



ESCOLA NAVAL

ta sainte & bief faire



Guilherme de Matos e Silva Neves Bento

Desenvolvimento de um Sistema de Aterragem Automática Baseado em Visão para Drones Multirrotores

Dissertação para obtenção do Grau de Mestre em
Ciências Militares Navais, na especialidade de Marinha



Alfeite
2025



ESCOLA NAVAL

talant de bi-faire



Guilherme de Matos e Silva Neves Bento

*Desenvolvimento de um Sistema de Aterragem Automática
Baseado em Visão para Drones Multirrotores*

Dissertação para obtenção do Grau de Mestre em
Ciências Militares Navais, na especialidade de Marinha

Orientação de: Nuno Alexandre Antunes Martins Pessanha Santos

Co-orientação de: Victor José de Almeida e Sousa Lobo

Alexandre Valério Rodrigues

O Aluno Mestrando,

O Orientador,

Guilherme Bento

Nuno Pessanha Santos

Alfeite

2025

“Para passar o Bojador tem que passar além da dor.”

Fernando Pessoa

Agradecimentos

Gostaria de iniciar esta dissertação expressando a minha mais sincera gratidão a todos aqueles que, ao longo do meu percurso académico, contribuíram de forma significativa para o meu desenvolvimento pessoal, militar e científico.

Aos meus orientadores, Doutor Nuno Pessanha Santos, Engenheiro Valério Rodrigues e Doutor Victor Lobo, agradeço o acompanhamento atento, o rigor técnico e a confiança depositada ao longo de todo este processo. O vosso apoio foi determinante para a concretização deste trabalho.

À guarnição da Célula de Inovação e Experimentação Operacional de Sistemas Não Tripulados (CEOV), deixo um agradecimento sentido pela forma como me acolheram, pelas condições que proporcionaram e pela disponibilidade constante demonstrada no decorrer da elaboração da dissertação.

Aos meus familiares e amigos, que sempre se mantiveram presentes com palavras de incentivo e gestos de apoio, expresso um agradecimento profundo. O vosso suporte contínuo foi essencial nos momentos de maior exigência.

Por fim, reconheço com apreço o papel dos meus camaradas do curso “Chefe de Divisão da Armada António Lopes da Costa e Almeida”. O espírito de camaradagem e entreaajuda que sempre cultivámos tornou este percurso não só mais sólido, mas também mais significativo.

Resumo

A aterragem de Veículos Aéreos Não Tripulados (VANTs) constitui uma das fases mais críticas do voo, representando um elevado risco de acidente, sobretudo em cenários marítimos e em ambientes de negação de Sistema Global de Navegação por Satélite (*Global Navigation Satellite System*, GNSS). Com o objetivo de aumentar a segurança e reduzir a dependência de operadores humanos, esta dissertação propõe o desenvolvimento de um sistema de aterragem automática de um VANT multirrotor, baseado exclusivamente em visão computacional. A escolha de um multirrotor deve-se à sua capacidade de voo estacionário, elevada manobrabilidade e requisitos reduzidos de espaço para a aterragem, características essenciais em plataformas navais. Deste modo, o sistema proposto recorreu a uma câmara embarcada no VANT, capaz de captar imagens a cores segmentadas nos canais vermelho, verde e azul (*Red, Green and Blue*, RGB), cujos dados são processados em tempo real numa estação terrestre. A partir desta informação é estimada a pose relativa ao alvo, que serve de base para a geração de uma trajetória de aterragem. Uma vez que o VANT dispõe da capacidade de executar comandos de alto nível, permitindo translações e rotações de forma autónoma, a trajetória é então executada através desses comandos, tendo como objetivo alcançar uma aterragem com erro inferior a 20 cm. A metodologia foi desenvolvida em ambiente *Software-In-The-Loop* (SITL), com o apoio de três componentes principais: *Robot Operating System 2* (ROS 2), que estrutura a comunicação modular do sistema, o simulador tridimensional Gazebo, que reproduz sensores, dinâmica e que permite simular de forma controlada o ambiente de estudo e o ArduPilot, um *firmware* de código aberto amplamente usado em VANTS. A metodologia foi validada em testes simulados atingindo uma precisão de aterragem com um erro mínimo de 5 cm, mantendo-se menor que os 20cm definidos como requisito, demonstrando a viabilidade da abordagem.

Palavras-chave: Veículos Aéreos Não Tripulados, Aterragem Automática, Visão Computacional, Veículos Militares

Abstract

The landing of Unmanned Aerial Vehicles (UAVs) is one of the most critical phases of flight, posing a high risk of accident, especially in maritime scenarios and in environments where Global Navigation Satellite System (GNSS) is denied. With the aim of increasing safety and reducing dependence on human operators, this dissertation proposes the development of an automatic landing system for a multirotor UAV, based exclusively on computer vision. The choice of a multirotor is due to its hovering capability, high maneuverability, and reduced landing space requirements, which are essential characteristics for naval platforms. Thus, the proposed system used a camera embedded in the UAV, capable of capturing color images segmented into red, green, and blue (RGB) channels, whose data are processed in real time at a ground station. Based on this information, the relative position of the target is estimated, which serves as the basis for generating a landing trajectory. Since the UAV has the ability to execute high-level commands, allowing autonomous translations and rotations, the trajectory is then executed through these commands, with the goal of achieving a landing with an error of less than 20 cm. The methodology was developed in a Software-In-The-Loop (SITL) environment, with the support of three main components: Robot Operating System 2 (ROS 2), which structures the system's modular communication; the Gazebo three-dimensional simulator, which reproduces the environment; and the Gazebo simulator, which reproduces the environment. The methodology was validated in simulated tests, achieving landing accuracy with a minimum error of 5 cm, remaining below the 20 cm requirement, demonstrating the feasibility of the approach.

Keywords: Unmanned Aerial Vehicles, Autonomous Landing, Computer Vision, Military Vehicles

Índice

Introdução	1
1.1 Contexto Geral	1
1.2 Motivação e relevância para a Marinha Portuguesa	3
1.3 Objetivos	4
1.4 Desafios	5
1.5 Trabalhos publicados	6
1.6 Estrutura	7
2 Fundamento Teórico	9
2.1 Veículos Aéreos Não Tripulados	9
2.1.1 Tipos e Características	10
2.1.2 Sistemas de Propulsão e Controlo	12
2.1.3 Navegação e Sensores a bordo	15
2.2 Visão Computacional	15
2.2.1 Modelo da câmara	16
Modelo <i>pinhole</i>	16
Distorções óticas	18
Calibração de câmara	19
2.2.2 Reconhecimento de Padrões e Segmentação	20
2.3 Marcadores	21
2.3.1 Marcadores Fiduciais	21
2.3.2 Marcador de Navegação H	23
2.4 Estimativa de pose	23
2.4.1 Fundamentos da estimativa baseada em visão	24
2.4.2 Filtragem da Estimativa de Pose	24
Fase de Predição do KF	25
Fase de Correção do KF	26
2.5 Aterragem Automática	27
2.5.1 Planeamento de Trajetória de aterragem	27
<i>Minimum Jerk</i>	28

<i>Minimum Snap</i>	29
2.5.2 Integração da Visão no Controlo de Aterragem	30
3 Estado da Arte	33
3.1 Panorama Geral das Estratégias de Aterragem Automática	33
3.1.1 Métodos Dependentes de GNSS	34
3.1.2 Métodos Independentes de GNSS	34
Métodos Baseados em Visão Computacional Embarcada	36
Métodos Baseados em Visão Computacional em Infraestruturas Externas	37
3.2 Aterragem Automática Baseada em Marcadores Visuais	37
3.2.1 Estimativa de Pose	38
3.2.2 Marcadores Visuais	39
3.3 Planeamento de Trajetória	40
3.4 Síntese Crítica	44
4 Formulação do problema	47
4.1 Definição do Cenário Operacional	47
4.2 Problemas Identificados	48
4.3 Objetivo Técnico do Sistema	49
4.4 Restrições do Sistema	50
4.5 Requisitos Funcionais	50
4.5.1 Conversão de referenciais	51
4.5.2 Perceção visual	52
4.5.3 Filtro de temporal	52
4.5.4 Planeamento da trajetória	52
4.5.5 Geração de comandos de alto nível	52
4.5.6 Autopiloto	53
4.5.7 Controlo e execução em tempo real	53
5 Metodologia	55
5.1 Perceção Visual	55
5.1.1 Calibração da Câmara	56
5.1.2 Detecção de Marcadores	57
Marcadores ArUco	57
Marcador H	58
5.1.3 Transformações de Referenciais	59
5.1.4 Filtragem Temporal	60

5.1.5	Refinamento da Detecção	62
5.2	Planeamento de Trajetória	63
5.2.1	Trajetória com <i>minimum snap</i>	63
5.2.2	Trajetória com <i>minimum jerk</i>	64
5.3	Controlo de Alto Nível	64
5.3.1	Interface de Controlo Manual	65
5.4	Autopiloto	66
6	Resultados experimentais	67
6.1	Resultados em Ambiente Simulado	67
6.1.1	<i>Ground Truth</i>	68
6.1.2	Estimativa de Pose	68
	Alcance e zona de segurança com ArUco 13.2 cm	68
	Alcance e zona de segurança com ArUco 70 cm	69
	Alcance e zona de segurança com marcador H 80 cm	69
	Trajetória com formato infinito com ArUco de 13.2 cm	69
	Trajetória com formato infinito com ArUco de 70 cm	69
	Trajetória com formato infinito com marcador H	70
6.1.3	Trajetória de Aterragem com <i>minimum jerk</i>	70
6.1.4	Trajetória de Aterragem com <i>minimum snap</i>	71
6.2	Resultados em Ambiente Real	73
6.2.1	RTK-GNSS	73
6.2.2	Estimativa de Pose	73
	Testes em bancada	73
	Testes em voo real	74
6.2.3	Trajetória de Aterragem	75
6.3	Discussão Geral	76
7	Conclusões e Trabalho Futuro	77
7.1	Conclusões	77
7.2	Trabalho Futuro	78
	Bibliografia	81
	Apêndices	89
	A Testes ao RTK-GNSS	89
	B Comparação <i>Ground Truth</i>	91

C Alcance e Zona de Segurança	93
C.1 ArUco de 13.2 cm	93
C.2 ArUco de 70 cm	94
C.3 H de 80cm	96
D Trajetória com formato infinito	97
D.1 ArUco de 13.2 cm	97
D.2 ArUco de 70 cm	99
D.3 H de 80 cm	102
E Detalhes Técnicos da Implementação	105
E.1 Implementação em Simulação	105
E.2 Implementação em Voo Real	105
E.3 <i>Logging</i> e Armazenamento de Dados	105
E.4 Interface de Controlo Manual	106
F Cálculo Simbólico do Jacobiano no EKF	107
G Definição de Avaliação de Desempenho	109

Lista de Figuras

1.1	Blocos Funcionais do Sistema.	4
2.1	Exemplo de VANT <i>tilt-rotor</i> (China Military 2024).	11
2.2	Diagrama de ligação dos principais componentes do autopiloto Pixhawk (ArduPilot Developers 2014).	14
2.3	Projeção de um ponto 3D no plano da imagem utilizando o modelo <i>pinhole</i> (Hartley e Zisserman 2003).	17
2.4	Marcador ArUco (esquerda) (OpenCV 2024) e Marcador de Navegação “H” (direita) (OpenMV 2020).	21
2.5	Marcador ArUco visualizado no gazebo com eixos coordenados desenhados (X-vermelho, Y-verde, Z-Azul).	22
2.6	Ilustração do marcador H e as suas dimensões (M. Vieira 2023) . .	23
2.7	Fases do Kalman Filter (Welch, Bishop et al. 1995).	25
3.1	Esquema de comunicações com um VANT utilizando RTK-GNSS (Adaptado de JOUAV 2023).	34
3.2	Classificação de aterragem automática independente de GNSS (Xin et al. 2022).	35
4.1	Ilustração de aterragem num navio com marcadores.	48
4.2	VANT Quadcopter da CEOV utilizado nos testes experimentais. . .	49
4.3	Diagrama de requisitos funcionais do sistema.	51
5.1	Arquitetura funcional do sistema de aterragem automática.	55
5.2	Comparação entre a imagem original (esquerda) e a imagem após correção da distorção (direita).	57
5.3	Deteções de marcador ArUco ID 23 de 13.2 cm.	58
5.4	Deteção de marcador H de 80 cm de lado.	59
5.5	Ilustração dos principais referenciais utilizados.	60
6.1	Ambiente simulado (esquerda) e ambiente real (direita).	67
6.2	Trajetórias estimadas (verde) com o método <i>minimum jerk</i> vs reais (azul) com ArUco 13.2 cm (i), ArUco 70 cm (ii) e H (iii).	71

6.3	Trajetórias estimadas (verde) com o método <i>minimum snap</i> vs reais (azul) com ArUco 13.2 cm (esquerda), ArUco 70 cm (centro) e H (direita).	72
6.4	Testes de bancada à estimativa de pose.	73
6.5	Erro de posição estimada (Ardupilot vs EKF).	74
6.6	Erro de orientação estimada (Ardupilot vs EKF).	75
A.1	Testes ao RTK-GNSS nas instalações da CEOV (esquerda) e da ETNA (direita).	89
A.2	Testes ao RTK-GNSS nas instalações do IST.	90
B.1	Erro da posição estimada pelo ArduPilot em relação ao <i>ground truth</i> do Gazebo, em função da altitude.	91
C.1	Erro da estimativa de pose em função da altura, com marcador ArUco de 13.2 cm.	93
C.2	Erro da estimativa de pose em função da altura, com marcador ArUco de 70 cm.	94
C.3	Erro da estimativa de pose em função da altura, com marcador H de 80 cm.	96
D.1	Erro de posição estimada (RAW vs EKF) — Trajetória em formato infinito.	98
D.2	Erro de orientação estimada (RAW vs EKF) — Trajetória em formato infinito.	98
D.3	Comparação da trajetória 3D estimada (RAW e EKF) com o <i>ground truth</i>	99
D.4	Erro de posição estimada (RAW vs EKF) — Trajetória em formato infinito, 4 m.	100
D.5	Erro de orientação estimada (RAW vs EKF) — Trajetória em formato infinito, 4 m.	101
D.6	Trajeto3ria 3D estimada (RAW e EKF) comparada com o <i>ground truth</i> , 4 m.	101
D.7	Erro de posição estimada (RAW vs EKF) — Trajetória em formato infinito, 4 m.	102
D.8	Erro de orientação estimada (RAW vs EKF) — Trajetória em formato infinito com marcador H de 80 cm.	103
D.9	Trajeto3ria 3D estimada (RAW e EKF) comparada com o <i>ground truth</i> com marcador H de 80 cm.	104

Lista de Tabelas

2.1	Resumo da classificação dos VANTs segundo diferentes critérios (Arjomandi et al. 2006; Hassanalian e Abdelkefi 2017).	12
3.1	Comparação de desempenho entre os marcadores.	40
3.2	Síntese das principais abordagens de aterragem automática baseadas em visão computacional	44
6.1	Precisão de aterragem com método <i>minimum jerk</i> à velocidade máxima de 1.1 m/s.	71
6.2	Precisão de aterragem com método <i>minimum snap</i> à velocidade máxima de 1.5 m/s.	72
6.3	Erros de pose para diferentes distâncias ao marcador.	74
6.4	Erros de posição (Ardupilot vs EKF).	74
6.5	Erros de orientação (Ardupilot vs EKF).	75
A.1	Resultados do processo de <i>survey-in</i> do RTK-GNSS	89
A.2	Resultados do processo de <i>survey-in</i> do RTK-GNSS do IST	90
B.1	Erro de posição tridimensional (RMSE) entre o ArduPilot e o Gazebo a diferentes altitudes.	91
C.1	Erro de estimativa de pose e frequência de detecção em função da altitude com marcador ArUco de 13.2 cm.	94
C.2	Erro de estimativa de pose e frequência de detecção em função da altura com marcador ArUco de 70 cm.	95
C.3	Erro de estimativa de pose e frequência de detecção em função da altura com marcador H de 80 cm.	96
D.1	Erro de posição (RMSE) para trajetória em formato infinito com marcador ArUco de 13.2 cm.	97
D.2	Erro de orientação (RMSE) para trajetória em formato infinito com marcador ArUco de 13.2 cm.	97

D.3	Erro de posição (RMSE) para trajetória em formato infinito com marcador ArUco de 70 cm.	99
D.4	Erro de orientação (RMSE) para trajetória em formato infinito com marcador ArUco de 70 cm.	100
D.5	Erro de posição (RMSE) para trajetória em formato infinito com marcador H de 80 cm.	102
D.6	Erro de orientação (RMSE) para trajetória em formato infinito com marcador H de 80 cm.	103

Lista de Abreviaturas

CEOV	Célula de Inovação e Experimentação Operacional de Sistemas Não Tripulados
DC	<i>Direct Current</i>
DOF	<i>Degrees of Freedom</i>
EKF	<i>Extended Kalman Filter</i>
ENU	<i>East-North-Up</i>
ESC	<i>Electronic Speed Controller</i>
ETNA	Escola de Tecnologias Navais
FOV	<i>Field of View</i>
FPV	<i>First Person View</i>
FRD	<i>Forward-Right-Down</i>
GNSS	<i>Global Navigation Satellite System</i>
GPS	<i>Global Positioning System</i>
HITL	<i>Hardware-In-The-Loop</i>
HSV	<i>Hue, Saturation, Value</i>
IBVS	<i>Image-Based Visual Servoing</i>
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
IUM	Instituto Universitário Militar
KF	<i>Kalman Filter</i>
LED	<i>Light Emitting Diode</i>
LiDAR	<i>Light Detection and Ranging</i>
LQR	<i>Linear Quadratic Regulator</i>
MAVLINK	<i>Micro Air Vehicle Link</i>
MPC	<i>Model Predictive Control</i>
NED	<i>North-East-Down</i>
PID	<i>Proportional-Integral-Derivative</i>
PNP	<i>Perspective-n-Point</i>
RANSAC	<i>Random Sample Consensus</i>
RDF	<i>Right-Down-Forward</i>
RGB	<i>Red, Green, Blue</i>
RMSE	<i>Root Mean Square Error</i>

ROS	<i>Robot Operating System</i>
RTK	<i>Real-time Kinematic Positioning</i>
RTK-GNSS	<i>Real-time Kinematic Positioning – Global Navigation Satellite System</i>
RTSP	<i>Real-Time Streaming Protocol</i>
SI	Sistema Internacional de Unidades
SITL	<i>Software-In-The-Loop</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SMC	<i>Sliding Mode Control</i>
SPP	<i>Standard Point Positioning</i>
UAS	<i>Unmanned Aerial System</i>
UAV	<i>Unmanned Aerial Vehicle</i>
UDP	<i>User Datagram Protocol</i>
VANT	Veículo Aéreo Não Tripulado
VTOL	<i>Vertical Take-Off and Landing</i>

Introdução

A crescente utilização de Veículos Aéreos Não Tripulados (VANTs), em setores como a defesa, segurança, investigação e indústria, tem impulsionado a necessidade de automatizar as várias fases das suas operações (Mohsan et al. 2022). Entre estas, a aterragem de VANTs destaca-se como uma das mais críticas, representando uma proporção significativa dos incidentes reportados durante missões VANTs, nos quais 47% a 68% dos casos estudados por Asim, Ehsan e Rafique 2005 foram causados por intervenções de humanos no sistema. A automatização deste processo constitui, portanto, um passo essencial para o aumento da segurança, da fiabilidade e da eficiência operacional (Nuno Pessanha Santos 2020).

A automatização da aterragem com base em visão computacional representa um contributo relevante para a modernização das operações com VANTs, especialmente em cenários de negação do sinal de Sistema Global de Navegação por Satélite (*Global Navigation Satellite System*, GNSS), onde a autonomia e a precisão se tornam imperativas. As aplicações potenciais incluem missões de vigilância, recolha de dados e busca e salvamento em ambientes adversos e de acesso difícil (Kakaletsis et al. 2021).

Este capítulo apresenta, em primeiro lugar, na Secção 1.1 o contexto geral da problemática abordada, destacando a evolução dos VANTs e os desafios operacionais enfrentados em cenários navais. Seguidamente, na Secção 1.2 discute-se a motivação e a relevância desta dissertação de mestrado. Já na Secção 1.3 são estabelecidos os objetivos da dissertação. A seguir, na Secção 1.4 são expostos os principais desafios técnicos operacionais presentes. Na Secção 1.5, são descritos os trabalhos científicos associados à dissertação. Por fim, na Secção 1.6 é apresentada a organização geral da dissertação.

1.1 Contexto Geral

Nos últimos anos, os VANTs têm registado uma evolução acelerada, impulsionada por avanços significativos ao nível tecnológico e pela diversificação progressiva dos seus domínios de aplicação. Estes, eram inicialmente utilizados para

missões de cariz militar, como vigilância, reconhecimento e empastelamento de sinais (Fahlstrom, Gleason e Sadraey 2022), no entanto, atualmente, são sistemas com capacidade para tarefas mais complexas e diversificadas. Pelo facto de serem tão versáteis e capazes de se ajustar, encontram-se hoje em dia presentes em diversas atividades do nosso quotidiano, como entrega de correio, encomendas e material médico (Mohsan et al. 2022), nas atividades da indústria destacam-se na fiscalização de infraestruturas, na agricultura, construção civil e na segurança pública (Chang et al. 2022) e na defesa, como por exemplo combate (Valavanis e Vachtsevanos 2014) e como já referido, em missões de vigilância, reconhecimento.

Relativamente ao meio militar, os VANTs mudaram o paradigma, conferindo uma nova vantagem estratégica. Neste momento, já é possível fazer a aquisição de dados em tempo real, realização de patrulhas com menor dependência de pessoal e a própria execução de operações em áreas de alto risco e de difícil acesso, sem a exposição direta de vidas humanas (Boyle 2012). Mais concretamente, estes sistemas demonstram capacidade para operar em cenários marítimos, mesmo expostos a fatores que habitualmente limitam os meios tripulados, como a visibilidade reduzida, o risco e custo operacional associado. Esta autonomia operacional proporciona uma maior flexibilidade, permitindo cobrir extensas áreas oceânicas e apoiar missões de inteligência, reconhecimento e vigilância (Lyu et al. 2023).

Mesmo com as suas vantagens, é de salientar os desafios técnicos significativos que estes apresentam em ambientes marítimos. Para voarem nas proximidades de navios e executarem a aterragem nos mesmos, é necessário sistemas de alta precisão e com boa fiabilidade, esta necessidade acresce ainda mais se tivermos em conta condições de instabilidade causados pela agitação marítima. Assim, a visão computacional surge como um recurso promissor, permitindo a perceção do ambiente e tomada de decisão independente de GNSS, que são sensíveis a empastelamento ou mesmo a indisponibilidade temporária. Deste modo, é um avanço para a modernização das operações de VANTs, o desenvolvimento de sistemas que garantam operações seguras e eficientes, de forma autónoma em ambientes marítimos (Wynn 2018).

Diversas abordagens têm sido propostas para garantir a aterragem automática de VANTs, incluindo soluções baseadas em radiofrequência (RF) (Kelner e Ziółkowski 2019), visão computacional, como o sistema de visão monocular terrestre (Nuno Pessanha Santos 2020), sensores de deteção e medição por luz (*Light Detection and Ranging*, LiDAR) (Liu et al. 2022) e sistemas de ultrassónicos e infravermelhos

(Iwakura et al. 2011). No entanto, muitas destas dependem de infraestrutura externa, são sensíveis a interferências e atrasos de transmissão ou implicam custos elevados. Por oposição, a visão computacional destaca-se como uma alternativa leve e de baixo custo, que permite a perceção do ambiente sem necessidade de comunicações e sensores mais complexos. Esta característica torna-a particularmente adequada para cenários marítimos e ambientes sem cobertura GNSS.

A visão computacional é uma das várias soluções tecnológicas que não só aumenta a capacidade de operação autónoma de VANTs como reforça a sua característica de versatilidade e capacidade de adaptação em cenários dinâmicos (Mohsan et al. 2022). Estes avanços impulsionam uma mudança no paradigma relativamente à condução de operações navais, levando a uma maior eficiência e eficácia da própria execução e da segurança.

Assim, a investigação e conceção de soluções para os desafios associados à execução de operações de VANTs em ambientes marítimos constituem uma área estratégica e de elevada relevância no domínio das ciências militares (Mohsan et al. 2022). Apesar do contributo para o avanço do conhecimento realizado nesta tese, trata-se de um problema em aberto que continua a merecer atenção da comunidade científica.

1.2 Motivação e relevância para a Marinha Portuguesa

Tanto a nível nacional como internacional, os VANTs passaram a ser um elemento crucial em operações de defesa e vigilância (Rosa, Marques e Victor Lobo 2016). Atualmente, para as Forças Armadas Portuguesas, mais concretamente para a Marinha de Guerra Portuguesa, o emprego de VANTs tem o potencial de proporcionar capacidades acrescidas de monitorização e apoio em operações de salvamento e vigilância marítima, nomeadamente através da redução dos custos operacionais, da diminuição da dependência de operadores humanos e da possibilidade de realizar monitorização contínua e persistente. No entanto, este tipo de operações suscita um conjunto de desafios próprios, exigindo o recurso a tecnologias mais inovadoras e robustas (Lyu et al. 2023).

A dependência dos VANTs em relação a sistemas GNSS nas diversas fases das suas operações, como a aterragem, é um dos primeiros obstáculos, uma vez que, em cenários de guerra estes sistemas podem ser afetados por empastelamento

ou por outro tipo de interferências. Este tipo de vulnerabilidade tem sido amplamente reportado em teatros de guerra como a Ucrânia e a região do Estreito de Taiwan (Defense 2024), levando a que a fiabilidade e segurança da operação possa ser comprometida (Zhou et al. 2024).

Atualmente, a Marinha de Guerra Portuguesa já emprega VANTs em diferentes missões, mas a capacidade de aterragem autónoma destes sistemas ainda não se encontra desenvolvida. A criação de um sistema automático e fiável, independente de GNSS e com precisão ao nível dos centímetros, revela-se assim essencial para aumentar a segurança e reduzir a dependência de operadores humanos durante a fase mais crítica do voo. Esta dissertação não se limita a um avanço tecnológico, mas constitui sobretudo uma contribuição para o reforço da capacidade operacional da Marinha, permitindo missões com VANT mais independentes e seguras em cenários navais.

1.3 Objetivos

A presente dissertação de mestrado tem como objetivo principal o desenvolvimento de um sistema de aterragem automática com base em visão computacional para VANTs multirotores. O sistema foi concebido para garantir aterragens precisas em plataformas tanto estáticas como móveis, considerando a dinâmica do navio e as restrições associadas a zonas de aterragem reduzidas, com um erro de aterragem inferior a 20 cm. Deste modo, pretende-se aumentar a fiabilidade e a segurança das operações em cenários complexos, como ambientes com negação do sinal de GNSS e/ou durante aterragens em plataformas móveis, como navios.



FIGURA 1.1: Blocos Funcionais do Sistema.

Para este fim, foi desenvolvido um sistema de aterragem automática baseado em visão computacional, estruturado em quatro blocos funcionais (Figura 1.1): (i) percepção visual, responsável pela deteção de um padrão conhecido no solo (marcadores visuais) e pela estimativa da pose relativa aos mesmos; (ii) planeamento de trajetória, gerando perfis de descida suaves e compatíveis com as restrições operacionais; (iii) controlo de alto nível, que converte as trajetórias em comandos de

posição para o VANT; e (iv) autopiloto, incumbido da estabilização e execução dos comandos de forma robusta.

Este sistema foi concebido para garantir aterragens precisas em plataformas tanto estáticas como móveis, considerando a dinâmica do navio e as restrições associadas a zonas de aterragem reduzidas, com um erro de aterragem inferior a 20 cm.

1.4 Desafios

O desenvolvimento de um sistema de aterragem automática para VANTs, com o objetivo final de operar em cenários complexos como a aterragem num navio, suscita um conjunto significativo de desafios técnicos e operacionais. A deteção visual do marcador utilizado como referência para a aterragem constitui um dos principais desafios. A presença de variações na iluminação, reflexos no convés, movimento relativo entre o VANT e a plataforma de aterragem e possíveis oclusões parciais tornam a estimativa de pose uma tarefa sensível a ruído (erro intrínseco do sensor) e a condições visuais adversas (fatores externos do ambiente).

Adicionalmente, a estimativa da pose relativa entre o VANT e o marcador requer elevada precisão. Mesmo erros de pequena magnitude na orientação ou na posição estimada podem comprometer a trajetória de aproximação, resultando em desvios críticos na fase final da aterragem.

Outro desafio fundamental reside na sincronização entre diferentes fontes de dados. Embora a estimativa de pose seja obtida a partir da câmara, a sua utilização eficaz requer alinhamento temporal com a telemetria do VANT, que fornece informações de estado relevantes para validação, filtragem e controlo. Como estes dados são gerados de forma assíncrona, com diferentes taxas de amostragem e atrasos associados, a correta sincronização é crucial para que cada pose estimada corresponda ao estado real do sistema no instante de cada *frame* de vídeo.

Adicionalmente, o sistema deve processar a informação em tempo real para gerar comandos de alto nível com baixa latência. Numa fase inicial, esta exigência é assegurada por processamento concorrente na estação terrestre, com *threads* dedicadas à aquisição de imagem, deteção do marcador visual, estimativa/filtragem da pose e planeamento/controlo. Contudo, com o propósito de assegurar a autonomia plena do VANT, está planeada a transição para uma plataforma computacional a bordo. Esta transição reduz a capacidade de processamento e o orçamento energético, impondo não só otimização de implementação, mas sobretudo a escolha de

algoritmos mais eficientes. A necessária simplificação algorítmica pode comprometer a fiabilidade.

A complexidade do sistema aumenta significativamente com a introdução do movimento da plataforma de aterragem. Mesmo quando o navio se encontre sem seguimento, o convés permanece sujeito a movimentos contínuos induzidos pela ação da ondulação e vento. Este tipo de cenário introduz variações na pose da plataforma exigindo, capacidade do sistema acompanhar a plataforma em tempo real, ajustando continuamente a trajetória do VANT em que se torna um requisito crítico que a precisão da trajetória de aterragem seja elevada.

Por fim, importa referir o desafio associado à transição do ambiente de simulação para o mundo real. Enquanto no âmbito da simulação é possível controlar todas as variáveis envolvidas, nos testes reais surgem inevitavelmente falhas de comunicação, variações atmosféricas e, sobretudo, limitações inerentes aos sensores, como ruído e erro de medição. Estas discrepâncias exigem uma etapa de validação meticulosa e a implementação de mecanismos de deteção de falhas e de redundância, com o objetivo de garantir a segurança da operação.

Além dos desafios técnicos inerentes à dissertação, é também importante salientar o caráter marcadamente multidisciplinar desta dissertação, que exigiu a aquisição e aplicação de conhecimentos que extravasam os conteúdos habitualmente abordados durante a formação académica do autor. Para além da exploração de conceitos de visão computacional, foi necessário compreender em profundidade os componentes físicos e eletrónicos dos VANTs, os seus sensores embarcados, os princípios de funcionamento e os métodos de controlo. Adicionalmente, revelou-se particularmente exigente o domínio das ferramentas de simulação e desenvolvimento utilizadas. A fusão destas ferramentas num ambiente de simulação do tipo *software-in-the-loop* (SITL), de forma funcional, sincronizada e realista, constituiu um dos grandes desafios técnicos do projeto, implicando um esforço considerável de aprendizagem, integração e resolução de problemas ao longo do desenvolvimento. Esta complexidade refletiu-se não apenas na configuração inicial dos sistemas, mas também na sua manutenção e compatibilidade com os restantes módulos da arquitetura desenvolvida.

1.5 Trabalhos publicados

Durante o desenvolvimento desta dissertação de mestrado, foram feitas tuas publicações científicas relacionados com o tema em estudo. A primeiro, elaborada

numa fase inicial da dissertação de mestrado, foi aceite e posteriormente apresentado pelo autor no *VI Encontro Anual de Investigação & Desenvolvimento em Ciências Militares (EI&DCM)*, um evento científico promovido pelo Instituto Universitário Militar (IUM), que visa dinamizar a investigação em Ciências Militares e fomentar a cooperação entre unidades militares e entidades externas, contribuindo para a missão das Forças Armadas e da Guarda Nacional Republicana. A segunda, foi aceite para apresentação na conferência internacional *OCEANS 2025 Great Lakes*, um evento de referência dedicado à partilha de conhecimento e inovação entre profissionais do setor marítimo, com especial ênfase na proteção e utilização sustentável dos recursos oceânicos. A conferência está agendada para setembro, na qual o artigo será apresentado publicamente e presencialmente no âmbito das sessões técnicas do programa científico.

1.6 Estrutura

Esta dissertação organiza-se em vários capítulos, que por sua vez abordam elementos fundamentais do estudo, expostos da seguinte forma:

Capítulo 2 – Fundamento Teórico: Este capítulo apresenta os conceitos fundamentais necessários à compreensão da tese, incluindo os princípios de funcionamento dos VANTs, visão computacional, deteção de marcadores, estimativa de pose e filtragem de dados. São também explicados o princípio de funcionamento dos sensores utilizados e os conceitos de controlo aplicados à aterragem automática;

Capítulo 3 – Estado da Arte: Neste capítulo é feita a revisão do estado da arte relativamente à aterragem automática de VANTs, com destaque para operações em plataformas móveis, como navios. São analisados trabalhos anteriores que utilizam câmaras a bordo do VANT, deteção visual de marcadores, estimativa de pose, sistemas de cinemática em tempo real e algoritmos de controlo em tempo real;

Capítulo 4 – Formulação do problema: Face à complexidade do problema abordado, foi dedicado um capítulo próprio. Este capítulo formaliza o problema, que consiste em desenvolver um sistema de aterragem automática de VANTs com base em visão computacional, capaz de operar inicialmente com processamento no solo e posteriormente a bordo do VANT. São descritos os requisitos do sistema, as suposições consideradas, bem como os principais desafios técnicos da aterragem numa plataforma móvel;

Capítulo 5 – Metodologia: O capítulo descreve as etapas de desenvolvimento do sistema, incluindo uma explicação sucinta da abordagem em ambiente simulado, a integração da câmara a bordo e a detecção dos marcadores. São abordados os métodos de calibração da câmara, a estimativa da pose com e a filtragem temporal, a sincronização dos dados de vídeo com a telemetria da aeronave e a geração da trajetória de aterragem e os comandos de controlo de alto nível;

Capítulo 6 – Resultados experimentais: Este capítulo apresenta a validação do sistema proposto através de testes em simulação e em voo real. Analisam-se os dados de voo do ArduPilot com os valores de referência do simulador para conferir a precisão da visão, com e sem filtragem e comparam-se os resultados obtidos para trajetórias de voo e manobras de aterragem.

Capítulo 7 – Conclusões e Trabalho Futuro: Este capítulo apresenta uma síntese dos principais contributos da dissertação, refletindo sobre os resultados obtidos e a eficácia do sistema proposto. São também discutidas as limitações identificadas durante os testes e sugeridas direções para trabalhos futuros, com vista ao aperfeiçoamento e operacionalização do sistema em cenários mais exigentes.

Capítulo 2

Fundamento Teórico

O presente capítulo apresenta os fundamentos teóricos indispensáveis à compreensão das metodologias aplicadas ao longo desta dissertação. Dada a natureza multidisciplinar do trabalho, procede-se à sistematização dos principais conceitos provenientes das áreas dos VANTs, visão computacional e controlo aplicado a sistemas de aterragem automática.

Neste capítulo apresentam-se os conceitos fundamentais que sustentam o trabalho desenvolvido. Na Secção 2.1 é feito um enquadramento dos VANTs, abordando a sua classificação e principais componentes. A Secção 2.2 introduz a visão computacional e o seu papel em sistemas autónomos, enquanto a Secção 2.3.1 descreve os marcadores fiduciais como ferramenta de apoio à deteção visual. Na Secção 2.4 discutem-se métodos de estimativa com base em imagens e técnicas de filtragem temporal. Por fim, a Secção 2.5 apresenta os princípios de controlo aplicados à aterragem automática e a necessidade de integração entre perceção e planeamento de trajetória

2.1 Veículos Aéreos Não Tripulados

Os VANTs são aeronaves concebidas para operar sem piloto a bordo, podendo ser comandados a partir de uma estação remota ou dotados de sistemas que lhes permitem executar tarefas de forma autónoma. O VANT representa apenas a componente aérea de um sistema mais abrangente designado por *Unmanned Aircraft System* (UAS). Este sistema engloba não só a aeronave, mas também todos os elementos necessários para a sua operação, nomeadamente: a aeronave, a estação de controlo, os sistemas de comunicação e os sensores (Fahlstrom, Gleason e Sadraey 2022). Um VANT é constituído por quatro principais componentes: a estrutura aerodinâmica, o sistema de propulsão, sensores de navegação e unidade de controlo

(Austin 2011). Pela capacidade de operar em ambientes de difícil acesso, versatilidade e um custo reduzido face às aeronaves convencionais, tem havido uma impulsão crescente da sua adoção em contextos civis e militares. Tais aplicações podem ser desde agricultura, monitorização de infraestruturas, transporte, segurança pública, missões de busca e salvamento e vigilância marítima (Mohsan et al. 2022).

Nesta dissertação de mestrado, utilizou-se um multirrotor, que dispõe de elevada manobrabilidade, voo estacionário e facilidade de controlo (Fahlstrom, Gleason e Sadraey 2022) e por ser o modelo atualmente disponível para experimentação e em utilização na Marinha Portuguesa. Em complemento, a configuração e dimensões deste VANT especificamente facilita a instalação de sensores adicionais e unidades de processamento a bordo, tornando-o assim adequado para tarefas complexas, como a aterragem automática.

2.1.1 Tipos e Características

Neste subcapítulo, são abordados os principais tipos de VANTs mais relevantes do ponto de vista técnico e operacional, com base na classificação proposta por (Hassanalian e Abdelkefi 2017), mas sem cobrir exaustivamente todas as variantes não convencionais. Os VANTs podem ser classificados com base em diversos critérios, tais como a sua configuração aerodinâmica, dimensões e peso, tipo de propulsão, autonomia de voo e aplicação (Hassanalian e Abdelkefi 2017).

No que diz respeito à configuração aerodinâmica, destacam-se três categorias principais: asa fixa, asa rotativa e sistemas híbridos. VANTs de asa fixa oferecem maior autonomia e velocidade, sendo adequados para missões de vigilância e cobertura de grandes áreas. VANTs de asa rotativa, que incluem helicópteros de rotor único e multirrotores, são marcados pela capacidade de voo estacionário, elevada manobrabilidade e capacidade de descolagem e aterragem vertical, (*Vertical Take-Off and Landing*, VTOL). Já os híbridos, como os *tilt-rotor* (representado na Figura 2.1), procuram combinar a eficiência aerodinâmica das asas fixas com a versatilidade operacional dos sistemas VTOL (Hassanalian e Abdelkefi 2017).



FIGURA 2.1: Exemplo de VANT *tilt-rotor* (China Military 2024).

Quanto à dimensão e peso, os VANTs são classificados também de diferentes formas, tendo Arjomandi et al. 2006 classificado da seguinte forma: Micro (<5 kg), leve (5-50 kg), médio (50-200 kg), pesado (200-2000 kg) e super pesado (>2000 kg).

Relativamente às aplicações podem variar desde o uso militar (reconhecimento, ataque, vigilância), até tarefas civis, como mapeamento agrícola, monitorização ambiental, entregas logísticas, entre outras. Estas aplicações influenciam diretamente o design, sensores embarcados e autonomia necessária do VANT (Hassanalian e Abdelkefi 2017).

Os VANTs também podem ser classificados segundo o tipo de propulsão, nomeadamente motores elétricos, motores a combustão ou sistemas híbridos, sendo esta escolha condicionada pela missão, eficiência energética e requisitos de carga útil (Hassanalian e Abdelkefi 2017).

Em relação à autonomia, os VANTs variam desde sistemas totalmente pilotados remotamente passando por configurações semiautónomas com seguimento de *waypoints*, até VANTs completamente autónomos, dotados de sensores e algoritmos de controlo que lhes permitem tomar decisões e adaptar-se dinamicamente ao ambiente durante a missão, sem necessidade de intervenção humana direta (Hassanalian e Abdelkefi 2017).

De seguida apresenta-se a Tabela 2.1, que resume a classificação dos VANTs segundo diferentes critérios.

TABELA 2.1: Resumo da classificação dos VANTs segundo diferentes critérios (Arjomandi et al. 2006; Hassanalian e Abdelkefi 2017).

Critério de Classificação	Categorias / Descrição
Configuração Aerodinâmica	- Asa fixa, Asa rotativa e Sistemas híbridos.
Dimensão e Peso	- Micro, leve, médio, pesado e super pesado.
Aplicação	- Militar: reconhecimento, vigilância e ataque; - Civil: mapeamento agrícola, monitorização ambiental, entregas logísticas, entre outras.
Tipo de Propulsão	- Motores elétricos, motores a combustão ou sistemas híbridos. A escolha depende da missão, eficiência energética e requisitos de carga útil.
Autonomia Operacional	- Controlado remotamente; - Semiautónomo (seguimento de pontos); - Completamente autónomo (com sensores e algoritmos de controlo para adaptação dinâmica ao ambiente).

2.1.2 Sistemas de Propulsão e Controlo

O sistema de propulsão de um VANT é responsável por gerar a força necessária para manter o voo e realizar manobras. Nos VANTs de asa rotativa, como multirrotores, predominam os motores elétricos sem escovas (*brushless*) devido à sua elevada eficiência, baixo peso e custo de manutenção reduzido. (Hassanalian e Abdelkefi 2017).

Cada motor elétrico de um VANT é regulado individualmente por um controlador de velocidade eletrónico (*Electronic Speed Controller*, ESC), responsável por converter os sinais de controlo recebidos do controlador de voo em ajustes precisos na velocidade de rotação do motor. Na prática, o controlador de voo envia sinais de comando, que são interpretados pelos ESCs para determinar a quantidade de corrente a ser fornecida ao motor correspondente. Este processo permite controlar a aceleração, a desaceleração e o torque de cada motor em tempo real, assegurando a execução das manobras desejadas pelo sistema de navegação e controlo.

A estrutura de controlo do VANT é coordenada por um sistema de autopiloto, como o Pixhawk, que integra diversos sensores e software embarcado. Este sistema inclui uma unidade de medição inercial (*Inertial Measurement Unit*, IMU), composta por acelerómetros e giroscópios triaxiais, responsáveis pela medição da aceleração linear e da velocidade angular, respetivamente. Frequentemente, incorpora ainda um magnetómetro, que contribui para a estimativa da orientação em relação ao campo magnético terrestre e um barómetro, utilizado para estimar a altitude relativa com base na pressão atmosférica. O autopiloto executa um *firmware* de navegação, como o PX4 ou o ArduPilot, carregado no microcontrolador, que monitoriza continuamente a atitude e a posição do VANT, processando os dados sensoriais para gerar comandos corretivos enviados aos motores. Em complemento, softwares externos executados num computador de bordo ou numa estação de controlo comunicam com esse *firmware*, permitindo ao operador supervisionar o voo, definir missões e ajustar parâmetros.(B. Zhang, Song e Zhao 2022).

Em cenários de negação do sinal de GNSS, como zonas interiores, áreas urbanas densas ou ambientes de conflito armado, torna-se necessário recorrer à integração de sensores alternativos, como câmaras, sensores óticos e inerciais, para estimar a pose da aeronave (Nguyen et al. 2017). Embora a fusão dos dados provenientes destes sensores possa ser realizada através de diversas abordagens os filtros temporais. Os filtros temporais são métodos matemáticos que utilizam medições presentes e passadas para estimar o estado atual de um sistema dinâmico (Ribeiro 2004). Deste modo, são amplamente utilizados devido à sua capacidade de integrar medições ruidosas e parciais ao longo do tempo, produzindo estimativas mais robustas e estáveis da pose. Para além da fusão multissensorial, estes filtros permitem a atenuação de ruído na trajetória estimada e a interpolação de dados perdidos, aspetos fundamentais em sistemas de controlo em tempo real e em ambientes operacionais degradados (Park et al. 2024).

A Figura 2.2 apresenta a arquitetura típica de ligação do sistema de controlo Pixhawk, evidenciando os principais módulos funcionais de um VANT multirrotor. São visíveis as ligações entre o controlador de voo Pixhawk e elementos como os ESCs, os módulos de receção de rádio e telemetria, utilizados na comunicação com a estação de controlo e o sistema de alimentação e distribuição de energia. O diagrama mostra ainda interfaces para sensores externos como o Sistema de Posicionamento Global (*Global Positioning System* GPS), *gimbal* e câmara, ilustrando a integração necessária para a operação autónoma ou remotamente assistida do veículo. Toda a gestão de sensores, atuadores e comunicações é executada por um *firmware* de controlo em tempo real, como o *ArduPilot*, instalado no controlador

Pixhawk (ArduPilot Developers 2014).

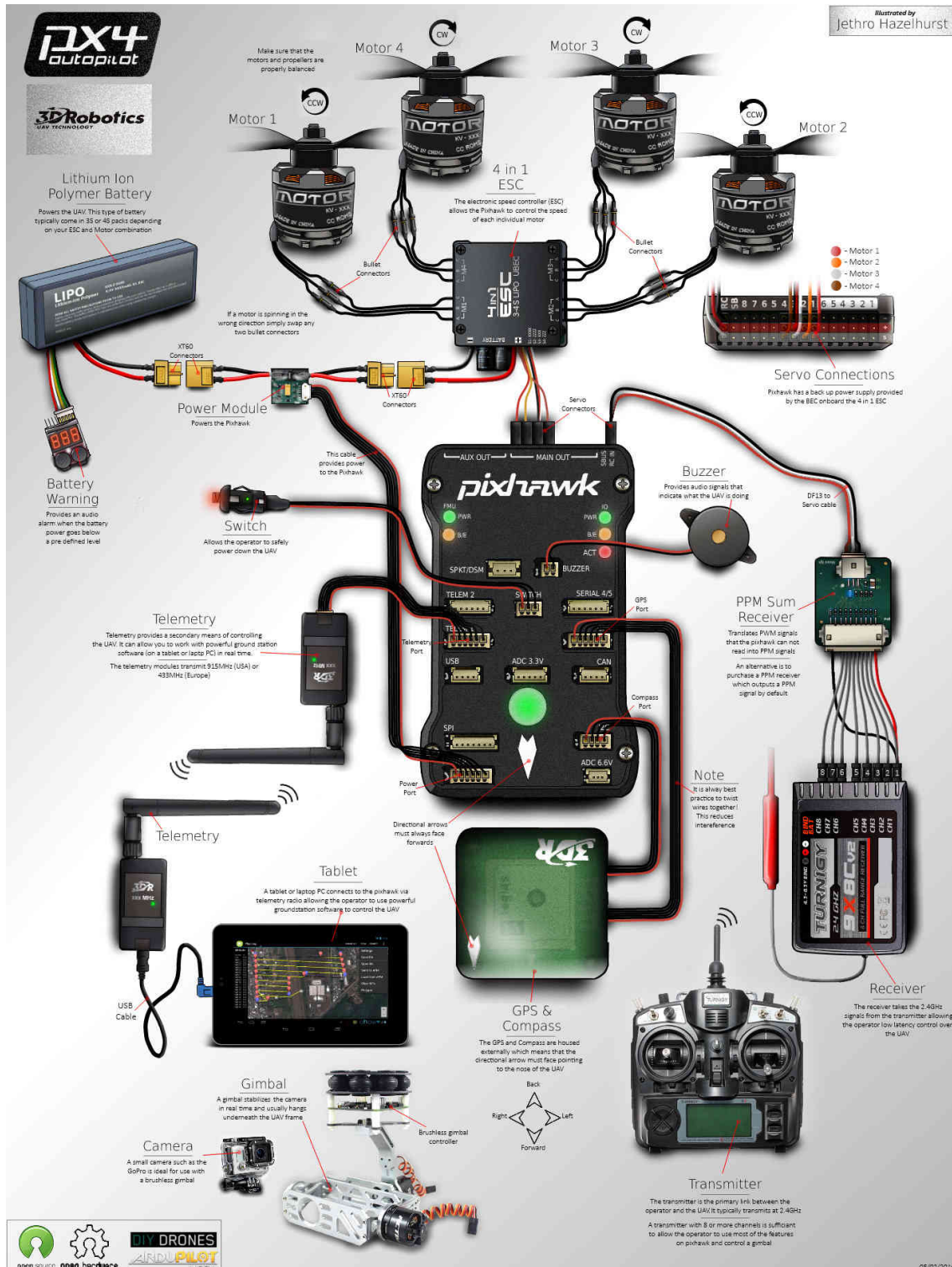


FIGURA 2.2: Diagrama de ligação dos principais componentes do autopiloto Pixhawk (ArduPilot Developers 2014).

Deste modo, o sistema de controlo de um VANT combina componentes eletromecânicos, sensores e algoritmos avançados de estimativa e controlo em tempo real, incluindo técnicas de filtragem temporal e fusão sensorial, com o objetivo de assegurar estabilidade, manobrabilidade e capacidade de adaptação a diferentes ambientes operacionais.

2.1.3 Navegação e Sensores a bordo

A navegação e os sensores a bordo dos VANTs constituem a base para operações autónomas e precisas. O posicionamento global é geralmente garantido por sistemas GNSS (como GPS, Galileo e BeiDou), mas a sua eficácia pode ser limitada em ambientes urbanos densos ou interiores devido à perda ou degradação do sinal ou por interferências. Para mitigar estas limitações, recorrem-se às IMUs, que permitem estimar o deslocamento e a orientação do VANT, embora sofram de deriva acumulada, isto é, um erro crescente resultante da soma de pequenas imprecisões nas medições. (P. Wang et al. 2025).

Além das IMUs e dos recetores GNSS, os VANTs recorrem frequentemente a altímetros barométricos para estimar a altitude. No entanto, estas medições podem ser afetadas por variações atmosféricas ou interferências eletromagnéticas. Em ambientes onde o sinal GNSS está ausente ou degradado, recorrem-se a métodos alternativos de perceção e navegação. Entre eles, destaca-se o uso de câmaras, que permitem a estimativa de pose através de técnicas de visão computacional, e sensores LiDAR, que emitem pulsos laser para medir distâncias e gerar nuvens de pontos tridimensionais do ambiente. Estas abordagens viabilizam técnicas de mapeamento e localização simultâneos (*Simultaneous Localization and Mapping*, SLAM) (Lou et al. 2023).

Apesar dos avanços recentes, persistem desafios técnicos relevantes em sistemas de navegação autónoma, nomeadamente a deriva inercial nos sensores da IMU, os erros por multipercurso em sistemas GNSS, resultantes da reflexão dos sinais em superfícies, como edifícios ou o solo e a dessincronização temporal entre sensores heterogéneos, como câmaras, LiDARs, magnetómetros, barómetros e IMUs (Lu 2018).

2.2 Visão Computacional

A visão computacional é uma área da ciência que procura capacitar sistemas a interpretar, processar e extrair informações relevantes de imagens e vídeos. Para

tal, são utilizados algoritmos matemáticos para se extrair informação tridimensional, como a forma, a cor e a iluminação (Szeliski 2022). Em VANTs, esta tecnologia desempenha um papel central ao ser uma alternativa para a navegação autónoma, o seguimento de alvos e a aterragem precisa, especialmente em ambientes de negação do sinal de GNSS (Kendall, Grimes e Cipolla 2015). Ao extrair informação contextual do ambiente a partir de imagens ou vídeos, a visão computacional torna-se um elemento crucial para a automatização e segurança de VANTs em cenários adversos. Esta abordagem permite identificar zonas seguras de aterragem mesmo em ambientes degradados, como os encontrados em situações de catástrofe, viabilizando operações de busca e salvamento sem dependência de GNSS ou mapas prévios (Zhu, Bétaille e Berbineau 2018).

2.2.1 Modelo da câmara

A modelação geométrica da câmara é essencial para projetar com precisão pontos tridimensionais no plano da imagem e, consequentemente, para realizar uma estimativa da pose. O modelo *pinhole* constitui a base dessa modelação, representando a câmara como um centro de projecção ideal e utilizando uma matriz intrínseca para descrever a relação entre o mundo 3D e a imagem 2D. No entanto, devido às imperfeições óticas das lentes reais, é necessário considerar distorções radiais e tangenciais que afetam a fidelidade das projecções. Estas distorções são modeladas matematicamente e corrigidas através de um processo de calibração, que permite estimar os parâmetros intrínsecos e os coeficientes de distorção da câmara. Este processo, habitualmente realizado com padrões planos como tabuleiros de xadrez, é suportado por bibliotecas como a OpenCV e assegura a correção das imagens adquiridas, aumentando a fiabilidade da estimativa de pose visual (Guo et al. 2025; OpenCV Documentation Team 2025).

Modelo *pinhole*

O modelo de câmara *pinhole* constitui o modelo utilizado para descrever a projecção de pontos tridimensionais no plano da imagem. Este modelo, amplamente adotado em visão computacional (Szeliski 2022), representa a câmara como um centro de projecção único, ignorando os efeitos físicos complexos de lentes reais. A formação da imagem é assumida como uma projecção central, em que um ponto tridimensional X do mundo é projetado linearmente no plano da imagem através do centro de projecção C , considerando uma distância focal f (Hartley e Zisserman

2003). A Figura 2.3 ilustra esta geometria, na qual a relação entre a cena tridimensional e a imagem é descrita de forma determinística. Em aplicações práticas, como a que se descreve nesta dissertação, é comum complementar este modelo com termos de distorção ótica, de modo a representar o comportamento real das lentes de uma câmara (Z. Zhang 2000).

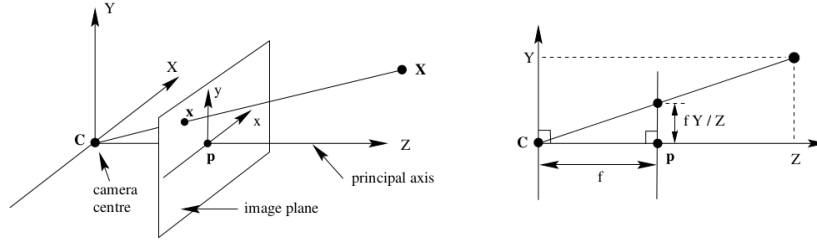


FIGURA 2.3: Projção de um ponto 3D no plano da imagem utilizando o modelo *pinhole* (Hartley e Zisserman 2003).

Este modelo oferece uma base matemática que descreve a relação entre as coordenadas de um ponto no mundo tridimensional $[X \ Y \ Z]^T$ e as suas coordenadas projetadas na imagem captada pela câmara $[u \ v]^T$. Esta relação pode ser expressa na forma matricial homogênea utilizando coordenadas homogêneas que acrescentam uma dimensão extra aos vetores de pontos, permitindo descrever projeções e transformações geométricas através de multiplicações matriciais, da seguinte forma:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{R} | \mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.1)$$

Nesta equação, λ é um fator de escala arbitrário, necessário devido ao uso de coordenadas homogêneas. A matriz $\mathbf{K}$ é a matriz intrínseca da câmara, tendo sido dada por:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Onde, f_x e f_y correspondem às distâncias focais expressas em píxeis ao longo dos eixos horizontal e vertical, respetivamente. Os termos c_x e c_y indicam as coordenadas do centro ótico (ou ponto principal) da imagem no referencial do sensor. O parâmetro s representa o fator de inclinação entre os eixos da imagem, geralmente assumido como nulo em câmaras com píxeis retangulares ortogonais.

A matriz $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ representa a rotação da câmara, definindo a sua orientação no espaço tridimensional. A sua forma geral é:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.3)$$

O vetor $\mathbf{t} \in \mathbb{R}^{3 \times 1}$ representa a translação da câmara, indicando a posição do seu centro ótico em relação ao referencial do mundo. A sua forma geral é:

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (2.4)$$

Com base nesta modelação, é possível estimar a pose da câmara em relação a pontos conhecidos no mundo tridimensional, desde que os parâmetros intrínsecos sejam previamente obtidos através do processo de calibração e que haja correspondência entre pontos do mundo e pontos da imagem. Os efeitos de distorção ótica, que afetam a precisão dessa projeção, serão descritos de seguida.

Distorções óticas

Apesar da simplicidade do modelo *pinhole*, as câmaras reais apresentam imperfeições óticas introduzidas pelas lentes. Estas imperfeições, conhecidas como distorções óticas (Guo et al. 2025), afetam a precisão da projeção dos pontos tridimensionais no plano da imagem e devem ser modeladas para que as estimativas geométricas se aproximem da realidade (Guo et al. 2025). As distorções mais relevantes em sistemas de visão computacional são:

- **Distorção radial:** Provoca o encurvamento de linhas retas, particularmente nas extremidades da imagem. Este fenómeno resulta da forma esférica das lentes;

- **Distorção tangencial:** É causada por pequenos desalinhamentos entre o eixo ótico da lente e o plano do sensor, fazendo com que os pontos da imagem pareçam deslocados.

Matematicamente, estas distorções são descritas por funções polinomiais em torno do centro ótico. A partir de coordenadas ideais (x, y) , os pontos projetados segundo o modelo *pinhole* sem imperfeições, obtêm-se as coordenadas distorcidas (x_d, y_d) , que correspondem às posições efetivamente registradas no sensor da câmara após a influência das distorções óticas das lentes.

$$\begin{aligned} x_d &= x \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \right) + 2p_1 xy + p_2 (r^2 + 2x^2) \\ y_d &= y \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \right) + p_1 (r^2 + 2y^2) + 2p_2 xy \end{aligned} \quad (2.5)$$

onde $r^2 = x^2 + y^2$, e os coeficientes k_1, k_2, k_3 representam a distorção radial, enquanto p_1, p_2 descrevem a distorção tangencial (Hartley e Zisserman 2003).

Este modelo de distorção é amplamente adotado em aplicações de visão computacional. Sendo que a forma como estes parâmetros são estimados é apresentada na secção seguinte.

Calibração de câmara

A calibração de câmara é o processo que permite estimar os parâmetros que caracterizam o sistema de formação de imagem, os parâmetros intrínsecos dados pela matriz $\mathbf{K}$ e os coeficientes de distorção ótica $\mathbf{d} = [k_1, k_2, p_1, p_2, k_3]^T$. Este processo é essencial para garantir que os modelos geométricos aplicados às imagens correspondem com precisão à realidade física da câmara.

No contexto desta dissertação, o método adotado para calibração é o proposto por Z. Zhang 2000, amplamente utilizado devido à sua flexibilidade e simplicidade experimental. Este método permite estimar os parâmetros da câmara a partir de múltiplas imagens de um padrão planar com geometria conhecida, tipicamente um tabuleiro de xadrez, observado sob diferentes poses. A única exigência é que o padrão seja rigidamente plano e bem visível nas imagens, sem necessidade de conhecer a posição relativa exata entre a câmara e o padrão em cada captura.

O algoritmo baseia-se na estimativa de várias homografias entre o padrão de calibração e o plano da imagem, a partir das quais se obtém a matriz intrínseca $\mathbf{K}$. Uma homografia corresponde a uma transformação projetiva que relaciona pontos de dois planos, neste caso o do padrão físico e o da imagem captada. Posteriormente,

são também estimados os coeficientes de distorção radial e tangencial (Z. Zhang 2000).

Este procedimento está implementado em bibliotecas como a OpenCV, que automatizam as etapas de detecção de padrões, estimativa dos parâmetros e avaliação do erro de reprojeção. A qualidade da calibração pode ser verificada através da análise desse erro, que quantifica a discrepância entre as posições reais dos pontos no plano da imagem e as posições previstas pelo modelo calibrado.

2.2.2 Reconhecimento de Padrões e Segmentação

O reconhecimento de padrões e a segmentação de imagens são tarefas centrais em visão computacional. O reconhecimento de padrões refere-se à capacidade de identificar automaticamente objetos, formas ou estruturas específicas em imagens, como marcadores visuais, alvos ou características geométricas relevantes. Esta capacidade é essencial mas desafiadora para tarefas como seguimento visual, detecção de zonas de aterragem ou identificação de obstáculos, permitindo que o sistema extraia significado a partir de dados visuais (Szeliski 2022).

Técnicas tradicionais de segmentação incluem limiarização, detecção de contornos, análise morfológica e agrupamento de pixéis com base em histogramas ou medidas estatísticas (Oliveira Filho 2017). A limiarização consiste na conversão de uma imagem em tons de cinzento para uma imagem binária, com base num valor de limiar, separando os pixéis em regiões de interesse e fundo conforme a sua intensidade luminosa (Szeliski 2022). Esta técnica é particularmente eficaz quando o contraste entre objeto e fundo é acentuado. A análise morfológica, por sua vez, aplica operadores sobre imagens binárias com o objetivo de remover ruído, preencher lacunas ou reforçar estruturas, com base na forma geométrica dos objetos presentes na imagem (Szeliski 2022).

Nos últimos anos, abordagens baseadas em redes neuronais convolucionais, (*Convolutional Neural Networks*, CNNs) têm demonstrado resultados promissores tanto em reconhecimento de padrões como em segmentação, especialmente em ambientes complexos ou não estruturados (Kakaletsis et al. 2021). No entanto, em sistemas com limitações computacionais, continuam a ser amplamente utilizadas soluções computacionalmente menos exigentes, como a detecção de marcadores (Alijani e Osman 2020).

Assim, o reconhecimento de padrões e a segmentação constituem tarefas fundamentais na perceção visual de VANTs, fornecendo informação crítica para o

controle, navegação e interação com o ambiente.

2.3 Marcadores

Os marcadores visuais usados nesta dissertação dividem-se em dois perfis complementares. Por um lado, os marcadores fiduciais, como por exemplo o ArUco e o AprilTag são concebidos para algoritmos, tendo geometria e dimensões conhecidas, criando identificadores únicos e permitem estimar a pose a partir de uma única imagem com elevada robustez (Garrido-Jurado et al. 2014). Por outro lado, o marcador de navegação “H” é uma marcação orientada ao humano, padronizada para apoiar pilotos, geralmente bordo de helicópteros, a sua alta visibilidade é vantajosa operacionalmente, mas não transporta um código próprio nem foi desenhada de raiz para processamento automático, podendo introduzir ambiguidades geométricas em certas condições (Morales et al. 2023).

2.3.1 Marcadores Fiduciais

Os marcadores fiduciais tornaram-se uma solução muito comum para permitir a aterragem automática precisa e fiável de VANTs. Estes marcadores são padrões visuais artificiais, normalmente colocados em plataformas de aterragem, concebidos para serem facilmente detetados e identificados de forma única por câmaras a bordo utilizando algoritmos de visão computacional (Guo et al. 2025). As suas características geométricas marcantes e dimensões conhecidas permitem uma estimativa precisa da posição e orientação relativas (pose) da câmara em relação ao marcador (Sousa 2022). Ao fornecer uma referência visual robusta, os marcadores fiduciais facilitam as fases de aproximação final e aterragem, reduzindo erros de aterragem e aumentando a segurança. Marcadores comuns, como *Augmented Reality University of Cordoba* (ArUco) e AprilTag, representados na Figura 2.4, são suportados por bibliotecas de código aberto que oferecem deteção eficiente e estimativa de pose, tornando-os práticos para aplicações de VANT em tempo real (Guo et al. 2025).

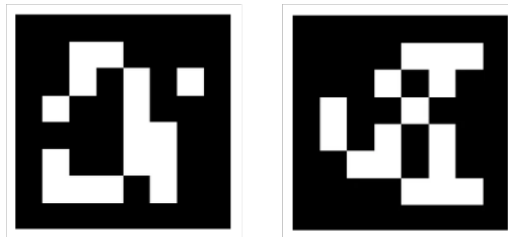


FIGURA 2.4: Marcador ArUco (esquerda) (OpenCV 2024) e Marcador de Navegação “H” (direita) (OpenMV 2020).

Um marcador ArUco padrão é representado por uma matriz binária de 7×7 , na qual a região central de 5×5 bits codifica a informação útil. Em cada linha dessa matriz interna, 2 bits correspondem ao identificador do marcador (ID) e 3 são reservados para detecção de erros, totalizando 10 bits de codificação. Esta estrutura permite gerar até 1024 marcadores distintos, cada um com um identificador único. A possibilidade de dispor de múltiplos IDs constitui uma vantagem face a marcadores visuais com formas únicas, que não permitem distinguir entre diferentes alvos (Sani e Karimian 2017). A estrutura de alto contraste facilita a sua localização em imagens, enquanto o código binário integrado permite a detecção e correção de erros, garantindo maior precisão (Garrido-Jurado et al. 2014).

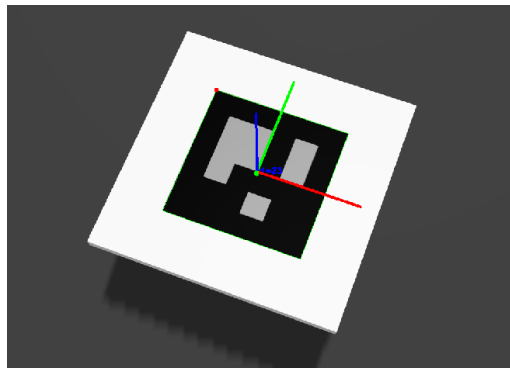


FIGURA 2.5: Marcador ArUco visualizado no gazebo com eixos coordenados desenhados (X-vermelho, Y-verde, Z-Azul).

Uma das principais vantagens dos marcadores ArUco é a sua capacidade de fornecer estimativas precisas da pose da câmara a partir de um único marcador. Tal é possível graças à identificação inequívoca dos quatro cantos do marcador, que permite calcular a posição e orientação relativas da câmara com elevada fiabilidade. Estas características tornam-nos particularmente adequados para aplicações em que a localização precisa é crítica, como na aterragem automática de VANTs, na navegação robótica e na realidade aumentada (Garrido-Jurado et al. 2014).

A biblioteca ArUco, integrada no OpenCV, disponibiliza algoritmos otimizados para a deteção de marcadores e para a estimativa da pose, bem como ferramentas para a calibração de câmaras. Isto facilita a sua adoção em sistemas em tempo real com recursos computacionais limitados (Garrido-Jurado et al. 2014).

Além disso, os marcadores ArUco são económicos, pois podem ser facilmente impressos e utilizados em diversos cenários. A sua versatilidade e escalabilidade tornam-nos uma ferramenta valiosa para tarefas como acompanhamento de objetos, localização e coordenação de múltiplos agentes em ambientes dinâmicos.

2.3.2 Marcador de Navegação H

O marcador H, símbolo padronizado para zonas de aterragem, pode ser utilizado como referência visual para detecção e orientação do VANT em relação à plataforma de aterragem como ilustrado na Figura 2.6. Trata-se de uma solução simples e eficaz, especialmente adequada para ambientes controlados embora não seja um marcador fiducial, mas sim uma marcação orientada ao piloto, este marcador pode ser adaptado à detecção automática a partir de visual computacional, em cenários controlados, sem recorrer a algoritmos avançados. No entanto, apresenta duas limitações relevantes em aplicações de visão computacional às quais se deve ter atenção: a estimativa de pose torna-se imprecisa ou inviável a curtas distâncias, devido à perda de geometria projetiva suficiente e em casos de marcação deficiente (por exemplo, baixo contraste ou desgaste da pintura) ou visualização parcial, pode ocorrer ambiguidade na orientação, dificultando o posicionamento correto do VANT com a plataforma de aterragem (Morales et al. 2023).

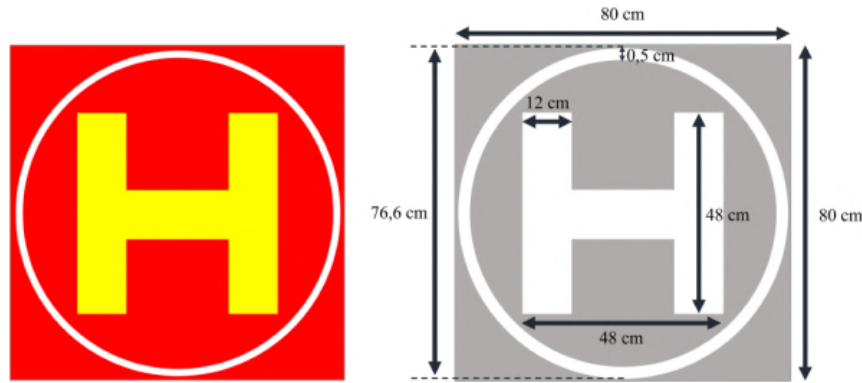


FIGURA 2.6: Ilustração do marcador H e as suas dimensões (M. Vieira 2023)

2.4 Estimativa de pose

A estimativa de pose representa uma componente fundamental em diversas aplicações de visão computacional, como a navegação robótica ou a realidade aumentada (Khazetdinov et al. 2021). Trata-se do processo de determinar a posição e orientação de um objeto no espaço com base em imagens ou vídeos (Khazetdinov et al. 2021; Nuno Pessanha Santos 2020). Em sistemas embarcados, como VANTs, esta tarefa é especialmente crítica durante manobras complexas, como a aterragem automática, onde é necessário garantir elevada precisão e continuidade da informação posicional.

2.4.1 Fundamentos da estimativa baseada em visão

A estimativa de pose visual baseada em pontos consiste em estabelecer correspondências entre pontos 3D conhecidos, como por exemplo, os vértices do marcador e as suas projeções 2D na imagem obtida. A partir dessas correspondências, é possível resolver o problema conhecido como *Perspective-n-Point* (PnP), que determina a transformação de seis graus de liberdade (*six degrees of freedom*, 6-DOF) entre o sistema de coordenadas da câmara e o do objeto conhecido (Szeliski 2022). Este cálculo requer a calibração prévia da câmara, com parâmetros intrínsecos conhecidos. Os seis graus de liberdade correspondem às três componentes de translação e rotação de um objeto (movimentos ao longo dos eixos X, Y e Z e os ângulos de roll, pitch e yaw) que descrevem a pose relativa da câmara em relação ao objeto (Northern Digital Inc 2025).

2.4.2 Filtragem da Estimativa de Pose

A estimativa da pose a partir de sensores visuais, como câmaras monoculares, é frequentemente afetada por ruído, oclusões parciais, variações de iluminação e limitações na precisão dos algoritmos de detecção. Estas perturbações tornam-se particularmente relevantes em ambientes dinâmicos ou com condições adversas, como cenários de negação de GNSS, onde não é possível recorrer a dados de outros sensores de posicionamento. Nestes casos, a variabilidade nas medições pode comprometer significativamente a estabilidade dos controladores embarcados, justificando a necessidade de técnicas de suavização e fusão sensorial.

Uma abordagem clássica para mitigar os efeitos de ruído e interrupções nas medições consiste na aplicação do filtro de Kalman (*Kalman Filter*, KF), um método de estimativa ótimo para sistemas lineares afetados por ruído branco Gaussiano (Maybeck 1979). Este filtro combina iterativamente previsões baseadas num modelo dinâmico com observações ruidosas dos sensores, de forma a obter uma estimativa mais precisa e robusta do estado do sistema. O funcionamento do KF está estruturado em duas etapas principais: Predição e correção, como ilustrado na Figura 2.7.

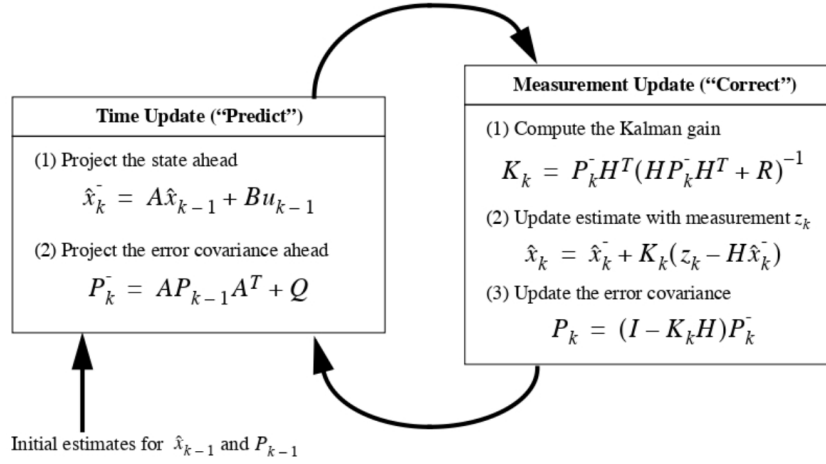


FIGURA 2.7: Fases do Kalman Filter (Welch, Bishop et al. 1995).

Fase de Predição do KF

A fase de predição tem como objetivo projetar o estado atual e a incerteza associada, com base no modelo dinâmico do sistema. O primeiro passo consiste na predição do vetor de estado:

$$\hat{\mathbf{x}}_{k+1}^- = A_k \hat{\mathbf{x}}_k^+ + B \mathbf{u}_k \quad (2.6)$$

Onde, $\hat{\mathbf{x}}_k^+$ representa a estimativa do vetor de estado no instante k , já corrigida com a medição, enquanto $\hat{\mathbf{x}}_{k+1}^-$ corresponde à estimativa *a priori* no instante $k+1$, obtida pela propagação do modelo dinâmico. A matriz A_k é a matriz de transição de estado, B é a matriz de controlo e $\mathbf{u}_k$ é o vetor de entrada de controlo.

De seguida, é feita a predição da matriz de covariância do erro, que estima a incerteza associada à previsão do estado:

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \quad (2.7)$$

Onde, P_k é a matriz de covariância do erro *a posteriori* no instante k , P_{k+1}^- é a covariância *a priori* prevista para o instante $k+1$, A_k é a matriz de transição de estado, e Q_k é a covariância do ruído de processo (Welch, Bishop et al. 1995).

Fase de Correção do KF

A fase de correção inicia-se com o cálculo do ganho de Kalman K_k , que pondera a confiança relativa entre a estimativa predita e a nova observação:

$$K_k = P_k^- H_k^\top (H_k P_k^- H_k^\top + R_k)^{-1} \quad (2.8)$$

onde H_k é a matriz de observação que relaciona o estado com o espaço das medições, R_k é a covariância do ruído de medição (assumido Gaussiano branco) e K_k quantifica o peso atribuído à medição na atualização da estimativa.

Em seguida, a estimativa do estado é atualizada, incorporando a diferença entre a medição real z_k e a previsão da mesma com base no modelo:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \quad (2.9)$$

onde z_k representa a medição observada no instante k , e o termo $z_k - H_k \hat{x}_k^-$ corresponde ao vetor de inovação (ou resíduo), que expressa a diferença entre a medição real e a prevista. A estimativa $\hat{x}_k^+$ corresponde ao estado *a posteriori*.

Por fim, a matriz de covariância do erro é atualizada para refletir a nova incerteza associada ao estado corrigido:

$$P_k = (I - K_k H_k) P_k^- \quad (2.10)$$

onde P_k é a matriz de covariância do estado corrigido e I é a matriz identidade utilizada no processo de correção da incerteza (Welch, Bishop et al. 1995).

O KF clássico é limitado à modelação de sistemas lineares. Contudo, a maioria dos sistemas robóticos, onde se inclui a estimativa de pose, exhibe comportamentos não lineares. Esta não linearidade decorre, por exemplo, da representação da rotação tridimensional por ângulos de Euler ou quatérniões, bem como da projeção de pontos tridimensionais para o plano da imagem segundo o modelo *pinhole*, que envolve relações não lineares entre coordenadas do mundo e coordenadas de imagem.

Nestes casos, o modelo de transição de estado e o modelo de observação são descritos por funções não lineares $f(\cdot)$ e $h(\cdot)$, respetivamente. Para tratar esta complexidade, recorre-se, por exemplo, ao filtro de Kalman estendido (*Extended Kalman Filter*, EKF) (Kalman 1960), que consiste numa generalização do KF clássico. O

EKF realiza uma linearização local das funções não lineares, recorrendo às suas derivadas de primeira ordem, conhecidas como Jacobiano (Apêndice F), avaliadas em torno da estimativa atual do estado (Welch, Bishop et al. 1995).

2.5 Aterragem Automática

A aterragem automática representa uma das fases mais críticas e exigentes na operação de VANTs, especialmente em cenários dinâmicos ou com restrições operacionais severas, como embarcações em movimento ou plataformas de aterragem com dimensões reduzidas (Grindley et al. 2024). Esta manobra requer um elevado grau de precisão e robustez, sendo fundamental garantir a continuidade da navegação, a estabilidade de controlo do VANT e a capacidade de resposta em tempo real.

Para alcançar estes objetivos, é necessária a integração de diversos subsistemas, incluindo estratégias de aproximação, algoritmos de controlo e sistemas de perceção. As estratégias de aproximação definem o perfil da trajetória até ao ponto de aterragem, considerando fatores como altitude, velocidade e orientação, já os controladores são responsáveis por garantir a resposta dinâmica adequada da aeronave face às variações do ambiente e às perturbações externas. Por sua vez, a visão computacional tem vindo a assumir um papel central na estimativa da pose relativa entre o VANT e o alvo de aterragem, fornecendo dados essenciais para a correção em tempo real da trajetória.

2.5.1 Planeamento de Trajetória de aterragem

A definição da trajetória de aproximação é um elemento central na aterragem automática de VANTs, sendo responsável por definir a trajetória e guiar o veículo de forma segura e eficiente até à plataforma de aterragem. Esta trajetória deve ser adaptada à dinâmica do alvo, às condições do ambiente e às capacidades do veículo, podendo assumir diferentes níveis de complexidade consoante a estratégia adotada.

No planeamento de trajetórias para VANTs podem distinguir-se duas abordagens principais: métodos com previsão do movimento do alvo e métodos sem previsão. No primeiro caso, a trajetória é gerada considerando modelos de movimento futuros da plataforma, enquanto no segundo se recorre apenas à pose estimada em tempo real. Em termos de implementação, existem estratégias que utilizam perfis

predefinidos, nos quais o veículo segue trajetórias estáticas e outras que atualizam a trajetória dinamicamente em função da estimativa de pose do alvo.

Para garantir suavidade e continuidade, é comum recorrer a técnicas baseadas em *splines*, como *B-splines* ou *Cubic-splines*, que permitem gerar trajetórias com continuidade de derivadas, facilitando o controlo dinâmico do VANT. Já o controlo preditivo baseado em modelo (MPC) constitui uma alternativa que introduz a capacidade de antecipar o movimento da plataforma e otimizar a trajetória em tempo real, oferecendo robustez acrescida face a perturbações externas.

Em contextos navais ou ambientes de negação de GNSS, a visão computacional é muitas vezes a única fonte de pose relativa, o que exige estratégias robustas ao desaparecimento temporário de marcadores ou à deteção intermitente do alvo (Damas, Santos e M. C. Vieira 2024).

A eficácia da aproximação depende da capacidade do sistema em adaptar-se ao movimento do alvo, em prever o seu comportamento futuro e em corrigir a trajetória de forma contínua e reativa, maximizando a precisão da manobra de aterragem e a segurança operacional do VANT. No contexto da aterragem automática de VANTs, o planeamento da trajetória de aproximação requer não apenas precisão posicional, mas também suavidade no perfil de movimento, de modo a garantir a estabilidade do controlo e a segurança da operação. Para tal, são frequentemente utilizados critérios de suavidade baseados na minimização de derivadas sucessivas da posição, como o *jerk* (terceira derivada da posição) e o *snap* (quarta derivada), os quais contribuem para reduzir transições abruptas na aceleração ou nas forças atuantes (Mellinger e Kumar 2011).

Minimum Jerk

O critério de *minimum jerk* consiste em minimizar a variação da aceleração ao longo do tempo, conduzindo a movimentos mais suaves e com menor risco de instabilidade dinâmica. Este perfil é particularmente adequado a aproximações finais em que o objetivo é garantir um toque suave na plataforma de aterragem, com prioridade dada à precisão e estabilidade, mesmo que à custa de uma menor agressividade nas manobras (Falanga et al. 2017). A função de custo associada é definida por:

$$J = \int_0^T \left\| \frac{d^3 \mathbf{p}(t)}{dt^3} \right\|^2 dt \quad (2.11)$$

onde $\mathbf{p}(t)$ representa a posição do VANT em função do tempo, e T a duração total da manobra. A solução deste problema de otimização corresponde a um polinómio de quinto grau. Este grau resulta da necessidade de satisfazer simultaneamente as condições de contorno de posição, velocidade e aceleração em instantes inicial e final, o que perfaz seis restrições. Um polinómio de quinto grau, com seis coeficientes, é assim a forma mínima capaz de acomodar estas restrições, assegurando continuidade até à aceleração. Esta propriedade conduz a trajetórias suaves, reduzindo variações bruscas de aceleração e garantindo movimentos mais estáveis e fisicamente realizáveis pelo VANT.

Minimum Snap

O critério de *minimum snap*, por sua vez, minimiza a quarta derivada da posição, permitindo gerar trajetórias ainda mais suaves em termos de forças de controlo requeridas. Esta abordagem é especialmente útil em manobras rápidas ou em ambientes sujeitos a rápidas mudanças de orientação e velocidade, onde é necessário assegurar transições contínuas até ao nível do *jerk* (Mellinger e Kumar 2011). A função de custo é dada por:

$$J = \int_0^T \left\| \frac{d^4 \mathbf{p}(t)}{dt^4} \right\|^2 dt \quad (2.12)$$

As soluções obtidas com esta formulação são polinómios de grau sete. Este grau resulta do facto de ser necessário satisfazer simultaneamente as condições de contorno de posição, velocidade, aceleração e *jerk* em instantes inicial e final, o que perfaz oito restrições. Um polinómio de sétimo grau, com oito coeficientes, é assim a forma mínima capaz de acomodar estas restrições, assegurando continuidade até à terceira derivada da posição. Esta suavidade elevada traduz-se numa menor exigência sobre os atuadores e maior robustez em ambientes altamente dinâmicos.

Em síntese, a escolha entre *minimum jerk* e *minimum snap* depende das exigências específicas da missão: o primeiro é preferido em aproximações finais lentas e estáveis, como na fase de aterragem, enquanto o segundo é geralmente mais apropriado para seguimento rápido de trajetórias em cenários com movimento relativo elevado. Ambos os métodos produzem trajetórias suaves e compatíveis com controladores típicos de multirrotores, sendo amplamente utilizados em sistemas de controlo de VANTs (Falanga et al. 2017).

2.5.2 Integração da Visão no Controlo de Aterragem

A pose estimada a partir da imagem é utilizada como entrada no sistema de controlo, permitindo o ajustamento contínuo da trajetória do veículo em função do erro entre a posição atual e a posição de referência. Esta abordagem baseia-se na realimentação da pose observada, possibilitando a correção da pose em tempo real.

Contudo, a pose extraída da imagem encontra-se expressa no referencial da câmara, cujo centro ótico raramente coincide com o centro de massa do VANT (Patruno et al. 2019). Simultaneamente, os comandos de controlo são definidos no referencial do corpo do VANT, geralmente associado ao centro de massa e orientado segundo uma convenção interna: Frente, direita, trás (*Front, Right, Down, FRD*). Para assegurar a coerência entre perceção e controlo, é necessário realizar uma sequência de transformações entre referenciais distintos, que inclui tipicamente a passagem do referencial do marcador para o da câmara e desta para o do VANT.

Estas conversões são realizadas por meio de transformações rígidas que combinam rotações e translações fixas, podendo ser formalizadas através de matrizes homogéneas:

$${}^B\mathbf{T}_M = {}^B\mathbf{T}_C \cdot {}^C\mathbf{T}_M \quad (2.13)$$

onde: ${}^C\mathbf{T}_M \in \mathbb{SE}(3)$ representa a pose do marcador no referencial da câmara, ${}^B\mathbf{T}_C \in \mathbb{SE}(3)$ é a transformação conhecida entre a câmara e o corpo do VANT, e ${}^B\mathbf{T}_M \in \mathbb{SE}(3)$ é a pose final do marcador expressa no referencial do VANT. Cada matriz homogénea tem a forma:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (2.14)$$

em que $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ representa a rotação, e $\mathbf{t} \in \mathbb{R}^3$ a translação correspondente.

A compensação do desfasamento entre sensores é especialmente relevante em configurações onde a câmara se encontra montada fora do eixo central do VANT, como em estruturas frontais ou inclinadas. A omissão desta correção pode introduzir erros sistemáticos no seguimento da trajetória e comprometer a precisão da aterragem.

A integração da visão no ciclo de controlo é realizada por meio de um *pipeline* que inclui a aquisição e pré-processamento da imagem, a deteção de características

visuais, a estimativa da pose tridimensional do alvo e por fim, a sua utilização nos módulos de planeamento e controlo. A pose transformada é introduzida em leis de controlo baseadas no erro de posição e orientação, assegurando o alinhamento do VANT e a convergência para o ponto de aterragem.

Assim, a visão computacional constitui um elo essencial entre perceção e ação, permitindo uma resposta adaptativa do sistema em ambientes independentes de GNSS e aumentando a robustez da aterragem em cenários dinâmicos ou não estruturados (Guo et al. 2025).

Capítulo 3

Estado da Arte

Este capítulo apresenta uma análise crítica e sistemática das principais abordagens de aterragem automática de VANTs multirrotores, com o objetivo de fundamentar o desenvolvimento da solução proposta nesta dissertação. Dada a natureza multidisciplinar do problema, que envolve navegação, percepção visual, planeamento de trajetória e controlo, são revistos métodos oriundos de diferentes domínios do estado da arte, desde sistemas de posicionamento por satélite até algoritmos de visão computacional avançada.

Este capítulo organiza-se de forma progressiva. A Secção 3.1 apresenta um panorama geral das estratégias de aterragem, distinguindo métodos dependentes e independentes de GNSS. A SubSecção 3.1.1 analisa as abordagens baseadas em GNSS e respetivas limitações, enquanto a SubSecção 3.1.2 aborda os métodos adequados a ambientes de negação de GNSS, suportados em sensores visuais e inerciais. Focando-se na visão computacional, a Secção 3.2 trata o uso de marcadores visuais e técnicas de estimativa de pose, e a Secção 3.3 discute o planeamento de trajetória e as estratégias de controlo com realimentação visual. Por fim, a Secção 3.4 apresenta uma síntese crítica, destacando lacunas e justificando a necessidade de uma solução robusta baseada em visão embarcada.

3.1 Panorama Geral das Estratégias de Aterragem Automática

Os métodos atuais para aterragem automática de VANTs podem ser organizados em duas grandes categorias, métodos dependentes de GNSS, descritos na Subsecção 3.1.1 e métodos independentes de GNSS, descritos na Subsecção 3.1.2.

3.1.1 Métodos Dependentes de GNSS

Historicamente, os primeiros métodos automáticos de aterragem basearam-se em sistemas GNSS, em particular o GPS. Embora eficazes para navegação global e localização inicial, estes métodos revelam-se inadequados para aterragens de alta precisão, devido a erros típicos de 1 a 3 metros (Khazetdinov et al. 2021) e à vulnerabilidade em ambientes onde o GNSS é inexistente ou degradado, como interiores, zonas urbanas densas, vales estreitos, florestas (Valavanis e Vachtsevanos 2014), bem como ao empastelamento ou falsificação de sinal (Fahlstrom, Gleason e Sandraey 2022; Nuno Pessanha Santos 2020). Estas limitações motivaram o desenvolvimento de soluções híbridas que combinam GNSS com sistemas de navegação inercial (*Inertial Navigation System*, INS), baseados em medições de sensores inerciais como acelerómetros e giroscópios (Valavanis e Vachtsevanos 2014), visão computacional, infraestruturas externas e técnicas diferenciais de posicionamento, como no caso do RTK-GNSS, ilustrado na Figura 3.1.

Uma dessas variantes é o sistema de Posicionamento Cinemático em Tempo Real (*Real-time Kinematic Positioning*, RTK), que permite obter posicionamento absoluto com precisão centimétrica. O RTK-GNSS é uma técnica diferencial que recorre a correções transmitidas em tempo real por uma estação base (Morales et al. 2023). Em geral, requer fusão com outros sistemas para reduzir o erro e aumentar a robustez em ambientes dinâmicos, sendo, no entanto, altamente suscetível a falhas de sinal (Hassanalian e Abdelkefi 2017).

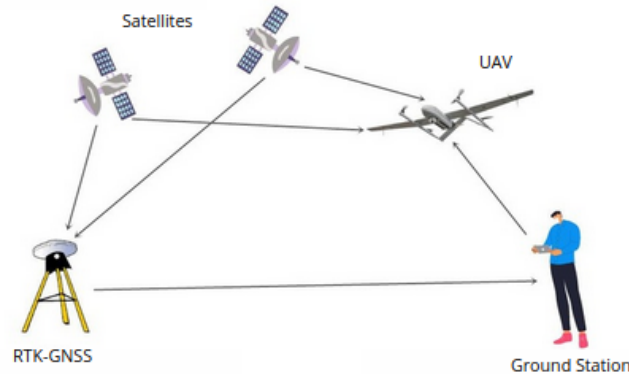


FIGURA 3.1: Esquema de comunicações com um VANT utilizando RTK-GNSS (Adaptado de JOUAV 2023).

3.1.2 Métodos Independentes de GNSS

As limitações associadas ao uso exclusivo de GNSS motivaram o desenvolvimento de métodos alternativos para guiar a aterragem de VANTs. Por outro lado,

os métodos independentes de GNSS recorrem a sensores embarcados no veículo, explorando informação proveniente de câmaras RGB ou térmicas, IMUs, LiDAR, entre outros, que são integrados através de algoritmos de fusão sensorial, estimativa de pose e controlo visual (Xin et al. 2022).

Xin et al. 2022 classificou a aterragem automática independente de GNSS, de acordo com a Figura 3.2, podendo ser classificada de acordo com três tipos de cenários operacionais: cenários estáticos, onde o alvo de aterragem é fixo (por exemplo, uma plataforma terrestre), cenários dinâmicos que envolvem alvos móveis, como navios ou veículos em movimento e cenários complexos relacionados a ambientes desorganizados, com obstáculos ou em condições adversas como vegetação densa ou áreas urbanas. Adicionalmente, cada cenário pode envolver dois tipos de alvos: alvos cooperativos, onde marcadores estão presentes no local de aterragem para facilitar a deteção e cenários naturais, onde o VANT deve identificar autonomamente uma zona de aterragem adequada com base nas características visuais existentes no ambiente.

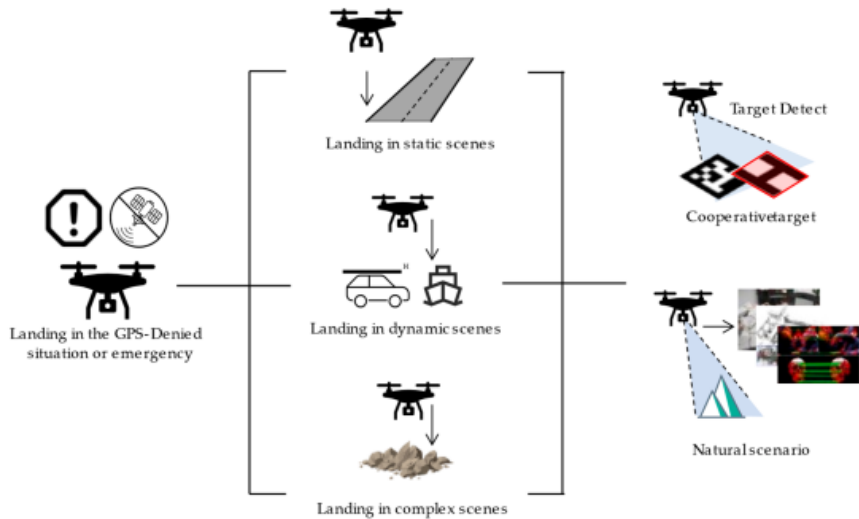


FIGURA 3.2: Classificação de aterragem automática independente de GNSS (Xin et al. 2022).

A partir desta classificação, é possível identificar um conjunto de técnicas e sensores que permitem realizar a aterragem de forma automática, mesmo na ausência de GNSS. As principais abordagens são descritas a seguir, agrupadas segundo o princípio tecnológico utilizado, os requisitos sensoriais e as suas limitações operacionais.

Métodos Baseados em Visão Computacional Embarcada

Os métodos baseados em visão computacional embarcada utilizam câmaras instaladas no próprio VANT, permitindo a detecção visual e a estimativa de pose de forma automática, sem depender de infraestruturas externas (Valavanis e Vachtsevanos 2014). Dentro desta categoria, uma abordagem comum é a utilização de uma câmara RGB com o complemento de marcadores, como os ArUco, April-Tag e marcadores em formato "H" (usados em heliportos) que funcionam como alvos visuais cooperativos colocados na zona de aterragem (Morales et al. 2023). Estes marcadores permitem uma estimativa precisa da pose relativa entre o VANT e o ponto de aterragem. Embora possam ser utilizados durante toda a aterragem, a sua eficácia aumenta nas fases em que o marcador ocupa maior área na imagem (Lee et al. 2020).

Outra técnica amplamente utilizada é o SLAM visual-inercial, que combina dados de câmaras com medições inerciais para construir um mapa do ambiente e estimar simultaneamente a pose do VANT (Qin, Li e Shen 2018). Este método é adequado para cenários não cooperativos e ambientes não estruturados, embora tenha maior exigência computacional (Dutrannois et al. 2022) e possa apresentar degradação de desempenho com variações rápidas de iluminação (Cao et al. 2024).

Complementarmente, têm sido desenvolvidos métodos baseados em CNNs, treinadas para identificar zonas de aterragem seguras em imagens RGB. Estas abordagens podem realizar segmentação semântica para distinguir áreas planas e livres de obstáculos, oferecendo boa generalização em ambientes naturais, embora dependam de conjuntos de dados anotados e apresentem elevada carga computacional (Kakaletsis et al. 2021).

Por fim, a detecção baseada em imagem térmica representa uma abordagem robusta para cenários com visibilidade reduzida, como ambientes noturnos, com fumo, poeiras ou nevoeiro. Este método recorre a câmaras que operam no espectro do infravermelho, permitindo identificar o alvo com base em contrastes de temperatura face ao fundo. Em certos casos, podem ser utilizados marcadores ativos, como o diodo emissor de luz (*light emitting diode*, LED), na gama dos infravermelhos, para facilitar a detecção. Apesar das suas vantagens em condições adversas, esta técnica apresenta limitações associadas à baixa resolução espacial (tipicamente inferior à das câmaras RGB) e da suscetibilidade a ruído térmico, além de implicar custos significativamente superiores face a sensores ópticos convencionais (Xuan-Mung et al. 2022).

Métodos Baseados em Visão Computacional em Infraestruturas Externas

Os métodos baseados em visão computacional recorrendo a infraestruturas externas utilizam sensores visuais instalados no ambiente envolvente, como plataformas terrestres, navios ou veículos móveis, em vez de dependerem exclusivamente de sensores embarcados no VANT (Nguyen et al. 2017). Um exemplo típico desta abordagem são os sistemas visuais no solo, nos quais câmaras fixas monitorizam a posição do VANT a partir de uma perspetiva externa (N. Pessanha Santos, Bernardino e V. Lobo 2020). Estas câmaras podem ser por exemplo monoculares ou estéreo, dependendo dos requisitos de profundidade e precisão. As soluções monoculares apresentam menor complexidade e custo, mas não fornecem informação de profundidade de forma direta (Nuno Pessanha Santos 2024). As configurações estéreo recorrem a duas câmaras para permitir triangulação e estimativa de distância, aumentando a fiabilidade em cenários tridimensionais. A informação captada é processada localmente e a pose estimada do VANT é então transmitida por rádio, permitindo que o veículo realize a aproximação com base em comandos externos (Damas, Santos e M. C. Vieira 2024; Nuno Pessanha Santos 2020).

Este tipo de sistema apresenta como principais vantagens a redução da carga computacional a bordo do VANT e uma maior estabilidade na deteção visual, especialmente em ambientes com condições visuais desafiantes. Esta estabilidade acrescida resulta do facto de o processamento ser realizado em unidades de computação no solo, que dispõem de maior capacidade de processamento e podem executar algoritmos mais complexos e robustos. Contudo, esta abordagem introduz uma dependência crítica de infraestruturas cooperativas no terreno, bem como da existência de canais de comunicação fiáveis e de baixa latência, capazes de transmitir dados sensoriais e comandos em tempo real.

3.2 Aterragem Automática Baseada em Marcadores Visuais

A aterragem automática baseada em marcadores visuais constitui uma abordagem consolidada na literatura para operações em ambientes de negação de GNSS, oferecendo elevada precisão na fase final da aproximação. A simplicidade de implementação, o baixo custo dos marcadores e a compatibilidade com câmaras de baixo custo tornam esta solução particularmente atrativa para aplicações em que a infraestrutura pode ser previamente preparada. Nesta secção, são analisadas as técnicas

associadas à estimativa de pose e os principais tipos de marcadores utilizados bem como as suas limitações em diferentes contextos operacionais.

3.2.1 Estimativa de Pose

A estimativa de pose visual é fundamental para a aterragem automática de VANTs, pois permite determinar a translação e orientação relativas do veículo em relação à zona de aterragem. No estado da arte, destacam-se diversas abordagens que combinam algoritmos geométricos, técnicas de filtragem estatística e fusão sensorial para assegurar precisão e robustez, mesmo em condições operacionais adversas.

No que concerne ao estado da arte alguma variantes do algoritmo do PnP têm sido desenvolvidas, destacando-se o *Efficient Perspective-n-Point* (EPnP), que oferece maior eficiência computacional e robustez ao ruído, permitindo a utilização de um número elevado de correspondências para melhorar a precisão da estimativa (Lu 2018). Outra abordagem amplamente utilizada é o PnP-RANSAC, que combina o algoritmo PnP com o método *Random Sample Consensus* (RANSAC) para reforçar a robustez da estimativa da pose. Esta técnica identifica e rejeita automaticamente correspondências inconsistentes (*outliers*), ao selecionar iterativamente subconjuntos de pontos válidos que maximizam a coerência geométrica entre as projeções 2D e os pontos 3D, resultando numa estimativa mais fiável mesmo em cenários com deteções ruidosas ou parcialmente incorretas. (Szeliski 2022).

Contudo, em aplicações reais, a deteção visual pode apresentar limitações significativas, como intermitência, oclusões parciais e ruído nas medições, comprometendo a estabilidade da estimativa da pose. Para mitigar estes efeitos, é comum complementar o algoritmo PnP e as suas variantes com técnicas de filtragem temporal, como por exemplo, o KF ou o EKF. Estes filtros permitem a fusão de medições provenientes de múltiplos sensores, como câmaras, IMUs e quando disponível, GNSS. Ao incorporar modelos dinâmicos do movimento do veículo, os filtros atenuam os efeitos do ruído nas estimativas, melhoram a continuidade da trajetória e permitem prever estados futuros. Além disso, mantêm uma estimativa coerente da pose mesmo em ocorrência de falha na deteção visual (Valavanis e Vachtsevanos 2014).

As soluções atualmente descritas na literatura reconhecem diversas limitações operacionais que afetam a fiabilidade dos sistemas de aterragem automática, nomeadamente a latência computacional, a baixa frequência de aquisição de imagem e a sincronização imperfeita entre os diferentes sensores. Estas limitações podem introduzir atrasos ou inconsistências nas estimativas de estado, comprometendo a estabilidade do processo de controlo (Xin et al. 2022). Por conseguinte, a seleção dos

algoritmos de percepção e estimativa deve procurar um equilíbrio rigoroso entre precisão, robustez e eficiência computacional, considerando simultaneamente as restrições impostas pelos sistemas embarcados e os requisitos de tempo real característicos de aplicações de navegação e controle automático.

3.2.2 Marcadores Visuais

A aterragem automática de VANTs multirrotores beneficia significativamente do uso de marcadores visuais, como os ArUco, amplamente adotados devido à sua robustez face a baixa resolução, oclusões, rotações e variações de iluminação, permitindo uma estimativa precisa da pose relativa entre o VANT e a plataforma de aterragem (Wubben et al. 2019). Foi criada uma extensão recente, o ArUco embutido (*embedded ArUco*, e-ArUco), que consiste na utilização de dois marcadores sobrepostos: um marcador externo e um marcador interno de dimensões diferentes. Este, assegura a detecção robusta numa ampla gama de distâncias, com um desempenho entre 20 cm a 30 m, superando os marcadores ArUco isolados (Kha-zetdinov et al. 2021). Em simulações, o e-ArUco alcançou uma precisão média de aterragem de 2.03 cm, com desvio padrão de 1.53 cm. Em testes de voo reais, sistemas baseados em ArUco reportaram erros de translação até 9.47 cm e um erro quadrático médio na orientação (*Root Mean Square Error*, RMSE), na orientação de aproximadamente 2.32° (Park et al. 2024). No entanto, noutro cenário, o uso de um marcador ArUco de 50×50 cm resultou num erro de aterragem de aproximadamente 80 cm, evidenciando a variabilidade do desempenho consoante a implementação e as condições operacionais (Mete 2018).

Delbene, Baglietto e Simetti 2022 utilizaram diversos AprilTags, de diversas dimensões a bordo de uma embarcação juntamente com sensores ultrassónicos a bordo do VANT, conseguindo aterrar o VANT automaticamente, com erros de estimativa de translação menores que 20 cm. No entanto, alguns artigos indiquem que as suas medições podem tornar-se menos fiáveis na fase final da aproximação, levando à necessidade de integrar sensores adicionais para obter erros menores (Morales et al. 2023). Por outro lado, marcadores no formato de “H”, comuns em contextos navais e em heliportos, têm mostrado bons resultados em ambientes visualmente congestionados (*cluttered*), sob oclusões parciais e variações de iluminação. Utilizando uma abordagem de *coarse-to-fine*, na qual a detecção evolui de características globais visíveis a maiores distâncias para detalhes mais finos em aproximações próximas e algoritmos baseados em distância ponto-a-linha, que estimam a pose minimizando a

distância entre os raios projetados pela câmara e as linhas 3D associadas ao marcador (Patruno et al. 2019). Estes marcadores alcançaram um RMSE médio de 0.014 m para a translação e 1.04° para a orientação em testes tanto em ambientes interiores com condições controladas como ambiente exterior com maior variabilidade de imprevisibilidade (Patruno et al. 2019).

A Tabela 3.1 apresenta uma comparação de desempenho entre os diferentes tipos de marcadores discutidos nesta secção, consolidando os principais resultados reportados na literatura. No geral, a capacidade destes marcadores para operar em condições adversas e a sua compatibilidade com sistemas de visão computacional consolidam-nos como uma solução de baixo custo e elevada precisão para a aterragem automática de VANTs.

TABELA 3.1: Comparação de desempenho entre os marcadores.

Descrição	ArUco	e-ArUco	AprilTag	“H”
Erro de Translação	80 cm	9.47 cm (max)	18,6 cm (média)	13.7 cm (RMSE)
Erro de Orientação	Não referido	2.3152° (RMSE)	Não referido	1.04° (média)
Notas	Utilizada uma câmara <i>Gimbal</i> ; Testes reais	Alinha posição e orientação sequencialmente; Testes reais	Limite de velocidade da embarcação de 0.35 m/s; Testes Reais	Menor precisão em <i>roll/pitch</i> ; Testes reais

3.3 Planeamento de Trajetória

O planeamento de trajetória para aterragem automática baseada em visão é um desafio amplamente abordado na literatura, com diversas estratégias que procuram equilibrar continuidade, robustez e viabilidade computacional. Entre os métodos mais utilizados destacam-se as curvas polinomiais, os perfis de movimento mínimo (Kakaletsis et al. 2021), o controlo preditivo por modelo (Valavanis e Vachtsevanos 2014) e, em algumas abordagens, estratégias de *visual servoing*, que consistem na utilização direta da informação visual, como a posição relativa do alvo na imagem, para gerar comandos de controlo em tempo real, ajustando continuamente a trajetória do VANT com base no erro obtido (Gautam et al. 2022).

Apesar da sua popularidade, as trajetórias polinomiais apresentam limitações, nomeadamente a sua estrutura estática e a reduzida capacidade de adaptação a alterações inesperadas no ambiente. Em conjunto, estes trabalhos demonstram que tais limitações podem ser mitigadas com técnicas complementares, como previsão de movimento, replaneamento por eventos ou escalonamento temporal.

Entre as abordagens baseadas em funções polinomiais, diversos trabalhos propõem soluções robustas e adaptativas para planeamento de trajetória em cenários visuais desafiantes. Pengyu Wang et al. 2022 utilizaram curvas B-spline como base para a fase de inicialização do seu método de planeamento local baseado em gradiente, criando trajetórias suaves, compatíveis com a dinâmica do VANT e livres de colisões em ambientes interiores e exteriores. A sua abordagem assume que o alvo está em movimento e por isso o VANT raramente executa a trajetória completa, em vez disso, novos planos de voo são gerados dinamicamente, adaptando-se ao contexto local.

Mellinger e Kumar 2011, por sua vez, utilizaram *minimum snap* para abordar o problema de geração e controlo de trajetórias em ambientes interiores complexos, demonstrando que a dinâmica de quadrotors é diferencialmente plana. Isto significa que a trajetória pode ser descrita diretamente em termos da posição (x, y, z) e do ângulo de *yaw*, a partir dos quais se obtêm todas as restantes variáveis de estado e controlos necessários. Com base nisso, recorrem a funções polinomiais segmentadas para minimizar o integral da norma do *snap*, definido como a quarta derivada da posição em relação ao tempo, isto é, a derivada temporal do *jerk*. Esta formulação assegura continuidade até ao nível do *jerk*, resultando em trajetórias ainda mais suaves e dinamicamente viáveis nos chamados *flat outputs*, isto é, as variáveis mínimas (no caso, (x, y, z) e *yaw*) que permitem reconstruir toda a dinâmica do sistema. A possibilidade de ajustar os tempos de segmento, escalar a trajetória para perfis mais agressivos ou seguros e impor restrições espaciais adicionais (como corredores de voo) confere a esta abordagem uma elevada flexibilidade operacional.

Uma estratégia distinta é apresentada por Acuna, D. Zhang e Willert 2018, que aplica *cubic-splines* para gerar trajetórias suaves num contexto sem GPS, em que o VANT acompanha um veículo terrestre. A trajetória é continuamente atualizada enquanto o alvo permanece visível e complementada com um algoritmo de previsão baseado em ajuste de curva, o que permite lidar com perdas momentâneas da deteção do marcador e assegurar a continuidade da missão. Esta integração entre interpolação e previsão mostra-se particularmente eficaz para aumentar a robustez do sistema face a limitações como intermitência na deteção visual e ruído.

Ainda no contexto de adaptação dinâmica, Kakaletsis et al. 2021 propôs uma arquitetura de segurança de voo com replaneamento em tempo real através de um algoritmo denominado *Simple Path Planner* (SPP), que permite recalcular rotas para zonas de aterragem alternativas em função de zonas de exclusão ou obstáculos

recém-detetados. Esta abordagem evidencia que, mesmo quando se utilizam planejadores polinomiais para gerar trajetórias suaves e otimizadas, torna-se necessário integrar mecanismos de reconfiguração capazes de lidar com alterações inesperadas no ambiente operacional.

Falanga et al. 2017 utilizou trajetórias com perfil de *minimum jerk* para guiar um VANT durante tarefas de seguimento visual de alvos em movimento. Este tipo de trajetória minimiza a terceira derivada da posição em relação ao tempo, o *jerk*, assegurando continuidade até à aceleração. Esta formulação resulta em movimentos mais suaves e estáveis, reduzindo variações bruscas de aceleração e, consequentemente, os esforços impostos sobre os atuadores. A escolha desta abordagem permitiu uma implementação computacional eficiente, o que viabilizou o replaneamento da trajetória em tempo real a uma frequência elevada de 50 Hz. Esta elevada taxa de atualização é particularmente relevante em cenários dinâmicos, pois permite ao sistema reagir rapidamente a alterações da pose do alvo, ajustando a trajetória com baixa latência e mantendo a estabilidade e precisão do seguimento. Além disso, foi também demonstrado que a verificação de viabilidade dinâmica dessas trajetórias pode ser realizada de forma eficiente, garantindo que os perfis de aceleração e velocidade respeitam os limites físicos do veículo, evitando saturação dos atuadores e assegurando seguimento estável mesmo em cenários dinâmicos com perturbações.

Em aplicações caracterizadas por restrições complexas e variáveis, o controle preditivo baseado em modelo (*Model Predictive Control*, MPC) tem sido amplamente adotado na criação e seguimento de trajetórias de VANTs. Esta abordagem permite a otimização em tempo real com base na minimização simultânea de múltiplos critérios de desempenho, como o erro de seguimento, o esforço de controle e a perda de informação sobre a localização da zona de aterragem. Uma das principais vantagens do MPC reside na sua capacidade de integrar de forma explícita as restrições físicas da plataforma, como os limites de velocidade, aceleração e campo de visão da câmara, permitindo gerar trajetórias viáveis do ponto de vista dinâmico e perceptivo (Mellinger e Kumar 2011). Além disso, a sua estrutura recursiva possibilita o replaneamento contínuo com base em informação proveniente de sensores atualizada, o que o torna particularmente adequado a ambientes dinâmicos e sujeitos a incertezas (Feng et al. 2018).

Neste contexto, Feng et al. 2018 propuseram um sistema de aterragem automática para VANTs em plataformas móveis, com particular foco em micro-VANTs sujeitos a perturbações externas e medições ruidosas. O sistema integra um KF

para estimativa da pose do alvo com base em marcadores visuais AprilTag detetados por uma câmara, bem como um controlador MPC linear concebido para realizar um seguimento da trajetória de aterragem de forma gradual e estável, minimizando manobras bruscas e respeitando as limitações dinâmicas do veículo. Este controlador otimiza, a cada iteração, a pose do VANT dentro de um horizonte de previsão curto, respeitando restrições como a velocidade máxima do veículo (18 m/s) e os ângulos máximos de inclinação (35° em *roll* e *pitch*). Para aumentar a robustez, foi ainda incorporado um controlador integral, responsável por compensar perturbações externas, como rajadas de vento. Um módulo de previsão da trajetória da plataforma móvel permitiu lidar com perdas temporárias de visibilidade, assegurando a continuidade da missão. Os resultados obtidos em simulações demonstraram uma precisão de aterragem inferior a 37 cm, mesmo em condições adversas, validando a eficácia e robustez da solução proposta.

Por fim, abordagens baseadas em *visual servoing*, ou mais recentemente em variantes adaptativas, utilizam diretamente a pose estimada para gerar comandos de controlo contínuos em malha fechada (Kakaletsis et al. 2021). Estas estratégias operam *frame a frame*, oferecendo elevada reatividade a movimentos do alvo e robustez perante variações na perceção visual. No entanto, a continuidade dos movimentos pode ser comprometida, uma vez que a resposta do sistema depende fortemente do ajuste dos ganhos do controlador. Além disso, estas técnicas são sensíveis ao ruído na imagem e podem apresentar oscilações ou divergências em condições de deteção instável, limitando a precisão da aterragem.

Um exemplo representativo desta abordagem é o trabalho de Serra et al. 2016, que propuseram uma estratégia de controlo em visão, denominada *Image-Based Visual Servoing* (IBVS) para a aterragem autónoma de um quadrotor sobre alvos móveis. A lei de controlo foi formulada diretamente no espaço da imagem bidimensional, regulando a velocidade da câmara com base no erro em pixels entre a posição atual e a desejada dos pontos de referência visuais. O sistema demonstrou robustez face a perturbações externas, como rajadas de vento, e a erros de medição, assegurando a convergência do veículo para o alvo em movimento. A estimativa da posição e da velocidade da plataforma de aterragem foi realizada com base no cálculo do centróide das características visuais e na análise do fluxo ótico translacional, eliminando a necessidade de sensores externos adicionais. Este trabalho evidencia o potencial do controlo por *visual servoing* para operar em ambientes dinâmicos e com perceção limitada, embora apresente limitações, nomeadamente a sensibilidade ao ruído visual e a dificuldade em garantir continuidade suave da trajetória.

3.4 Síntese Crítica

Com base na revisão realizada, apresenta-se na Tabela 3.2 uma síntese comparativa dos principais trabalhos relevantes no domínio da aterragem automática de VANTs em plataformas móveis. Esta análise contempla múltiplas dimensões técnicas, incluindo o tipo de marcador fiducial utilizado, a estratégia de localização ou detecção visual adotada, os métodos de planeamento e controlo aplicados, o ambiente de validação (simulado e/ou real) e os principais resultados e vantagens reportadas.

TABELA 3.2: Síntese das principais abordagens de aterragem automática baseadas em visão computacional

Estudo/Método	Marcador Fiducial	Estratégia de Localização / Detecção	Estratégia de Planeamento / Controlo	Ambiente de Teste	Resultados Chave / Vantagem Principal
Khazetdinov et al. 2021	Marcadores e-ArUco	Detecção de e-ArUco (exterior de longe, interior de perto) com algoritmos ArUco padrão.	Alinhamento direto com base na pose estimada (sem planeamento explícito).	Simulado (Gazebo).	Detecção robusta do marcador em amplas gamas de distância (0,2 m a 30 m), garantindo visibilidade durante toda a aterragem. Precisão média de 2,03 cm.
Mellinger e Kumar 2011	N/A (usa sistema de captura de movimento Vicon para localização).	N/A.	Geração de trajetórias ótimas (<i>minimum snap</i>).	Real (Quadrotor AscTec, Vicon).	Gera trajetórias suaves e ótimas para manobras agressivas, com um erro máximo de posição de 8 cm.
Pengyu Wang et al. 2022	Marcadores ArUco.	Detecção baseada em ArUco para estimativa rápida de pose relativa (PNP).	Planeador de movimento local baseado em gradiente com <i>B-Spline</i> .	Simulado (Pro-metheus/Gazebo) e Real (Indoor/Outdoor).	Aterragem precisa e de baixo custo em plataformas móveis (até 3,24 km/h) com capacidade de evitar obstáculos durante a aterragem.
Morales et al. 2023	Padrão personalizado de marcadores ArUco.	Estimativa de pose relativa (posição e rumo) do VANT em relação ao UGV baseada no ArUco (PNP).	Seguimento com perfis de velocidade trapezoidais em linha reta (sem planeamento otimizado).	Simulado (ROS/Gazebo/SITL) e Real (Outdoor).	A detecção de marcadores foi possível entre uma altura máxima de 4 metros e uma altura mínima de 0.23 centímetros. Demonstrações bem-sucedidas de seguimento e aterragem em ambientes simulados e reais.
Acuna, D. Zhang e Willert 2018	Marcadores ArUco.	Detecção baseada em visão com previsão de movimento do UGV para lidar com a perda temporária do campo de visão (FOV).	Máquina de estados para gestão de modos de operação. Controlo em cascata de 6-DOF recorrendo a <i>cubic-splines</i> .	Simulado (ROS/Gazebo)	Lida com cenários onde o alvo sai temporariamente do FOV da câmara, prevendo seu movimento. Taxa de sucesso de 100% a baixas velocidades.
Feng et al. 2018	Marcadores AprilTags.	Detecção baseada em AprilTag para estimativa de pose relativa (PNP).	Controlo Preditivo por Modelo (MPC) com controlo integral para robustez.	Simulado (DJI HITL Simulation).	Aterragem automática em plataformas móveis de alta velocidade (até 12 m/s) sob medições ruidosas e com perturbações do vento.
Falanga et al. 2017	Marcador personalizado (cruz preta dentro de círculo)	Estimativa contínua da pose do alvo com odometria visual-inercial monocular e modelo dinâmico (velocidade constante).	Planeamento de trajetória ótimo com perfil de <i>minimum jerk</i> , resolvido em malha fechada.	Simulado (RotorS/Gazebo) e Real.	Sistema modular e leve, robusto a perda temporária de detecção. Planeamento suave, computacionalmente eficiente, replaneamento eficiente a 50 Hz e validado em missões com alvos móveis até 4.2 m/s.

Os trabalhos foram selecionados com base na sua representatividade metodológica e diversidade de cenários de aplicação, abrangendo tanto abordagens baseadas em controlo clássico como estratégias mais avançadas de planeamento e controlo preditivo. Esta comparação visa evidenciar as forças e limitações das soluções existentes, bem como identificar lacunas metodológicas e operacionais que justificam o desenvolvimento da abordagem proposta nesta dissertação.

A análise detalhada dos principais trabalhos na área de aterragem automática de VANTs evidencia avanços significativos no uso de marcadores visuais, planeamento polinomial de trajetórias e controlo embarcado em ambientes de negação de GNSS. Estes sistemas recorrem frequentemente a sensores de baixo custo, como câmaras e IMUs, integrados com arquiteturas modulares e estratégias adaptativas como previsão de movimento ou replaneamento dinâmico. Em particular, muitos estudos demonstram capacidades robustas de seguimento e aterragem em plataformas móveis, combinando algoritmos eficientes com execução embarcada em hardware limitado.

Apesar destes progressos, persistem lacunas relevantes na literatura. A maioria dos trabalhos abordados é conduzida com VANTs de pequena escala, tipicamente com menos de 1.5 kg. As implicações de inércia aumentada, atuadores com limitações mais restritivas e dinâmica mais lenta, comuns em VANTs de maiores dimensões usados em operações de longa duração ou transporte de carga, não são analisadas de forma sistemática no contexto da aterragem baseada em visão. Embora existam referências que demonstram a capacidade de controlo reduzida em quadrotors maiores (Omar et al. 2023) a forma como estas características influenciam a deteção visual e o desempenho do controlo não é devidamente explorada.

Por fim, as questões associadas à comunicação e ao fluxo de vídeo embarcado, como perdas de pacotes, compressão, latência ou falhas temporárias de deteção, também não são ignoradas pela literatura. Em vários estudos, problemas com ligações sem infraestrutura fixa e limites na taxa de transmissão foram identificados como entraves à operação contínua. A latência é apontada como fator crítico em aterragem de alta velocidade, e a robustez da comunicação, incluindo requisitos de disponibilidade e integridade é considerada essencial para garantir segurança. Estratégias de compensação, como a previsão do movimento através de por exemplo, modelos dinâmicos, são frequentemente aplicadas para mitigar falhas temporárias de deteção. Assim, longe de assumir um canal visual estável e ideal, grande parte da literatura reconhece explicitamente os efeitos de ruído, erros de estimativa e limitações computacionais.

Com base nas limitações identificadas na literatura, o presente estudo propõe uma abordagem integrada para a aterragem automática de VANTs multirrotores em plataformas móveis, com especial enfoque em veículos de média dimensão e em cenários operacionais realistas. A solução desenvolvida recorre exclusivamente a sensores embarcados de baixo custo, nomeadamente uma câmara monocular e integra algoritmos de deteção de marcadores, estimativa de pose e planeamento de trajetórias com perfil de *minimum jerk* e *minimum snap*, combinados com um EKF com o objetivo de reforçar a fiabilidade face ao ruído nas estimativas de pose obtidas. De forma a colmatar as lacunas associadas à escala do veículo, o sistema é aplicado a um VANT com características dinâmicas representativas de missões reais no domínio marítimo. Neste tipo de operações, tipicamente realizadas a partir de navios ou plataformas móveis, a utilização de VANTs de pequena escala é limitada pela autonomia reduzida, suscetibilidade a perturbações atmosféricas, acabando a não serem habitualmente empregues. Assim, a adoção de um VANT de média dimensão permite uma análise mais realista dos desafios envolvidos, nomeadamente no seguimento visual e na execução precisa da trajetória de aproximação. Adicionalmente, são consideradas as restrições impostas pelo canal de comunicação utilizando entre o VANT e a estação em terra, incluindo latência, compressão e perdas de pacotes, avaliando a viabilidade do processamento em tempo real. Esta configuração permite validar a autonomia do sistema em condições próximas do contexto operacional real, demonstrando a sua capacidade de adaptação a ambientes com comunicação limitada e estimativa de pose ruidosa.

Capítulo 4

Formulação do problema

Devido à elevada complexidade do problema, o capítulo que se segue visa delimitar de forma clara o contexto do problema, bem como os seus principais desafios técnicos. Inicialmente, na Secção 4.1 é caracterizado o cenário operacional, com ênfase nas particularidades do ambiente marítimo e nas exigências impostas à aterragem autónoma em movimento. De seguida, na Secção 4.2 identificam-se os problemas críticos que motivam esta investigação, como as limitações na disponibilidade de sensores, a instabilidade da plataforma e as dificuldades inerentes à estimativa contínua e precisa da pose. É ainda definido, na Secção 4.3 o objetivo técnico do sistema a desenvolver, tendo em consideração a viabilidade prática e os requisitos funcionais impostos pela aplicação. Na Secção 4.4 são abordadas as restrições impostas pelo meio e pela arquitetura do sistema. Por fim, na Secção 4.5 são explicados os requisitos do sistema.

4.1 Definição do Cenário Operacional

O presente sistema foi concebido para ser implementado em VANTs operados pela Marinha Portuguesa, com o objetivo de realizar a fase final da missão, uma aterragem automática em contexto naval. Após a execução de tarefas primárias, como vigilância ou recolha de informação, o VANT deve regressar e aterrar de forma precisa e segura. Neste contexto, o sistema desenvolvido visa responder a esse requisito operacional, assegurando a fiabilidade da missão até ao seu término.

Neste cenário, o algoritmo assume controlo total da manobra a partir do momento em que o marcador é detetado, como ilustrado na Figura 4.1 baseando-se exclusivamente em visão computacional para estimar a pose relativa do VANT face à plataforma de aterragem. Esta abordagem foi escolhida para eliminar a dependência de GNSS, frequentemente comprometido por empastelamento em contexto militar. A missão ocorre durante o período diurno, com vento moderado e variações

de iluminação natural, representado assim um ambiente realista para validação do sistema.

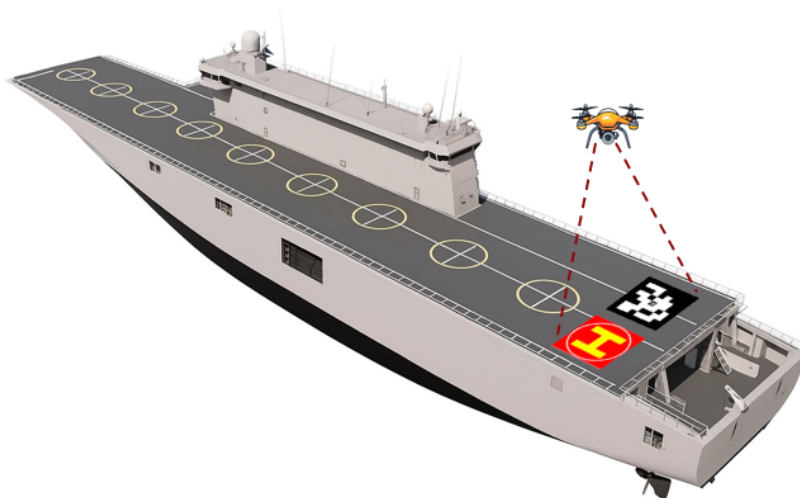


FIGURA 4.1: Ilustração de aterragem num navio com marcadores.

4.2 Problemas Identificados

A implementação de um sistema de aterragem automática baseado exclusivamente em visão computacional, no contexto naval definido, levanta diversos desafios técnicos e operacionais.

Em primeiro lugar, a exclusão de GNSS, motivada pelo risco de empastelamento, elimina uma importante fonte de dados de posicionamento. O sistema passa assim a depender exclusivamente da estimativa de pose relativa obtida através da deteção visual de marcadores. Esta dependência impõe fortes exigências à fiabilidade e continuidade da deteção visual, que pode ser afetada por variações de iluminação provocadas pelas estruturas do navio, pelas condições atmosféricas e pela operação apenas em período diurno.

Outro desafio crítico é a instabilidade da plataforma de aterragem. O movimento contínuo do navio, causado pela ondulação introduz variações na pose do alvo de aterragem. Isto dificulta o seguimento visual contínuo do marcador e exige mecanismos de compensação por parte do sistema de controlo. Pelo facto da deteção do marcador se poder tornar intermitente ou mesmo inexistente a dado momento, há o comprometimento da continuidade da estimativa de pose do VANT.

Adicionalmente, a operação em contexto marítimo implica uma área de aterragem significativamente reduzida, quando comparada com ambientes em terra.

A precisão exigida é, por isso, consideravelmente superior, especialmente tendo em conta os efeitos de vento lateral e perturbações aerodinâmicas durante a fase de aproximação final. Em conjunto, estes fatores colocam elevados requisitos na estabilidade da estimativa de pose e na capacidade de resposta do controlador.

4.3 Objetivo Técnico do Sistema

O objetivo técnico deste sistema consiste na realização de uma aterragem automática precisa de um VANT multirrotor sobre uma plataforma móvel, utilizando exclusivamente visão computacional embarcada no VANT. O VANT considerado (Figura 4.2) é um quadrotor com uma distância entre centros das hélices opostas de 1,30 metros e um peso aproximado de 8 kg, desenvolvido Célula de Inovação e Experimentação Operacional de Sistemas Não Tripulados (CEOV). Este, está equipado com os sensores típicos de uma plataforma deste tipo (GPS, IMU, altímetro, barómetro, entre outros), embora a lógica de aterragem adotada nesta tese se baseie exclusivamente na informação visual fornecida por uma câmara RGB, montada a bordo da aeronave e orientada na direção do solo.



FIGURA 4.2: VANT Quadcopter da CEOV utilizado nos testes experimentais.

As imagens captadas são processadas em tempo real para detetar marcadores e estimar a pose relativa entre o VANT e a plataforma de aterragem. Esta estimativa é complementada por um EKF, que reduz o impacto do ruído e assegura continuidade em caso de falhas momentâneas de deteção.

A partir da pose estimada, é gerada uma trajetória de aproximação, composta por uma sequência de posições intermédias que conduzem o VANT até ao alvo. Em cada iteração, o sistema calcula vetores de deslocamento relativos ($\Delta x, \Delta y, \Delta z$),

expressos no referencial do corpo, os quais são transmitidos ao controlador de voo Pixhawk via protocolo *Micro Air Vehicle Link* (MAVLink), em relação ao referencial do VANT. Compete ao *firmware* interno do Pixhawk interpretar esses comandos de alto nível e executar as manobras correspondentes, com base no controlador de baixo nível Proporcional–Integral–Derivativo (PID). Este tipo de controlador ajusta continuamente os sinais de controlo com base no erro atual (componente proporcional), na acumulação histórica do erro (componente integral) e na sua taxa de variação (componente derivativa), assegurando uma resposta estável e adaptada às perturbações do sistema.

Atualmente, todo o processamento necessário à perceção visual, estimativa da pose e geração de trajetória é realizado na estação terrestre, sendo prevista a sua migração futura para um computador embarcado, de forma a viabilizar a operação autónoma em ambientes com comunicação limitada.

4.4 Restrições do Sistema

A deteção visual do marcador depende fortemente das condições de iluminação. Sombras, reflexos ou baixa luminosidade podem comprometer a qualidade da imagem, dificultando a identificação do marcador e afetando a estimativa de pose. Adicionalmente, o campo de visão da câmara limita a área útil de deteção, sendo o sistema sensível a variações de atitude do VANT ou aproximações fora do eixo visual. Oclusões temporárias e perdas momentâneas de deteção são também esperadas, exigindo mecanismos de filtragem que assegurem continuidade. Embora atualmente o processamento seja realizado na estação terrestre, a futura execução a bordo imporá restrições adicionais à complexidade dos algoritmos, dado o carácter limitado dos recursos computacionais disponíveis

Por fim, a precisão do sistema depende da calibração rigorosa da câmara. Pequenos erros nos parâmetros intrínsecos e coeficientes de distorção podem comprometer a estimativa da pose, sobretudo em aproximações com ângulos elevados ou em zonas periféricas do campo visual.

4.5 Requisitos Funcionais

A Figura 4.3 apresenta o diagrama de requisitos funcionais do sistema de aterragem automática, organizada em blocos modulares que representam as principais etapas do processo. O sistema inicia-se com a conversão de referenciais, de

seguida a percepção visual, responsável pela detecção do marcador e estimativa da pose relativa. Segue-se a aplicação de um EKF, que suaviza a estimativa antes do planeamento da trajetória. A trajetória gerada é convertida em comandos de alto nível, enviados ao autopiloto Pixhawk, que executa as manobras necessárias através do seu sistema interno de controlo. Esta estrutura modular permite uma visão clara do funcionamento do sistema e facilita a sua descrição detalhada seguidamente.



FIGURA 4.3: Diagrama de requisitos funcionais do sistema.

4.5.1 Conversão de referenciais

Para que a pose estimada a partir da percepção visual possa ser efetivamente utilizada pelo sistema de navegação e controlo do VANT, é necessário assegurar a conversão entre diferentes sistemas de coordenadas. A deteção visual baseada em marcadores fornece a pose relativa no referencial da câmara. No entanto, os módulos de controlo embarcados, como o autopiloto Pixhawk com *firmware* ArduPilot, operam num referencial distinto, FRD. Torna-se, por isso, necessário relacionar a pose estimada entre referenciais, garantindo a relação correta entre os dados visuais e os comandos de controlo.

Em contexto de simulação, acrescenta-se ainda uma camada de complexidade, dado que o ambiente virtual Gazebo utiliza a convenção (*East-North-Up* (ENU)). Esta discrepância impõe a aplicação de conversões adicionais, de modo a permitir a comparação direta entre as estimativas visuais e o *ground truth* fornecido pelo simulador.

A correta transformação entre os diferentes referenciais da câmara, do simulador, do autopiloto e do corpo do VANT, ilustrados na Figura 5.5, constitui, assim, um requisito funcional indispensável para assegurar consistência espacial.

4.5.2 Percepção visual

Um requisito funcional fundamental do sistema desenvolvido consiste na aquisição e processamento de dados visuais em tempo real, com o objetivo de detectar a presença do marcador e estimar a pose relativa. Esta estimativa é obtida utilizando imagens captadas por uma câmara RGB embarcada no VANT, orientada para o solo. A detecção do marcador, que se encontra localizado na plataforma de aterragem, deve ser robusta a variações de iluminação, distância e ângulo de visualização, assegurando a fiabilidade do sistema em cenários realistas.

4.5.3 Filtro de temporal

A integração de um mecanismo de filtragem temporal constitui outro requisito funcional essencial do sistema proposto, com vista a garantir a robustez e a estabilidade das estimativas de pose ao longo do tempo. Dado que a estimativa visual obtida a partir da detecção de marcadores pode estar sujeita a ruído, oclusões parciais e variações abruptas devido à dinâmica do voo ou limitações da percepção, é necessário garantir um método que permita suavizar essas medições ruidosas e minimizar o erro obtido.

Deste modo, o sistema deve integrar um modelo dinâmico simples do movimento do VANT, que permita suavizar medições ruidosas e manter estimativas consistentes mesmo durante curtas interrupções na detecção visual. Este requisito é essencial para assegurar a continuidade do planeamento e controlo da trajetória de aterragem.

4.5.4 Planeamento da trajetória

O sistema desenvolvido requer a capacidade de gerar trajetórias estáveis e dinamicamente viáveis, que permitam ao VANT aproximar-se do alvo de forma precisa e segura. Para esse efeito, é necessário estimar a pose a uma taxa compatível com o controlador de bordo. Neste contexto, é requisito funcional a utilização de modelos de planeamento baseados na minimização de critérios de estabilidade, que assegurem perfis de movimento adequados à execução estável da manobra de aterragem.

4.5.5 Geração de comandos de alto nível

Após a definição da trajetória de aterragem, o sistema deve ser capaz de convertê-la em comandos de alto nível, compatíveis com o autopiloto embarcado.

Este requisito funcional assegura a ligação entre o planeamento e a execução, permitindo que o VANT siga a trajetória desejada com base em referências de pose.

4.5.6 Autopiloto

O autopiloto Pixhawk com o *firmware* ArduPilot, é responsável pela execução dos comandos recebidos, assegurando a estabilização da atitude, o controlo de altitude e a realização das manobras. Este requisito funcional implica garantir a compatibilidade entre os comandos gerados pelo gerador de trajetórias e a estrutura de controlo embarcada, bem como a correta configuração dos modos de voo, parâmetros de controlo e interfaces de comunicação. O autopiloto gere os atuadores em tempo real, assegurando o seguimento das referências e a estabilidade do voo até à aterragem.

4.5.7 Controlo e execução em tempo real

O funcionamento do sistema exige que todas as componentes, desde a perceção visual, estimativa de pose, filtragem temporal, planeamento e controlo, operem de forma integrada e em tempo real. É, por isso, requisito funcional que o sistema seja capaz de processar a informação adquirida e gerar comandos com latência suficientemente baixa para garantir a estabilidade e precisão da aterragem.

A execução em tempo real é particularmente crítica nas fases finais de aproximação, onde pequenas variações na posição e orientação do VANT exigem atualizações rápidas e fiáveis. Adicionalmente, para efeitos de avaliação experimental do sistema, torna-se necessário assegurar a sincronização temporal entre as imagens captadas pela câmara e a telemetria proveniente do autopiloto e os dados de referência do simulador, de modo a permitir a obtenção de *ground truth* e a análise rigorosa do desempenho do algoritmo.

Capítulo 5

Metodologia

Este capítulo descreve a metodologia adotada para o desenvolvimento do sistema de aterragem automática baseado em visão computacional, conforme ilustrado na Figura 5.1. A metodologia é apresentada segundo os quatro blocos funcionais que compõem o sistema: percepção visual, na Secção 5.1, planeamento de trajetória, na Secção 5.2, controlo de alto nível, na Secção 5.3 e autopiloto, na Secção 5.4.

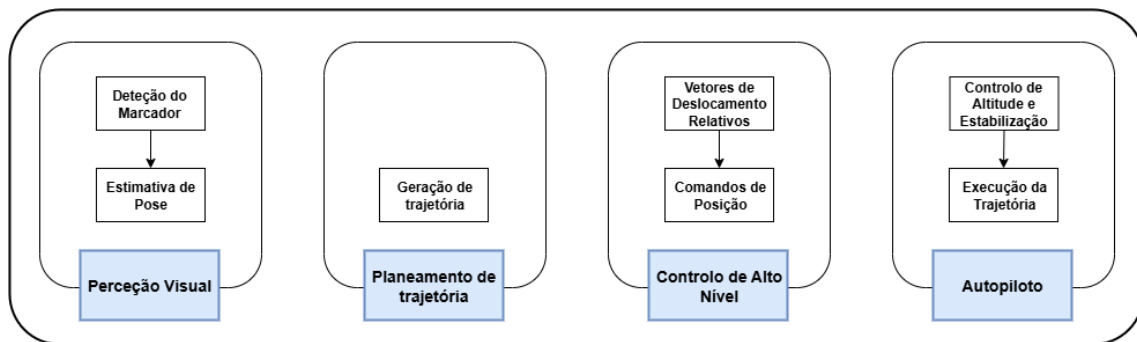


FIGURA 5.1: Arquitetura funcional do sistema de aterragem automática.

5.1 Percepção Visual

O módulo de percepção visual tem como principal objetivo estimar a pose relativa entre o VANT e o alvo de aterragem a partir da imagem captada por uma câmara RGB embarcada. Para este fim, é utilizada uma câmara SIYI R1M, instalada na estrutura do VANT, com um desfasamento de 11.5 cm abaixo e 25 cm à frente do recetor GNSS. A câmara transmite vídeo comprimido em *Motion JPEG* (MJPEG), com baixa latência, utilizando uma ligação Ethernet. O vídeo é transmitido em tempo real para a estação terrestre utilizando um rádio dedicado, com resolução até 1080p a 30 fps, possibilitando processamento na estação de controlo em terra.

A estimativa de pose é obtida através da detecção de marcadores, sendo posteriormente filtrada por um EKF para maior fiabilidade e continuidade. Esta estimativa de pose constitui a base para o planeamento da trajetória e para o controlo em tempo real do sistema.

5.1.1 Calibração da Câmara

A calibração da câmara constitui um pré-requisito fundamental para assegurar a precisão da estimativa de pose visual, uma vez que permite corrigir distorções óticas e estabelecer a correspondência entre o mundo real e o plano da imagem. Sem esta etapa, a estimativa da pose a partir de marcadores sofreria erros sistemáticos, sobretudo nas regiões periféricas da imagem. Estes parâmetros são usados para corrigir as imagens antes da extração de marcadores, eliminando as distorções introduzidas pela câmara.

O processo de calibração seguiu o método de Zhang (Zhengyou Zhang 2000), suportado pela biblioteca OpenCV. Foi utilizado um padrão planar de xadrez com 8 colunas por 5 linhas de cantos internos, cada quadrado medindo 2.54 cm. Foram adquiridas 54 imagens em diferentes orientações, distâncias e ângulos de visão, garantindo diversidade suficiente para uma estimativa robusta dos parâmetros. Para cada imagem, o algoritmo detetou automaticamente os cantos do tabuleiro e ajustou o modelo de câmara, minimizando o erro entre as projeções teóricas e as observadas.

O processo resultou na seguinte matriz intrínseca e coeficientes de distorção:

$$\mathbf{K} = \begin{bmatrix} 696.42 & 0 & 678.95 \\ 0 & 691.63 & 353.23 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$\mathbf{d} = [0.145 \quad -0.266 \quad -0.00089 \quad 0.00326 \quad 0.152] \quad (5.2)$$

O erro médio de reprojeção obtido foi de 0.057 píxeis, valor bastante reduzido e indicativo da elevada consistência entre o modelo geométrico e as observações. A Figura 5.2 mostra a diferença entre a imagem original e a imagem corrigida, sendo possível verificar que, particularmente nos cantos da imagem, a compensação da distorção reduz as deformações introduzidas pela lente.



FIGURA 5.2: Comparação entre a imagem original (esquerda) e a imagem após correção da distorção (direita).

5.1.2 Detecção de Marcadores

O sistema foi desenvolvido para a detecção de dois tipos de marcadores: ArUco e marcador em "H". Cada tipo apresenta características distintas que podem ser adequadas a diferentes cenários operacionais. Os marcadores ArUco baseiam-se em padrões binários codificados, oferecendo detecção robusta através de algoritmos de reconhecimento de padrões. Para o marcador em "H" é implementada uma detecção baseada em segmentação por cor (vermelho e amarelo) e validação de simetria da geometria.

Marcadores ArUco

A pose do marcador é estimada a partir das suas coordenadas 2D na imagem, recorrendo ao modelo de câmara *pinhole* e à resolução do problema PnP. Dada a correspondência entre os pontos 3D conhecidos do marcador (no referencial do próprio marcador) e os pontos 2D observados na imagem, é calculada a matriz de rotação $\mathbf{R}$ e o vetor de translação $\mathbf{t}$, fornecendo assim a pose 6D do marcador relativamente ao referencial da câmara (função `cv2.aruco.estimate PoseSingleMarkers`).

Para aumentar a precisão da detecção, foi aplicado um método de refinamento sub-pixel dos cantos, que itera sobre a vizinhança de cada ponto para ajustar a sua posição e melhorar a consistência geométrica da estimativa. Foi considerada uma janela de 7×7 píxeis e até 50 iterações de otimização, de forma a maximizar a estabilidade das estimativas, sobretudo em cenários de maior distância de detecção (utilizando a opção `CORNER_REFINE_APRILTAG` na função `cv2.aruco.ArucoDetector`). Adicionalmente, definiu-se uma taxa mínima e máxima do perímetro do marcador, respetivamente de 2% e 400% relativamente ao tamanho da imagem, restringindo a detecção a marcadores de grande dimensão e reduzindo a ocorrência de falsos positivos.

Na presente dissertação foram utilizados dois dicionários ArUco distintos. Um dicionário constituído por marcadores binários de dimensão 5×5 bits e 50 identificadores únicos, foi aplicado em marcadores físicos de 13.2 cm, adequados a testes em curta distância (DICT_5X5_50). E um outro dicionário que contém 1000 padrões de 7×7 bits, foi utilizado em marcadores de 70 cm, mais apropriados para detecção a maiores distâncias devido à sua maior área projetada na imagem (DICT_7X7_1000). Foi selecionado aleatoriamente o identificador 23, assegurando consistência nos ensaios. A Figura 5.3 exemplifica a detecção em condições reais, incluindo a projeção dos eixos de coordenadas estimados.



FIGURA 5.3: Detecções de marcador ArUco ID 23 de 13.2 cm.

Marcador H

Foi utilizado um marcador em forma de "H", com 80 cm de lado, composto por um fundo circular vermelho e uma estrutura interna com formato de H amarela. A detecção foi implementada através de segmentação por cor no espaço *Hue*, *Saturation*, *Value* (HSV), selecionado pela sua robustez sob diferentes condições de iluminação (Sayeed e Ayesha 2020). Foram definidos intervalos baseados em testes experimentais para a representação das cores vermelho e amarelo. No caso do vermelho, devido à sua representação em ambos os extremos do espaço de cores, foram usados dois intervalos: $H \in [0, 10] \cup [160, 180]$, $S \in [150, 255]$ e $V \in [100, 255]$. Para o amarelo, definiu-se $H \in [20, 35]$, $S \in [150, 255]$ e $V \in [100, 255]$. Os valores de saturação (S) e valor (V) foram intencionalmente limitados a níveis elevados, de forma a restringir a detecção a cores intensas e reduzir falsos positivos causados por sombras ou baixa iluminação. O fundo vermelho foi validado por ajuste de elipse, aceitando apenas formas aproximadamente circulares (razão de eixos > 0.8). Da estrutura amarela foram definidos seis pontos característicos (quatro cantos externos e dois pontos centrais das hastes horizontais), associados a coordenadas tridimensionais conhecidas. A Figura 5.4 exemplifica a detecção em simulação, incluindo a projeção dos eixos de coordenadas estimados.

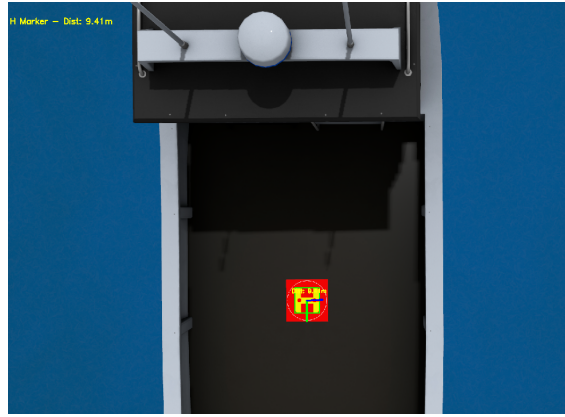


FIGURA 5.4: Detecção de marcador H de 80 cm de lado.

5.1.3 Transformações de Referenciais

A estimativa inicial é obtida relativamente ao referencial do marcador ArUco, definido com origem no centro do marcador, eixo X orientado para a direita, Y para baixo e Z perpendicular ao plano do marcador (para fora). Para integração no sistema de controlo, esta informação deve ser convertida sequencialmente para o referencial do corpo do VANT (FRD) e, por fim, para o sistema global *North-East-Down* (NED) utilizado pelo ArduPilot. O processo de transformação de referenciais é realizado em três etapas principais:

1. **Rotação:** aplicação da matriz de rotação que converte o sistema de coordenadas da câmara para o sistema de coordenadas do corpo do VANT;
2. **Translação:** correção do desfasamento físico entre a câmara e o centro do VANT, correspondente a um deslocamento de 11.5 cm abaixo e 25 cm à frente do recetor GNSS;
3. **Conversão para NED:** transformação final para o referencial compatível com os comandos de alto nível enviados para o piloto automático (utilizando a opção `BODY_OFFSET_NED`).

Formalmente, e de acordo com o ilustrado na Figura 5.5, a transformação é implementada através da seguinte relação:

$$\mathbf{T}_{\text{body}}^{\text{marker}} = \mathbf{T}_{\text{body}}^{\text{cam}} \mathbf{T}_{\text{cam}}^{\text{marker}} = \begin{bmatrix} \mathbf{R}_{\text{body}}^{\text{cam}} & \mathbf{t}_{\text{body}}^{\text{cam}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\text{cam}}^{\text{marker}} & \mathbf{t}_{\text{cam}}^{\text{marker}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (5.3)$$

onde $\mathbf{R}_{\text{body}}^{\text{cam}}$ representa a matriz de rotação que converte do referencial da câmara para o referencial do corpo do VANT e é definida por:

$$\mathbf{R}_{\text{body}}^{\text{cam}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

e $\mathbf{t}_{\text{body}}^{\text{cam}}$ representa o vetor de translação entre os dois referenciais de acordo com:

$$\mathbf{t}_{\text{body}}^{\text{cam}} = \begin{bmatrix} 0.25 \\ 0 \\ -0.115 \end{bmatrix} \quad (5.5)$$

sendo que esta rotação adicional é necessária para acomodar a montagem física da câmara (orientada para baixo) e garantir consistência com os comandos no referencial do piloto automático (NED).

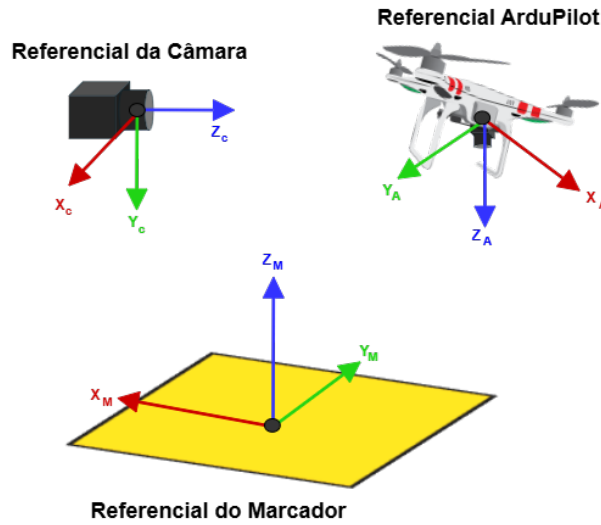


FIGURA 5.5: Ilustração dos principais referenciais utilizados.

5.1.4 Filtragem Temporal

A estimativa direta de pose revelou-se ruidosa, com variações abruptas entre frames, sobretudo em mudanças de direção e em movimentos rápidos do VANT. Para mitigar estas flutuações, foi implementado um EKF, que funde observações sucessivas e gera uma estimativa de pose mais estável e contínua.

O EKF foi selecionado devido à não linearidade existente na dinâmica do sistema $f(\cdot)$ e na relação entre o estado e as observações $h(\cdot)$ (Welch, Bishop et al. 1995). Isto ocorre pois a projeção de pontos tridimensionais no plano da imagem constitui uma função racional da profundidade, as transformações entre referenciais incluem relações trigonométricas, e o algoritmo PnP resolve um problema de mínimos quadrados não linear através de otimização iterativa. Estas características inviabilizam a utilização de um filtro de Kalman linear clássico.

O vetor de estado é definido por:

$$\mathbf{x} = [x \ y \ z \ v_x \ v_y \ v_z \ \phi \ \theta \ \psi \ \omega_\phi \ \omega_\theta \ \omega_\psi]^\top \quad (5.6)$$

onde (x, y, z) representam a posição do VANT no referencial *body*, (v_x, v_y, v_z) as respectivas velocidades lineares, (ϕ, θ, ψ) os ângulos de roll, pitch e yaw, e $(\omega_\phi, \omega_\theta, \omega_\psi)$ as velocidades angulares.

A evolução temporal do estado assume um modelo de movimento constante, descrito por:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \Delta t) + \mathbf{w}_k, \quad (5.7)$$

com $f(\cdot)$ representando as equações de propagação temporal:

$$\begin{cases} p_k = p_{k-1} + v_{k-1} \Delta t \\ \theta_k = \theta_{k-1} + \omega_{k-1} \Delta t \\ v_k = v_{k-1} \\ \omega_k = \omega_{k-1} \end{cases}$$

e $\mathbf{w}_k$ modelando o ruído de processo gaussiano.

As observações correspondem à pose estimada pelo sistema de visão, composta por posição e orientação $[x, y, z, \phi, \theta, \psi]$:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (5.8)$$

onde $h(\cdot)$ representa a relação não linear entre o estado do VANT e as observações obtidas pelo PnP, e $\mathbf{v}_k$ é o ruído de medição. Na implementação, a Jacobiana de medição assume observabilidade direta dos seis primeiros estados ($H = [I_6 \ 0]$), com normalização dos ângulos para o intervalo $[-\pi, \pi]$ antes da atualização.

Os ruídos de processo $\mathbf{w}_k$ e de medição $\mathbf{v}_k$ são modelados como ruído gaussiano, com matrizes de covariância $\mathbf{Q}$ e $\mathbf{R}$, definidas diagonais e constantes ao longo

da execução:

$$\mathbf{Q} = \text{diag}\left(\underbrace{10^{-5}, 10^{-6}, 10^{-5}}_{\text{pos } [x,y,z] \text{ (m}^2\text{)}}, \underbrace{10^{-6}, 10^{-7}, 10^{-6}}_{\text{ori } [\phi,\theta,\psi] \text{ (rad}^2\text{)}}, \underbrace{10^{-2}, 10^{-2}, 10^{-2}}_{\text{vel. lin. (m}^2\text{/s}^2\text{)}}, \underbrace{10^{-2}, 10^{-2}, 10^{-2}}_{\text{vel. ang. (rad}^2\text{/s}^2\text{)}}\right),$$

$$\mathbf{R} = \text{diag}\left(\underbrace{0.02, 0.002, 0.01}_{\text{pos } [x,y,z] \text{ (m}^2\text{)}}, \underbrace{0.005, 0.005, 2 \times 10^{-5}}_{\text{ori } [\phi,\theta,\psi] \text{ (rad}^2\text{)}}\right).$$

Estes valores foram ajustados empiricamente para equilibrar suavização temporal e responsividade. Em termos de desvio-padrão, correspondem aproximadamente a $[0.141, 0.045, 0.100]$ m para a posição e a $[4.1^\circ, 4.1^\circ, 0.26^\circ]$ para a orientação.

5.1.5 Refinamento da Detecção

No caso do marcador ArUco de 70 cm, a robustez da detecção baseou-se exclusivamente nos parâmetros internos do detetor do OpenCV já descritos na anteriormente, incluindo o refinamento sub-pixel dos cantos (`CORNER_REFINE_APRILTAG`), os limites de perímetro mínimo e máximo do marcador (`minMarkerPerimeterRate` = 0.02, `maxMarkerPerimeterRate` = 4.0) e os restantes parâmetros de segmentação e correção de perspectiva. Estes filtros asseguram a rejeição automática de detecções incoerentes, pelo que não foi implementada nenhuma métrica de qualidade adicional nem cálculo de pontuação associado.

Já para o marcador em forma de H, foi implementado um processo de avaliação que atribui a cada detecção uma pontuação de qualidade Q , combinando quatro fatores: (i) área relativa, (ii) distribuição dos pontos, (iii) simetria e (iv) proximidade às bordas. A equação global é dada por:

$$Q = w_{\text{área}} Q_{\text{área}} + w_{\text{distribuição}} Q_{\text{distribuição}} + w_{\text{simetria}} Q_{\text{simetria}} + w_{\text{bordas}} Q_{\text{bordas}}, \quad (5.9)$$

com pesos $(w_{\text{área}}, w_{\text{distribuição}}, w_{\text{simetria}}, w_{\text{bordas}}) = (0.3, 0.3, 0.2, 0.2)$. Cada métrica individual é normalizada em $[0, 1]$, sendo definidas como:

- **Área relativa:** $Q_{\text{área}} = \min\left(\frac{A_{\text{marcador}}}{A_{\text{imagem}}} \times 20, 1.0\right)$, com A_{marcador} a área projetada e A_{imagem} a área da imagem.
- **Distribuição dos pontos:** $Q_{\text{distribuição}} = 1 - \min\left(\frac{\sigma_d}{100}, 1.0\right)$, com σ_d o desvio padrão das distâncias dos pontos ao centro geométrico.

- **Simetria:** $Q_{\text{simetria}} = 1 - \min(E_{\text{simetria}}, 1.0)$, onde E_{simetria} representa o erro médio de alinhamento entre os lados esquerdo e direito.
- **Proximidade às bordas:** $Q_{\text{bordas}} = \prod f(p)$, sendo $f(p) = 0.9$ se um ponto p estiver a menos de 50 píxeis de uma borda, e 1.0 caso contrário.

O candidato com maior Q é selecionado para a estimativa de pose, descartando observações inconsistentes. Adicionalmente, é exigida uma validação temporal: 10 deteções consecutivas dentro de uma tolerância de 10 cm (filtro de média móvel) antes de iniciar a sequência de aterragem, garantindo robustez na estimativa e evitando respostas instáveis.

5.2 Planeamento de Trajetória

O módulo de planeamento de trajetória é responsável por criar sequências de posições dinamicamente viáveis que conduzem o VANT até uma aterragem suave e precisa, com base na estimativa de pose relativa ao marcador. Neste trabalho, foram implementados e comparados dois perfis distintos: um baseado em *minimum snap* e outro em *minimum jerk*. Ambas as implementações partilham a mesma arquitetura: a trajetória é calculada com base no erro de pose relativamente ao alvo. Sempre que uma trajetória termina e o objetivo ainda não foi atingido, é criada uma nova a partir da pose corrente até ao alvo fixo. Esta estratégia assegura adaptação contínua às variações de pose, mantendo estabilidade e baixo custo computacional. O objetivo foi avaliar o desempenho de cada abordagem em termos de suavidade, tempo de execução e precisão de aterragem. Ambas as estratégias geram trajetórias ponto-a-ponto, compostas por um único segmento polinomial entre a posição atual do VANT e o ponto-alvo da aterragem.

5.2.1 Trajetória com *minimum snap*

A trajetória *minimum snap* é definida como:

$$p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 + a_6t^6 + a_7t^7, \quad (5.10)$$

com restrições impostas em $t = 0$ e $t = T$, incluindo posição p , velocidade $\dot{p}$, aceleração $\ddot{p}$ e *jerk* $\dddot{p}$. Estas condições originam um sistema linear resolvido para os coeficientes $a_0 \dots a_7$, resultando numa trajetória contínua até à terceira derivada, compatível com as limitações dinâmicas de um quadcóptero. As trajetórias são

criadas de forma independente para cada eixo $x(t), y(t), z(t)$ e discretizadas num intervalo fixo de $\Delta t = 0.02 \text{ s}$ (50 Hz). A duração total é adaptativa:

$$T = \max\left(\frac{d}{v}, 0.5\right), \quad (5.11)$$

onde d representa a distância euclidiana até ao alvo (considerado $\mathbf{p}_{\text{alvo}} = [0, 0, 0]$) e v a velocidade de aproximação desejada. O limite inferior de 0.5 s previne trajetórias demasiado curtas e manobras agressivas.

Cada ponto é transmitido como comando MAVLink de posição no referencial NED localizado no VANT, que expressa deslocamentos relativos à posição atual do VANT.

5.2.2 Trajetória com *minimum jerk*

A expressão normalizada do *minimum jerk* utilizada é:

$$x(t) = x_0 + (x_f - x_0) \left(35\tau^4 - 84\tau^5 + 70\tau^6 - 20\tau^7 \right), \quad (5.12)$$

onde $\tau = \frac{t}{T}$ com $0 \leq t \leq T$.

Tal como no *minimum snap*, as trajetórias são criadas separadamente para cada eixo e discretizadas num intervalo fixo de $\Delta t = 0.02 \text{ s}$ (50 Hz). A duração total segue igualmente a formulação adaptativa descrita na Equação 5.11

A principal diferença em relação ao *minimum snap* é que esta formulação fornece uma solução fechada, isto é, uma expressão polinomial com coeficientes já conhecidos de forma universal, dispensando a resolução de sistemas lineares a cada nova trajetória e reduzindo assim a complexidade computacional. Embora não minimize explicitamente o *snap*, gera perfis de aceleração suaves e movimentos naturais, revelando-se adequados em cenários com ruído de perceção ou restrições de processamento.

5.3 Controlo de Alto Nível

O módulo de controlo de alto nível tem como função converter a trajetória contínua criada pelo planeador num conjunto de comandos discretos compatíveis com o autopiloto. Estes comandos são transmitidos ciclicamente via protocolo MAVLink, utilizando a mensagem SET_POSITION_TARGET_LOCAL_NED. Este tipo de

mensagem contempla múltiplos campos de comando, desde posição, velocidade, aceleração, orientação em *yaw* e respetiva velocidade angular (*yaw rate*). Na presente dissertação, foram utilizados apenas os campos de posição, enviados no formato `BODY_OFFSET_NED`, garantindo que os vetores de deslocamento são interpretados relativamente à orientação atual da aeronave e simplificando o seguimento da trajetória.

A trajetória é discretizada em pontos de referência tridimensionais com intervalo temporal fixo de $\Delta t = 0.02\text{ s}$ (50 Hz). Em cada iteração do ciclo de controlo, o próximo ponto é convertido num vetor de erro de posição representado por:

$$\Delta \mathbf{p}_t = \mathbf{p}_t^{\text{ref}} - \mathbf{p}_t^{\text{current}}, \quad (5.13)$$

onde $\mathbf{p}_t^{\text{ref}}$ corresponde ao ponto seguinte da trajetória e $\mathbf{p}_t^{\text{current}}$ à posição atual estimada do VANT. Onde o vetor é expresso inicialmente em ENU e convertido para NED.

O módulo opera em modo *open-loop*, enviando um comando por ciclo sem correção entre pontos consecutivos. Para reduzir o impacto de erros acumulados ou distúrbios externos, sempre que uma trajetória é teoricamente concluída mas o alvo permanece por atingir, é criada uma nova trajetória a partir da pose atual. Este mecanismo assegura convergência contínua, mantendo suavidade temporal e eficiência computacional.

5.3.1 Interface de Controlo Manual

Para apoiar a validação experimental e garantir segurança durante os ensaios, foi ainda implementada uma interface de controlo manual por teclado integrada numa interface gráfica OpenCV. Esta funcionalidade permite iniciar ou interromper a sequência de aterragem de forma imediata, assegurando replicabilidade dos testes e a possibilidade de abortar a descida em situações imprevistas. A descrição detalhada desta interface encontra-se no Apêndice E.

5.4 Autopiloto

O controlo de baixo nível do VANT é assegurado por um autopiloto Pixhawk a executar o *firmware* ArduPilot. Este módulo constitui a camada final da arquitetura de aterragem, sendo responsável por interpretar os comandos de posição recebidos via MAVLink e convertê-los em sinais de atuação para os motores, assegurando a estabilização e a execução física da trajetória. O autopiloto gere todas as dinâmicas de baixo nível, incluindo a atuação nos motores, a estabilização da orientação e o controlo de atitude, impondo ainda limites predefinidos de velocidade, inclinação e aceleração.

Adicionalmente, a arquitetura beneficia dos mecanismos de segurança nativos do ArduPilot, como limites de inclinação, deteção de perda de sinal, *timeout* de comandos externos e capacidade de *failsafe*. O sistema foi configurado para operar com receção contínua de comandos a 50 Hz, interrompendo automaticamente a aterragem caso a deteção do marcador seja perdida por mais de 2 segundos.

No contexto da simulação, o ambiente ArduPilot SITL replica o comportamento do autopiloto real, recebendo exatamente as mesmas mensagens MAVLink. Esta consistência garante que os testes em simulação sejam diretamente transferíveis para voo real, reduzindo a necessidade de ajustes no controlador. A separação clara entre perceção, planeamento e controlo interno confere modularidade ao sistema, permitindo evoluções futuras na componente computacional externa sem impacto direto no autopiloto.

Capítulo 6

Resultados experimentais

Este capítulo apresenta os resultados experimentais obtidos no decorrer do desenvolvimento e validação do sistema proposto. A análise está dividida em três secções principais, de forma a refletir as diferentes fases do processo de validação.

O capítulo encontra-se dividido em três secções em que, na Secção 6.1 são explorados os resultados obtidos em ambiente simulado e na Secção 6.2 são explorados os resultados obtidos em ambiente real. Nestas duas secções é feita uma comparação entre *ground truth*, a avaliação da estimativa de pose através da deteção de marcadores visuais e os primeiros testes de trajetória de aterragem. Por fim, apresenta-se uma discussão global na Secção 6.3 com as principais observações, limitações e implicações para o desenvolvimento futuro do sistema. Complementarmente, no Apêndice G são descritas as métricas estatísticas utilizadas para avaliar a qualidade da estimativa de pose, como RMSE, MAE, erro médio, desvio padrão e erro máximo. Os ambientes de teste são ilustrados na Figura 6.1

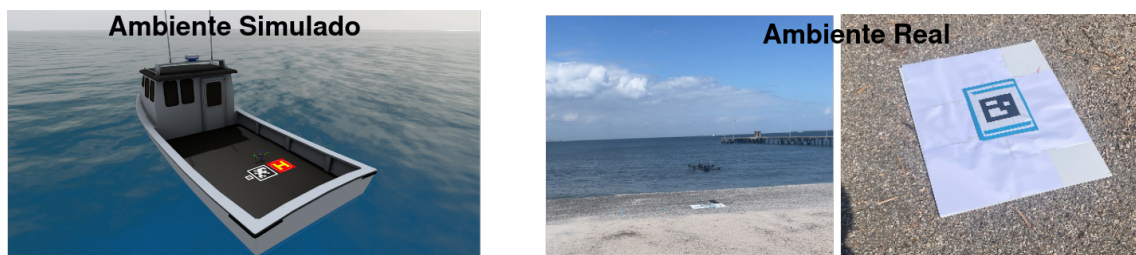


FIGURA 6.1: Ambiente simulado (esquerda) e ambiente real (direita).

6.1 Resultados em Ambiente Simulado

Esta secção apresenta os resultados obtidos através dos testes realizados no ambiente simulado. São analisadas comparações entre dados de *ground truth* fornecidos pelo ArduPilot e pelo Gazebo, seguidas da avaliação da estimativa de

pose a partir dos marcadores. Por fim, é examinada a trajetória de aterragem gerada com base nessa estimativa.

6.1.1 *Ground Truth*

Para avaliar a fiabilidade da telemetria do ArduPilot como referência em voo real, analisou-se o erro de posição 3D (RMSE) face ao *ground truth* geométrico do simulador Gazebo, considerado isento de erro. De acordo com o ilustrado no Apêndice B, apresentam-se os resultados para diferentes altitudes.

O erro oscila entre 0.21 m e 0.36 m, mantendo-se relativamente estável até cerca de 8 m (valores entre 0.26 m e 0.29 m). A partir dos 9 m, observa-se um aumento para 0.357 m, associado à diminuição da resolução espacial do marcador na imagem, resultante da menor densidade de píxeis disponíveis para a sua deteção.

Estes resultados são relevantes para os testes em voo real, nos quais a referência do ArduPilot será a única disponível. Assim, desvios inferiores a 30 cm poderão refletir limitações da própria estimativa de pose e não necessariamente erros na estimativa de pose visual.

6.1.2 Estimativa de Pose

Nesta parte são apresentados os resultados da estimativa de pose usando diferentes marcadores visuais. Para cada caso, analisa-se o erro de posição e orientação, bem como a frequência de deteção, comparando os valores obtidos diretamente (RAW) com os filtrados pelo EKF. O objetivo é perceber até que altura cada marcador garante uma deteção fiável e de que forma o filtro temporal contribui para tornar as estimativas mais suaves e consistentes para aplicações de controlo em tempo real.

Alcance e zona de segurança com ArUco 13.2 cm

No Apêndice C é apresentado o erro de pose (RMSE) e a frequência de deteção do marcador ArUco de 13.2 cm, em função da altura. Entre os 2 m e os 6 m, o sistema apresenta bom desempenho, com RMSE inferior a 8 cm e frequências acima de 9 Hz, sendo este o intervalo seguro para controlo visual. O erro mínimo ocorre a 5.06 m, com apenas 1 cm após filtragem EKF. Acima dos 6 m, o erro cresce progressivamente e, a partir dos 9 m, a frequência de deteção cai para valores insuficientes (1.08 Hz), inviabilizando a operação. O EKF não altera significativamente o RMSE, mas contribui para maior suavidade na trajetória estimada.

Alcance e zona de segurança com ArUco 70 cm

No Apêndice C apresentam-se o RMSE da pose e a frequência média de detecção do marcador ArUco de 70 cm em função da altura. Até cerca de 6 m, o sistema mantém erros inferiores a 20 cm e frequências acima de 9 Hz, garantindo desempenho robusto para controlo visual. Entre os 7 m e os 16 m, o erro cresce gradualmente até cerca de 27 cm, mas com frequências ainda superiores a 7 Hz, permitindo operação viável. Acima dos 17 m, verifica-se uma degradação acentuada, com erros superiores a 30 cm e frequências abaixo de 5 Hz, o que inviabiliza o uso em tempo real. O EKF traz melhorias marginais (2–5 cm) em distâncias intermédias, mas com impacto reduzido a maiores alturas. Conclui-se que a operação fiável se estende até cerca de 16 m, sendo prudente restringir aplicações automáticas a valores inferiores a 20 m.

Alcance e zona de segurança com marcador H 80 cm

No Apêndice C apresentam-se o RMSE da pose e a frequência média de detecção do marcador H de 80 cm em função da altura. Até cerca de 6 m, o desempenho é robusto, com erros inferiores a 26 cm. Em alturas próximas de 3 m a 6 m, destacam-se resultados particularmente favoráveis, com RMSE abaixo de 12 cm e frequências superiores a 6.7 Hz. No entanto, acima dos 7 m, a frequência de detecção cai abruptamente para menos de 1 Hz e o EKF torna-se instável, com erros que podem ultrapassar os 50 cm. A partir dos 10 m, a detecção torna-se residual, com erros superiores a 70 cm, inviabilizando a utilização prática. Assim, a operação fiável com este marcador encontra-se limitada a alturas até cerca de 6 m.

Trajetória com formato infinito com ArUco de 13.2 cm

No teste da trajetória em formato infinito a 3 m, a aplicação do EKF não reduziu o erro posicional (RMSE), registando-se até um ligeiro aumento face aos dados RAW. Contudo, verificou-se uma melhoria clara na estimativa da orientação, com reduções de erro até 1.01° nos ângulos *roll*, *pitch* e *yaw*, conforme apresentado no Apêndice C.

Trajetória com formato infinito com ArUco de 70 cm

No teste da trajetória em formato infinito a 4 m com o marcador de 70 cm, a aplicação do EKF não trouxe ganhos de precisão, verificando-se até um ligeiro aumento do RMSE tanto na posição como na orientação, em especial nos ângulos *roll* e *pitch*, conforme ilustrado no Apêndice C.

Trajetória com formato infinito com marcador H

No teste da trajetória em formato infinito a 4 m com o marcador H de 80 cm, o EKF não reduziu o erro de posição, registrando-se apenas um ligeiro aumento marginal (<1.5 cm) face aos dados RAW. Em contrapartida, verificou-se uma melhoria consistente na orientação, com reduções de aproximadamente 4.6° em *roll* e *pitch*, conforme apresentado no Apêndice C.

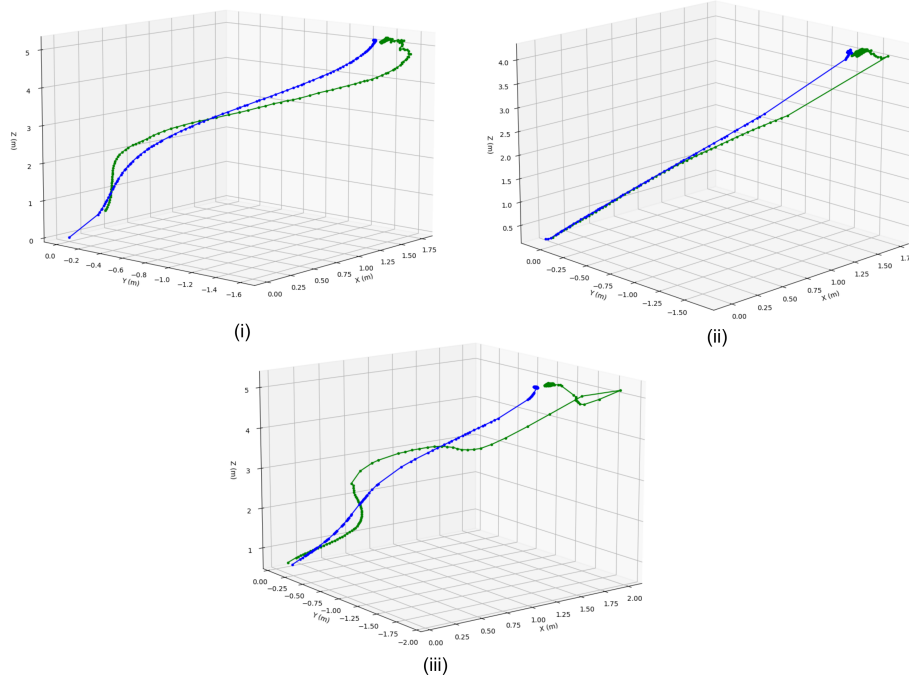
6.1.3 Trajetória de Aterragem com *minimum jerk*

Foram realizados aproximadamente 50 testes em ambiente simulado com o objetivo de validar a precisão do sistema de aterragem autónoma com base em trajetórias do tipo *minimum jerk*. Para tal, foram utilizados três tipos de marcadores visuais colocados no solo: dois marcadores fiduciais ArUco com dimensões laterais de 13,2 cm e 70 cm, respetivamente, e um marcador em formato de “H” com 80 cm de lado. Estes elementos serviram como referências visuais para o sistema de navegação baseado em visão computacional. Os ensaios abrangeram diversas configurações de posição inicial relativa entre o VANT e o marcador, permitindo avaliar a robustez do algoritmo face a variações na geometria inicial da aproximação. Em todas as execuções, o VANT iniciou a manobra com orientação nula nos ângulos de *roll*, *pitch* e *yaw*, garantindo condições iniciais consistentes para análise comparativa.

A precisão da aterragem gerada para uma velocidade nominal de 2 m/s apresentada na Tabela 6.1 e Figura 6.2, foi quantificada com base na norma euclidiana do vetor de posição final (Apêndice G), assumindo que o objetivo seria o centro do marcador, posicionado na origem. Os resultados indicaram um erro de posição máximo de aproximadamente 17,5 cm (com o ArUco de 70 cm) e um erro mínimo de 4 cm (com o marcador H de 80 cm). A média dos desvios posicionais situou-se abaixo dos 15 cm, validando a aplicabilidade do método para cenários de aterragem de precisão em ambientes controlados. No que respeita à orientação, o maior desvio observado foi de apenas $0,022^\circ$, registado com o ArUco de 13,2 cm, valor negligenciável para a fase terminal da manobra. O tempo médio de aterragem foi de aproximadamente 12 segundos, enquanto o valor mínimo registado foi de 11 segundos, observado nos testes realizados com o marcador em formato de H.

TABELA 6.1: Precisão de aterragem com método *minimum jerk* à velocidade máxima de 1.1 m/s.

Marcador	Posição inicial (m)	Posição final (m)	Orientação inicial ($^{\circ}$)	Orientação final ($^{\circ}$)
ArUco 13.2 cm	(1.5, 1.5, 5)	(0.14, 0.04, 0)	(0, 0, 0)	(0, 0.02, 0.01)
ArUco 70 cm	(1.5, 1.5, 5)	(0.16, 0.07, 0)	(0, 0, 0)	(0, 0, 0)
H 80 cm	(1.5, 1.5, 5)	(0, 0.04, 0)	(0, 0, 0)	(0, 0, 0)

FIGURA 6.2: Trajetórias estimadas (verde) com o método *minimum jerk* vs reais (azul) com ArUco 13.2 cm (i), ArUco 70 cm (ii) e H (iii).

6.1.4 Trajetória de Aterragem com *minimum snap*

Foram realizados aproximadamente 60 testes em ambiente simulado com o objetivo de avaliar a precisão do sistema de aterragem autónoma utilizando trajetórias do tipo *minimum snap*. As condições de ensaio foram mantidas idênticas às utilizadas nos testes com o perfil *minimum jerk*, de modo a permitir uma comparação direta entre os dois métodos. A trajetória foi gerada para uma velocidade nominal de 2 m/s, e foram novamente utilizados três tipos de marcadores visuais: dois marcadores fiduciais ArUco com dimensões laterais de 13,2 cm e 70 cm, respetivamente, e um marcador em formato de “H” com 80 cm de lado. Os ensaios abrangeram diferentes posições iniciais relativas entre o VANT e o marcador, avaliando a consistência do algoritmo face a variações na geometria inicial da aproximação. Em todas

as execuções, o VANT iniciou a manobra com orientação nula nos ângulos de *roll*, *pitch* e *yaw*, assegurando condições de partida equivalentes.

Os resultados, apresentados na Tabela 6.2 e Figura ??, indicaram um erro na posição máximo de aproximadamente 7,8 cm, correspondente ao teste com o ArUco de 13,2 cm, e um erro mínimo de 5,0 cm com o marcador em H. Estes valores foram calculados com base na norma euclidiana da posição final estimada em relação ao centro do marcador, assumido como origem. Todos os valores se mantiveram abaixo dos 8 cm, indicando uma melhoria significativa face ao método anterior. No que respeita à orientação, o desvio máximo foi de apenas 0,02°, registado com o ArUco de 13,2 cm, o que reforça a estabilidade da atitude durante a manobra. O tempo médio de aterragem foi de aproximadamente 8 segundos, enquanto o valor mínimo registado foi de 7 segundos, observado nos testes realizados com o marcador em formato de H.

TABELA 6.2: Precisão de aterragem com método *minimum snap* à velocidade máxima de 1.5 m/s.

Marcador	Posição inicial (m)	Posição final (m)	Orientação inicial (°)	Orientação final (°)
ArUco 13.2 cm	(1.5, 1.5, 5)	(0.05, 0.06, 0)	(0, 0, 0)	(0, 0, 0.02)
ArUco 70 cm	(1.5, 1.5, 5)	(0.06, 0.03, 0)	(0, 0, 0)	(0, 0, 0)
H 80 cm	(1.5, 1.5, 5)	(0.03, 0.04, 0)	(0, 0, 0)	(0, 0, 0)

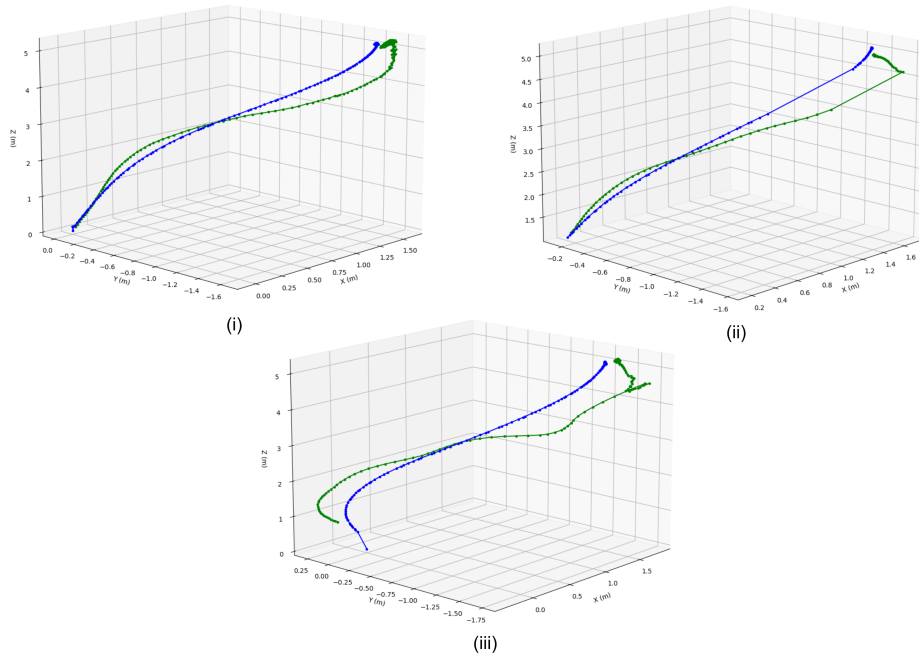


FIGURA 6.3: Trajetórias estimadas (verde) com o método *minimum snap* vs reais (azul) com ArUco 13.2 cm (esquerda), ArUco 70 cm (centro) e H (direita).

6.2 Resultados em Ambiente Real

Nesta secção apresentam-se os resultados obtidos em ambiente real. Começa-se pela análise dos testes ao sistema RTK-GNSS, seguindo-se a avaliação da estimativa de pose utilizando o marcador ArUco de 13.2cm, tanto em bancada como em voo e, por fim, os resultados da aterragem com recurso à trajetória *minimum snap*.

6.2.1 RTK-GNSS

Foram realizados diversos testes para avaliar a precisão do sistema RTK (Here+) da CEOV, com o objetivo de o utilizar como *ground truth*. Conforme apresentado no Apêndice A, após um *survey-in* de 3 horas e 30 minutos, foi obtida uma precisão de apenas 0.37 m. Para efeito de comparação, testou-se igualmente o RTK (Here+) do Instituto Superior Técnico de Lisboa, tendo sido alcançada a mesma precisão em apenas 32 minutos.

6.2.2 Estimativa de Pose

Testes em bancada

Inicialmente estimou-se a pose em bancada como representado na Figura 6.4 e na Tabela 6.3, na qual se recorreu a uma fita métrica para medir distâncias entre o marcador e a câmara. Observa-se que os erros de posição em X , Y e Z se mantêm reduzidos (1–2 cm) para todas as distâncias referidas, demonstrando boa consistência na estimativa translacional. Em contraste, os erros de orientação apresentam maior variação, com o *roll* e o *pitch* a aumentarem progressivamente à medida que a distância cresce de 0.5 m para 1.5 m. Já o *yaw* manteve-se estável, com variações inferiores a 1.5° , o que indica maior robustez neste eixo.



FIGURA 6.4: Testes de bancada à estimativa de pose.

TABELA 6.3: Erros de pose para diferentes distâncias ao marcador.

Distância (m)	X (cm)	Y (cm)	Z (cm)	Roll (°)	Pitch (°)	Yaw (°)
0.5	1	1	1	-3.4	0.9	-1.1
1.0	1	1	1	-5.6	2.4	-1.1
1.5	1	1	2	-8.2	3.1	-0.9

Testes em voo real

Relativamente aos testes de estimativa de pose em voo real, o cálculo do erro foi realizado tendo como referência a telemetria fornecida pelo ArduPilot, a qual apresenta uma incerteza típica de 1 a 3 m devido à utilização de GNSS convencional. Conforme apresentado na Tabela 6.4 e na Figura 6.5, os resultados de posição indicam diferenças entre os eixos analisados. Nos eixos X e Y , os valores de RMSE situaram-se entre 0.23 m e 0.33 m, com erros máximos de 0.52 m. Já no eixo Z verificou-se um erro mais acentuado, com RMSE de 0.70 m e máximo de 1.32 m.

TABELA 6.4: Erros de posição (Ardupilot vs EKF).

Eixo	RMSE EKF (m)	Máximo EKF (m)
X	0.33	0.52
Y	0.23	0.52
Z	0.70	1.32

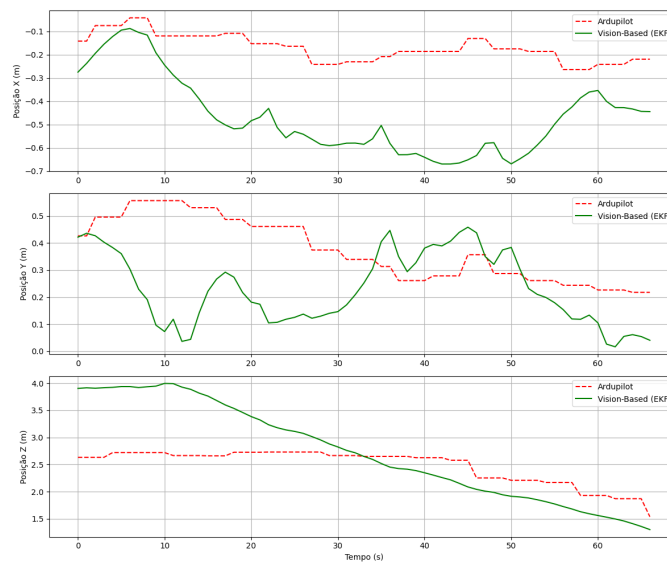


FIGURA 6.5: Erro de posição estimada (Ardupilot vs EKF).

Relativamente à orientação, a Tabela 6.5 e a Figura 6.6 mostram que os três eixos apresentaram valores médios semelhantes, próximos de 5° , embora os erros máximos tenham atingido 14.31° no *roll* e 16.49° no *pitch*. No *yaw*, a variação foi mais limitada, com máximo de 7.12° .

TABELA 6.5: Erros de orientação (Ardupilot vs EKF).

Eixo	RMSE [$^\circ$]	Máximo [$^\circ$]
Roll	5.08	14.31
Pitch	5.39	16.49
Yaw	5.42	7.12

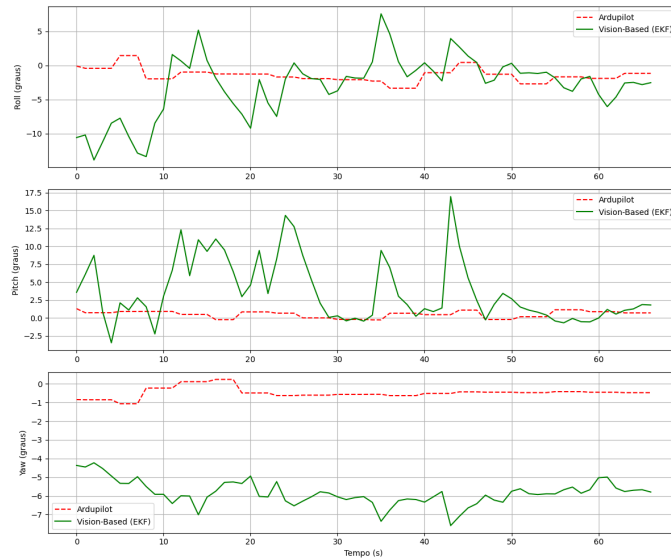


FIGURA 6.6: Erro de orientação estimada (Ardupilot vs EKF).

6.2.3 Trajetória de Aterragem

Foram realizados cinco testes de aterragem em ambiente real. Durante os mesmos, o VANT apresentou anomalias de voo e instabilidade dinâmica, o que condicionou a execução consistente da trajetória gerada. Apesar destas limitações, a melhor aterragem registada ocorreu a aproximadamente 15 cm do centro do marcador.

6.3 Discussão Geral

Os resultados em simulação permitiram validar a exequibilidade do sistema de aterragem automática. Verificou-se uma discrepância consistente entre a telemetria do ArduPilot e o *ground truth* do Gazebo, com erros entre 0.21 m e 0.36 m, o que mostrou ser relevante para a interpretação dos testes em voo real, onde a telemetria do ArduPilot foi a única referência disponível. Na estimativa de pose, o desempenho variou consoante o marcador utilizado: o ArUco de 13.2 cm revelou maior precisão a baixas altitudes, o ArUco de 70 cm garantiu detecções mais robustas até cerca de 16 m, e o marcador em H de 80 cm apresentou boa estabilidade até cerca de 6 m. O EKF mostrou-se útil sobretudo para suavizar a trajetória estimada, ainda que não tenha reduzido significativamente o RMSE. Nos testes de aterragem, ambos os perfis de trajetória se mostraram viáveis, destacando-se o *minimum snap*, que apresentou erros finais inferiores a 8 cm e maior consistência na orientação.

Em ambiente real, os ensaios em bancada evidenciaram boa consistência nos erros de posição, da ordem de 1–2 cm, mas revelaram crescimento dos erros de orientação em *roll* e *pitch* à medida que a distância ao marcador aumentava. Em voo, a análise da estimativa de pose, tomando como referência a telemetria do ArduPilot (com incerteza típica de 1–3 m), revelou diferenças entre os eixos: nos eixos X e Y os RMSE ficaram entre 0.23 m e 0.33 m, enquanto no eixo Z o erro foi mais acentuado, com RMSE de 0.70 m. Quanto à orientação, os três eixos apresentaram valores próximos de 5° , mas com erros máximos que chegaram a 16.49° em *pitch*. Finalmente, nos testes de aterragem, condicionados por instabilidades de voo, a melhor execução registou uma precisão máxima de aterragem de 15 cm.

Em síntese, os resultados em simulação confirmaram a viabilidade técnica do sistema e permitiram identificar a trajetória mais adequada (*minimum snap*). Os testes em voo real, ainda que limitados por restrições de fiabilidade do *ground truth*, instabilidade do VANT e número reduzido de ensaios, demonstraram o potencial de aplicação prática da abordagem proposta, reforçando a necessidade de melhorar a robustez do sistema de percepção e a estabilidade da aeronave para consolidar o desempenho em cenários reais mais exigentes.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões

A presente dissertação teve como principal objetivo o desenvolvimento e validação de um sistema de aterragem automática baseado em visão computacional, destinado a VANTs multirrotores em cenários independentes de GNSS. Esta abordagem surge em resposta às exigências operacionais da Marinha Portuguesa, onde a robustez e fiabilidade do sistema são cruciais para garantir a segurança durante a fase de aterragem, uma das mais críticas e propensas a acidentes em operações com VANTs. Adicionalmente, a adoção de um sistema automático contribui para a redução do número de operacionais necessários por missão, aumentando significativamente a eficiência dos recursos humanos. Esta vantagem torna-se particularmente relevante em plataformas navais de nova geração, como o futuro NRP D. João II, concebido como uma plataforma naval multifuncional com elevada taxa de operações com VANTs, substancialmente superior à verificada nas restantes unidades da esquadra.

Foi desenvolvida uma arquitetura modular que integra perceção visual com marcadores do tipo ArUco e em forma de H, estimativa de pose obtida por visão computacional, filtragem temporal com um EKF e planeamento de trajetória com perfis de *minimum jerk* e *minimum snap* e o controlo de alto nível implementado utilizando o protocolo MAVLink.

A validação foi conduzida em duas fases: Simulação e voo real. Em simulação, com o Gazebo, ROS 2 e ArduPilot SITL, o sistema foi submetido a dezenas de testes com diferentes posições iniciais, demonstrando elevada precisão (erro máximo inferior a 18 cm) e estabilidade na execução da trajetória. Em voo real, foram testadas as componentes principais do sistema em condições controladas ao ar livre, com um erro mínimo de 15 cm.

Contudo, a validação quantitativa da precisão foi condicionada por diversas limitações. A ausência de uma plataforma móvel, combinada com dificuldades logísticas na instalação do sistema RTK-GNSS, comprometeu a recolha de dados de *ground truth* com precisão comparável à do simulador. Acresce que o sistema de RTK-GNSS disponível exigia um período mínimo de estabilização superior a três horas, alcançando apenas uma precisão na ordem dos 37 cm, o que inviabilizou a sua utilização sistemática em campo. Durante os ensaios práticos, registou-se ainda instabilidade estrutural da aeronave utilizada, fator alheio ao sistema desenvolvido, que afetou a consistência de alguns testes. Adicionalmente, quando foi possível aceder ao serviço de transporte em rede de mensagens RTCM via protocolo de internet (*Networked Transport of RTCM via Internet Protocol*, NTRIP) para suporte ao RTK-GNSS, a aeronave encontrava-se inoperacional devido a uma avaria, impossibilitando a realização de novos testes até ao final da dissertação. Apesar destas limitações, as estimativas de pose revelaram-se estáveis e as trajetórias executadas mostraram-se coerentes com os resultados obtidos em simulação.

O contributo desta dissertação para o estado da arte reside na demonstração experimental, tanto em simulação como em testes reais, da viabilidade de um sistema de aterragem visual automático sem dependência de GNSS, com execução modular, utilizando apenas sensores de baixo custo e algoritmos otimizados. Para além da demonstração funcional, esta investigação incluiu uma análise comparativa entre dois tipos de marcadores, o ArUco e um marcador em forma de H, avaliando o seu desempenho na estimativa de pose. Foram geradas e comparadas trajetórias com perfis de *minimum jerk* e *minimum snap*, permitindo analisar o impacto de cada abordagem na suavidade do movimento e na precisão da aterragem. A comparação com a literatura evidencia que os resultados obtidos se inserem dentro dos intervalos de precisão alcançados por abordagens similares, apresentando vantagens adicionais em termos de simplicidade, flexibilidade e integração do sistema.

7.2 Trabalho Futuro

Com base na experiência adquirida e nas limitações enfrentadas, identificam-se várias linhas de desenvolvimento futuro. A mais imediata consiste na transição da execução dos algoritmos da estação terrestre para uma unidade computacional embarcada, como o Raspberry Pi, já testado preliminarmente, a fim de reduzir a latência e aumentar a autonomia do sistema.

A integração do sistema com plataformas móveis, como veículos terrestres

ou plataformas navais, é fundamental para completar a validação do sistema em cenários realistas de aplicação final. Essa transição permitirá validar o desempenho do sistema com a variável do movimento relativo da plataforma.

Adicionalmente, deve também ser integrada uma câmara *gimbal* no lugar da câmara fixa, de modo a compensar movimentos bruscos e vibrações durante o voo, aumentando a estabilidade da imagem e consequentemente, a fiabilidade da estimativa de pose ao longo da trajetória de aterragem.

Finalmente, o sistema pode ser expandido para permitir a colaboração multi-agente entre VANTs, por exemplo, com partilha de pose e coordenação para aterragem em sequência ou missões cooperativas, alinhando-se com as tendências atuais de sistemas aéreos em enxame.

Bibliografia

- Acuna, Raul, Ding Zhang e Volker Willert (2018). «Vision-based UAV landing on a moving platform in GPS denied environments using motion prediction». Em: *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*. IEEE, pp. 515–521.
- Alijani, Morteza e Anas Osman (2020). «Autonomous landing of uav on moving platform: A mathematical approach». Em: *2020 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE, pp. 1–6.
- ArduPilot Developers (2014). *Advanced Pixhawk Quadcopter Wiring Chart*. Acesso em: 21 maio 2025. URL: <https://ardupilot.org/copter/docs/advanced-pixhawk-quadcopter-wiring-chart.html>.
- Arjomandi, Maziar, Shane Agostino, Matthew Mammone, Matthieu Nelson e Tong Zhou (2006). «Classification of unmanned aerial vehicles». Em: *Report for Mechanical Engineering class, University of Adelaide, Adelaide, Australia*, pp. 1–48.
- Asim, Muhammad, DN Ehsan e Khalid Rafique (2005). «Probable causal factors in uav accidents based on human factor analysis and classification system». Em: *history* 1905, p. 5.
- Austin, Reg (2011). *Unmanned aircraft systems: UAVS design, development and deployment*. John Wiley & Sons.
- Boyle, Ashley (2012). *The US and its UAVs: The Financial Cost versus Strategic Value of Drones*. <https://www.americansecurityproject.org/the-us-and-its-uavs-the-financial-cost-versus-strategic-value-of-drones/>. Accessed: 2024-05-20.
- Cao, Yanan, Minglei Li, Li Xu e Mingqiang Wei (2024). «Autonomous Landing Spot Detection for Unmanned Aerial Vehicles Based on Monocular Vision». Em: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 48, pp. 51–57.
- Chai, Tianfeng e Roland R Draxler (2014). «Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature». Em: *Geoscientific model development* 7.3, pp. 1247–1250.

- Chang, C.W., L.Y. Lo, H.C. Cheung, Y. Feng e A.S. Yang (jan. de 2022). «Pro-active Guidance for Accurate UAV Landing on a Dynamic Platform: A Visual-Inertial Approach». Em: *Sensors* 22.1 (1), p. 404. ISSN: 14248220. URL: <https://www.mdpi.com/1424-8220/22/1/404>.
- China Military (2024). *UAV jammed and downed: A new confrontation mode emerges*. Acedido em: 19 junho 2025. URL: http://eng.chinamil.com.cn/CHINA_209163/WeaponryEquipment/News_209182/9328294.html.
- Damas, Bruno, Nuno Pessanha Santos e Matilde Correia Vieira (2024). «A Visual Tracking System for UAV Landing on Ships». Em: pp. 2627–2632. DOI: 10.1109/CoDIT62066.2024.10708450.
- Defense, Breaking (jul. de 2024). *From Ukraine to Taiwan: Jamming of 50-Year-Old GPS Is a Defense Tech Nightmare*. Acedido em: 19 de junho de 2025. URL: <https://breakingdefense.com/2024/07/from-ukraine-to-taiwan-jamming-of-50-year-old-gps-is-a-defense-tech-nightmare/>.
- Delbene, Andrea, Marco Baglietto e Enrico Simetti (2022). «Visual Servoed Autonomous Landing of an UAV on a Catamaran in a Marine Environment». Em: *Sensors* 22.9. ISSN: 1424-8220. URL: <https://www.mdpi.com/1424-8220/22/9/3544>.
- Dutrannois, Thomas, Tien-Thanh Nguyen, Charles Hamesse, Geert De Cubber e Bart Janssens (2022). «Visual SLAM for autonomous drone landing on a maritime platform». Em: *2022 International Symposium on Measurement and Control in Robotics (ISMCR)*. IEEE, pp. 1–7.
- Fahlstrom, Paul G, Thomas J Gleason e Mohammad H Sadraey (2022). *Introduction to UAV systems*. John Wiley & Sons.
- Falanga, Davide, Alessio Zanchettin, Alessandro Simovic, Jeffrey Delmerico e Davide Scaramuzza (2017). «Vision-based autonomous quadrotor landing on a moving platform». Em: *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, pp. 200–207.
- Feng, Yi, Cong Zhang, Stanley Baek, Samir Rawashdeh e Alireza Mohammadi (2018). «Autonomous landing of a UAV on a moving platform using model predictive control». Em: *Drones* 2.4, p. 34.
- Garrido-Jurado, S., R. Muñoz-Salinas, F.J. Madrid-Cuevas e M.J. Marín-Jiménez (2014). «Automatic generation and detection of highly reliable fiducial markers under occlusion». Em: *Pattern Recognition* 47.6, pp. 2280–2292. ISSN: 0031-3203. URL: <https://www.sciencedirect.com/science/article/pii/S0031320314000235>.

- Gautam, Alvika, Mandeep Singh, Pedda Baliyarasimhuni Sujit e Srikanth Saripalli (fev. de 2022). «Autonomous Quadcopter Landing on a Moving Target». Em: *Sensors* 22.3 (3). Applies polynomial velocity profile for controlled descent using visual pose, p. 1116. ISSN: 14248220. DOI: 10.3390/s22031116.
- Grindley, Ben, Katie Phillips, Katie J. Parnell, Tom Cherrett, James Scanlan e Katherine L. Plant (2024). «Over a decade of UAV incidents: A human factors analysis of causal factors». Em: *Applied Ergonomics* 121, p. 104355. ISSN: 0003-6870. URL: <https://www.sciencedirect.com/science/article/pii/S0003687024001327>.
- Guo, Ziqing, Jianhua Wang, Xiang Zheng, Yuhang Zhou e Jiaqing Zhang (mai. de 2025). «A Visual Guidance and Control Method for Autonomous Landing of a Quadrotor UAV on a Small USV». Em: *Drones* 9.5 (5). Atira se a uma rede , nao e grande coisa, p. 364. ISSN: 2504-446X. URL: <https://www.mdpi.com/2504-446X/9/5/364>.
- Hartley, Richard e Andrew Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.
- Hassanalian, M. e A. Abdelkefi (2017). «Classifications, applications, and design challenges of drones: A review». Em: *Progress in Aerospace Sciences* 91, pp. 99–131. ISSN: 0376-0421. URL: <https://www.sciencedirect.com/science/article/pii/S0376042116301348>.
- Iwakura, Daisuke, Wei Wang, Kenzo Nonami e Mark Haley (2011). «Movable Range-Finding Sensor System and Precise Automated Landing of Quad-Rotor MAV». Em: *Journal of System Design and Dynamics* 5.1, pp. 17–29.
- JOUAV (2023). *RTK Drone: The Ultimate Guide*. Acedido em 19 de junho de 2025. URL: <https://www.jouav.com/blog/rtk-drone.html>.
- Kakaletsis, Efstratios, Charalampos Symeonidis, Maria Tzelepi, Ioannis Mademlis, Anastasios Tefas, Nikos Nikolaidis e Ioannis Pitas (out. de 2021). «Computer Vision for Autonomous UAV Flight Safety: An Overview and a Vision-based Safe Landing Pipeline Example». Em: *ACM Comput. Surv.* 54.9. ISSN: 0360-0300. URL: <https://doi.org/10.1145/3472288>.
- Kalman, R. E. (mar. de 1960). «A New Approach to Linear Filtering and Prediction Problems». Em: *Journal of Basic Engineering* 82.1, pp. 35–45. ISSN: 0021-9223. URL: <https://doi.org/10.1115/1.3662552>.
- Kelner, JM e Cezary Ziółkowski (2019). «Errors of UAV autonomous landing system for different radio beacon configurations». Em: *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation* 13.2, pp. 429–435.

- Kendall, Alex, Matthew Grimes e Roberto Cipolla (2015). «PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization». Em: pp. 2938–2946. URL: <https://arxiv.org/abs/1505.07427>.
- Khazetdinov, Artur, Aufar Zakiev, Tatyana Tsoy, Mikhail Svinin e Evgeni Magid (mai. de 2021). «Embedded ArUco: A novel approach for high precision UAV landing». Em: DOI: 10.1109/SIBCON50419.2021.9438855.
- Lee, Bochan, Vishnu Saj, Moble Benedict e Dileep Kalathil (2020). «A Vision-Based Control Method for Autonomous Landing of Vertical Flight Aircraft On a Moving Platform Without Using GPS». Em: *CoRR* abs/2008.05699. URL: <https://arxiv.org/abs/2008.05699>.
- Liu, F., J. Shan, B. Xiong e Z. Fang (2022). «A Real-Time and Multi-Sensor-Based Landing Area Recognition System for UAVs». Em: *Drones* 6.5. Exemplo direto de fusão sensorial para aterragem, p. 118. DOI: 10.3390/drones6050118.
- Lou, L., Y. Li, Q. Zhang e H. Wei (2023). «SLAM and 3D Semantic Reconstruction Based on the Fusion of Lidar and Monocular Vision». Em: *Sensors (Basel, Switzerland)* 23.3, p. 1502. URL: <https://doi.org/10.3390/s23031502>.
- Lu, Xiao Xin (2018). «A review of solutions for perspective-n-point problem in camera pose estimation». Em: *Journal of Physics: Conference Series*. Vol. 1087. 5. IOP Publishing, p. 052009.
- Lyu, Mingyang, Yibo Zhao, Chao Huang e Hailong Huang (jun. de 2023). «Unmanned Aerial Vehicles for Search and Rescue: A Survey». Em: *Remote Sensing 2023, Vol. 15, Page 3266* 15 (13), p. 3266. ISSN: 2072-4292. URL: <https://www.mdpi.com/2072-4292/15/13/3266/htm%20https://www.mdpi.com/2072-4292/15/13/3266>.
- Maybeck, Peter S. (1979). *Stochastic Models, Estimation, and Control*. Vol. 1. New York: Academic Press.
- Mellinger, Daniel e Vijay Kumar (2011). «Minimum snap trajectory generation and control for quadrotors». Em: pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.
- Mete, Mustafa (2018). «Vision Based Automated Landing of Quadrotors». Em: *ResearchGate*. URL: https://www.researchgate.net/publication/388589418_Vision_Based_Automted_Landing_of_Quadrotors.
- Mohsan, Syed Agha Hassnain, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah e Mohammed H. Alsharif (2022). «Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review». Em: *Drones* 6.6. ISSN: 2504-446X. URL: <https://www.mdpi.com/2504-446X/6/6/147>.

- Morales, Jesús, Isabel Castelo, Rodrigo Serra, Pedro U. Lima e Meysam Basiri (jan. de 2023). «Vision-Based Autonomous Following of a Moving Platform and Landing for an Unmanned Aerial Vehicle». Em: *Sensors* 23 (2). proximo a ler, p. 829. ISSN: 14248220. URL: <https://www.mdpi.com/1424-8220/23/2/829>.
- Nguyen, Phong Ha, Ki Wan Kim, Young Won Lee e Kang Ryoung Park (set. de 2017). «Remote marker-based tracking for uav landing using visible-light camera sensor». Em: *Sensors (Switzerland)* 17.9 (9). Referência crucial para uso de marcadores ArUco ou visuais, p. 1987. ISSN: 14248220. DOI: 10.3390/s17091987.
- Northern Digital Inc (2025). *6 DOF Explained*. <https://www.ndigital.com/6dof-explained/>. Acedido em 21 de agosto de 2025.
- Oliveira Filho, José de Arimatéia de (2017). «Sistema de processamento de imagens aéreas de baixo custo embarcado em computadores de porte leve e de baixo consumo de energia aplicados ao monitoramento de áreas agrícolas ou regiões ambientais em tempo real». Dissertação de Mestrado. João Pessoa, Brasil: Universidade Federal da Paraíba. URL: <https://repositorio.ufpb.br/jspui/bitstream/tede/9262/2/arquivototal.pdf>.
- Omar, Hanafy M., Rizwan Akram, Saad M.S. Mukras e Ahmed Alaa Mahvouz (2023). «Recent advances and challenges in controlling quadrotors with suspended loads». Em: *Alexandria Engineering Journal* 63, pp. 253–270. ISSN: 1110-0168. URL: <https://www.sciencedirect.com/science/article/pii/S1110016822005245>.
- OpenCV (2024). *Detection of ArUco Markers*. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html. Acedido em: 1 de junho de 2025.
- OpenCV Documentation Team (2025). *Camera Calibration with OpenCV*. Acesso em: 19 de junho de 2025. URL: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html.
- OpenMV (2020). *AprilTag Marker Tracking – The Future is Here*. Acedido em 20 de junho de 2025. URL: <https://openmv.io/blogs/news/apriltag-marker-tracking-the-future-is-here>.
- Park, Younghoo, Chanhwi Park, Wooseok Song, Chulyong Lee, Junsoo Kwon, Jihoon Park, Geemoon Noh e Daewoo Lee (jan. de 2024). «Fiducial Marker-Based Autonomous Landing Using Image Filter and Kalman Filter». Em: *International Journal of Aeronautical and Space Sciences* 25 (1), pp. 190–199. ISSN:

20932480. URL: <https://link.springer.com/article/10.1007/s42405-023-00635-y>.
- Patruno, Cosimo, Massimiliano Nitti, Antonio Petitti, Ettore Stella e Tiziana D’Orazio (2019). «A vision-based approach for unmanned aerial vehicle landing». Em: *Journal of Intelligent & Robotic Systems* 95.2, pp. 645–664.
- Pessanha Santos, N., A. Bernardino e V. Lobo (2020). «Directional Statistics for 3D Model-Based UAV Tracking». Em: *IEEE Access* 8, pp. 33884–33897. DOI: 10.1109/ACCESS.2020.2973970.
- Pessanha Santos, Nuno (2020). «Fixed-wing UAV tracking in outdoor scenarios for autonomous landing». Tese de doutoramento. Universidade de Lisboa, Instituto Superior Técnico.
- (2024). «Fixed-Wing UAV Pose Estimation Using a Self-Organizing Map and Deep Learning». Em: *Robotics* 13.8, p. 114. DOI: 10.3390/robotics13080114.
- Qin, Tong, Peiliang Li e Shaojie Shen (2018). «Vins-mono: A robust and versatile monocular visual-inertial state estimator». Em: *IEEE transactions on robotics* 34.4, pp. 1004–1020.
- Ribeiro, Maria Isabel (2004). «Kalman and extended kalman filters: Concept, derivation and properties». Em: *Institute for Systems and Robotics* 43.46, pp. 3736–3741.
- Rosa, Gonalo Castanheira, Mario Monteiro Marques e Victor Lobo (2016). «Unmanned aerial vehicles in the Navy: Its benefits». Em: *Sci. Bull. Nav. Acad* 19.1, pp. 39–43.
- Sani, Mohammad Fattahi e Ghader Karimian (2017). «Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors». Em: *2017 International Conference on Computer and Drone Applications (ICONDA)*, pp. 102–107. DOI: 10.1109/ICONDA.2017.8270408.
- Sayeed, Azam e Noor Ayesha (abr. de 2020). «Detecting Crows on Sowed Crop Fields using Simplistic Image processing Techniques by OpenCV in comparison with TensorFlow Image Detection API». Em.
- Serra, Pedro, Rita Cunha, Tarek Hamel, David Cabecinhas e Carlos Silvestre (2016). «Landing of a quadrotor on a moving target using dynamic image-based visual servo control». Em: *IEEE Transactions on Robotics* 32.6, pp. 1524–1535.
- Sousa, Joo Pedro Jacinto (2022). «Fiducial Markers for Autonomous UAV Landing». Dissertao de Mestrado. Coimbra, Portugal: Universidade de Coimbra. URL: https://estudogeral.uc.pt/retrieve/260729/dissertacao_joao_sousa.pdf.

- Szeliski, Richard (2022). *Computer Vision: Algorithms and Applications*. Springer. URL: <https://link.springer.com/book/10.1007/978-3-030-34372-9>.
- Valavanis, Kimon P e George J Vachtsevanos (2014). *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated.
- Vieira, Matilde (2023). *Integração de um sistema de apoio à aterragem de Veículos Aéreos Não Tripulados (UAVs) em navios*.
- Wang, P., Y. Gao, Q. Zhao, Y. Wang, F. Zhou e D. Zhang (2025). «An Enhanced, Real-Time, Low-Cost GNSS/INS Integrated Navigation Algorithm and Its Platform Design». Em: *Sensors (Basel, Switzerland)* 25.7, p. 2119. URL: <https://doi.org/10.3390/s25072119>.
- Wang, Pengyu, Chaoqun Wang, Jiankun Wang e Max Q.-H. Meng (2022). «Quadrotor Autonomous Landing on Moving Platform». Em: *Procedia Computer Science* 209. Proceedings of the 2022 International Symposium on Biomimetic Intelligence and Robotics (ISBIR), pp. 40–49. ISSN: 1877-0509. URL: <https://www.sciencedirect.com/science/article/pii/S1877050922015460>.
- Welch, Greg, Gary Bishop et al. (1995). «An introduction to the Kalman filter». Em.
- Wubben, Jamie, Francisco Fabra, Carlos T. Calafate, Tomasz Krzeszowski, Johann M. Marquez-Barja, Juan-Carlos Cano e Pietro Manzoni (2019). «A vision-based system for autonomous vertical landing of unmanned aerial vehicles». Em: *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pp. 1–7. DOI: 10.1109/DS-RT47707.2019.8958701.
- Wynn, Jesse Stewart (2018). *Visual servoing for precision shipboard landing of an autonomous multirotor aircraft system*. Brigham Young University.
- Xin, Long, Zimu Tang, Weiqi Gai e Haobo Liu (nov. de 2022). «Vision-Based Autonomous Landing for the UAV: A Review». Em: *Aerospace* 9.11 (11). Revisão ampla sobre técnicas de aterragem com visão computacional, pp. 1–11. ISSN: 2226-4310. URL: <https://www.mdpi.com/2226-4310/9/11/634>.
- Xuan-Mung, Nguyen, Ngoc Phi Nguyen, Tan Nguyen, Dinh Ba Pham, Mai The Vu, Ha Le Nhu Ngoc Thanh e Sung Kyung Hong (2022). «Quadcopter Precision Landing on Moving Targets via Disturbance Observer-Based Controller and Autonomous Landing Planner». Em: *IEEE Access* 10, pp. 83580–83590. ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3197181.
- Zhang, B., Z. Song e F. Zhao (2022). «Overview of Propulsion Systems for Unmanned Aerial Vehicles». Em: *Energies* 15.2. Classificação e avaliação de sistemas de propulsão — útil no Cap. 2.1.2, pp. 1–25. DOI: 10.3390/en15020455.

- Zhang, Z. (2000). «A flexible new technique for camera calibration». Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11. Desenvolvimento clássico em calibração de câmera — importante na visão computacional, pp. 1330–1334. DOI: 10.1109/34.888718.
- Zhang, Zhengyou (2000). *A Flexible New Technique for Camera Calibration*. Rel. téc.
- Zhou, Jianwei, Wenjie Wang, Xi Hong e Chenhao Zhang (2024). «Multi-UAV Cooperative Anti-Jamming for GNSS Signals Based on Frequency-Domain Power Inversion». Em: *IEEE Sensors Journal* 24.19, pp. 30778–30791. DOI: 10.1109/JSEN.2024.3437643.
- Zhu, N., D. Bétaille e M. Berbineau (2018). «GNSS Position Integrity in Urban Environments: A Review of Literature». Em: *IEEE Transactions on Intelligent Transportation Systems* 19.9. Limitações do GNSS em ambientes urbanos — aplicável a cenários GNSS-denied, pp. 2762–2778. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8248655&isnumber=8458340>.

Apêndice A - Testes ao RTK-GNSS

Neste apêndice são inicialmente apresentados os resultados e a instalação de testes ao RTK (Here+) da CEOV, realizados nas instalações da CEOV e da Escola de Tecnologias Navais (ETNA), no qual a melhor precisão foi de 37 cm (Tabela A.1), após aproximadamente três horas e trinta minutos de *survey-in* nas instalações da ETNA (Figura A.1). Seguidamente, são apresentados os resultados e a instalação de testes ao RTK (Here+) do Instituto Superior Técnico de Lisboa, realizados nas suas instalações, com uma precisão de 30 cm (Tabela A.2), após aproximadamente trinta e dois minutos de *survey-in* no Instituto Superior Técnico (Figura A.2).

TABELA A.1: Resultados do processo de *survey-in* do RTK-GNSS

Duração <i>survey-in</i>	3 h 30 min
Precisão	0.37 m
Número de Satélites	20



FIGURA A.1: Testes ao RTK-GNSS nas instalações da CEOV (esquerda) e da ETNA (direita).

TABELA A.2: Resultados do processo de *survey-in* do RTK-GNSS do IST

Duração <i>survey-in</i>	32 min
Precisão	0.27 m
Número de Satélites	20

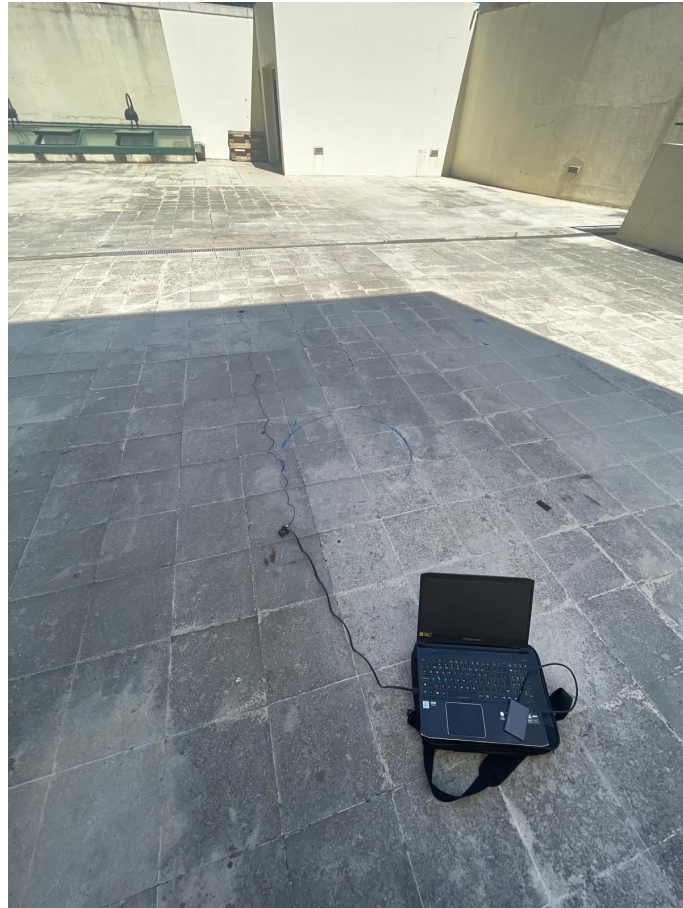


FIGURA A.2: Testes ao RTK-GNSS nas instalações do IST.

Apêndice B - Comparação *Ground Truth*

Este apêndice apresenta a análise do erro de posição tridimensional (RMSE) entre a estimativa de posição fornecida pelo ArduPilot e o *Ground Truth* obtido através do simulador Gazebo, para diferentes altitudes. Os testes foram realizados em ambiente simulado com o objetivo de avaliar a fiabilidade da telemetria do ArduPilot como referência relativa em contexto real. A Tabela B.1 e a Figura B.1 mostram que o erro se mantém estável entre os 2m e os 8m, com valores de RMSE entre 0.26 m e 0.29 m, aumentando ligeiramente a partir dos 9 m.

TABELA B.1: Erro de posição tridimensional (RMSE) entre o ArduPilot e o Gazebo a diferentes altitudes.

Altura (m)	0.97	1.97	2.97	3.97	4.97	5.98	6.97	7.97	8.98	9.37
RMSE (m)	0.213	0.281	0.259	0.286	0.281	0.273	0.293	0.276	0.291	0.357

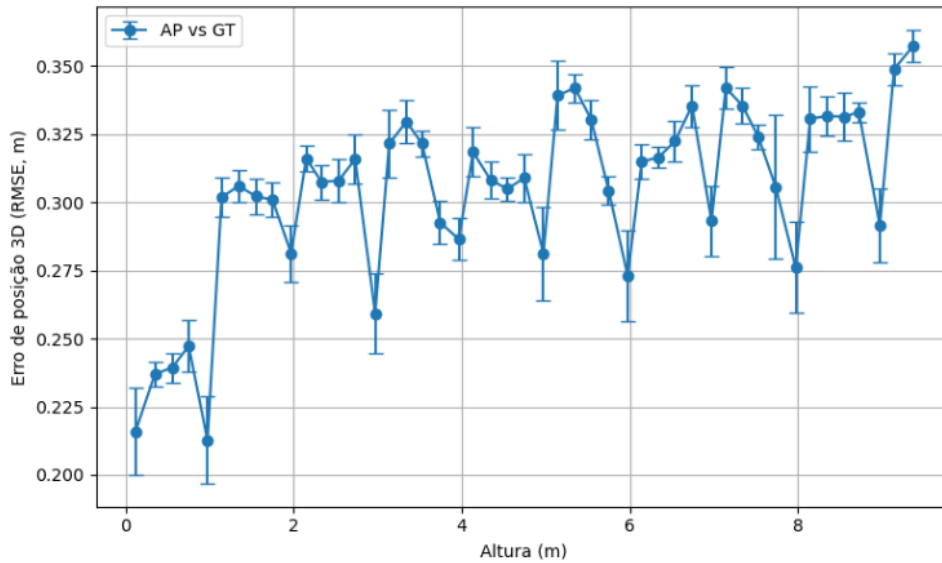


FIGURA B.1: Erro da posição estimada pelo ArduPilot em relação ao *ground truth* do Gazebo, em função da altitude.

Apêndice C - Alcance e Zona de Segurança

Este apêndice apresenta os resultados dos testes de estimativa de pose em relação à altitude, realizados em simulação com diferentes tipos de marcadores (ArUco de 13.2 cm, ArUco de 70 cm e marcador em forma de H com 80 cm). O objetivo é caracterizar o alcance útil de detecção e a zona de segurança operacional, analisando a evolução do erro de pose em função da altitude e a frequência média de detecção em cada caso. As tabelas e figuras seguintes reúnem os valores de RMSE obtidos para os dados brutos e após filtragem com EKF, permitindo avaliar a fiabilidade de cada marcador em diferentes condições de operação.

C.1 ArUco de 13.2 cm

Esta secção apresenta os resultados da estimativa de pose em função da altura com o marcador ArUco de 13.2 cm em ambiente simulado. A Tabela C.1 e a Figura C.1 mostram o erro quadrático médio (RMSE) da posição estimada em função da altura, para os dados brutos (RAW) e após aplicação de um EKF, bem como a frequência média de detecção.

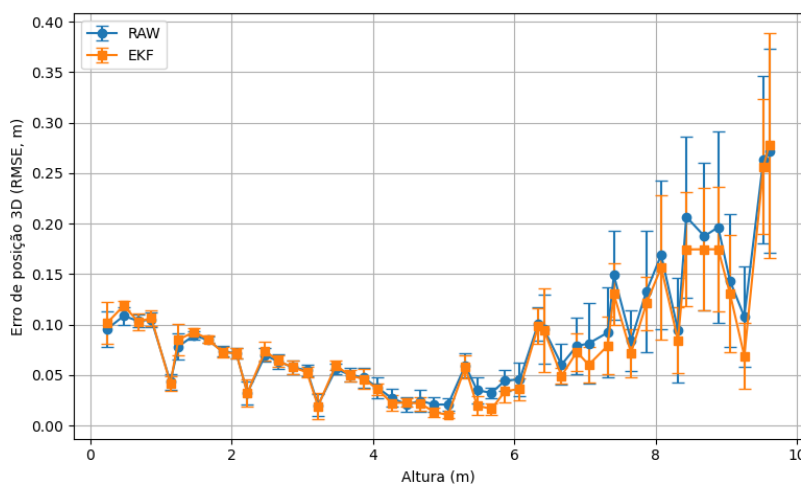


FIGURA C.1: Erro da estimativa de pose em função da altura, com marcador ArUco de 13.2 cm.

TABELA C.1: Erro de estimativa de pose e frequência de detecção em função da altitude com marcador ArUco de 13.2 cm.

Altitude (m)	RMSE RAW (m)	RMSE EKF (m)	Freq. detecção (Hz)
0.87	0.105	0.107	9.51
2.07	0.072	0.071	9.48
3.07	0.055	0.053	9.82
4.07	0.037	0.036	9.64
5.06	0.021	0.010	10.24
6.08	0.046	0.036	10.26
7.05	0.081	0.060	10.71
8.08	0.169	0.157	9.53
9.05	0.143	0.131	1.08

C.2 ArUco de 70 cm

Apresentam-se os resultados da estimativa de pose em função da altura com o marcador ArUco de 70 cm, em ambiente simulado. A Tabela C.2 e a Figura C.2 indicam os valores do erro quadrático médio (RMSE) da posição estimada, para dados brutos (RAW) e filtrados com EKF, bem como a frequência média de detecção do marcador.

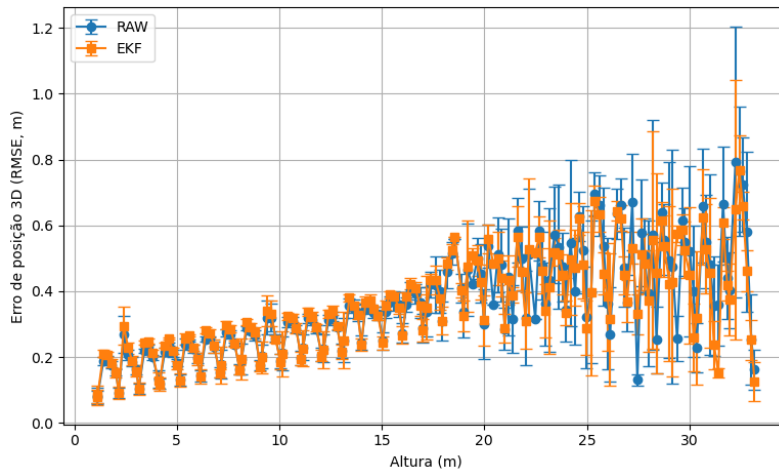


FIGURA C.2: Erro da estimativa de pose em função da altura, com marcador ArUco de 70 cm.

TABELA C.2: Erro de estimativa de pose e frequência de detecção em função da altura com marcador ArUco de 70 cm.

Altura (m)	RMSE RAW (m)	RMSE EKF (m)	Freq. detecção (Hz)
1.09	0.078	0.080	7.56
2.03	0.154	0.152	11.72
3.02	0.162	0.158	11.25
4.09	0.133	0.129	5.05
5.03	0.182	0.175	11.88
6.02	0.197	0.194	12.42
7.09	0.153	0.151	9.45
8.07	0.163	0.161	9.70
9.07	0.175	0.173	9.62
10.05	0.189	0.186	9.61
11.07	0.198	0.195	9.37
12.06	0.208	0.204	9.30
13.02	0.220	0.216	9.47
14.00	0.243	0.238	10.10
15.05	0.250	0.244	9.42
16.01	0.273	0.267	9.01
16.98	0.288	0.282	7.86
17.96	0.310	0.311	4.43
18.98	0.339	0.325	4.91
19.99	0.300	0.311	2.56
21.00	0.286	0.288	4.34
22.01	0.319	0.311	4.39
22.98	0.362	0.341	1.85
23.96	0.343	0.335	3.46
24.99	0.321	0.289	1.58
26.01	0.363	0.383	2.83
26.95	0.365	0.369	2.91
28.00	0.394	0.370	1.68
29.03	0.494	0.423	1.04
30.03	0.521	0.449	0.92
31.01	0.482	0.456	1.76
31.95	0.404	0.375	4.04
33.02	0.254	0.253	1.46

C.3 H de 80cm

Apresentam-se os resultados da estimativa de pose com o marcador H de 80 cm em ambiente simulado. A Tabela C.3 e a Figura C.3 mostram os valores do erro quadrático médio (RMSE) da posição estimada em função da altura, para dados brutos (RAW) e filtrados com EKF, bem como a frequência média de detecção.

TABELA C.3: Erro de estimativa de pose e frequência de detecção em função da altura com marcador H de 80 cm.

Altura (m)	RMSE RAW (m)	RMSE EKF (m)	Freq. detecção (Hz)
0.94	0.356	0.350	7.14
1.91	0.256	0.253	4.36
2.90	0.077	0.066	6.78
3.93	0.153	0.154	7.96
4.94	0.065	0.065	9.65
5.93	0.111	0.111	8.69
6.93	0.064	0.114	4.16
7.95	0.780	0.477	0.45
8.92	0.061	0.324	0.63
9.91	0.081	0.554	0.62
10.95	1.720	0.712	0.12
11.96	1.418	1.110	0.28
12.96	0.896	0.817	0.41

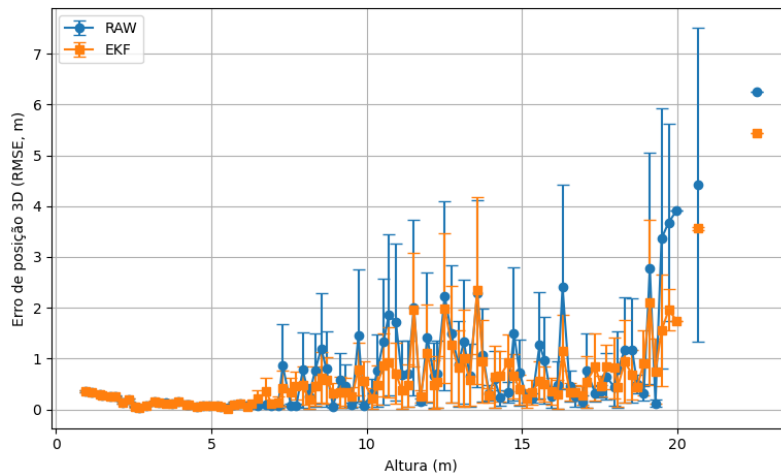


FIGURA C.3: Erro da estimativa de pose em função da altura, com marcador H de 80 cm.

Apêndice D - Trajetória com formato infinito

Este apêndice apresenta os resultados dos ensaios realizados em simulação com uma trajetória em formato de “infinito”, concebida para avaliar o desempenho do sistema em movimento lateral e com variação contínua de direção. São analisados três tipos de marcadores (ArUco de 13.2 cm, ArUco de 70 cm e marcador em forma de H de 80 cm) em diferentes alturas de voo. As tabelas e figuras seguintes mostram os valores de erro quadrático médio (RMSE) da posição e orientação estimadas, comparando os dados brutos (RAW) com os filtrados por EKF, bem como as trajetórias 3D obtidas face ao *ground truth*. O objetivo é caracterizar a robustez do sistema de estimação em cenários dinâmicos, onde a trajetória não se limita a movimentos retos ou verticais.

D.1 ArUco de 13.2 cm

Apresentam-se os resultados obtidos para a estimativa de pose durante a execução de uma trajetória em forma de “infinito” a 3 metros de altura, utilizando o marcador ArUco de 13.2 cm. As Tabelas D.1 e D.2 mostram os valores de erro quadrático médio (RMSE) da posição e orientação estimadas, comparando os dados brutos (RAW) com os filtrados por EKF. As Figuras D.1, D.2 e D.3 ilustram os erros e a trajetória 3D resultante.

TABELA D.1: Erro de posição (RMSE) para trajetória em formato infinito com marcador ArUco de 13.2 cm.

Eixo	RMSE (RAW) [m]	RMSE (EKF) [m]	Diferença [m]
X	0.1586	0.1697	+0.0111
Y	0.1535	0.1612	+0.0077
Z	0.0872	0.0887	+0.0015

TABELA D.2: Erro de orientação (RMSE) para trajetória em formato infinito com marcador ArUco de 13.2 cm.

Eixo	RMSE (RAW) [°]	RMSE (EKF) [°]	Melhoria [°]
Roll	3.32	2.31	1.01
Pitch	3.90	2.99	0.91
Yaw	0.32	0.19	0.13

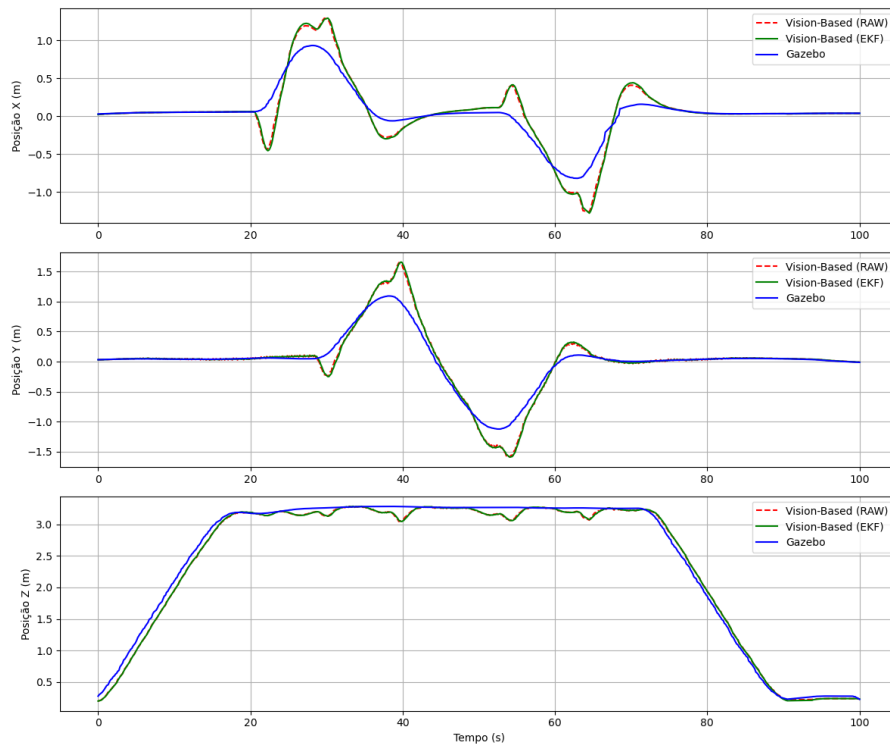


FIGURA D.1: Erro de posição estimada (RAW vs EKF) — Trajetória em formato infinito.

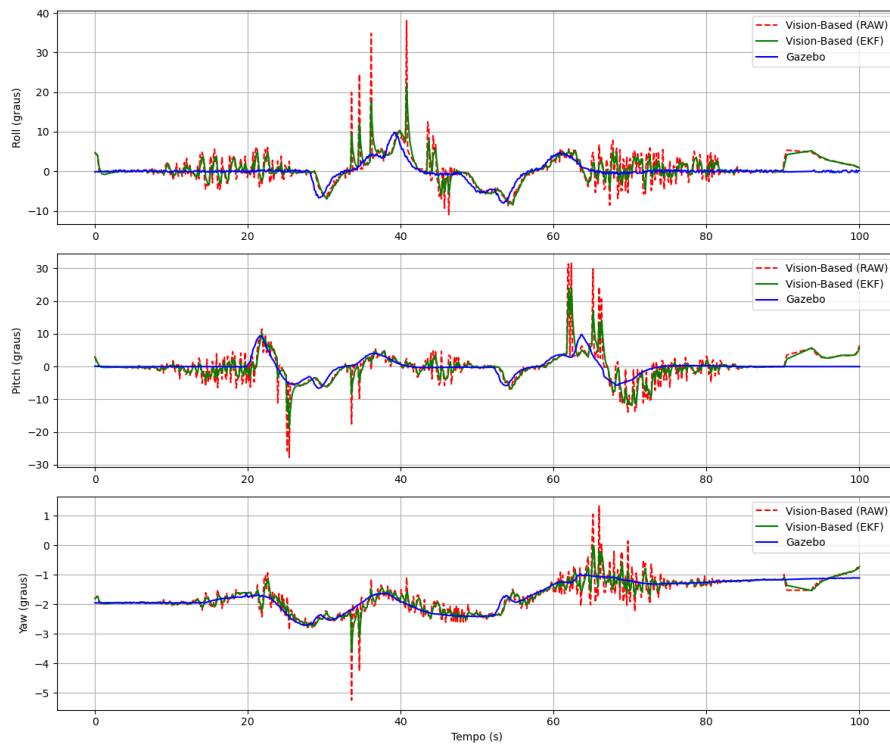


FIGURA D.2: Erro de orientação estimada (RAW vs EKF) — Trajetória em formato infinito.

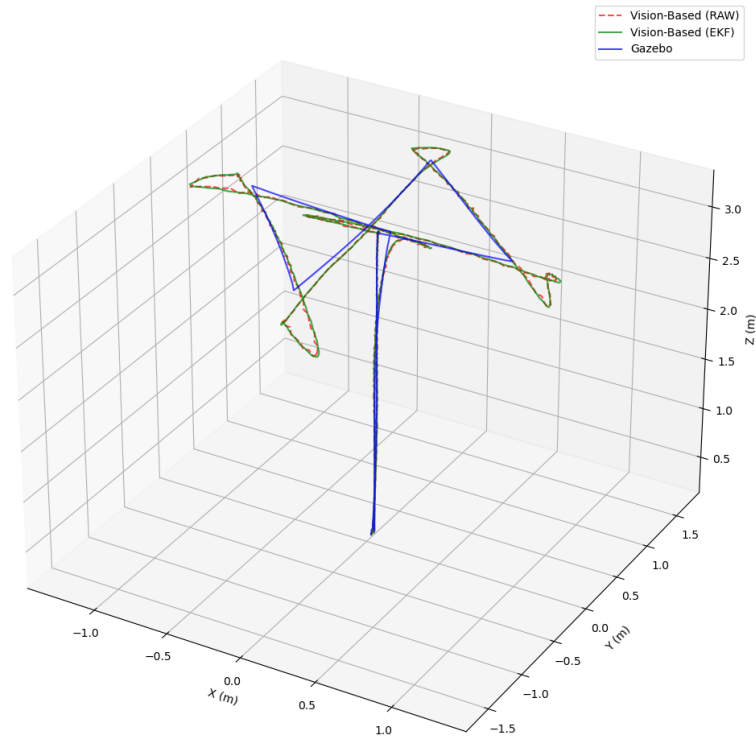


FIGURA D.3: Comparação da trajetória 3D estimada (RAW e EKF) com o *ground truth*.

D.2 ArUco de 70 cm

Apresentam-se os resultados obtidos para a estimativa de pose durante a execução de uma trajetória em forma de “infinito” a 4 metros de altura, utilizando o marcador ArUco de 70 cm. As Tabelas D.3 e D.4 mostram os valores de erro quadrático médio (RMSE) da posição e orientação estimadas, comparando os dados brutos (RAW) com os filtrados por EKF. As Figuras D.4, D.5 e D.6 ilustram os erros e a trajetória 3D estimada.

TABELA D.3: Erro de posição (RMSE) para trajetória em formato infinito com marcador ArUco de 70 cm.

Eixo	RMSE (RAW) [m]	RMSE (EKF) [m]	Diferença [m]
X	0.1866	0.1987	+0.0121
Y	0.1948	0.2034	+0.0086
Z	0.1583	0.1586	+0.0003

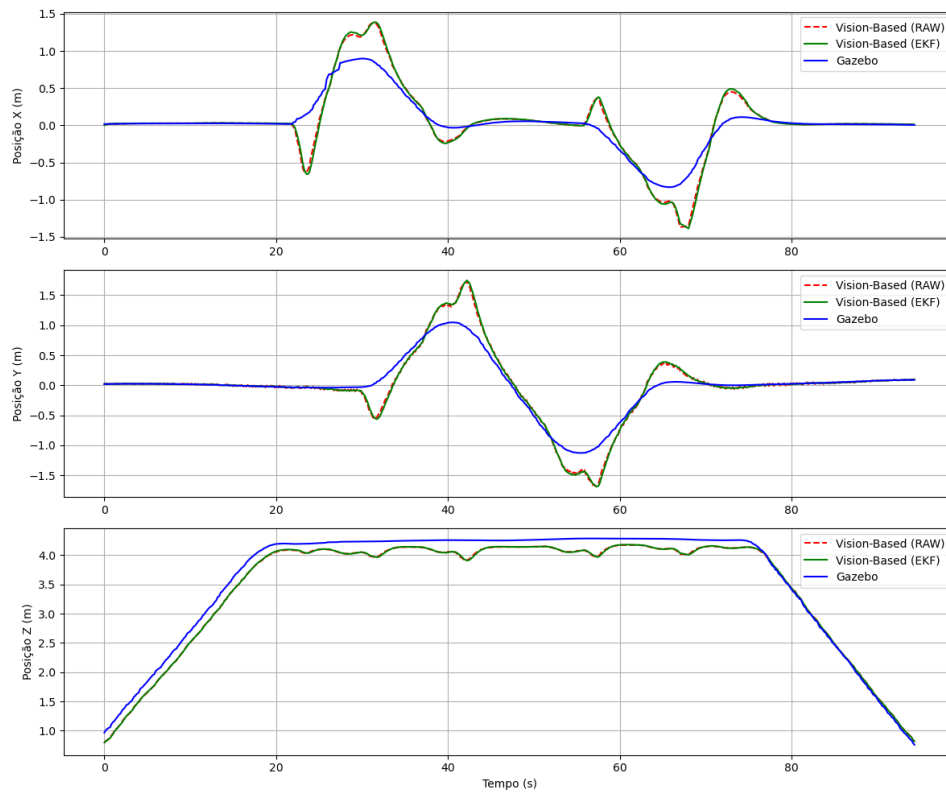


FIGURA D.4: Erro de posição estimada (RAW vs EKF) — Trajetória em formato infinito, 4 m.

TABELA D.4: Erro de orientação (RMSE) para trajetória em formato infinito com marcador ArUco de 70 cm.

Eixo	RMSE (RAW) [°]	RMSE (EKF) [°]	Diferença [°]
Roll	0.805	0.929	+0.124
Pitch	1.266	1.398	+0.132
Yaw	0.069	0.074	+0.005

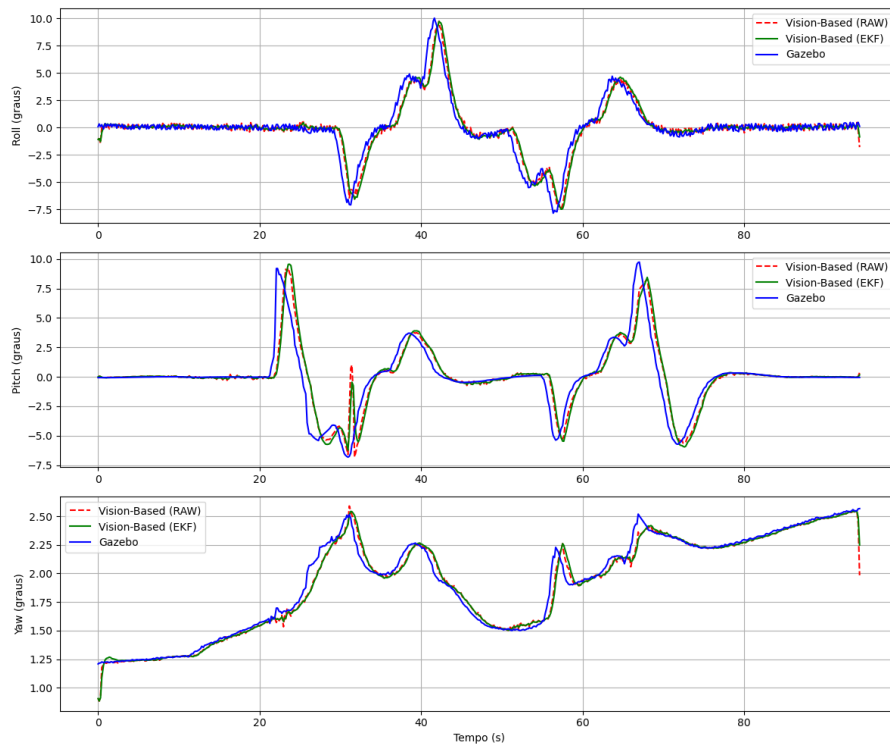


FIGURA D.5: Erro de orientação estimada (RAW vs EKF) — Trajetória em formato infinito, 4 m.

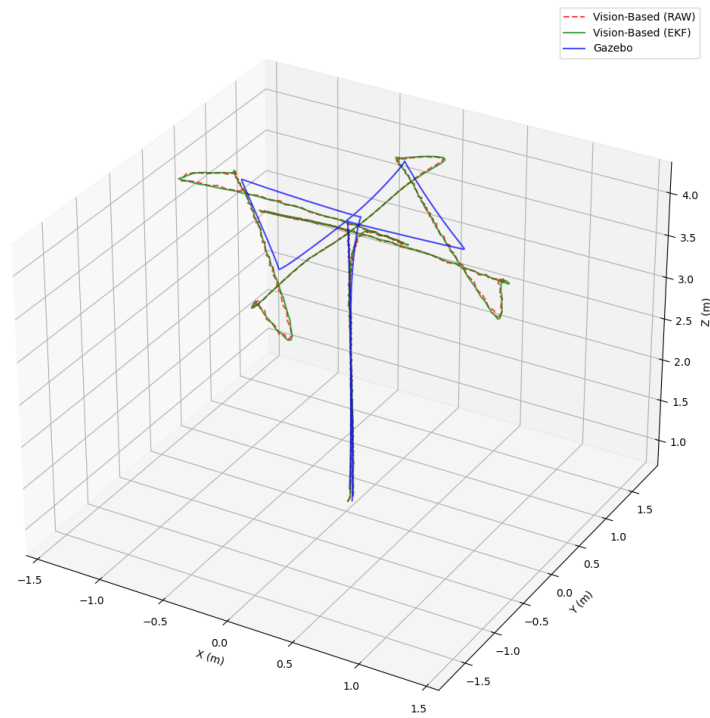


FIGURA D.6: Trajetória 3D estimada (RAW e EKF) comparada com o *ground truth*, 4 m.

D.3 H de 80 cm

Apresentam-se os resultados obtidos para a estimativa de pose durante a execução de uma trajetória em forma de “infinito” a 4 metros de altura, utilizando o marcador H de 80 cm. As Tabelas D.5 e D.6 mostram os valores de erro quadrático médio (RMSE) da posição e orientação estimadas, comparando os dados brutos (RAW) com os filtrados por EKF. As Figuras D.7, D.8 e D.9 ilustram os erros e a trajetória 3D estimada.

TABELA D.5: Erro de posição (RMSE) para trajetória em formato infinito com marcador H de 80 cm.

Eixo	RMSE (RAW) [m]	RMSE (EKF) [m]	Diferença [m]
X	0.215838	0.230284	+0.0144
Y	0.313521	0.323163	+0.0096
Z	0.181782	0.181980	+0.0002

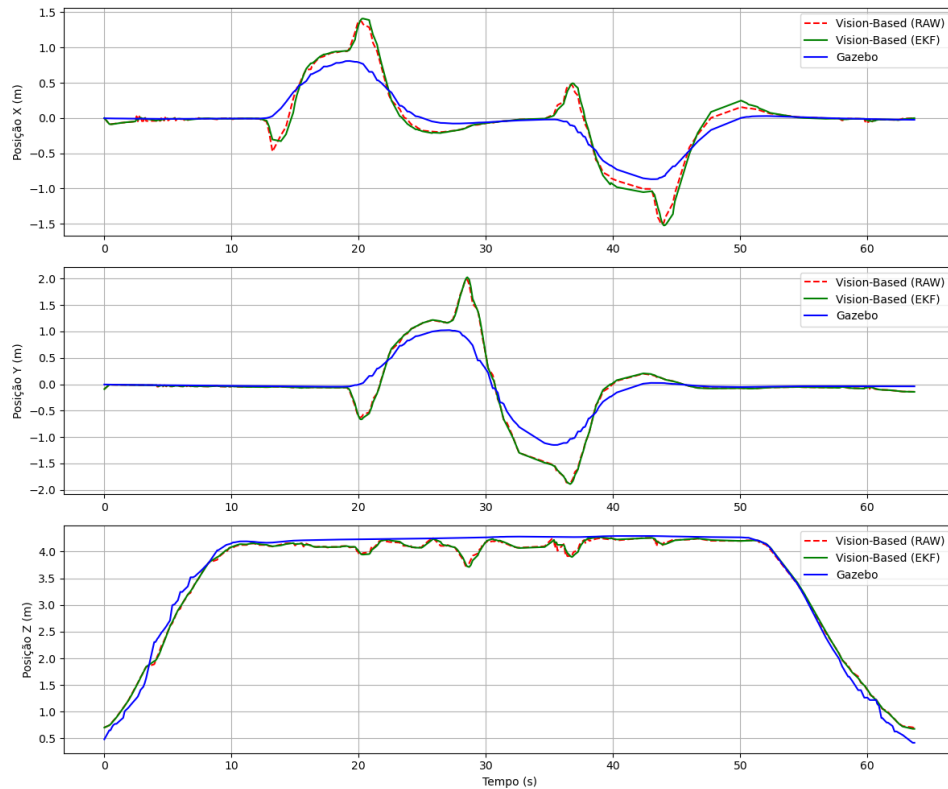


FIGURA D.7: Erro de posição estimada (RAW vs EKF) — Trajetória em formato infinito, 4 m.

TABELA D.6: Erro de orientação (RMSE) para trajetória em formato infinito com marcador H de 80 cm.

Eixo	RMSE (RAW) [°]	RMSE (EKF) [°]	Diferença [°]
Roll	28.10	23.47	-4.63
Pitch	49.22	44.61	-4.62
Yaw	47.51	46.65	-0.86



FIGURA D.8: Erro de orientação estimada (RAW vs EKF) — Trajetória em formato infinito com marcador H de 80 cm.

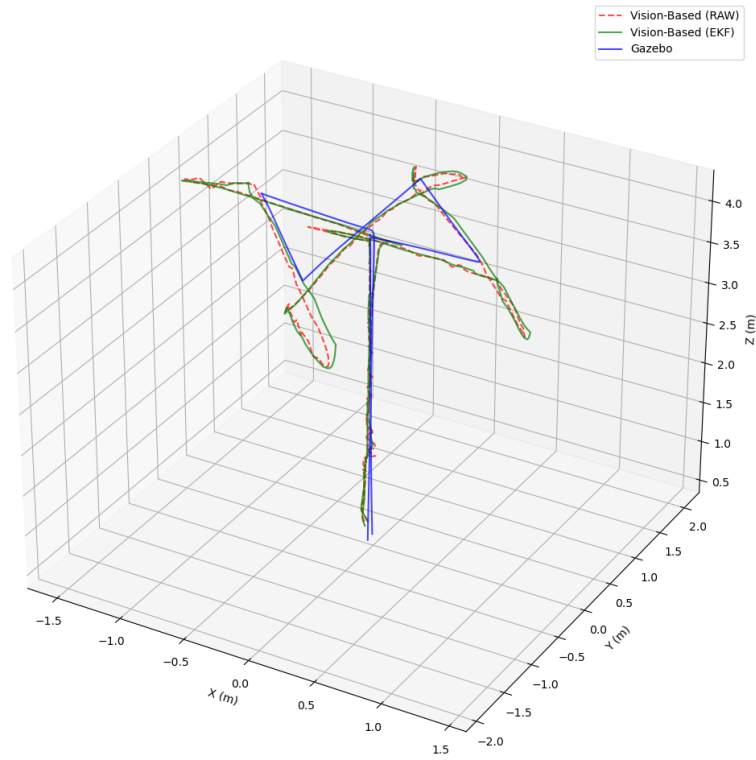


FIGURA D.9: Trajetória 3D estimada (RAW e EKF) comparada com o *ground truth* com marcador H de 80 cm.

Apêndice E - Detalhes Técnicos da Implementação

Este apêndice descreve os aspetos técnicos da implementação do sistema de aterragem automática, tanto em ambiente simulado como em voo real. São apresentados os principais elementos de integração com o ROS 2 e o Gazebo, bem como a adaptação do código para execução direta em Python durante os testes práticos. Adicionalmente, detalham-se os mecanismos de registo e armazenamento de dados utilizados para validação experimental, e a interface de controlo manual que foi desenvolvida para garantir segurança e flexibilidade durante os ensaios.

E.1 Implementação em Simulação

Na simulação, o processo foi implementado como um nó ROS 2 em `rclpy`, com integração no Gazebo e utilização da câmara virtual. Foram utilizados tópicos específicos para transmissão de dados de imagem (`/camera/image`), odometria (`/odometry`) e pose do autopiloto (`/ap/pose/filtered`). O sistema utilizou perfis de *Quality of Service (QoS)* distintos para cada tipo de mensagem, assegurando compatibilidade e robustez na comunicação. A câmara virtual do Gazebo foi configurada para simular as características da SIYI R1M (1080p, 30 fps).

E.2 Implementação em Voo Real

Nos testes reais, a perceção e o controlo foram executados como aplicação Python independente, comunicando diretamente com o ArduPilot via MAVLink. Esta abordagem reduziu a latência, eliminou dependências de middleware e assegurou maior responsividade. A ligação é estabelecida na porta *User Datagram Protocol (UDP)* 14560, configurando automaticamente o modo GUIDED e permitindo comunicação bidirecional com o autopiloto.

E.3 *Logging* e Armazenamento de Dados

O sistema regista em ficheiros *Comma-Separated Values (CSV)* todos os dados relevantes, incluindo pose estimada, *ground truth*, telemetria e comandos enviados. Estes registos permitem análise pós-processamento e validação experimental detalhada.

E.4 Interface de Controlo Manual

Para facilitar os testes experimentais e permitir maior controlo durante a validação, o sistema inclui uma interface de controlo manual por teclado, implementada em conjunto com uma interface gráfica baseada na biblioteca OpenCV. Esta solução assegura uma camada de interação direta com o utilizador, permitindo iniciar, interromper ou encerrar a sequência de aterragem automática de forma controlada e segura. A interface de controlo está mapeada da seguinte forma:

- **Tecla L:** Inicia a sequência de aterragem visual automática, ativando o planeador de trajetória e iniciando a descida controlada;
- **Tecla S:** Interrompe a trajetória em execução, permitindo pausar a aterragem para análise ou correção;
- **Tecla ESC:** Encerra o sistema de forma segura, fechando todas as conexões MAVLink e libertando recursos.

Esta funcionalidade foi particularmente útil em cenários de teste, permitindo validar o comportamento do sistema em diferentes condições e interromper a descida sempre que necessário, garantindo segurança tanto no ambiente simulado como nos ensaios com o VANT real.

Apêndice F - Cálculo Simbólico do Jacobiano no EKF

Considerando que a função de observação $\mathbf{h}(\cdot)$ modela a projeção pinhole de um ponto 3D (x_C, y_C, z_C) na imagem, a matriz Jacobiana $\mathbf{H}_k$ com respeito ao vetor de estado é dada por (Welch, Bishop et al. 1995):

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} = \begin{bmatrix} \frac{\partial u}{\partial x_C} & \frac{\partial u}{\partial y_C} & \frac{\partial u}{\partial z_C} \\ \frac{\partial v}{\partial x_C} & \frac{\partial v}{\partial y_C} & \frac{\partial v}{\partial z_C} \end{bmatrix} \quad (\text{F.1})$$

desenvolvendo cada termo, obtém-se:

$$\mathbf{H}_k = \begin{bmatrix} A + 2\frac{x_C^2}{z_C^2}B & 2\frac{x_C y_C}{z_C^2}B & -\frac{f x_C}{z_C^2}L - 2\frac{x_C(x_C^2 + y_C^2)}{z_C^3}B \\ 2\frac{x_C y_C}{z_C^2}B & A + 2\frac{y_C^2}{z_C^2}B & -\frac{f y_C}{z_C^2}L - 2\frac{y_C(x_C^2 + y_C^2)}{z_C^3}B \end{bmatrix} \quad (\text{F.2})$$

onde:

$$A = \frac{f}{z_C} (1 + k_1 R + k_2 R^2), \quad (\text{F.3})$$

$$B = \frac{f}{z_C} (k_1 + 2k_2 R), \quad (\text{F.4})$$

$$R = \frac{x_C^2 + y_C^2}{z_C^2}, \quad (\text{F.5})$$

$$L = 1 + k_1 R + k_2 R^2. \quad (\text{F.6})$$

Apêndice G - Definição de Avaliação de Desempenho

Neste anexo são apresentadas as principais métricas estatísticas utilizadas para avaliar a qualidade da estimativa de pose do UAV, tanto em simulação como em voo real. Estas métricas comparam os valores estimados p_i com os valores de referência y_i , ao longo de n amostras temporais (Chai e Draxler 2014).

- A medida do erro *Root Mean Square Error* (RMSE), ou Erro Quadrático Médio, quantifica o desvio médio quadrático entre as estimativas e os valores reais:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2} \quad (G.1)$$

- A métrica *Mean Absolute Error* (MAE), ou Erro Médio Absoluto, corresponde à média das diferenças absolutas:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - p_i| \quad (G.2)$$

- O erro médio (*Mean*) representa o desvio sistemático ao longo do tempo:

$$\text{Mean} = \frac{1}{n} \sum_{i=1}^n (y_i - p_i) \quad (G.3)$$

- O desvio padrão (*Standard Deviation*) mede a variabilidade dos erros:

$$\text{STD} = \sqrt{\frac{1}{n} \sum_{i=1}^n ((y_i - p_i) - \text{Mean})^2} \quad (G.4)$$

- O erro máximo (*Max*) representa o maior erro observado durante o intervalo analisado:

$$\text{Max} = \max_i |y_i - p_i| \quad (G.5)$$