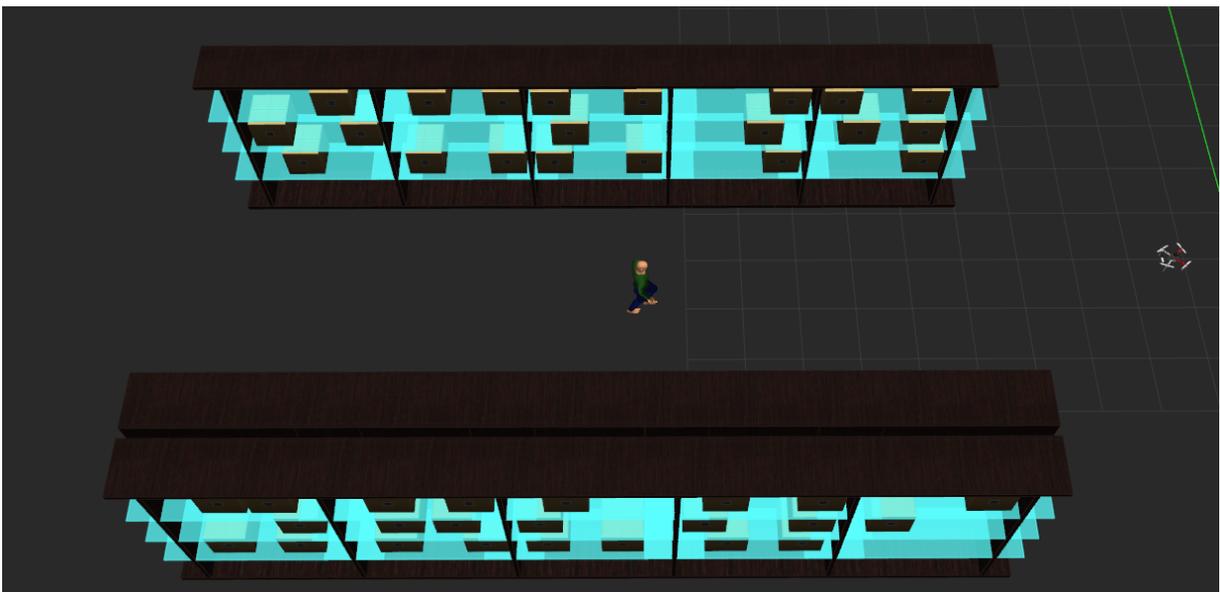
O resultado é uma visível diferença na determinação dos obstáculos, mesmo com uma reconstrução proporcional do U-map não oferece uma definição de altura precisa para o obstáculo.

4.3 Ambiente de testes preliminares do método proposto

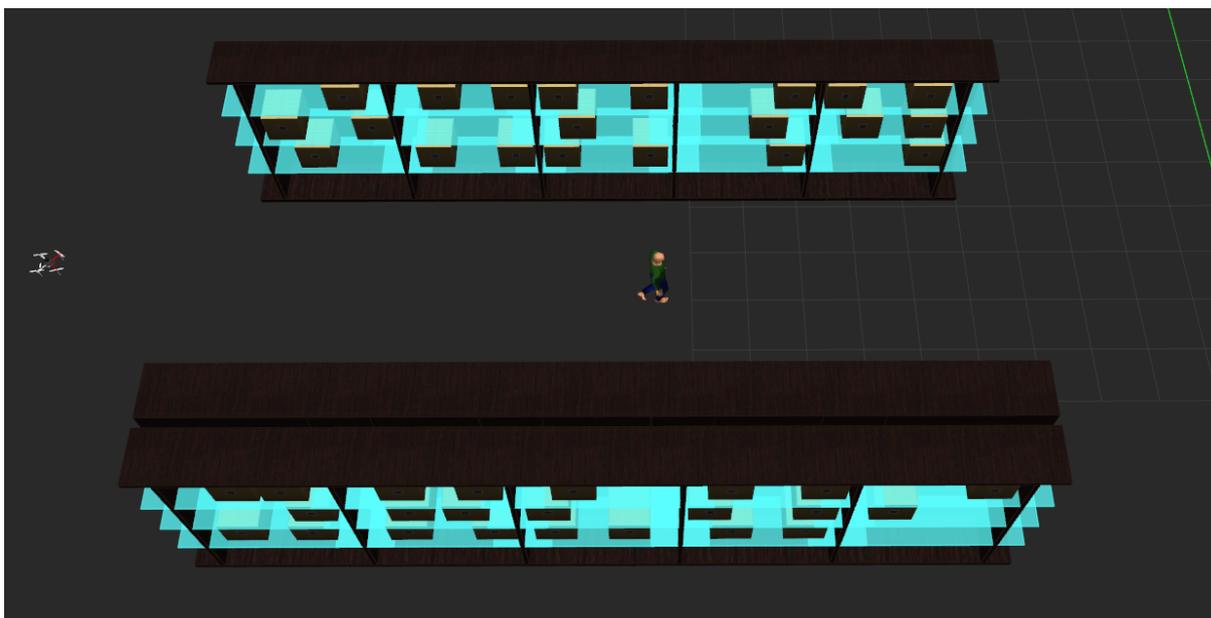
O primeiro teste realizado para avaliar o funcionamento global do algoritmo e uma hipótese de ambiente para teste com obstáculo dinâmico. O desvio de obstáculos do MRS foi testado

com e sem o uso do UV-map em condições semelhantes, detalhe o qual é importante ressaltar, que é que o UAV é submetido a condições similares, mas dada a natureza dinâmica do ambiente, qualquer alteração no início do teste pode ocasionar numa mudança das condições que o UAV terá de lidar, de maneira que a única circunstância garantida nesse teste é que dado o ponto de partida, enquanto o UAV estiver virado em direção ao objetivo e dentro das limitações do corredor, ele ao menos terá o obstáculo em vista em algum momento mesmo se o obstáculo não estiver em rota de colisão com o UAV. O ambiente de teste consistiu em um corredor de armazém com um obstáculo dinâmico e diversos estáticos, onde uma pessoa se deslocava pelo corredor, indo e voltando a uma velocidade média de 1,5 metros por segundo. As prateleiras do corredor apresentavam caixas dispostas de maneira randômica, criando uma superfície irregular. Esse cenário foi escolhido para simular uma situação realista e desafiadora, permitindo uma análise abrangente do desempenho do algoritmo em condições dinâmicas e com obstáculos complexos.

FIGURA 4.2 – Configuração no início da simulação da simulação



O UAV foi designado para percorrer o corredor entre as coordenadas (x,y,z) inicial e final, respectivamente $(2m, 8m, 1.5m)$ e $(21m, 8m, 1.5m)$, a uma velocidade de 2 m/s , conforme representado nas Figuras 4.2 e 4.3, considerando evitar a colisão com obstáculos. Desta forma, neste trajeto, embora o caminho mais curto para o UAV fosse uma linha reta de 19 metros até o ponto de chegada, seguir essa rota resultaria em uma colisão com o obstáculo dinâmico. Os obstáculos estáticos, por sua vez, não estão localizados na rota direta, mas o algoritmo empregado precisa detectá-los corretamente, caso contrário, poderiam ocorrer desvios desnecessários na trajetória.

FIGURA 4.3 – Configuração no fim da simulação da simulação

Essa configuração de teste foi estabelecida para desafiar o UAV a navegar por um corredor com obstáculos desconhecidos e requerer a ativação do algoritmo de desvio de obstáculos para garantir uma passagem segura. Isso ocorre porque, como mencionado anteriormente, a rota mais óbvia de seguir em linha reta até o objetivo resultaria em uma colisão com o obstáculo, independentemente do tempo de processamento do algoritmo e do início da execução da missão. Embora este ambiente de teste possa parecer inicialmente limitado, ele foi projetado de forma a permitir que as paredes atuem como obstáculos que não interferem na trajetória e, mais importante ainda, para simular colisões com os obstáculos dinâmicos, eventos estes que são, na verdade, raros, pois a colisão entre corpos em movimento é uma coincidência improvável.

4.4 Testes sem o UV-Map

A princípio, foram realizados testes com o modelo MPC do pacote MRS sem nenhuma implementação adicional. O propósito dessas simulações foi testar a capacidade de concluir a missão proposta com os recursos existentes disponibilizados pelo pacote MRS sozinho. A princípio, não há nenhum motivo para o UAV tomar medidas preventivas contra o obstáculo, dado que, embora existam recursos que permitam o ajuste da trajetória (BACA et al., 2018) para desviar o obstáculo, o mesmo é desprovido de conhecimento prévio dos obstáculos. Logo esse teste visa garantir que qualquer resultado que demonstre sucesso no desvio do obstáculo se dá pelas alterações feitas nesse trabalho. Portanto, este teste visa ga-

rantir que qualquer resultado que demonstre sucesso no desvio do obstáculo seja atribuído às alterações realizadas neste trabalho. No caso em questão, o UAV respondeu como esperado, porém falhou em todas as tentativas de missão de percorrer o corredor enquanto voava a $2m/s$. Como resultado, não foi possível obter valores de AGV para comparação, já que o UAV não conseguiu completar com sucesso as missões propostas. Esses resultados iniciais destacaram a necessidade de implementar melhorias para garantir um desempenho adequado na tarefa de desvio de obstáculos dinâmicos.

4.5 MPC Auxiliado do U-map e V-map

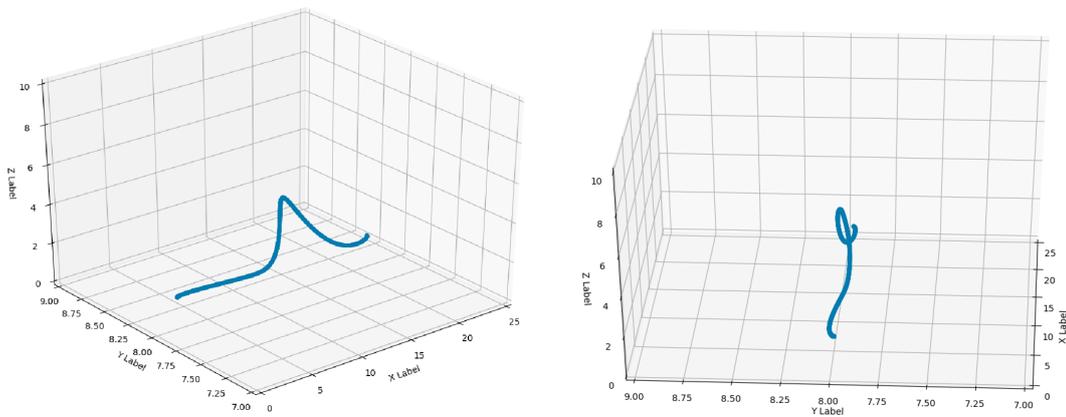
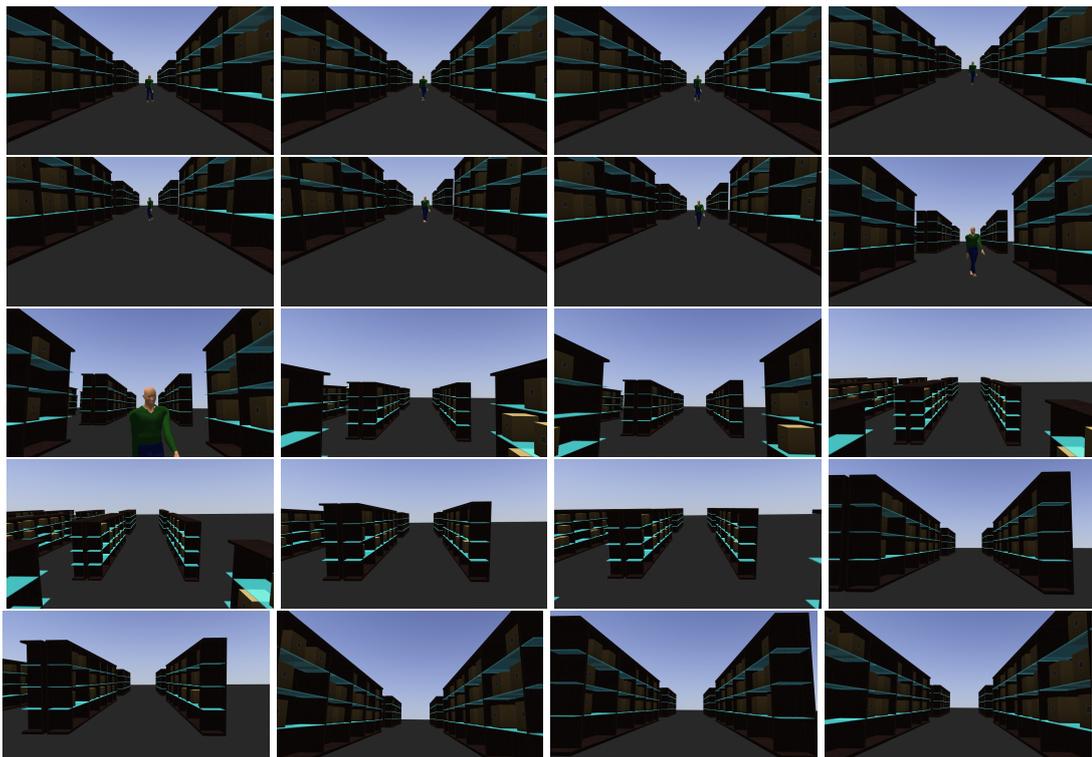
Subsequentemente aos testes realizados na Seção 4.4, foram conduzidos cinco testes com o método proposto neste trabalho com o UAV também a uma velocidade de $2 m/s$, utilizando a combinação do método de detecção de obstáculos empregando o UV-map com o MPC. Durante esses testes, o método de desvio de obstáculos demonstrou sucesso em quase todas as tentativas, resultando em $SR = 0.8$, ou seja, conseguindo completar a missão em quatro das cinco vezes. Além disso, os valores da AGV foram calculados conforme a Equação 4.2, resultando nos valores apresentados na Tabela 4.1.

Tabela 4.1: Resultados de AGV por simulação dos testes preliminares.

Teste	AGV
1	- m/s
2	$1,88 m/s$
3	$1,82 m/s$
4	$1,96 m/s$
5	$1,94 m/s$

Dos testes realizados, conclui-se que não apenas a combinação do UV-map com o sistema de verificação de trajetórias do MRS são métodos viáveis para o desvio de obstáculos dinâmicos, mas também que o ambiente em si requer que o UAV seja capaz de desviar desses obstáculos dinâmicos.

Um exemplo de uma trajetória bem-sucedida pode ser observado na Figura 4.4, enquanto uma captura da gravação feita durante o voo está representada na Figura 4.5.

FIGURA 4.4 – Trajetória de voo bem sucedido a 2 m/s.**FIGURA 4.5 – Exemplo de voo bem sucedido dos testes preliminares a 2 m/s. Link para o video das imagens: <https://youtu.be/uJgZj4NwSmY>**

4.6 Análise comparativa

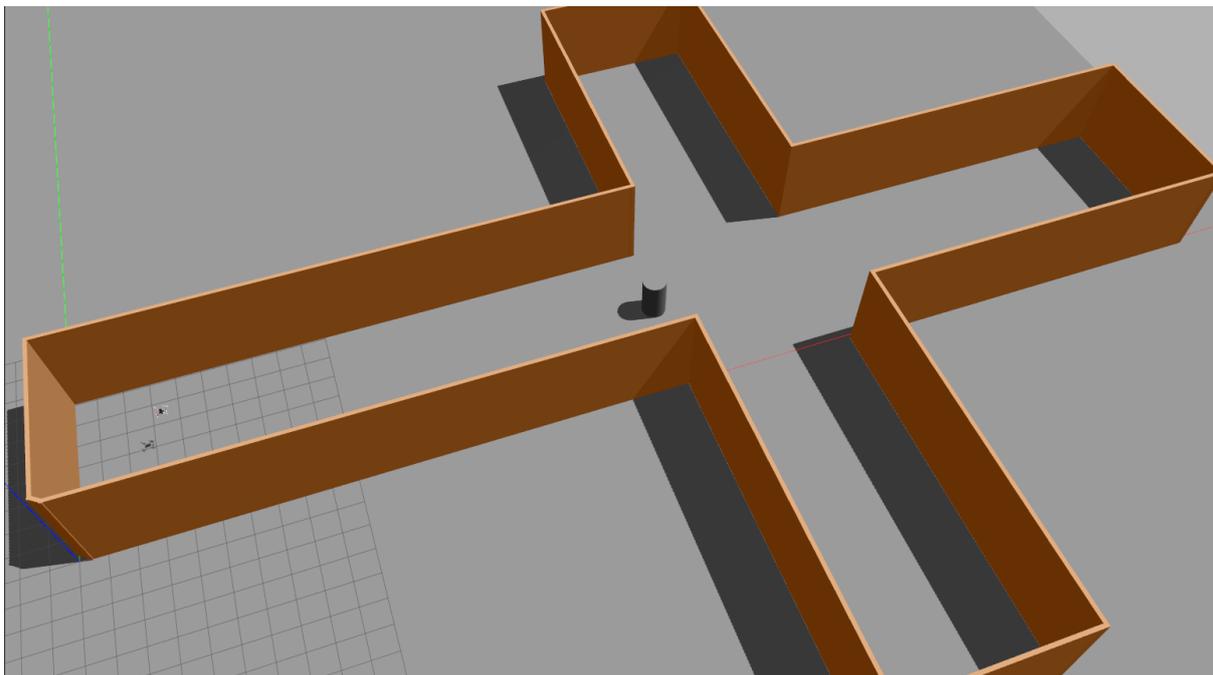
Como parte da validação do método proposto, foram realizados testes do algoritmo em um novo ambiente, visando a comparação com outros recursos de sensores disponíveis que não

podem ser explorados com os recursos no ambiente inicial, mais especificamente os recursos de LiDARS e restrições de altitude de voo que demandaram a introdução de um obstáculo mais alto e com configurações dos obstáculos da simulação que permitiam a detecção por LiDAR. Além do UV-map em conjunto com o sistema MRS, foram examinadas configurações que envolvem o U-map com o sistema de verificação de trajetórias do MRS e o UV-map com o sistema MRS, este último recebendo comandos de um sistema de *path planning*, o Plannie (ROCHA, 2021). Os testes foram conduzidos em um ambiente semelhante ao descrito na Seção 4.3 para garantir as mesmas condições especificadas de colisões, porém com modificações para incluir características físicas adicionais ao obstáculo. Para fins comparativos, cada algoritmo foi submetido à mesma missão em cinco repetições, simuladas com o UAV em diferentes velocidades, a saber, 1 *m/s*, 2 *m/s* e 3 *m/s*.

Esta Seção apresentará inicialmente o ambiente de testes, seguido da descrição de como os resultados de PT foram obtidos e os valores calculados. Por último, serão apresentados os resultados de AGV e SR obtidos com diferentes combinações de algoritmos: U-map com o algoritmo de correção de trajetórias do MRS, UV-map com o algoritmo de correção de trajetórias do MRS, e o UV-map com o algoritmo de correção de trajetórias do MRS, juntamente com o Plannie, um algoritmo de planejamento de missões.

4.6.1 Ambiente para análise comparativa

O ambiente de testes para análise comparativa foi construído como mostrado na Figura 4.6, com a observação de que as paredes da imagem foram diminuídas nessa foto para expor melhor o interior do ambiente.

FIGURA 4.6 – Ambiente de testes para comparação de métodos de desvio

No caso, o ambiente de testes possui paredes de 6 metros de altura que formam um cruzamento entre dois corredores com 8 metros de largura e um obstáculo dinâmico é um cilindro de 2.5 metros de altura e 1 metro de diâmetro, que se desloca entre as coordenadas $(8\text{ m}, 4\text{ m})$ e $(20\text{ m}, 4\text{ m})$, a uma velocidade constante de 1.5 m/s .

4.6.2 Testes de PT

Uma vez que a natureza do algoritmo de desvio, que busca ativamente por colisões constantemente e se move caso haja algum obstáculo dinâmico vindo em sua direção, foi optado por realizar esse teste com o UAV parado na posição $(4\text{ m}, 4\text{ m}, 2\text{ m})$ e tendo de identificar e desviar de um obstáculo vindo em sua direção, no caso o desvio e a trajetória em si não foi considerada, somente o tempo que o algoritmo demora para realizar a identificação e tomar uma decisão, isso foi realizado a fim de facilitar o processamento dos dados e poder medir com mais precisão os dados obtidos.

Dada a clara distinção entre o tempo gasto durante o processo de detecção de obstáculos e o tempo necessário para o UAV recalculer uma rota livre de colisão, foram medidos os tempos de processamento (PT) de cada um separadamente. Quanto ao tempo de cálculo do U-map e UV-map, trata-se de um processo contínuo ao longo de toda a simulação, resultando em várias medições. Dado o interesse em determinar o limite do recurso, optou-se por utilizar a pior medição obtida como o tempo de execução. Os resultados estão apresentados nas Tabelas 4.2 e

4.3.

Tabela 4.2: Resultados de tempo para recalcular a rota e para detectar os obstáculos com o UV-map

Teste	PT para recalcular da rota	PT UV-map
1	0.320s	0.092s
2	0.247s	0.070s
3	0.291s	0.059s
4	0.160s	0.060s
5	0.544s	0.069s
6	0.234s	0.130s
7	0.282s	0.131s
8	0.480s	0.106s
9	0.355s	0.102s
10	0.472s	0.102s
Média	$0.338 \pm 0.123s$	$0.092 \pm 0.026s$

Tabela 4.3: Resultados de tempo para recalcular a rota e para detectar os obstáculos U-map

Teste	PT para recalcular da rota	PT U-map
1	0.230s	0.010s
2	0.310s	0.009s
3	0.300s	0.009s
4	0.221s	0,008s
5	0.444s	0.010s
6	0.484s	0.007s
7	0.426s	0.005s
8	0.498s	0.006s
9	0.538s	0.009s
10	0.393s	0.009s
Média	0.384 ± 0.113	0.008 ± 0.002

Essa subsecção demonstrou portanto que embora haja uma melhoria clara na definição dos limites físicos dos obstáculos, existe uma perda considerável na velocidade de detecção dos

obstáculos, deixando então o questionamento se esse custo adicional prejudica ou ajuda no desvio de obstáculos dinâmicos, hipótese que será testada na Subseção 4.6.3.

4.6.3 Testes de SR e AGV

Para avaliar o desempenho dos algoritmos, foi atribuída ao UAV a missão de se deslocar da posição inicial (4 m, 4 m, 2 m) até a posição final (30 m, 4 m, 2 m), inicialmente seguindo uma trajetória linear. Essa configuração simula as condições dos testes realizados na seção 4.4, onde é impossível alcançar o objetivo em linha reta devido à presença de um obstáculo dinâmico, exigindo assim desvios.

Conforme mencionado anteriormente, ainda não há um padrão estabelecido para esses testes na bibliografia e portanto, foi adotado um critério que assegura a ocorrência de colisões com obstáculos dinâmicos, caso não haja detecção do obstáculo e adaptação da trajetória.

Durante esses testes são medidos dois valores, o SR e o AGV de dos testes enquanto o UAV voa em três velocidades distintas, 1 m/s, 2 m/s e 3 m/s em 5 tentativas, o que vai testar os limites do método de detecção, detecção de colisão e limites dos atuadores do UAV.

Primeiramente o U-map em conjunto com o sistema de correção de trajetórias do MRS foi testado e apresentou os seguintes resultados:

- Inicialmente, foram conduzidos testes utilizando o algoritmo U-map e o MRS em três velocidades diferentes, os quais resultaram em variações nos valores de SR. A uma velocidade de 1 m/s, observou-se um $SR = 0.4$, indicando que apenas uma das tentativas obteve êxito na conclusão da missão. No entanto, é relevante destacar que dentre as 4 falhas em completar a missão, três delas ocorreram pelo fato do UAV foi impactado por estar direcionado para um ponto que não permitia a visualização do obstáculo, e com a exceção de uma vez, o UAV foi capaz de desviar do obstáculo quando o mesmo estava exposto para a detecção do UAV, este foi capaz de desviar com sucesso. Exemplos de gravações de voos retratando os eventos de falha e sucesso são demonstrados respectivamente nas Figuras 4.7 e 4.9 com suas respectivas trajetórias representadas nas Figuras 4.8 e 4.10;
- Durante os experimentos com o UAV operando a uma velocidade de 2 m/s, alcançou-se um $SR = 0,87$, indicando que o UAV conseguiu evitar todos os obstáculos, atingindo os objetivos das missões como demonstrado nas Figuras 4.11 e com sua respectiva trajetória representada na Figura 4.12;

- Por último, foram conduzidos testes em que o UAV voou a uma velocidade de 3 m/s e registrou um $SR = 0.33$, desviando-se do obstáculo apenas uma vez e apresentando comportamento similar aos demonstrados nas Figuras 4.11 e 4.12. Importante salientar que, em diversas ocasiões, o UAV foi capaz de detectar tanto o obstáculo quanto a iminência de colisão, no entanto, as limitações físicas dos atuadores do UAV não permitiram a execução do desvio a tempo.

FIGURA 4.7 – Exemplo de voo que falhou enquanto o obstáculo atingia na parte traseira do UAV a 1 m/s . Link para o vídeo das imagens: <https://youtu.be/r1BZpuWflg0>

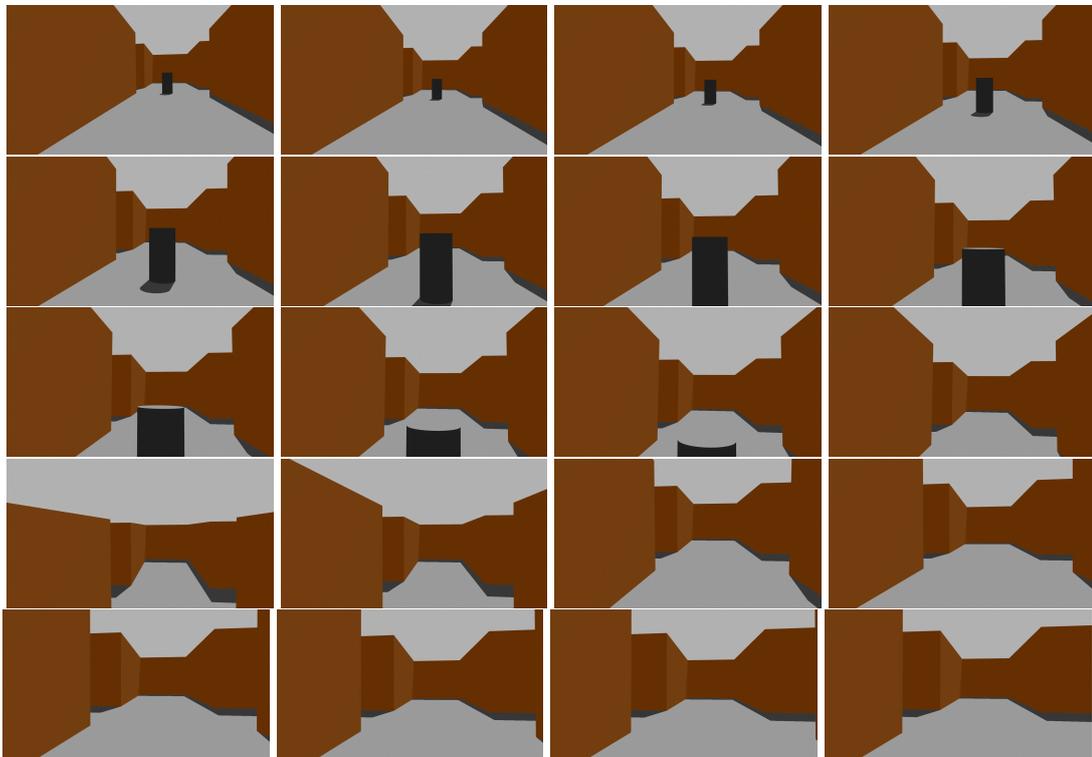


FIGURA 4.8 – Trajetória de voo que falhou enquanto o obstáculo atingia na parte traseira do UAV a 1 m/s.

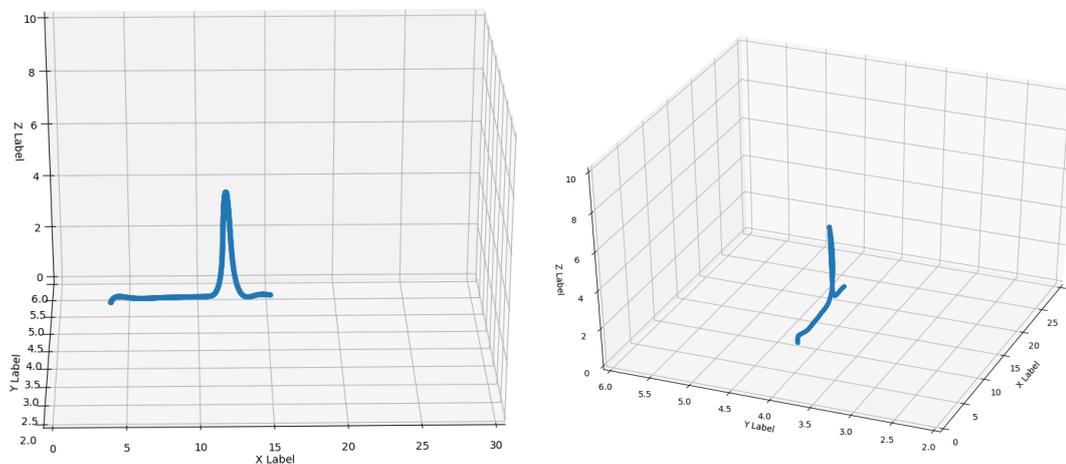


FIGURA 4.10 – Trajetória de voo que foi bem sucedido no voo a 1 m/s.

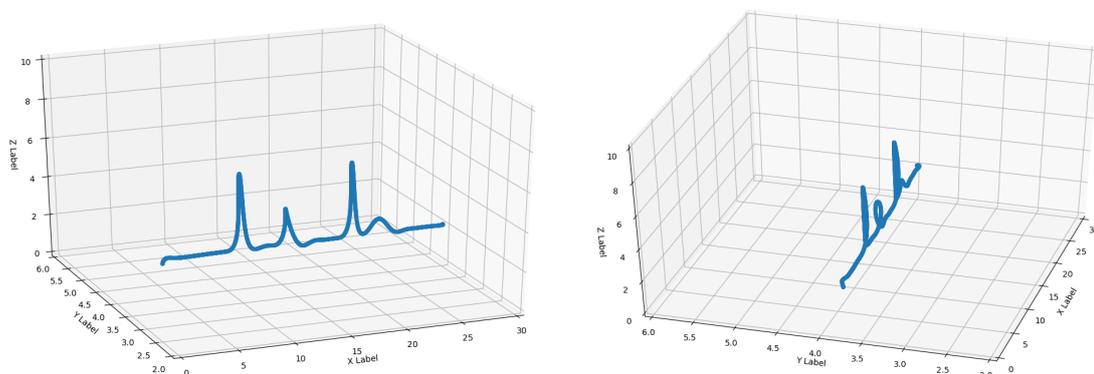


FIGURA 4.9 – Exemplo de voo que foi bem sucedido no voo a 1 m/s. Link para o video das imagens: <https://youtu.be/LcdCtfYK-ko>

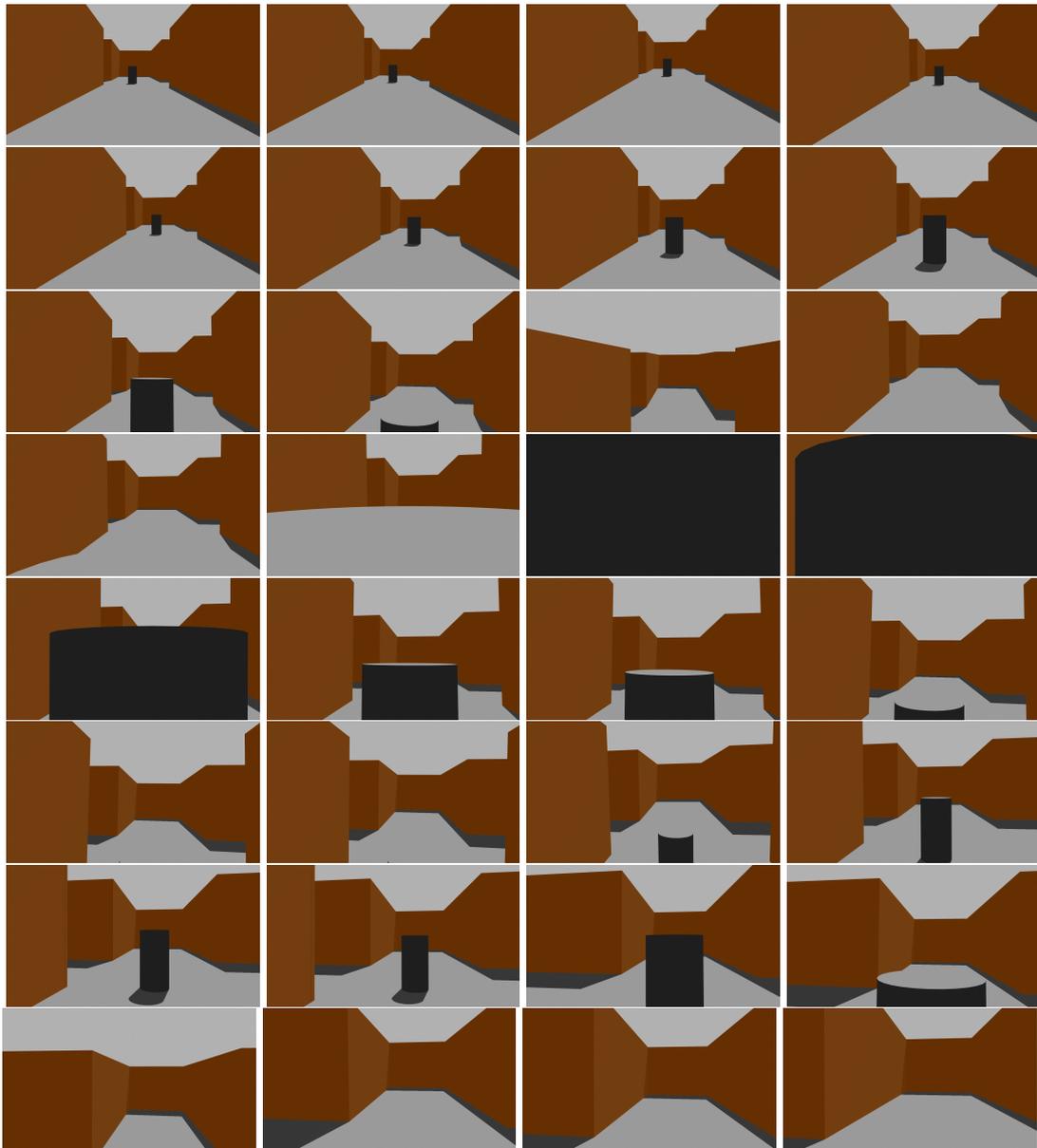


FIGURA 4.11 – Exemplo de voo a 2 m/s que foi bem sucedido no voo. Link para o video das imagens: <https://youtu.be/WDECSvbOP3o>

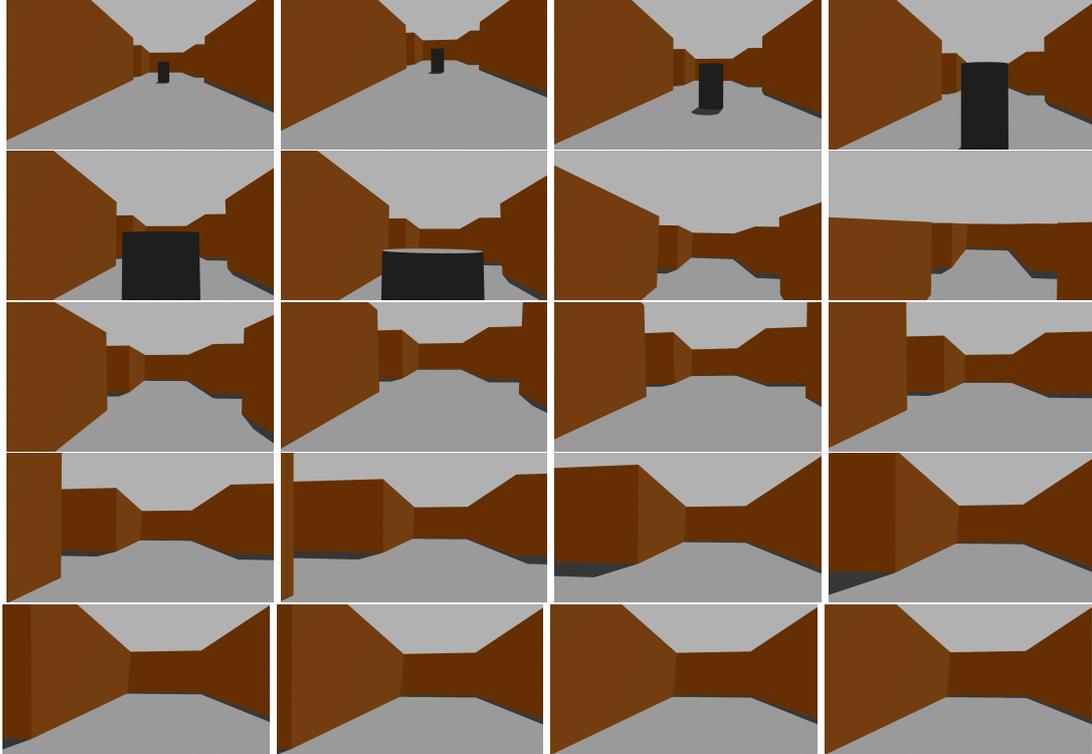
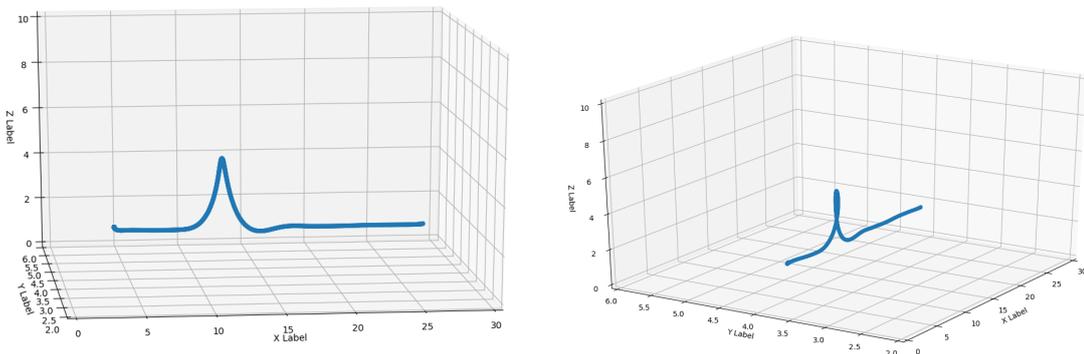


FIGURA 4.12 – Trajetória de voo que falhou enquanto o obstáculo atingia na parte traseira do UAV a 1 m/s.



Os resultados dos AGVs de cada simulação dos testes realizados utilizando o U-map com o MRS nas velocidades testadas estão dispostos na Tabela 4.4, onde cada linha representa um teste e cada uma das colunas seus respectivos resultados de AGV e a última linha mostra uma média dos testes bem sucedidos.

Tabela 4.4: Resultados de AGV por simulação utilizando o U-map

Teste de AGV	1 <i>m/s</i>	2 <i>m/s</i>	3 <i>m/s</i>
1	0.74 <i>m/s</i>	1.88 <i>m/s</i>	-
2	-	1.63 <i>m/s</i>	-
3	-	1.52 <i>m/s</i>	-
4	-	1.51 <i>m/s</i>	2.21 <i>m/s</i>
5	-	1.64 <i>m/s</i>	-
6	-	1.58 <i>m/s</i>	-
7	0.80 <i>m/s</i>	1.60 <i>m/s</i>	2.17 <i>m/s</i>
8	-	1.57 <i>m/s</i>	-
9	-	1.58 <i>m/s</i>	-
10	0.80 <i>m/s</i>	1.57 <i>m/s</i>	2.30 <i>m/s</i>
11	0.77 <i>m/s</i>	1.58 <i>m/s</i>	-
12	0.80 <i>m/s</i>	-	-
13	0.77 <i>m/s</i>	1.73 <i>m/s</i>	-
14	-	1.58 <i>m/s</i>	2.27 <i>m/s</i>
15	-	-	2.15 <i>m/s</i>
Média	0.78 <i>m/s</i>	1,61 <i>m/s</i>	2.22 <i>m/s</i>

Em seguida dos testes que utilizavam o U-map como método de detecção de obstáculos dinâmicos, o algoritmo UV-map com o sistema de correção de trajetórias do MRS foram testados nas mesmas condições:

- Quando submetido a voos utilizando o algoritmo UV-map a uma velocidade de 1 *m/s*, resultando em um $SR = 0.40$. O UAV manifestou o mesmo padrão de comportamento observado com o U-map, representados nas Figuras 4.7, 4.9, 4.8 e 4.10. Mas no caso, todas as instâncias em que o obstáculo foi detectado dentro do campo de visão da câmera de profundidade, o UAV foi capaz de desviar do obstáculo. No entanto, devido à velocidade reduzida do UAV em comparação com a do obstáculo, este último foi capaz de alcançar o UAV, resultando em colisões;
- Igualmente ao caso do U-map os testes com o UAV a uma velocidade de 2 *m/s*, ele obteve um $SR = 0.87$ e obteve resultados similares aos exemplificado nas Figuras 4.11 e 4.12;
- Também demonstrando um comportamento similar ao U-map com o sistema de correção

de trajetórias do MRS e a velocidade de 3 *m/s* o algoritmo também obteve $SR = 0.33$ com um comportamento similar ao do U-map.

Os resultados de AGV de cada teste estão dispostos na Tabela 4.5 e dispostos de maneira idêntica aos apresentados na Tabela 4.4.

Tabela 4.5: Resultados de AGV por simulação utilizando o UV-map

Teste de AGV	1 <i>m/s</i>	2 <i>m/s</i>	3 <i>m/s</i>
1	0.78 <i>m/s</i>	1.53 <i>m/s</i>	2.30 <i>m/s</i>
2	-	1.53 <i>m/s</i>	-
3	-	1.53 <i>m/s</i>	-
4	-	1.52 <i>m/s</i>	-
5	-	1.64 <i>m/s</i>	-
6	-	-	-
7	0.75 <i>m/s</i>	1.53 <i>m/s</i>	-
8	0.79 <i>m/s</i>	1.54 <i>m/s</i>	2.25 <i>m/s</i>
9	-	1.56 <i>m/s</i>	-
10	-	1.53 <i>m/s</i>	-
11	0.75 <i>m/s</i>	1.53 <i>m/s</i>	-
12	-	-	-
13	0.81 <i>m/s</i>	1.54 <i>m/s</i>	-
14	0.79 <i>m/s</i>	-	-
15	-	1.54 <i>m/s</i>	-
Média	0.78 <i>m/s</i>	1.54 <i>m/s</i>	2.27 <i>m/s</i>

Na fase final dos experimentos, o algoritmo UV-map e o algoritmo de correção de trajetórias foi empregado em conjunto com o Plannie, mais especificamente utilizando o RRT-C para realizar o planejamento de trajetória e foram constatados os seguintes resultados:

- No cenário em análise, a uma velocidade de 1 *m/s*, o algoritmo apresentou uma taxa de sucesso $SR = 0.67$. Apesar de essa taxa ser superior às obtidas em outros testes em velocidades mais baixas, é importante notar que, do ponto de vista do comportamento do UAV, os resultados são bastante semelhantes aos observados em outras simulações. Em todas as instâncias em que o obstáculo estava visível para o UAV, o sistema foi capaz

de desviar dele. Nas situações em que o UAV conseguiu contornar o obstáculo, ele, na verdade, seguiu o obstáculo por grande parte da trajetória até o momento em que o obstáculo alterou sua direção, situações similares como as vistas nas Figuras 4.10 e 4.9, evitando assim uma colisão. Essas coincidências podem ser atribuídas ao tempo inicial que o UAV leva para calcular a rota inicial que fazia o UAV iniciar o voo na mesma direção do obstáculo, em vez de ir em direção ao obstáculo. Isso resultava na falta de tempo para que o obstáculo completasse a volta na trajetória programada e alcançasse o UAV para uma possível colisão traseira.

- Nos testes realizados à velocidade de 2 m/s , o UAV apresentou um $SR = 0.33$. No entanto, observa-se que o mesmo fator que proporcionou resultados positivos em velocidades mais baixas acarretou repercussões negativas em velocidades mais elevadas. Desta vez, o atraso no início do deslocamento fez com que o UAV precisasse desviar do obstáculo em momentos nos quais o obstáculo alterava sua direção, mas dada a velocidade do voo e a distância reduzida, não houve tempo o suficiente para identificar a nova rota e desviar do obstáculo e conseqüentemente, ocasionou as duas colisões, inclusive uma delas, o UAV teve tempo de recalculer sua rota mas uma das pernas do UAV tocaram no obstáculo dinâmico.
- Nos testes realizados à velocidade de 3 m/s , foi observado um $SR = 0.13$ resultaram em colisões pelos mesmos motivos observados nos testes anteriores, ou seja, a velocidade relativa do UAV com relação ao obstáculo não permitia que o algoritmo do MRS processasse o recálculo da rota em tempo suficiente para que o UAV executasse o desvio.

Os resultados dos AGVs de cada simulação com o *Plannie* como sistema de gerenciamento de missões estão dispostos na Tabela 4.6 e dispostos de maneira idêntica aos apresentados na Tabela 4.5.

Tabela 4.6: Resultados de AGV por simulação utilizando o UV-map e o Plannie

Teste de AGV	1 m/s	2 m/s	3 m/s
1	-	1.29 m/s	-
2	-	-	-
3	0.80 m/s	-	-
4	0.81 m/s	1.57 m/s	-
5	0.81 m/s	1.61 m/s	-
6	-	- m/s	-
7	0.86 m/s	1.60 m/s	2.10 m/s
8	0.78 m/s	1.52 m/s	-
9	0.85 m/s	-	-
10	0.75 m/s	-	-
11	0.80 m/s	-	-
12	0.76 m/s	-	-
13	0.82 m/s	-	-
14	-	-	-
15	-	-	-
Média	0.80 m/s	1.52 m/s	2.10 m/s

4.7 Considerações finais

Neste capítulo a integração de algoritmos proposta foi colocada em diversas circunstâncias diferentes, desde ser comparada com a versão simplificada do U-map até o teste do algoritmo em combinação com um algoritmo de planejamento de missão. Muito embora o sistema proposto mostre adaptabilidade e robustez para desviar de obstáculos, fica também uma ressalva quanto a importância de se avaliar cautelosamente cada simulação dada a configuração dinâmica, dada a natureza mutável do ambiente, uma conclusão sobre o desempenho do UAV não pode depender exclusivamente dos resultados mensuráveis.

Também foi constatado que, embora o PT do U-map seja significativamente menor do que o PT do UV-map, ainda assim, ele apresenta latência suficiente para a execução dos desvios de obstáculos dinâmicos, além de proporcionar uma melhor determinação das dimensões do obstáculo detectado, mesmo considerando os limites de alcance das câmeras de profundidade.

Esses resultados fortalecem a hipótese de que o UV-map é uma ferramenta adequada para a detecção de obstáculos dinâmicos.

Aprofundando a observação sobre os resultados obtidos além dos resultados numéricos em si, observa-se o desempenho mais veloz do AGV do U-map em comparação com o UV-map a 2 *m/s*. Primeiramente, é necessário notar que o melhor desempenho do primeiro não pode ser atribuído pelo fato dele calcular a rota mais rapidamente, dado que o valor médio do tempo necessário para recalculá-la é irrisório para alterar o tempo de percurso suficientemente para resultar na alteração significativa de AGV, mesmo se considerando o tempo de detecção de obstáculos, esse aumento de PT continua desprezível, o mais provável é que seja apenas uma variação esperada da resposta, sendo assim justificado dizer que em ambos os casos o algoritmo obteve o mesmo resultado.

Quanto aos resultados do SR, embora seja o valor que, do ponto de vista prático, possui uma relevância maior, este ainda é uma variável que precisa ser ponderada antes de ser tomada em consideração. Questionamentos sobre como ocorreram as colisões e se o ambiente oferece obstáculos detectáveis pelo UAV são questões importantes para avaliar se o SR está realmente avaliando o desempenho do algoritmo. No caso em questão, o SR se mostrou relevante nas simulações a 2 *m/s* e 3 *m/s*, indicando uma dificuldade elevada em desviar dos obstáculos a uma velocidade de 3 *m/s*. Sugerindo que velocidades superiores também resultem em falhas no desvio de obstáculos. Os resultados dos SR estão dispostos para cada velocidade e algoritmo de teste como disposto na Tabela 4.7.

Tabela 4.7: Resultados de SR para cada velocidade e algoritmo

	U-map	UV-map	UV-map + Plannie
1 <i>m/s</i>	0.4	0.4	0.33
2 <i>m/s</i>	0,87	0,87	0.13
3 <i>m/s</i>	0.33	0.13	0.07

Tabela 4.8: Resultados de AGV para cada velocidade e algoritmo

	U-map	UV-map	UV-map + Plannie
1 <i>m/s</i>	0.74	0.78	0.81
2 <i>m/s</i>	1.63	1.55	1.485
3 <i>m/s</i>	2.21	2.3	—

Logo, em relação às observações sobre a comparação de desempenho, os dados coletados não revelam discrepâncias significativas entre os métodos propostos que permitam inferir qual

deles apresentou o melhor desempenho em termos de AGV. Quanto ao SR, também não foram encontradas evidências contundentes sobre qual método se destaca, com uma exceção: o sistema UV-map de correção de trajetórias do MRS com o uso do Plannie como planejador de missões a 2 *m/s*. Como explicado, essa integração de sistemas nessa velocidade falhou em duas ocasiões. Em uma delas, o UAV não conseguiu executar a trajetória de desvio por uma margem pequena, e na outra, não foi capaz de recalcular a rota de colisão a tempo devido à proximidade e à mudança abrupta de direção realizada pelo obstáculo.

Capítulo 5

CONCLUSÕES E TRABALHOS FUTUROS

Nesse capítulo serão apresentadas as conclusões obtidas desse trabalho e potenciais trabalhos futuros com os recursos desenvolvidos e avaliados.

5.1 Considerações Finais

Neste estudo, realizou-se a implementação da integração de um sistema de detecção de obstáculos dinâmicos, mais especificamente, adotou-se o método UV-map. Esse método constrói um histograma com os pixels que ultrapassam um determinado valor, aglutinando em seguida as colunas adjacentes que apresentam valores semelhantes. Tal abordagem permite a identificação de obstáculos, possibilitando, com a detecção em múltiplos frames, determinar não apenas a posição e dimensão do obstáculo, mas também sua direção e velocidade de deslocamento, utilizando o EKF. Esse filtro permite construir uma projeção das futuras posições do obstáculo, que em conjunto com o MPC Tracker disponibilizado pelo MRS permite o desvio de obstáculos. A escolha desse método resultou de uma revisão da literatura mais recente sobre o tema, optando-se pelas técnicas que apresentaram maior compatibilidade com os recursos disponíveis.

Após a implementação desta proposta, o método foi comparado com outras duas circunstâncias. Em uma delas, o UV-map foi substituído pelo U-map, enquanto na outra, o Plannie foi empregado como planejador de missões, atribuindo um objetivo ao sistema de controle de trajetórias do MRS, em conjunto com o UV-map. Neste contexto, observou-se que, apesar da maior velocidade de processamento do algoritmo U-map, o UV-map conseguiu fornecer ao algoritmo de detecção de colisões do MRS informações com frequência suficiente para desviar dos obstáculos sem aparentes prejuízos operacionais, além de apresentar uma definição substancialmente mais precisa dos obstáculos.

Posteriormente, o sistema integrado do MRS com o UV-map foi acoplado ao Plannie, atuando como planejador de missões. Nesse cenário, o Plannie fornecia um conjunto de pontos de uma trajetória a ser seguida, e a implementação proposta ajustava essa trajetória do UAV quando detectava um caminho de obstáculos que coincidia com o do UAV. Os resultados obtidos demonstraram pouca alteração nos resultados de AVG, apenas observando uma redução no SR, que, como explicado, decorreu da natureza do ambiente de testes.

Diante desses resultados, o método proposto neste trabalho mostrou-se como uma solução viável para a detecção e desvio de obstáculos em diversas circunstâncias. No entanto, é crucial ressaltar que as condições não são universais para todos os UAVs, uma vez que diferentes meios apresentam características bastante específicas e alterações nas condições, como variações na iluminação, podem impactar os resultados de maneira significativa.

5.2 Trabalhos Futuros

Trabalhos futuros com os recursos desenvolvidos por esse trabalho incluem a utilização do UV-map em outros métodos de desvio de obstáculos, a integração do UV-map com outro método de desvio de obstáculos dentro do próprio Plannie, testes do UV-map em ambientes reais e outras aplicações que necessitam de identificação de obstáculos dinâmicos. No campo de IA, o algoritmo de UV-map também pode auxiliar na criação de dados para treinamento de rede neurais, para uso tanto de redes que usam o mapa de profundidade quanto redes que usam imagens RGB ou RGB-D para a detecção de obstáculos. No campos de identificação de obstáculos dinâmicos, outros métodos de detecção de obstáculos poderiam substituir o UV-map e seria possível comparar os resultados obtidos.

REFERÊNCIAS

- ARULAMPALAM, M. S. et al. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, v. 50, p. 174–188, 2002.
- BACA, T. et al. Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 6753–6760. ISBN 9781538680940. ISSN 21530866.
- BACA, T. et al. The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *Journal of Intelligent and Robotic Systems: Theory and Applications*, Journal of Intelligent Robotic Systems, v. 102, 2021. ISSN 15730409.
- CHEN, X.; ZHAO, M.; YIN, L. Dynamic Path Planning of the UAV Avoiding Static and Moving Obstacles. *Journal of Intelligent and Robotic Systems: Theory and Applications*, Journal of Intelligent Robotic Systems, v. 99, p. 909–931, 2020. ISSN 15730409.
- DAI, X. et al. Automatic obstacle avoidance of quadrotor UAV via CNN-based learning. *Neurocomputing*, Elsevier B.V., v. 402, p. 346–358, 2020. ISSN 18728286. Disponível em: <https://doi.org/10.1016/j.neucom.2020.04.020>.
- ESS, A. et al. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, v. 29, p. 1707–1724, 2010.
- FALANGA, D.; KIM, S.; SCARAMUZZA, D. How Fast Is Too Fast? the Role of Perception Latency in High-Speed Sense and Avoid. *IEEE Robotics and Automation Letters*, v. 4, n. 2, p. 1884–1891, 2019. ISSN 23773766.
- FREIMUTH, H.; KÖNIG, M. Planning and executing construction inspections with unmanned aerial vehicles. *Automation in Construction*, Elsevier, v. 96, p. 540–553, 2018. ISSN 09265805. Disponível em: <https://doi.org/10.1016/j.autcon.2018.10.016>.
- GANDHI, D.; PINTO, L.; GUPTA, A. Learning to fly by crashing. *IEEE International Conference on Intelligent Robots and Systems*, p. 3948–3955, 4 2017. Disponível em: <http://arxiv.org/abs/1704.05588>.
- GONZÁLEZ-DESANTOS, L. M. et al. Indoor path-planning algorithm for uav-based contact inspection. *Sensors (Switzerland)*, v. 21, n. 2, p. 1–24, 2021. ISSN 14248220.
- JAFARI, O. H.; MITZEL, D.; LEIBE, B. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In: . [s.n.], 2014. p. 5636–5643. Disponível em: <http://www.vision.rwth-aachen.de/software>.

- KAMEL, M. et al. Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. v. 2017-September, p. 236–243. ISBN 9781538626825. ISSN 21530866.
- LAVALLE, S. M. et al. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and computational robotics: new directions*, Wellesley, v. 5, p. 293–308, 2001.
- LEE, H. Y.; HO, H. W.; ZHOU, Y. Deep Learning-based Monocular Obstacle Avoidance for Unmanned Aerial Vehicle Navigation in Tree Plantations. *Journal of Intelligent Robotic Systems*, v. 101, n. 1, p. 18, 2021. ISSN 0921-0296. Disponível em: <http://link.springer.com/10.1007/s10846-020-01284-z>.
- LEE, T.; LEOK, M.; MCCLAMROCH, N. H. Geometric tracking control of a quadrotor uav on $se(3)$. In: *49th IEEE Conference on Decision and Control (CDC)*. [S.l.: s.n.], 2010. p. 5420–5425.
- LIN, J.; ZHU, H.; ALONSO-MORA, J. Robust Vision-based Obstacle Avoidance for Micro Aerial Vehicles in Dynamic Environments. In: *IEEE International Conference on Robotics and Automation*. [s.n.], 2020. p. 2682–2688. ISBN 9781728173955. ISSN 23318422. Disponível em: <http://arxiv.org/abs/2002.04920>.
- LOQUERCIO, A. et al. Deep Drone Racing: From Simulation to Reality with Domain Randomization. *IEEE Transactions on Robotics*, v. 36, n. 1, p. 1–14, 2019. ISSN 19410468.
- LU, L. et al. A robust and fast collision-avoidance approach for micro aerial vehicles using a depth sensor. *Remote Sensing*, v. 13, n. 9, p. 1–14, 2021. ISSN 20724292.
- LU, L. et al. Fixed-wing UAV path planning in a dynamic environment via dynamic RRT algorithm. In: *Mechanism and Machine Science: Proceedings of ASIAN MMS 2016 CCMMS 2016*. [S.l.]: Springer, Singapore, 2017. v. 408, p. 271–282. ISBN 9789811028748.
- OLEYNIKOVA, H.; HONEGGER, D.; POLLEFEYS, M. Reactive avoidance using embedded stereo vision for MAV flight. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle: IEEE, 2015. p. 50–56. ISBN 978-1-4799-6923-4. Disponível em: <http://ieeexplore.ieee.org/document/7138979/>.
- PETRLÍK, M. et al. A robust uav system for operations in a constrained environment. *IEEE Robotics and Automation Letters*, IEEE, v. 5, n. 2, p. 2169–2176, 2020.
- QIAN, X. et al. Dynamic obstacle avoidance path planning of UAVs. In: *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015. p. 8860–8865. ISBN 978-9-8815-6389-7. ISSN 21612927. Disponível em: <http://ieeexplore.ieee.org/document/7261039/>.
- ROCHA, L. G. S. da. Desenvolvimento de um algoritmo de planejamento de trajetória em ambientes desconhecidos e não estruturados para uavs. 2021.
- SADEGHI, F.; LEVINE, S. CAD2RL: Real Single-Image Flight without a Single Real Image. *Robotics: Science and Systems*, v. 13, p. 1–12, nov 2016. ISSN 2330765X. Disponível em: <http://arxiv.org/abs/1611.04201>.
- SAHA, A. et al. Efficient obstacle detection and tracking using rgb-d sensor data in dynamic environments for robotic applications. *Sensors*, MDPI, v. 22, 9 2022. ISSN 14248220.

- SANDINO, J. et al. UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments. *Remote Sensing*, v. 12, n. 20, p. 1–31, 2020. ISSN 20724292.
- WANG, D. et al. UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. *Computers and Electronics in Agriculture*, Elsevier, v. 175, p. 1–11, 2020. ISSN 01681699. Disponível em: <https://doi.org/10.1016/j.compag.2020.105523>.
- XI, Y.-G.; ZHANG, C.-G. Rolling path planning of mobile robot in a kind of dynamic uncertain environment. v. 28, p. 161–175, 03 2002.
- YAO, M.; ZHAO, M. Unmanned aerial vehicle dynamic path planning in an uncertain environment. *Robotica*, v. 33, n. 3, p. 611–621, 2015. ISSN 14698668.
- YAO, Y. et al. Dynamic trajectory planning for unmanned aerial vehicle based on sparse a* search and improved artificial potential field. *Kongzhi Lilun Yu Yingyong/Control Theory and Applications*, v. 27, n. 7, p. 953–959, 2010.
- YU, H.; CROON, G. C. H. E. de; WAGTER, C. D. Avoidbench: A high-fidelity vision-based obstacle avoidance benchmarking suite for multi-rotors. In: . IEEE, 2023. p. 9183–9189. Disponível em: <http://arxiv.org/abs/2301.07430>.
- YUAN, J. et al. Dynamic obstacle detection method based on u–v disparity and residual optical flow for autonomous driving. *Scientific Reports*, Nature Research, v. 13, 12 2023. ISSN 20452322.
- ZAMMIT, C.; KAMPEN, E.-J. V. Comparison between A* and RRT Algorithms for UAV Path Planning. In: *2018 AIAA Guidance, Navigation, and Control Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2018. p. 1–23. ISBN 978-1-62410-526-5. Disponível em: <https://arc.aiaa.org/doi/10.2514/6.2018-1846>.
- ZHONG, J. et al. A safer vision-based autonomous planning system for quadrotor uavs with dynamic obstacle trajectory prediction and its application with llms. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. [S.l.: s.n.], 2024. p. 920–929.
- ZHU, H.; ALONSO-MORA, J. Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments. *IEEE Robotics and Automation Letters*, v. 4, n. 2, p. 776–783, 2019. ISSN 23773766.

Kim, Pedro Henrique Corrêa

Desvio de Obstáculos para UAVs em Ambientes
Dinâmicos / Pedro Henrique Corrêa Kim -- 2024.
62f.

Dissertação (Mestrado) - Universidade Federal de São
Carlos, campus Sorocaba, Sorocaba

Orientador (a): Kelen Cristiane Teixeira Vivaldini

Banca Examinadora: Kelen Cristiane Teixeira Vivaldini,
Tatiana de Figueiredo Pereira Alves Taveira Pazelli,
Denis Fernando Wolf

Bibliografia

1. Robôs autônomos, Evitar colisões, Detecção de
obstáculos dinâmicos. I. Kim, Pedro Henrique Corrêa. II.
Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática
(SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Maria Aparecida de Lourdes Mariano -
CRB/8 6979



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Pedro Henrique Correa Kim, realizada em 28/02/2024.

Comissão Julgadora:

Profa. Dra. Kelen Cristiane Teixeira Vivaldini (UFSCar)

Profa. Dra. Tatiana de Figueiredo Pereira Alves Taveira Pazelli (UFSCar)

Prof. Dr. Denis Fernando Wolf (USP)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.