



# Inspeção Autónoma de Turbinas Eólicas Offshore com UAVs Residentes

**SAMUEL FERNANDO BARROS DA COSTA**

Setembro de 2025

POLITÉCNICO DO PORTO  
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

---

# Inspeção Autónoma de Turbinas Eólicas Offshore com UAVs Residentes

---

Samuel Fernando Barros da Costa

Mestrado em Engenharia Electrotécnica e de Computadores  
Área de Especialização em Automação e Sistemas

**ISEP** INSTITUTO SUPERIOR  
DE ENGENHARIA DO PORTO

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA  
Instituto Superior de Engenharia do Porto

Setembro, 2025



*Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Automação e Sistemas.*

**Candidato:** Samuel Fernando Barros da Costa, N.º 1201578,  
1201578@isep.ipp.pt

**Orientação Científica:** André Miguel Pinheiro Dias, APD@isep.ipp.pt

**ISEP** INSTITUTO SUPERIOR  
DE ENGENHARIA DO PORTO

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA  
Instituto Superior de Engenharia do Porto  
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Setembro, 2025



*Dedico este trabalho à minha família, pelo amor incondicional e pelo apoio constante em todos os momentos da minha vida. À minha namorada, pela compreensão, paciência e incentivo, que foram fundamentais para ultrapassar os desafios deste percurso. Aos meus amigos, pela amizade verdadeira, motivação e companheirismo, que tornaram esta caminhada mais leve e inspiradora. A todos os professores que me acompanharam ao longo do caminho, pelo conhecimento transmitido e pela orientação que, direta ou indiretamente, contribuíram para a realização desta dissertação. A todos, deixo a minha sincera gratidão por acreditarem em mim e por tornarem possível a concretização deste trabalho.*



# Agradecimentos

Gostaria de expressar a minha profunda gratidão ao meu orientador, pela dedicação, disponibilidade e orientação científica, que foram fundamentais para a realização deste trabalho.

Agradeço igualmente a todos os professores que, ao longo do meu percurso acadêmico, compartilharam o seu conhecimento e experiência, contribuindo de forma direta ou indireta para a concretização desta dissertação.

Dirijo também um agradecimento especial aos meus colegas de curso, pelo apoio, companheirismo e momentos de partilha, que tornaram este percurso mais enriquecedor e motivador.

A todos, deixo o meu sincero reconhecimento pelo incentivo, pela ajuda e pelos contributos que tornaram possível a conclusão desta dissertação.



# Resumo

A crescente adoção de turbinas eólicas **offshore** exige soluções para a sua inspeção, capazes de reduzir custos e riscos associados a métodos convencionais. Esta dissertação apresenta um **pipeline de inspeção autónoma com UAVs residentes**, estruturado em várias etapas que vão desde a aproximação à turbina até ao retorno seguro, incluindo deteção de obstáculos, seguimento das pás e coordenação multi-UAV.

O sistema integra sensores LiDAR e câmara RGB com algoritmos de visão computacional e segmentação, permitindo planear trajetórias seguras e garantir cobertura completa das pás. A abordagem foi validada em ambiente simulado, recorrendo às ferramentas *Gazebo*, *ROS* e *PX4*, onde foram avaliadas métricas de desempenho como tempo de missão, distância percorrida e robustez em diferentes cenários operacionais.

Os resultados demonstraram que a solução proposta possibilita inspeções completas e seguras, mesmo perante restrições impostas pela geometria da turbina, e que a utilização de múltiplos UAVs contribui para ganhos significativos de eficiência na execução da missão.

**Palavras-Chave:** inspeção autónoma, UAVs, turbinas eólicas offshore, LiDAR, visão computacional



# Abstract

The increasing adoption of **offshore** wind turbines demands innovative inspection solutions that can reduce costs and risks associated with conventional methods. This dissertation presents an **autonomous inspection pipeline with resident UAVs**, structured in several stages ranging from the approach to the turbine to the safe return, including obstacle detection, blade tracking, and multi-UAV coordination.

The system integrates LiDAR and RGB camera sensors with computer vision and segmentation algorithms, enabling safe trajectory planning and ensuring complete blade coverage. The approach was validated in a simulated environment using *Gazebo*, *ROS*, and *PX4*, where performance metrics such as mission time, traveled distance, and robustness under different operational scenarios were evaluated.

The results demonstrated that the proposed solution enables complete and safe inspections, even under geometric constraints imposed by the turbine, and that the use of multiple UAVs provides significant efficiency gains in mission execution.

**Keywords:** autonomous inspection, UAVs, offshore wind turbines, LiDAR, computer vision



# Índice

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Listagens</b>	<b>xiii</b>
<b>Lista de Algoritmos</b>	<b>xv</b>
<b>Lista de Acrónimos</b>	<b>xvii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	2
1.2 Modelos de Manutenção em Parques Eólicos Offshore . . . . .	4
1.3 Logística e Níveis de Autonomia em UAVs para Inspeção Offshore . . . . .	6
1.4 Turbinas Eólicas Offshore . . . . .	8
1.5 Definição do Problema . . . . .	12
1.5.1 Objetivos . . . . .	13
1.5.2 Resultados Esperados . . . . .	13
1.6 Plano de Trabalho . . . . .	13
1.7 Organização da Dissertação . . . . .	14
<b>2 Estado da Arte</b>	<b>17</b>
2.1 Otimização Global . . . . .	17
2.2 Inspeção Local . . . . .	18
2.2.1 Técnicas de Inspeção na Manutenção Proativa Offshore . . . . .	18
2.2.2 Tipos de UAVs para Inspeção de Turbinas Eólicas Offshore . . . . .	20
2.2.3 Sensores Integrados nos UAVs . . . . .	22
2.2.4 Métodos Visuais . . . . .	24
2.2.5 Métodos Multissensoriais . . . . .	27
2.2.6 Soluções Comerciais com UAVs na Inspeção Offshore . . . . .	32
2.3 Síntese e Considerações Finais . . . . .	33
<b>3 Fundamentação Técnica</b>	<b>35</b>
3.1 Robot Operating System (ROS) . . . . .	35
3.2 Gazebo . . . . .	36

3.3	PX4 Autopilot . . . . .	36
3.4	Câmaras RGB e Modelo de Projeção Pinhole . . . . .	37
3.5	LiDAR 3D . . . . .	38
3.6	Controlo PID . . . . .	39
3.7	RANSAC . . . . .	40
3.8	DBSCAN . . . . .	41
3.9	Ângulos de Rotação em Turbinas Eólicas . . . . .	42
<b>4</b>	<b>Implementação</b>	<b>45</b>
4.1	Arquitetura do Sistema . . . . .	46
4.2	Ambiente de Simulação com UAVs . . . . .	49
4.2.1	Turbina <i>Offshore</i> . . . . .	49
4.2.2	UAV Yuneec Typhoon H480 com LiDAR Velodyne VLP-16 . . . . .	51
4.3	Condições Iniciais . . . . .	55
4.3.1	Condição 1: Turbina fixa, ambiente controlado e comunicações com o planeador de missões . . . . .	55
4.3.2	Condição 2: Limites do ângulo de <i>roll</i> das pás . . . . .	55
4.3.3	Condição 3: Limites do ângulo de <i>yaw</i> da nacelle . . . . .	58
4.3.4	Condição 4: Limites do ângulo de <i>pitch</i> das pás . . . . .	58
4.4	Distância de Inspeção e Projeção do Campo de Visão . . . . .	59
4.5	Etapa 0 — Descolagem . . . . .	61
4.6	Etapa 1 — Alinhamento com a torre . . . . .	62
4.7	Etapa 2 — Subida até ao <i>hub</i> . . . . .	65
4.8	Etapa 3 — Alinhamento com o <i>hub</i> . . . . .	67
4.9	Etapa 4 — Inspeção da pá . . . . .	70
4.10	Etapa 5 — Descida controlada . . . . .	72
4.11	Etapa 6 — Localização da plataforma de aterragem . . . . .	74
4.12	Etapa 7 — Aterragem . . . . .	75
4.13	Planeador de Missão . . . . .	77
<b>5</b>	<b>Resultados</b>	<b>79</b>
5.1	Testes do Ângulo de <i>Yaw</i> da Nacelle . . . . .	80
5.1.1	Caso Ideal: Offset de <i>Yaw</i> de $0^\circ$ . . . . .	81
5.1.2	Caso Limite Viável: Offset de <i>Yaw</i> de $40^\circ$ . . . . .	83
5.1.3	Caso Inviável: Offset de <i>Yaw</i> de $50^\circ$ . . . . .	84
5.2	Testes do Ângulo de <i>Roll</i> das Pás . . . . .	85
5.2.1	Caso Inviável: Offset de <i>Roll</i> de $60^\circ$ . . . . .	86
5.2.2	Caso Limite: Offset de <i>Roll</i> de $30^\circ$ . . . . .	88
5.2.3	Caso Inviável: Offset de <i>Roll</i> de $40^\circ$ . . . . .	91
5.3	Múltiplos UAVs . . . . .	92

<b>6</b>	<b>Conclusão</b>	<b>95</b>
<b>7</b>	<b>Appendix A</b>	<b>103</b>
7.1	Configuração da Turbina Eólica . . . . .	103
7.2	Janelas de Inspeção . . . . .	104



# Lista de Figuras

1.1	Classificação dos principais modelos de manutenção em parques eólicos offshore [1]. . . . .	4
1.2	Comparação entre diferentes estratégias de manutenção. Verde: turbinas operacionais; vermelho: falha; amarelo: manutenção programada [1]. . . . .	5
1.3	Esquema representativo de um parque eólico offshore flutuante [4]. . . . .	6
1.4	Diferentes tipos de subestruturas para turbinas eólicas offshore em função da profundidade [7]. . . . .	8
1.5	Componentes internos e externos típicos de uma pá de turbina eólica [9]. . . . .	10
1.6	Componentes mecânicos e elétricos principais da <i>nacelle</i> [9]. . . . .	11
1.7	Evolução histórica e projeções futuras das dimensões das turbinas eólicas [8]. . . . .	12
2.1	Classificação dos principais tipos de UAVs [18], [19]. . . . .	21
3.1	Modelo simplificado de uma turbina eólica <i>offshore</i> , indicando as três rotações principais: <i>yaw</i> da nacelle, <i>roll</i> das pás e <i>pitch</i> das pás [29]. . . . .	43
3.2	Diferentes configurações de rotação em turbinas: (a) <i>roll</i> das pás; (b) <i>yaw</i> da nacelle [28]. . . . .	44
4.1	Arquitetura do sistema proposto. . . . .	47
4.2	Representação visual das etapas da missão autónoma de inspeção . . . . .	48
4.3	Segmentação da turbina em componentes independentes (pás, <i>nacelle</i> , <i>hub</i> e torre) e cálculo das envolventes geométricas de cada peça, utilizadas para estabelecer limites de inspeção. . . . .	50
4.4	Modelo tridimensional da turbina eólica <i>offshore</i> , adaptado a partir de [51] e integrado no ambiente de simulação <i>Gazebo</i> . . . . .	51
4.5	Drone Yuneec Typhoon H480 original [52]. . . . .	52
4.6	Sensor LiDAR Velodyne VLP-16 [53]. . . . .	52
4.7	Offset vertical de 0,312 m entre a câmara RGB e o LiDAR no modelo Typhoon H480 adaptado. . . . .	53
4.8	Três UAVs residentes na plataforma junto à turbina eólica <i>offshore</i> . . . . .	54
4.9	Configuração ideal das pás: pá central vertical, Y invertido. As zonas <i>Central</i> , <i>Esquerda</i> e <i>Direita</i> são claramente distinguíveis. . . . .	56

4.10	Configuração das pás com offset de $30^\circ$ no ângulo de <i>roll</i> : a identificação das zonas <i>Central</i> , <i>Esquerda</i> e <i>Direita</i> permanece válida. . . .	57
4.11	Configuração das pás com offset de $40^\circ$ no ângulo de <i>roll</i> : a identificação das zonas torna-se ambígua, ultrapassando o limite admissível. . . .	58
4.12	Projeção do campo de visão da câmara RGB e do LiDAR (FOV horizontal limitado a $20^\circ$ ) a 12 m de distância . . . . .	60
4.13	Vista superior da plataforma de aterragem com os marcadores <i>ArUco</i> individuais. . . . .	62
4.14	Etapa 1: Exemplo de trajetória circular seguida pelo UAV . . . . .	65
4.15	Etapa 2: Subida incremental mantendo distância igual ou superior a 25 m da torre. . . . .	67
4.16	Etapa 3: Alinhamento em <i>yaw</i> . . . . .	68
4.17	Etapa 3: Alinhamento em <i>y</i> após alinhamento em <i>yaw</i> . . . . .	69
4.18	Etapa 4: Ajuste no plano das pás com <i>RANSAC</i> . . . . .	70
4.19	Etapa 7: Aterragem no <i>ArUco</i> exclusivo da sua plataforma. . . . .	76
5.1	Configuração da turbina com pás em formação de <i>Y invertido</i> , sem offset do ângulo de <i>roll</i> . . . . .	81
5.2	Trajetória 3D do UAV1 no teste com erro de <i>yaw</i> de $0^\circ$ . . . . .	82
5.3	Trajetória 3D do UAV1 no teste com erro de <i>yaw</i> de $40^\circ$ . . . . .	83
5.4	Trajetória 3D do <i>Unmanned Aerial Vehicle</i> (UAV) no teste com erro de <i>yaw</i> de $50^\circ$ . . . . .	85
5.5	Configuração da turbina com offset de <i>roll</i> de $60^\circ$ ( <i>Y normal</i> ), em que uma pá bloqueia parcialmente a torre, tornando a inspeção inviável. . . . .	86
5.6	Trajetória 3D do UAV1 no teste com offset de <i>roll</i> de $60^\circ$ . . . . .	87
5.7	Configuração da turbina com offset de <i>roll</i> de $30^\circ$ . . . . .	89
5.8	Trajetória 3D do UAV1 no teste com offset de <i>roll</i> de $30^\circ$ . As etapas são representadas por cores distintas. . . . .	90
5.9	Configuração da turbina com offset de <i>roll</i> de $40^\circ$ . . . . .	91
5.10	Trajetória 3D do UAV1 no teste com offset de <i>roll</i> de $40^\circ$ . . . . .	92
5.11	Resultado no teste multi-drone. . . . .	94

# Lista de Tabelas

1.1	Domínios funcionais no ciclo de vida de parques eólicos offshore [2]. . . . .	3
1.2	Classificação dos níveis de autonomia em UAVs aplicados à inspeção offshore [6]. . . . .	7
2.1	Comparação entre diferentes métodos de inspeção [4], [17]. . . . .	20
2.2	Comparação entre os principais tipos de UAVs [18], [19]. . . . .	22
2.3	Comparação dos principais sensores [17], [20]. . . . .	24
4.1	Características originais do UAV Yuneec Typhoon H480 [45]. . . . .	52
4.2	Características do sensor LiDAR Velodyne VLP-16 [53]. . . . .	53
4.3	Síntese das portas de comunicação e <i>namespaces</i> de cada UAV. . . . .	54
5.1	Resultado no teste com erro de <i>yaw</i> de 0°. . . . .	82
5.2	Resultado no teste com erro de <i>yaw</i> de 40°. . . . .	84
5.3	Resultado no teste com erro de <i>yaw</i> de 50°. . . . .	85
5.4	Resultado no teste com offset de <i>roll</i> de 60°. . . . .	88
5.5	Resultado no teste com offset de <i>roll</i> de 30°. . . . .	90
5.6	Resultado no teste com offset de <i>roll</i> de 40°. . . . .	92
5.7	Resultado no teste multi-drone. . . . .	94



# Listagens

7.1	Cálculo das janelas de inspeção do LiDAR e da câmera . . . . .	104
-----	--	-----



# Lista de Algoritmos

4.1	Etapa 1 – Aproximação e alinhamento preliminar . . . . .	64
4.2	Etapa 2 – Subida ao longo da torre . . . . .	66
4.3	Etapa 3 – Ajuste no plano do <i>hub</i> . . . . .	69
4.4	Etapa 4 — Inspeção das pás . . . . .	72
4.5	Etapa 5 — Descida segura . . . . .	73
4.6	Etapa 6 — Localização da plataforma de aterragem . . . . .	75
4.7	Etapa 7 — Aterragem no marcador <i>ArUco</i> . . . . .	77
4.8	Planeador de Missão — Lançamento e Monitorização de UAVs . . . . .	78



# Lista de Acrónimos

<b>AFS</b>	Automatic Flight System
<b>APC</b>	Automatic Parameter Calculation
<b>ASV</b>	Autonomous Surface Vessel
<b>ATSP</b>	Asymmetric Traveling Salesman Problem
<b>ATSP-PL</b>	Asymmetric Traveling Salesman Problem with Precedence Loss
<b>BVLOS</b>	Beyond Visual Line of Sight
<b>C-CPP</b>	Collaborative Coverage Path Planning
<b>CFRP</b>	Carbon-Fiber Reinforced Polymer
<b>CNN</b>	Convolutional Neural Network
<b>CTV</b>	Crew Transfer Vessel
<b>DBSCAN</b>	Density-Based Spatial Clustering of Applications with Noise
<b>DKF</b>	Discrete Kalman Filter
<b>EEPPA</b>	Energy Efficient Path Planning Algorithm
<b>EKF</b>	Extended Kalman Filter
<b>FAST</b>	Features from Accelerated Segment Test
<b>FCU</b>	Flight Control Unit
<b>GFRP</b>	Glass-Fiber Reinforced Polymer
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	<i>Global Positioning System</i>
<b>HBB</b>	Horizontal Bounding Box
<b>HLV</b>	Heavy Lift Vessel
<b>IA</b>	Inteligência Artificial

<b>IMU</b>	Inertial Measurement Unit
<b>INESC TEC</b>	Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência
<b>KF</b>	Kalman Filter
<b>LCOE</b>	Levelized Cost of Energy
<b>LiDAR</b>	Light Detection and Ranging
<b>Mask R-CNN</b>	Mask Region-Based Convolutional Neural Network
<b>ML</b>	<i>Machine Learning</i>
<b>MPC</b>	Model Predictive Control
<b>NDT</b>	<i>Non-Destructive Testing</i> (Inspeção Não Destrutiva)
<b>NMPC</b>	Nonlinear Model Predictive Control
<b>O&amp;M</b>	Operação e Manutenção
<b>OctoMap</b>	Probabilistic 3D Mapping Framework using Octrees
<b>PAMPC</b>	Perception-Aware Model Predictive Control
<b>PD</b>	Proportional–Derivative Controller
<b>PID</b>	Proportional–Integral–Derivative
<b>PnP</b>	Perspective-n-Point
<b>PX4</b>	Open-Source Flight Control Platform
<b>RANSAC</b>	Random Sample Consensus
<b>RBB</b>	Rotated Bounding Box
<b>RGB</b>	Red Green Blue
<b>RGB-D</b>	Red Green Blue - Depth
<b>ROS</b>	Robot Operating System
<b>RTK-GNSS</b>	Real-Time Kinematic Global Navigation Satellite System
<b>SBC</b>	Single-Board Computer
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>SOV</b>	Service Operation Vessel

<b>STL</b>	Standard Tessellation Language
<b>ToF</b>	Time of Flight
<b>TRL</b>	<i>Technology Readiness Level</i>
<b>TSP</b>	Traveling Salesman Problem
<b>UAV</b>	<i>Unmanned Aerial Vehicle</i>
<b>USV</b>	<i>Unmanned Surface Vehicle</i>
<b>UWB</b>	Ultra-Wideband
<b>VLOS</b>	Visual Line of Sight
<b>VT-NMPC</b>	Visual Tracking Nonlinear Model Predictive Control
<b>WTD</b>	Wind Turbine Detection Dataset
<b>YOLO</b>	You Only Look Once



## Capítulo 1

# Introdução

A energia eólica offshore tem vindo a assumir um papel cada vez mais importante na produção de eletricidade a nível global, devido à sua capacidade de gerar energia de forma sustentável e eficiente. A inspeção regular das turbinas eólicas, especialmente das pás, é fundamental para assegurar a fiabilidade, segurança e desempenho destes sistemas, sendo tradicionalmente realizada por operadores humanos em condições difíceis e muitas vezes arriscadas.

O uso de veículos aéreos não tripulados (UAVs) surge como uma solução promissora para apoiar estas inspeções, permitindo reduzir custos, aumentar a segurança e facilitar o acesso a zonas de difícil alcance. Contudo, a implementação de sistemas autónomos de inspeção continua a apresentar desafios, relacionados com a perceção do ambiente, planeamento de trajetórias, controlo de voo e coordenação entre múltiplos UAVs.

Esta dissertação foca-se no desenvolvimento de um sistema de inspeção autónoma de turbinas eólicas offshore utilizando UAVs residentes, abordando o desenvolvimento de um ambiente de simulação, a integração de sensores e o desenho de algoritmos de navegação e controlo. O trabalho combina uma abordagem científica, fundamentada no estado da arte, com a implementação prática em cenário de testes, permitindo avaliar a eficácia das soluções propostas e preparar o caminho para aplicações futuras em contextos reais.

## 1.1 Contextualização

A energia eólica offshore tem-se afirmado como uma das tecnologias-chave para a transição energética global, conjugando elevado potencial de geração com uma pegada territorial reduzida. Entre os principais fatores que explicam o seu crescimento destacam-se a escassez de terrenos onshore, a maior intensidade e regularidade dos ventos em ambiente marítimo, bem como a diminuição do impacto visual e acústico sobre populações humanas [1], [2].

Segundo o *Offshore Wind Market Report: 2024 Edition* [3], a capacidade instalada global de energia eólica offshore atingiu **68,3 GW até ao final de 2023**, correspondendo a um acréscimo anual de 6,3 GW — o quarto maior valor de sempre. Este resultado confirma uma tendência de aceleração que deverá manter-se ao longo da próxima década. A China lidera o setor, com mais de **38,6 GW** instalados, seguida pelo Reino Unido com cerca de **14,8 GW**, reforçando o seu papel como segundo maior mercado mundial.

O potencial de expansão é igualmente expressivo: o pipeline global de projetos — incluindo empreendimentos em fase de planeamento, licenciamento ou construção — ultrapassa atualmente os **453 GW**. As projeções mais conservadoras apontam para uma capacidade instalada de **421 GW até 2035**, enquanto estimativas mais otimistas indicam que esse valor poderá alcançar **486 GW**, representando um crescimento superior a cinco vezes face ao estado atual [3].

O ciclo de vida de um parque eólico offshore integra múltiplas atividades, habitualmente agrupadas em quatro grandes domínios operacionais [2], conforme sintetizado na Tabela 1.1.

Tabela 1.1: Domínios funcionais no ciclo de vida de parques eólicos offshore [2].

Domínio	Principais Atividades
<b>Operações</b>	Monitorização remota; previsão (produção, meteorologia, manutenção); otimização e manutenção preditiva; planeamento e agendamento; coordenação logística e marítima; integração na rede elétrica; formação e treino; resposta a emergências.
<b>Manutenção</b>	Inspeções (pás, torre, nacelle, fundações); manutenção planeada e não planeada; reparações (menores a maiores); diagnóstico e resolução de falhas; ferramentas e equipamentos especializados; segurança em operações no local; apoio de embarcações e transporte.
<b>Suporte</b>	Planeamento de fim de vida; desmantelamento; finanças e seguros; gestão de partes interessadas; considerações regulatórias; armazenamento de componentes críticos; gestão da cadeia de abastecimento.
<b>Investigação e Desenvolvimento (I&amp;D)</b>	Robótica e sistemas autónomos; inspeção e reparação remota; economia circular e reciclagem; exploração multiuso do espaço marítimo; descarbonização do setor marítimo.

Embora todos estes domínios sejam relevantes para o desempenho de longo prazo dos parques offshore, é nas atividades de **operação** e, sobretudo, de **manutenção** que se concentra a maior parcela dos custos. Ren et al. [1] indicam que as atividades de Operação e Manutenção (O&M) representam, em média, cerca de **23% do Levelized Cost of Energy (LCOE)** de um parque eólico offshore — um valor significativamente superior aos aproximadamente **5%** registados em parques onshore. Dentro deste bloco, a **manutenção** assume particular preponderância, absorvendo entre **60% e 75% dos custos de O&M**, o que corresponde a aproximadamente **14% a 17% do LCOE total**. Este diferencial justifica a crescente aposta em

soluções tecnológicas avançadas, como a utilização de UAVs, algoritmos de detecção precoce de falhas e sistemas autónomos de inspeção, que procuram otimizar a eficiência e reduzir o impacto económico da manutenção offshore.

## 1.2 Modelos de Manutenção em Parques Eólicos Offshore

A estratégia de manutenção adotada em parques eólicos offshore influencia diretamente o custo nivelado de energia (LCOE), a fiabilidade dos equipamentos e a eficiência global do sistema. Conforme descrito por Ren et al. [1], os modelos de manutenção podem ser classificados em três grandes grupos: **corretiva**, **proativa** e **oportunista**. A Figura 1.1 apresenta esta taxonomia.

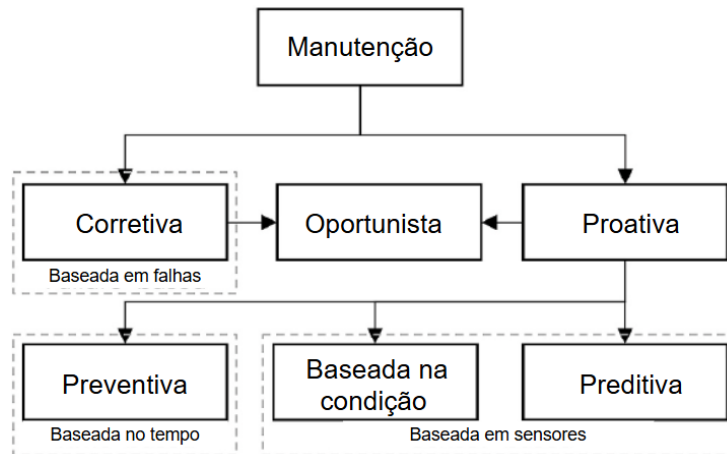


Figura 1.1: Classificação dos principais modelos de manutenção em parques eólicos offshore [1].

- **Manutenção corretiva (baseada em falhas)** — Também designada reativa, ocorre apenas após a falha de um componente ou subsistema. Apesar de exigir pouco planeamento inicial, está associada a elevados períodos de indisponibilidade, reparações urgentes e riscos operacionais acrescidos.
- **Manutenção proativa** — Recorre a técnicas de monitorização e prognóstico para antecipar falhas e otimizar a logística das intervenções. Subdivide-se em:
  - **Manutenção preventiva (baseada no tempo)**: calendarizada de acordo com a vida útil estimada dos componentes, reduzindo o risco de falhas críticas, embora possa originar substituições desnecessárias;
  - **Manutenção baseada na condição**: acionada por sensores que monitorizam parâmetros críticos (vibração, temperatura, ruído, etc.), permitindo intervenções quando é detetada degradação funcional;
  - **Manutenção preditiva**: baseada em algoritmos de prognóstico e inteligência artificial, capazes de antecipar falhas com base em padrões

históricos e tendências, possibilitando intervenções planeadas de forma otimizada.

- **Manutenção oportunista** — Explora janelas de oportunidade em que já existe presença no local (por exemplo, uma embarcação em operação), realizando múltiplas intervenções numa única deslocação e reduzindo custos logísticos.

A Figura 1.2 ilustra de forma comparativa o impacto dos diferentes modelos na disponibilidade dos aerogeradores e nos tempos de resposta operacional.

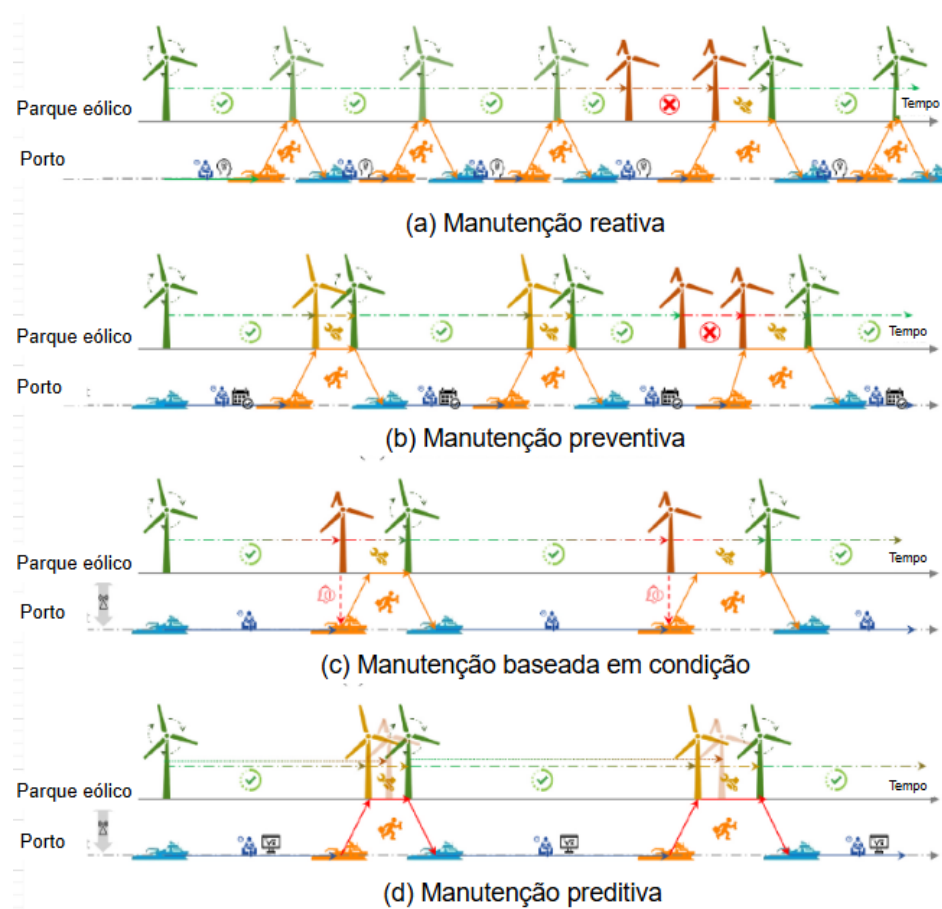


Figura 1.2: Comparação entre diferentes estratégias de manutenção. Verde: turbinas operacionais; vermelho: falha; amarelo: manutenção programada [1].

Como se observa, os modelos corretivos tendem a gerar longos períodos de indisponibilidade e custos elevados. Em contraste, as abordagens proativas — em particular a manutenção baseada na condição e a preditiva — asseguram maior fiabilidade operacional, reduzem o impacto económico associado a falhas inesperadas e contribuem para a diminuição do LCOE. Atualmente, verifica-se uma tendência clara

para a adoção de estratégias **proativas**, com especial destaque para a manutenção preditiva, potenciada pelos avanços em sensorização, *digital twins* e algoritmos de *Machine Learning* (ML). Estas abordagens alinham-se com os objetivos de maximização da disponibilidade e redução dos custos de operação, justificando o seu crescente interesse por parte de operadores de grande escala [1].

### 1.3 Logística e Níveis de Autonomia em UAVs para Inspeção Offshore

A logística associada à inspeção de turbinas eólicas offshore com veículos aéreos não tripulados (UAVs) está intimamente relacionada com o seu grau de autonomia, a distância dos parques em relação à costa e os métodos disponíveis para lançamento, recuperação e operação. À medida que os parques eólicos offshore avançam para localizações cada vez mais remotas e exigentes, torna-se necessário repensar as estratégias logísticas e operacionais tradicionalmente adotadas [4].

Atualmente, a maioria das missões com UAVs ocorre a partir de embarcações tripuladas, como os Crew Transfer Vessel (CTV) ou os Service Operation Vessel (SOV). Estas embarcações funcionam como base para o transporte de técnicos, ferramentas e UAVs, permitindo a descolagem manual nas proximidades das turbinas. Embora funcional, esta abordagem apresenta limitações claras: elevado custo operacional, elevada complexidade logística, exposição de operadores humanos a riscos e forte dependência de condições meteorológicas [4]. Para intervenções de maior escala, recorrem-se ainda a Heavy Lift Vessel (HLV) (*Heavy Lift Vessels*), também designados *jack-up vessels*, cuja utilização implica custos ainda mais elevados e restrições logísticas significativas.

A Figura 1.3 ilustra o exemplo típico de um parque eólico offshore flutuante, evidenciando a sua arquitetura física e os principais meios logísticos envolvidos.

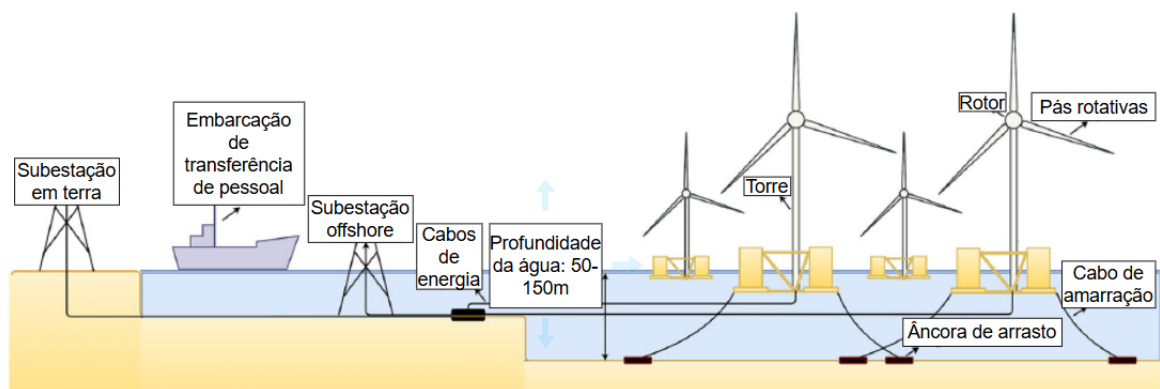


Figura 1.3: Esquema representativo de um parque eólico offshore flutuante [4].

Com o aumento das distâncias e a necessidade de reduzir custos operacionais, tem-se observado um interesse crescente em soluções mais autônomas e sustentáveis, como os Autonomous Surface Vessel (ASV) (*Autonomous Surface Vessels*) e os chamados **UAVs residentes** [5] — drones concebidos para permanecer em ambiente offshore durante longos períodos, operando a partir de plataformas flutuantes ou torres automatizadas. Apesar do seu elevado potencial, estas soluções encontram-se ainda em estágios experimentais, apresentando um nível de maturidade tecnológica reduzido (*Technology Readiness Level* (TRL)), o que limita, por enquanto, a sua aplicação em cenários comerciais de larga escala.

Neste enquadramento, assume particular relevância a análise do **nível de autonomia** dos UAVs, que permite avaliar até que ponto estes podem operar sem supervisão direta. Esta classificação afeta diretamente a capacidade de realizar voos para além da linha de visão (Beyond Visual Line of Sight (BVLOS)) e de responder a situações imprevistas [6]. Por contraste, as operações de linha de visão direta (Visual Line of Sight (VLOS)) limitam-se à observação visual contínua por parte do operador.

Segundo Mitchell et al. [6], os níveis de autonomia em UAVs aéreos podem ser organizados em seis categorias distintas, conforme sintetizado na Tabela 1.2.

Tabela 1.2: Classificação dos níveis de autonomia em UAVs aplicados à inspeção offshore [6].

Nível	Designação	Descrição e Características
0	Sem Autonomia	Teleoperação total; operação manual sob VLOS; controlo contínuo pelo operador humano.
1	Assistência Operacional	Voo manual com assistência básica (ex.: estabilização, retorno automático, travagem de emergência).
2	Automação Parcial	Execução de tarefas programadas sob supervisão contínua; operação semi-autónoma.
3	Operação Condicional	Execução autónoma de missões pré-planeadas em ambientes previsíveis; intervenção humana apenas em exceções.
4	Automação Elevada	Autonomia quase total; replaneamento dinâmico; operação segura em BVLOS com redundâncias de segurança.
5	Autonomia Total	Capacidade de operar em qualquer ambiente de forma totalmente autónoma; sem supervisão direta de operadores humanos.

A correta identificação do nível de autonomia de uma plataforma UAV é crucial para o planeamento das inspeções offshore, dado que impacta a sua viabilidade

regulatória, o custo de operação, a cobertura alcançável e a complexidade técnica. A transição de VLOS para BVLOS representa um marco estratégico, permitindo missões de longo alcance sem embarcações tripuladas. Contudo, a concretização de níveis mais elevados de autonomia exige avanços na navegação em ambientes não estruturados, na deteção robusta de obstáculos e na integração com sistemas logísticos autónomos, como os ASVs [4].

Assim, a evolução dos níveis de autonomia, combinada com o desenvolvimento de plataformas logísticas não tripuladas, abre caminho para uma nova geração de inspeções offshore mais seguras, eficientes e integralmente automatizadas.

## 1.4 Turbinas Eólicas Offshore

As turbinas eólicas offshore são instaladas em ambientes marinhos, sujeitas a condições geotécnicas e meteorológicas adversas. A escolha da subestrutura depende da profundidade da água e das características do solo marinho, sendo possível distinguir cinco tipos principais ilustrados na Figura 1.4. Estas soluções abrangem desde estruturas fixas até plataformas flutuantes, adaptando-se a diferentes profundidades e exigências operacionais [7].

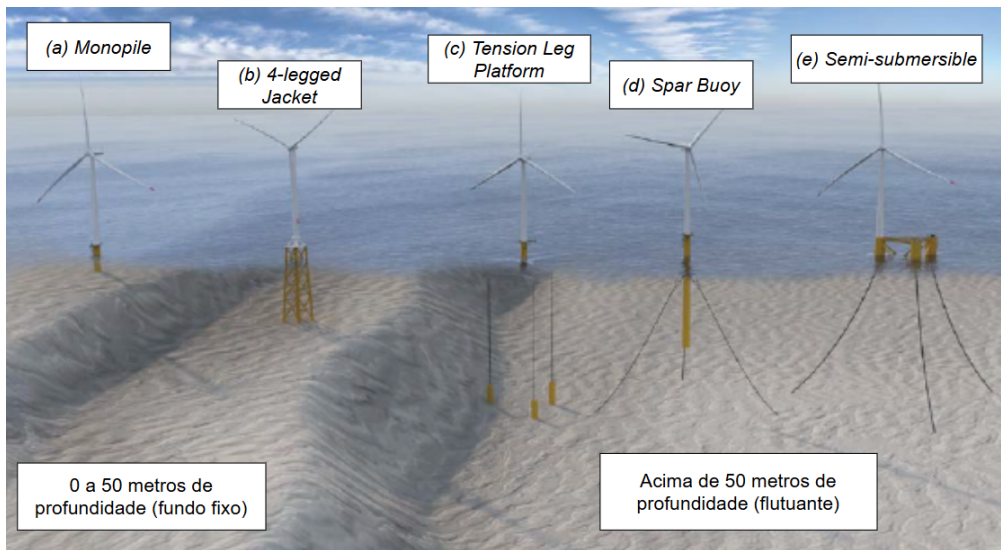


Figura 1.4: Diferentes tipos de subestruturas para turbinas eólicas offshore em função da profundidade [7].

A *Monopile* (a) é uma estrutura fixa, composta por um único pilar cravado no leito marinho, ideal para profundidades até 50 metros. A *4-legged Jacket* (b) utiliza uma estrutura de suporte de quatro pernas, também fixa, adequada a águas intermédias com maior estabilidade. A *Tension Leg Platform* (c) é uma plataforma flutuante ancorada por cabos tensionados, projetada para águas profundas acima de

50 metros. A *Spar Buoy* (*d*) consiste numa boia cilíndrica flutuante, estabilizada verticalmente por contrapesos, sendo outra solução para profundidades elevadas. Por fim, a *Semi-submersible* (*e*) é uma plataforma flutuante com múltiplos suportes submersos, oferecendo flexibilidade geográfica e escalabilidade em águas profundas [7].

### **Rotor**

O rotor, composto pelo *hub* e pelas pás, constitui o elemento mais dispendioso e vulnerável da turbina eólica, representando um investimento significativo devido à complexidade da sua conceção e fabricação. As pás, fabricadas tipicamente em materiais compósitos (Glass-Fiber Reinforced Polymer (GFRP)/Carbon-Fiber Reinforced Polymer (CFRP)), integram reforços internos que garantem rigidez e otimizam o desempenho aerodinâmico, sendo, no entanto, altamente suscetíveis a diversos tipos de danos. Estes incluem delaminações, fissuras, fraturas por fadiga, impactos de granizo, colisões com aves e descargas atmosféricas, o que as coloca como foco prioritário em programas de manutenção [8]. Dados indicam que as pás estão associadas a aproximadamente 19,4% dos incidentes de avarias reportados em turbinas eólicas, constituindo uma proporção crítica que reforça a necessidade de estratégias de monitorização rigorosas. Além disso, estes componentes representam cerca de 15-20% do custo total de fabrico, destacando a sua relevância económica e a urgência de minimizar falhas [8]. As pás apresentam duas superfícies principais, a superfície superior e a superfície inferior, que correspondem aos lados maiores e opostos da estrutura, sendo estas as áreas mais expostas aos elementos e, por conseguinte, as mais frequentemente inspecionadas. A borda de ataque, onde o ar incide diretamente, e a borda de fuga, por onde o ar é expelido, complementam estas superfícies, sendo igualmente suscetíveis a impactos e desgaste, mas é nas superfícies superior e inferior que se concentram os principais esforços de monitorização devido à sua extensão e exposição prolongada [9]. A substituição ou reparação em campo, especialmente em ambientes offshore, é extremamente dispendiosa, justificando o investimento em sistemas de deteção precoce de danos, como inspeção visual, termografia, ultrassons e sensores embutidos.

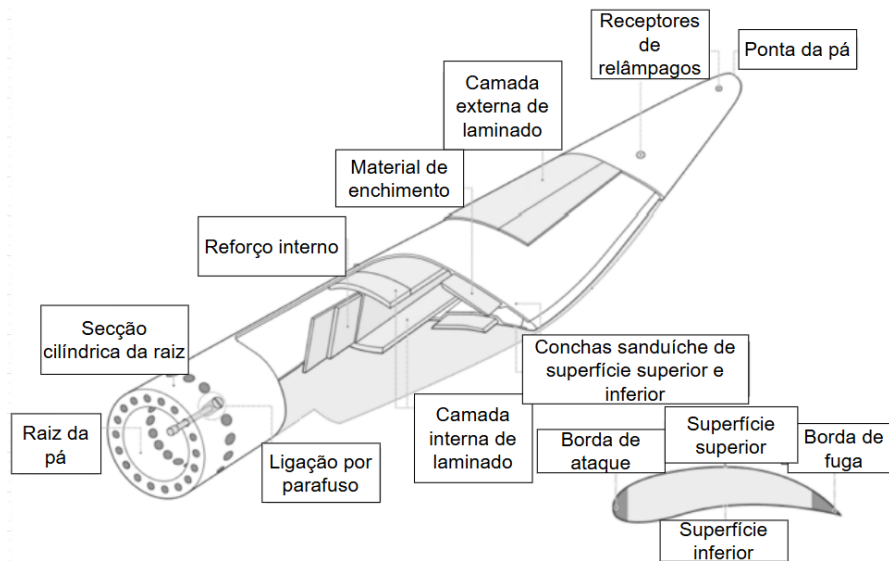


Figura 1.5: Componentes internos e externos típicos de uma pá de turbina eólica [9].

### ***Nacelle* e Sistema de Geração**

A *nacelle* constitui o núcleo central da turbina eólica, abrigando os principais sistemas responsáveis pela conversão e gestão da energia gerada. Este componente crítico, montado no topo da torre, integra uma série de elementos mecânicos e elétricos cuja performance influencia diretamente a eficiência e a longevidade da turbina. Entre os principais sistemas destacam-se:

- A caixa de velocidades, que desempenha um papel essencial na transmissão de energia do veio principal para o gerador, ajustando a rotação para níveis adequados à geração elétrica. Contudo, é frequentemente identificada como uma das fontes mais comuns de falhas no *drivetrain*, devido ao desgaste intensivo e às cargas variáveis impostas pelas condições ambientais.
- O gerador, em conjunto com os sistemas de controlo e eletrónica de potência, converte a energia mecânica captada pelas pás em energia elétrica, sendo um elemento fundamental para a produtividade da turbina. Estes sistemas asseguram a estabilização da saída de energia e a sua integração na rede elétrica, sendo suscetíveis a sobreaquecimento ou falhas eletrónicas em condições adversas.
- O sistema de *yaw* e o travão mecânico são cruciais para a orientação da *nacelle* face à direção do vento, garantindo a otimização da captura de energia, e para a segurança da turbina, permitindo a imobilização em situações de ventos extremos ou manutenção. A precisão do sistema de *yaw* é vital para minimizar

stresses estruturais, enquanto o travão mecânico atua como mecanismo de emergência.

- Os sistemas hidráulicos e de arrefecimento suportam a operação contínua dos componentes internos, dissipando o calor gerado pelo gerador e pela *gearbox*, além de lubrificar e manter a integridade dos rolamentos do veio principal, que suportam as cargas dinâmicas transmitidas pelo rotor.

A robustez e a manutenção proativa destes sistemas são determinantes para a fiabilidade global da turbina, especialmente em ambientes offshore, onde as condições marítimas acentuam o desgaste [9].

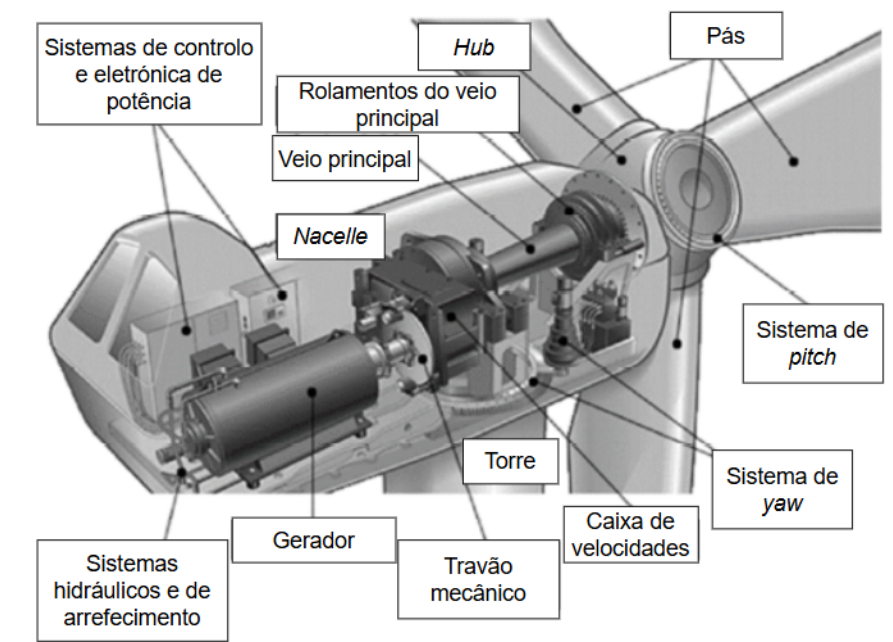


Figura 1.6: Componentes mecânicos e elétricos principais da *nacelle* [9].

### Dimensões

Nas últimas décadas, as turbinas eólicas têm vindo a registar um crescimento notável em termos de potência nominal e dimensões estruturais. Este incremento tem como objetivo maximizar a captação de energia e reduzir o custo nivelado de energia (LCOE), sendo particularmente relevante no contexto offshore, onde a maior regularidade dos ventos justifica o investimento em turbinas de grande escala [8], [10].

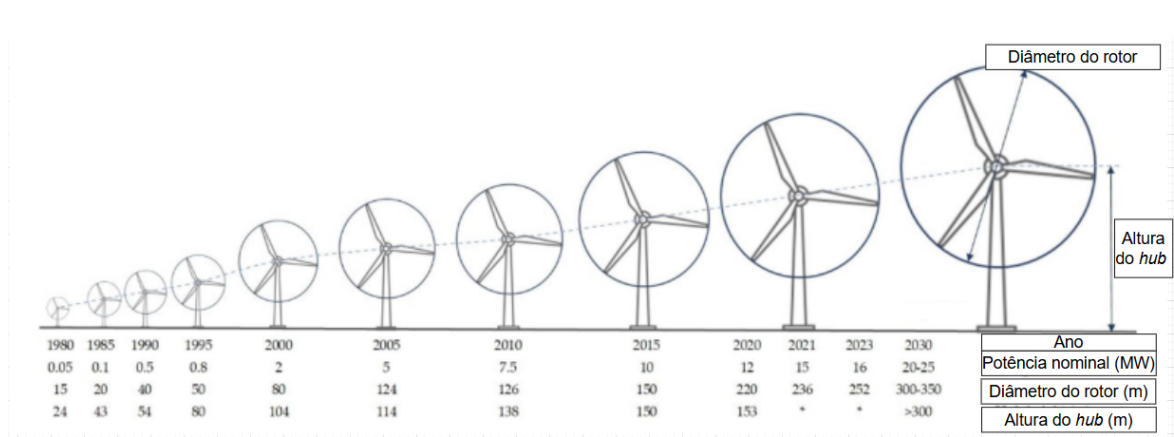


Figura 1.7: Evolução histórica e projeções futuras das dimensões das turbinas eólicas [8].

Entre 1980 e 2023, a potência nominal evoluiu de 0,05 MW para 16 MW, com projeções que indicam o desenvolvimento de turbinas com potências entre 20 e 25 MW até 2030. Paralelamente, o diâmetro do rotor aumentou de 15 m para valores superiores a 250 m, enquanto a altura do *hub* passou de 24 m para mais de 150 m, com perspectivas de atingir ou ultrapassar 300 m em projetos futuros. Apesar das vantagens energéticas associadas a estas dimensões, emergem desafios significativos na operação e manutenção (O&M), especialmente em ambientes remotos e sujeitos a condições adversas, como corrosão marítima e acessibilidade limitada [8].

## 1.5 Definição do Problema

A inspeção de turbinas eólicas *offshore* continua a ser um desafio complexo, especialmente no que respeita às pás, que são estruturas longas, expostas a condições ambientais adversas e suscetíveis a diferentes tipos de danos. Tradicionalmente, estas inspeções dependem de intervenções humanas, com utilização de embarcações de suporte, guindastes ou plataformas elevatórias, expondo os operadores a riscos significativos e gerando custos elevados.

Apesar do potencial dos veículos aéreos não tripulados (UAVs) para reduzir riscos e aumentar a eficiência, verificam-se várias lacunas na literatura e na prática:

- **Falta de soluções integradas e acessíveis:** não existem plataformas open-source consolidadas que permitam testar de forma segura e sistemática técnicas de inspeção autónoma em UAVs, integrando perceção, planeamento de trajetória e controlo de voo;
- **Dificuldade em validar técnicas sem hardware real:** o elevado custo, risco e complexidade logística associados à utilização de UAVs em ambiente offshore limitam a possibilidade de experimentação direta com hardware, tornando difícil avaliar a viabilidade de novas abordagens;

- **Desafios de percepção e navegação:** a detecção precisa das pás, a interpretação do seu estado e a navegação próxima a estas estruturas longas e rotativas representam obstáculos técnicos ainda por superar, sobretudo em condições de vento e mar agitado.

### 1.5.1 Objetivos

O objetivo central desta dissertação é propor uma metodologia para a inspeção de turbinas eólicas *offshore* com UAVs residentes, estruturando o processo desde a aproximação às pás até à coordenação multi-UAV. Para tal, definem-se os seguintes objetivos específicos:

- Desenvolver um ambiente de simulação realista que permita reproduzir cenários típicos de operação em parques eólicos offshore;
- Configurar UAVs virtuais com sensores identificados como relevantes na literatura, assegurando deteção e caracterização fiável das turbinas;
- Conceber e implementar algoritmos de percepção, controlo e planeamento de trajetória, integrados num pipeline de inspeção autónoma;
- Explorar, ainda que de forma preliminar, estratégias de inspeção multi-UAV que evidenciem o potencial da colaboração entre veículos;
- Avaliar o desempenho da solução proposta em métricas de robustez, eficiência e aplicabilidade em cenários reais.

### 1.5.2 Resultados Esperados

Com este trabalho espera-se demonstrar que é possível construir e validar, em simulação, um sistema de inspeção autónoma de turbinas offshore baseado em UAVs residentes. Pretende-se que os principais resultados incluam:

- Um **pipeline funcional e integrado** que combine percepção multissensorial, planeamento e controlo;
- A **demonstração experimental em simulação** da viabilidade do sistema, reduzindo custos e riscos associados a testes com *hardware* real;

## 1.6 Plano de Trabalho

O plano de trabalho desta dissertação foi estruturado em fases sequenciais, de forma a garantir uma progressão lógica desde a análise do estado da arte até à avaliação final das estratégias estudadas. Esta abordagem permite organizar o processo de investigação, reduzir redundâncias e assegurar que cada etapa sustenta a seguinte.

As fases principais são as seguintes:

1. **Análise do Estado da Arte:** Revisão das soluções existentes para inspeção de turbinas eólicas offshore com UAVs, incluindo técnicas de percepção, navegação e controlo. Esta fase permite identificar lacunas e oportunidades de investigação.
2. **Criação do Ambiente de Simulação:** Desenvolvimento de um ambiente virtual que reproduza as turbinas e os UAVs, incluindo a configuração de sensores representativos, de modo a possibilitar testes seguros e controlados.
3. **Testes de Técnicas de Percepção e Controlo:** Avaliação de diferentes abordagens de percepção e controlo para a inspeção autónoma das turbinas, permitindo analisar a capacidade do sistema em detectar e responder às condições do ambiente.
4. **Exploração de Estratégias de Cooperação Multi-UAV:** Investigação de possíveis métodos de coordenação entre vários UAVs, com o objetivo de melhorar a cobertura e a eficiência da inspeção.
5. **Avaliação e Análise dos Resultados:** Verificação da viabilidade das estratégias propostas, análise do desempenho das técnicas estudadas e identificação de direções para investigações futuras.

Este plano procura equilibrar a vertente científica, através da análise do estado da arte e da definição de estratégias fundamentadas, com a vertente prática, assegurando que a investigação se mantém focada na resolução do problema identificado e na avaliação das suas implicações.

## 1.7 Organização da Dissertação

Esta dissertação encontra-se estruturada de forma a guiar o leitor desde o contexto da investigação até às conclusões obtidas. O Capítulo 1 apresenta a introdução, contextualizando a importância da inspeção de turbinas eólicas *offshore* e os desafios associados, assim como os objetivos gerais e específicos do trabalho.

O Capítulo 2 aborda o estado da arte, descrevendo as diferentes soluções existentes para inspeção de turbinas, incluindo métodos tradicionais e técnicas com UAVs, bem como as vantagens e limitações de cada abordagem. No final deste capítulo é feita uma síntese crítica, que orienta a seleção de sensores, algoritmos e técnicas a implementar na metodologia proposta.

O Capítulo 3 apresenta a fundamentação teórica, detalhando os conceitos, algoritmos e técnicas utilizados na implementação, fornecendo a base técnica para o desenvolvimento da solução proposta.

O Capítulo 4 descreve a implementação da metodologia, detalhando a configuração dos UAVs, sensores, processamento de dados e coordenação multi-UAV.

No Capítulo 5 são apresentados os resultados obtidos, incluindo análises de desempenho, métricas de voo, inspeção das pás e avaliação da eficiência da abordagem multi-UAV.

Por fim, o Capítulo 6 apresenta as conclusões, refletindo sobre os principais contributos do trabalho, limitações identificadas e possíveis direções para trabalhos futuros.



## Capítulo 2

# Estado da Arte

O presente capítulo apresenta uma revisão do estado da arte sobre a inspeção de turbinas eólicas offshore com recurso a UAVs, com foco em publicações científicas e soluções comerciais datadas de apartir de 2015. Foram analisados artigos, conferências e documentação técnica que abordam diferentes aspetos da operação autónoma de drones em ambientes offshore.

A revisão organiza-se em duas grandes vertentes. A primeira aborda a **otimização global**, centrando-se em estratégias de planeamento e coordenação de múltiplos UAVs, bem como na logística associada às missões. A segunda foca-se na **inspeção local**, explorando métodos de perceção, controlo e navegação para efetuar inspeções autónomas das turbinas, especialmente das pás.

O capítulo termina com uma síntese que integra os principais resultados encontrados, identifica lacunas na literatura e evidencia oportunidades para investigação futura, servindo de base para a implementação adotada nesta dissertação.

### 2.1 Otimização Global

Chung et al. [11] exploram o **dimensionamento e posicionamento ótimo dos UAVs**, resolvendo problemas de alocação e planeamento de rotas com o objetivo de minimizar tanto o número de UAVs necessários como o tempo total da missão. A abordagem foca-se na distribuição eficiente dos veículos aéreos sobre a área de inspeção, considerando restrições de cobertura e autonomia.

Fontenla-Carrera et al. [12] estendem esta linha de investigação propondo um modelo **UAV–embarcação**, no qual os drones partem de um CTV ou embarcação similar. Este conceito permite ultrapassar limitações de autonomia e reforça a conectividade, mas depende fortemente da disponibilidade e posição da embarcação, podendo ser menos flexível em operações prolongadas ou em parques dispersos.

Noutra perspetiva, Yan et al. [13] apresentam uma abordagem de **alocação de tarefas multi-UAV** utilizando algoritmos híbridos de otimização combinando heurísticas e métodos exatos. Esta solução é adequada a cenários dinâmicos, onde zonas de risco ou mudanças inesperadas podem ocorrer, permitindo redistribuir as tarefas entre UAVs de forma adaptativa.

Fan et al. [14] focam-se em modelar de forma realista o **consumo energético e a gestão de baterias**, propondo heurísticas evolutivas para otimizar a distribuição de tarefas e o recarregamento entre múltiplos UAVs.

Baik e Valenzuela [15] propõem uma solução **híbrida UAV + veículo terrestre**, adequada sobretudo a parques onshore, em que UAVs e veículos terrestres colaboram para inspecionar diferentes partes das turbinas, podendo este conceito ser adaptado a operações offshore com as devidas modificações logísticas.

Ismail et al. [16] introduzem a coordenação de *Unmanned Surface Vehicle (USV)* e UAVs, utilizando embarcações autónomas como plataformas móveis de recarga. Esta abordagem permite maior autonomia e tempo de voo prolongado, mas depende de controlo marítimo avançado e de sincronização precisa entre veículos aéreos e aquáticos.

Finalmente, a proposta *Lilypad* da Marshall Futureworx [5] representa um salto conceptual, ao defender um ecossistema de **UAVs residentes com estações fixas junto às turbinas**, eliminando a necessidade de embarcações de apoio e permitindo inspeções contínuas e autónomas. Este modelo garante maior disponibilidade e eficiência, especialmente em parques offshore dispersos, mas implica investimento inicial elevado em infraestrutura e manutenção das estações residentes.

## 2.2 Inspeção Local

### 2.2.1 Técnicas de Inspeção na Manutenção Proativa Offshore

A inspeção regular e eficiente constitui um pilar essencial da **manutenção proativa** em turbinas eólicas offshore, permitindo antecipar falhas, reduzir períodos de indisponibilidade e maximizar a disponibilidade dos ativos [17].

As abordagens atualmente aplicadas em ambiente offshore podem ser agrupadas em seis categorias principais:

- **Inspeção manual** — Envolve a deslocação de técnicos ao local, apoiados por plataformas ou cordas. Embora seja a forma mais tradicional de inspeção, é dispendiosa, arriscada e logisticamente pouco eficiente.
- **Sensores permanentes** — Consistem em sensores embutidos (ex.: vibração, temperatura, tensão) que monitorizam continuamente componentes críticos. Apesar de possibilitarem a deteção precoce de anomalias, oferecem cobertura limitada e exigem integração desde a fase de projeto.
- **ROVs (*Remotely Operated Vehicles*)** — Utilizados sobretudo para inspeção subaquática de fundações, linhas de amarração e cabos, fornecendo dados visuais e sonar de elevada precisão. São eficazes, mas implicam logística especializada e operação embarcada [4].
- **Robôs trepadores** — Aplicados em torres e pás, deslocam-se autonomamente pela estrutura recolhendo dados visuais, térmicos ou ultrassónicos. Apesar da elevada precisão, enfrentam desafios técnicos relacionados com aderência, vento lateral e geometrias complexas [4].
- **UAVs (*Unmanned Aerial Vehicles*)** — Representam uma solução eficiente, versátil e de baixo custo. São amplamente utilizados na inspeção visual e termográfica de pás, torres e nacelles, podendo ser equipados com câmaras Red Green Blue (RGB), Light Detection and Ranging (LiDAR) e sensores térmicos. Destacam-se pela rapidez de mobilização, pela segurança e pela integração com algoritmos de deteção automática [4], [17].
- **Sistemas híbridos e autónomos** — Combinam UAVs, ROVs, robôs terrestres e, cada vez mais, ASVs (*Autonomous Surface Vessels*). Estes últimos emergem como plataformas complementares para batimetria, apoio logístico e posicionamento dinâmico, reduzindo a dependência de embarcações tripuladas [4].

A Tabela 2.1 apresenta uma síntese comparativa destas abordagens, integrando dados de Zhang et al. [17] e Khalid et al. [4].

Tabela 2.1: Comparação entre diferentes métodos de inspeção [4], [17].

Método	Tipo de Sensor	Custo	Cobertura	Autonomia
Inspeção manual	Visual	Alto	Média	Baixa
Sensores permanentes	Vibração, Temp., Tensão	Médio	Baixa	Alta (contínua)
ROVs	Visual, Sonar	Alto	Submarina	Média
Robôs trepadores	Visual, US, AE	Médio-Alto	Alta (vertical)	Baixa
UAVs	RGB, LiDAR, Térmica	Baixo	Alta (aérea)	Média
Sistemas híbridos / ASVs	Combinados	Alto	Muito alta	Alta

De forma geral, reconhece-se que **sistemas híbridos**, que integram UAVs, ROVs, robôs trepadores e veículos de superfície autônomos, oferecem o maior potencial em termos de cobertura, automação e versatilidade. No entanto, considerando limitações de orçamento, tempo ou logística, os **UAVs** mantêm-se como a solução isolada mais eficiente, ao aliarem **ampla cobertura visual, baixo custo, mobilidade elevada e integração com algoritmos de detecção**, consolidando-se como a técnica dominante nas abordagens atuais de manutenção proativa offshore [4], [17].

### 2.2.2 Tipos de UAVs para Inspeção de Turbinas Eólicas Offshore

A seleção do tipo de UAV a utilizar em missões de inspeção offshore influencia diretamente a qualidade dos dados recolhidos, a eficiência da operação e os requisitos de segurança. Conforme analisado por Heo e Na [18] e complementado por Shafiee et al. [19], os UAVs aplicados a este domínio dividem-se em quatro categorias principais: **multirotor, asa fixa, híbrido** e **rotor único**, conforme ilustrado na Figura 2.1.

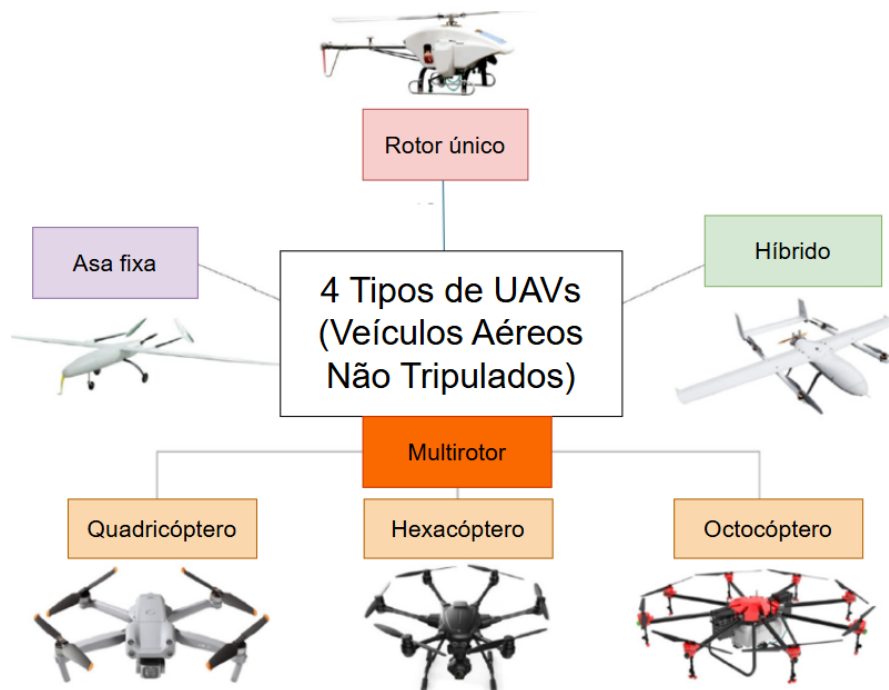


Figura 2.1: Classificação dos principais tipos de UAVs [18], [19].

- **Multirotores** — Englobam quadricópteros, hexacópteros e octocópteros. Destacam-se pela capacidade de pairar (*hovering*), elevada manobrabilidade e simplicidade de operação em ambientes confinados, sendo a escolha predominante para inspeções de proximidade, como pás, torres e *nacelles*. As principais limitações incluem a reduzida autonomia de voo (geralmente entre 20 e 30 minutos) e a sensibilidade a condições de vento adverso.
- **Asa fixa** — Funcionam com superfícies sustentadoras fixas, requerendo movimento contínuo para gerar sustentação. São ideais para missões de longa distância e cobertura de grandes áreas, beneficiando de uma elevada eficiência energética. Contudo, a ausência de capacidade de pairar restringe a sua utilização em inspeções detalhadas de proximidade.
- **Híbridos** — Integram as vantagens dos multirotores, como a descolagem e aterragem vertical, com as características das asas fixas, possibilitando voo de cruzeiro eficiente. Esta versatilidade torna-os adequados a uma ampla gama de cenários operacionais, embora se associem a maior complexidade técnica e custos operacionais mais elevados.
- **Rotor único** — Conforme descrito por Shafiee et al. [19], utilizam um rotor principal e um rotor de cauda para controlo direcional. Sobressaem pela capacidade de pairar com cargas úteis elevadas, sendo adequados para aplicações exigentes. Apesar da boa autonomia e desempenho de voo, requerem maior

perícia operacional, apresentam custos superiores e implicam riscos acrescidos devido ao tamanho das lâminas.

A Tabela 2.2 sintetiza as principais características comparativas entre estas quatro categorias de UAVs.

Tabela 2.2: Comparação entre os principais tipos de UAVs [18], [19].

<b>Critério</b>	<b>Multirotor</b>	<b>Asa fixa</b>	<b>Híbrido</b>	<b>Rotor único</b>
Capacidade de pairar	Sim	Não	Sim	Sim
Precisão de manobra	Alta	Baixa	Média	Média
Autonomia de voo	Baixa	Alta	Média	Alta
Cobertura de área	Média	Alta	Alta	Média
Custo e complexidade	Baixo	Médio	Elevado	Elevado
Inspeção detalhada	Excelente	Fraca	Boa	Boa
Carga útil	Baixa	Média	Média	Alta

Com base na análise comparativa e nas tendências atuais da indústria, os **UAVs multirotores** afirmam-se como a solução dominante em inspeções de proximidade de turbinas eólicas offshore. A sua capacidade de pairar, elevada manobrabilidade e compatibilidade com múltiplos sensores (RGB, câmaras térmicas e LiDAR) tornam-nos ideais para missões que exigem detalhe visual, rapidez de mobilização e flexibilidade operacional. Apesar da limitação em termos de autonomia, oferecem a melhor relação entre **custo, precisão e versatilidade**, sendo amplamente adotados em contextos offshore [18], [19].

### 2.2.3 Sensores Integrados nos UAVs

A eficácia da inspeção de turbinas eólicas offshore com recurso a UAVs depende diretamente da qualidade e versatilidade dos sensores **integrados** (*payloads*) nestes veículos. Conforme discutido por Zhang et al. [17] e reforçado por Márquez e Chacón [20], a combinação de diferentes sensores permite a aquisição simultânea de múltiplos tipos de dados físicos, estruturais e visuais, aumentando significativamente a precisão, a cobertura e a robustez do processo de inspeção.

Os principais sensores e respetivas funções podem ser descritos da seguinte forma:

- **Câmaras RGB** — Capturam imagens visuais de alta resolução, possibilitando inspeções superficiais, detecção de fissuras e documentação do estado estrutural. UAVs equipados com câmaras panorâmicas e zoom ótico são amplamente utilizados em inspeções rápidas e de baixo custo [17].
- **Termografia** — Aplicada em inspeções não destrutivas (*Non-Destructive Testing* (Inspeção Não Destrutiva) (NDT)), permite identificar anomalias internas, como delaminações e fissuras ocultas, através da análise da distribuição de calor à superfície. Segundo Márquez e Chacón [20], técnicas de termografia ativa com laser, combinadas com algoritmos avançados de visualização, revelam elevado potencial para detecção remota em condições operacionais.
- **LiDAR** — Geram modelos tridimensionais precisos das turbinas, possibilitando análise geométrica, detecção de deformações e suporte ao planeamento de trajetórias. UAVs com LiDAR têm sido integrados em sistemas de mapeamento e localização simultâneos (Simultaneous Localization and Mapping (SLAM)), assegurando reconstruções digitais das pás com elevada fidelidade [17].
- **Sensores ultrassônicos** — Tradicionalmente utilizados em medições de espessura e detecção de falhas internas, requerem contacto físico com a superfície. Contudo, Márquez e Chacón [20] destacam avanços recentes em sensores ultrassônicos sem contacto, baseados em laser, acoplados a UAVs.
- **Sistemas multissensoriais** — Combinam câmaras Red Green Blue - Depth (RGB-D), LiDAR, Inertial Measurement Unit (IMU), sensores de fluxo ótico e Global Navigation Satellite System (GNSS), permitindo navegação precisa, mapeamento 3D e aquisição de dados integrados numa única missão. Márquez e Chacón [20] sublinham a crescente incorporação de algoritmos de Inteligência Artificial (IA) para fusão de dados e suporte à decisão autónoma.

A Tabela 2.3 apresenta uma síntese comparativa das principais tecnologias de sensores integrados nos UAVs, considerando o tipo de dados, a necessidade de contacto, a complexidade de integração e o nível de maturidade tecnológica.

Tabela 2.3: Comparação dos principais sensores [17], [20].

Sensor	Tipo de dado	Contacto	Complexidade	Maturidade
Câmara RGB	Imagem 2D (visual)	Não	Baixa	Alta
Termografia	Imagem térmica	Não	Média	Alta
<i>LiDAR</i>	Nuvem de pontos 3D	Não	Alta	Alta
Ultrassónico	Sinal acústico	Sim / Parcial	Alta	Baixa
Sistema multissensorial	Dados combinados ( <i>RGB, LiDAR, GNSS</i> )	Não	Muito alta	Média

A adoção de **sistemas multissensoriais** tem vindo a ganhar destaque na inspeção de turbinas eólicas offshore, devido à sua capacidade de proporcionar avaliações mais completas e autónomas, reduzindo o número de voos necessários. As câmaras visuais continuam a ser amplamente utilizadas, dada a sua acessibilidade, baixo custo e facilidade de operação, sendo frequentemente a solução escolhida para inspeções preliminares. No entanto, face às exigências crescentes dos ambientes offshore, os sistemas multissensoriais apresentam-se como a abordagem mais promissora, oferecendo maior precisão, integração de dados diversificados e robustez em condições adversas, consolidando-se como uma estratégia eficaz para otimizar a manutenção de turbinas eólicas em contextos desafiantes [17], [20].

#### 2.2.4 Métodos Visuais

Em [21], é apresentado um sistema de inspeção autónoma baseado num controlador preditivo não linear orientado por informação visual, designado Visual Tracking Nonlinear Model Predictive Control (VT-NMPC).

Ao contrário de métodos convencionais de Model Predictive Control (MPC), que seguem trajetórias pré-definidas, o VT-NMPC ajusta continuamente a posição e a atitude do UAV para manter a superfície da pá centrada no campo de visão e a uma distância de referência. O processo combina construção de uma malha triangular simplificada da turbina e cálculo de uma sequência de pontos de inspeção resolvendo um Traveling Salesman Problem (TSP) de VT-NMPC para gerar sinais de controlo em tempo real, assegurando a manutenção da distância de segurança e do ângulo adequado de inspeção.

Os resultados em simulação e em ensaios experimentais indicam que esta abordagem permite manter a segurança do voo, garantir elevada cobertura e assegurar a

qualidade das imagens, quando comparada com Nonlinear Model Predictive Control (NMPC) e Perception-Aware Model Predictive Control (PAMPC).

Numa outra linha de investigação, é proposto um esquema de planeamento de cobertura colaborativa designado Collaborative Coverage Path Planning (C-CPP) (*Collaborative Coverage Path Planning*), concebido para coordenar múltiplos UAVs na inspeção de estruturas de grande escala [22]. O método baseia-se na decomposição da geometria da infraestrutura em fatias horizontais, obtendo regiões de interesse que são atribuídas a diferentes UAVs, de modo a assegurar a cobertura completa da superfície com sobreposição de campos de visão e distâncias de segurança adequadas. Esta abordagem foi posteriormente validada em cenários reais de inspeção de turbinas eólicas [23], onde múltiplos UAVs do tipo AscTec NEO foram utilizados para a recolha de dados visuais. Foram usadas diversas câmaras (*Visual-Inertial*, GoPro e PlayStation Eye) para obtenção de imagens de alta resolução, enquanto a localização foi garantida através da fusão de dados de Ultra-Wideband (UWB) e IMU, processados num Extended Kalman Filter (EKF), devido às limitações do *Global Positioning System* (GPS) em torno da torre. O controlo do voo foi realizado com base em MPC linear implementado em Robot Operating System (ROS), permitindo compensar perturbações externas. As missões incluíram a inspeção circular da torre (8–24 m) e semicircular das pás (30–45 m), alcançando erros médios de seguimento de aproximadamente 0,55 m para a torre e 1,36 m para as pás. Os resultados demonstraram a viabilidade do sistema, com reconstrução tridimensional parcial da turbina e imagens adequadas para avaliação estrutural, embora limitadas nas pás devido à baixa textura da superfície.

Uma outra contribuição relevante no domínio dos métodos visuais é apresentada em [24], onde é desenvolvido um módulo de navegação autónoma destinado a posicionar o UAV em frente ao hub da turbina eólica, constituindo uma etapa preliminar antes da inspeção detalhada. O sistema baseia-se exclusivamente em informação visual para reconhecer os elementos principais da turbina (torre, hub e pás), sem necessidade de integração com sistemas externos ou medições de elevada precisão do GNSS.

A abordagem recorre ao *Canny edge detector* e à transformada de Hough para extrair as linhas dominantes da imagem, correspondentes à torre e às pás. A interseção das linhas permite estimar a posição do hub, enquanto a largura aparente da torre é utilizada para calcular a distância relativa do UAV. A orientação das pás é igualmente inferida pela simetria angular de 120°, característica da sua configuração estrutural. Para robustez, as estimativas são filtradas por meio de um Kalman Filter (KF), permitindo compensar falhas temporárias de deteção ou condições visuais adversas.

O sistema foi implementado em tempo real utilizando uma câmara GoPro acoplada a um hexacóptero e processado numa Single-Board Computer (SBC) do tipo

PandaBoard. Ensaios realizados num parque eólico em Bessakerfjellet, Noruega, demonstraram que o algoritmo operava a uma frequência entre 7–11 Hz, sendo capaz de fornecer em voo as estimativas necessárias para alinhar autonomamente o UAV com a turbina e iniciar a inspeção. Embora não realize a inspeção propriamente dita, esta solução constitui um contributo relevante para a fase de navegação e posicionamento inicial em missões de inspeção visual.

Em [25] é proposta uma abordagem de inspeção visual que combina, de forma conjunta, a estimação da pose do UAV com a modelação paramétrica da turbina eólica durante o voo. O sistema tem como objetivo aumentar a consistência da localização relativa e, em paralelo, ajustar os parâmetros geométricos da turbina de modo a garantir uma correspondência próxima entre o modelo e a estrutura real.

A turbina é representada por um modelo esquelético minimalista, composto pela base, topo da torre, nacelle, hub e pás. Este modelo é parametrizado por grandezas como altura da torre, comprimento da nacelle e ângulo/comprimento das pás, constituindo uma representação compacta que facilita a otimização não linear. O problema é formulado como um grafo de poses tridimensional, onde as medições de GPS e IMU são usadas como restrições suaves, enquanto as correspondências visuais assumem um papel central na correção da trajetória estimada.

As imagens capturadas pela câmara RGB são processadas através de uma Convolutional Neural Network (CNN), responsável por detetar a estrutura da turbina e projetar o modelo esquelético no plano da imagem. A correspondência é estabelecida entre os pontos e linhas do modelo projetado e as detecções visuais. São definidas duas funções de custo: (i) uma baseada em correspondências geométricas tradicionais (2D), e (ii) uma baseada na interpolação direta das probabilidades fornecidas pela rede neuronal em forma de *heatmaps*. A otimização conjunta é resolvida por mínimos quadrados não lineares com recurso ao Ceres Solver, permitindo atualizar simultaneamente a pose do UAV e os parâmetros do modelo da turbina.

A validação foi realizada tanto em simulação como com dados de voos reais de inspeção. Nos cenários sintéticos, a abordagem demonstrou reduções significativas no erro de localização e de modelação em comparação com estratégias que estimavam apenas a pose. Em dados reais, verificou-se que as reprojeções do modelo ajustado se alinhavam de forma mais consistente com as imagens capturadas, confirmando a robustez do método. O sistema foi implementado em CPU (Intel i7) com uma taxa de atualização próxima de 5 Hz e em plataforma embarcada Jetson TX2 a aproximadamente 1 Hz, revelando-se viável para operação em tempo real, embora com limitações de frequência de execução.

Em [26] é proposta uma solução baseada exclusivamente em visão computacional para a deteção e posicionamento das pontas das pás de turbinas eólicas, tendo em vista a navegação autónoma de UAVs em missões de inspeção. O trabalho parte da premissa de que sensores como LiDAR, sonar ou radar apresentam custos elevados e

suscetibilidade a ruído em cenários offshore, pelo que a utilização de câmaras RGB se torna uma alternativa mais acessível e robusta.

A abordagem recorre a uma rede neuronal do tipo Mask Region-Based Convolutional Neural Network (Mask R-CNN), com backbone ResNet50 pré-treinado no *ImageNet*, para detetar os principais elementos da turbina (torre, hub, pás e respetivas pontas). Para reduzir erros de classificação, foram introduzidas restrições de forma (*shape constraints*) que permitem distinguir entre estruturas semelhantes, como por exemplo entre a torre e uma pá em posição vertical. Após a deteção, a posição da ponta é refinada com recurso ao detetor de cantos Features from Accelerated Segment Test (FAST), aplicado em janelas locais de  $20 \times 20$  píxeis, obtendo uma localização de precisão subpíxel.

As coordenadas das pontas das pás são então calculadas em espaço tridimensional através da resolução do problema de Perspective-n-Point (PnP), utilizando os parâmetros intrínsecos da câmara, a altura da torre, o comprimento das pás e a posição GPS da turbina como referências geométricas. O resultado é a determinação das coordenadas geográficas (latitude, longitude, altitude) das pontas, informação essencial para o planeamento de trajetórias de inspeção.

Para validar a proposta, os autores compilaram uma base de dados própria, designada Wind Turbine Detection Dataset (WTD), composta por 584 imagens 4K de turbinas eólicas obtidas com um UAV DJI Phantom 4 Pro. As anotações incluem não apenas caixas horizontais (Horizontal Bounding Box (HBB)), mas também caixas rotacionadas (Rotated Bounding Box (RBB)), que representam de forma mais realista a geometria das pás. Os resultados experimentais evidenciaram que a Mask R-CNN superou outras arquiteturas de referência, como Faster R-CNN, Cascade R-CNN e RetinaNet, atingindo uma precisão média de 0,989 ( $\text{IoU} > 0,5$ ) e um erro médio de 2,27 píxeis na localização das pontas. Em testes em hardware embarcado Jetson TX2, o sistema alcançou aproximadamente 1,46 s por imagem a 720p, próximo de tempo real. A validação em campo, realizada com recurso a uma estação total, confirmou que as coordenadas obtidas eram suficientemente precisas para suportar a navegação autónoma dos UAVs durante a inspeção.

### 2.2.5 Métodos Multissensoriais

Claro et al. [27] apresentam um método de planeamento de trajetória para inspeções aéreas tridimensionais com foco na eficiência energética. A abordagem propõe a formulação do problema de planeamento como uma variante do Asymmetric Traveling Salesman Problem (ATSP), designada Asymmetric Traveling Salesman Problem with Precedence Loss (ATSP-PL) (*Asymmetric Travelling Salesman Problem with Precedence Loss*), onde o custo associado à transição entre pontos de inspeção depende não apenas da distância, mas também do movimento anterior do UAV, refletindo o impacto das acelerações e desacelerações no consumo energético.

O algoritmo desenvolvido, denominado Energy Efficient Path Planning Algorithm (EEPPA) (*Energy Efficient Path Planning Algorithm*), combina heurísticas de inicialização, resoluções do ATSP clássico e técnicas de melhoria local, procurando reduzir o gasto energético total durante a missão de inspeção. Para estimar este consumo, foi integrado um modelo energético do UAV que considera deslocamentos lineares, ascensões, descidas, rotações e variações de velocidade.

A solução contempla diferentes sensores, incluindo câmaras monoculares (RGB) e LiDAR 3D. O processo de definição de pontos de vista de inspeção baseia-se nas características de ambos os sensores, nomeadamente campo de visão, resolução e frequência de aquisição de dados. Esta configuração permite recolher simultaneamente informação visual e dados volumétricos da turbina, possibilitando a reconstrução tridimensional da estrutura.

A validação foi realizada em dois contextos distintos. Em ambiente de simulação no *Gazebo*, foram modeladas turbinas eólicas completas, permitindo testar o EEPPA em cenários realistas de inspeção. Os resultados mostraram reduções na ordem dos 25–28% no consumo energético quando comparado com métodos de planeamento convencionais, assegurando cobertura adequada da superfície inspecionada. Em ambiente real, o algoritmo foi testado com o UAV CROW, desenvolvido no Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC), tendo como alvo experimental uma torre de iluminação. Este ensaio confirmou a aplicabilidade prática do método, registando ganhos próximos de 20% na eficiência energética, ao mesmo tempo que garantiu a aquisição consistente de dados visuais e tridimensionais.

Oliveira et al. [28] apresentam um sistema autónomo de inspeção de turbinas eólicas offshore com recurso a LiDAR, que constitui um pipeline completo de perceção, modelação e recolha de dados. O objetivo é permitir que o UAV identifique e modele autonomamente a estrutura da turbina, assegurando a manutenção de distâncias de segurança e a aquisição de informação multimodal relevante para a deteção de defeitos.

A arquitetura proposta organiza-se em dois pipelines principais. O primeiro corresponde ao pipeline de perceção e modelação, no qual os dados adquiridos por um sensor LiDAR Velodyne VLP-16 são processados de forma a identificar os componentes estruturais da turbina (torre, nacelle, rotor e pás). Para tal, são aplicados algoritmos de segmentação e *clustering*, nomeadamente Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (*Density-Based Spatial Clustering of Applications with Noise*), que permitem isolar a nuvem de pontos associada a cada componente e, conseqüentemente, ajustar um modelo paramétrico tridimensional que representa a turbina. Durante esta fase inicial, o UAV mantém uma distância de segurança de aproximadamente 25 m até que o modelo esteja suficientemente definido.

O segundo corresponde ao pipeline de inspeção, em que o UAV ajusta dinamicamente a sua trajetória em torno das pás, guiado pelo centro da nuvem de pontos previamente segmentada pelo LiDAR. O voo decorre desde a raiz até à ponta da pá, garantindo o alinhamento adequado e a distância de referência. Nesta fase são também ativados sensores complementares, nomeadamente uma câmara ótica e uma câmara termográfica, que permitem recolher informação visual e térmica para a deteção de defeitos superficiais, incluindo fissuras, corrosão, erosão ou delaminações.

A navegação é assegurada pela fusão de dados de IMU e Real-Time Kinematic Global Navigation Satellite System (RTK-GNSS), processados por um filtro de Kalman estendido (EKF), fornecendo estimativas de pose com elevada precisão. Esta configuração permite ao sistema lidar com perturbações externas e garantir estabilidade mesmo em cenários dinâmicos. O pipeline foi validado em dois contextos: (i) simulação em *Gazebo*, com diferentes configurações de yaw e ângulos das pás, incluindo cenários de turbinas flutuantes sujeitas a oscilações em três níveis de intensidade; e (ii) ambiente de *mixed reality*, combinando o voo real do UAV STORK II com a representação virtual da turbina no *Gazebo*. Estes ensaios confirmaram a capacidade do sistema para efetuar modelação tridimensional e inspeção autónoma de pás, assegurando tanto a segurança do voo como a qualidade dos dados adquiridos.

Esta solução distingue-se por recorrer ao LiDAR como sensor principal para a modelação da turbina, complementado por dados visuais e térmicos apenas na fase final de inspeção. Tal configuração revela-se particularmente adequada a cenários offshore, onde as condições de iluminação e textura da superfície podem limitar a eficácia de métodos puramente visuais. Ao integrar perceção, planeamento e inspeção num único pipeline, este trabalho estabelece uma base sólida para sistemas futuros de inspeção autónoma baseados em UAVs.

Em [29] é descrito um sistema completo de inspeção autónoma de turbinas eólicas com recurso a um quadrotor, cobrindo desde a fase de aproximação até à aquisição de imagens detalhadas das pás. A proposta distingue-se por integrar tanto técnicas de visão computacional como sensores adicionais, de forma a lidar com os desafios inerentes à geometria complexa da turbina e à fiabilidade limitada do GPS em proximidade da estrutura.

O sistema utiliza como sensores principais uma câmara RGB de alta resolução montada num gimbal e um LiDAR 2D baseado em Time of Flight (ToF), igualmente montado num gimbal, responsável por manter a distância constante em relação às pás e compensar a sua curvatura. Os dados de GPS e IMU são fundidos para navegação de alto nível, mas a deteção e posicionamento da turbina são assegurados maioritariamente por visão computacional. Para tal, recorre-se a uma rede neuronal do tipo You Only Look Once (YOLO)v3, capaz de detetar em tempo real os principais componentes da turbina (torre, hub, nacelle e pás), complementada por uma

transformada de Hough personalizada para estimar com maior precisão a orientação das pás e calcular os ângulos de rotação.

A estratégia de inspeção inicia-se com a determinação autónoma do lado frontal da turbina, obtida através de uma trajetória circular em torno da torre, combinada com a análise das deteções visuais do hub e da nacelle. Posteriormente, cada pá é inspecionada de forma sistemática, abrangendo tanto a **aresta de ataque** (*leading edge*) como a **aresta de fuga** (*trailing edge*), assegurando a cobertura completa das superfícies principais. O percurso inclui manobras em torno da ponta da pá, seguidas de transições seguras entre pás consecutivas. Todo o processamento foi realizado num processador ARM quad-core com GPU integrada (256 núcleos), permitindo a execução em tempo real dos módulos de visão e controlo.

Os resultados experimentais evidenciaram que o sistema era capaz de realizar a inspeção completa das pás de forma autónoma, assegurando distâncias de segurança e recolha de imagens adequadas para posterior análise de defeitos. A integração da visão com o LiDAR 2D revelou-se determinante para a robustez do método, permitindo compensar falhas temporárias de deteção visual e garantir a continuidade da missão mesmo em condições adversas.

Outra linha de trabalho é explorada em [30], onde se propõe o uso de um UAV multicóptero equipado com LiDAR para inspeção automatizada de turbinas eólicas. A solução tem como foco central o mapeamento tridimensional da estrutura, a geração de trajetórias suaves e a implementação de mecanismos de controlo de distância para evitar colisões durante a missão.

O sistema recorre a um sensor LiDAR Hokuyo UTM-30LX, com alcance de 30 m e campo de visão de 270°, responsável pela construção de mapas 3D probabilísticos em tempo real. Estes mapas são representados através de octrees utilizando a framework Probabilistic 3D Mapping Framework using Octrees (OctoMap), permitindo uma compressão eficiente e a utilização em algoritmos de planeamento. Para garantir uma descrição precisa da pose, foram integrados ainda dados de GPS diferencial, IMU e, em ambiente indoor, um sistema de rastreamento infravermelho ART Trackpack.

Com base no mapa construído, são geradas trajetórias contínuas e diferenciáveis através de splines, ajustadas dinamicamente à orientação da nacelle e ao ângulo das pás. O controlo de distância é assegurado por um regulador Proportional–Derivative Controller (PD), alimentado por medições de IMU e filtragem mediana dos dados de LiDAR, permitindo compensar ruído e variações abruptas. Adicionalmente, a dinâmica do quadrotor foi simulada em ROS e *Gazebo*, incluindo efeitos de vento e erros no sinal de GPS, de modo a validar a robustez do planeamento e do controlo antes de testes práticos.

A validação experimental foi realizada em ambiente indoor com objetos substitutos de menor escala, confirmando a capacidade do sistema em detetar obstáculos, gerar trajetórias seguras e manter distâncias adequadas à superfície. Em cenários

de simulação de turbinas em escala real, o método demonstrou viabilidade para inspeção autônoma, embora sem validação direta em voo outdoor junto de turbinas reais. Esta limitação não invalida a relevância do trabalho, que representa uma etapa importante no desenvolvimento de soluções de inspeção baseadas em LiDAR, combinando percepção, mapeamento e planeamento de forma integrada.

Em [31] é apresentada uma solução para inspeção autônoma de pás de turbinas eólicas com recurso a um UAV equipado com LiDAR. O objetivo central consiste em permitir que o veículo acompanhe autonomamente o perfil da pá, desde a raiz até à ponta, garantindo a manutenção da distância de segurança e a aquisição de dados visuais e geométricos relevantes para a análise estrutural.

O sistema utiliza como sensor principal um LiDAR Velodyne VLP-16, responsável pela aquisição contínua da nuvem de pontos da pá. Para extrair a sua geometria em tempo real, é aplicado o algoritmo Random Sample Consensus (RANSAC), que identifica o plano predominante associado à superfície inspecionada. A trajetória do UAV é então ajustada de forma dinâmica, de modo a manter uma distância de referência ao plano estimado. A navegação é suportada por medições de IMU e GPS, fundidas através de um Discrete Kalman Filter (DKF), assegurando a continuidade da estimação mesmo em situações de perda temporária de dados de LiDAR. Complementarmente, uma câmara RGB embarcada é utilizada para registo visual em alta resolução.

O controlo é implementado através de uma máquina de estados hierárquica, que organiza a missão em etapas distintas (idle, voo estacionário, inspeção e regresso). Em cada etapa, são ativados controladores dedicados do tipo Proporcional-Integral-Derivative (PID) em cascata, ajustados às necessidades da fase em curso. Esta arquitetura híbrida permite manter a estabilidade de atitude, yaw e altitude, ao mesmo tempo que garante a adaptação do voo às exigências da inspeção.

A validação foi conduzida em duas fases: (i) simulação em ROS e *Gazebo*, utilizando modelos de turbinas realistas; e (ii) ensaios experimentais com uma pá em escala real em Split (Croácia), bem como protótipos em menor escala. Os resultados reportados indicam um erro médio de seguimento da distância de aproximadamente 0,19 m, confirmando a viabilidade do método para a inspeção autônoma de pás. Para além da robustez face a falhas momentâneas de deteção, o sistema demonstrou ser capaz de gerar dados adequados para reconstrução tridimensional e posterior análise estrutural.

O trabalho de [32] introduz o **BladeView**, um dos sistemas mais completos e robustos de inspeção autônoma de turbinas eólicas desenvolvido até à data. A proposta integra num único pipeline os módulos de percepção, planeamento e execução, assegurando não apenas a cobertura integral das pás, mas também a adaptação a diferentes condições ambientais e operacionais.

O sistema organiza-se em dois blocos principais. O primeiro corresponde ao

Automatic Parameter Calculation (APC) (*Automatic Parameter Calculation*), responsável pela estimação automática de cinco parâmetros críticos da turbina: altura do hub, posição relativa, orientação do rotor, ângulo de paragem das pás e raio do rotor. Esta etapa é realizada através da aplicação do algoritmo RANSAC a dados obtidos por LiDAR, complementado com técnicas de triangulação e regressão linear para aumentar a robustez da estimação. Desta forma, o sistema consegue adaptar-se a diferentes turbinas e iniciar a missão mesmo quando as pás se encontram em ângulos subótimos.

O segundo bloco, designado Automatic Flight System (AFS) (*Automatic Flight System*), gere a execução da missão de inspeção. Para cada pá, são geradas trajetórias que abrangem tanto a **aresta de ataque** (*leading edge*) como a **aresta de fuga** (*trailing edge*), assegurando a recolha de imagens adequadas para a deteção de defeitos. O planeamento combina algoritmos de procura baseados em *Hybrid A\** com a suavização de trajetórias via *B-Splines*, criando corredores dinâmicos de voo que mantêm a distância de segurança à superfície. O controlo de voo é assegurado por um controlador robusto integrado no autopiloto PX4, em articulação com medições de IMU, GPS e estimativas obtidas por EKF.

A plataforma experimental utilizada foi um UAV multirrotor DJI-M600, equipado com LiDAR Velodyne VLP-16, câmara ótica e gimbal estabilizado. A componente de planeamento foi executada num computador embarcado Intel NUC a correr ROS, em paralelo com o sistema de controlo Pixhawk/Ardupilot.

A validação incluiu mais de 100.000 simulações em *Gazebo* e 9.239 voos reais em diferentes cenários (desertos, áreas agrícolas, ambientes urbanos, montanhas, margens costeiras e offshore). Os resultados reportados demonstram taxas de sucesso entre 96% e 99%, com tempos médios de inspeção entre 19 e 28 minutos por turbina. O erro médio de trajetória manteve-se dentro de 0,25 m na ponta da pá e 0,64 m na zona da raiz, valores que asseguram imagens de elevada qualidade para reconstrução 3D e análise estrutural. Quando comparado com controladores PID convencionais, o BladeView registou um aumento de 6,7% na taxa de sucesso e uma redução de 15% no consumo energético, confirmando o seu potencial como solução de referência para inspeções autónomas em turbinas eólicas.

### 2.2.6 Soluções Comerciais com UAVs na Inspeção Offshore

Para além das contribuições académicas, existem já diversas **soluções comerciais** que demonstram a utilização de UAVs na inspeção de turbinas eólicas offshore. Estas plataformas diferenciam-se em termos de autonomia, integração com sistemas de análise de dados e aplicabilidade em cenários remotos.

A **SkySpecs** [33], fundada em 2012, consolidou-se como líder mundial em inspeções automatizadas de pás. A empresa reporta mais de 33.000 inspeções realizadas em 19 países, com drones totalmente autónomos capazes de completar a inspeção

de uma turbina em menos de 15 minutos. O sistema está associado à plataforma digital **Horizon**, que permite análise de danos, priorização de reparações e gestão integrada de ativos ao longo do ciclo de vida.

A empresa suíça **Sulzer Schmid** [34] disponibiliza a plataforma **3DX**, orientada para inspeções de alta resolução com drones de navegação autónoma. Embora a decolagem e a aterragem exijam assistência humana, a solução é reconhecida pela elevada qualidade dos dados obtidos e pela sua portabilidade, o que facilita a aplicação em ambientes offshore.

Outra oferta relevante é apresentada pela **DSLRPros** [35], que atua como integrador de hardware e software, fornecendo pacotes prontos com UAVs comerciais, como o Autel EVO Max 4T XE. Estas soluções permitem inspeções visuais e térmicas, embora dependam de controlo assistido por piloto, sendo mais adequadas a operações em pequena escala ou como redundância em parques offshore.

Por fim, a **ABJ Drones** [36] disponibiliza serviços completos de inspeção com UAVs equipados com câmaras RGB e térmicas. A sua oferta distingue-se pela possibilidade de realizar tanto inspeções externas (pás, torre) como internas (nacelle), em ambientes onshore e offshore.

## 2.3 Síntese e Considerações Finais

A revisão apresentada neste capítulo permitiu identificar as principais abordagens de inspeção de turbinas eólicas offshore com recurso a UAVs, abrangendo desde metodologias de otimização global até soluções locais de perceção, controlo e aquisição de dados, bem como plataformas comerciais já consolidadas no setor. De forma geral, constatou-se que a inspeção baseada em UAVs multirotores se afirmou como a solução predominante em cenários offshore, destacando-se pela capacidade de pairar, elevada manobrabilidade e facilidade de integração com diferentes sensores. Apesar disso, UAVs de asa fixa ou híbridos oferecem maior autonomia, embora a sua limitação na realização de inspeções de detalhe próximo restrinja a aplicabilidade em tarefas de proximidade às pás e à *nacelle*. Relativamente aos sensores, verificou-se que as câmaras RGB continuam a ser a tecnologia mais utilizada, devido à simplicidade e baixo custo, sendo frequentemente complementadas por termografia infravermelha para deteção de defeitos não visíveis. Contudo, a literatura mais recente aponta para uma tendência crescente na adoção de sistemas multissensoriais, que combinam RGB, LiDAR, IMU e GNSS, embora ainda enfrentem desafios relacionados com complexidade e custo. No que diz respeito aos algoritmos, identificaram-se avanços significativos na aplicação de métodos de *clustering* (como o DBSCAN), de regressão robusta (como o RANSAC) e de técnicas de visão computacional suportadas por CNNs. Estas abordagens contribuem para melhorar a fiabilidade na deteção de componentes da turbina e para sustentar pipelines mais completos que integram

perceção, planeamento e execução. Apesar disso, persiste uma lacuna na disponibilização de soluções *open-source* e replicáveis, capazes de acelerar a transição destas técnicas para contextos aplicados.

Adicionalmente, destaca-se que a validação em cenários reais permanece limitada devido aos elevados custos de hardware e operação offshore, justificando o recurso frequente a ambientes de simulação como o *Gazebo*. Esta abordagem permite desenvolver e testar pipelines complexos, minimizar riscos associados a protótipos reais e avaliar soluções em diversas condições antes da sua implementação prática.

Assim, conclui-se que a literatura oferece uma base sólida de metodologias e ferramentas, mas ainda carece de soluções integradas, acessíveis e especificamente orientadas para o conceito de UAVs residentes em parques eólicos offshore. É neste contexto que se insere a presente dissertação, que propõe um *pipeline* integrado de inspeção autónoma com UAVs multissensoriais, concebido e validado em simulação, contribuindo para o avanço desta área emergente.

## Capítulo 3

# Fundamentação Técnica

### 3.1 Robot Operating System (ROS)

O ROS (*Robot Operating System*) constitui um dos principais ecossistemas de software utilizados no desenvolvimento de aplicações robóticas. Apesar do nome, o ROS não corresponde a um sistema operativo completo, mas antes a um **middleware de robótica** que disponibiliza bibliotecas, ferramentas e uma arquitetura distribuída orientada a mensagens, concebida para facilitar a integração de sensores, atuadores e algoritmos em sistemas robóticos complexos [37].

A sua principal funcionalidade assenta na **abstração da comunicação entre processos**, através do paradigma *publisher-subscriber*, onde diferentes módulos (nós) podem trocar informação de forma assíncrona, segura e escalável. Esta abordagem modular permite o desenvolvimento colaborativo e a reutilização de código, sendo atualmente suportada por uma vasta comunidade académica e industrial.

O ROS é amplamente aplicado em tarefas que vão desde a perceção multissensorial até ao planeamento de movimento, controlo e simulação de robôs. Entre as suas vantagens destacam-se a **portabilidade**, a **extensibilidade** e a compatibilidade com ambientes de simulação como o *Gazebo*, bem como a integração nativa com controladores de voo como o Open-Source Flight Control Platform (PX4).

No presente trabalho, foi adotada a versão **ROS Noetic Ninjemys**, última versão estável da série *ROS 1*, suportada até 2025. Embora exista já a transição para o **ROS 2**, a escolha do Noetic justifica-se pela sua maior maturidade, estabilidade

e compatibilidade com bibliotecas e pacotes relevantes para a inspeção de turbinas eólicas offshore.

## 3.2 Gazebo

O **Gazebo** é um simulador de código aberto para robótica, amplamente utilizado em investigação e desenvolvimento devido à sua capacidade de recriar ambientes realistas em 3D. Permite modelar tanto a física (dinâmica, colisões, fricção, gravidade) como a interação com sensores e atuadores, proporcionando uma plataforma segura e flexível para testar algoritmos antes da sua implementação em hardware real [38].

A integração nativa com o ROS torna o Gazebo especialmente adequado para cenários de prototipagem e validação de missões robóticas. No contexto desta dissertação, o simulador foi utilizado para adicionar modelos de turbinas eólicas offshore e cenários ambientais, possibilitando a avaliação de estratégias de percepção, controlo e planeamento em condições próximas da realidade. Este processo é particularmente relevante face às dificuldades logísticas e ao elevado custo de testar diretamente em hardware real no mar.

## 3.3 PX4 Autopilot

O **PX4 Autopilot** é uma plataforma de controlo de voo de código aberto, desenvolvida e mantida pela comunidade internacional de robótica aérea [39]. Trata-se de um sistema modular e escalável que suporta uma ampla gama de veículos não tripulados, incluindo multirotores, helicópteros, aeronaves de asa fixa e veículos híbridos.

Entre as suas principais características destacam-se:

- suporte a múltiplas arquiteturas de hardware (placas Pixhawk, Snapdragon, Intel NUC, entre outras);
- integração nativa com o ROS, permitindo comunicação bidirecional com simuladores como o Gazebo;
- controladores de voo robustos (PID, MPC, NMPC) configuráveis de acordo com a aplicação;
- flexibilidade para operar tanto em ambiente real como em simulação, assegurando continuidade entre prototipagem e implementação prática.

No contexto desta dissertação, o PX4 desempenha um papel essencial como **elo comum entre simulação e realidade**. Por um lado, permite correr em conjunto com o ROS e o Gazebo, simulando de forma realista a dinâmica do UAV e o comportamento dos sensores. Por outro, trata-se do mesmo *firmware* utilizado em

controladores de voo reais, garantindo que as soluções desenvolvidas no ambiente virtual são transferíveis para drones físicos sem alterações estruturais significativas.

Esta característica é particularmente valiosa no caso em estudo, dado que os ensaios com hardware real em ambiente offshore implicam custos e riscos elevados. Assim, a utilização do PX4 em simulação oferece uma plataforma realista para validar algoritmos de percepção, controlo e planeamento em UAVs residentes destinados à inspeção de turbinas eólicas offshore.

### 3.4 Câmaras RGB e Modelo de Projeção Pinhole

As câmaras RGB são os sensores visuais mais amplamente utilizados em sistemas de inspeção com recurso a UAVs, fornecendo informação em três canais de cor (*Red*, *Green*, *Blue*). Pela sua simplicidade, baixo custo e elevada resolução, constituem ferramentas fundamentais para a deteção de defeitos superficiais, tais como fissuras, delaminações ou sinais de degradação estrutural nas pás de turbinas eólicas offshore.

O funcionamento de uma câmara pode ser descrito matematicamente pelo **modelo pinhole**, que estabelece a transformação entre um ponto tridimensional (3D) no espaço e a sua projeção bidimensional (2D) no plano da imagem. Esta transformação é expressa por:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{R} \mid \mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3.1)$$

onde:

- $(X, Y, Z)$  são as coordenadas do ponto no espaço 3D no referencial do mundo;
- $(u, v)$  são as coordenadas do ponto projetado no plano da imagem (em píxeis);
- $s$  é um fator de escala;
- $\mathbf{R}$  e  $\mathbf{t}$  representam, respetivamente, a matriz de rotação e o vetor de translação, que definem a posição e orientação da câmara no espaço (**parâmetros extrínsecos**);
- $\mathbf{K}$  é a matriz dos **parâmetros intrínsecos** da câmara, definida por:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.2)$$

em que:

- $f_x, f_y$  correspondem às distâncias focais expressas em píxeis, relacionando a distância focal física da lente com o tamanho do píxel do sensor;
- $(c_x, c_y)$  representam as coordenadas do *ponto principal* (*principal point*), isto é, o centro ótico da imagem.

A geometria da câmara permite ainda calcular o **campo de visão** (*Field of View*, FOV), que está diretamente relacionado com a resolução da imagem e com os parâmetros intrínsecos. Para uma câmara de resolução  $(W, H)$  e distâncias focais  $f_x, f_y$ , o FOV é obtido por:

$$\text{FOV}_x = 2 \arctan \left( \frac{W}{2f_x} \right), \quad \text{FOV}_y = 2 \arctan \left( \frac{H}{2f_y} \right). \quad (3.3)$$

Deste modo, assumindo uma distância  $d$  entre o UAV e a superfície inspecionada, é possível estimar a **janela de inspeção**, ou seja, a área efetivamente visível numa captura de imagem, através de:

$$A_x = 2d \cdot \tan \left( \frac{\text{FOV}_x}{2} \right), \quad A_y = 2d \cdot \tan \left( \frac{\text{FOV}_y}{2} \right). \quad (3.4)$$

Este enquadramento matemático é de elevada relevância para o planeamento de missões de inspeção, uma vez que permite estabelecer a relação entre a resolução da câmara, a distância de operação e a área efetivamente coberta numa única imagem, assegurando a otimização entre segurança de voo, qualidade visual e eficiência da missão [40].

### 3.5 LiDAR 3D

O LiDAR (*Light Detection and Ranging*) é uma tecnologia de sensorização ativa que utiliza feixes de laser para medir distâncias com elevada precisão. O princípio fundamental baseia-se na emissão de pulsos de luz na direção de um objeto e na medição do tempo de voo (ToF, *Time of Flight*) até à receção do eco refletido. A distância  $d$  entre o sensor e o ponto de reflexão é calculada pela seguinte relação:

$$d = \frac{c \cdot \Delta t}{2}, \quad (3.5)$$

onde  $c$  representa a velocidade da luz no ar e  $\Delta t$  o intervalo de tempo entre a emissão e a receção do pulso. O fator  $1/2$  deve-se ao percurso de ida e volta do feixe.

Ao efetuar varrimentos sistemáticos em diferentes ângulos, o LiDAR gera uma nuvem de pontos tridimensional que descreve a geometria do ambiente. Esta nuvem pode ser representada em coordenadas cartesianas  $(x, y, z)$  a partir de medições polares  $(d, \theta, \phi)$ , segundo:

$$\begin{aligned}x &= d \cdot \cos \theta \cdot \cos \phi, \\y &= d \cdot \cos \theta \cdot \sin \phi, \\z &= d \cdot \sin \theta,\end{aligned}\tag{3.6}$$

em que  $d$  é a distância calculada pela Eq. 3.5,  $\theta$  o ângulo de elevação e  $\phi$  o ângulo de azimute.

Tal como nas câmaras, o LiDAR possui um **campo de visão** (FOV, *Field of View*) definido pelos ângulos máximos de varrimento horizontal (FOV<sub>h</sub>) e vertical (FOV<sub>v</sub>). Considerando uma distância de inspeção  $d$ , a área projetada de cobertura é dada por:

$$A_h = 2d \cdot \tan\left(\frac{\text{FOV}_h}{2}\right), \quad A_v = 2d \cdot \tan\left(\frac{\text{FOV}_v}{2}\right).\tag{3.7}$$

Deste modo, o sensor consegue abranger uma janela de inspeção ( $A_h \times A_v$ ) que depende tanto da sua configuração ótica como da distância ao alvo.

Segundo Waykar [41], o LiDAR destaca-se pela sua elevada precisão métrica (tipicamente na ordem dos centímetros), robustez em ambientes com baixa iluminação e capacidade de reconstrução tridimensional de superfícies complexas. Estas características tornam-no especialmente adequado para a inspeção de turbinas eólicas offshore, onde a geometria das pás e a grande escala da estrutura exigem sensores capazes de fornecer dados geométricos detalhados e consistentes.

Além disso, a integração do LiDAR com sistemas de navegação e visão computacional possibilita a implementação de algoritmos de mapeamento e localização simultâneos (SLAM), bem como técnicas de planeamento de trajetória baseadas na nuvem de pontos. Este enquadramento justifica a sua adoção no presente trabalho, não apenas como sensor de suporte visual, mas como elemento central no processo de perceção e inspeção autónoma.

### 3.6 Controlo PID

O controlo PID (*Proportional-Integral-Derivative*) é um dos métodos mais utilizados em sistemas de controlo para robótica, aeronaves não tripuladas, controlo de atitude, altitude e trajetória. A sua popularidade baseia-se na simplicidade, robustez e eficácia para corrigir erros entre valores desejados (set-points) e valores medidos.

Denotando por  $e(t)$  o erro entre o valor desejado e o valor atual do sistema, o sinal de controlo  $u(t)$  dado por um controlador PID em tempo contínuo é:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt},\tag{3.8}$$

onde:

- $K_p$  é o ganho proporcional: corrige o erro instantâneo, respondendo de forma rápida às discrepâncias atuais;
- $K_i$  é o ganho integral: acumula erros passados para eliminar erro estacionário (*steady-state error*);
- $K_d$  é o ganho derivativo: antecipa variações do erro, ajudando a atenuar oscilações e melhorar a resposta transitória.

No domínio de Laplace, assumindo linearidade e sistema SISO, o controlador pode ser representado por:

$$U(s) = \left( K_p + \frac{K_i}{s} + K_d s \right) E(s), \quad (3.9)$$

com  $E(s)$  e  $U(s)$  as transformadas de Laplace do erro  $e(t)$  e do controlo  $u(t)$ , respetivamente.

Em UAVs, o PID é usado para controlar atitude (roll, pitch, yaw), altitude, bem como manter trajectórias definidas entre objectos ou seguir paths. Os sensores como acelerómetros, giroscópios, IMU fornecem medições contínuas do estado, sendo o erro calculado em cada instante. A actuação (por exemplo, no motor ou nas superfícies de controle) é ajustada em conformidade.

Segundo o estudo de Ørsted e Ma [42], embora existam controladores mais sofisticados (MPC, adaptativos, etc.), o PID continua a ser uma referência devido à sua implementação simples e capacidade de responder bem a perturbações moderadas.

A eficácia do PID depende fortemente da escolha dos ganhos  $K_p, K_i, K_d$ . Algumas das técnicas de sintonia incluem:

- métodos heurísticos como Ziegler-Nichols;
- sintonia experimental (“trial and error”) em simulação;
- análise de resposta em frequência;
- algoritmos automáticos ou adaptativos (como ganho adaptativo) nos casos em que o modelo não é totalmente conhecido.

### 3.7 RANSAC

O RANSAC (*Random Sample Consensus*) é um dos algoritmos mais utilizados em visão computacional e robótica para estimação robusta de modelos matemáticos a partir de dados contaminados por ruído ou *outliers*. Foi introduzido por Fischler e Bolles em 1981 [43] e desde então tornou-se um paradigma fundamental para problemas como estimação de transformações geométricas, segmentação de planos em nuvens de pontos e reconstrução tridimensional.

A ideia central do RANSAC é que, mesmo em conjuntos de dados com elevada proporção de *outliers*, é possível encontrar um modelo adequado escolhendo amostras aleatórias mínimas e avaliando o seu grau de consistência com o conjunto total. O processo segue tipicamente os seguintes passos:

1. Seleciona-se aleatoriamente um subconjunto mínimo de pontos suficientes para estimar o modelo;
2. Ajusta-se o modelo a esses pontos;
3. Determina-se o número de pontos do conjunto total que se enquadram no modelo segundo um limiar de tolerância (*inliers*);
4. Repete-se o processo durante  $N$  iterações, guardando o modelo com maior número de *inliers*;
5. Opcionalmente, recalcula-se o modelo final com todos os *inliers* identificados.

Considerando um conjunto de observações  $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$ , deseja-se ajustar um modelo  $M(\theta)$  com parâmetros  $\theta$ . Em cada iteração:

$$\theta^* = \arg \max_{\theta} |\{x_i \in \mathcal{D} \mid \text{dist}(x_i, M(\theta)) < \tau\}|, \quad (3.10)$$

onde  $\tau$  é o limiar de tolerância que define se um ponto é considerado um *inlier*. O modelo escolhido  $\theta^*$  é aquele que maximiza o número de pontos consistentes.

No contexto de inspeção de turbinas eólicas com UAVs, o RANSAC desempenha um papel essencial em diferentes tarefas:

- **Segmentação de superfícies:** identificação de planos dominantes em nuvens de pontos captadas por sensores LiDAR, como as pás ou a torre da turbina;
- **Ajuste de linhas e círculos:** usado para detetar bordos estruturais ou o contorno do hub;
- **Filtragem robusta:** separação entre *inliers* e *outliers*, garantindo maior fiabilidade na reconstrução 3D e no planeamento de trajetórias.

## 3.8 DBSCAN

O DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é um algoritmo de agrupamento não supervisionado amplamente utilizado em análise de dados espaciais e visão computacional. Foi introduzido por Ester et al. [44] e distingue-se por identificar **clusters de forma arbitrária** em grandes bases de dados, sendo simultaneamente robusto à presença de *outliers*.

O funcionamento do DBSCAN baseia-se na noção de densidade de pontos. Dado um raio de vizinhança  $\varepsilon$  e um número mínimo de pontos  $MinPts$ , os pontos são classificados em três categorias:

- **Pontos núcleo:** possuem pelo menos  $MinPts$  vizinhos dentro de um raio  $\varepsilon$ ;
- **Pontos de fronteira:** não atingem o critério de núcleo, mas estão na vizinhança de um ponto núcleo;
- **Ruído:** pontos que não pertencem a qualquer cluster.

Com base nesta definição, o algoritmo agrupa iterativamente os pontos núcleo e os seus vizinhos, formando regiões de elevada densidade.

Seja  $D = \{x_1, x_2, \dots, x_n\}$  o conjunto de pontos, define-se a vizinhança de  $\varepsilon$  de um ponto  $x_i$  como:

$$N_\varepsilon(x_i) = \{x_j \in D \mid \|x_i - x_j\| \leq \varepsilon\}. \quad (3.11)$$

Um ponto  $x_i$  é considerado núcleo se:

$$|N_\varepsilon(x_i)| \geq MinPts. \quad (3.12)$$

Clusters são então construídos como a união transitiva de pontos núcleo e fronteira conectados pela densidade.

### Aplicações em UAVs e Inspeção Offshore

No contexto da inspeção de turbinas eólicas com UAVs, o DBSCAN tem especial relevância para:

- **Segmentação de nuvens de pontos LiDAR**, separando automaticamente diferentes partes da turbina (torre, nacelle e pás);
- **Filtragem de ruído**, descartando pontos espúrios causados por reflexões, interferências ou condições ambientais adversas;
- **Deteção de objetos**, ao agrupar dados sensoriais em regiões coerentes, simplificando a modelação e a navegação autónoma.

## 3.9 Ângulos de Rotação em Turbinas Eólicas

A operação e a inspeção de turbinas eólicas *offshore* requerem uma compreensão das possíveis rotações associadas à sua estrutura. Uma turbina típica de eixo horizontal pode ser descrita através de três rotações fundamentais, que influenciam diretamente a geometria da inspeção aérea com UAVs [29]:

- **Ângulo de *yaw* da nacelle** — rotação em torno do eixo vertical da torre, permitindo alinhar o rotor com a direção do vento;
- **Ângulo de *roll* das pás** — movimento rotacional do conjunto das três pás em torno do eixo do hub, que define a velocidade angular do rotor;
- **Ângulo de *pitch* da pá** — rotação de cada pá em torno do seu próprio eixo longitudinal, controlando o ângulo de ataque e, conseqüentemente, a potência extraída do vento.

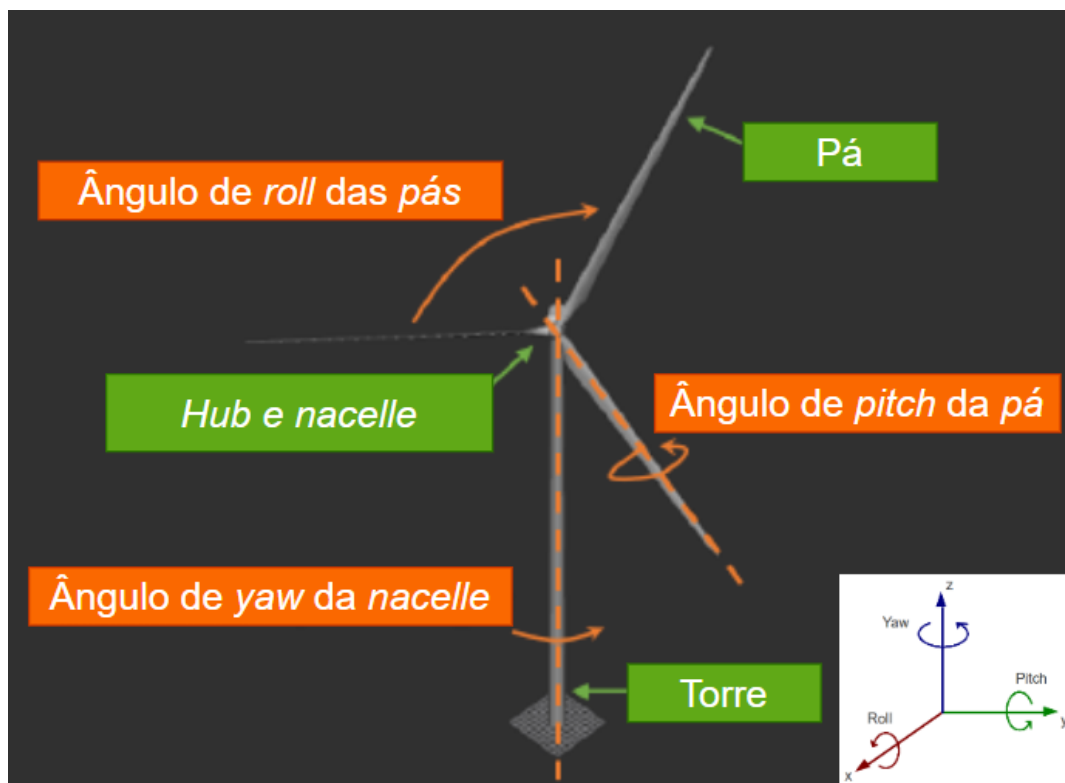


Figura 3.1: Modelo simplificado de uma turbina eólica *offshore*, indicando as três rotações principais: *yaw* da nacelle, *roll* das pás e *pitch* das pás [29].

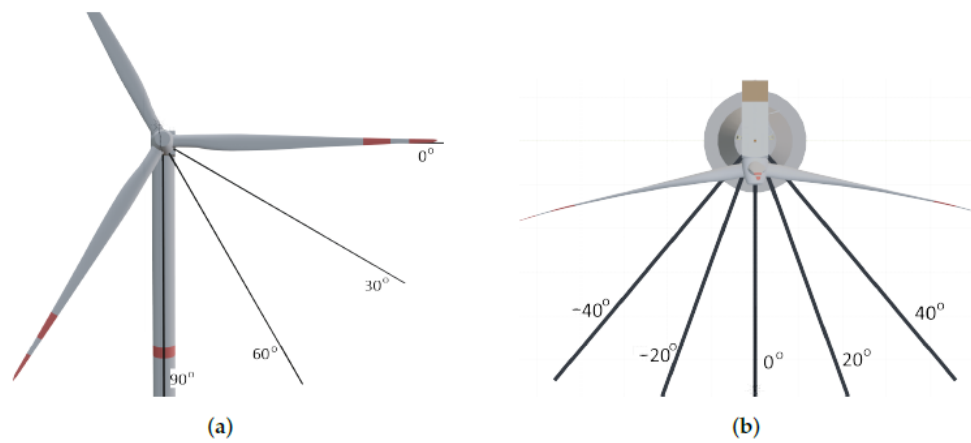


Figura 3.2: Diferentes configurações de rotação em turbinas: (a) *roll* das pás; (b) *yaw* da nacelle [28].

## Capítulo 4

# Implementação

A implementação do sistema proposto centra-se na inspeção das pás das turbinas eólicas, elemento mais crítico a inspecionar conforme verificado anteriormente. Contudo, considerando o conceito de UAVs residentes [5], que iniciam a missão a partir de uma base localizada na própria turbina, a torre também é inspecionada de forma complementar durante o percurso de subida até ao *hub*.

Em relação ao planeamento, perceção e controlo, esta abordagem foi concebida a partir de trabalhos prévios, como Oliveira et al. [28], Gu et al. [29] e Car et al. [31], que apresentam metodologias validadas de inspeção autónoma de turbinas eólicas *offshore* com UAVs multissensoriais. A solução proposta visa colmatar lacunas identificadas na literatura, focando em estratégias integradas, acessíveis e replicáveis.

O UAV selecionado é o *Yuneec Typhoon H480* [45], escolhido pela disponibilidade de um modelo funcional no simulador Gazebo [46], compatível com o *PX4 Autopilot* [47], permitindo comunicação direta com o ROS Noetic [48] através do pacote MAVROS [49], que utiliza mensagens MAVLink para controlo e sensorização. O modelo base já integra uma câmara visual de alta resolução e, nesta solução, foi complementado com um LiDAR 3D Velodyne VLP-16 [50] para perceção em tempo real. A nuvem de pontos processada pelo LiDAR guia o drone ao longo da missão, dispensando o armazenamento de dados ou a construção de mapas. Além disso, o sistema suporta a operação de múltiplos UAVs, permitindo uma abordagem escalável a missões de inspeção.

## 4.1 Arquitetura do Sistema

A arquitetura distingue componentes *offboard* e *onboard*, conforme ilustrado na Figura 4.1. O módulo *offboard* corresponde ao planejador central, responsável pelo lançamento faseado de até três UAVs (UAV1, UAV2, UAV3). Este planejador atribui a cada UAV a pá a inspecionar (esquerda, direita ou central) e monitoriza o progresso da missão através de mensagens publicadas em tópicos específicos. À medida que cada UAV conclui etapas do seu pipeline, publica atualizações, e com base nessas informações o planejador central decide quando lançar o próximo drone.

Cada UAV executa o *pipeline* da missão de forma *onboard*, incluindo descolagem, alinhamento com a torre, subida até ao hub, alinhamento com o hub, inspeção de uma pá, descida controlada, localização da plataforma de aterragem e aterragem. Neste módulo *onboard* foi adotada uma abordagem reativa, de forma a maximizar a independência de cada UAV relativamente ao planejador de missão e aos restantes veículos. Assim, após receber a atribuição da pá a inspecionar, o UAV executa a missão de forma autónoma, adaptando-se em tempo real ao ambiente e reagindo a potenciais obstáculos ou estruturas imprevistas. Esta estratégia reduz a necessidade de coordenação intensiva e de trocas constantes de informação, aumentando a robustez do sistema perante as condições hostis e falhas de comunicação típicas de cenários *offshore*.

O processamento dos dados do LiDAR utiliza técnicas de segmentação como DBSCAN [44] e RANSAC [43], enquanto a câmara é dedicada à captura de imagens para inspeção e à deteção de marcadores ArUco durante a aterragem. A estimativa de pose é assegurada pelo filtro EKF do PX4. A Figura 4.2 representa as etapas da missão, organizadas de Etapa 0 a Etapa 7, desde a descolagem até à aterragem, passando pela inspeção da torre e das pás.

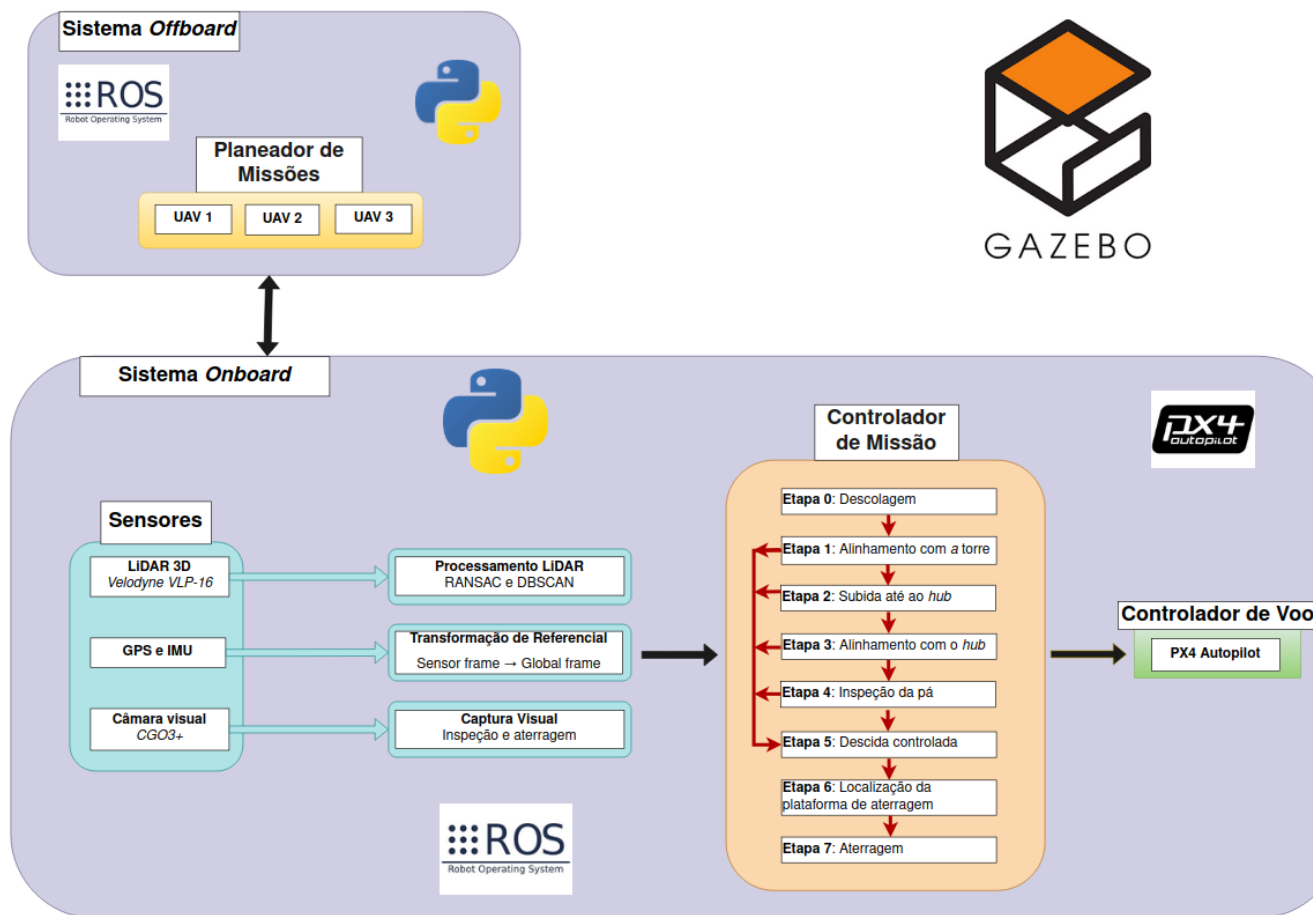


Figura 4.1: Arquitetura do sistema proposto.

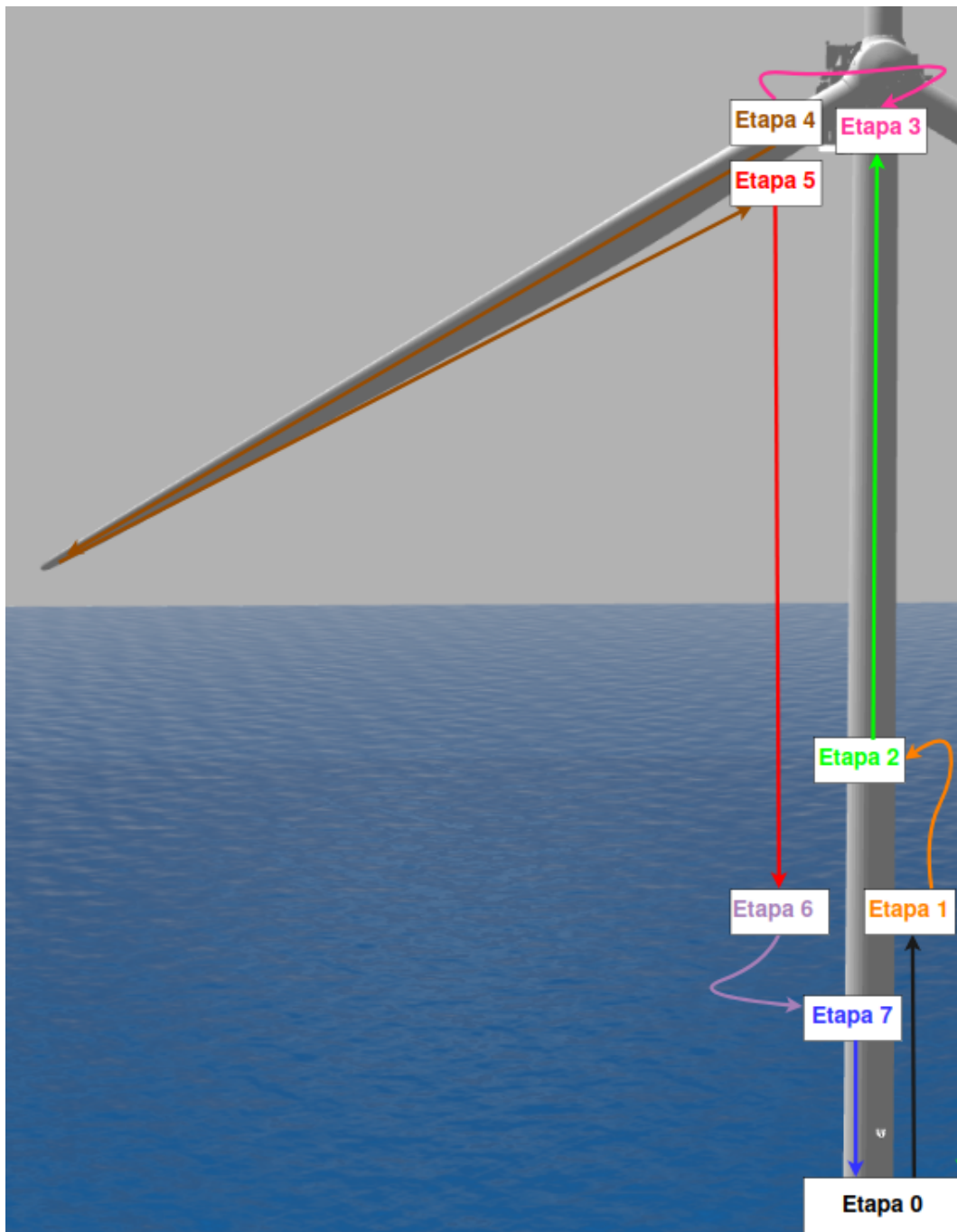


Figura 4.2: Representação visual das etapas da missão autónoma de inspeção

## 4.2 Ambiente de Simulação com UAVs

### 4.2.1 Turbina *Offshore*

Para representar a turbina eólica *offshore* foi utilizado um modelo tridimensional obtido na plataforma *CGTrader* [51], posteriormente segmentado em várias componentes individuais: pás, *nacelle*, *hub* e torre. Esta segmentação permitiu maior flexibilidade na configuração do modelo, permitindo que cada componente fosse manipulado de forma independente.

Numa fase inicial de preparação do modelo, foi implementado um processo de cálculo de envolventes geométricas para cada componente (Figura 4.3). Estas envolventes representam o maior caso dimensional de cada peça, isto é, a largura máxima que cada componente pode apresentar. No caso das pás, o cálculo da respetiva envolvente permite determinar que, mesmo em diferentes configurações do ângulo de *pitch* — ou seja, na rotação da pá em torno do seu eixo longitudinal —, a largura máxima observada não excede um determinado valor. Considera-se que esta largura máxima corresponde à distância entre a borda de ataque e a borda de fuga da pá, quando observada de frente pelo UAV alinhado com o *hub*. Esta informação permite garantir que durante a inspeção a câmara visual consegue captar toda a largura da pá numa única imagem.

Com recurso a um *código Python* desenvolvido para este efeito (ver Secção 7.1 do Anexo 7), foi possível aplicar rotações às pás em torno do *hub* (ângulo de *roll*) e alterar a orientação da *nacelle* (*yaw*). O *código Python* gera automaticamente novos ficheiros no formato Standard Tessellation Language (STL), cada um representando uma configuração geométrica distinta da turbina. Desta forma, foi possível simular diferentes cenários de inspeção, nomeadamente:

- variações no ângulo de *yaw* da *nacelle* em relação à torre;
- diferentes ângulos de *roll* das pás;
- combinações entre ambas as transformações.

Embora a rotação dinâmica em tempo real não tenha sido implementada no *Gazebo*, esta metodologia permitiu criar múltiplas condições iniciais, fundamentais para avaliar a robustez do pipeline de inspeção. A Figura 4.4 ilustra o modelo tridimensional da turbina eólica, adaptado a partir do recurso obtido em [51] e posteriormente integrado no ambiente de simulação *Gazebo*, servindo como base para todos os cenários de teste considerados.

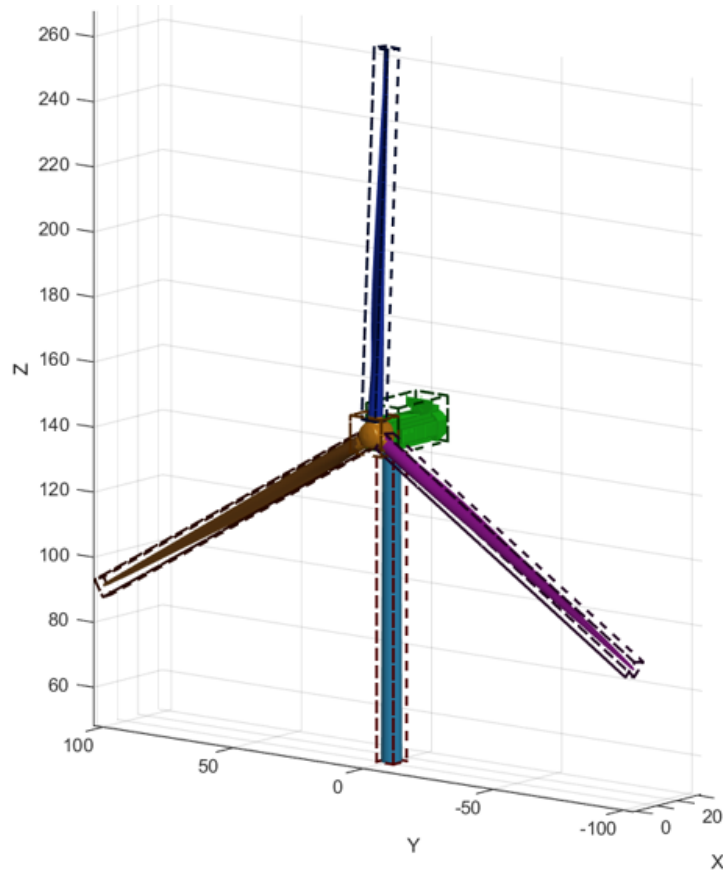


Figura 4.3: Segmentação da turbina em componentes independentes (pás, *nacelle*, *hub* e torre) e cálculo das envolventes geométricas de cada peça, utilizadas para estabelecer limites de inspeção.

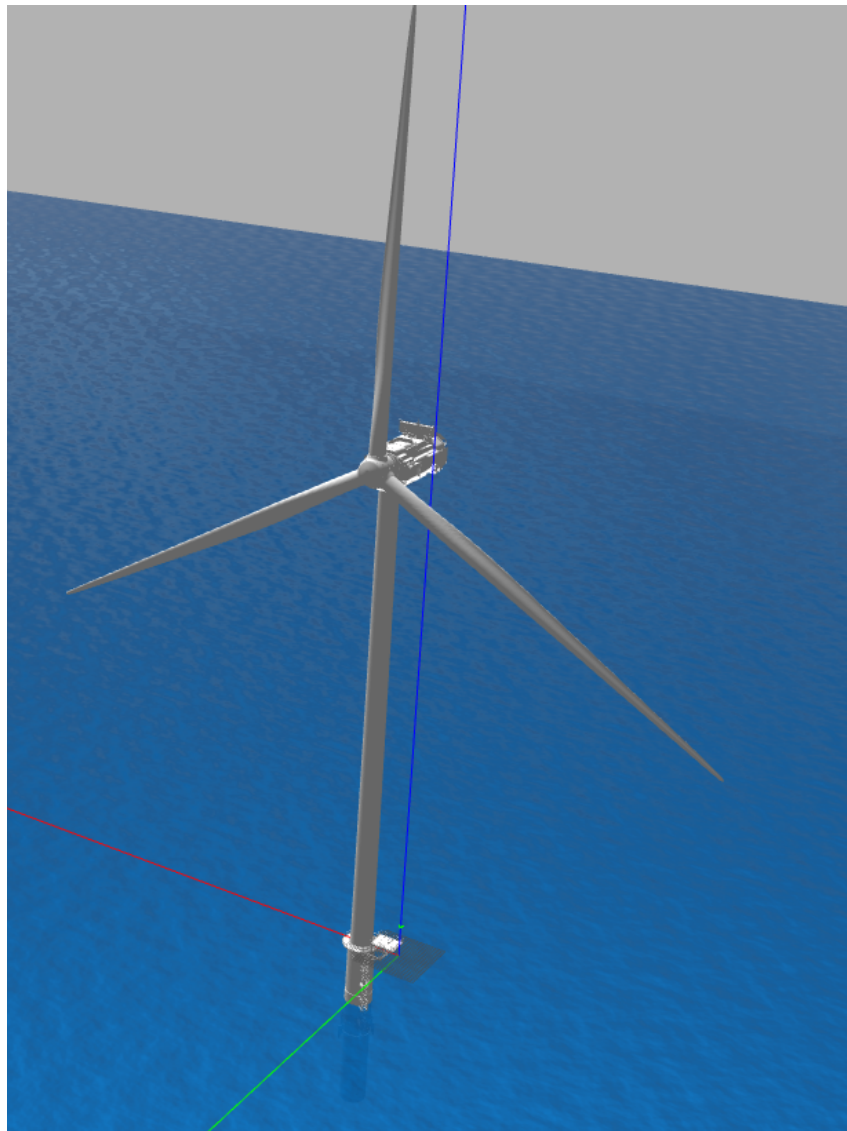


Figura 4.4: Modelo tridimensional da turbina eólica *offshore*, adaptado a partir de [51] e integrado no ambiente de simulação *Gazebo*.

#### 4.2.2 UAV Yuneec Typhoon H480 com LiDAR Velodyne VLP-16

O UAV Yuneec Typhoon H480, desenvolvido pela Yuneec [52], é um *hexacóptero* amplamente utilizado em inspeções profissionais devido à sua estabilidade, redundância estrutural (seis rotores) e à integração de uma câmara CGO3+ montada num *gimbal* de três eixos.

A Figura 4.5 apresenta o UAV original, enquanto a Tabela 4.1 resume as principais características do dispositivo, incluindo dimensões, peso, autonomia e sensores integrados.



Figura 4.5: Drone Yuneec Typhoon H480 original [52].

Propriedade	Especificação
Peso do <i>airframe</i>	1,695 kg
Peso total	2,001 kg (com bateria)
Dimensões (C x L x A)	32,9 cm x 27,9 cm x 25,6 cm
Bateria	LiPo 4S 5400 mAh, 14,8 V (79,9 Wh)
Câmara	CGO3+ 12 MP, 4K UHD 30 fps, <i>gimbal</i> 3 eixos
Transmissão de vídeo	720p, 5,8 GHz WiFi

Tabela 4.1: Características originais do UAV Yuneec Typhoon H480 [45].

Como referido anteriormente, para complementar a perceção tridimensional do ambiente, foi seleccionado o sensor LiDAR Velodyne VLP-16, cuja representação encontra-se na Figura 4.6. A Tabela 4.2 resume as suas características, como número de canais, alcance, precisão e campo de visão.



Figura 4.6: Sensor LiDAR Velodyne VLP-16 [53].

Propriedade	Especificação
Canais	16
Alcance máximo	100 m
Precisão	$\pm 3$ cm
Campo de visão vertical	$30^\circ$ ( $\pm 15^\circ$ )
Campo de visão horizontal	$360^\circ$
Velocidade de rotação	5–20 Hz
Pontos por segundo	300.000
Peso	830 g
Dimensões	$\varnothing 103$ mm x 72 mm
Laser	905 nm, Classe 1
Classificação IP	IP67

Tabela 4.2: Características do sensor LiDAR Velodyne VLP-16 [53].

A integração digital do LiDAR no modelo do Typhoon H480 no simulador *Gazebo* exigiu algumas alterações. Originalmente descrito em *URDF*, o LiDAR foi convertido para *SDF*, permitindo a sua incorporação direta no modelo do UAV. O sensor foi fixado ao corpo do *hexacóptero* através de uma *link* dedicada, configurado com taxa de atualização de 10 Hz, alcance máximo de 100 m e ruído gaussiano de 0,8 cm.

O offset vertical entre a câmara RGB e o LiDAR foi definido como 0,312 m, sem deslocamento nos eixos x e y, garantindo alinhamento facilitado para a perceção (Figura 4.7).

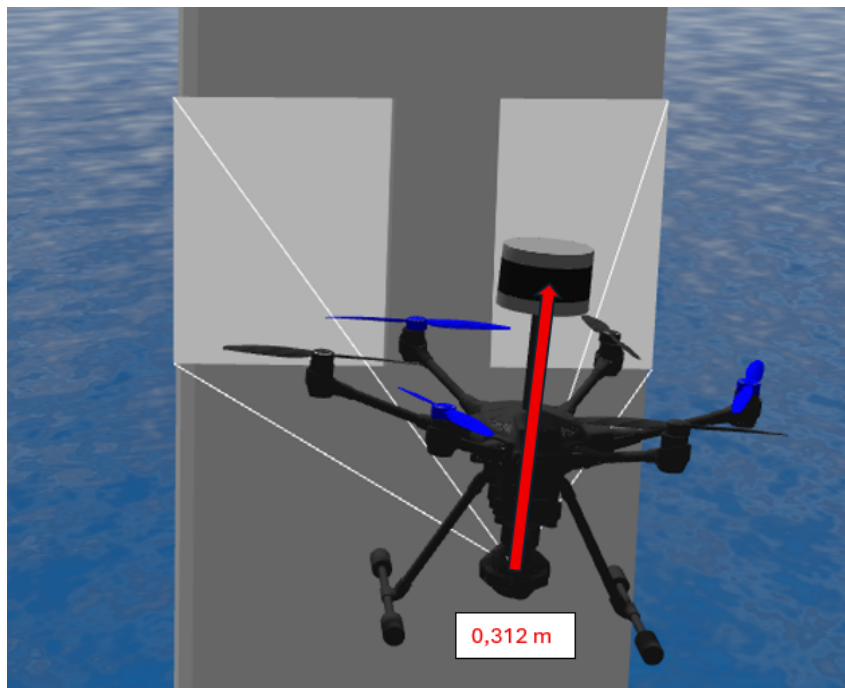


Figura 4.7: Offset vertical de 0,312 m entre a câmara RGB e o LiDAR no modelo Typhoon H480 adaptado.

Cada UAV possui uma configuração independente de comunicação, essencial para garantir a operação simultânea de três Typhoon H480 adaptados com LiDAR Velodyne VLP-16 em missões coordenadas. A configuração é implementada através de *launch files* em ROS e PX4, permitindo instanciar cada UAV com *namespaces* próprios e tópicos distintos.

A Tabela 4.3 sintetiza as principais portas de comunicação para cada UAV, incluindo portas MAVLink para controlo, portas GStreamer para vídeo e portas específicas para a transmissão de imagens das câmaras.

Parâmetro	UAV1	UAV2	UAV3
Namespace ROS	/uav1	/uav2	/uav3
Porta MAVLink UDP	14540	14541	14542
Porta MAVLink TCP	4560	4561	4562
Porta GStreamer Vídeo	5600	5601	5602
Porta MAVLink Câmera UDP	14530	14531	14532

Tabela 4.3: Síntese das portas de comunicação e *namespaces* de cada UAV.

No modelo digital em *SDF*, cada LiDAR Velodyne VLP-16 está configurado para publicar num tópico ROS exclusivo para o respectivo UAV:

- UAV1: /velodyne\_points\_1
- UAV2: /velodyne\_points\_2
- UAV3: /velodyne\_points\_3

Esta configuração assegura que cada UAV opera de forma autónoma, com sensorização e comunicação isoladas, permitindo missões simultâneas e coordenadas dentro do mesmo ambiente de simulação (Figura 4.8).

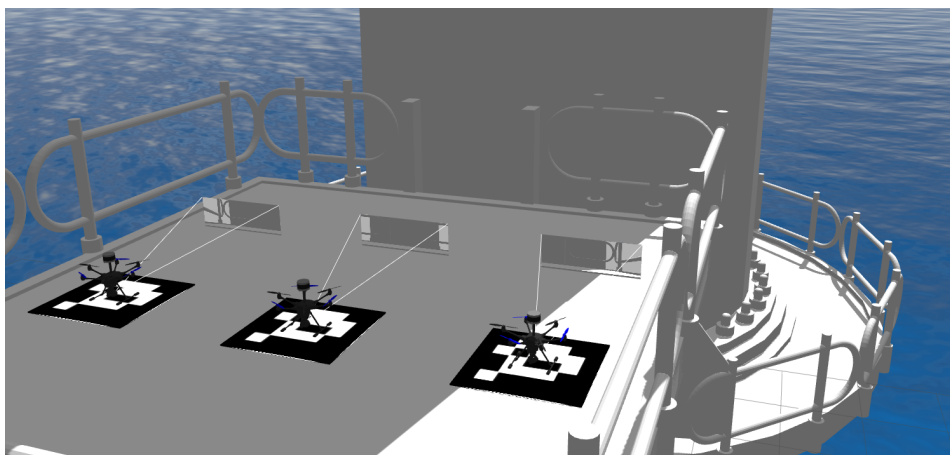


Figura 4.8: Três UAVs residentes na plataforma junto à turbina eólica *offshore*.

## 4.3 Condições Iniciais

Para o correcto funcionamento do pipeline de inspecção, é essencial definir condições iniciais claras para os UAVs e para a turbina eólica *offshore*. Estas condições garantem resultados consistentes e reproduzíveis, permitindo que o planeador de missões execute o lançamento dos UAVs apenas quando as condições são seguras.

### 4.3.1 Condição 1: Turbina fixa, ambiente controlado e comunicações com o planeador de missões

Nesta condição, consideram-se turbinas eólicas *offshore* fixas e paradas, com ausência de vento anormal, representando um ambiente de teste ideal.

Os UAVs comunicam exclusivamente com o **planeador de missões**, o módulo central da arquitectura do sistema responsável pela gestão da missão. Este planeador tem acesso aos dados da turbina e fornece aos UAVs informações aproximadas sobre os seguintes graus de rotação (ver Seção 3.9):

- **Ângulo de yaw da nacelle**
- **Ângulo de roll das pás**
- **Ângulo de pitch das pás**

O planeador assegura que cada UAV inicie a missão apenas quando os ângulos de yaw, roll e pitch estiverem dentro dos limites adequados para uma inspecção segura. Assim, os UAVs podem planear a trajectória, otimizar a captura de imagens e dados LiDAR, e executar a missão sem necessidade de sincronização directa entre si.

### 4.3.2 Condição 2: Limites do ângulo de *roll* das pás

A segunda condição crítica refere-se ao ângulo de *roll* das pás em torno do eixo do *hub*. Esta rotação determina a posição relativa das pás e é essencial para a correcta identificação das zonas *Central*, *Esquerda* e *Direita* durante a inspecção.

Idealmente, a pá central deve estar alinhada verticalmente, apontando para cima, representando a configuração de *Y invertido*, considerada a mais adequada na literatura, pois minimiza o risco de obstrução visual por outras pás ou pela torre (Figura 4.9).

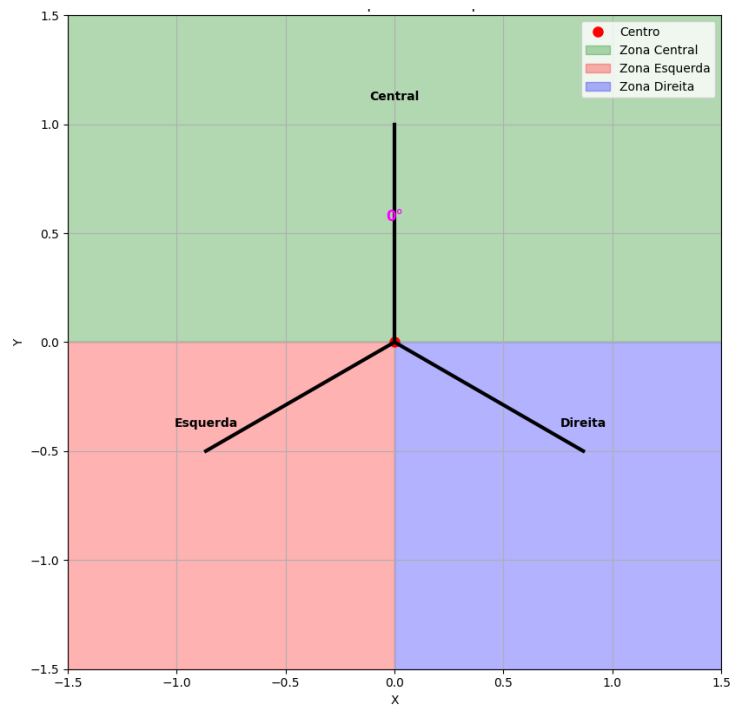


Figura 4.9: Configuração ideal das pás: pá central vertical, Y invertido. As zonas *Central*, *Esquerda* e *Direita* são claramente distinguíveis.

Pequenos desvios no ângulo de *roll* até  $\pm 30^\circ$  ainda permitem a identificação correcta das zonas (Figura 4.10). Desvios superiores, como  $40^\circ$ , tornam a classificação ambígua (Figura 4.11), comprometendo a inspecção.

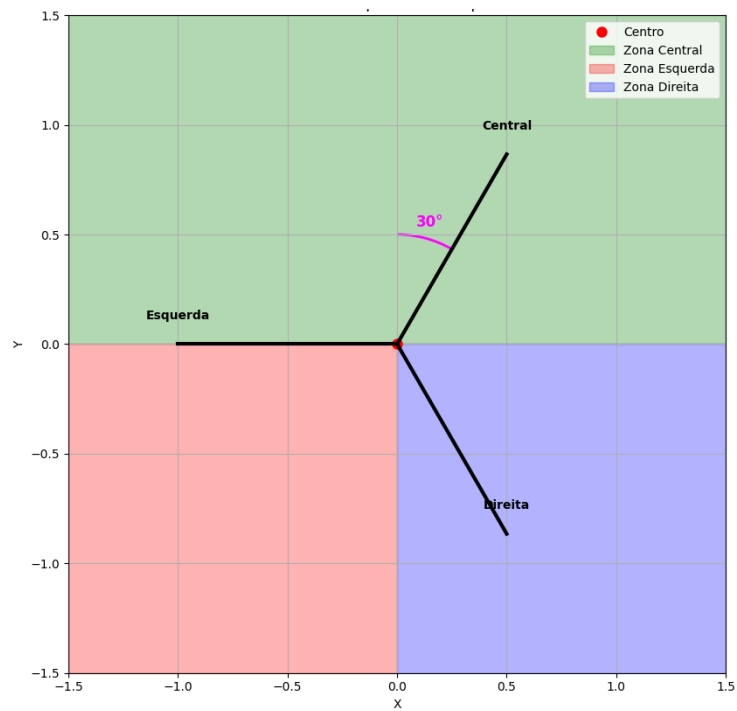


Figura 4.10: Configuração das pás com offset de  $30^\circ$  no ângulo de *roll*: a identificação das zonas *Central*, *Esquerda* e *Direita* permanece válida.

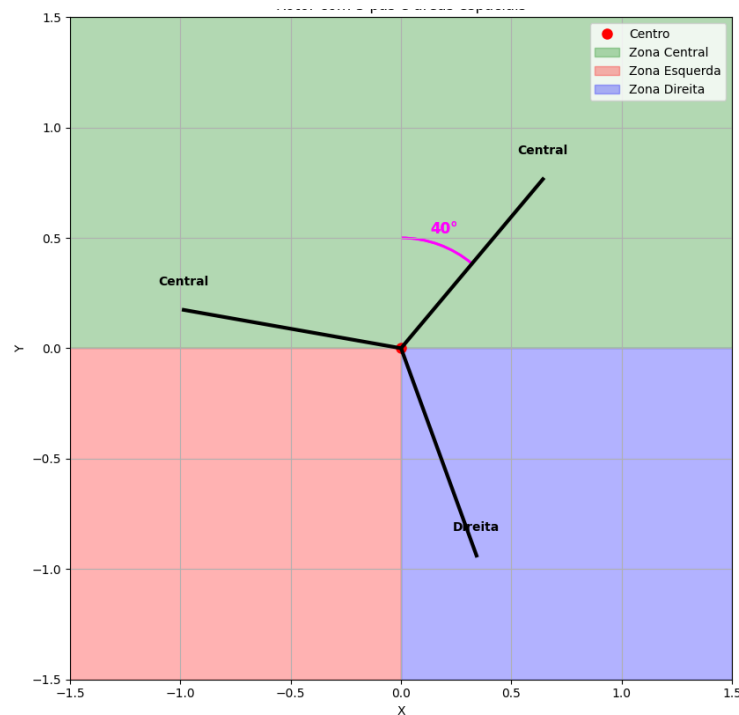


Figura 4.11: Configuração das pás com offset de  $40^\circ$  no ângulo de *roll*: a identificação das zonas torna-se ambígua, ultrapassando o limite admissível.

### 4.3.3 Condição 3: Limites do ângulo de *yaw* da nacelle

Esta condição refere-se à orientação da *nacelle*, correspondente ao grau de rotação em *yaw*. A estimativa inicial do ângulo de *yaw* é essencial para que cada UAV se aproxime do *hub* devidamente alinhado, garantindo uma inspeção frontal eficiente sobre o plano horizontal.

Caso o ângulo de *yaw* esteja fora dos limites previstos, a aproximação torna-se insegura, comprometendo a captura de imagens e os dados do LiDAR. Nos resultados experimentais serão apresentados os limites de *yaw* dentro dos quais os UAVs conseguem realizar a inspeção de forma segura e confiável.

### 4.3.4 Condição 4: Limites do ângulo de *pitch* das pás

A quarta condição refere-se ao *pitch* das pás, que define a orientação do lado de maior área da pá em relação ao UAV. Para inspeção segura, assume-se que, quando o UAV está alinhado frontalmente ao *hub*, a superfície de maior visibilidade esteja voltada para o UAV.

O **planeador de missões**, além de receber dados da turbina, coordena o ajuste do ângulo de *pitch* das pás. Quando o UAV conclui a Etapa 3 e sinaliza ao planeador

que está correctamente alinhado com o *hub*, a turbina ajusta o *pitch* para expor a superfície desejada.

Durante a Etapa 4, o UAV percorre a pá do *hub* até à extremidade. Ao atingir a ponta da pá, o planeador envia o comando à turbina para rodar o *pitch* em  $180^\circ$ , expondo a outra face ampla. O UAV percorre então a pá de volta ao *hub*, garantindo cobertura completa de ambos os lados sem necessidade de contornar o rotor, evitando colisões com a *nacelle* ou outras estruturas. Esta abordagem maximiza a segurança e a eficácia da inspeção.

## 4.4 Distância de Inspeção e Projeção do Campo de Visão

A definição da distância de inspeção é um aspeto crítico para assegurar que a superfície da turbina eólica seja captada integralmente pelos sensores embarcados. Neste trabalho, foi avaliada a projeção do campo de visão (*Field of View* – FOV) da câmara RGB e do sensor LiDAR, considerando a sua disposição no modelo adaptado do Typhoon H480.

O sensor **LiDAR Velodyne VLP-16** apresenta um FOV nominal horizontal de  $360^\circ$  e vertical de  $30^\circ$ . Contudo, no caso da inspeção das pás, adotou-se um enquadramento restrito com FOV horizontal de  $20^\circ$  e vertical de  $30^\circ$ , permitindo focar na área de interesse e garantir precisão nos dados.

Para a **câmara RGB CGO3** (resolução  $640 \times 360$  px), foi assumido um FOV horizontal de aproximadamente  $57.3^\circ$  (equivalente a 1 radiano). Mantendo a proporção da imagem, o FOV vertical é dado por:

$$\text{FOV}_v = 2 \arctan \left( \frac{H}{W} \cdot \tan \left( \frac{\text{FOV}_h}{2} \right) \right), \quad (4.1)$$

com  $W = 640$  e  $H = 360$ , resultando em  $\text{FOV}_v \approx 33^\circ$ .

A largura ( $L$ ) e altura ( $A$ ) do campo de visão de qualquer sensor, para uma dada distância  $d$ , são calculadas através de:

$$L = 2d \cdot \tan \left( \frac{\text{FOV}_h}{2} \right), \quad A = 2d \cdot \tan \left( \frac{\text{FOV}_v}{2} \right), \quad (4.2)$$

onde  $\text{FOV}_h$  e  $\text{FOV}_v$  correspondem ao campo de visão horizontal e vertical, respetivamente.

Como evidenciado na Figura 4.12, uma distância de  $d = 12$  m permite que a câmara capture uma largura de 13.12 m e altura de 7.37 m, enquanto o LiDAR cobre  $4.23 \times 6.43$  m. Esta distância é suficiente para inspeccionar integralmente a largura da pá ( $\approx 4$  m) e definir o passo de inspeção da pá, garantindo algum *overlap* nas capturas consecutivas.

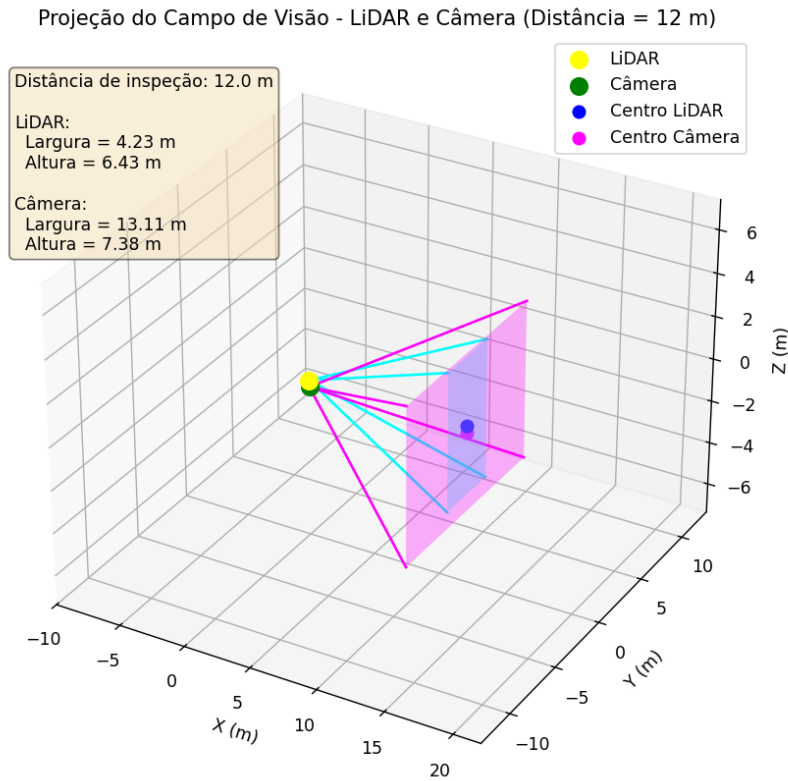


Figura 4.12: Projeção do campo de visão da câmara RGB e do LiDAR (FOV horizontal limitado a  $20^\circ$ ) a 12 m de distância

No entanto, o pipeline na Etapa 3, que inclui a segmentação utilizando DBSCAN, exige uma distância mínima de aproximadamente 25 m para que a nuvem de pontos seja corretamente processada. A largura e altura do campo de visão da câmara RGB a esta distância podem ser calculadas através da Equação 4.2:

$$L = 2 \cdot d \cdot \tan\left(\frac{\text{FOV}_h}{2}\right) = 2 \cdot 25 \cdot \tan\left(\frac{57.3^\circ}{2}\right) \approx 27.1 \text{ m},$$

$$A = 2 \cdot d \cdot \tan\left(\frac{\text{FOV}_v}{2}\right) = 2 \cdot 25 \cdot \tan\left(\frac{33^\circ}{2}\right) \approx 14.4 \text{ m}.$$

Assim, a câmara RGB cobre uma área de aproximadamente  $27.1 \times 14.4$  m a 25 m de distância, o que permite definir o espaçamento vertical entre as posições consecutivas do UAV durante a inspeção da torre, garantindo cobertura completa com *overlap* adequado entre capturas consecutivas.

Para o LiDAR Velodyne VLP-16, não é necessário efetuar este cálculo nesta fase, uma vez que, na Etapa 3, o sensor opera com um FOV horizontal de  $360^\circ$ , assegurando a captação completa da nuvem de pontos.

O cálculo das dimensões do campo de visão da câmara RGB e do LiDAR, para diferentes distâncias de inspeção, é apresentado no Anexo 7.2.

## 4.5 Etapa 0 — Descolagem

A etapa inicial corresponde ao procedimento de **descolagem** do UAV, em que o veículo assume o modo *offboard* e começa a receber *setpoints* de posição para o controlador de voo. Antes de levantar, o *script onboard* realiza inicializações essenciais para garantir operação autônoma segura.

O UAV é iniciado exclusivamente através do **planeador de missões**, que ativa o *script onboard* e envia como argumento a pá a inspecionar ("esquerda", "direita" ou "central"). O *script* cria um diretório específico para armazenar os dados da missão. Caso nenhum argumento seja fornecido, o programa termina com uma mensagem de erro.

São definidos objetos globais, como mensagens de pose (`PoseStamped`) e controle do gimbal (`MountControl`), usando o *frame* "map". Configuram-se **publicadores** para envio de posições (`uav/mavros/setpoint_position/local`), comandos de gimbal (`uav/mavros/mount_control/command`) e estado da missão (`/uav/status`).

**Subscritores** monitorizam o estado do UAV (`uav/mavros/state`), pose local (`uav/mavros/local_position/pose`) e nuvem de pontos do LiDAR (`/velodyne_points`).

Também são inicializados **serviços ROS** para armamento (`uav/mavros/cmd/arming`) e alteração de modo (`uav/mavros/set_mode`). O sistema espera pela ligação com a Flight Control Unit (FCU) (*Flight Control Unit*) antes de prosseguir, garantindo comunicações estáveis.

Para captura de vídeo e detecção de marcadores ArUco, são utilizadas bibliotecas como OpenCV (`cv2.aruco`) para processamento de imagens, GStreamer (`Gst`) para streaming de vídeo, NumPy (`numpy`) para operações matemáticas e scikit-learn (`sklearn`) para algoritmos como DBSCAN e RANSAC. A classe `Video` gere a captura de frames via UDP, enquanto a classe `MarkerDetectionSystem` detecta marcadores *ArUco* com IDs únicos, permitindo identificação precisa dos UAVs na plataforma.

Após todas as as inicializações efetuadas, UAV sobe até à altitude de segurança predefinida, verificando a posição através da função `reached_target` (tolerância de 0.5 m). Após confirmação do sucesso da decolagem, o estado atualizado é publicado no tópico `/uav/status`.

A Figura 4.13 mostra a plataforma de aterragem com os marcadores ArUco, cada um com ID único para detecção e alinhamento visual.

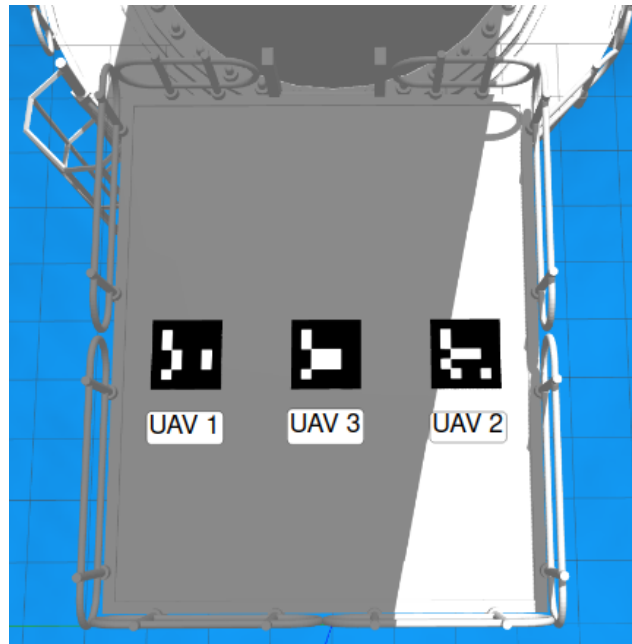


Figura 4.13: Vista superior da plataforma de aterragem com os marcadores *ArUco* individuais.

## 4.6 Etapa 1 – Alinhamento com a torre

Após a descolagem (Etapa 0), o UAV encontra-se a uma altitude de aproximadamente 20 m, considerada altura segura de operação inicial. A partir desta posição, inicia-se a aproximação à torre da turbina, recorrendo às medições do sensor LiDAR.

Nesta fase, assume-se que o UAV ainda não possui informação precisa sobre a orientação da nacelle e do hub, mas dispõe de uma estimativa inicial da sua direção global. O objetivo é realizar um alinhamento preliminar, próximo da base da torre, para posicionar o UAV corretamente antes da subida na etapa seguinte.

A detecção da torre é realizada aplicando DBSCAN sobre a nuvem de pontos do LiDAR. Apenas são considerados *clusters* cuja largura mínima seja de 5 m, compatível com a dimensão real da torre. Se nenhum cluster válido for detetado durante várias iterações consecutivas, a missão é abortada ativando a Etapa 5.

Uma vez identificada a torre, o UAV estabelece uma distância de referência de 25 m no eixo  $x$ , considerada segura para inspeção. São calculados dois erros principais:

- erro de yaw: diferença angular entre a orientação atual do UAV e a direção do centro estimado da torre, corrigido primeiro;
- erro em  $x$ : diferença entre a distância desejada e a distância efetiva ao centro da torre, corrigido apenas quando o yaw estiver dentro da tolerância.

O controle é realizado com dois PID independentes em sequência: primeiro o regulador de yaw até alinhar o UAV com a torre; depois, se o erro de yaw for  $\leq 5^\circ$ , o regulador de  $x$  ajusta a distância radial.

Depois de corrigidos os erros, o UAV executa passos incrementais em torno da torre de forma a alinhar com a estimativa inicial do ângulo *yaw* da *nacelle*. Considerando que a torre tem 3.5 m de raio e o drone deve manter 25 m de distância da torre, o raio da trajetória circular é  $r_{tower} = 3.5 + 25 = 28.5$  m. O processo de estimação e correção dos erros é repetido a cada passo até atingir o ângulo total predefinido.

A etapa é considerada concluída quando:

- $|erro_x| \leq 0.5$  m;
- $|erro_{yaw}| \leq 5^\circ$ ;
- a rotação total em torno da torre atinge o ângulo predefinido.

O pseudocódigo do procedimento é apresentado a seguir:

```

1  Receber nuvem de pontos do LiDAR
2  Aplicar DBSCAN com parâmetros:  $\text{eps} = 0.75$ ,  $\text{min\_samples} = 60$ 
3  Remover clusters ruidosos ( $\text{label} = -1$ )
4  if nenhum cluster válido then
5      Incrementar contador de falhas
6      if falhas consecutivas  $\geq 20$  then abortar missão
7      else
8          Selecionar cluster mais próximo da origem
9          Calcular centroide  $(x_{center}, y_{center})$  e largura  $y_{max} - y_{min}$ 
10         if largura do cluster  $\geq 5$  m then
11             Estimar erro de yaw:  $\text{yaw\_target} = \arctan 2(y_{center}, x_{center})$ 
12             Aplicar PID de yaw para alinhar UAV com torre
13             if  $|\text{erro\_yaw}| \leq 5^\circ$  then
14                 Estimar erro em X:  $x_{target} - x_{center}$ 
15                 Aplicar PID de X para ajustar distância
16             end if
17             Executar passo incremental na trajetória circular de raio  $r_{tower} =$ 
18             28.5 m
19             Atualizar rotação acumulada
20             if erro em X  $\leq 0.5$  m, erro de yaw  $\leq 5^\circ$  e rotação total alcançada
21         then
22             Avançar para Etapa 2
23         end if
24     else
25         Incrementar contador de falhas de largura
26         if contador  $\geq 10$  then abortar missão
27         end if
28     end if

```

Algoritmo 4.1: Etapa 1 – Aproximação e alinhamento preliminar

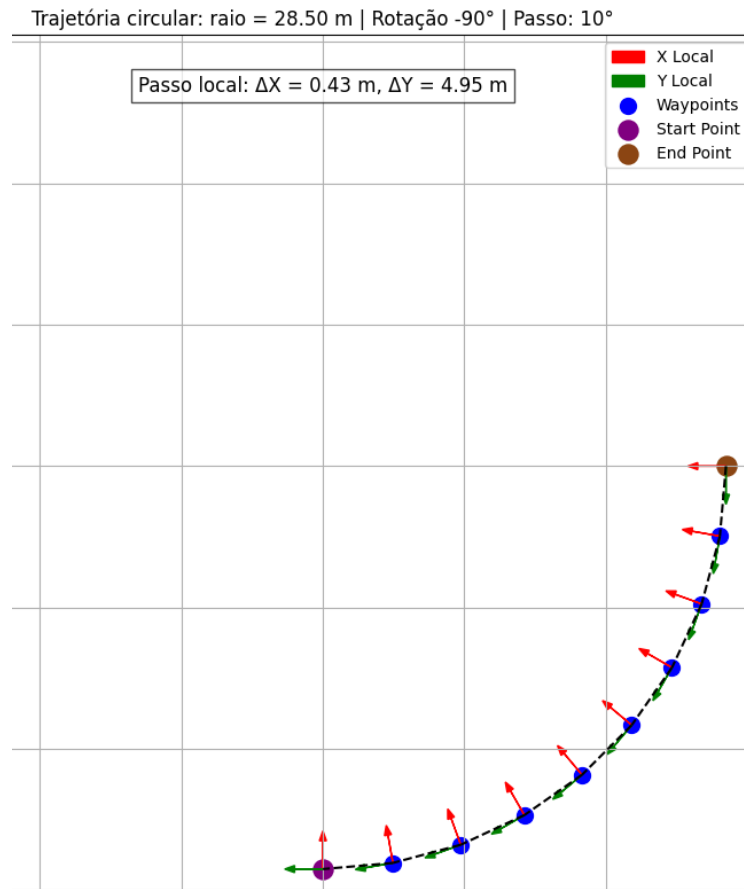


Figura 4.14: Etapa 1: Exemplo de trajetória circular seguida pelo UAV

## 4.7 Etapa 2 – Subida até ao *hub*

Concluída a fase de alinhamento preliminar (Etapa 1), o UAV inicia a subida vertical ao longo da torre, mantendo a referência obtida pelo LiDAR. O objetivo desta etapa é alcançar o plano do *hub*, garantindo que a estrutura da torre é continuamente monitorizada e que as imagens captadas apresentem algum *overlap*.

Tal como na etapa anterior, a deteção da torre é efetuada com recurso ao algoritmo de clusterização *DBSCAN*, com parâmetros definidos como  $\text{eps} = 0.75$  e  $\text{min\_samples} = 60$ . Apenas são considerados válidos os *clusters* cuja largura mínima seja de 5 m, condição que assegura correspondência com a dimensão real da torre. Caso não seja detetado nenhum cluster válido durante iterações consecutivas, a missão é abortada e o sistema transita para a Etapa 5 (retorno seguro).

O UAV estabelece como distância alvo no eixo  $x$  o valor de 25 m, idêntico ao definido na Etapa 1. O processo de controlo segue a mesma lógica sequencial:

1. correção do erro de yaw, de modo a manter o UAV orientado para o centro da torre;

2. após o yaw se encontrar dentro da tolerância ( $\leq 5^\circ$ ), correção do erro em  $x$ , ajustando a distância radial para 25 m.

Uma vez alinhado, o UAV executa incrementos de subida vertical tendo em conta o campo de visão projetado, garantindo a existência de *overlap* entre as imagens adquiridas. Assim, a cada incremento vertical é registada uma imagem da superfície da torre, assegurando cobertura contínua da inspeção.

A etapa é considerada concluída quando o LiDAR deteta mais de dois *clusters* distintos, indicando que, além da torre, surgem também elementos adicionais como o rotor e *nacelle*. Esta condição é interpretada como evidência de que o UAV atingiu o plano do *hub*, encerrando assim a Etapa 2.

O procedimento pode ser descrito no seguinte pseudocódigo:

```

1 Receber nuvem de pontos do LiDAR
2 Aplicar DBSCAN com eps = 0.75, min_samples = 60
3 if nenhum cluster válido com largura  $\geq 5$  m then
4     Incrementar contador de falhas
5     if falhas consecutivas  $\geq 20$  then abortar missão (Etapa 5)
6     else
7         if número de clusters  $> 2$  then
8             Avançar para Etapa 3
9         else
10            Selecionar cluster mais próximo (torre)
11            Estimar erro de yaw e aplicar  $PID_{yaw}$ 
12            if  $|erro_{yaw}| \leq 5^\circ$  then
13                Estimar erro em X e aplicar  $PID_x$ 
14            end if
15            if  $|erro_{yaw}| \leq 5^\circ$  e  $|erro_x| \leq 0.5$  m then
16                Executar incremento vertical
17                Registrar captura de imagem da torre
18            end if
19        end if
20    end if

```

Algoritmo 4.2: Etapa 2 – Subida ao longo da torre

A Figura 4.15 ilustra o comportamento do UAV durante a Etapa 2. Observa-se que o veículo mantém a distância alvo de 25 m relativamente ao centro da torre (linha vermelha), enquanto a largura estimada do cluster (linha verde) confirma a deteção da estrutura com dimensões compatíveis com a torre real. Durante a subida, esta configuração é preservada a cada incremento vertical, assegurando que o UAV permanece alinhado com a torre e que as imagens adquiridas apresentam cobertura contínua.

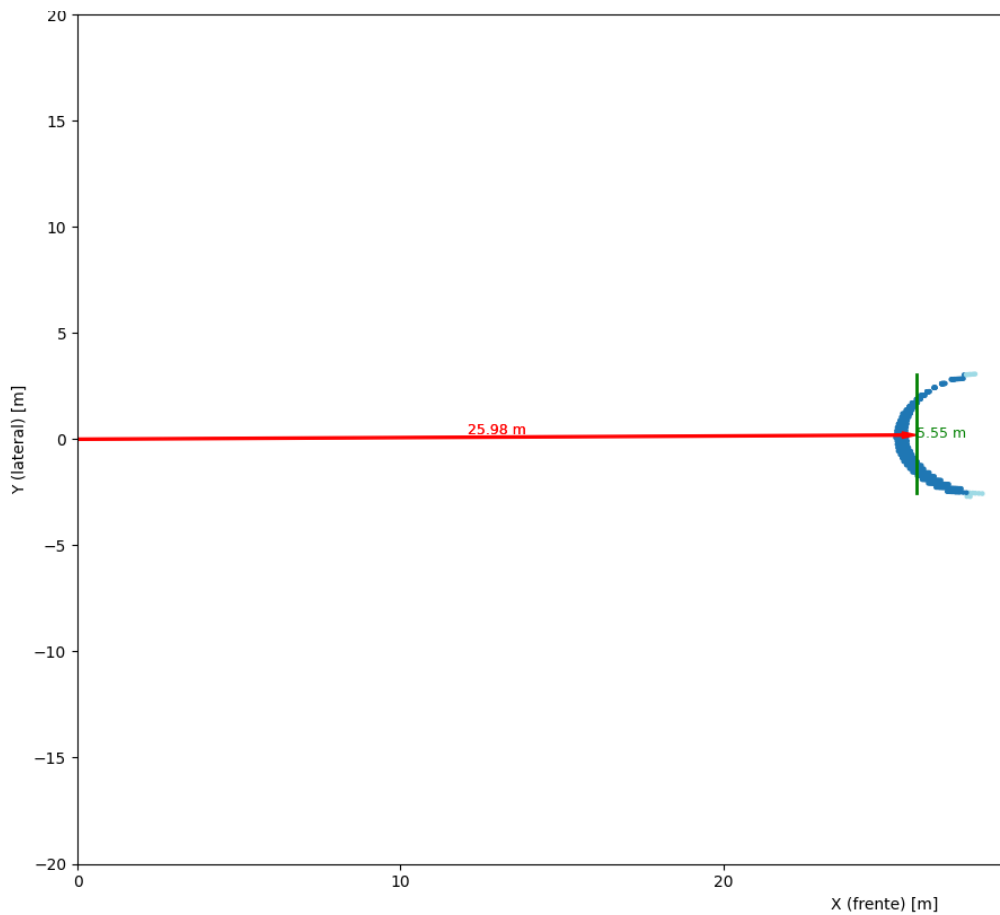


Figura 4.15: Etapa 2: Subida incremental mantendo distância igual ou superior a 25 m da torre.

## 4.8 Etapa 3 – Alinhamento com o *hub*

Ao atingir o plano do *hub* (conclusão da Etapa 2), o UAV encontra-se numa zona onde o LiDAR já consegue detetar múltiplos *clusters*, correspondentes não apenas à torre mas também às pás da turbina. Nesta etapa, o objetivo é realizar o ajuste fino de pose para alinhar o UAV com a estrutura global e centrar-se relativamente às pás, antes de avançar para a inspeção detalhada.

A deteção das pás é realizada através do algoritmo *DBSCAN*, aplicado à nuvem de pontos do LiDAR. São identificados pelo menos dois *clusters* distintos, que correspondem às pás mais à esquerda e mais à direita no referencial do sensor. Os centróides destes *clusters* são então utilizados como referência para duas fases sequenciais de alinhamento, ilustradas na Figura 4.16 e na Figura 4.17.

1. **Alinhamento em yaw:** conforme ilustrado na Figura 4.16, o UAV corrige iterativamente a sua orientação até que as coordenadas  $x$  dos centróides esquerdo e direito sejam aproximadamente iguais. Esta condição garante que o

UAV está alinhado com o plano do *hub*.

2. **Centragem em  $y$ :** uma vez estabilizado o *yaw*, é calculado o erro lateral como a diferença entre as coordenadas  $y$  absolutas dos *clusters* esquerdo e direito. O UAV aplica um controlador PID para deslocar-se lateralmente até que o ponto médio entre os dois *clusters* coincida com o eixo central do veículo, como demonstrado na Figura 4.17.

Concluídas estas duas fases, o UAV encontra-se posicionado e centrado no plano do *hub*, pronto para se deslocar em direção à pá selecionada para inspeção (esquerda, direita ou central).

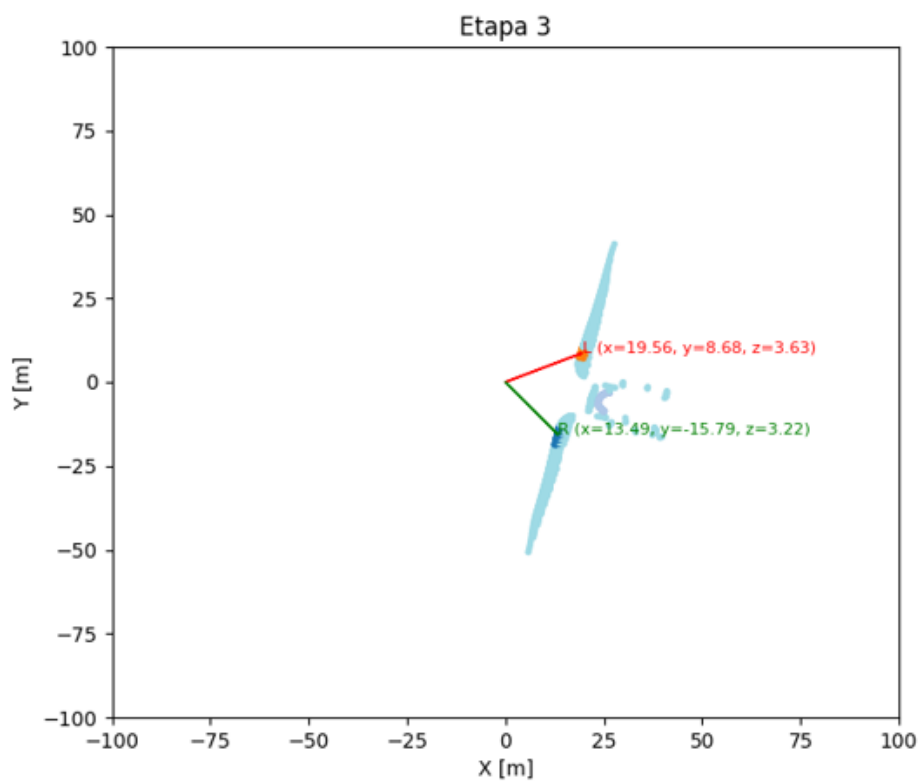


Figura 4.16: Etapa 3: Alinhamento em *yaw*.

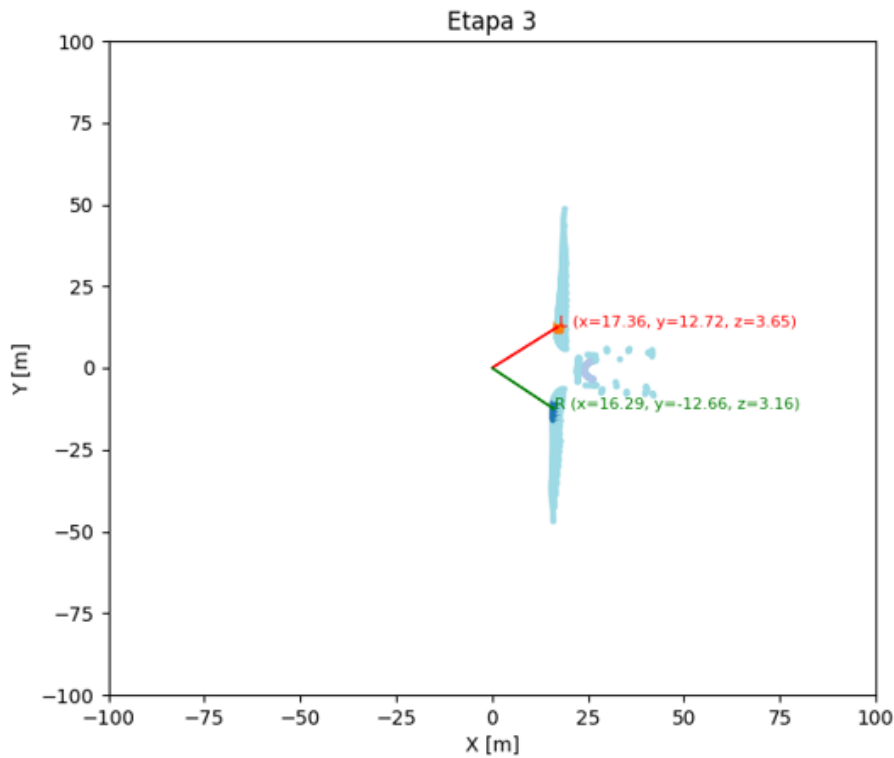


Figura 4.17: Etapa 3: Alinhamento em  $y$  após alinhamento em  $yaw$ .

O procedimento pode ser descrito no seguinte pseudocódigo:

```

1  Receber nuvem de pontos do LiDAR
2  Aplicar DBSCAN para identificar clusters
3  if menos de 2 clusters válidos then
4      Incrementar contador de falhas
5      Subir passo de 1 m em  $z$ 
6      if falhas consecutivas > limiar then abortar missão (Etapa 5)
7      else
8          Selecionar clusters mais à esquerda e mais à direita
9          Calcular centróides esquerdo e direito
10         Estimar erro de yaw =  $x_{left} - x_{right}$ 
11         Aplicar  $PID_{yaw}$  até  $|erro_{yaw}| \leq \text{limiar}$ 
12         if alinhamento em yaw concluído then
13             Estimar erro lateral em  $y = |y_{left}| - |y_{right}|$ 
14             Aplicar  $PID_y$  até  $|erro_y| \leq \text{limiar}$ 
15             Etapa concluída  $\rightarrow$  avançar para Etapa 4
16         end if
17     end if

```

Algoritmo 4.3: Etapa 3 – Ajuste no plano do *hub*

## 4.9 Etapa 4 — Inspeção da pá

A Etapa 4 corresponde à fase mais relevante da missão, dedicada à inspeção das pás da turbina eólica.

A nuvem de pontos adquirida pelo sensor LiDAR é inicialmente filtrada para reter apenas os pontos dentro do campo de visão útil. De seguida, aplica-se o algoritmo *DBSCAN* para a identificação de *clusters* que correspondem às superfícies planas das pás. Uma vez identificado um cluster válido, utiliza-se o método *RANSAC* para ajustar um plano, a partir do qual são extraídos: (i) o vetor normal, (ii) os ângulos de roll, pitch e yaw da pá, (iii) as dimensões principais (largura e altura) e (iv) o centróide da superfície. A Figura 4.18 ilustra este processo, mostrando o plano ajustado, o centróide, a origem do LiDAR e os próximos passos de inspeção.

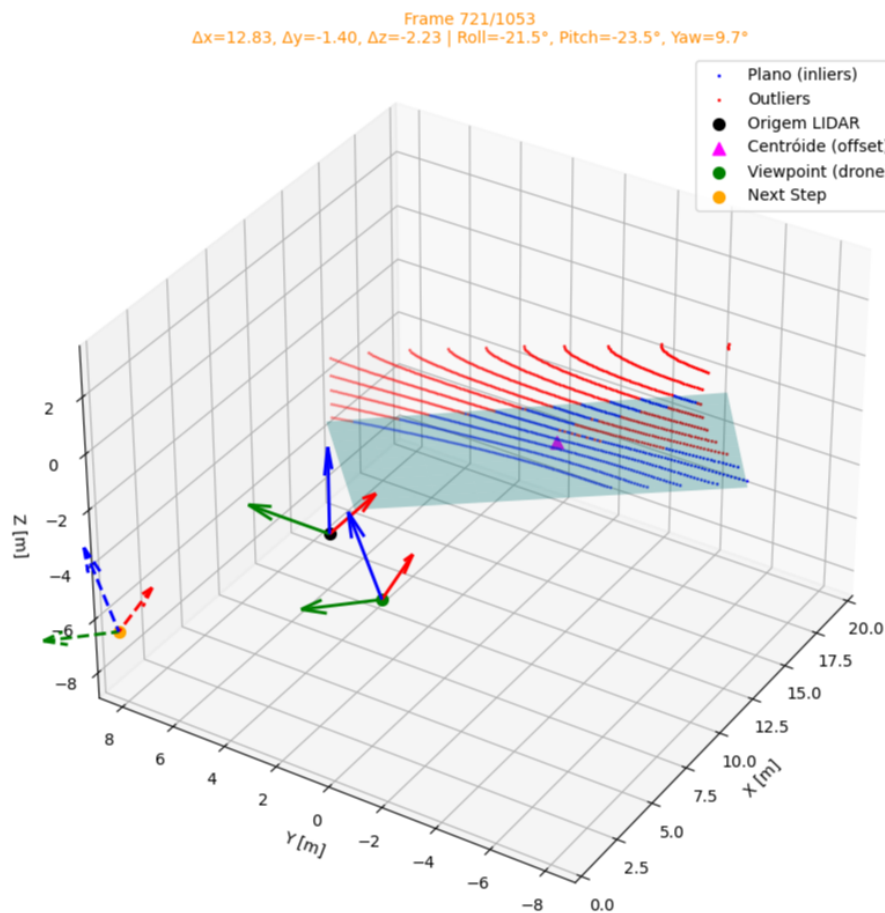


Figura 4.18: Etapa 4: Ajuste no plano das pás com *RANSAC*.

O UAV alinha-se em relação ao plano estimado, corrigindo a sua pose nos diferentes eixos de acordo com a pá em análise:

- Em todas as pás: correção do yaw e da distância ao plano no eixo  $X$  (mantendo cerca de 12 m).

- Na pá esquerda e direita: ajuste adicional da altitude ( $Z$ ).
- Na pá central: ajuste do deslocamento lateral ( $Y$ ).

Após o alinhamento, o UAV progride ao longo da pá, definindo *waypoints* que respeitam a inclinação real obtida pelo roll do plano. O espaçamento entre *waypoints* é definido de modo a garantir **overlap horizontal ou vertical** entre imagens consecutivas, calculado a partir da janela útil da câmara (campo de visão e distância de 12 m ao plano da pá). Assim, assegura-se cobertura contínua da superfície sem falhas de inspeção.

Durante a etapa Etapa 4, dependendo número de *clusters* detectados, podem ocorrer os seguintes casos:

- **Primeira perda de detecção:** considera-se que o UAV atingiu a ponta da pá. O planeador emite um comando à turbina para rodar o pitch da pá, expondo a face oposta para inspeção. Em simultâneo, o UAV recua no sentido oposto ao último movimento, tentando restabelecer a detecção da pá.
- **Nenhum *cluster* detectado após a manobra:** se, mesmo após o movimento de recuo, o UAV não detectar qualquer *cluster*, a Etapa 5 (retorno seguro) é ativada, indicando que a pá não foi localizada e a missão deve ser interrompida de forma segura.
- **Deteção de múltiplos *clusters*:** quando são detectados múltiplos *clusters*, isto pode significar que o UAV já regressou próximo do *hub*. Neste caso, a Etapa 5 é igualmente ativada, seguindo a sequência normal de retorno seguro.

O procedimento pode ser descrito no seguinte pseudocódigo:

```

1 Inicializar controladores PID e parâmetros
2 while missão ativa do
3   Capturar nuvem de pontos do LiDAR
4   Filtrar pontos no campo de visão
5   Aplicar DBSCAN para identificar clusters
6   if nenhum cluster encontrado then
7     if primeira perda then
8       Assumir ponta da pá atingida
9       Enviar comando → rodar pitch da pá
10    else
11      Incrementar contador de perda
12      if perda prolongada then recuar em direção ao hub
13        if perda total then avançar para Etapa 5
14        end if
15      else if mais de um cluster encontrado then
16        Abortar inspeção (condição insegura)
17        Avançar para Etapa 5
18      else
19        Aplicar RANSAC e estimar plano da pá
20        Extrair roll, pitch, yaw e centróide
21        Alinhar UAV em yaw e eixo  $X$ 
22        if pá esquerda ou direita then
23          Ajustar altitude ( $Z$ )
24        else if pá central then
25          Ajustar deslocamento lateral ( $Y$ )
26        end if
27        if alinhamento bem sucedido then
28          Capturar imagem
29          Calcular próximo waypoint garantindo overlap horizontal
30          Avançar para o próximo ponto de inspeção
31        end if
32      end if
33

```

Algoritmo 4.4: Etapa 4 — Inspeção das pás

## 4.10 Etapa 5 — Descida controlada

A Etapa 5 é ativada em dois cenários: (i) quando a inspeção de uma pá termina com sucesso (após a Etapa 4), ou (ii) quando a inspeção é abortada por motivos de segurança. O objetivo principal desta fase é permitir que o UAV realize uma descida controlada e segura, mantendo-se afastado da turbina e atento a possíveis obstáculos adicionais.

A área em que normalmente se inicia a Etapa 5 corresponde a uma zona de maior probabilidade de encontro entre UAVs que se encontram próximos do hub durante as Etapas 2, 3 e 4. Consequentemente, é essencial que o UAV, nesta fase, não só

mantenha a distância de segurança em relação à turbina, mas também seja capaz de detectar pequenos obstáculos e reagir em tempo útil, evitando colisões.

O sensor LiDAR mantém-se ativo, mas nesta etapa o algoritmo *DBSCAN* é aplicado com parâmetros mais sensíveis (raio  $\varepsilon$  reduzido e menor número de pontos mínimos). Este ajuste permite a identificação de *clusters* de menor dimensão, aumentando a capacidade de deteção de objetos potencialmente perigosos, como outros UAVs em operação.

O UAV executa dois controlos em paralelo:

- Controlo lateral: o yaw é ajustado continuamente de forma a orientar a frente local para o centróide do cluster mais próximo, mantendo uma distância horizontal de segurança. Para isso, é utilizado um controlador PID que atua sobre o eixo  $X$  (distância ao objeto) e yaw (orientação).
- Descida vertical: a descida no eixo  $Z$  é realizada com base na estimação interna de pose. A precisão não é crítica, pois o objetivo é apenas reduzir a altitude de forma segura até cerca de 20 m.

A etapa considera-se concluída quando o UAV atinge uma altitude inferior a 20 m, mantendo simultaneamente uma distância lateral de pelo menos 10 m em relação ao obstáculo mais próximo. Nessa condição, a missão progride para a Etapa 6.

O procedimento pode ser descrito no seguinte pseudocódigo:

```

1  while missão ativa do
2    Capturar nuvem de pontos do LiDAR
3    Aplicar DBSCAN com parâmetros refinados
4    if nenhum cluster detetado then
5      Continuar descida com base na pose interna
6    else
7      Selecionar cluster mais próximo
8      Calcular centróide e distância
9      Ajustar yaw para apontar ao centróide (PID)
10     if distância ao cluster < 10 m then
11       Recuar para manter afastamento
12     end if
13   end if
14   Atualizar altitude de descida
15   if altitude  $\leq$  20 m e distância lateral > 10 m then
16     Concluir Etapa 5 e avançar para Etapa 6
17   end if
18 end while=0

```

Algoritmo 4.5: Etapa 5 — Descida segura

## 4.11 Etapa 6 — Localização da plataforma de aterragem

A Etapa 6 tem como objetivo localizar a plataforma de aterragem através da deteção de marcadores visuais do tipo *ArUco*. Estes marcadores bidimensionais permitem determinar a pose relativa entre o UAV e a plataforma com elevada precisão. Cada UAV possui um identificador único de *ArUco*, o que assegura que apenas reconhece a sua própria plataforma de aterragem.

O procedimento desta etapa é semelhante ao da Etapa 1: o UAV executa uma trajetória circular em torno da torre, mantendo altitude constante e utilizando o LiDAR para identificar o cluster correspondente à torre. A diferença principal é que, em vez de completar uma rotação para alinhar com o *yaw* inicial estimado, o UAV continua a deslocar-se circularmente até que o marcador *ArUco* seja detetado pela câmara RGB.

O raio da trajetória circular é definido como  $r = 8.5$  m, correspondendo à soma da distância entre o *ArUco* e a superfície da torre (5 m) com o raio da torre (3.5 m). Durante o movimento, dois controladores PID asseguram o controlo:

- **Yaw:** garante que o UAV mantém alinhamento angular com o centróide da torre;
- **Distância radial (X):** assegura que o UAV se mantém no raio desejado.

A etapa considera-se concluída quando o marcador *ArUco* correspondente ao UAV é identificado na imagem da câmara, momento em que se avança para a Etapa 7 (aproximação e aterragem).

O procedimento pode ser descrito no seguinte pseudocódigo:

```

1  while missão ativa do
2      Receber nuvem de pontos do LiDAR
3      Identificar cluster da torre via DBSCAN
4      Calcular erro de yaw e erro radial ( $X$ )
5      Aplicar PID de yaw
6      if yaw dentro da tolerância then
7          Aplicar PID radial para manter  $r = 8.5$  m
8      end if
9      if frame disponível da câmara then
10         Processar frame
11         Procurar marcador ArUco com ID do UAV
12         if marcador detetado then
13             Etapa concluída → avançar para Etapa 7
14             break
15         end if
16     end if
17     Executar incremento circular em torno da torre ( $\Delta\theta$ )
18     Atualizar pose do UAV
19 end while

```

Algoritmo 4.6: Etapa 6 — Localização da plataforma de aterragem

## 4.12 Etapa 7 — Aterragem

A última etapa da missão corresponde ao procedimento de aterragem na plataforma residente. O UAV utiliza a sua **câmara visual** orientada verticalmente para baixo (através do gimbal, ajustado a  $-90^\circ$ ) para detetar o marcador *ArUco* correspondente à sua plataforma. Cada UAV possui um identificador (*ID*) de *ArUco* único, o que elimina ambiguidades quando várias plataformas se encontram na mesma área de operação.

A deteção é realizada com recurso à biblioteca `OpenCV`, que disponibiliza funções para identificar marcadores *ArUco* e estimar a sua pose tridimensional. Para tal, são utilizados os parâmetros intrínsecos da câmara (matriz de calibração e coeficientes de distorção). A partir das coordenadas dos cantos do marcador, calcula-se a posição relativa (erros em  $x$ ,  $y$ ,  $z$ ) e a orientação (erro em yaw).

O controlo é feito com controladores PID independentes para cada eixo:

- **Correção lateral ( $x, y$ ):** o UAV desloca-se de forma a centrar o marcador no campo de visão da câmara.
- **Correção de yaw:** o UAV ajusta a sua orientação até alinhar o eixo longitudinal com o marcador.
- **Descida ( $z$ ):** o UAV reduz gradualmente a altitude, mantendo-se alinhado com o marcador.

Quando os erros em  $x$ ,  $y$  e yaw se encontram dentro de uma margem de tolerância e a altura é suficientemente baixa, o UAV considera-se alinhado sobre a plataforma. Neste momento, é emitido o comando `AUTO.LAND`, que inicia a descida automática final e o pouso controlado.

A Figura 4.19 exemplifica este processo, mostrando o UAV a alinhar-se com o marcador *ArUco* instalado na plataforma.

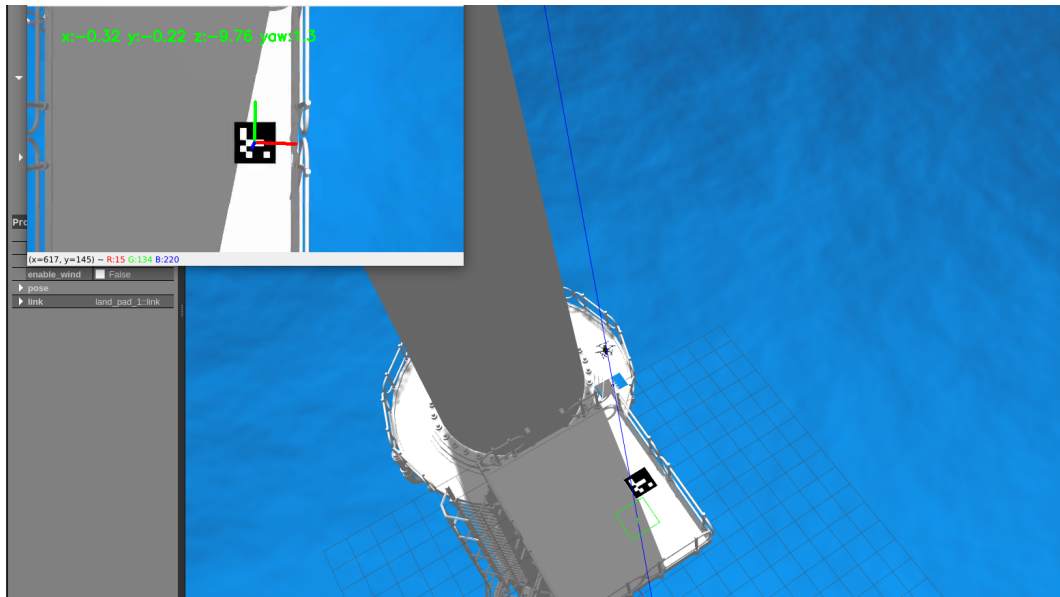


Figura 4.19: Etapa 7: Aterragem no *ArUco* exclusivo da sua plataforma.

O procedimento pode ser descrito no seguinte pseudocódigo:

```

1 Rodar gimbal para baixo ( $-90^\circ$ )
2 while missão ativa e não concluída do
3   if frame disponível da câmara then
4     Detetar marcador ArUco com OpenCV
5     if marcador com ID do UAV encontrado then
6       Estimar pose relativa (erros  $x, y, z, yaw$ )
7       Aplicar PID para corrigir  $x, y, yaw$ 
8       if erros  $x, y, yaw$  dentro da tolerância then
9         Reduzir altitude ( $z$ )
10      end if
11      if  $z \leq$  limiar (altura segura para pouso) then
12        Enviar comando AUTO.LAND
13        Missão concluída
14        break
15      end if
16    else
17      Manter posição e procurar marcador
18    end if
19  end if
20 end while

```

Algoritmo 4.7: Etapa 7 — Aterragem no marcador *ArUco*

## 4.13 Planeador de Missão

O planeador de missão constitui o módulo central responsável pela coordenação dos UAVs na inspeção cooperativa de uma turbina eólica. Este módulo corre em modo *offboard* e comunica diretamente com cada UAV através de tópicos ROS, bem como com a infraestrutura da turbina, de forma a orquestrar as diferentes etapas da missão. Embora nesta implementação não exista ligação direta a sensores da turbina para recolha de parâmetros de *roll*, *pitch* e *yaw*, assume-se que esta interface poderá ser futuramente integrada, permitindo sincronização com o estado da máquina.

A principal função do planeador é decidir **quando** cada UAV deve ser lançado, **qual** região da turbina deve inspecionar (atribuição de pás) e **quando** a missão individual deve ser considerada concluída. Para tal, o sistema mantém um estado interno com a última etapa atingida por cada UAV, registado em tempo real a partir do tópico `/status`. Paralelamente, o planeador subscreve as posições de cada UAV, armazenando a trajetória em ficheiros para posterior análise.

A orquestração segue uma lógica sequencial ou hierárquica. Por exemplo, pode ser definido que um UAV inicia a inspeção de uma pá, outro UAV só é lançado quando o primeiro atinge determinada etapa, e assim sucessivamente. No entanto, a ordem de lançamento dos drones pode ser ajustada de forma a determinar a estratégia mais eficiente.

O planeador pode ser descrito em pseudocódigo formal:

```
1 Iniciar nó ROS do planeador
2 while missão não concluída do
3   for cada UAV do
4     if UAV ainda não iniciado e condição de lançamento satisfeita then
5       Lançar UAV
6     end if
7   end for
8   for cada UAV ativo do
9     Subscriver posição e estado (/status)
10    Registrar trajetória em ficheiro
11    if UAV completou todas as etapas da sua missão then
12      Marcar UAV como concluído
13    end if
14  end for
15 end while
16 Missão concluída
```

Algoritmo 4.8: Planeador de Missão — Lançamento e Monitorização de UAVs

## Capítulo 5

# Resultados

Este capítulo apresenta os resultados experimentais obtidos durante a avaliação do sistema proposto para inspeção autónoma de turbinas eólicas *offshore* utilizando UAVs residentes. Os ensaios foram estruturados em dois grupos principais, permitindo uma análise abrangente da performance do pipeline de missão e da estratégia multi-UAV.

O primeiro grupo de testes foca-se na execução individual das etapas do pipeline de missão, avaliando os limites operacionais do sistema em função dos ângulos críticos da turbina, nomeadamente:

- o ângulo de *yaw* da nacelle, que influencia a orientação inicial do UAV em relação ao eixo da turbina;
- o ângulo de *roll* das pás, que impacta a visibilidade da torre e a capacidade de inspeção completa.

Estes testes permitiram validar a robustez do sistema e identificar condições inviáveis para inspeção.

O segundo grupo de testes aborda a coordenação de múltiplos UAVs, com foco na otimização da estratégia de lançamento e execução faseada. Nestes ensaios, foram exploradas diferentes abordagens de planeamento, culminando na estratégia final adotada.

Cada secção subsequente detalha os resultados obtidos nos cenários de ângulo de *yaw* da nacelle (Seção 5.1), ângulo de *roll* das pás (Seção 5.2) e na coordenação de múltiplos UAVs (Seção 5.3). Os resultados incluem trajetórias 3D, distâncias

percorridas, tempos de execução por etapa e análises comparativas de condições ideais, limites viáveis e cenários inviáveis.

## 5.1 Testes do Ângulo de *Yaw* da Nacelle

Os testes relacionados com o ângulo de *yaw* da nacelle têm como objetivo avaliar os limites operacionais do sistema de inspeção em função da orientação inicial da turbina, fator crítico para o correto alinhamento e para a qualidade da captura de dados pelo UAV.

Para estes ensaios, a turbina foi configurada com as pás em formação de *Y invertido*, conforme ilustrado na Figura 5.1. Esta é considerada a disposição ideal para inspeção, já que não existe offset introduzido pelo ângulo de *roll* das blades.

O procedimento experimental consistiu em introduzir erros iniciais de *yaw* na nacelle durante a Etapa 1, forçando o UAV a corrigir autonomamente a orientação sem necessidade de intervenção manual. Foram analisados três cenários distintos:

- o caso ideal, sem erro inicial de *yaw*;
- o caso inviável, em que a inspeção não pode ser realizada devido ao desalinhamento elevado;
- e o caso limite, em que ainda é possível efetuar a inspeção de forma viável.

A avaliação do valor limite concentrou-se particularmente na Etapa 3, em que ocorre o alinhamento com o *hub*, pois é nesta fase que a precisão da orientação tem impacto direto na subsequente inspeção das pás.

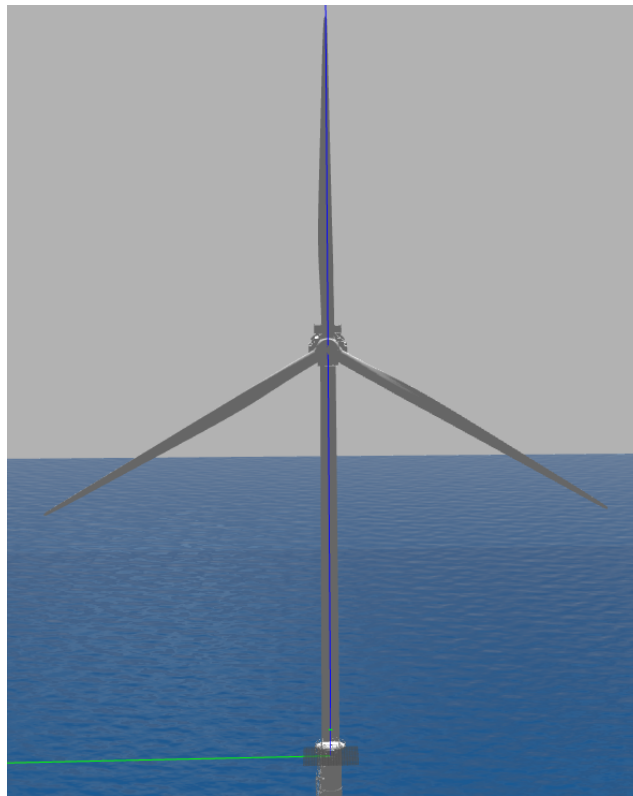


Figura 5.1: Configuração da turbina com pás em formação de *Y invertido*, sem offset do ângulo de *roll*.

### 5.1.1 Caso Ideal: Offset de *Yaw* de $0^\circ$

No cenário ideal, sem erro inicial de *yaw* na nacelle, o teste foi realizado utilizando apenas um UAV, o UAV1, dedicado à inspeção da pá esquerda da turbina e da torre. Nesta configuração, a orientação inicial da nacelle encontrava-se perfeitamente alinhada com o esperado, o que permitiu validar o pipeline de missão em condições ótimas.

O UAV executou todas as etapas da missão com sucesso, desde a decolagem até à aterragem. A câmara visual registou imagens que cobriram toda a largura da pá inspecionada de ponta a ponta, assim como toda a extensão da torre.

A trajetória completa do UAV1 encontra-se ilustrada na Figura 5.2, que apresenta o percurso em coordenadas 3D, com distância total percorrida de 574,1 metros e tempo total de execução de 9 minutos e 7 segundos.

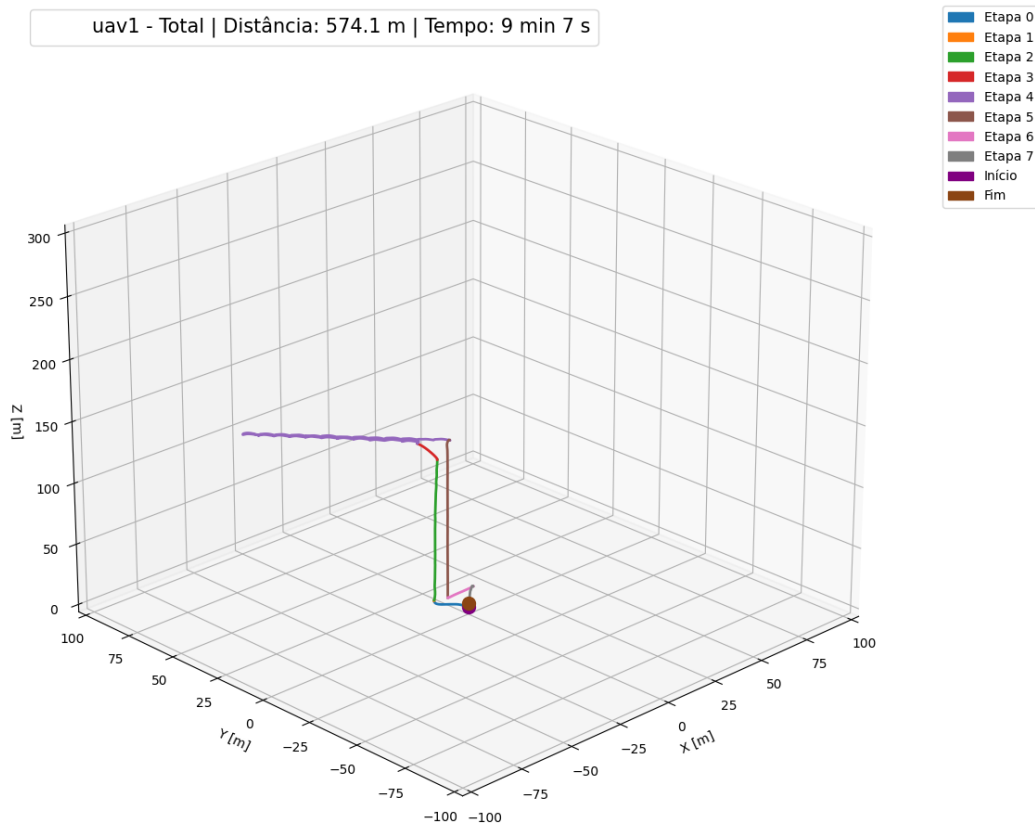


Figura 5.2: Trajetória 3D do UAV1 no teste com erro de  $yaw$  de  $0^\circ$ .

Para sintetizar o desempenho por etapa, apresenta-se a Tabela 5.1, que detalha as distâncias percorridas e os tempos de execução em cada fase. Observa-se que todas as etapas foram concluídas de forma eficiente, confirmando a robustez do sistema .

Tabela 5.1: Resultado no teste com erro de  $yaw$  de  $0^\circ$ .

Etapa	Distância (m)	Tempo
Etapa 0	29,3	0 min 30 s
Etapa 1	0,6	0 min 2 s
Etapa 2	115,2	1 min 17 s
Etapa 3	17,3	0 min 11 s
Etapa 4	252,2	3 min 14 s
Etapa 5	128,3	3 min 4 s
Etapa 6	15,0	0 min 17 s
Etapa 7	16,3	0 min 27 s
<b>Total</b>	<b>574,1</b>	<b>9 min 7 s</b>

### 5.1.2 Caso Limite Viável: Offset de *Yaw* de 40°

Tal como no teste anterior, utilizou-se um UAV dedicado à inspeção da pá esquerda da turbina e da torre. Neste ensaio, o offset de *yaw* foi incrementado progressivamente até atingir o limite em que o sistema ainda consegue completar a missão, identificado como 40°. Este valor provocou maiores ajustes durante o voo, especialmente na Etapa 3, responsável pelo alinhamento com o hub.

Comparado ao teste ideal, verificou-se um aumento do tempo de execução principalmente na Etapa 3 devido à correção do offset. A câmara visual captou corretamente toda a largura da pá e a extensão da torre, assegurando a inspeção completa.

A trajetória completa encontra-se ilustrada na Figura 5.3, com distância total percorrida de 571,3 metros e tempo total de execução de 11 minutos e 55 segundos.

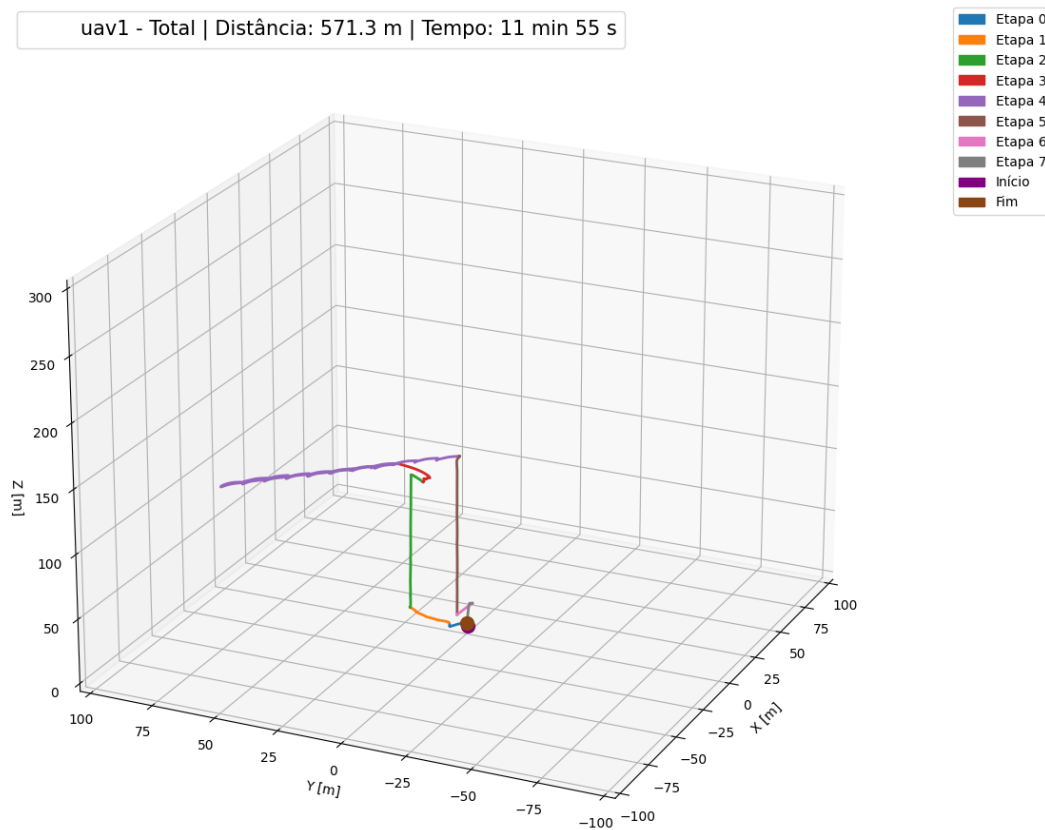


Figura 5.3: Trajetória 3D do UAV1 no teste com erro de *yaw* de 40°.

A Tabela 5.2 apresenta as distâncias e tempos por etapa, confirmando a execução eficiente do teste.

Tabela 5.2: Resultado no teste com erro de *yaw* de 40°.

<b>Etapa</b>	<b>Distância (m)</b>	<b>Tempo</b>
Etapa 0	29,4	0 min 31 s
Etapa 1	21,7	0 min 35 s
Etapa 2	113,1	1 min 52 s
Etapa 3	22,2	2 min 57 s
Etapa 4	229,4	3 min 43 s
Etapa 5	124,6	2 min 2 s
Etapa 6	18,3	0 min 52 s
Etapa 7	12,6	0 min 23 s
<b>Total</b>	<b>571,3</b>	<b>11 min 55 s</b>

### 5.1.3 Caso Inviável: Offset de *Yaw* de 50°

Mais uma vez, utilizou-se um UAV dedicado à inspeção da pá esquerda da turbina e da torre. Neste ensaio, foi introduzido um offset de *yaw* de 50°, que ultrapassou o limite operacional do sistema, resultando em falha.

O UAV progrediu normalmente nas Etapas 0 a 2, mas falhou na Etapa 3, não conseguindo alinhar corretamente com o *hub* devido ao desalinhamento excessivo. Como medida de segurança, o sistema ativou o modo de retorno, pulando a Etapa 4 e prosseguindo diretamente para a Etapa 5, regressando à base sem completar a inspeção das pás.

A trajetória completa, limitada às etapas executadas, encontra-se ilustrada na Figura 5.4, apresentando o percurso em coordenadas 3D, com distância total percorrida de 327,4 metros e tempo total de execução de 8 minutos e 30 segundos.

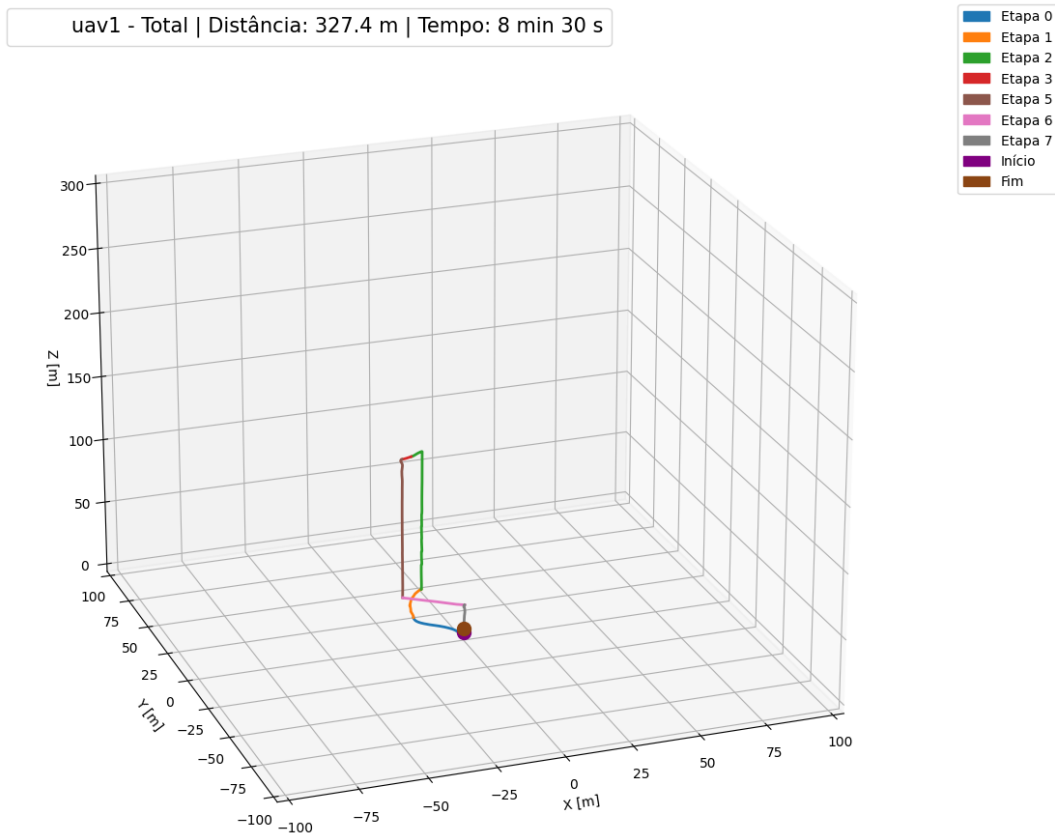


Figura 5.4: Trajetória 3D do UAV no teste com erro de *yaw* de  $50^\circ$ .

A Tabela 5.3 sintetiza as distâncias e tempos das etapas executadas, destacando a ausência da Etapa 4 devido à falha na Etapa 3.

Tabela 5.3: Resultado no teste com erro de *yaw* de  $50^\circ$ .

Etapa	Distância (m)	Tempo
Etapa 0	29,3	0 min 34 s
Etapa 1	27,1	0 min 46 s
Etapa 2	115,2	1 min 44 s
Etapa 3	4,1	0 min 3 s
Etapa 5	108,0	2 min 40 s
Etapa 6	29,5	0 min 25 s
Etapa 7	19,3	2 min 15 s
<b>Total</b>	<b>327,4</b>	<b>8 min 30 s</b>

## 5.2 Testes do Ângulo de *Roll* das Pás

Nesta secção são apresentados os testes relativos ao ângulo de *roll* das pás da turbina, mantendo o erro de *yaw* nulo, ou seja, sem desalinhamento da *nacelle*. O objetivo destes ensaios é avaliar o impacto do *offset* no *roll* das pás sobre a capacidade

do UAV adotar a estratégia de posicionamento *Esquerda-Central-Direita* conforme explicado na Secção 4.3.2.

Foram realizados três cenários de offset de *roll*:

- **Offset de 60°:** As pás estão rodadas numa configuração em que uma pá bloqueia parcialmente a visão da torre, correspondendo a uma situação inviável para inspeção.
- **Offset de 30°:** Mantém-se a ideologia de *Esquerda-Central-Direita*, permitindo ainda uma inspeção viável.
- **Offset de 40°:** Excede o conceito de *Esquerda-Central-Direita*, representando uma condição em que a inspeção deve ser recusada.

### 5.2.1 Caso Inviável: Offset de *Roll* de 60°

Neste teste, foi avaliado o impacto de um offset de *roll* de 60° nas pás da turbina, mantendo o erro de *yaw* em zero. A configuração da turbina, com as pás em *Y normal* e uma pá bloqueando parcialmente a visão da torre, é apresentada na Figura 5.5.



Figura 5.5: Configuração da turbina com offset de *roll* de 60° (*Y normal*), em que uma pá bloqueia parcialmente a torre, tornando a inspeção inviável.

Durante a execução, o pipeline seguiu normalmente até à Etapa 2, responsável pela detecção da torre. Contudo, nesta etapa o UAV identificou incorretamente a ponta da pá como torre. Como a ponta da pá não possui largura suficiente para ser considerada torre, a inspeção foi rejeitada. Como medida de segurança, o UAV ativou a Etapa 5, retornando diretamente à base sem prosseguir com a inspeção das pás e da torre. O comportamento ocorreu conforme esperado, confirmando que esta configuração não é adequada para inspeção e garantindo a aterragem segura do UAV.

A trajetória completa do UAV1 encontra-se ilustrada na Figura 5.6, com distância total percorrida de 86,3 m e tempo total de execução de 2 min 14 s.

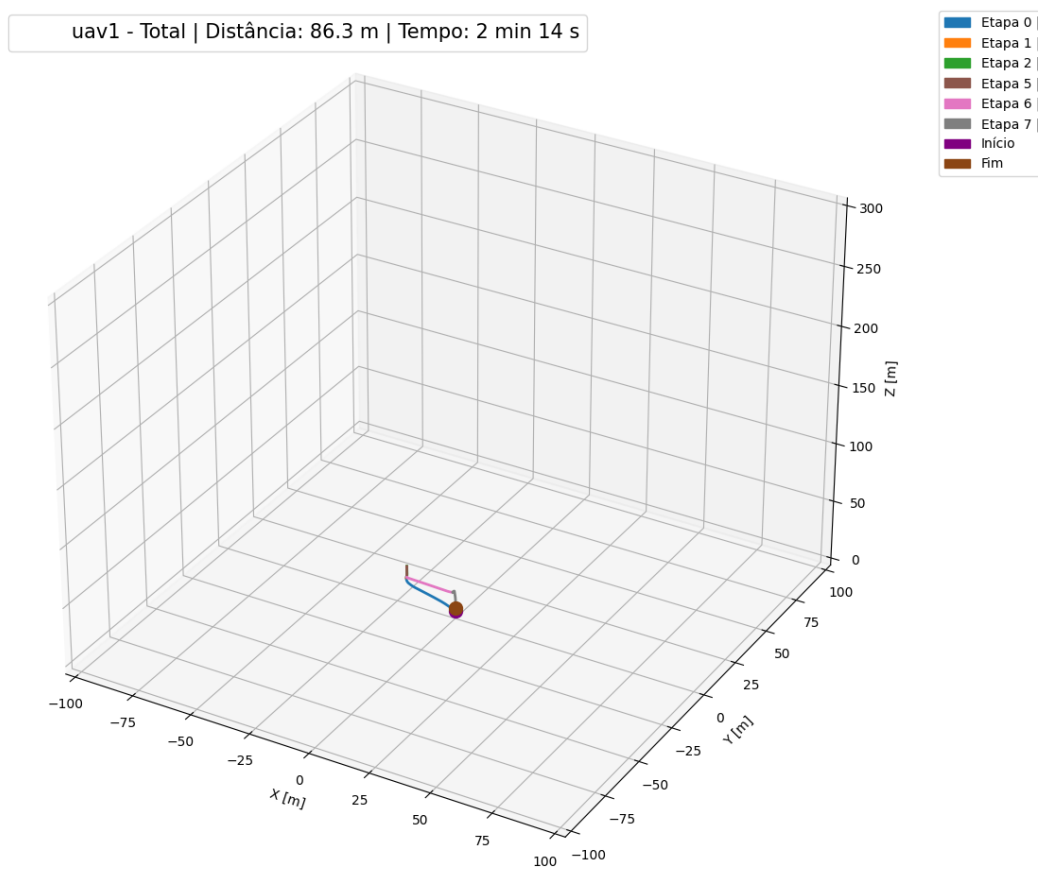


Figura 5.6: Trajetória 3D do UAV1 no teste com offset de *roll* de  $60^\circ$ .

Tabela 5.4: Resultado no teste com offset de *roll* de 60°.

<b>Etapa</b>	<b>Distância (m)</b>	<b>Tempo (min:s)</b>
Etapa 0	29,1	0:30
Etapa 1	0,4	0:02
Etapa 2	9,8	0:10
Etapa 5	9,7	0:12
Etapa 6	20,1	0:22
Etapa 7	17,1	0:54
<b>Total</b>	<b>86,3</b>	<b>2:14</b>

### 5.2.2 Caso Limite: Offset de *Roll* de 30°

Neste teste, avaliou-se o impacto de um offset de *roll* de 30° nas pás da turbina, mantendo o erro de *yaw* nulo. Esta configuração, ainda dentro do conceito de *Esquerda-Central-Direita*, permitiu verificar a capacidade do UAV em completar a inspeção, conforme descrito na Secção 4.3.2. A disposição das pás é ilustrada na Figura 5.7.

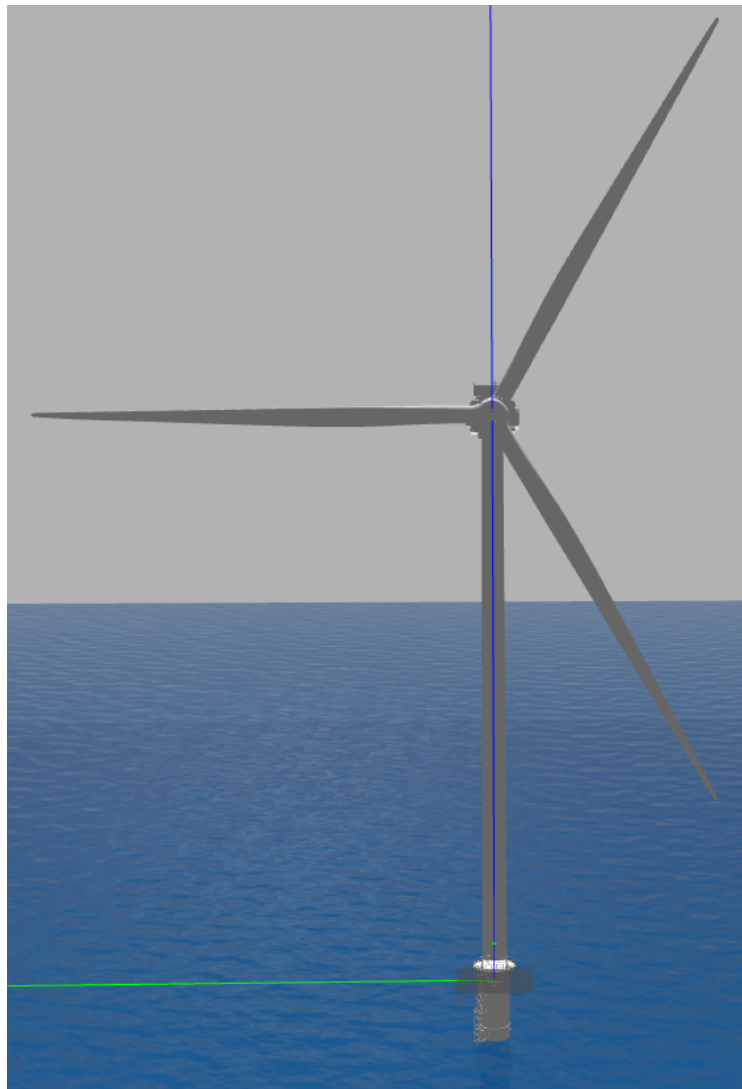


Figura 5.7: Configuração da turbina com offset de *roll* de  $30^\circ$ .

O UAV completou todas as etapas da missão desde a decolagem até à aterragem, inspecionando a torre e a pá esquerda, confirmando que um offset de  $30^\circ$  ainda é viável para o sistema.

A trajetória completa do UAV1 encontra-se ilustrada na Figura 5.8, com distância total percorrida de 563,3 m e tempo total de execução de 10 min 30 s.

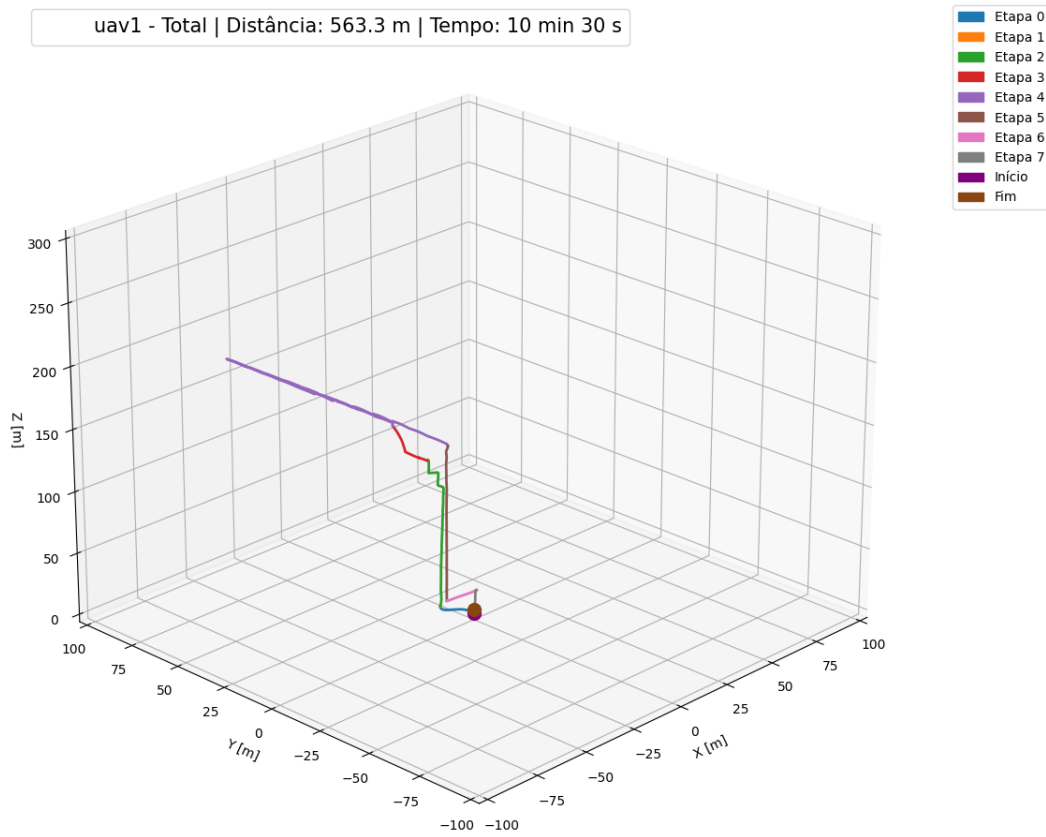


Figura 5.8: Trajetória 3D do UAV1 no teste com offset de *roll* de  $30^\circ$ .  
As etapas são representadas por cores distintas.

A Tabela 5.5 sintetiza as distâncias percorridas e os tempos de cada etapa.

Tabela 5.5: Resultado no teste com offset de *roll* de  $30^\circ$ .

Etapa	Distância (m)	Tempo (min:s)
Etapa 0	29,6	0:35
Etapa 1	0,5	0:02
Etapa 2	123,4	2:02
Etapa 3	33,4	0:48
Etapa 4	219,0	3:10
Etapa 5	124,8	2:37
Etapa 6	15,5	0:20
Etapa 7	17,0	0:52
<b>Total</b>	<b>563,3</b>	<b>10:30</b>

### 5.2.3 Caso Inviável: Offset de *Roll* de 40°

Neste teste, avaliou-se o impacto de um offset de *roll* de 40° nas pás da turbina, mantendo o erro de *yaw* nulo. Esta configuração excede o conceito de *Esquerda-Central-Direita*, tornando a inspeção inviável, conforme descrito na Secção 4.3.2. A disposição das pás, com um deslocamento que compromete a distinção das zonas, é ilustrada na Figura 5.9.

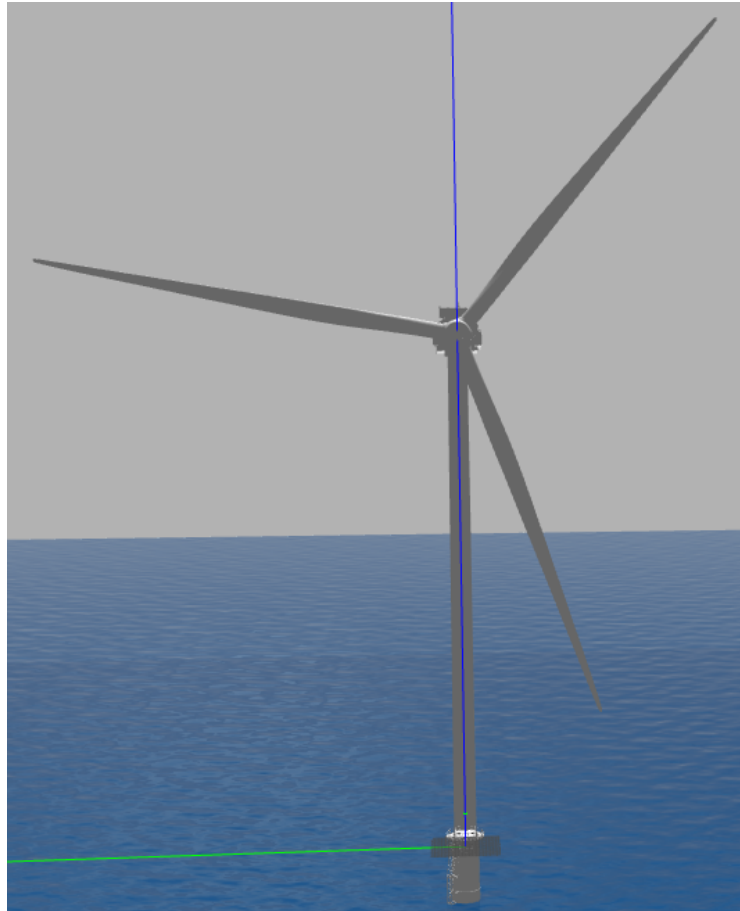


Figura 5.9: Configuração da turbina com offset de *roll* de 40°

Durante a execução, o pipeline avançou normalmente até à Etapa 3, responsável pelo alinhamento com o *hub*. Nesta etapa, o sistema rejeitou prosseguir para a Etapa 4 devido ao offset excessivo, ativando como medida de segurança a Etapa 5 para retornar à base sem completar a inspeção da pá. O comportamento ocorreu conforme esperado, confirmando que esta configuração não é compatível com o sistema e garantindo a aterragem segura do UAV.

A trajetória completa do UAV1 encontra-se ilustrada na Figura 5.10, com distância total percorrida de 313,5 m e tempo total de execução de 6 min 55 s.

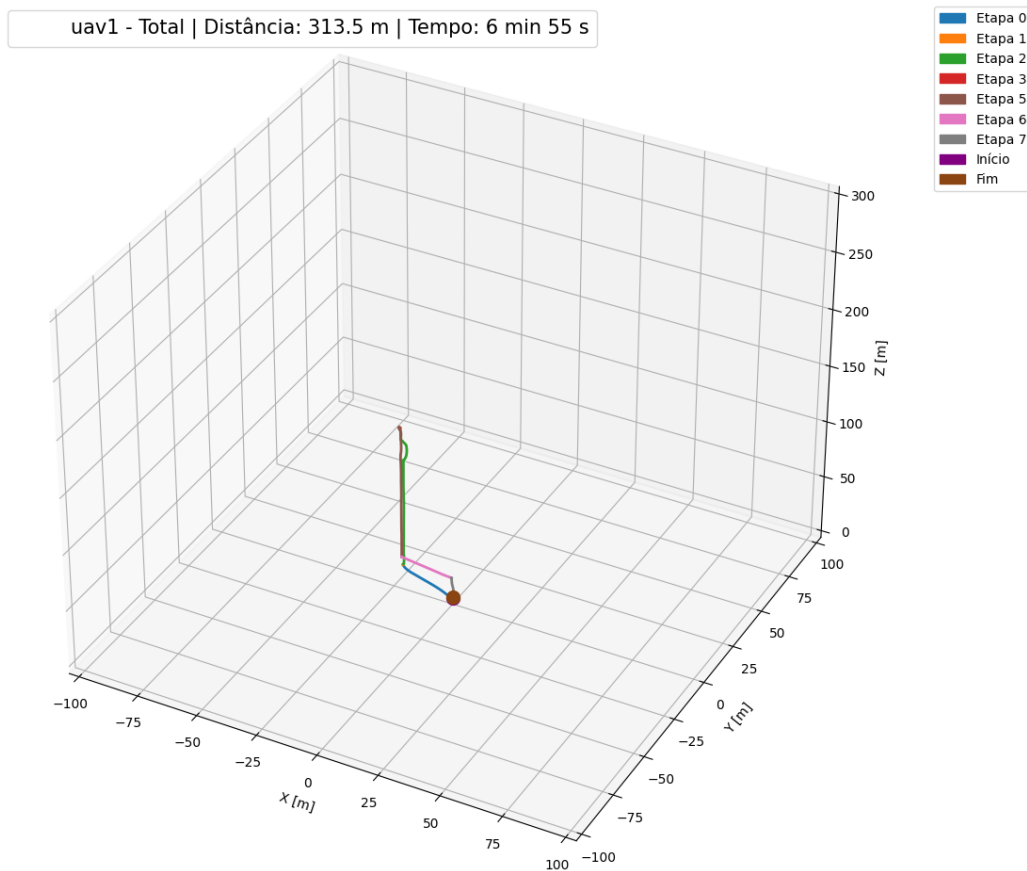


Figura 5.10: Trajetória 3D do UAV1 no teste com offset de *roll* de  $40^\circ$ .

Tabela 5.6: Resultado no teste com offset de *roll* de  $40^\circ$ .

Etapa	Distância (m)	Tempo (min:s)
Etapa 0	29,6	0:29
Etapa 1	0,7	0:03
Etapa 2	122,8	1:56
Etapa 3	0,3	0:00
Etapa 5	117,6	2:45
Etapa 6	23,3	0:25
Etapa 7	19,1	1:14
<b>Total</b>	<b>313,5</b>	<b>6:55</b>

### 5.3 Múltiplos UAVs

Por fim, os últimos testes e resultados apresentados referem-se à estratégia de coordenação de múltiplos UAVs desenvolvida ao longo do projeto. Neste ensaio, definiu-se

que o UAV 1 seria responsável pela pá esquerda, o UAV 2 pela pá direita e o UAV 3 pela pá central da turbina, garantindo cobertura distribuída.

Ao longo do desenvolvimento, várias abordagens para a coordenação de múltiplos UAVs foram avaliadas. Entre as alternativas, verificou-se que a estratégia mais eficaz consistia em lançar o próximo UAV apenas quando o anterior completava a Etapa 3. Esta abordagem faseada simplifica a coordenação, permitindo que cada UAV opere de forma reativa e independente, sem necessidade de comunicação direta entre eles, garantindo segurança e eficiência.

A estratégia escolhida foi então testada de forma tentativa-erro, de forma a que a melhor estratégia encontrada consistia em lançar o próximo UAV apenas após o anterior terminar a Etapa 3. Esta metodologia permitiu observar que a inspeção completa das três pás e da torre da turbina era realizada, em média, em cerca de 20 minutos. Este tempo representa uma redução significativa em relação ao lançamento sequencial (um a um), que exigiria aproximadamente 30 minutos.

A Etapa 5 revelou-se crítica nesta estratégia, pois, durante esta fase, os UAVs realizam manobras de desvio de obstáculos mais ativas, afastando-se de qualquer obstáculo próximo. Esta funcionalidade garante que, mesmo que a Etapa 5 de um UAV coincida com as Etapas 0 a 3 de outro, não ocorram colisões entre drones ou com a estrutura da turbina.

Outro benefício desta abordagem faseada é a sua robustez em situações imprevistas: caso a turbina apresente um offset elevado de *roll* ou *yaw*, o primeiro UAV detecta a anomalia e impede o lançamento dos restantes drones, evitando riscos desnecessários e aumentando a eficiência global da inspeção. Apesar de simples, esta estratégia é eficaz, beneficiando-se da natureza reativa dos UAVs e garantindo inspeções seguras e rápidas.

Para os testes, foi utilizada uma turbina em formação de *Y invertido*, sem offset de *roll* nas pás e sem offset de *yaw* na nacelle, confirmando a eficácia da estratégia em condições ideais.

As trajetórias dos três UAVs encontram-se ilustradas na Figura 5.11, apresentando os percursos em coordenadas 3D.

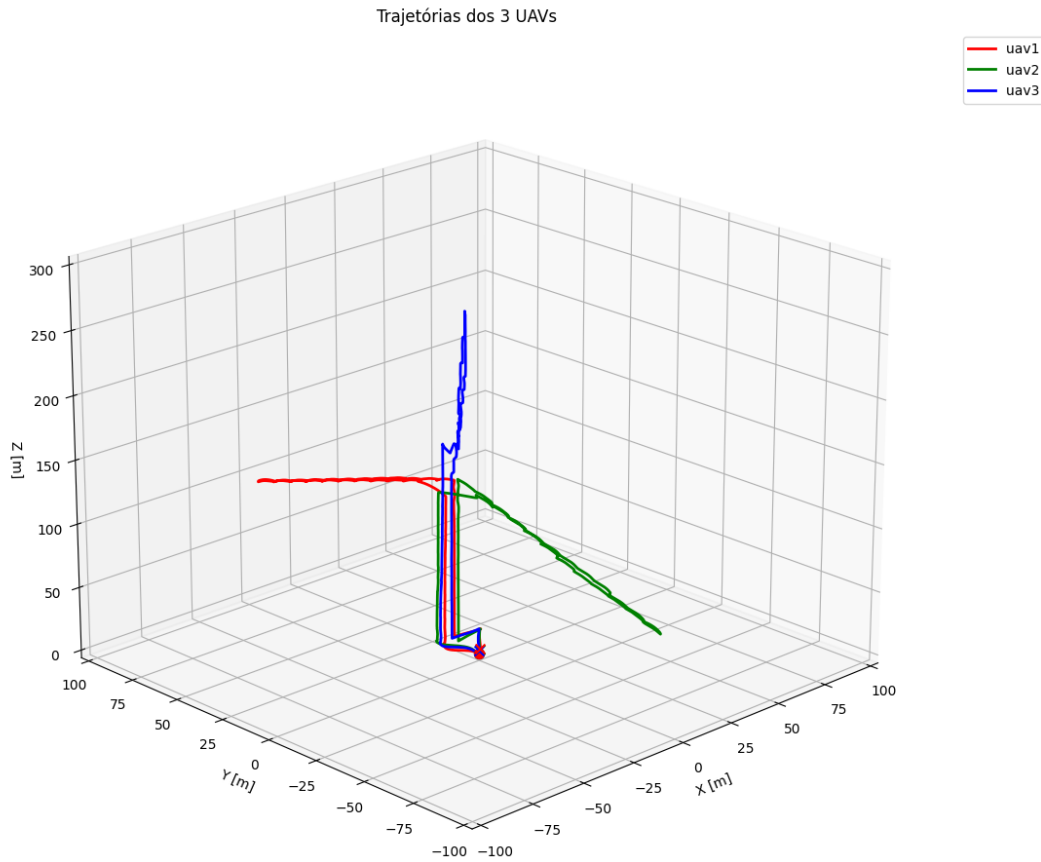


Figura 5.11: Resultado no teste multi-drone.

A Tabela 5.7 sintetiza as distâncias percorridas e os tempos de execução para cada UAV.

Tabela 5.7: Resultado no teste multi-drone.

UAV	Distância (m)	Tempo
UAV1	549.5	8 min 52 s
UAV2	559.9	9 min 32 s
UAV3	613.3	12 min 9 s

Com base nestes testes, conclui-se que a estratégia de lançamento faseado permite uma inspeção eficiente, segura e adaptativa da turbina, garantindo que cada UAV contribua de forma otimizada para a missão global.

## Capítulo 6

# Conclusão

O trabalho desenvolvido permitiu concretizar os objetivos inicialmente definidos, nomeadamente a conceção e implementação de um **pipeline de inspeção autónoma com UAVs** para turbinas eólicas *offshore*. A estrutura proposta, composta por várias etapas — desde a aproximação à turbina até ao retorno seguro — demonstrou ser capaz de garantir a cobertura completa das pás e assegurar a segurança da missão em diferentes cenários simulados.

Ao longo do desenvolvimento surgiram diversos desafios, sobretudo relacionados com a preparação do ambiente de simulação e a integração de múltiplas ferramentas, como *ROS*, *Gazebo* e *PX4*. A instalação de bibliotecas, a adaptação de pacotes existentes e a necessidade de criar soluções de compatibilidade exigiram um esforço considerável, tornando o processo exigente, mas também enriquecedor do ponto de vista técnico e académico. Estes obstáculos foram superados, contribuindo para uma implementação funcional e robusta do sistema.

Os resultados obtidos validaram o pipeline proposto, evidenciando a sua eficácia e a possibilidade de extensão para cenários mais complexos. Apesar disso, reconhece-se que existem limitações inerentes à utilização exclusiva de simulação, bem como a simplificações adotadas para viabilizar o estudo no âmbito desta dissertação.

**Como trabalhos futuros**, destaca-se a oportunidade de:

- explorar alternativas em cada etapa do pipeline, nomeadamente diferentes algoritmos de segmentação, métodos de planeamento de trajetórias e estratégias de controlo;

- investigar estratégias de lançamento e coordenação multi-UAV em cenários de maior escala;
- realizar a implementação em plataformas reais, através da integração em UAVs físicos, de modo a validar em ambiente experimental o desempenho observado em simulação.

Em suma, a dissertação confirmou a viabilidade de uma abordagem autónoma para a inspeção de turbinas eólicas com UAVs, contribuindo para a investigação nesta área e abrindo caminho para futuros avanços em direção a sistemas aplicáveis em contexto real.

# Bibliografia

- [1] Z. Ren, A. S. Verma, Y. Li, J. J. Teuwen e Z. Jiang, «Offshore wind turbine operations and maintenance: A state-of-the-art review,» *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110 886, 2021 (ver pp. 2–6).
- [2] A. J. Foster, M. Gianni, A. Aly, H. Samani e S. Sharma, «Multi-Robot Coverage Path Planning for the Inspection of Offshore Wind Farms: A Review,» *Drones*, vol. 8, n.º 1, p. 10, 2023 (ver pp. 2, 3).
- [3] A. McCoy, W. Musial, R. Hammond et al., «Offshore Wind Market Report: 2024 Edition,» National Renewable Energy Laboratory (NREL), Golden, CO (United States), rel. téc., 2024. URL: <https://www.nrel.gov/docs/fy24osti/90525.pdf> (ver p. 2).
- [4] O. Khalid, G. Hao, C. Desmond et al., «Applications of robotics in floating offshore wind farm operations and maintenance: Literature review and trends,» *Wind Energy*, vol. 25, n.º 11, pp. 1880–1899, 2022 (ver pp. 6, 8, 19, 20).
- [5] M. Futureworx, *Lilypad: A Network of Resident Autonomous UAVs for Offshore Wind Inspection*, <https://marshallgroup.com/media/pages/air/autonomous-systems/561c49c078-1712824389/lilypad-brochure-marshall-website.pdf>, White paper da Marshall Futureworx, 2023 (ver pp. 7, 18, 45).
- [6] D. Mitchell, J. Blanche, S. Harper et al., «A review: Challenges and opportunities for artificial intelligence and robotics in the offshore wind sector,» *Energy and AI*, vol. 8, p. 100 146, 2022 (ver p. 7).
- [7] M. Kreider, F. Oteri, A. Robertson, C. Constant e E. Gill, «Offshore Wind Energy: Technology Below the Water [Slides],» National Renewable Energy Lab.(NREL), Golden, CO (United States), rel. téc., 2022 (ver pp. 8, 9).
- [8] Z. Zhang e Z. Shu, «Unmanned aerial vehicle (UAV)-assisted damage detection of wind turbine blades: A review,» *Energies*, vol. 17, n.º 15, p. 3731, 2024 (ver pp. 9, 11, 12).
- [9] M. Civera e C. Surace, «Non-destructive techniques for the condition and structural health monitoring of wind turbines: A literature review of the last 20 years,» *Sensors*, vol. 22, n.º 4, p. 1627, 2022 (ver pp. 9–11).

- 
- [10] A. Alnmr e M. Mayassah, «Innovations in offshore wind: reviewing current status and future prospects with a parametric analysis of helical pile performance for anchoring mooring lines,» *Journal of Marine Science and Engineering*, vol. 12, n.º 7, p. 1040, 2024 (ver p. 11).
- [11] H.-M. Chung, S. Maharjan, Y. Zhang, F. Eliassen e K. Strunz, «Placement and routing optimization for automated inspection with unmanned aerial vehicles: A study in offshore wind farm,» *IEEE Transactions on Industrial Informatics*, vol. 17, n.º 5, pp. 3032–3043, 2020 (ver p. 17).
- [12] G. Fontenla-Carrera, E. Aldao, F. Veiga-López e H. González-Jorge, «Route Optimization of Vessel-UAV Tandem Systems for Offshore Wind Farm Inspections,» em *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2024, pp. 99–106 (ver p. 18).
- [13] M. Yan, H. Yuan, J. Xu, Y. Yu e L. Jin, «Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm,» *EURASIP Journal on Advances in Signal Processing*, vol. 2021, pp. 1–23, 2021 (ver p. 18).
- [14] T. Fan, L. Fu, C. Guo, Y. Zhang e L. Sun, «Multi UAV Inspection Optimization for Offshore Wind Farms Considering Battery Exchange Process,» *IEEE Transactions on Intelligent Vehicles*, 2024 (ver p. 18).
- [15] H. Baik e J. Valenzuela, «An optimization drone routing model for inspecting wind farms,» *Soft Computing*, vol. 25, n.º 3, pp. 2483–2498, 2021 (ver p. 18).
- [16] A. H. Ismail, X. Song, D. Ouelhadj, M. Al-Behadili e A. Fraess-Ehrfeld, «Unmanned surface vessel routing and unmanned aerial vehicle swarm scheduling for off-shore wind turbine blade inspection,» *Expert Systems with Applications*, p. 127 534, 2025 (ver p. 18).
- [17] K. Zhang, V. Pakrashi, J. Murphy e G. Hao, «Inspection of floating offshore wind turbines using multi-rotor unmanned aerial vehicles: literature review and trends,» *Sensors*, vol. 24, n.º 3, p. 911, 2024 (ver pp. 18–20, 22–24).
- [18] S.-J. Heo e W. S. Na, «Review of Drone-Based Technologies for Wind Turbine Blade Inspection,» *Electronics*, vol. 14, n.º 2, p. 227, 2025 (ver pp. 20–22).
- [19] M. Shafiee, Z. Zhou, L. Mei, F. Dinmohammadi, J. Karama e D. Flynn, «Unmanned aerial drones for inspection of offshore wind turbines: A mission-critical failure analysis,» *Robotics*, vol. 10, n.º 1, p. 26, 2021 (ver pp. 20–22).
- [20] F. P. G. Márquez e A. M. P. Chacón, «A review of non-destructive testing on wind turbines blades,» *Renewable Energy*, vol. 161, pp. 998–1010, 2020 (ver pp. 22–24).

- 
- [21] A. Amer, M. Mehndiratta, J. le Fevre Sejersen, H. X. Pham e E. Kayacan, «Visual Tracking Nonlinear Model Predictive Control Method for Autonomous Wind Turbine Inspection,» em *2023 21st International Conference on Advanced Robotics (ICAR)*, IEEE, 2023, pp. 431–438 (ver p. 24).
- [22] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak e G. Nikolakopoulos, «Cooperative coverage path planning for visual inspection,» *Control Engineering Practice*, vol. 74, pp. 118–131, 2018 (ver p. 25).
- [23] C. Kanellakis, E. Fresk, S. S. Mansouri, D. Kominiak e G. Nikolakopoulos, «Autonomous visual inspection of large-scale infrastructures using aerial robots,» *arXiv preprint arXiv:1901.05510*, 2019 (ver p. 25).
- [24] M. Stokkeland, K. Klausen e T. A. Johansen, «Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection,» em *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2015, pp. 998–1007 (ver p. 25).
- [25] O. Moolan-Feroze, K. Karachalios, D. N. Nikolaidis e A. Calway, «Simultaneous drone localisation and wind turbine model fitting during autonomous surface inspection,» em *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 2014–2021 (ver p. 26).
- [26] H. Guo, Q. Cui, J. Wang, X. Fang, W. Yang e Z. Li, «Detecting and positioning of wind turbine blade tips for uav-based automatic inspection,» em *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, pp. 1374–1377 (ver p. 26).
- [27] R. M. Claro, M. I. Pereira, F. S. Neves e A. M. Pinto, «Energy efficient path planning for 3d aerial inspections,» *IEEE Access*, vol. 11, pp. 32 152–32 166, 2023 (ver p. 27).
- [28] A. Oliveira, A. Dias, T. Santos, P. Rodrigues, A. Martins e J. Almeida, «LiDAR-based unmanned aerial vehicle offshore wind blade inspection and modeling,» *Drones*, vol. 8, n.º 11, p. 617, 2024 (ver pp. 28, 44, 45).
- [29] W. Gu, D. Hu, L. Cheng, Y. Cao, A. Rizzo e K. P. Valavanis, «Autonomous wind turbine inspection using a quadrotor,» em *2020 international conference on unmanned aircraft systems (ICUAS)*, IEEE, 2020, pp. 709–715 (ver pp. 29, 42, 43, 45).
- [30] B. E. Schäfer, D. Picchi, T. Engelhardt e D. Abel, «Multicopter unmanned aerial vehicle for automated inspection of wind turbines,» em *2016 24th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2016, pp. 244–249 (ver p. 30).

- [31] M. Car, L. Markovic, A. Ivanovic, M. Orsag e S. Bogdan, «Autonomous wind-turbine blade inspection using LiDAR-equipped unmanned aerial vehicle,» *IEEE access*, vol. 8, pp. 131 380–131 387, 2020 (ver pp. 31, 45).
- [32] C. Yang, H. Zhou, X. Liu et al., «BladeView: Toward Automatic Wind Turbine Inspection With Unmanned Aerial Vehicle,» *IEEE Transactions on Automation Science and Engineering*, 2024 (ver p. 31).
- [33] SkySpecs, *Automated Wind Turbine Blade Inspections*, <https://skyspecs.com/product/inspections/>, Acedido em 2025-09-08, 2023 (ver p. 32).
- [34] Sulzer Schmid, *3DX™ Blade Inspection Platform*, <https://sulzerschmid.com>, Acedido em 2025-09-08, 2023 (ver p. 33).
- [35] DSLRPros, *Industrial Drone Inspection Solutions*, <https://www.dslrpros.com/>, Acedido em 2025-09-08, 2023 (ver p. 33).
- [36] ABJ Drones, *Wind Turbine Inspection Services*, <https://abjdrones.com>, Acedido em 2025-09-08, 2023 (ver p. 33).
- [37] O. Robotics, *ROS: Robot Operating System*, <https://www.ros.org/>, Página oficial do ROS, jan. de 2025 (ver p. 35).
- [38] O. Robotics, *Gazebo Simulator*, <http://gazebo.org>, Página oficial do simulador Gazebo, jan. de 2025 (ver p. 36).
- [39] P. D. Team, *PX4 Autopilot*, <https://px4.io>, Plataforma de código aberto para controlo de veículos aéreos não tripulados, jan. de 2025 (ver p. 36).
- [40] ScienceDirect Topics, *Pinhole Camera Model — An Overview*, <https://www.sciencedirect.com/topics/engineering/pinhole-camera-model>, página de tópico informativo, sem autor especificado; acedido em: setembro 2025, 2025 (ver p. 38).
- [41] Y. A. Waykar, «LiDAR Technology: A Comprehensive Review and Future Prospects,» *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 9, n.º 7, pp. —, 2022, Revisão abrangente dos princípios, componentes, tipos e aplicações do LiDAR (ver p. 39).
- [42] H. Ørsted e Y. Ma, «Review of PID Controller Applications for UAVs,» *arXiv preprint arXiv:2311.06809*, 2023, Revisão abrangente sobre uso de controladores PID em UAVs (ver p. 40).
- [43] M. A. Fischler e R. C. Bolles, «Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,» *Communications of the ACM*, vol. 24, n.º 6, pp. 381–395, 1981. DOI: 10.1145/358669.358692 (ver pp. 40, 46).

- [44] M. Ester, H.-P. Kriegel, J. Sander e X. Xu, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,» em *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, AAAI Press, 1996, pp. 226–231 (ver pp. 41, 46).
- [45] Yuneec, *Typhoon H480 Advanced Dual Electric Edition Aerial Imaging Pro Drone*, <https://ottostore.com.au/products/yuneec-typhoon-h480-pro-drone-with-intel-real-sense-aerial-imaging-drone-4k-uhd-collision-avoidance-professional?srsltid=AfmB0oou4TMSnqWsSnnoIKurXMue4vEgroeUiQH23> Acedido: 2025-09-17 (ver pp. 45, 52).
- [46] O. Robotics, *Gazebo Simulator*, <https://gazebo.org/home>, Acedido: 2025-09-17 (ver p. 45).
- [47] PX4, *PX4 Autopilot*, <https://github.com/PX4/PX4-Autopilot>, Acedido: 2025-09-17 (ver p. 45).
- [48] O. S. R. Foundation, *ROS Noetic Ninjemys*, <https://wiki.ros.org/noetic>, Acedido: 2025-09-17 (ver p. 45).
- [49] V. Ermakov, *MAVROS: MAVLink extendable communication node for ROS*, <https://wiki.ros.org/mavros>, Acedido: 2025-09-17 (ver p. 45).
- [50] lmark1, *Velodyne Simulator*, [https://github.com/lmark1/velodyne\\_simulator](https://github.com/lmark1/velodyne_simulator), Acedido: 2025-09-17 (ver p. 45).
- [51] *Offshore Wind Turbine 3D Model*, <https://www.cgtrader.com/items/5695101/download-page>, Acedido em: setembro 2025, 2025 (ver pp. 49, 51).
- [52] Yuneec, *Typhoon H Plus*, <https://yuneec.online>, Acedido: 2025-09-17 (ver pp. 51, 52).
- [53] *Velodyne VLP-16 LiDAR Sensor*, <https://docs.pal-robotics.com/sdk/23.12/hardware/extra/velodyne.html>, 2025 (ver pp. 52, 53).



## Capítulo 7

# Appendix A

### 7.1 Configuração da Turbina Eólica

---

```
1 import pyvista as pv
2
3 # Ficheiros STL das componentes da turbina
4 files = {
5     "blades": "blades_hub.stl",
6     "nacelle": "nacelle_1.stl",
7     "tower_inspectable": "tower_inspectable.stl",
8     "tower_uninspectable": "tower_uninspectable.stl"
9 }
10
11 # Pontos de referência para rotação
12 hub_center = (0, 0, 150) # Centro de rotação das blades
13 yaw_axis_point = (0, 0, 0) # Ponto de referência para yaw
14
15 # Ângulos de rotação
16 blade_angle = 0 # Roll das pás
17 yaw_angle = 90 # Yaw da nacelle
18 rotation_final_z = 90 # Rotação global da turbina
19
20 # Lista para armazenar as malhas transformadas
21 meshes_transformed = []
22
23 # Carregamento e transformação de cada componente
```

---

```

24 for name, file in files.items():
25     mesh = pv.read(file)
26
27     if name == "blades":
28         mesh.rotate_y(blade_angle, point=hub_center, inplace=True)
29         # Roll das blades
30         mesh.rotate_z(yaw_angle, point=yaw_axis_point, inplace=
31             True) # Yaw aplicado s blades
32     elif name == "nacelle":
33         mesh.rotate_z(yaw_angle, point=yaw_axis_point, inplace=
34             True)
35
36     meshes_transformed.append(mesh)
37
38 # Combinação de todas as malhas
39 combined_mesh = pv.PolyData()
40 for mesh in meshes_transformed:
41     combined_mesh = combined_mesh.merge(mesh)
42
43 # Rotação final de toda a turbina
44 combined_mesh.rotate_z(rotation_final_z, point=(0, 0, 0), inplace=
45     True)
46
47 # Visualização interativa
48 plotter = pv.Plotter()
49 plotter.add_mesh(combined_mesh, color='lightgray', opacity=0.8,
50     show_edges=False)
51 plotter.add_axes(line_width=2, labels_off=False)
52 plotter.set_background("white")
53 plotter.show()
54
55 # Gravação do STL final
56 filename = f"wind_turbine_{blade_angle}_{yaw_angle}.stl"
57 combined_mesh.save(filename)
58 print(f"Arquivo STL gravado: {filename}")

```

---

## 7.2 Janelas de Inspeção

---

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d.art3d import Poly3DCollection
4
5 # --- Configuracoes de inspeccao ---
6 d = 12 # distancia de inspeccao (m)
7 lidar_height = 0.0
8 camera_offset = -0.312 # 31.2 cm abaixo do LiDAR

```

```
9 lidar_pitch_deg = 0
10 camera_pitch_deg = 0
11 lidar_pitch_rad = np.radians(lidar_pitch_deg)
12 camera_pitch_rad = np.radians(camera_pitch_deg)
13
14 # --- Janela do LiDAR ---
15 fov_h_lidar = 20 # campo de visao horizontal (graus)
16 fov_v_lidar = 30 # campo de visao vertical (graus)
17
18 width_lidar = 2 * d * np.tan(np.radians(fov_h_lidar)/2)
19 height_lidar = 2 * d * np.tan(np.radians(fov_v_lidar)/2)
20
21 x_window_lidar = np.array([d]*4)
22 y_window_lidar = np.array([-width_lidar/2, width_lidar/2,
23                             width_lidar/2, -width_lidar/2])
23 z_window_lidar = np.array([-height_lidar/2, -height_lidar/2,
24                             height_lidar/2, height_lidar/2])
24
25 # Funcao para aplicar rotacao em pitch
26 def rotate_pitch(x, y, z, pitch):
27     xr = x * np.cos(pitch) + z * np.sin(pitch)
28     yr = y
29     zr = -x * np.sin(pitch) + z * np.cos(pitch)
30     return xr, yr, zr
31
32 # Rotacao da janela do LiDAR
33 x_window_lidar_rot, y_window_lidar_rot, z_window_lidar_rot =
34     rotate_pitch(
35         x_window_lidar, y_window_lidar, z_window_lidar,
36         lidar_pitch_rad)
35 verts_lidar = [list(zip(x_window_lidar_rot, y_window_lidar_rot,
36                         z_window_lidar_rot))]
36 x_center_lidar, y_center_lidar, z_center_lidar = rotate_pitch(d,
37     0, 0, lidar_pitch_rad)
37
38 # --- Janela da camera ---
39 width_px = 640
40 height_px = 360
41 fov_h_cam = 1 # radianos (~57.3 deg)
42
43 # Campo de visao vertical mantendo proporcao
44 fov_v_cam = 2 * np.arctan((height_px / width_px) * np.tan(
45     fov_h_cam / 2))
45
46 # Dimensoes fisicas da janela a 12 m
47 width_cam = 2 * d * np.tan(fov_h_cam / 2)
48 height_cam = 2 * d * np.tan(fov_v_cam / 2)
49
50 x_window_cam = np.array([d]*4)
```

```

51 y_window_cam = np.array([-width_cam/2, width_cam/2, width_cam/2, -
    width_cam/2])
52 z_window_cam = np.array([-height_cam/2, -height_cam/2, height_cam
    /2, height_cam/2])
53
54 # Rotacionar e deslocar para posicao relativa ao LiDAR
55 x_window_cam_rot, y_window_cam_rot, z_window_cam_rot =
    rotate_pitch(
56     x_window_cam, y_window_cam, z_window_cam, camera_pitch_rad)
57 z_window_cam_rot += camera_offset
58
59 verts_cam = [list(zip(x_window_cam_rot, y_window_cam_rot,
    z_window_cam_rot))]
60 x_center_cam, y_center_cam, z_center_cam = rotate_pitch(d, 0, 0,
    camera_pitch_rad)
61 z_center_cam += camera_offset
62
63 # --- Visualizacao ---
64 fig = plt.figure(figsize=(12,8))
65 ax = fig.add_subplot(111, projection='3d')
66
67 # Sensores
68 ax.scatter(0, 0, lidar_height, color='yellow', s=100, label='LiDAR
    ')
69 ax.scatter(0, 0, camera_offset, color='green', s=100, label='
    Camera')
70
71 # Pontos centrais das janelas
72 ax.scatter(x_center_lidar, y_center_lidar, z_center_lidar, color='
    blue', s=50, label='Centro LiDAR')
73 ax.scatter(x_center_cam, y_center_cam, z_center_cam, color='
    magenta', s=50, label='Centro Camera')
74
75 # Janelas de projecao
76 poly_lidar = Poly3DCollection(verts_lidar, facecolors='cyan',
    alpha=0.3)
77 poly_cam = Poly3DCollection(verts_cam, facecolors='magenta', alpha
    =0.3)
78 ax.add_collection3d(poly_lidar)
79 ax.add_collection3d(poly_cam)
80
81 # Linhas do sensor aos cantos
82 for i in range(4):
83     ax.plot([0, x_window_lidar_rot[i]], [0, y_window_lidar_rot[i]
        ], [lidar_height, z_window_lidar_rot[i]], color='cyan',
            linewidth=1.5)
84     ax.plot([0, x_window_cam_rot[i]], [0, y_window_cam_rot[i]], [
        camera_offset, z_window_cam_rot[i]], color='magenta',
            linewidth=1.5)

```

```
85
86 # Texto informativo
87 textstr = (f"Distancia de inspeccao: {d:.1f} m\n\n"
88           f"LiDAR:\n Largura = {width_lidar:.2f} m\n Altura = {
89           height_lidar:.2f} m\n\n"
90           f"Camera:\n Largura = {width_cam:.2f} m\n Altura = {
91           height_cam:.2f} m")
92
93 # Eixos e limites
94 ax.set_xlabel('X (m)')
95 ax.set_ylabel('Y (m)')
96 ax.set_zlabel('Z (m)')
97 ax.set_xlim(-10, d+10)
98 ax.set_ylim(-max(width_lidar, width_cam), max(width_lidar,
99           width_cam))
100 ax.set_zlim(-max(height_lidar, height_cam), max(height_lidar,
101           height_cam))
102
103 ax.legend()
104 ax.set_title(f'Projecao do Campo de Visao - LiDAR e Camera (
105           Distancia = {d} m)')
```

Listagem 7.1: Cálculo das janelas de inspeção do LiDAR e da câmera