

# REVISTA CIENTÍFICA

## ACADEMIA DA FORÇA AÉREA

E não menos por armas que por letras



N.º 15 | Publicação Anual | 2025

# **Revista Científica**

## **Academia da Força Aérea**

---



Publicada anualmente com textos em português e em inglês, a Revista Científica da Academia da Força Aérea é uma publicação da Força Aérea.

Destina-se a divulgar a atividade da Academia da Força Aérea, no domínio da investigação científica, designadamente Teses de Mestrado, Doutoramento e Trabalhos de Final de Curso.

As opiniões expressas nesta revista pertencem aos respetivos autores.

**N.º 15, publicação anual, 2025**





# Ficha Técnica

## Diretor

Comandante da Academia da Força Aérea

Major-General Piloto Aviador Fernando Manuel Lourenço da Costa

## Conselho Editorial

Chefe do Gabinete de Avaliação e Qualidade

Coronel Técnico de Pessoal e Apoio Administrativo Luís Ribeiro

Chefe do Gabinete de Estudos e Planeamento

Coronel Psicóloga Cristina Fachada

Comandante do Corpo de Alunos

Coronel Piloto Aviador José Dias

Diretor da Direção de Ensino da Academia da Força Aérea

Coronel Engenheiro Eletrotécnico Gonçalo Beato de Carvalho

Diretora do Centro de Investigação da Academia da Força Aérea

Tenente-Coronel Técnica de Manutenção de Armamento e Equipamento Paula Gonçalves

## Conselho Consultivo

Tenente-Coronel Técnica de Manutenção de Armamento e Equipamento Paula Gonçalves

Tenente-Coronel Engenheiro Eletrotécnico Ricardo Sousa

Tenente-Coronel Técnico de Operações de Meteorologia Pedro Guerreiro

Major Engenheiro de Aeródromos Nuno Garcez

Major de Administração Aeronáutica João Pina

Capitão Engenheira Aeronáutica Marta Santos

Capitão Engenheira Eletrotécnica Anna Agamyryzansc

---

|                            |                                                                                                                                            |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Propriedade:</b>        | Academia da Força Aérea                                                                                                                    |
| <b>Título:</b>             | Revista Científica da Academia da Força Aérea                                                                                              |
| <b>Periodicidade:</b>      | Anual, Nº 15, setembro de 2025                                                                                                             |
| <b>Composição da Capa:</b> | Sargento-Ajudante Secretariado e Apoio dos Serviços Miguel Duarte Ferreira                                                                 |
| <b>Paginação:</b>          | Tenente Engenheiro Aeronáutico Vasco Coelho<br>Academia da Força Aérea, Força Aérea Portuguesa<br><i>What Colour is this?</i><br>2182-2174 |
| <b>Impressão:</b>          |                                                                                                                                            |
| <b>ISSN:</b>               | 2182-2174                                                                                                                                  |
| <b>Depósito legal:</b>     |                                                                                                                                            |
| <b>Tiragem</b>             | 50 exemplares                                                                                                                              |
| <b>Morada:</b>             | Academia da Força Aérea, Granja do Marquês<br>2715-021 Pêro Pinheiro, Portugal                                                             |
| <b>Correio eletrónico:</b> | gabcmd@academiafa.edu.pt                                                                                                                   |





## Mensagem de Abertura



E com singular agrado que, nesta 15.<sup>a</sup> edição da *Revista Científica da Academia da Força Aérea*, revejo mais uma vez fortemente espelhada a Visão que, no passado dia 21 de maio expressei na minha Diretiva Estratégica 2025-2027 “ ‘Afirmar uma Força Aérea forte, resiliente, inovadora, sustentável, como Instituição de referência’, no País e no Mundo”, e posteriormente em 05 de julho, ecoei no meu discurso na Figueira da Foz, por ocasião da Cerimónia Militar do 73.<sup>o</sup> aniversário da Força Aérea. Em concreto, quando sublinhei os mais elevados padrões de exigência pelos quais a Força Aérea é regida, sua absoluta disponibilidade para responder a todas as solicitações que o país dela necessitar e o facto de que:

investir na inovação é investir no futuro, garantindo [uma] Força Aérea [que] continua a ser uma referência de modernidade, eficiência e prontidão, [e] uma Força Aérea 5.3, voltada para os grandes desafios do século XXI, mas ancorada nos valores que sempre definiram: competência, integridade e serviço à nação.

Em primeiro lugar, e iniciando na forma, pelo facto de oito dos 10 artigos nela publicados estarem redigidos na língua fortemente comungada pela comunidade científica, inglesa.

Em segundo lugar, e mais centrado no conteúdo, em virtude de congregar um conjunto de investigações de elevado rigor metodológico e científico, desenvolvidas por oficiais-alunos das diversas áreas de especialização dos Cursos de Mestrado em Ciências Militares e Aeronáuticas. Investigações/dissertações, igualmente reconhecidas pela sua qualidade e relevância, académica e profissional, aquando das respetivas provas públicas de defesa, e que, pelo somatório dos seus atributos, constituem um valioso acréscimo ao corpo de conhecimento existente, quer no seio da comunidade académica, quer no espaço mais alargado da sociedade contemporânea.

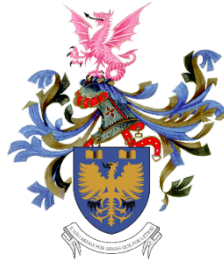
É hoje amplamente consensual que a qualidade da educação e da formação se encontra intrinsecamente ligada ao progresso sustentável das instituições e das nações, constituindo esta realidade a terceira razão da minha elevada satisfação em ver publicados estes 10 estudos. Porque a sua solidez científica e relevância temática atestam, de forma inequívoca, o compromisso contínuo da Academia da Força Aérea com os mais elevados padrões de exigência e competência, repercutido na excelência da formação de quadros altamente qualificados nas áreas das Ciências Militares. E porque são um produto de um trabalho conjunto, orientado para a valorização do conhecimento, para a promoção da investigação aplicada e para o fortalecimento da capacidade intelectual das Forças Armadas.



Enquanto Chefe do Estado-Maior da Força Aérea, cumpre-me, pois, enaltecer e endereçar as mais elevadas felicitações a todos quantos, na Academia da Força Aérea, contribuíram direta ou indiretamente para o êxito de mais esta meritória iniciativa, que prestigia a Instituição e realça o esforço coletivo em prol da ciência, da educação e da defesa nacional.

Aos leitores – académicos, investigadores, decisores, militares, civis, e todos mais que se revêem nas matérias aqui tratadas –, formulo os meus mais sentidos votos de uma leitura profícua, inspiradora, intelectualmente estimulante, profundamente enriquecedora e geradora de novas reflexões no domínio estratégico e científico que estes estudos procuram aclarar.

João Guilherme Rosado Cartaxo Alves, General  
Chefe do Estado-Maior da Força Aérea



## Editorial



A Academia da Força Aérea (AFA) orgulha-se em apresentar a 15.<sup>a</sup> edição da sua *Revista Científica*, dando continuidade ao caminho trilhado nas edições anteriores e mantendo o compromisso de divulgação de trabalhos de investigação, sempre pautados por elevados padrões de exigência, rigor e relevância para o ramo, em particular, e para as Forças Armadas e sociedade académica e civil, em geral.

Num cenário de acelerada evolução tecnológica e transformação operacional, a Força Aérea 5.3 continua a adaptar-se com agilidade e visão estratégica, sustentada nas Pessoas, Processos e Tecnologia. Três pilares para os quais a AFA, como Estabelecimento de Ensino Superior Público Universitário Militar dos futuros Oficiais dos Quadros Permanentes, desempenha um papel essencial, porque primeiramente responsável pela missão de formação de recursos humanos altamente qualificados, preparados para operar sistemas de armas de 5.<sup>a</sup> geração, que possam enfrentar com sucesso os desafios do 5.<sup>o</sup> domínio operacional – o Espaço. Uma missão que exige não apenas excelência técnica, mas também uma sólida formação académica e científica, que

permita aos futuros oficiais compreender, antecipar e responder aos desafios complexos que caracterizam o ambiente operacional contemporâneo.

É, assim, no resultado deste esforço académico que resulta a presente edição, composta por artigos desenvolvidos por finalistas dos cursos de mestrado de diversas especialidades, cuja pertinência e reconhecido valor dos temas abordados contribui para o desenvolvimento do conhecimento científico, com impacto na realidade operacional da Força Aérea e das Forças Armadas. Um conjunto de trabalhos que evidenciam não só o domínio técnico e metodológico dos seus autores, mas também a capacidade para pensar de forma inovadora e propor soluções para problemas reais.

Tal como nas edições anteriores, esta revista é o resultado de um esforço coletivo que envolve alunos, orientadores e todos os que, com dedicação e empenho, tornaram possível a concretização dos projetos de investigação aqui apresentados. A todos, o meu profundo reconhecimento e agradecimento.

Fiéis ao lema da AFA “E não menos por armas que por letras”, convido todos os leitores a explorar esta edição com espírito crítico e curiosidade, certo de que o conhecimento aqui partilhado contribuirá para uma Força Aérea 5.3 cada vez mais preparada para servir a Nação.

Fernando Manuel Lourenço da Costa, Major-General Piloto Aviador  
Comandante da Academia da Força Aérea



# Introdução



A Revista Científica da Academia da Força Aérea (AFA) apresenta, nesta 15.<sup>a</sup> edição anual, dez (10) artigos científicos, tendo estes sido selecionados pela sua pertinência, validade e relevância científica. Os artigos escolhidos foram selecionados entre um total de sessenta e dois (62) trabalhos de investigação, elaborados pelos alunos de Mestrado das especialidades Piloto Aviador, Engenharia Aeronáutica, Engenharia Eletrotécnica, Engenharia de Aeródromos, Administração Aeronáutica e Medicina, nos anos letivos de 2023-2024 e de 2024-2025, contemplando este último apenas a especialidade de Piloto Aviador.

A presente publicação pretende divulgar, em fonte aberta e amplamente acessível, os trabalhos de destaque entre todos os cursos de Mestrado da AFA, representando a diversidade e a qualidade das diversas áreas científicas, assim como o especial contributo para a consolidação das ciências militares aeronáuticas, nomeadamente, nas áreas de comando e controlo em operações aéreas (3), de tecnologias aeronáuticas (2), no domínio aeroespacial (2), áreas de eletrotecnia e telecomunicações (2) e de infraestruturas aeronáuticas (1).

Na área de comando e controlo em operações aéreas são também apresentados três artigos.

O artigo “Detecção Automática de Navios Não Identificados com Recurso a Imagens Satélite” procura desenvolver uma ferramenta capaz de detetar navios em imagens satélite geolocalizadas, através de inteligência artificial, para posteriormente cruzar essa informação com um Sistema de Identificação Automática (AIS). Subsequentemente, caso a fusão dos dados radar com os espaciais obtenha discrepâncias, será gerado um alerta de inconformidade. A validação da fusão com os dados AIS foi testada através de uma ferramenta desenvolvida no ecossistema do Python®, posteriormente validada por simulações de cenários em diversos contextos temporais e espaciais.

O artigo “Implementação de sistema de localização para navegação sem recurso a GNSS” desenvolve um sistema de localização, com o intuito de ser utilizado na navegação de um Unmanned Aircraft System (UAS), de pequenas dimensões e em ambientes onde não exista sinal GNSS. Para o desenvolvimento do sistema de localização compara-se o Light Detection and Ranging (LIDAR) com uma câmara estereoscópica, tendo-se optado pelo LIDAR, como sensor do sistema desenvolvido.

O artigo “User Interface for UAV Object Detection and Geolocation” versa sobre uma aplicação em ambiente web, com o propósito de garantir acesso a dados de geolocalização e a deteção de intrusos em tempo real. Neste sentido, desenvolveu-se um protótipo de um Unmanned Aerial Vehicle (UAV), para duas principais aplicações, a vigilância aérea de áreas restritas e a inspeção de infraestruturas.

Na área de tecnologias aeronáuticas são apresentados dois artigos.

O artigo “Criação de um simulador de voo híbrido, de realidade virtual e mista, e proposta de adaptação do Syllabus de instrução de voo” propõe-se delinear a metodologia subjacente ao desenvolvimento e implementação de um simulador de voo, que utilize a realidade virtual (VR), em simultâneo com a realidade mista (MR). Estuda assim a criação e integração de duas plataformas distintas, uma recuperando a já existente na Esquadra 802, que empregará tecnologia MR, permitindo a visualização real do interior do cockpit e a visualização virtual do exterior e outra, criada de origem, para simulação VR de qualquer aeronave de asa fixa ou rotativa.

O artigo “Design of a Fault Tolerant Active Flutter Suppression System using Model Predictive Control” desenvolve sistemas de controlo adaptativos, para estruturas de asas aeroelásticas flexíveis, abordando o desafio de manter

a estabilidade em asas que, apesar de seus benefícios de eficiência, exibem efeitos aeroelásticos complexos, tais como a vibração. Desenvolveu-se um sistema de Controlo Preditivo de Modelo Adaptativo (AMPC), para que exista um ajuste dinâmico em função das mudanças estruturais e aerodinâmicas, melhorando assim a estabilidade e a tolerância a eventuais falhas.

No âmbito do domínio aeroespacial são apresentados dois artigos.

O artigo “Design and Analysis of the Structure of a Sounding Rocket” tem como objetivo analisar e projetar a estrutura de um foguetão, utilizando propelente sólido, capaz de atingir uma altitude de 100 km, com uma carga útil de 49 kg. No desenvolvimento deste projeto foram utilizados os softwares AutoCAD® e SOLIDWORKS®.

O artigo “Study of the trajectory of the stages of a rocket launched from the Santa Maria spaceport” explora a trajetória ideal para um foguete lançado da ilha de Santa Maria, assim como as potenciais zonas de impacto dos estágios de queda. O cálculo da trajetória ideal envolve várias etapas, sendo a primeira o seu voo vertical.

Na área de eletrotecnia e telecomunicações são também apresentados dois artigos. O artigo “Military Boot Vibration Energy Harvesting” procura desenvolver um sistema de energia vestível, capaz de converter o movimento humano em energia elétrica, fornecendo desta forma uma solução sustentável, para reduzir a dependência de baterias externas e minimizar a carga transportada pelos soldados durante as missões. Utilizando transdutores eletromagnéticos, o sistema captura as vibrações geradas pelos impactos do calcanhar, convertendo-as em energia elétrica útil.

O artigo “Benchmarking Real-Time Fire and Smoke Segmentation in RGB Images for Wildfire Monitoring” desenvolve a captação de registos e consequente criação de base de dados, para utilização no âmbito de combate a incêndios rurais. Esta captação e registo são efetuados de forma eficiente, semiautomática, combinando o modelo YOLOv9®, na deteção de objetos, e o modelo de inteligência artificial Segment Anything Model, na captura de imagens relevantes.

Por último, o artigo “The use of UAV based photogrammetry and BIM in construction supervision. Application to a practical case”, na área de infraestruturas aeronáuticas, procura desenvolver e implementar um projeto, para que a Força Aérea possa reforçar a supervisão da construção e da manutenção, através da adoção de tecnologias avançadas para a supervisão diária. A solução proposta pretende colmatar eventuais lacunas e discrepâncias entre as fases de projeto e de construção de infraestruturas, concentrando-se no desenvolvimento de uma metodologia de monitorização de empreitadas, interligando a fotogrametria realizada por UAV e a Modulação de Informações da Construção (BIM®).

Todos os artigos científicos, com particular relevância para os que são hoje publicados nesta edição, além de constituírem um contributo importante da AFA para o conhecimento científico, no âmbito das Ciências Militares Aeronáuticas, representam também uma forma de valorizar e prestigiar o ensino superior militar, com reconhecimento dos pares, nacionais e estrangeiros.

O mais elevado reconhecimento, nacional e internacional, da qualidade do ensino e da investigação, que a AFA tem sabido desenvolver ao longo da sua existência, como Instituição de Ensino Superior, bem como aquele que resulta das parcerias com entidades homólogas de referência, tais como o IST, o ISEG e a FMUL que, no seu conjunto, conferem grande prestígio, contribuem para o conhecimento científico e fomentam o lema da nossa Academia: “E NÃO MENOS POR ARMAS QUE POR LETRAS”.

Gonçalo Correia Fernandes Beato de Carvalho, Coronel Engenheiro Eletrotécnico  
Diretor de Ensino da Academia da Força Aérea



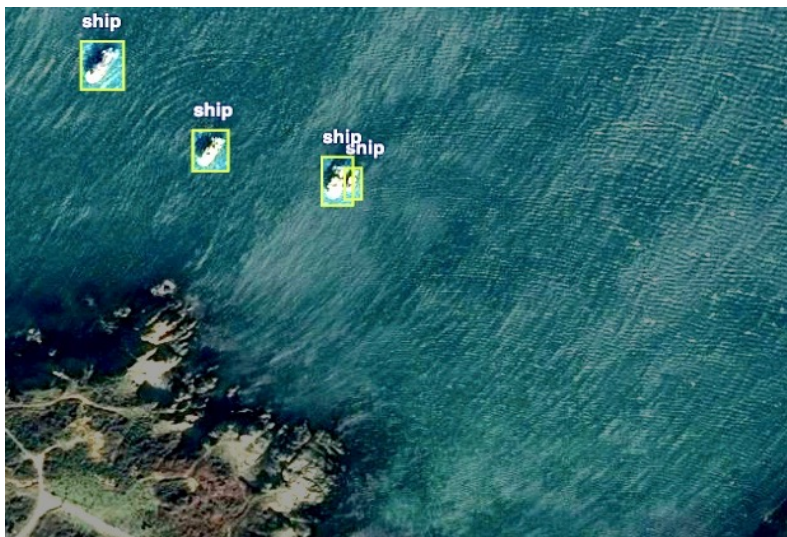
### **Índice**

|                                                                                                                                                |            |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>Deteção Automática de Navios Não Identificados com Recurso a Imagens Satélite</b>                                                           | <b>13</b>  |
| ALF/PILAV Marta Faria Capito                                                                                                                   |            |
| <b>Implementação de sistema de localização para navegação sem recurso a GNSS</b>                                                               | <b>29</b>  |
| ALF/PILAV Miguel Castanhinha Carrasqueira                                                                                                      |            |
| <b>User Interface for UAV Object Detection and Geolocation . . . . .</b>                                                                       | <b>43</b>  |
| ALF/ENGEL Tiago André Martins Vera-Cruz de Abreu                                                                                               |            |
| <b>Criação de um simulador de voo híbrido, de realidade virtual e mista, e proposta de adaptação do Syllabus de instrução de voo . . . . .</b> | <b>59</b>  |
| ALF/PILAV Francisco José Lopes Félix                                                                                                           |            |
| <b>Design of a Fault Tolerant Active Flutter Suppression System using Model Predictive Control . . . . .</b>                                   | <b>75</b>  |
| ALF/ENGEL Francisco Manuel Costa Ribeiro                                                                                                       |            |
| <b>Design and Analysis of the Structure of a Sounding Rocket . . . . .</b>                                                                     | <b>91</b>  |
| ALF/ENGAER Henrique Brás do Carmo Fernandes                                                                                                    |            |
| <b>Study of the trajectory of the stages of a rocket launched from the Santa Maria spaceport . . . . .</b>                                     | <b>107</b> |
| ALF/ENGAER José Miguel Alves Gonçalves Mota                                                                                                    |            |
| <b>Military Boot Vibration Energy Harvesting . . . . .</b>                                                                                     | <b>123</b> |
| ALF/ENGEL Eduardo Filipe Rocha Pinto                                                                                                           |            |
| <b>Benchmarking Real-Time Fire and Smoke Segmentation in RGB Images for Wild-fire Monitoring . . . . .</b>                                     | <b>139</b> |
| ALF/ENGEL Hugo Daniel Sousa Lima                                                                                                               |            |

The use of UAV based photogrammetry and BIM in construction supervision: Application to a practical case . . . . . 153

ALF/ENGAED Maria Catarina Sousa Paulino Libano Monteiro Wemans

# Deteção Automática de Navios Não Identificados com Recurso a Imagens Satélite



Autor: **ALF/PILAV Marta Faria Capito**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

Orientador: **MAJ/ENGEL Gonçalo Charters Santos Cruz**  
Direção de Engenharia e Programas, Alfragide

Coorientador: **CAP/ENGEL Diogo Alexandre Oliveira Silva**  
Academia da Força Aérea, Sintra



## Resumo

A Força Aérea tem como missão participar na defesa militar nacional, integrando a vigilância e patrulhamento marítimo na área sob soberania nacional e interagindo, de forma ativa, na constante monitorização de atividades navais. Neste contexto, a organização tem vindo a incluir um acompanhamento direto de emissões AIS através de fontes abertas, realizada por militares, que analisam se os comportamentos de embarcações são considerados suspeitos. Subsequentemente, as aeronaves designadas ao tipo de missão desempenham um papel crucial ao fornecer suporte à operação no terreno, uma vez que verificam, em tempo real, o *status* do AIS de cada embarcação presente na área e, caso existam alvos específicos, inspecionam detalhadamente a atividade a ser realizada por meios de câmaras integradas na aeronave.

Uma vez que estamos perante uma extensa Zona Económica Exclusiva, que, segundo Cajarabille (2009), contempla 18 vezes a dimensão terrestre portuguesa, torna-se complexo manter uma vigilância contínua e eficaz. Este obstáculo advém da falta de capacidade de um ser humano monitorizar constantemente as fontes de dados, ou das aeronaves cobrirem toda a área marítima com a frequência necessária. Por conseguinte, o objetivo da presente dissertação é o desenvolvimento de uma ferramenta capaz de detetar navios em imagens satélite geolocalizadas através de inteligência artificial e, posteriormente, possua a capacidade de cruzar essa informação com uma base de dados de sinais AIS. Subsequentemente, caso a fusão de dados radar com espaciais obtenha discrepâncias, será gerado um alerta de inconformidade.

A componente de deteção de navios foi implementada com base em *Convolutional Neural Network*, particularmente a versão do 8 da família YOLO. Para a recolha de imagens satélite geolocalizadas optou-se como fonte de dados as imagens obtidas pelas constelações *Sentinel 2*, uma vez que são fontes abertas desenvolvidas pela ESA e retratam imagens do território Europeu, proporcionando a obtenção de uma panóplia de imagens da área de interesse, através do sensor *MultiSpectral Instrument* de 13 bandas espectrais. A validação da fusão com dados AIS foi testada com dados sintéticos. A ferramenta foi desenvolvida no ecossistema do *Python*, e validada por simulações de cenários em diversos contextos temporais e espaciais. Por último, é apresentado um conjunto de resultados dos testes experimentais realizados, comprovando a eficácia do método proposto.

**Palavras-chave:** Patrulhamento marítimo; Inteligência artificial; Deteção de navios; Imagens satélite.

## 1 Introdução

A geoestratégia de Portugal destaca-se como uma mais valia à vista da defesa, turismo e economia, redobrando a importância do espaço marítimo (Cunha, 2004). A abundância de recursos que este meio direciona para Portugal reflete-se num culminar de fatores indispensáveis a ter em consideração aquando fins civis ou militares. Neste sentido, com o incito de interesses pelos bens provenientes do mar, surgiu a necessidade de um controlo exaustivo aos eventos nele praticados, uma vez que estes recursos apresentam um caráter limitativo (Carvalho, 1982). Ora, sendo este controlo efetuado maioritariamente pelos militares, torna-se notório que as múltiplas atividades desempenhadas atualmente, assim como as numerosas rotas marítimas que cruzam o espaço naval português, aumentaram a complexidade da missão das Forças Armadas, sendo necessária uma atenção redobrada ao tráfico de estupearfacientes, migrantes, entre outros (Silveira, 2019).

Neste âmbito, a Força Aérea Portuguesa, fora os seus compromissos perante a defesa do espaço aéreo português, assume-se como um dos protagonistas principais pela busca e salvamento, vigilância, reconhecimento e patrulhamento naval, ainda que a marinha portuguesa também represente igualmente presença nestas missões, embora sejam independentes uma da outra (Dias & do Mestrado Integrado, 2014). Neste sentido, a dita cuja área de responsabilidade é assinalada como Região de Busca e Salvamento (SRR), que, até ao presente momento, coincide a nível espacial com a *Flight Information Region* de Lisboa e Santa Maria (Maia, 2021). Definido pela Convenção *Search and Rescue* de 1979, aquando a divisão das 13 áreas de busca e salvamento, e, posteriormente revisto, Portugal é responsável por uma área que se estende 5.754.848 km<sup>2</sup>, cuja dimensão supera 18 vezes a dimensão da área terrestre portuguesa (Cajarabille, 2009) (Branco, 2022) (DGRM, n.d.).

Seguindo uma linha de pensamento direcionada para a realidade da Força Aérea, a falta de pessoal sentida no ramo afeta todas as áreas e, o Centro de Fusão de Informação, não é exceção. A incorporação da inteligência artificial seria vista então como uma ferramenta de apoio à análise e decisão, permitindo a cobertura de uma maior área, em menos tempo,

e libertando os operadores para tarefas *high-level*. Atualmente, a inteligência artificial é uma ferramenta acessível que permite uma contínua operacionalidade eficiente. Como tal, através do desenvolvimento e evolução destas tecnologias, com especial ênfase na identificação e deteção de objetos através de imagens trabalhadas, foi possível facilitar a recolha e leitura de dados, tratando e analisando o objeto predefinido que é procurado (dos Santos Quinta Albuquerque, 2022).

## 1.1 Formulação do Problema

O objetivo deste trabalho remete para o desenvolvimento de um sistema que possua a aptidão de detetar navios em imagens de satélite e, posteriormente, tendo em atenção o *delay* do satélite, seja capaz de cruzar essa informação com bases de dados de sinais de AIS. Esta ferramenta tem em vista, como objetivo final, alertar as áreas onde foram feitas deteções, sem a existência de um sinal AIS correspondente. Este objetivo será concretizado tendo em conta o esquema de ações descrito seguidamente: primeiramente seleciona-se as fontes abertas de imagens satélite e a recolha de múltiplas imagens; de seguida, procede-se à escolha de fontes abertas de dados AIS e à recolha destes dados; Por último, segue-se o desenvolvimento de uma ferramenta capaz de detetar e classificar navios, cruzar informação com os dados AIS recolhidos e alerta de não correspondência.

## 2 Estado de Arte

No contexto marítimo, os instrumentos obrigatórios a bordo, como sistemas de navegação e comunicação, desempenham um papel essencial para a segurança. O Sistema de Identificação Automática (AIS) é uma dessas ferramentas, permitindo um *tracking* e *situational awareness* de embarcações. No entanto, nem todos os navios utilizam ou estão equipados com AIS, criando desafios na vigilância. Tal como referenciado no Decreto de Lei nº 180/2004, de 27 de Julho a notificação e identificação de um navio é obrigatória sempre que esteja implementado num espaço marítimo sob jurisdição nacional, sempre que os navios se dirijam a um porto nacional e para quaisquer embarcações de pesca com um comprimento de fora a fora superior a 15 metros, salvo exceções.

As imagens satélite surgem como uma solução importante para monitorizar áreas vastas como a nossa ZEE, complementando a falta de dados AIS. Através da deteção de objetos em imagens, é possível identificar e classificar embarcações, mesmo em zonas remotas. Tecnologias como inteligência artificial tornam essa análise mais eficiente, sendo um auxílio no patrulhamento, busca e salvamento e combate a atividades ilícitas.

### 2.1 Automatic Identification System

O *automatic identification system* é um sistema de transmissão *Very High Frequency*, dividido em dois canais, o primeiro *ship-to-ship* (161.975 MHz) e o segundo *ship-to-shore* (162.025 MHz), que permite a recessão e transmissão de identificação/informação dos navios, tanto entre navios como de navios para estações de controlo, ou ainda de navio para satélite. Este sistema obrigatório deu origem a inúmeras aplicações, nomeadamente, a uma análise tanto prévia como subsequente de uma ocorrência, uma vez que, segundo o Decreto de Lei nº 180/2004, de 27 de Julho, artigo 6º-A, alínea 2, as embarcações que possuem o AIS têm a obrigação de o manter ligado e completamente operacional quando operam sob águas de jurisdição nacional (Lee, Mokashi, Moon, & Kim, 2019).

Neste sentido, para cada uma das embarcações é lhes atribuído um número *Maritime Mobile Satellite Identity*. Este é um número único para cada navio que equivale a um conjunto de 9 dígitos, compreendido entre 201000000 e 775999999. O facto deste número ser único para cada embarcação permite às autoridades verificarem se o número introduzido no sistema corresponde aos dados reais (Tunaley, 2013).

### 2.2 Imagens Satélite

As imagens satélite têm vindo a ser amplamente utilizadas, nomeadamente no que toca à manipulação de imagens pois surgem como uma potencial fonte de informação que dificilmente oculta dados de interesse propositadamente. As imagens satélite demonstram uma forte capacidade de resolução espectral aquando a identificação de navios, essencialmente pela preferência destes de usarem materiais de aço, alumínio e madeira. Não obstante, estas estão sujeitas às condições

climáticas e reflexos provenientes do sol, que deterioram significativamente a resolução da imagem e podem criar falsos alarmes (Kanjir, Greidanus, & Oštir, 2018).

A ingestão deste tipo de fonte de informação no meio educacional tem vindo a ser visto como uma ferramenta didática, aplicável a um vasto leque de temas tanto no meio terrestre como marítimo, que contempla uma variedade de características temporais, espaciais e técnicas (Crispim & Albano, 2016). Sendo o patrulhamento marítimo uma das missões primárias executada pelas esquadras da Força Aérea, são naturalmente exigidos sistemas capazes de abranger uma vasta área de cobertura marítima e de forma sistemática e económica, incluindo ambientes com interferência ou ruído. Deste modo, as imagens satélite surgem como uma fonte espacial de vigilância marítima superior aos meios terrestres e aéreos, que através da sua flexibilidade e automatização, conseguem recolher imagens fidedignas de uma extensa área de interesse como é o espaço marítimo português, sendo possível aceder via fontes abertas como o *Copernicus Data Space Ecosystem* e *SentinelHub* (Falqueto, Paes, & Passaro, 2021).

### 2.3 Detecção de Objetos

Ainda que os sistemas AIS auxiliem no patrulhamento e vigilância marítima, surge um obstáculo quando as embarcações navegam com os sistemas de identificação desligados, tornando-se indetetáveis aos radares e sistemas de controlo, uma vez que existe a limitação da linha de vista por parte das transmissões e recessões radar. Neste âmbito, e através do uso de imagens satélite, torna-se possível voltar a identificar estas embarcações quando são detetadas nas fotografias. A deteção de navios é feita, nomeadamente, através do contraste entre a cor deste e o fundo marítimo, ou, como descreve a literatura, encontrar um ponto branco num fundo preto (Kanjir et al., 2018). Esta deteção em específico, pode ser obtida essencialmente através de técnicas como *machine learning* e *deep learning* (Kirbas & Quek, 2004).

#### 2.3.1 Machine Learning

A técnica de *machine learning* é um método avançado de inteligência artificial amplamente utilizado na deteção de objetos. Neste contexto, a ferramenta vai aprendendo a funcionar de forma autónoma, prevendo comportamentos através de padrões extraídos dos dados. Este método aprimora os algoritmos iniciais, automatizando a correção dos modelos através de três formas diferentes: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço (Mahesh, 2020).

Embora muitos destes métodos estejam amplamente desenvolvidos nos dias de hoje, tornam-se complexos de pôr em prática perante tarefas que implicam grandes volumes de dados não estruturados ou padrões de dificuldade extravagante. É neste sentido que se destacam as redes neuronais artificiais, um dos métodos de aprendizagem supervisionada, uma vez que possuem uma maior capacidade de resposta perante cenários de maior complexidade (Mishra & Srivastava, 2014). O potencial de uma rede neuronal reside no ajuste sistemático dos seus parâmetros, baseado na diferença entre os valores desejados e os obtidos. Os erros são reintroduzidos no sistema, ajustando os parâmetros iniciais até alcançar uma performance satisfatória (Dongare, Kharde, Kachare, et al., 2012).

#### 2.3.2 Deep Learning

O *deep learning* está integrado na técnica *machine learning* e, tal como o nome indica, é uma técnica de inteligência artificial mais aprofundada que possui uma elevada capacidade de interpretação de dados, sejam estes mais complexos e/ou numerosos, assim como a receção e interpretação de informação proveniente de diversas fontes e formatos. A criação do *deep learning* está amplamente associada à superação de limitações impostas pelo *machine learning*, nomeadamente através do desenvolvimento e aplicação de redes neuronais profundas, ainda que existam métodos que não contemplem a integração destas (Ibrahim et al., 2020).

Uma rede neuronal profunda, em muito é semelhante às artificiais, uma vez que também aprende a melhorar a performance através do treino e da reintrodução dos erros, essencialmente, através do método de *back propagation* (Cichy & Kaiser, 2019). A diferença entre redes neuronais artificiais e profundas, equivale ao número de camadas ocultas que constituem a rede neuronal e, como tal, à medida que adicionamos camadas ocultas, a rede passa a ser profunda. Em cada camada, o *input* da camada de entrada sofre uma transformação geométrica, que preserva as linhas e paralelismos

mas não as distâncias e ângulos, e uma transformação não linear, que enaltece específicas características (Liu et al., 2021).

Os modelos de *deep learning* dividem-se em métodos de *two-stage approaches* e *one-stage approaches*. No primeiro método estamos perante uma rede dividida em duas fases: a primeira foca-se em gerar propostas de áreas de interesse ou possíveis objetos; e, a segunda, concentra-se na classificação e deteção através de caixas delimitadoras, como é o caso do modelo YOLO (Sultana, Sufian, & Dutta, 2020).

### 2.3.3 YOLO

Neste âmbito, os modelos de deteção baseados em *one-stage*, nomeadamente o *You Only Look Once* (YOLO), foram desenvolvidos atendendo ao problema de deteção de objetos com uma performance mais rápida e eficaz. Desde 2015 até 2023 já foram apresentadas 8 versões díspares do modelo YOLO, com a mais recente lançada em 2023, pela *Ultralytics*, tornando possível tirar proveito de vários sistemas de treino de imagens por serem conciliáveis com a maioria das variantes do YOLOv8 (Terven, Córdova-Esparza, & Romero-González, 2023).

Este modelo apresenta uma arquitetura *CSPDarknet53*, semelhante à versão YOLOv5 (Hussain, 2024). O YOLOv8 tem como base uma rede *Cross Stage Partial Network* (CSPNet), que se define como uma arquitetura de alta eficiência a partir da propagação do fluxo de gradiente e subdivisão do mapa de características. A componente de Darknet53 é uma especificidade de redes neuronais baseadas em CSPNet, que permite um melhor desempenho na deteção de objetos e aplicação direta em modelos como o YOLO. Esta metodologia permite evitar o aumento da capacidade de computação e a redução de precisão (Li et al., 2024; Wang et al., 2020).

Algumas características adicionais do YOLOv8 são o módulo utilizado, as âncoras e as funções. O YOLOv8 utiliza um módulo *cross-stage partial bottleneck* com duas convoluções, que combina as características de alto nível com informações contextuais, o que permite uma melhor exatidão nas deteções realizadas. Nesta versão, estamos perante um modelo sem âncora com ramos desacoplados, cujas funções de objetividade, regressão e classificação encontram-se independentes umas das outras por forma a garantir foco numa única tarefa e, no geral, assegurar uma melhor performance. A nível de funções, o YOLOv8 utiliza maioritariamente as funções de perda *Varifocal Loss* e a combinação entre *Distribution Focal Loss* e *Complete Intersection over Union*, que calculam a perda das caixas e da classificação. As funções de ativação do YOLO vão regular os pesos do modelo consoante o *feedback* das funções de perda. As funções aplicadas variam consoante o objetivo final, sendo as mais comuns a sigmóide e *softmax* (Terven et al., 2023).

O YOLO para além de se enquadrar no âmbito desta dissertação, também introduz outras vantagens. Estas revertem-se na possível aplicação do modelo em plataformas que não exigem programação e funcionam totalmente de forma visual, como o *Roboflow*, uma ferramenta de *no-code*, ou ainda, para plataformas de *low-code*, onde é exigido uma base de código reduzida e básica.

### 2.3.4 Deteção em Imagens Satélite

As imagens satélite são altamente ricas em detalhes pormenorizados o que as viabiliza como um alvo interessante para deteções automatizadas. Não obstante, esta fonte traz adversidades que não são comuns noutros tipos de fontes, como variabilidade nas condições de iluminação, obstáculos meteorológicos e resolução espacial. Estes contratempos afetam em muito a performance dos modelos de deteção, e, como tal, vê-se a necessidade de aplicar redes profundas devido ao elevado grau de aprendizagem sem a necessidade de extrair manualmente as características (Tahir et al., 2022).

Adicionalmente, os objetos a detetar são de tamanhos reduzidos pelo que induzem o modelo de deteção em erro mais facilmente. Objetos como carros e navios conseguem ter aproximadamente 15 pixels em imagens de alta resolução, direcionados de forma variável. Posto isto, a deteção de objetos em imagens satélite é um problema extremamente relevante uma vez, fora as condições climáticas adversas, os objetos encontram-se densamente agrupados, de tamanho reduzido e multi-orientados (Shermeyer & Van Etten, 2019).

### 2.3.5 Métricas de Avaliação

As métricas de avaliação determinam a capacidade de um modelo identificar e localizar objetos nas imagens fornecidas. É através destas que nos é possível entender como é que o modelo está a abordar falsos positivos e falsos negativos.

Em antemão, é essencial dar a conhecer alguns conceitos presentes na matriz de confusão, que nos permitem calcular os parâmetros de problemas de classificação binária, tais como, *accuracy*, *Precision*, Recall, Pontuação-F1, *Average Precision* e *mean Average Precision* (Casas, Ismail, Limeira, da Silva, & Leite, 2023).

A *Precision* avalia a capacidade de o modelo evitar falsos positivos, quantificando a percentagem de Verdadeiros Positivos entre todas as deteções positivas existentes. Este valor é quantificado através da razão entre Verdadeiros Positivos e a soma entre Verdadeiros Positivos e Falsos Positivos, tal como presente na equação 1.

$$Precision = \frac{VP}{VP + FP} \quad (1)$$

O *Recall* mede a aptidão do modelo para detetar corretamente um objeto. Esta métrica é calculada pela razão entre os Verdadeiros Positivos e todos os positivos reais presentes no modelo, sendo o segundo representado pela soma de Verdadeiros Positivos e Falsos Negativos, exibido na equação 2 (Sapkota et al., 2024).

$$Recall = \frac{VP}{VP + FN} \quad (2)$$

A pontuação-F1 é uma média ponderada entre a *Precision* e o *Recall*. A curva relativa a esta pontuação dá-nos um ponto de situação do equilíbrio do modelo em relação a falsos positivos e falsos negativos. Por sua vez, a *Average Precision* corresponde à área sob a curva descrita anteriormente e pode ser calculada através da equação 3. Deste modo, esta sumariza a curva *Precision-Recall* a um valor escalar entre [0,1] e representa a capacidade do modelo de classificar corretamente as amostras positivas, sem classificar muitas amostras negativas como positivas.

$$AP = \int_0^1 P(r) dr \quad (3)$$

### 3 Metodologia

A organização desta ferramenta encontra-se faseada por etapas sequenciais, onde a conclusão de cada passo facultada dados para o passo seguinte, gerando uma linha de continuação ao longo de todo o sistema. Descrevendo, de forma sucinta, o processo pelo qual o sistema será desenvolvido, o primeiro passo resume-se à receção de dados que determinam o escopo para todas as etapas restantes. É nesta fase que se circunscreve parâmetros como a área de interesse, espaço temporal, entre outros critérios relevantes a delinear para o *input* inicial.

De seguida, segue-se a obtenção de imagens satélite de acordo com o delineado previamente e, atendendo a critérios essenciais, como a resolução espacial. Após a obtenção de imagens, o sistema irá proceder à deteção de objetos, através do uso de redes neurais profundas. No âmbito desta dissertação, a geolocalização no geral desempenha um papel fundamental, tornando essencial estabelecer uma ligação programática entres as localizações das imagens extraídas e as deteções realizadas pelo modelo. Deste modo, o objetivo passa por associar estas deteções às respetivas coordenadas geoespaciais, permitindo validar o posicionamento em concreto do objeto detetado.

Os critérios para a escolha de fontes de dados AIS são bastante simples, sendo que o principal foca-se num formato que seja compatível com as geolocalizações das deteções das imagens extraídas, por forma a possibilitar um cruzamento de dados. Adicionalmente, pretende-se que esta fonte contenha a localização individual de cada embarcação, acompanhada da identificação *standard* AIS. Deste modo, propõem-se a criação de dados AIS sinteticamente que advêm diretamente de uma correspondência parcial às geolocalizações das deteções anteriormente feitas. Este ficheiro seria obrigatoriamente compatível entre ambas as fontes de informação, uma vez que são provenientes do mesmo meio, e, embora não sejam dados reais, exemplificam o normal funcionamento da ferramenta através de uma simulação. A fase final do sistema reflete-se no cruzamento de dados que contempla um alerta de não correspondência entre os dados AIS e as deteções das imagens satélite, originando um detetor de navios não identificados.

## 4 Implementação

### 4.1 Extração de Dados

#### 4.1.1 Input Inicial

O ponto inicial desta ferramenta foca-se em extrair as imagens satélite de acordo com a localização determinada. Neste sentido, as imagens que serão obtidas terão uma coordenada determinada como canto superior esquerdo e as restantes serão extraídas até completar uma área definida previamente. O objetivo deste método é garantir que as imagens são todas retiradas continuamente em relação a uma área de interesse, sem sobreposição. Adicionalmente, é nesta fase inicial que também se determina o intervalo temporal à qual se irá realizar a extração de imagens satélite. A definição destas características garante que a ferramenta trabalhe com imagens contextualizadas a nível temporal e espacial, otimizando os resultados da deteção.

#### 4.1.2 Fonte de Imagens Satélite

Para a obtenção de imagens satélite, inicialmente tentou-se compreender as opções em causa, que se dividem entre recolher bases de dados completos, ou extrair imagens programaticamente. Os critérios para esta base de dados incidem sobre a necessidade de obter imagens de zonas costeiras e/ou rotas marítimas acompanhadas de um ficheiro compatível com as geolocalizações correspondentes a cada imagem. Após a análise de vários catálogos, conclui-se que a zona pretendida nunca era a área de maior foco, acabando por atrair muitos dados desnecessários que poderiam possivelmente gerar falsos alarmes na ferramenta. Para além desta adversidade, muitos destes catálogos não contemplavam a carência de um ficheiro a geolocalizar as imagens devidas. Deste modo, ao invés de se tentar obter um catálogo completo, propõe-se a recolha de imagens programaticamente, definindo manualmente a data, área, cobertura de nuvens, satélite e resolução, sendo a fonte com maior qualidade de imagem, o *SentinelHub*.

#### 4.1.3 Método de Recolha de Imagens

Estabelecida a fonte de extração e a sua resolução, determinou-se que a recolha de imagens satélite seguiria os seguintes passos a ser descritos. Em primeira instância delinear uma área de parametrizável e, posteriormente, dividir essa área em subáreas de uma dimensão compatível com o maior tamanho extraível pela API do *SentinelHub* e evitando sobreposições. O passo seguinte foca-se em extrair imagens de todas as subáreas para o mesmo dia, até a área estar completa. Por último, o objetivo é realizar o passo anterior de forma contínua para um determinado período de tempo, a fim de obter um *track* seguido do tráfego marítimo, tendo em consideração que a ferramenta só extrai imagens do dia seguinte após completar o mosaico do dia em questão. De realçar que o satélite em causa não possui cobertura das áreas para todos os dias, pelo que, para os dias em que não existe captura de imagens, o *SentinelHub* retribui imagens a preto. Neste caso, para imagens sem conteúdo, a ferramenta descarta-as automaticamente.

### 4.2 Deteção de Navios

#### 4.2.1 Dataset e Modelo de Deteção

Após uma revisão de literatura, acredita-se que o uso de modelos de *deep learning one-stage* serão os mais apropriados a aplicar para a deteção de navios. Esta opção deve-se ao facto de serem métodos caracterizados pela elevada performance de deteção rápida em tempo real, não exigindo excessivamente de competências computacionais. Outros motivos que também proporcionaram a escolha revertem-se, nomeadamente, na compatibilidade com outras ferramentas de *low code*, onde seria possível treinar o modelo de deteção, assim como a eliminação de outras características que não trariam nenhum benefício à sua aplicação, como a classificação, uma vez que a única classe incutida serão os navios no geral.

Para efeitos de treino do modelo de deteção procedeu-se à escolha do *dataset* e ao treino deste. A escolha esteve perante 3 *datasets* distintos em todos os parâmetros, sejam estes a fonte da imagem, resolução, dimensões de objetos e obstáculos presentes. Após o treino de ambos na plataforma *Roboflow*, verificou-se a performance dos modelos através

das métricas de avaliação e testes com imagens satélite que não estão contempladas nos treinos do modelo, constatando que o modelo correspondente ao *dataset* da *Airbus* apresentava uma melhor performance em todos os parâmetros.

### 4.2.2 Treino e Performance

Em relação ao *dataset* da *Airbus*, para além de se tratar de um conjunto de dados facilmente acessível por ser público, a utilização de um número substancialmente maior de imagens originou uma base de dados com maior variedade no grau de aproximação do objeto e tipo de imagem utilizado, dando azo a um processo mais robusto e eficaz. Depois de destacado o modelo que contém o *dataset* da *Airbus* pelos motivos anteriormente mencionados, procedeu-se ao treino deste modelo no *Jupyter*, com a biblioteca *Ultralytics* (Jocher, Chaurasia, & Qiu, 2023), através de métodos de *low-code*, o que permitiu verificar uma melhoria em determinadas métricas.

Após o treino final do modelo de deteção, deu-se início aos testes de performance. Uma das conclusões mais importantes reflete-se no impacto da marca de água. Embora esta marca esteja presente num tamanho relativamente pequeno, acaba por confundir o modelo em algumas das imagens. Deste modo, considerou-se que um dos primeiros passos após a extração das imagens, seria esconder a marca de água no canto inferior esquerdo e apenas posteriormente dar-se a deteção, tal como se verifica na imagem 1. Após diversos testes realizados na mesma zona, com apenas alterações de datas e *thresholds*, verificou-se que o modelo não consta mais deteções no local da marca de água, concentrando-se melhor na deteção de embarcações.



Figura 1: Processo de deteção.

### 4.3 Geolocalização das Embarcações

Após a extração e deteção de navios nas imagens satélite, procede-se à identificação da posição geográfica de cada embarcação. Esta posição é obtida através da correlação entre os dados das fotografias e os das deteções. O cerne da questão é garantir esta compatibilidade programática entre os dois dados, de modo a gerar um ficheiro que contém exclusivamente as localizações de cada deteção associada à imagem correspondente.

Deste modo, o processo consiste num mapeamento das coordenadas das deteções feitas em pixels. Cada imagem satélite é submetida a um sistema de referência geográfico que relaciona cada pixel com uma posição real no globo terrestre. Após a deteção, cria-se uma *bounding box* com as coordenadas em pixels, delimitando o objeto detetado e, posteriormente, associa-se cada limite da caixa de deteção a uma coordenada geográfica. Assim, cada uma das deteções passa a representar uma posição no espaço real.

### 4.4 Dados AIS

Desde o início do desenvolvimento desta ferramenta que as opções foram bastante limitadas, uma vez que foi definido que um dos principais critérios passava por ter um acesso programático gratuito aos dados AIS. Apesar de existirem inúmeras fontes de dados AIS *online* com atualizações em tempo real, o acesso a tal informação via programática exige

um plano pago, quer sejam estes dados atualizados ou históricos, impedindo assim o uso das informações por este meio. Posto isto, tentou-se obter um registo antigo de bases de dados da Força Aérea, não obtendo uma resposta positiva devido à falta de capacidade de extração de amostras e de acessibilidade de forma programática a esta fonte.

Dito isto, foi proposta uma criação sintética de dados AIS, sendo que determinou-se que a sua origem provinha dos dados já obtidos nas deteções das imagens satélite. Este processo tem como fim simular a presença de embarcações não identificadas no contexto desta dissertação, uma vez que não foi possível aceder a informação real. Deste modo, após a recolha das localizações de cada deteção, procede-se a uma amostragem aleatória, incluindo apenas uma determinada percentagem dos dados obtidos. A escolha desta percentagem não implica que esta seja uma representação fidedigna de um cenário real.

Os restantes dados correspondem a embarcações que, por motivos diversos, não estão visíveis para este sistema de monitorização. Estes casos podem ser devidos a embarcações que propositadamente desligaram a emissão *transponder* AIS ou que a sua comunicação não foi rececionada, não coincidindo com a informação da deteção nas imagens satélite, resultando numa discrepância de dados.

### 4.5 Fusão de Deteções e AIS

O cruzamento de dados tem como objetivo notificar discrepâncias de geolocalizações relativas a navios não identificados. Após a obtenção de ambas as geolocalizações, tanto das deteções como dos sinais AIS, os ficheiros são comparados programaticamente e, de seguida, é criado um novo ficheiro que contém exclusivamente os dados relativos à falta de compatibilidade.

Deste modo, o ficheiro resultante contém os detalhes referentes às embarcações não devidamente identificadas incluindo a posição geográfica, imagem correspondente, dia da obtenção da imagem satélite e coordenadas da *bounding box* da deteção, tanto em pixels como geográficas. O propósito deste novo ficheiro é devolver as imagens satélite originais com novas caixas delimitadoras azuis, a destacar o navio não corretamente identificado.

### 4.6 Análise de Ruído de Emissão

Embora o AIS seja amplamente reconhecido pela sua elevada precisão em condições ideais, apresenta limitações significativas em cenários adversos. Entre as principais desvantagens estão a possibilidade de desligar intencionalmente o sinal, interferências de rádio, atenuação do sinal e outros fatores ambientais que comprometem a transmissão correta (Wolsing, Roeper, Bauer, & Wehrle, 2022). Em acréscimo, o movimento do navio, nomeadamente a altas velocidades, pode gerar discrepâncias entre o posicionamento transmitido e o real. O posicionamento da antena de transmissão, especialmente em embarcações de grande comprimento, também pode causar desvios, uma vez que nem sempre reflete o centro da embarcação. Adicionalmente, atrasos de transmissão típicos dos sistemas de comunicação podem agravar essas imprecisões (Xiao, Yang, Xu, Li, & Jiang, 2024).

A inclusão deste subcapítulo deve-se ao facto da necessidade de prever uma possível resolução para mitigar este problema de desvios de geolocalização AIS, ainda que não esteja implementado na ferramenta tendo em conta a natureza dos dados AIS utilizados, uma vez que são sintéticos e baseados nas próprias deteções. De acordo com a literatura, a discrepância máxima de localização geográfica pode atingir os 100 metros em situações de tempestades ou outros fatores externos excessivos. Como tal, considera-se, para efeitos de certeza, um desvio máximo de 150 metros como critério, implementando o seguimento dos passos a ser descritos:

1. Conversão das Coordenadas da *Bounding Box*: Em primeira instância, para efeitos de compatibilidade, converter as coordenadas das deteções para coordenadas GPS, assumindo a coordenada central da caixa delimitadora como a posição geográfica aproximada do navio;
2. Definição de um Raio de Confiança: Considerar um raio de 150 metros a partir de cada coordenada GPS, representando uma área de localização estimada do navio;
3. Comparação entre Coordenadas: Realizar uma comparação entre cada coordenada do ficheiro de deteções com cada coordenada do ficheiro AIS, utilizando o raio definido para verificação de compatibilidades;



4. Separação de Embarcações Distintas: Em casos de múltiplas embarcações presentes no mesmo raio, fazer uso do número MMSI para as diferenciar, prevenindo que sejam consideradas como apenas uma.

Através deste método, é possível minimizar efeitos de desvios inerentes à transmissão do sistema AIS, aumentando a viabilidade da ferramenta na correspondência de detecções via satélite e dados AIS, e, evitando inúmeros falsos alarmes, especialmente em cenários adversos.

## 5 Teste Experimental e Resultados

A arquitetura deste sistema segue as etapas pormenorizadas na figura 2. A ordem deste sistema tem em vista procurar nas imagens satélite, embarcações que emitam um sinal AIS, e não o oposto. A ferramenta desenvolvida tem como requisito um funcionamento contínuo que contemple todas as áreas previamente definidas. O desempenho esperado implica uma recolha de imagens individual em cada subárea para todos os dias contemplados no intervalo de tempo estabelecido e, posteriormente, todo o processo de detecção, geolocalização e resultados de dados que não entrem em concordância.

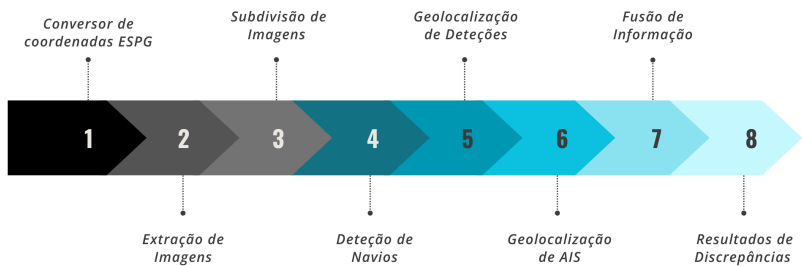


Figura 2: Arquitetura do Sistema.

### 5.1 Teste Experimental

#### 5.1.1 Extração e Subdivisão de Imagens

A nível de praticabilidade, determinou-se que as coordenadas não seriam introduzidas manualmente para a extração de imagens, tornando a ferramenta mais eficiente e menos propensa a erros de fator humano. Deste modo, recorreu-se ao uso de um mapa interativo, onde o utilizador selecciona diretamente no mapa a área de interesse, e, de seguida, as coordenadas são diretamente transferidas para a ferramenta, dando início a todo o processo. Deste modo, os utilizadores não precisam de recorrer a outros sistemas para encontrar a coordenada exata do pretendido e têm um *feedback* visual imediato, podendo confirmar no momento se o local em questão corresponde com o pretendido.

As imagens foram retiradas da plataforma *SentinelHub* com uma resolução de 10 metros por pixel. O teste experimental ocorreu em três áreas localizadas ao longo da zona costeira espanhola e intersectando rotas marítimas, com vista a diversificar o tipo de fundo à qual a deteção de objetos será imposta. As três áreas correspondem a 1000Km<sup>2</sup> cada, sendo divididas posteriormente em nove subáreas. As imagens extraídas correspondentes a estas subáreas e estão contempladas no intervalo de tempo compreendido entre 1 de junho e 15 de agosto de 2019.

### 5.1.2 Detecção e Geolocalização de Navios

Após a obtenção de imagens nos três locais díspares, deu-se início à detecção dos navios presentes nas imagens. Esta detecção foi realizada através do modelo treinado previamente com uma *confidence threshold* de 20%. O resultado foi a criação de um ficheiro com as imagens detetadas, identificando as detecções através de *bounding boxes* vermelhas, com a respetiva classe e *confidence* de cada detecção. Em simultâneo, a ferramenta também gerou um ficheiro com as geolocalizações das respetivas detecções, identificando a que imagem pertence, a classe respetiva, assim como as coordenadas por pixel. Este passo é crucial, pois organiza a informação espacial, facilitando o mapeamento entre os dados das imagens e as detecções realizadas.

### 5.1.3 Geolocalização de AIS

Uma vez que a ferramenta já criou um ficheiro com todas as geolocalizações das embarcações detetadas, o passo seguinte recai sobre a simulação dos dados AIS. Nesta fase, um novo ficheiro é gerado, contendo 80% da informação do documento original, através de um critério de aleatoriedade. Esta simulação tem em vista preservar o formato e a estrutura dos dados do documento original, uma vez que o procedimento que se segue foca-se numa fusão e correlação de dados.

### 5.1.4 Fusão de Informação

A etapa seguinte consiste em correlacionar a geolocalização de cada detecção com a de cada sinal AIS simulado, gerando um novo ficheiro a partir da interverificação entre ambos. Este ficheiro destaca as incompatibilidades de geolocalização, apresentando apenas as disparidades no documento final. As imagens com irregularidades são realçadas com *bounding boxes* azuis, sinalizando navios sem emissão AIS detetada. Este procedimento ajuda a evidenciar os dados finais e permite analisar mais detalhada e visualmente, com o benefício de indicar áreas de interesse em investigações posteriores.

### 5.1.5 Constrangimentos

Mantendo em consideração que uma das bases desta dissertação são imagens satélite, e estando o seu controlo fora de alcance, existem diversos constrangimentos que se devem a este fator. A nível de imagens obtidas é notável que existe uma clara distinção entre as cores das imagens tendo em conta a localização, o que dificulta a detecção. Em casos em que o cenário é captado totalmente cinzento pelo satélite, e adicionando o fator da embarcação também o ser, acaba por se camuflar o objeto em causa, impedindo a sua detecção. Outro fator de enaltecer, são os casos em que o meio marítimo se encontra demasiado agitado, criando formas de arrasto de embarcações e, por vezes, confundindo o modelo. O mesmo acontece em situações em que as nuvens cobrem a maioria da imagem mas são relativamente transparentes em determinados sítios, o que permite ao olho humano identificar embarcações, mas não ao detetor, uma vez que não consegue compreender o formato do objeto por detrás das nuvens.

Aquando a extração de imagens em zonas portuárias, é relativamente frequente a existência de pequenas embarcações que não são detetadas, nem através da *wake*. Ainda que estas não sejam relevantes para o caso de estudo em causa por não reunirem as condições de obrigatoriedade de emissão de AIS, seria interessante o modelo treinado conseguir captar todo o tipo de embarcação presente na área de interesse. Adicionalmente, em embarcações com elevado relevo e cores distintas, o modelo acaba por realizar duas detecções na mesma embarcação.

## 5.2 Resultados

### 5.2.1 Modelo de Detecção

A escolha do modelo de treino consoante os *datasets* apresentados sucedeu-se consoante determinadas características mencionadas anteriormente, especialmente, o facto de abranger muitos casos que correspondem à realidade de uma imagem satélite, sendo que verificou-se que o modelo do *dataset* da *Airbus* apresentou uma melhor performance. Através do treino realizado com o modelo da *Airbus* com 200 *epochs* no *Jupyter*, foi possível obter o modelo final com

bastantes respostas positivas aos critérios necessários, reconhecendo embarcações em cenários tanto simples como ruidosos e, por vezes, unicamente através da *wake*. Verificou-se que este modelo final apresenta uma razão aceitável de verdadeiros com falsos positivos com uma *confidence threshold* de 20%.

O treino foi realizado com o modelo de deteção YOLOv8, uma vez que esta versão possibilitava uma deteção rápida e eficaz, e, posteriormente, exportar num formato correspondente com o desejado. A exportação consiste na obtenção de imagens com as *bounding boxes* correspondentes a cada deteção, assim como toda a informação de treino do modelo, incluindo o necessário para aplicar o modelo noutros conjuntos de imagens. Na tabela 1, as métricas correspondentes ao Modelo *Roboflow*, equivalem às do treino inicial no *Roboflow* e estão expostas unicamente para efeitos de comparação. As métricas relativas ao Modelo Final correspondem às obtidas no final do treino em causa.

Tabela 1: Modelo final treinado no *Jupyter*

| Imagens de Treino      | Imagens | AP    | Precision | Recall |
|------------------------|---------|-------|-----------|--------|
| Modelo <i>Roboflow</i> | 9900    | 71,2% | 84,2%     | 59,7%  |
| Modelo Final           | 9900    | 75%   | 80%       | 66%    |

O modelo final de deteção demonstrou uma alta performance em diversos cenários com condições climatéricas adversas. As imagens deste *dataset* têm dimensões de 768X768, extraídas de satélites de alta resolução, onde cada pixel corresponde a 1 metro real. Neste caso, as dimensões das embarcações podem variar entre 30 a 200 pixels de comprimento e, 10 a 50 pixels de largura, sendo que cada navio encontra-se posicionado aleatoriamente. O desafio deste *dataset* reverte-se nesta variabilidade de tamanho, onde os barcos de menor dimensão acabam por não ser detetados ou, quando detetados, têm uma *confidence* bastante reduzida. Adicionalmente, uma grande parte das imagens não apresentava qualquer tipo de navio enquanto que, por vezes, uma imagem tinha cerca de 10 embarcações. O modelo quando testado, apresentou-se sempre como um modelo robusto que não se confunde facilmente, mesmo em cenários bastante ruidosos em alto mar. Por outro lado, quando aplicado em imagens de zonas portuárias, o término do cais é habitualmente detetado como uma embarcação.

A análise objetiva da performance concentra-se nas métricas oriundas do desempenho do modelo. Uma AP de 75% diz-nos que o modelo consegue obter uma precisão elevada na classificação dos objetos relativamente à *ground truth*. Adicionalmente, com uma *Precision* de 80% é possível concluir que o modelo de deteção tem um baixo índice de falsos positivos. Não obstante, e em contraste, o valor de *Recall*, de 66% indica-nos que existe uma dificuldade em detetar todas as embarcações presentes nas imagens.

5.2.2 Análise do Modelo Aplicado nas Deteções

As imagens à qual foram aplicadas o modelo anteriormente descrito são imagens satélite retiradas do *SentinelHub*, de tamanho 1000X1000, totalizando um conjunto de 393 imagens que não constituem o *dataset* de treino. O valor utilizado como *confidence* foi de 20%, tornando notório que o modelo tem alguns *overlaps* em embarcações de comprimentos superiores, originando falsos positivos. A nível de resultados visuais do modelo, este detetou diversas embarcações, que variavam entre as dimensões de 15 a 200 pixels de comprimento e 25 a 70 pixels de largura, onde as embarcações à semelhança do *dataset* de treino também se encontram posicionadas aleatoriamente. À semelhança do modelo de treino, verificou-se que em todas as imagens que continham portos, o modelo confundia-se, realizando deteções como falsos positivos. Todas as embarcações que não foram detetadas e eram inferiores a 15 metros de comprimento, não foram consideradas como uma falha do modelo de deteção, tendo em consideração que não são obrigadas a emitir um sinal AIS.

Nesta análise, o objetivo é interpretar os resultados do modelo previamente treinado aplicado nas imagens extraídas pela ferramenta. As métricas obtidas no final da deteção encontram-se representadas na tabela 2. Verificou-se que com uma AP de 89,7% o modelo consegue obter bastante precisão em classificar devidamente os navios relativamente à *ground truth*. Uma *Precision* de 88,9% indica-nos que o modelo tem um índice de falsos positivos reduzido e uma *Recall* de 85,8% indica-nos que uma elevada capacidade de detetar as embarcações presentes nas imagens.

Tabela 2: Métricas de validação do modelo final treinado no *Jupyter* aplicado na ferramenta

| Imagens de Validação                    | Nº Imagens | AP    | Precision | Recall |
|-----------------------------------------|------------|-------|-----------|--------|
| Imagens Extraídas do <i>SentinelHub</i> | 393        | 89,7% | 88,9%     | 85,8%  |

Interpretando as curvas de *Precision-Confidence*, *Recall-Confidence* e *Precision-Recall* para o modelo experimental, representadas na figura 3, também é possível verificar valores positivos. Iniciando pelo gráfico de *Precision* em função da *Confidence*, verificamos uma grande diferença aquando comparado com o modelo original. Neste modelo experimental obtemos uma confiança de 45,3% para uma precisão de 100%. O significado destes valores, é que estamos perante um modelo mais tolerante em relação ao limiar de confiança e que deteta mais objetos sem comprometer a precisão, contrariando o modelo anterior que se revela mais seletivo e tenta evitar um acréscimo de falsos positivos.

Analisando a curva de *Recall* em função de *Confidence*, esta tem como valor inicial de 0.94, cujo indica que o modelo tem a capacidade de detetar 94% de todos os objetos presentes. Sendo que o modelo do *dataset* inicial apresenta uma *Recall* de 86%, estamos perante uma diferença de 0.08, o que, embora pareça insignificativo a níveis práticos, indica-nos que a cada 100 objetos reais, são detetados mais 8 do que o modelo original. A curva deste gráfico também cai mais gradualmente que a do modelo original, sugerindo uma maior eficácia para evitar falsos negativos, mesmo perante níveis de confiança mais restritos.

Por último, através do gráfico *Precision versus Recall* verificamos que o valor médio de precisão é de 89,7%, resultando num aumento de 14,7% para o modelo anterior. Um valor de AP mais alto significa que o modelo tem uma probabilidade mais alta de detetar corretamente os navios. Deste modo, verificamos uma redução de falsos positivos e falsos negativos, elevando a performance do modelo pela melhor capacidade de deteção de navios. Uma vez que a *Precision* deste gráfico mantém-se elevada para níveis de *Recall* superiores, estamos perante um modelo mais equilibrado e com uma melhor capacidade de deteção no geral.

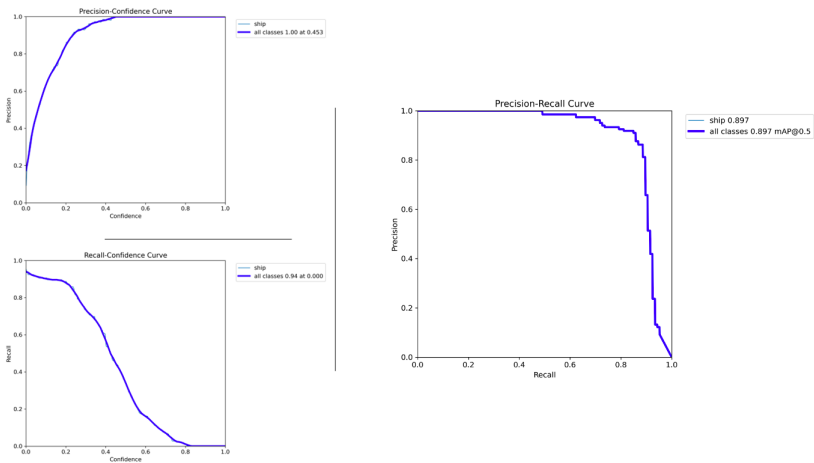


Figura 3: Métricas do Modelo Experimental.

## 6 Conclusões

A Força Aérea enfrenta desafios na missão de patrulhamento marítimo devido à vasta área a cobrir e ao aumento da atividade naval, dificultando o controlo de práticas ilícitas e a manutenção de *situational awareness*. Em acréscimo, a desativação do AIS por parte dos utilizadores para a prática de atividades ilegais incrementa a dificuldade sentida uma vez que inviabiliza o mapeamento em tempo real.

Nesta dissertação, foi desenvolvida uma ferramenta para detetar navios não identificados com recurso a imagens de satélite. A abordagem dividiu-se em quatro etapas: extração de imagens do satélite *Sentinel-2A*, deteção de objetos com *YOLOv8*, simulação de dados AIS, e fusão de geolocalizações a fim de identificar discrepâncias. A ferramenta tem como resultado final o destaque de navios não identificados.

Os testes experimentais apresentaram uma alta precisão (89,7%), mesmo com a introdução de limitações como a diversidade de dimensões das embarcações e condições meteorológicas adversas. As imagens em que foram feitas as deteções apresentam-se com uma menor resolução do que as de treino, o que demonstra uma boa generalização do modelo de deteção. Por outro lado, é de manter em consideração que as embarcações do teste não têm dimensões tão reduzidas como algumas do *dataset* de treino e as imagens foram direcionadas para épocas de verão onde as condições meteorológicas não são extremamente adversas à deteção. Ainda que estejam presentes muitos fatores, estes não são uma maioria.

Em suma, o modelo de deteção desenvolvido apresenta uma performance elevada, evidenciada pela sua capacidade de identificar navios de forma precisa e eficiente, mesmo em cenários com diferentes condições ambientais ou características variáveis das embarcações. Além de cumprir os objetivos inicialmente propostos, esta ferramenta possui a flexibilidade e o potencial para ser aprimorada e adaptada para aplicações mais específicas, sendo uma base sólida para futuros trabalhos na área de monitorização e vigilância marítima.

Como mencionado ao longo deste trabalho, a temática de deteção de objetos em prol das missões da Força Aérea tem vindo ser um alvo de estudo, sendo esta dissertação um dos projetos iniciais, com valores iniciais que servem de base. Ao longo do desenvolvimento deste estudo optou-se por escolhas relativas à limitação temporal e programática, sendo que nem todas as possíveis opções foram testadas. Neste âmbito, foi deixado espaço para futuros trabalhos. Considerando os modelos de deteção e o desenvolvimento destes, embora o *YOLO* seja um método com maior equilíbrio performance-velocidade, existem diversos métodos de deteção que apresentam uma precisão média superior, pelo qual é recomendado testar devido às dimensões dos objetos a detetar. Neste âmbito, tendo em consideração que as imagens extraídas apresentam uma resolução média, acaba por reduzir a qualidade de deteção pelo que recomenda-se que a ferramenta seja testada em imagens com resoluções superiores. No que toca à base de dados AIS, ao criar uma amostragem simulada perdeu-se o fator de ruído de geolocalizações. Ainda que esteja previsto uma solução caso os dados fossem reais, recomenda-se que esta seja testada na ferramenta, a fim de garantir que a fusão de dados tem em consideração o movimento da embarcação e o possível desvio de emissão.

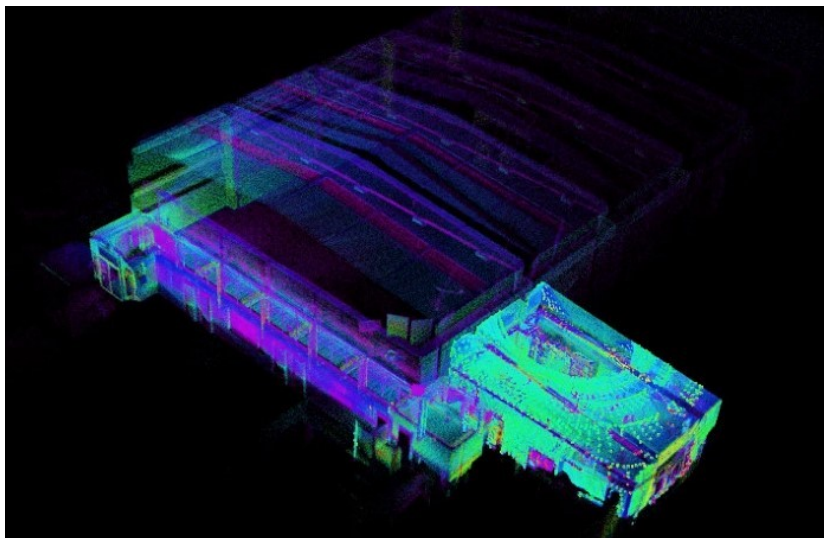
## Referências Bibliográficas

- Branco, G. F. C. (2022). *A busca e salvamento marítimo em portugal no século xxi* (Ph.D. Thesis). Instituto Universitário Militar (IUM).
- Cajarabille, V. L. (2009). A segurança no mar português. *Nação e Defesa*.
- Carvalho, V. d. (1982). O poder marítimo. *Nação e Defesa*.
- Casas, G. G., Ismail, Z. H., Limeira, M. M. C., da Silva, A. A. L., & Leite, H. G. (2023). Automatic detection and counting of stacked eucalypt timber using the yolov8 model. *Forests*, 14(12), 2369.
- Cichy, R. M., & Kaiser, D. (2019). Deep neural networks as scientific models. *Trends in cognitive sciences*, 23(4), 305–317.
- Crispim, L. C., & Albano, A. (2016). O uso das imagens de satélite como recurso didático no ensino de geografia. *Pesquisar—Revista de Estudos e Pesquisas em Ensino de Geografia*, 3(4), 46–57.
- Cunha, T. P. (2004). A importância estratégica do mar para portugal. *Nação e Defesa*.
- DGRM. (n.d.). *Zonas marítimas sob jurisdição e ou soberania nacional - dgrm*, url=<https://www.dgrm.pt/am-ec-zonas-maritimas-sob-jurisdicao-ou-soberania-nacional>.

- Dias, L., & do Mestrado Integrado, A. A. (2014). Isr e conhecimento da força aérea na zona económica exclusiva. *Revista Científica Academia da Força Aérea*, 4, 38–46.
- Dongare, A., Kharde, R., Kachare, A. D., et al. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1), 189–194.
- dos Santos Quinta Albuquerque, S. I. (2022). *O impacto da tecnologia no policiamento* (Master's Thesis, Instituto Universitário Militar (IUM)). Retrieved from <https://comum.rcaap.pt/handle/10400.26/41640> (Disponível no Repositório Comum)
- Falqueto, L. E., Paes, R. L., & Passaro, A. (2021). A importância de dados polarimétricos para detecção de alvos artificiais no mar utilizando imagens sar. *Aplicações Operacionais em Áreas de Defesa*, 22, 53–58.
- Hussain, M. (2024). YoloV1 to v8: Unveiling each variant—a comprehensive review of yolo. *IEEE Access*, 12, 42816–42833.
- Ibrahim, Y., Wang, H., Liu, J., Wei, J., Chen, L., Rech, P., ... Guo, G. (2020). Soft errors in dnn accelerators: A comprehensive review. *Microelectronics Reliability*, 115, 113969.
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics yoloV8*. Retrieved from <https://github.com/ultralytics/ultralytics> (Available under AGPL-3.0 license. ORCID: 0000-0001-5950-6979, 0000-0002-7603-6750, 0000-0003-3783-7069.)
- Kanjir, U., Greidanus, H., & Oštir, K. (2018). Vessel detection and classification from spaceborne optical images: A literature survey. *Remote sensing of environment*, 207, 1–26.
- Kirbas, C., & Quek, F. (2004). A review of vessel extraction techniques and algorithms. *ACM Computing Surveys (CSUR)*, 36(2), 81–121.
- Lee, E., Mokashi, A. J., Moon, S. Y., & Kim, G. (2019). The maturity of automatic identification systems (ais) and its implications for innovation. *Journal of Marine Science and Engineering*, 7(9), 287.
- Li, F., Sun, T., Dong, P., Wang, Q., Li, Y., & Sun, C. (2024). Msf-cspnet: A specially designed backbone network for faster r-cnn. *IEEE Access*.
- Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., Kochenderfer, M. J., et al. (2021). Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4), 244–404.
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet], 9(1), 381–386.
- Maia, J. (2021). Busca e salvamento nas regiões de informação de voo de lisboa e de santa maria. *IUM*.
- Mishra, M., & Srivastava, M. (2014). A view of artificial neural network. In *2014 international conference on advances in engineering & technology research (icaetr-2014)* (pp. 1–3).
- Sapkota, R., Meng, Z., Ahmed, D., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2024). Comprehensive performance evaluation of yoloV10, yoloV9 and yoloV8 on detecting and counting fruitlet in complex orchard environments. *arXiv preprint arXiv:2407.12040*.
- Shermeyer, J., & Van Etten, A. (2019). The effects of super-resolution on object detection performance in satellite imagery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 0–0).
- Silveira, B. (2019). Vigilância e patrulhamento marítimo: uma visão para 2030. *IUM*.
- Sultana, F., Sufian, A., & Dutta, P. (2020). A review of object detection models based on convolutional neural network. *Intelligent computing: image processing based applications*, 1–16.
- Tahir, A., Munawar, H. S., Akram, J., Adil, M., Ali, S., Kouzani, A. Z., & Mahmud, M. P. (2022). Automatic target detection from satellite imagery using machine learning. *Sensors*, 22(3), 1147.
- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of yolo architectures in computer vision: From yoloV1 to yoloV8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716.
- Tunaley, J. (2013). Utility of various ais messages for maritime awareness. *contract*, 9, 7009458.
- Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 390–391).
- Wolsing, K., Roepert, L., Bauer, J., & Wehrle, K. (2022). Anomaly detection in maritime ais tracks: A review of recent approaches. *Journal of Marine Science and Engineering*, 10(1), 112.
- Xiao, G., Yang, D., Xu, L., Li, J., & Jiang, Z. (2024). The application of artificial intelligence technology in shipping: A bibliometric review. *Journal of Marine Science and Engineering*, 12(4), 624.



# Implementação de sistema de localização para navegação sem recurso a GNSS



**Autor: ALF/PILAV Miguel Castanhinha Carrasqueira**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

**Orientador: TCOR/TMAEQ/123650-E Paula Alexandra Veiga Gonçalves**  
Academia da Força Aérea, Sintra

**Coorientador: CAP/ENGEL/136787-A Diogo Alexandre Oliveira Silva**  
Academia da Força Aérea, Sintra



## Resumo

Os *Unmanned Aircraft System* UAS têm-se desenvolvido em larga escala, sendo de elevada importância que estes tenham um sistema em que seja possível estimar a sua posição e velocidade. Isto é possível no exterior através de, por exemplo, *Global Navigation and Satellite Systems* (GNSS). No entanto, nem todos os ambientes têm sinal de satélite suficientemente forte, como é o caso do interior de edifícios, ou num cenário onde exista interferência eletromagnética, como, por exemplo, *jamming*. Nestes cenários, os UAS terão de obter a sua localização através de outro tipo de sistemas. O objetivo primário desta dissertação é o desenvolvimento de um sistema de localização, de modo a ser utilizado na navegação de um UAS de pequenas dimensões em ambientes onde não exista sinal GNSS. Para o desenvolvimento do sistema de localização foi comparado um *Light Detection and Ranging* (LIDAR), a uma câmara estereoscópica, sendo posteriormente escolhido o LIDAR como sensor deste sistema. O LIDAR tem o objetivo de capturar a profundidade do ambiente e funciona em conjunto com um algoritmo *Simultaneous Localization and Mapping* (SLAM) que mapeia o ambiente envolvente e localiza-se no mesmo. Para a comunicação entre estes dois componentes foi utilizado o *Robot Operating System* (ROS). Foi feita a integração deste sistema com um piloto automático, de forma a verificar a viabilidade da implementação deste sistema de localização na navegação do UAS.

Este sistema foi testado utilizando dois tipos de abordagens. Inicialmente, realizaram-se testes manuais, com os motores do UAS desligados, para avaliar a precisão do sistema de localização. Posteriormente, foram realizados testes de voo para validar a viabilidade da implementação deste sistema de localização em condições semelhantes às reais.

**Palavras-chave:** UAS, GNSS, SLAM, LIDAR, *Pixhawk*, ROS.

## 1 Introdução

Os UAS têm-se desenvolvido ao longo dos últimos anos e são cada vez mais importantes, devido às suas várias aplicações (Jeon, Jung, Lee, Choi, & Myung, 2021). A maior parte dos voos são feitos em *Visual Line of Sight* (VLOS), ou seja, quem está no controlo do UAS tem de manter contacto visual com o mesmo, para o comandar. Contudo, isto torna-se uma limitação e têm sido desenvolvidos projetos com o objetivo de expandir a operação dos UAS para além da linha de vista. Para isso acontecer, é necessário aumentar a segurança e a autonomia dos UAS. Podem acontecer falhas de comunicação entre o UAS e a estação de controlo e, por isso, é necessário que o UAS tenha um sistema de localização, um sistema de navegação e um sistema para detetar obstáculos e evitá-los (Aldao, González-de Santos, & González-Jorge, 2022).

No caso de ambientes exteriores, os UAS baseiam-se em sistemas de GNSS e em sistemas de *Positioning Navigation and Timing* (PNT). Se este sistema tiver sinal de satélite suficiente, é possível serem dadas informações fidedignas quanto à sua localização e velocidade. No entanto, nem todos os ambientes têm sinal GNSS. Um dos desafios da navegação por parte do UAS é em locais como o interior de edifícios, em que o sinal GNSS é fraco, ou inexistente, impreciso e altamente suscetível a interferência, como reflexões do sinal. Por esse motivo, os UAS terão de obter a sua localização através de outro tipo de sensores. Devido a isso, os investigadores estão a estudar novas formas de navegação, nomeadamente uma navegação autónoma (Gao, Lang, & Ren, 2020). Para que o mesmo aconteça, os UAS terão de realizar diversas tarefas, nomeadamente, conseguir localizar-se no meio onde se encontram, controlo e planeamento, deteção de obstáculos de forma a evitá-los e tomada de decisão (Suriano, 2021). Contudo, existem diversas limitações para a implementação desta nova forma de navegação num UAS, em particular o tamanho, o peso e limites de potência (Jeon et al., 2021).

O objetivo primário desta dissertação de mestrado é a implementação de um sistema de localização que não tenha recurso a sistemas GNSS, nem ambientes cooperativos ou sistemas externos desenvolvidos para ajudar no processo de localização. Serão utilizadas ferramentas de *software* de *open-source*. Serão preferencialmente utilizados sensores que forneçam informação sobre a profundidade. Este sistema de localização poderá fazer uso de mapas previamente criados ou através de SLAM. O sistema de localização será implementado em um UAS de pequenas dimensões, o que vai restringir o peso dos diferentes componentes *hardware* utilizados, devido ao *payload* do UAS. O sistema de localização terá também de ter uma capacidade computacional baixa, uma vez que a potência que é possível fornecer ao sistema é limitada.

## 2 Estado de arte

### 2.1 Sensores

Existem diversos sensores onde foi estudada a sua implementação em um sistema de localização, sendo o primeiro abordado, o *Inertial Measurement Unit* (IMU). Estes sensores baseiam-se em acelerômetros, giroscópios e magnetômetros, que conseguem detectar a aceleração, podendo assim detectar a distância percorrida. Como vantagem, este sensor necessita de pouca potência para ser alimentado, tem pequenas dimensões e não necessita de qualquer marcador externo ou dispositivo para o seu normal funcionamento. A grande desvantagem é que este tipo de sensores acumulam erros, o que faz com que haja uma redução da precisão da estimativa de localização do UAS.

As câmaras monoculares são outro tipo de sensores em que já foi estudada a sua implementação em um sistema de localização (Troll, Szipka, & Archenti, 2020). As câmaras monoculares são sensores económicos, compactos e eficientes quanto à potência requerida. Não obstante, na atualidade, ainda não é possível obter de forma direta, apenas com uma câmara monocular, a profundidade do meio, de modo a permitir que o UAS se localize autonomamente. A forma mais comum para criar um sistema que seja capaz de medir a profundidade do meio envolvente é combinar a câmara monocular com um IMU (Jeon et al., 2021). Outra forma de resolver o problema da obtenção de profundidade através de uma câmara monocular é com algoritmos, em que o sistema vai aprender, de forma autónoma, a perceber qual a profundidade dos objetos que perceciona, através das características de determinados *pixels*. No entanto, este tipo de abordagem necessita de bastante informação para treinar o sistema, o que leva a uma elevada complexidade computacional (Huyh, Nguyen-Ha, Matas, Rahtu, & Heikkilä, 2020). Ademais, a câmara monocular depende das condições de iluminação para detectar a profundidade do meio envolvente, não conseguindo fazê-lo quando não existe luminosidade ou quando, por exemplo, existe uma fonte de luz que incida diretamente na câmara (Petrík, Krajník, & Saska, 2021).

Através das câmaras estereoscópicas foi também já testada a obtenção de uma estimativa de localização para um sistema de navegação. Esta abordagem funciona inspirada na visão dos humanos, sendo este tipo de câmaras compostas por duas câmaras que apontam para direções paralelas, separadas por uma determinada distância. Quando é detetado um determinado ponto no espaço, este é captado pelas duas câmaras, em posições e planos diferentes. A profundidade é calculada através da disparidade de posição do ponto de referência nas duas imagens (Sanchez-Rodriguez & Aceves-Lopez, 2018).

A grande vantagem deste tipo de câmaras, face às câmaras monoculares, é que não necessitam de outros algoritmos ou sensores para obter diretamente a profundidade. Outra vantagem é que este tipo de câmaras está diretamente ligado às plataformas de processamento de imagem, permitindo que o cálculo de profundidade seja realizado de forma autónoma, sem a necessidade de algoritmos intermediários, o que contribui para que não haja perda de informação (Mustafah, Azman, & Akbar, 2012).

As câmaras TOF (*Time-of-Flight*) têm ganho bastante importância ao longo do tempo, pois, ao contrário das câmaras monoculares, esta câmara consegue detectar a profundidade diretamente, não necessitando de qualquer algoritmo. Estas câmaras são de fácil utilização, de pequenas dimensões e elevada precisão. Este tipo de câmaras iluminam o ambiente através de uma onda que irá ser projetada a partir da câmara, e posteriormente, esta é refletida. Através da diferença de fases da onda, é calculada a distância a que os diversos objetos se encontram (Li et al., 2014).

Este tipo de câmara consegue colmatar a principal desvantagem das câmaras monoculares e das câmaras estereoscópica, uma vez que consegue medir a profundidade mesmo em condições de iluminação reduzidas, ou em casos onde existe uma variação de intensidade da luminosidade (Chen, Chang, & Wen, 2020).

O LIDAR é um sensor com alta resolução e precisão de informação, quando comparado com os restantes. Este emite feixes de radiação infravermelha para diferentes direções, conseguindo fazer uma identificação do ambiente onde se desloca (Aldao et al., 2022). Como desvantagem, este tipo de sensores que se baseiam em *laser scanners* são mais dispendiosos do que as câmaras monoculares e estereoscópicas e podem sofrer de erros devido ao movimento da plataforma em que estão a ser utilizados e a reflexões que possam existir no meio envolvente (Bhoi, 2019). Este tipo de sensores possui uma vantagem comparativamente com as câmaras monoculares e estereoscópicas, pois pode ser utilizado independentemente da luminosidade existente (Bosse, Zlot, & Flick, 2012).

## 2.2 Algoritmo de localização

Uma das formas de obter a localização é através de algoritmos de SLAM. Os algoritmos de SLAM permitem que uma determinada plataforma se localize num mapa que não existe previamente e que é construído à medida que essa mesma plataforma se desloca. Esse processo é essencial para que o UAS planeie e execute rotas de forma autónoma, evitando obstáculos ao longo do trajeto (Taheri & Xia, 2021). Foram estudados apenas dois tipos de SLAM: SLAM com tecnologia visual (V-SLAM) e SLAM que utiliza o LIDAR (Hening, Ippolito, Krishnakumar, Stepanyan, & Teodorescu, 2017).

Uma das técnicas utilizadas para aperfeiçoar o SLAM é o *Loop Closure*. Esta abordagem é utilizada com o objetivo de reduzir o erro que se acumula ao longo do tempo. Através deste método, o sistema consegue detetar se já esteve no local onde se encontra, ou se é a primeira vez. Caso já tenha passado por um determinado ponto do mapa, este sistema vai aperfeiçoar o mapa criado (Yang, Lin, You, Yan, & Cheng, 2021).

### 2.2.1 LIDAR SLAM

Este tipo de tecnologia, quando comparado com o V-SLAM, tem uma melhor precisão, visto que não é afetado pela luminosidade, podendo operar em condições onde não existe qualquer tipo de iluminação. Todavia, este tipo de algoritmos não é capaz de identificar as cores do meio envolvente (Roch & Najjaran, 2023). Uma das vantagens que o LIDAR SLAM tem em comparação com o V-SLAM é o facto de, no V-SLAM, quando o sistema volta a um local onde já esteve anteriormente, este local poderá estar com condições de luminosidade diferentes, o que irá fazer com que o *Loop Closure* nem sempre seja executado (Arshad & Kim, 2021).

Um dos algoritmos SLAM é o *Cartographer* que interliga a informação de um LIDAR e de um IMU de forma a construir um mapa e obter uma estimativa de localização robusta. O facto de se basear em dois sensores permite mitigar as limitações de cada um dos tipos de sensores, permitindo voar em locais onde não existam muitos pontos de referência (Milijas, Markovic, Ivanovic, Petric, & Bogdan, 2021).

Outro algoritmo SLAM que funciona com base no LIDAR e no IMU é o FASTLIO. Este utiliza um *Extended Kalman Filter* (EKF) iterativo que corrige a distorção da nuvem de pontos e facilita o processo de correspondência entre varreduras do LIDAR. A grande vantagem deste algoritmo é que permite a criação de mapas densos, o que, por sua vez, permite uma correspondência eficiente entre a varredura e o mapa, por intermédio do *Loop Closure*, o que vai fazer com que os desvios que o mesmo tenha sejam mínimos (Vultaggio, d'Apolito, Sulzbachner, & Fanta-Jende, 2023).

### 2.2.2 V-SLAM

Este tipo de SLAM tem-se desenvolvido devido à informação que este consegue fornecer do meio ambiente, nomeadamente a construção de um mapa com cores, algo que os sensores que funcionam à base de *laser scans*, como é o caso do LIDAR, não têm capacidade. Por outro lado, este sistema tem um custo computacional mais elevado, precisando de algoritmos mais sofisticados para o processamento de imagem. Inicialmente, desenvolveram-se métodos SLAM 2D e 3D, devido a diversas limitações dos recursos computacionais. Na atualidade, já é possível fazer uma reconstrução 3D do ambiente, de uma forma mais densa, devido à disponibilidade de novos métodos mais eficientes e ao avanço tecnológico (Al-Tawil, Hempel, Abdelrahman, & Al-Hamadi, 2024).

Um dos algoritmos V-SLAM é o RTAB-Map (*Real-time appearance-based mapping*) que é uma biblioteca *open-source* que implementa a tecnologia de deteção através de *Loop Closure*. Este método adota uma abordagem de gestão de memória que limita o tamanho do mapa, permitindo a execução do *Loop Closure* dentro de um intervalo de tempo pré-determinado. O RTAB-Map distingue-se de outros algoritmos SLAM, pois este consegue ser utilizado com diferentes tipos de sensores, desde câmaras a sensores que utilizam *laser scans*. Com este algoritmo também é possível combinar várias informações provenientes de diferentes sensores, o que permite que as desvantagens dos vários sensores sejam colmatadas, ao mesmo tempo que facilita a execução de *Loop Closure* e torna mais precisa a estimativa de localização do UAS (Roch & Najjaran, 2023).

Outro algoritmo utilizado é o ORB-SLAM, que é um algoritmo de SLAM que é bastante reconhecido por operar em tempo real em vários ambientes, quer no interior de edifícios, quer no exterior. Este sistema consegue manter consistência, ao mesmo tempo que é eficiente computacionalmente. Este algoritmo constrói um mapa através de diversos pontos, administrando de forma equilibrada a criação abundante de pontos iniciais, com a remoção de pontos redundantes que

foram anteriormente criados, permitindo uma melhor estabilidade e duração das operações (Mur-Artal, Montiel, & Tardos, 2015).

### 3 Metodologia

#### 3.1 Arquitetura do sistema

O sistema de localização foi concebido de modo a combinar diversos elementos de *hardware* e *software* para que seja possível a sua operação em tempo real. Inicialmente, é necessária a presença de um sensor, que terá o propósito de conseguir calcular a profundidade entre o mesmo e o ambiente em seu redor, de uma forma direta. Esta informação será acoplada a um IMU, de forma a ser possível obter uma estimativa de localização mais precisa, compensando movimentos do UAS e corrigindo possíveis distorções nas medidas do sensor, causadas pela dinâmica de voo. Posteriormente, a informação proveniente dos dois sensores é fundida e utilizada por um algoritmo de localização. Este terá o objetivo de mapear o local onde o UAS se encontra e, ao mesmo tempo, localizar-se no mapa que está a ser construído. O algoritmo de localização e as informações obtidas pelos sensores podem comunicar de forma direta e são interpretadas através de um computador de bordo. Posteriormente, as informações provenientes do algoritmo de localização são enviadas para o piloto automático. No piloto automático, existe um EKF que, além de processar as informações fornecidas pelo algoritmo de localização, também integra os dados recebidos de um IMU do próprio piloto automático. Posteriormente, as informações processadas pelo controlador do piloto automático fornecem os dados para que possam ser utilizados pelo motor. Por fim, o piloto automático é ligado a uma antena de telemetria, que será responsável por fornecer as informações do estado do UAS a um computador externo. A arquitetura do sistema encontra-se representada na Figura 1.

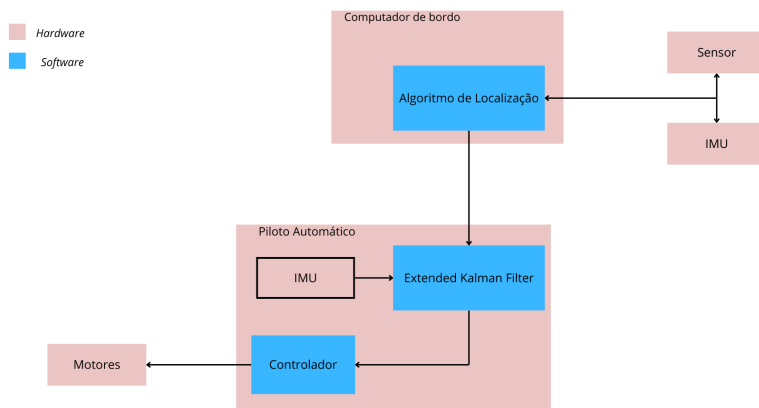


Figura 1: Arquitetura do sistema de localização

#### 3.2 Seleção do sensor e algoritmo de localização

A escolha do sensor e do algoritmo de localização são duas partes cruciais no desenvolvimento do sistema de localização, pois esta seleção vai influenciar de forma direta a robustez, eficiência e precisão do sistema de localização.

No caso da escolha do sensor é importante ter em consideração a capacidade de este gerar uma nuvem de pontos para ambientes tridimensionais, sendo que esta tem de ter detalhe suficiente para uma caracterização pormenorizada do

ambiente ao redor do sensor. O sensor necessita de ter um alcance considerável e foi ponderada a robustez do sensor e a possibilidade de operar em qualquer tipo de ambiente.

Para a escolha do algoritmo de localização, é necessário que este possua uma robustez elevada perante a perda de referências, ou seja, caso o sensor não consiga detectar no ambiente pontos-chave, este algoritmo terá de ter a capacidade de não perder a estimativa de localização. Para além disso, é importante que este algoritmo tenha uma precisão elevada, tendo de fornecer uma baixa margem de erro, quer na localização do UAS, quer na construção do mapa. A solução escolhida tem de ser capaz de lidar com alterações do meio ao seu redor, como identificar objetos em movimento.

### 3.3 Teste e validação

Inicialmente, os testes foram realizados, apenas com o sistema de localização, de forma a escolher o sensor e o algoritmo de localização. Foram realizados testes preliminares através da plataforma *Rviz*, que permite avaliar a qualidade dos mesmos. Posteriormente, o sistema foi implementado em um UAS para a realização de testes. Primeiramente, os motores encontram-se desligados e o UAS foi transportado manualmente ao longo de um percurso predefinido. Antes da realização do teste serão feitas as medições do percurso que o mesmo vai percorrer de modo a verificar a precisão do sistema de localização. Desta forma, são garantidos os requisitos mínimos para ser realizado um ensaio de voo. Os ensaios de voo são realizados com o objetivo de verificar a capacidade do UAS voar de forma autónoma, mantendo-se na mesma posição em um voo estacionário, apenas com as informações fornecidas pelo sistema de localização. Em todos os testes, foram gravados *rosbag*, o que permite a análise do voo, podendo ser analisados os resultados do sistema de localização. Por fim, caso o UAS consiga manter o voo estacionário, seria realizado um teste com um algoritmo de planeamento de trajetória, em que se verificará se o UAS tem a capacidade de voar de um ponto para outro, desviando-se dos vários obstáculos presentes ao longo do seu percurso.

## 4 Implementação

### 4.1 Componentes do sistema de localização

Inicialmente, optou-se por utilizar alguns componentes já disponíveis no Centro de Investigação da Academia da Força Aérea (CIAFA), devido a estes já terem sido utilizados em projetos anteriores ou que se encontravam disponíveis para utilização.

Foi utilizado o UAS *Holybro PX4 Development Kit - X500 v2* como plataforma de testes deste sistema de localização. Este UAS é fabricado em fibra de carbono, tornando-o leve e, ao mesmo tempo, robusto. As suas características tornam-no uma plataforma ideal para o desenvolvimento de um UAS autónomo, em virtude da sua versatilidade e estabilidade de voo, permitindo diferentes tipos de missões. Este possui um *payload* máximo de 1500 g (*PX4 Development Kit - X500 v2*, n.d.).

Para piloto automático foi utilizado o *Pixhawk 6c*, pois é recomendado para o uso com o *Holybro PX4 Development Kit - X500 v2*. O *Pixhawk 6c* permite a integração com diversos sensores e sistemas de navegação, permitindo a obtenção de odometria através de sensores externos. A partir deste, é possível testar o sistema de localização (*Holybro Pixhawk 6C | PX4 Guide (main)*, n.d.). Este piloto automático está representado na Figura 2.

De forma a permitir que o algoritmo de localização utilize os dados provenientes do sensor, foi utilizado o ROS. Este suporta uma grande variedade de sensores e de algoritmos, sendo compatível com uma grande quantidade de plataformas *hardware* (Quigley et al., 2009). O ROS é uma ferramenta *open-source* que conta com uma vasta comunidade ativa que disponibiliza várias bibliotecas e ferramentas para diferentes aplicações, reduzindo significativamente o tempo de desenvolvimento (Koubaa et al., 2017).

Como computador de bordo foi utilizada a *NVIDIA Jetson Tx2*, pois já tinha sido utilizada em projetos anteriores no CIAFA. A *NVIDIA Jetson Tx2* é um módulo de computação desenvolvido para sistemas autónomos, fornecendo um elevado desempenho e, simultaneamente, tendo uma elevada eficiência energética (NVIDIA, n.d.). Este computador de bordo foi utilizado com o módulo de suporte *Astro Carrier*, sendo este robusto, e tendo já sido aplicado em diversos ambientes, como UAS e veículos autónomos. É possível ser observado na Figura 3.



Figura 2: Pixhawk 6c

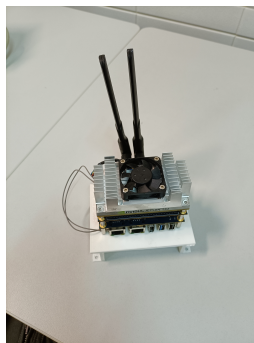


Figura 3: NVIDIA Jetson Tx2

#### 4.1.1 Escolha do sensor e algoritmo de localização

Inicialmente foi testada a câmara *Intel Realsense D435i*. Esta pode ser observada na Figura 4. Esta é uma câmara estereoscópica capaz de obter, de uma forma direta, a profundidade do meio ambiente onde se encontra. Possui um IMU o que irá facilitar a sua implementação num sistema de localização (Corporation, n.d.). Como algoritmo de localização, foi utilizado um algoritmo de SLAM, o RTAB-MAP, sendo que este, para além de fazer uso das imagens RGB-D, também utiliza as informações provenientes do IMU integrado na câmara estereoscópica, executando a fusão das duas informações. Adicionalmente, este algoritmo SLAM oferece uma fácil integração com o ROS.

Foram posteriormente realizados testes preliminares, em que foi feito um mapeamento no qual o sistema iniciou numa sala, prosseguiu com o mapeamento ao longo de um corredor e, no final, retornou ao ponto inicial. Na Figura 5 é possível observar uma parte do mapeamento realizado pela câmara *Intel Realsense D435i*. Observou-se que a estimativa de localização do ponto inicial diferiu da do ponto final, embora ambos estivessem na mesma localização. Foi possível concluir também que a tecnologia de *Loop Closure* não desempenhou a sua função, no contexto em que estamos a operar, uma vez que mapeou o mesmo local duas vezes, o que criou uma distorção no mapa. Foi observado que a estimativa de localização sofre desvios e saltos inesperados durante o mapeamento e que, quando apontada para locais onde não existiam pontos de referência como paredes lisas, perdia a sua estimativa de localização, algo que poderia colocar em risco a utilização deste sistema em um UAS. Conclui-se desta forma que este sistema não seria viável para o desenvolvimento de um sistema de localização para a navegação de um UAS em ambientes sem recurso a GNSS.



Figura 4: Câmera *Intel Real-sense D435i*

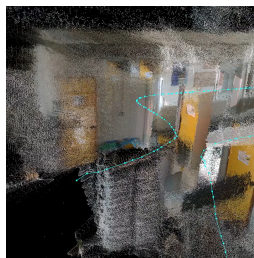


Figura 5: SLAM com a câmara *Intel Realsense D435i*

Posteriormente, foi testado um LIDAR, o *Livox Mid-360*. Este tem um campo de visão de 360° horizontal, e verticalmente consegue obter pontos entre os -7° e os 52°, em relação ao plano horizontal do mesmo (Limited, n.d.). Em conjunto com este LIDAR foi escolhido o FASTLIO, um algoritmo SLAM, visto que este é de fácil integração com os LIDAR do grupo

*Livox*, sendo um algoritmo *open-source*, capaz de fundir a informação proveniente da nuvem de pontos do LIDAR e da informação proveniente do IMU, através de um EKF iterativo, permitindo que exista uma diminuição de erros de localização e uma diminuição da distorção da nuvem de pontos criada (Vultaggio et al., 2023).

Foram realizados os testes preliminares à semelhança do que foi feito com a câmara estereoscópica e com o RTAB-MAP e foi observada uma elevada robustez do sistema, sendo possível identificar com facilidade o ambiente ao redor do LIDAR. Ademais, a localização do sistema foi atualizada com maior rapidez, não estando dependente das condições luminosas existentes nem da obtenção de pontos de referência. Quando comparado com o teste preliminar da câmara estereoscópica, podemos observar que a nuvem de pontos criada tem uma densidade superior, e não existem locais repetidos no mapeamento. O facto de conseguir obter a estimativa de localização sem elevados desvios e saltos ao longo da trajetória, indica que a fusão dos dados da nuvem de pontos e IMU, feita através do EKF iterativo do FASTLIO, está a ser executada com sucesso. Com base nestes testes preliminares, demonstrou-se ser viável avançar com a implementação deste sistema de localização para a navegação de um UAS sem recurso a GNSS.



Figura 6: *Livox Mid-360*

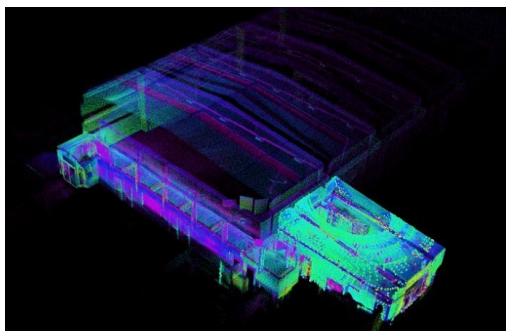


Figura 7: SLAM com LIDAR *Livox Mid-360*

## 4.2 Integração de componentes

### 4.2.1 Integração do sistema de localização com o UAS

Sendo que a plataforma escolhida para a execução dos testes de voo foi o *Holybro PX4 Development Kit - X500 v2*, esta não era compatível com a *NVIDIA Jetson Tx2*, nem com o LIDAR escolhido. Por isso, foram desenhadas e impressas peças 3D. Ficou definido que se iriam construir as mesmas peças de modo a que se instalassem diretamente por cima e por baixo do módulo do *Pixhawk*. Foi também definido que com esta implementação seria dada prioridade à estabilidade do UAS, em vez da sua manobrabilidade. Outro aspeto relevante é que o LIDAR necessita de maximizar o campo de visão, de modo a obter uma boa estimativa de localização e mapeamento. Foi tomada a decisão de que o LIDAR ficaria por baixo do *Pixhawk* virado para baixo, enquanto a *NVIDIA Jetson Tx2* ficaria por cima do *Pixhawk*.

### 4.2.2 Integração do sistema com o piloto automático

O sistema de localização, por forma a ser integrado na navegação de um UAS, necessita que os diferentes componentes computacionais possam comunicar entre si. Neste caso em particular, é necessário que os dados obtidos pelo LIDAR e processados pelo algoritmo SLAM sejam posteriormente comunicados ao piloto automático, de forma a que o mesmo tome as decisões necessárias para um voo seguro. O processo de comunicação utilizado é conhecido como MAVLink (*Micro Air Vehicle*).

Através do MAVROS, as informações de telemetria, como por exemplo o estado do UAS e a sua posição, podem ser convertidas em tópicos do ROS (*maavros - ROS Wiki*, n.d.), funcionando como um meio de comunicação entre os diferentes



módulos do sistema de localização, permitindo a troca de mensagens entre os mesmos (*ROS/Tutorials/UnderstandingTopics - ROS Wiki*, n.d.).

Sendo que o *payload* máximo do UAS é de 1500 g, foi verificado o peso de cada um dos componentes, tendo todos os componentes somados o peso de 1016 g, existindo uma margem de 484 g para o *payload* máximo.

## 4.2.3 Configuração do Pixhawk

O MAVROS por defeito, utiliza sinal GNSS (*mavros - ROS Wiki*, n.d.). Todavia, o objetivo deste sistema de localização é o seu funcionamento em condições onde não exista sinal GNSS e, por esse motivo, é necessário ajustar o MAVROS.

Assim, foi executado um *remaping* entre o tópico *Odometry* para o tópico *mavros/odometry/out*. permitindo que a informação do LIDAR seja utilizada pelo *Pixhawk* (*Using Vision or Motion Capture Systems for Position Estimation | PX4 Guide (main)*, n.d.). Uma vez que o LIDAR encontra-se virado para baixo, foi preciso realizar alterações na árvore de transformações do sistema de coordenadas, indicando a posição relativa do LIDAR em relação ao *Pixhawk*, de modo a que o piloto automático consiga utilizar a informação que está a ser fornecida pelo sistema de localização.

## 4.2.4 Montagem dos sistema de localização no UAS

Uma das peças 3D criadas foi montada na parte inferior do UAS. Esta possuía um apoio para o LIDAR e para uma bateria. O LIDAR foi ligado a uma bateria de modo a obter energia para o seu funcionamento e ligado à *NVIDIA Jetson Tx2* para ser fornecida a informação de profundidade ao algoritmo SLAM. Diretamente colocado na plataforma do UAS, encontrava-se o *Pixhawk 6c* que, através de um cabo *USB-c* era fornecida energia diretamente do computador de bordo. O *Pixhawk* encontrava-se ligado a uma antena de telemetria, que fornecia informação diretamente para a estação de controlo, onde era possível examinar os comportamentos do *Pixhawk*, e a um recetor rádio, que recebia informação de um controlador rádio. Na parte superior do UAS, foi instalada uma peça impressa em 3D que serviu de suporte para a *NVIDIA Jetson TX2*, que recebia energia através de uma bateria. Esta mesma bateria, para além de fornecer energia ao computador de bordo e ao LIDAR, também fornecia energia aos motores do UAS. Nas Figuras 4, 5 e 6 é possível verificar onde cada um dos componentes do sistema de localização foi colocado no UAS.

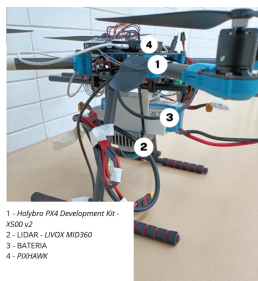


Figura 8: Imagem legendada dos componente na parte inferior do UAS



Figura 9: Imagem legendada dos componente da plataforma do UAS

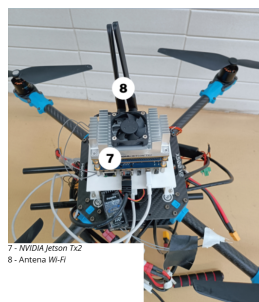


Figura 10: Imagem legendada dos componente da parte superior do UAS

## 4.2.5 Teste com o UAS com os motores desligados

Foram realizados dois tipos de testes, primeiro, um teste onde o objetivo seria averiguar a precisão deste sistema de localização no plano horizontal e, posteriormente, um teste para verificar a precisão do sistema de localização no plano vertical.

O teste para validar a precisão do sistema de localização no plano horizontal consistiu em transportar o UAS manualmente ao longo de um trajeto horizontal e verificar a precisão do sistema de localização no plano horizontal, bem como a



sua reação a pequenas variações de altura. Durante o teste, o sistema de localização processou os dados em tempo real e mapeou o ambiente ao redor do UAS, regressando, no final, ao mesmo local de onde o trajeto teve início.

O teste para validar a precisão do sistema de localização no plano vertical consistiu em transportar o UAS por um trajeto horizontal e, de seguida, elevá-lo a uma altura considerável, neste caso a uma plataforma de 2,7 m, de modo a verificar se a estimativa de posição fornecida pelo sistema coincide com a diferença de altura.

### 4.2.6 Testes de voo

Antes de cada voo, terão de ser executadas ações prévias de forma a garantir a segurança do voo. Inicialmente, a *NVIDIA Jetson TX2* foi acessada remotamente. Posteriormente, foi utilizado o *QGroundControl*, um *software* que funciona como estação de controlo de terra, desenvolvido para configurar e controlar os parâmetros do *Pixhawk*. Adicionalmente, pode-se alterar os modos de voo e configurar os canais do controlador de rádio através do mesmo. No *QGroundControl* os parâmetros do *Pixhawk* são ajustados para que o piloto automático utilize exclusivamente a informação proveniente do algoritmo SLAM, permitindo que o voo seja baseado apenas nesses dados (*QGroundControl User Guide | QGC Guide (master)*, n.d.). Também neste *software* foram criados *failsafes* e definido um canal do controlador rádio que funcionou de *kill switch*, sendo os referidos, mecanismos de segurança que são implementados com o intuito de proteger o UAS em situações de perigo durante o voo (*Safety (Failsafe) Configuration | PX4 Guide (main)*, n.d.).

No *QGroundControl* também é possível alterar os modos de voo. Todavia, para validar o sistema de localização foi utilizado o *Position Mode*. Este modo de voo é o mais seguro, pois é fixada a posição do UAS, assim que o piloto colocar a alavanca da potência numa posição fixa e não fornecer mais nenhum comando, o UAS irá estabilizar no local onde foi dado o último *input*, compensando qualquer força externa como o vento ou um possível desvio do centro de gravidade (*Position Mode (Multicopter) | PX4 Guide (main)*, n.d.).

### 4.2.7 Teste de voo com algoritmo de planeamento de trajetória

Caso os testes de voo estacionário revelassem resultados positivos e robustos, seria utilizado um algoritmo de deteção e desvio de obstáculos, que iria permitir que o UAS voasse de forma autónoma, apenas com a estimativa de localização. O algoritmo utilizado possibilita que, após o envio de uma posição pelo operador, o UAS se desloque para o local desejado, desviando-se de obstáculos estáticos ou em movimento, quer no interior de edifícios, quer no exterior (Kong, Xu, Cai, & Zhang, 2021).

## 5 Resultados

### 5.1 Resultados do teste manual no plano horizontal

Foi gravado o *rosbag* e inicialmente analisado, sendo possível observar qualquer tipo de falha de publicação de mensagens nos tópicos executados, em particular de onde a odometria é extraída, ou seja, do tópico *mavros/odometry/out*. Este tópico sofreu pequenas quebras, sendo as mesmas de 0,432 s, 0,574 s e 0,75 s, existindo uma taxa de publicação média de 10 Hz. Por outro lado, foi também verificado o tópico *mavros/local.position/pose* e apurado que não existe nenhuma falha, sendo publicado a uma taxa média de 30 Hz, sendo este o tópico a partir do qual o *Pixhawk* obtém a sua estimativa de localização. Apesar de existirem pequenas quebras no tópico *mavros/odometry/out*, a estimativa de localização do tópico *mavros/local.position/pose* não sofreu saltos nem desvios ao longo destes períodos de quebras.

Como foi referido na Implementação, foram comparadas as distâncias reais dos trajetos realizados com a estimativa de localização. Foi observado que ao ser executado o SLAM, os eixos de orientação nem sempre coincidiam com o percurso percorrido, ou seja, os eixos *x* e *y* não são paralelos ao solo e por isso à medida que o UAS se deslocou pelo trajeto, a estimativa de localização ao longo do eixo *z* variou. Foi observado que o maior erro, não relacionado com a posição dos eixos *x* e *y*, foi de 0,027 m. Também foi verificado que quando o UAS voltou à posição inicial, existiu apenas uma diferença de 0,009 m no eixo *x*, de 0,027 m no eixo *y* e de 0,008 m no eixo *z*. É possível afirmar que durante este teste existiu um mapeamento correto do ambiente envolvente.

## 5.2 Resultados do teste manual no plano vertical

Através do *rosbag* gravado durante o teste, observou-se que o tempo máximo que o tópico *mavros/odometry/out* demorou a ser publicado foi 0,113 s e a uma frequência média de 10 Hz. No caso do *mavros/local.position/pose* o tempo máximo que esse tópico demorou a ser publicado foi 0,041 s e a frequência média de publicação foi de 30,007 Hz.

Devido ao facto de os eixos não se encontrarem diretamente alinhados com a trajetória executada, antes de subir para a plataforma de 2,7 m, o UAS foi colocado no chão com o objetivo de verificar a diferença de estimativas de posição. Posteriormente, seria comparada a diferença de estimativas de posição ao longo do eixo z comparativamente com a altura da plataforma.

A posição inicial e final foram iguais. A diferença da estimativa de posição no eixo x foi de 0,0238 m. No eixo y foi de 0,0016 m e no eixo z de 0,0626 m. A diferença da estimativa de posição do chão e de quando colocado no topo da plataforma ao longo do eixo z, foi de 2,7352 m. Sendo que a altura da plataforma é de 2,7 m, existiu um erro de estimativa de localização de 0,0352 m.

Também foi constatado, por meio da análise do *rosbag*, que durante a execução do teste, o algoritmo SLAM construiu o mapa do ambiente envolvente de maneira consistente com a realidade.

Devido aos resultados verificados, foi concluído que seria viável avançar com os testes de voo, pois a estimativa de localização fornecida possuía uma elevada precisão.

## 5.3 Análise do teste de voo com resultados positivos

Foram analisados os testes de voo com resultados favoráveis, verificando-se que não existiram desvios na posição no plano horizontal quando o UAS foi colocado em *Position Mode*.

Durante o voo, foi observado que, com diferentes comandos de potência, o UAS não variou a sua posição ao longo do plano horizontal, compensando dessa forma um possível desequilíbrio do centro de gravidade. Isto foi corroborado após a análise do *rosbag*, em que foi observado um desvio no eixo x de apenas 0,062 m entre a posição inicial e a posição final, e no eixo y de 0,043 m. No entanto, apesar de ter sido observado este comportamento positivo neste teste de voo, foram executados outros testes de voo em que o comportamento observado não foi o esperado.

## 5.4 Análise de falhas nos testes de voo

Nestes testes de voo, apesar dos resultados do sistema de localização serem satisfatórios, alguns comportamentos por parte do UAS foram inesperados. Estas falhas evidenciam desafios associados à integração do sistema de localização com o UAS.

Observou-se que, logo no início do voo, o UAS girava sobre o eixo z, ou seja, sobre ele mesmo. No entanto, o UAS deveria permanecer imóvel, pois estava a voar em *Position Mode*. Foi também observado que, em certos momentos do voo, o UAS aumentava de forma autónoma a quantidade de potência dos motores, com mais frequência na descida.

Através do *rosbag* gravado durante os voos foi possível observar por meio do tópico *mavros/local.position/pose*, a constante variação do *Yaw*. Verificou-se que, durante os picos de potência dos motores, a estimativa de localização ao longo do eixo z apresenta oscilações, especialmente perceptíveis durante as descidas. Esses picos estão diretamente relacionados aos momentos de maior potência do motor.

Posteriormente, foi observado o *log* de voo do *Pixhawk* e primeiramente foi analisada a posição do UAS no plano horizontal. Foi verificado que no plano horizontal, o UAS não permaneceu na mesma posição, como era suposto acontecer. Foi confirmado que não foi fornecido qualquer comando no controlador rádio, que pudesse justificar a origem dos picos de potência nem da rotação do UAS sobre ele mesmo.

### 5.4.1 Medidas implementadas

Para reduzir o comportamento evidenciado por parte do UAS, foram tomadas algumas medidas. Primeiramente, foi trocado um dos braços do UAS, assim como o respetivo motor, de modo a verificar se existia uma falha. De seguida, foi alterado o *Pixhawk 6c* utilizado, para verificar se existia uma falha no piloto automático. Foi instalado o *Firmware* do *Pixhawk* mais recente. Foram calibrados os acelerómetros, giroscópios, bússolas, foram *Eletronic Speed Control* (ESC)

dos motores e desligados os magnetômetros, tendo sido utilizada apenas a estimativa do SLAM para orientação, pois poderiam estar a influenciar os comandos de potência do motor. À medida que iam sendo realizadas estas alterações, era realizado um teste de voo. No entanto, o comportamento por parte do UAS manteve-se.

### 5.4.2 Conclusão da análise de falhas nos testes de voo

Em suma, o sistema de localização está funcional, e o *Pixhawk* recebe as informações de localização. No entanto, por algum motivo, possivelmente devido a um problema elétrico no UAS, a um motor com limitações ou até mesmo a uma falha na placa de energia, o UAS não conseguiu manter a posição de forma consistente no *Position Mode*.

O passo seguinte para tentar contrariar os comportamentos inesperados por parte do UAS passaria por realizar testes mais extensos aos ESC, motores e placa de energia. Ensaios de voo com outro conjunto de equipamentos eletrônicos não foram realizados por falta de equipamentos suplentes.

Devido ao facto de não existir robustez ao longo dos testes em *Position Mode*, não foi utilizado o algoritmo de detecção e desvio de obstáculos em um ensaio de voo.

## 6 Conclusões

O avanço da tecnologia relacionada com os UAS, tem apresentado um aumento significativo, por ter diversas aplicações, tanto no ambiente civil como no ambiente militar. Têm sido desenvolvidos sistemas autónomos que permitem o reconhecimento em ambientes onde não é possível a presença humana ou em ambientes onde a presença humana acarreta elevados riscos, como é o caso de ambientes hostis. Na atualidade, a maioria dos UAS baseia a sua navegação em sinal GNSS. Contudo, existem diversos locais onde este sinal é reduzido ou inexistente e altamente suscetível a interferências, como é o caso do interior de edifícios ou em ambientes onde exista *jamming* de sinal GNSS. Deste modo, a navegação do UAS é colocada em causa, sendo por isso necessário criar uma navegação autónoma.

Nesta dissertação de tese de mestrado, foi desenvolvido um sistema de localização, para a navegação autónoma de um UAS de pequenas dimensões, em ambientes onde não existe sinal GNSS, com o objetivo de ser implementado em um UAS. Assim, foram utilizados um LIDAR, que tem a capacidade de calcular a profundidade de forma direta, e foi aplicado um algoritmo SLAM, o FASTLIO. Este permite mapear um local desconhecido enquanto tem a capacidade de se localizar no mapa construído, à medida que o UAS se desloca. Para garantir que o UAS tivesse a capacidade de processar a informação, foi necessário incorporar um computador de bordo, a *NVIDIA Jetson Tx2*. Neste computador de bordo está instalado o ROS, que permite a comunicação dos diversos componentes deste sistema, em tempo real.

Para verificar a integração do sistema de localização com um sistema de navegação, é necessário avaliar a comunicação de dados com o piloto automático, neste caso, o *Pixhawk*. Para garantir a comunicação entre o sistema de localização e o *Pixhawk* foi utilizado o MAVROS. Foram editados os vários parâmetros de modo a que o *Pixhawk* baseasse a sua obtenção de localização através da estimativa visual que estava a ser produzida pelo FASTLIO. Este sistema de localização foi posteriormente implementado em um UAS.

Inicialmente, foram realizados testes de forma manual com os motores desligados, onde foi averiguada a precisão da estimativa de localização ao longo dos 3 eixos. Os valores dos erros dos testes realizados revelam precisão e robustez do sistema, pois ao longo dos testes nunca foi perdida a estimativa de posição, como foi possível constatar através da avaliação da publicação de mensagens nos tópicos responsáveis por fornecer a estimativa de localização.

Foram também realizados testes de voo em que o objetivo foi avaliar a viabilidade da implementação deste sistema de localização em UAS, para verificar se o piloto automático era capaz de processar as mensagens provenientes do FASTLIO de forma a poder localizar-se. Como tal, o UAS foi colocado em *Position Mode* e verificado se apenas com a informação proveniente do sistema de localização, este iria ser capaz de permanecer em voo estacionário, corrigindo qualquer desvio ou posição inadequada do centro de gravidade. Foi possível observar em apenas um dos testes que o UAS permaneceu em voo estacionário, existindo inicialmente pequenos desvios ao longo do plano horizontal, mas sendo os mesmos rapidamente corrigidos. Apesar de nos restantes testes, o UAS não ter mantido a posição, foi verificado que o sistema de localização estava a funcionar. Após análise dos vários parâmetros de voo, e aplicação de medidas de correção, não foi identificada uma causa concreta para o comportamento observado, sendo necessários mais testes.

Em síntese, o voo do UAS apresenta desafios significativos para a implementação de um sistema de localização. Contudo, o sistema desenvolvido demonstrou robustez e precisão ao longo dos testes, revelando-se uma opção viável para a integração num sistema de navegação de UAS em condições onde não há sinal GNSS.

Um dos principais desenvolvimentos que este sistema de localização permite, é a criação de um sistema de navegação autónomo baseado em IA (Inteligência Artificial), sendo que a integração destes algoritmos permite que o UAS não estime apenas a sua localização, mas também seja capaz, em tempo real, de tomar decisões de forma autónoma durante o voo. Um exemplo seria a aplicação deste sistema de localização a um outro algoritmo que seria capaz de identificar obstáculos dinâmicos e ajustar a sua trajetória em tempo real, decidindo de forma autónoma qual a trajetória mais segura e eficiente. Este sistema iria permitir a operação do UAS independente da intervenção humana, aumentando a sua eficácia em cenários imprevisíveis.

A criação de sistemas de navegação autónoma baseados em IA iria ser uma vantagem estratégica no contexto militar, onde os UAS passariam a ter a capacidade de operar, com maior autonomia, em ambientes hostis. Este seria um elemento essencial na execução de missões de vigilância, reconhecimento e apoio às operações em zona de combate, aumentando a segurança das forças envolvidas e reduzindo a exposição dos recursos humanos a situações de elevado risco.

Para trabalhos futuros, seria vantajoso estudar a integração deste sistema com outros tipos de sensores, visto que este sistema não oferece qualquer informação quanto à cor do meio ambiente envolvente. Em suma, a partir deste sistema de localização, foi criada uma base sólida para o desenvolvimento de UAS completamente autónomos, capazes de operar em ambientes desafiadores, com elevada eficiência e precisão.

## Referências Bibliográficas

- Aldao, E., González-de Santos, L. M., & González-Jorge, H. (2022). Lidar based detect and avoid system for uav navigation in uam corridors. *Drones*, 6(8), 185.
- Al-Tawil, B., Hempel, T., Abdelrahman, A., & Al-Hamadi, A. (2024). A review of visual slam for robotics: evolution, properties, and future applications. *Frontiers in Robotics and AI*, 11, 1347985.
- Arshad, S., & Kim, G.-W. (2021). Role of deep learning in loop closure detection for visual and lidar slam: A survey. *Sensors*, 21(4), 1243.
- Bhoi, A. (2019). Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*.
- Bosse, M., Zlot, R., & Flick, P. (2012). Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5), 1104–1119.
- Chen, S., Chang, C.-W., & Wen, C.-Y. (2020). Perception in the dark; development of a tof visual inertial odometry system. *Sensors*, 20(5), 1263.
- Corporation, I. (n.d.). *Intel realsense depth camera d435i*. Retrieved 2024-12-09, from <https://www.intelrealsense.com/depth-camera-d435i/> (Acesso em: 9 de dezembro de 2024)
- Gao, B., Lang, H., & Ren, J. (2020). Stereo visual slam for autonomous vehicles: A review. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 1316–1322).
- Hening, S., Ippolito, C. A., Krishnakumar, K. S., Stepanyan, V., & Teodorescu, M. (2017). 3d lidar slam integration with gps/ins for uavs in urban gps-degraded environments. In *Aiaa information systems-aiaa infotech@ aerospace* (p. 0448).
- Holybro pixhawk 6c | PX4 guide (main). (n.d.). Retrieved 2024-12-21, from [https://docs.px4.io/main/en/flight\\_controller/pixhawk6c.html](https://docs.px4.io/main/en/flight_controller/pixhawk6c.html)
- Huynh, L., Nguyen-Ha, P., Matas, J., Rahtu, E., & Heikkilä, J. (2020). Guiding monocular depth estimation using depth-attention volume. In *Computer vision—eccv 2020: 16th european conference, glasgow, uk, august 23–28, 2020, proceedings, part xxvi 16* (pp. 581–597).
- Jeon, J., Jung, S., Lee, E., Choi, D., & Myung, H. (2021). Run your visual-inertial odometry on nvidia jetson: Benchmark tests on a micro aerial vehicle. *IEEE robotics and automation letters*, 6(3), 5332–5339.
- Kong, F., Xu, W., Cai, Y., & Zhang, F. (2021). Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots. *IEEE Robotics and Automation Letters*, 6(4), 7869–7876.
- Koubaa, A., et al. (2017). *Robot operating system (ros)*. (Vol. 1). Springer.

- Li, L., et al. (2014). Time-of-flight camera—an introduction. *Technical white paper*(SLOA190B). Limited, L. T. C. (n.d.). *Livox mid-360 lidar sensor*. Retrieved 2024-12-09, from <https://www.livoxtech.com/mid-360> (Acesso em: 9 de dezembro de 2024)
- mavros - ROS wiki*. (n.d.). Retrieved 2024-12-10, from <https://wiki.ros.org/mavros>
- Milijasic, R., Markovic, L., Ivanovic, A., Petric, F., & Bogdan, S. (2021). A comparison of lidar-based slam systems for control of unmanned aerial vehicles. In *2021 international conference on unmanned aircraft systems (icuas)* (pp. 1148–1154).
- Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5), 1147–1163.
- Mustafah, Y. M., Azman, A. W., & Akbar, F. (2012). Indoor uav positioning using stereo vision sensor. *Procedia Engineering*, 41, 575–579.
- NVIDIA. (n.d.). *Jetson tx2 module*. Retrieved 2024-12-09, from <https://developer.nvidia.com/embedded/jetson-tx2> (Accessed on December 9, 2024)
- Petrlik, M., Krajník, T., & Saska, M. (2021). Lidar-based stabilization, navigation and localization for uavs operating in dark indoor environments. In *2021 international conference on unmanned aircraft systems (icuas)* (pp. 243–251).
- Position mode (multicopter) | PX4 guide (main)*. (n.d.). Retrieved 2024-12-12, from [https://docs.px4.io/main/en/flight\\_modes\\_mc/position.html](https://docs.px4.io/main/en/flight_modes_mc/position.html)
- PX4 development kit - x500 v2*. (n.d.). Retrieved 2024-12-10, from <https://holystone.com/products/px4-development-kit-x500-v2>
- QGroundControl user guide | QGC guide (master)*. (n.d.). Retrieved 2024-12-11, from <https://docs.qgroundcontrol.com/master/en/qgc-user-guide/index.html>
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... others (2009). Ros: an open-source robot operating system. In *Icra workshop on open source software* (Vol. 3, p. 5).
- Roch, J., & Najjaran, H. (2023, January). *DOPESLAM: A high-precision ROS-based semantic SLAM system*. Research Square. Retrieved 2024-11-13, from <https://www.researchsquare.com/article/rs-1895628/v2> (ISSN: 2693-5015) doi: 10.21203/rs.3.rs-1895628/v2
- ROS/tutorials/UnderstandingTopics - ROS wiki*. (n.d.). Retrieved 2024-12-10, from <https://wiki.ros.org/ROS/Tutorials/UnderstandingTopics>
- Safety (failsafe) configuration | PX4 guide (main)*. (n.d.). Retrieved 2024-12-14, from <https://docs.px4.io/main/en/config/safety.html>
- Sanchez-Rodriguez, J.-P., & Aceves-Lopez, A. (2018). A survey on stereo vision-based autonomous navigation for multi-rotor muavs. *Robotica*, 36(8), 1225–1243.
- Suriano, F. (2021). *Study of slam state of the art techniques for uavs navigation in critical environments* (Unpublished doctoral dissertation). Politecnico di Torino.
- Taheri, H., & Xia, Z. C. (2021). Slam; definition and evolution. *Engineering Applications of Artificial Intelligence*, 97, 104032.
- Troll, P., Szipka, K., & Archenti, A. (2020). Indoor localization of quadcopters in industrial environment. In *9th swedish production symposium, sps 2020* (pp. 453–464).
- Using vision or motion capture systems for position estimation | PX4 guide (main)*. (n.d.). Retrieved 2024-12-10, from [https://docs.px4.io/main/en/ros/external\\_position\\_estimation.html](https://docs.px4.io/main/en/ros/external_position_estimation.html)
- Vultaggio, F., d'Apolito, F., Sulzbachner, C., & Fanta-Jende, P. (2023). Simulation of low-cost mems-lidar and analysis of its effect on the performances of state-of-the-art slams. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 539–545.
- Yang, J.-C., Lin, C.-J., You, B.-Y., Yan, Y.-L., & Cheng, T.-H. (2021). Rtlto: Real-time lidar-inertial odometry and mapping for uavs. *Sensors*, 21(12), 3955.

# User Interface for UAV Object Detection and Geolocation



**Autor: ALF/ENGEL Tiago André Martins Vera-Cruz de Abreu**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

**Orientador: António Manuel Raminhos Cordeiro Grilo, Professor Associado**  
INESC ID – Instituto de Engenharia de Sistemas e Computadores:  
Investigação e Desenvolvimento em Lisboa, Lisboa  
Instituto Superior Técnico, Lisboa

### Abstract

The work presented in this paper, funded by KPMG Portugal, and done in collaboration with NOS and Vodafone Portugal, is a component of a Capstone Project to develop a prototype of a 5G-enabled Unmanned Aerial Vehicle (UAV) for two principal applications: restricted area surveillance and infrastructure inspection. This thesis is specifically concerned with the development of a web application that provides consumers with access to real-time geolocation and intruder detection data. Despite the obstacles that prevented the thesis's initial goals from being fully realized, extensive research was conducted on potential hardware and software solutions for this purpose. The most suitable components for the proposed system were selected based on this comprehensive analysis. To ensure the optimal functioning of the selected systems, a comprehensive testing procedure was then undertaken, considering the constraints of the design process. While not achieving all the initial objectives, this study significantly advances the field of UAVs in surveillance applications, providing a robust foundation for future advancements in this domain. The resulting web application represents a key step in the integration of detection and geolocation technologies into unmanned aerial surveillance systems.

**Keywords:** Unmanned Aerial Vehicle Prototype, Internet of Drones, People Detection and Geolocation, User Interface Web App, Capstone Project.

## 1 Introduction

This work is part of the Capstone Project, a multi-company and multidisciplinary project and it was proposed and funded by KPMG Portugal, a global network of independent companies that offer audit, tax, and consultancy services. It was being conducted in collaboration with NOS and Vodafone Portugal, two telecommunications operators that offer services.

The students who comprise the project are Francisco Alves assigned to develop the conceptual and structural design analysis and also prototype the UAV model; Sasha Mot assigned to enhancement the aerodynamic and propulsion systems designs; Bernardo Albano assigned to the design, and hardware selection, implementation, and testing of avionics, auxiliary external support, and communication payload systems and Tiago Vera-Cruz assigned to implement the computer vision system for intrusion detection and web application for the user interface.

UAVs can monitor and patrol the airspace around infrastructure, detecting potential threats. Real-time network activity monitoring can be provided by UAVs, which can alert operators to any suspicious activities or anomalies (Abdalla, Powell, Marojovic, & Geraci, 2020). On the other hand, the telecommunications industry has rapidly adopted UAVs due to their benefits, including increased efficiency, cost savings, and safety. The increasing utilization of drones in this sector is primarily propelled by technological advancements resulting from investments made by major industry players, both in military and commercial contexts (Silver, Mazur, Wiśniewski, & Babicz, 2017).

The overarching objective of the Capstone Project is to develop a UAV capable of fulfilling two distinct operational roles: surveillance and security, and mobile network detection and infrastructure inspection. The specific objective of this thesis is to develop a detection and geolocation system for objects (people and vehicles); to develop an infrastructure inspection system; to develop of a web application system interface for the user; and to create an image and video database for real-time analysis and post-operation of the aerial system.

## 2 Internet of Drones

The Internet of Things (IoT) can be defined as a global network infrastructure that is self-configuring and based on standard and interoperable communication protocols. It facilitates the provision of sophisticated services by establishing connections between physical and virtual entities through the utilization of intelligent interfaces, which both exist and evolve, and which are integrated into the information network. The IoT offers a range of services to diverse applications, leveraging capabilities in identification, data capture, processing, and communication while upholding the confidentiality of user data. This system comprises both the network infrastructure and the integration of diverse objects into a unified framework (Fraga-Lamas, Lopes, & Fernández-Caramés, 2021). The IoT is extended into the atmosphere through the Internet of Drones (IoD), creating a global network of IoT sensors and UAVs. This enables the monitoring of atmospheric conditions

across the globe, facilitating the formation of a highly interconnected global community and providing individuals with a range of benefits (Beekman, 2017).

The IoD represents a revolutionary shift of the paradigm in UAV technology, integrating UAVs into a networked ecosystem that enables seamless communication, control, and data exchange. This innovative concept is structured around a five-layer architecture, comprising the following layers: the UAV layer, the communication layer, the network layer, the cloud layer, and the client layer. Each layer is of critical importance in creating a robust and efficient system that can support a diverse range of applications (Derhab et al., 2023).

As IoD continues to evolve, it promises to transform various industries and sectors by leveraging the power of interconnected UAVs. The UAV layer provides the physical foundation, while the Communication layer, particularly through 5G networks, ensures high-speed, low-latency connectivity. The Network layer manages traffic and routing, the Cloud layer offers scalable computing and storage resources, and the Client layer serves as the interface for end-users and applications. Collectively, these layers constitute a comprehensive ecosystem that optimizes the potential of UAV technology, thereby paving the way for innovative solutions and services (Derhab et al., 2023).

In (Derhab et al., 2023), the authors present an architectural framework of the IoD ecosystem, which employs a multilayer system paradigm. Each layer is associated with a system actor, and the proposed architectural model, illustrated in Figure 1, is generic and can be implemented on any IoD platform. The architectural design comprises five principal layers: Unmanned Aerial Vehicle, Communication, Network, Cloud and Client.

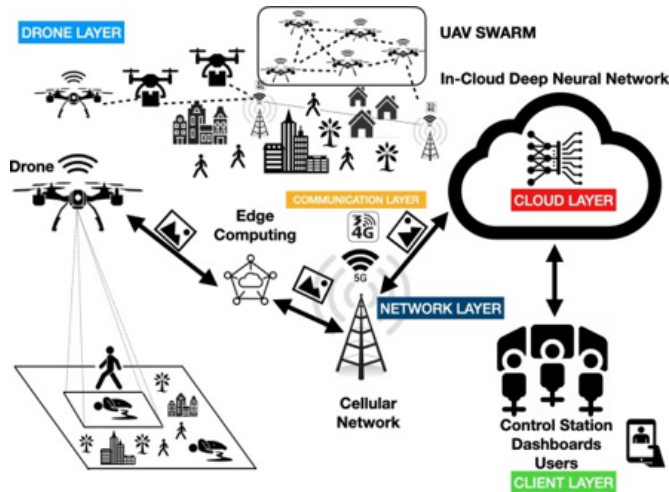


Figure 1: Internet-of-Drones Architecture (Derhab et al., 2023).

### 3 Capstone Project

This chapter examines the concept of the Capstone Project, and it will examine the scenarios that students have created and researched to operate a UAV prototype.

#### 3.1 Description

As previously stated, this work constituted a component of a Capstone Project, which was a multidisciplinary, multi-company financial endeavor supported by KPMG (around 7000€) and in collaboration with NOS and Vodafone. The project's objectives were to develop a UAV prototype that could communicate with telecommunications equipment and



to construct an advanced graphical user interface, analytical data models, and a data collection infrastructure. Each of the previously stated students was responsible for developing their work, which would be integrated with the collective project. The characteristics of UAVs, including weight, autonomy, dimensions, and others, were discussed in group to facilitate collective input and, when necessary, enable individuals to adapt their work to align with the project's objectives. Integrating individual and group efforts was a crucial aspect of the Capstone Project.

These students conducted the development of three UAV action scenarios during the Projeto Integrador de 2º Ciclo (PIC2) course through the investigation of concepts for this Capstone Project in collaboration with industry partners as these scenarios serve as the foundation for the project, delineating the requirements that the UAS must fulfill the mission in each situation.

### 3.2 Concept of Operations

This section will present the three scenarios that have been the subject of study and will set out the actions that the UAV is required to perform in each case. The first scenario focuses on security surveillance in industrial areas, where the UAV is employed for intruder detection and perimeter monitoring. The second scenario involves mobile network infrastructure assessment, using the UAV to inspect and evaluate cellular towers and addresses emergencies in remote areas, where the UAV is deployed to rapidly assess damage and temporarily replace an inactive telecommunication tower to restore vital communication links

#### 3.2.1 Scenario 1 – Surveillance in Industrial Areas

In this scenario, a UAV is deployed to enhance security surveillance in a restricted large-size industrial area, specifically the Sines Refinery, located in Portugal's Setúbal region. The principal objective is to prevent unauthorized access and detect potential intruders, thereby ensuring critical infrastructure protection and operational safety maintenance. This sophisticated security system capitalizes on the UAV's mobility and sophisticated sensing capabilities to provide comprehensive coverage of the facility, offering a more efficient and cost-effective alternative to traditional ground-based surveillance methods.

The UAV is equipped with a camera system that is capable of real-time person detection, which works in conjunction with an NB-IoT or LTE Machine Type Communication (LTE-M) tag monitoring system. Personnel granted access to the refinery are provided with NB-IoT or LTE-M tags, thus establishing a dual-layer verification process. By correlating visual detection with tag information, the system can distinguish accurately between authorized workers and potential intruders. Suppose an individual is detected without a corresponding tag generated signal. In that case, the system immediately alerts security personnel, enabling a rapid response to potential security breaches and significantly enhancing the overall safety and security of the industrial complex.

#### 3.2.2 Scenario 2 - Mobile Network Infrastructure Assessment

Based on the Lisbon area, the UAV is employed to perform anomaly detection in the mobile network coverage and assessment in telecommunications infrastructures. When a fault is reported or detected in the network, the UAV is promptly dispatched from the operations center to the affected location, thereby facilitating a swift and effective preliminary examination and inspection.

Upon reaching the error site, the UAV conducts a comprehensive visual inspection using its onboard camera systems and sensors. Simultaneously, it performs real-time analysis of the collected data, leveraging Artificial Intelligence and Machine Learning algorithms to identify and categorize potential issues. The UAV's findings are streamed live to the remote operator, who can access both the visual feed and the automated analysis results via an appropriate interface.

#### 3.2.3 Scenario 3 – Telecom Assessment and Link Re-establishment

In the wake of natural disasters, rapid assessment and restoration of critical infrastructure are crucial for effective emergency response. This scenario employs UAVs to evaluate telecommunication networks rapidly and comprehensively in affected areas. Upon deployment, the UAV performs an initial aerial inspection of the disaster zone, utilizing advanced

imaging technologies and real-time data analysis to assess the extent of damage to communication towers and associated infrastructure.

Following the initial assessment, the UAV transitions into its secondary role as a temporary network restoration solution. Equipped with compact, yet powerful communication relay technology, the UAV can function as an airborne telecommunication node. By positioning itself strategically, the UAV can effectively replace an inactive tower, re-establishing vital communication links between surviving network infrastructure.

## 4 Hardware and Software Selection

The analysis of various UAV components, computer vision algorithms, and web application frameworks revealed the most suitable technologies for the proposed solution. Each selection was driven by the specific needs of the project, balancing performance, feasibility, and future development potential.

The NVIDIA Jetson Orin Nano Developer Kit was selected for the onboard computer due to its robust processing power and compatibility with deep learning applications. The chosen hardware is well-suited to the demands of real-time computer vision tasks, which are fundamental to both person detection and infrastructure inspection. The integration of the hardware mentioned above into the UAV system ensures that the prototype can manage the requisite computational load for these operations without compromising its mobility or energy efficiency. The Jetson Orin Nano Developer Kit offers a comprehensive set of interfaces and connectivity options, rendering it highly suitable for integration into UAV systems. Moreover, these features facilitate seamless integration into UAV platforms, allowing the development of sophisticated, AI-driven systems (NVIDIA, 2023b).

The camera choice, the SIYI ZR10 gimbal camera, was made based on its ability to provide high-quality video and image stabilization, which is crucial for accurate object detection in dynamic environments. This camera allows for smooth and clear capture of images, which are essential inputs for the subsequent stages of computer vision processing, ensuring reliable performance in diverse operational conditions.

In terms of the object detection algorithm, NVIDIA DeepStream SDK, was chosen because it is a flexible Software Development Kit (SDK) that supports various object detection pre-trained models and algorithms optimized for the Jetson platform. As this project is a proof-of-concept, the focus is on demonstrating the feasibility of the system without developing new detection algorithms. By leveraging DeepStream, the project can achieve state-of-the-art performance in detecting people while minimizing development time. This choice aligns with the project's goals, emphasizing rapid deployment using reliable, well-optimized tools.

However, when exploring infrastructure inspection, no open-source, ready-to-use algorithms were identified. As a result, this area remains underdeveloped in the current scope of the project. Despite this limitation, the analysis highlights a clear opportunity for future research and development. Exploring customized or open-source solutions for infrastructure inspection could expand the system's functionality, offering greater versatility for broader UAV applications.

For the web application framework, Flask was chosen due to its simplicity and versatility. Flask enables quick deployment and easy integration with the broader system, facilitating the development of a user-friendly interface for real-time data visualization and control. Its flexibility makes it an ideal choice for creating a web-based interface without unnecessary complexity, ensuring the prototype remains lightweight yet functional.

In summary, the choices made in this project reflect a careful balance between using pre-existing technologies and leaving room for future innovation. Each decision was made with the proof-of-concept nature of the project in mind, focusing on reliability, compatibility, and ease of use, while also identifying gaps where future developments can push the boundaries of UAV applications for both person detection and infrastructure inspection.

## 5 Project Methodology and Design

The people detection and geolocation system is comprised of three interrelated components: object detection with the NVIDIA DeepStream SDK, an estimation of the computational capacity required to geolocate the individual based on the distance to the UAV camera, and the location of the person's GPS. When multiple objects are present in the camera's field

of view, these two settings facilitate more precise object detection and recognition. The final element of the detection and geolocation system is the NB-IoT tag, which is affixed to everyone permitted to be in the restricted area.

Integrating these three elements enables the system to identify individuals, determine their GPS location, and compute their distance. The utilization of the GPS position and the NB-IoT tag enables the implementation of screening procedures, the identification of individuals within the field of vision, and the subsequent assessment of their NB-IoT tag status. Provided that they are duly accredited as employers, individuals in possession of the requisite identification credentials are thereby permitted to enter the designated area. Those who have not been identified by a tag are considered as trespassers.

The web application allows authorized users to access the system data. Following the input of their login credentials, users can enjoy live streaming with DeepStream-performed detection and geolocation of detected items. Additionally, the GPS positions of each NB-IoT tag are displayed on a map of the restricted area where the UAV operates.

5.1 Object Detection via NVIDIA DeepStream

NVIDIA DeepStream is an SDK designed for real-time streaming analytics, particularly in the realm of computer vision and object detection, enabling efficient and scalable AI-based visual analytics. DeepStream integrates seamlessly with deep learning frameworks to perform object detection tasks with high accuracy and speed (Abdulghafoor & Abdullah, 2021). It is worth noting that DeepStream is highly extensible since it enables real-time object detection on live video streams with various camera inputs, including USB cameras, CSI cameras, IP cameras streaming over RTSP, or even pre-recorded video files. There are no specific camera requirements beyond compatibility with these standard input methods, making DeepStream flexible for different deployment scenarios. When implementing NVIDIA DeepStream for person detection, the system processes input video streams through a series of stages, including video decoding, object detection, tracking, and visualization (NVIDIA, 2023a).

The process begins when the DeepStream application from the terminal is executed. This command initializes the DeepStream pipeline, which is built on GStreamer, a powerful multimedia framework. The pipeline for live stream object detection consists of the stages exposed in Figure 2.

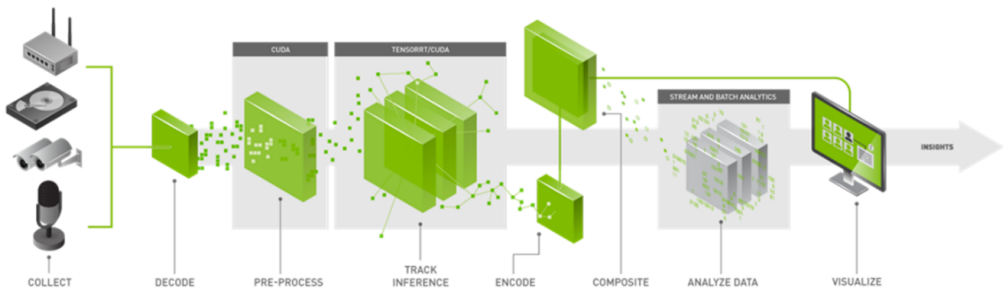


Figure 2: DeepStream SDK Workflow (NVIDIA, 2023a).

The process begins with video input capture. After DeepStream uses hardware-accelerated decoders like NVDEC to efficiently convert the compressed video into raw frame data. The raw frames undergo pre-processing using CUDA-accelerated operations. If enabled, the 'nvtracker' plugin performs multi-object tracking. It associates detections across frames, assigning consistent IDs to tracked objects, and understanding object movement and persistence in the video stream. The pre-processed frames are then passed to the inference engine, typically using the 'nvinfer' plugin, leveraging NVIDIA TensorRT for optimized GPU-accelerated inference. If the output needs to be streamed or saved, the processed frames are encoded using hardware-accelerated encoders (NVIDIA, 2023a).

For multi-stream applications, DeepStream can gather multiple processed streams into a single output. This is useful for creating video walls or tiled displays of multiple camera feeds. Postprocessing of inference results occurs at this stage. This may include filtering detections based on confidence thresholds, applying non-maximum suppression, or extracting specific

metadata from the detections for further analysis. The 'nvdsosd' (On-screen Display) plugin overlays visual elements on the processed frames. This typically includes drawing bounding boxes around detected objects, displaying labels, and showing tracking IDs if tracking is enabled, overlaying the detection results onto the original video frames (NVIDIA, 2023a).

## 5.2 Object Geolocalization Estimation

Object detection and depth estimation are crucial components in computer vision, enabling machines to perceive and understand their environment in three dimensions. When object detection algorithms are combined with depth estimation techniques, these systems can determine the 3D positions of detected objects relative to the UAV.

### 5.2.1 Camera Calibration

Before using the camera in the experiment, it must undergo calibration to determine the camera parameters. The camera model used for calibration is the pinhole model, the most basic camera model, which establishes the mathematical connection between the picture plane and the projection of points in the three-dimensional environment.

This model calibration is based on the process of calculating coordinate transformation matrices, which are essential for accurate image processing: Extrinsic and Intrinsic Matrices. The Intrinsic Matrix, expressed on  $3 \times 3$  K matrix, transforms the 3D camera coordinate system to the 2D image coordinate system, while the Extrinsic Matrix transforms the 3D world coordinate system to the 3D camera coordinate system (Hartley & Zisserman, 2004; Li & Yoon, 2023; Sanyal, Bhushan, & Sivayazi, 2020). Figure 3 allows us to have a perspective of the problem. The calculation of the camera calibration using the pinhole model is made with more detail in the Dissertation document (Vera-Cruz, 2024).

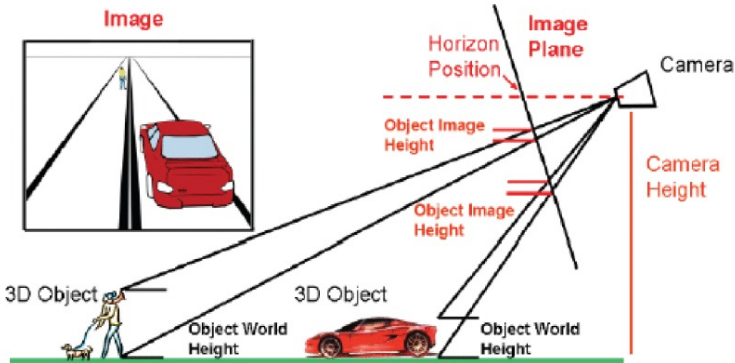
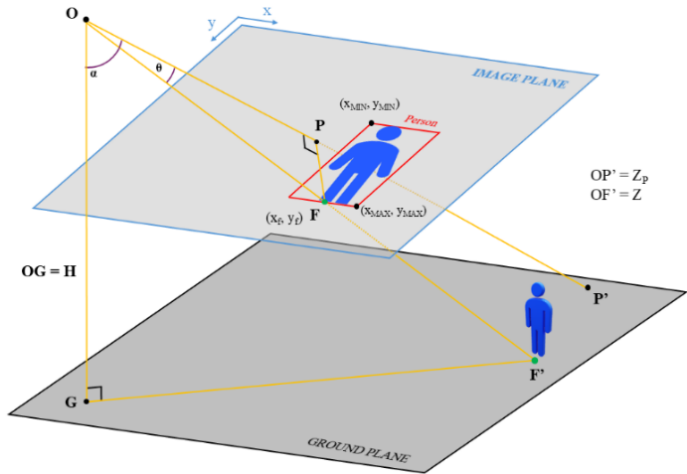


Figure 3: Perspective of 3D world compared to 2D image plane from a UAV camera (Sanyal et al., 2020).

### 5.2.2 Object Depth Estimation

Since the calculations are made without the requirement for altitude information, the distance will be determined using the coordinates of the foot of the person. It shall be assumed for this discussion that the earth is a flat plane and that the bounding boxes are accurate enough that the estimated coordinates of the foot match the actual coordinates of the foot in the real world. Figure 4 provides a visual representation of the issue from the camera perspective. The references (Li & Yoon, 2023) and (Sanyal et al., 2020) serve as the foundation for the calculations exposed in the Dissertation (Vera-Cruz, 2024).



5.2.3 Object Geolocation

It should be noted that the bottom center of the detection bounding box is represented by point F. In scenarios involving the detection of intruders, there will be no crossover of information with NB-IoT tags; however, the location of the intruder may prove to be crucial information. The pixel frame was calculated using the same algorithms that were previously employed to determine the coordinates of point F. The following reference transformations must be conducted to ascertain the GPS coordinates of point F, based on (Sanyal et al., 2020).

- 1. Pixel coordinates → 2D normalized camera coordinates;
- 2. 2D normalized camera coordinates → 3D camera frame coordinates;
- 3. 3D camera frame coordinates → UAV frame coordinates;
- 4. UAV frame coordinates → East, North, Up coordinates;
- 5. East, North, Up coordinates → GPS coordinates.

5.3 NB-IoT Tags

In the current project, NB-IoT tags transmit the user's GPS location to the server. These tags symbolize the GPS-identifying technology of every employee in the restricted region. Bernardo Albano, a team member of this Capstone Project, managed to integrate this tracking solution with the UAV prototype.

5.4 User Interface Web App

Integrated into the surveillance of a restricted zone within the capstone project, this interface is designed to streamline access to the real-time video feed of the UAV. The individual responsible for overseeing the operation accesses the system and, via this interface mentioned above, can view the live streaming, which includes people detection capabilities. This allows for the monitoring of workers in the designated area and the detection of potential intruders. In parallel, the GPS positions of the tags can be managed, including those that are not visible by the UAV. The interface was developed using the Flask framework, with HTML for the page design, to define the locations and dimensions of textual elements, images, action buttons, and other factors related to the page design.

## 6 Project Implementation and Results

By dividing the project into three distinct phases— object detection, object geolocation, and web app user interface— a comprehensive testing of each constituent component was possible.

### 6.1 Project Implementation and Results

The functionality of NVIDIA DeepStream was evaluated through a comprehensive analysis of pre-recorded video footage and live-streaming data. To assess the performance of DeepStream, two distinct tests were implemented. The initial test employed DeepStream samples with pre-recorded videos, thereby providing a controlled setting for the assessment of the algorithm's accuracy and consistency. The second test was designed to assess the camera's object detection capabilities in broadcast footage, thereby simulating more dynamic and challenging conditions.

#### 6.1.1 Sample Tests

The initial phase of testing was conducted with the inclusion of video data within the DeepStream framework. The objective was to evaluate the performance of the algorithm on a pre-recorded video instead of a live-stream video. The issue of movement differs between the two tests due to the speed of the objects relative to the camera, which also differs due to their displacement relative to the camera. Another parameter to be taken into consideration is camera stabilization. The body carrying the camera in a moving situation can infer instability to the camera with its movement. Figure 5 is an example of object detection with DeepStream on a pre-recorded video. In this case, the camera is mounted on a movement body.

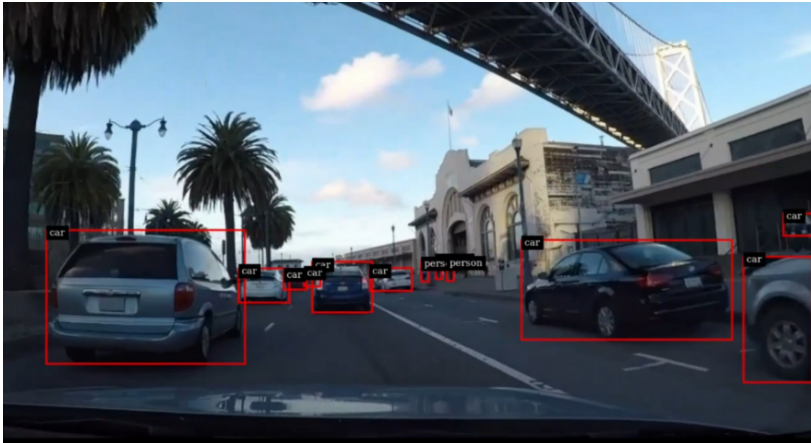


Figure 5: NVIDIA DeepStream SDK– Samples tests

#### 6.1.2 Livestreaming Tests

This section presents the results of tests conducted using live streaming from a USB Trust Exis Webcam (Trust.com, 2024). This approach was adopted as integration with the ZR10 gimbal could not be achieved within the project period. This alternative approach was adopted to simulate real-time video input and evaluate the system's performance under dynamic conditions.

Vehicle Detection

As this category is one that DeepStream can identify and which may be present in any of the situations, a qualitative evaluation of the algorithm’s ability to identify objects in this category was conducted. In this section it presents the algorithm’s performance in vehicle detection in two cases: a comparison of the camera image with (Figure 6 (a)) and without (Figure 6 (b)) window glass.



Figure 6: NVIDIA DeepStream SDK – Detection of Parked Vehicles

People Detection

This section will present examples of people detection. This category is the most applicable primarily for Scenario 1 of this Capstone Project, which deals with the security and surveillance of a limited area. The capacity to detect human presence will facilitate the differentiation between employees and intruders.

Due to the remote location of the testing facility, individuals only visit it at the start and end of working hours. The intensity of sunlight during these periods is too high to conduct tests without the use of glass. To do this test, it was necessary to place a glass between the camera and the items to be identified. Figure 7 below presents an example of people detection test.



Figure 7: NVIDIA DeepStream SDK — Livestream Tests — Detection of Persons.



### Vehicle and People Detection

In the preceding sections, the detection of vehicles and people was examined in isolation. In the context of the UAV prototype operation, scenarios may emerge wherein both categories, from an image perspective, exhibit overlapping characteristics. Accordingly, the objective of this section is to examine the algorithm's capacity to discern between people and vehicles within the same pixel space. In other words, the aim is to assess DeepStream's capability to differentiate between overlapping objects. Figure 8 illustrates an example of the results of the tests conducted in this context.



Figure 8: NVIDIA DeepStream SDK– Livestream Tests– Detection of Vehicles and Persons.

DeepStream demonstrated the capacity to differentiate between people and vehicles, even near the camera, indicating the presence of robust classification capabilities contingent upon the quality of the camera. In scenarios comprising multiple objects, such as instances where individuals are superimposed upon vehicles, the DeepStream algorithm demonstrated superior performance in prerecorded video data compared to live streaming. This highlights the significant challenges that object detection algorithms face in such complex scenarios.

The use of a consumer-grade webcam introduced constraints in image quality and processing, which occasionally resulted in inaccuracies and anomalies in object detection. The resolution and focusing capabilities of the webcam at longer distances affected the performance of the algorithm, particularly in the detection of distant objects.

## 6.2 Object Geolocation

The objective of this component is to calculate the GPS coordinates of objects that have been identified within the surveillance system. The location of an object can be determined by calculating the coordinates of its detection bounding box. The process of obtaining bounding box coordinates from DeepStream is of vital importance for the subsequent calculation of GPS coordinates for identified objects.

The tests were conducted at sunset, which eliminated glass in front of the camera, thus facilitating them. A simulation of the UAV's GPS location was conducted with the coordinates  $38.8355116^\circ$  latitude,  $-9.3353939^\circ$  longitude and  $4m$  of height. A resulting frame is presented in Figure 9.



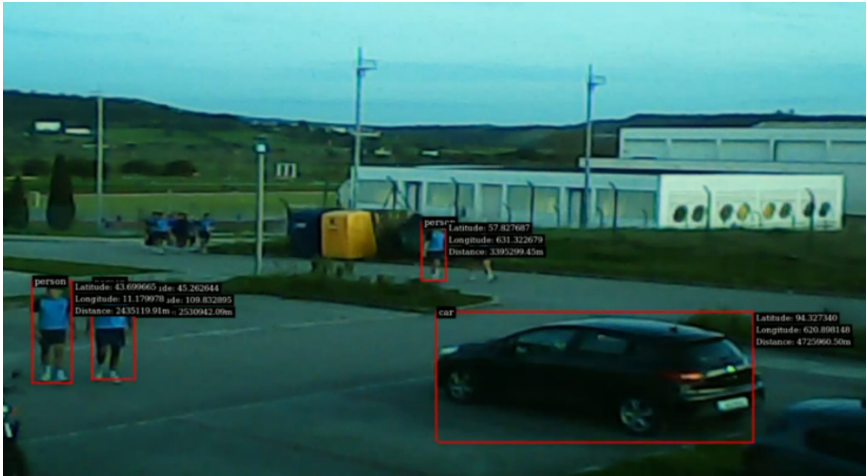


Figure 9: Geolocation of Objects.

To evaluate the accuracy of the object localization approach, a series of tests were conducted using real-world coordinates and estimated positions obtained through the system, to assess the accuracy of the proposed method in determining object coordinates and distance. The results highlighted significant relative errors for each of the measurements when compared to ground references, providing a quantitative basis for analyzing the system's performance and, for that so, demonstrating the necessity for further calibration and refinement of the estimation algorithm.

6.3 User Interface Web App

The website represents the project's most significant strength. The program is currently operational; however, a single application has been developed on the host, which serves to demonstrate that the integration and development of a Flask application to establish an interface is a viable proposition. The general layout of the web app is illustrated in Figure 10.

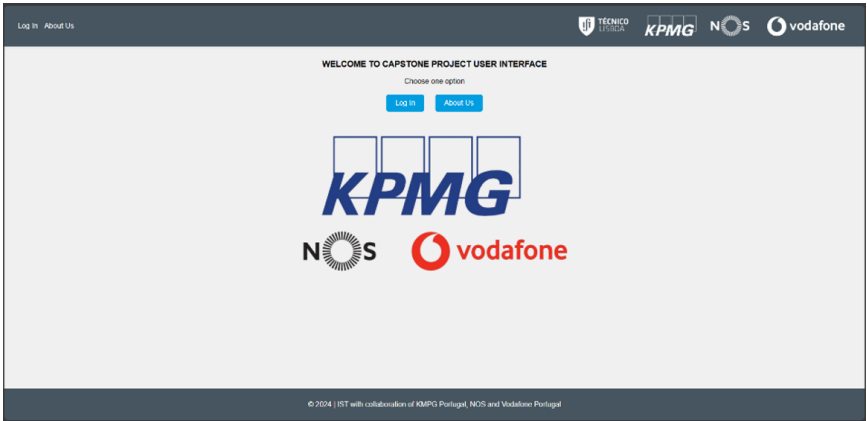


Figure 10: General Web App Layout.

Because of the necessity for a user ID and password to gain access to resources, the app can limit access. This type of access approach serves multiple purposes like ensuring that users have access only to the functions necessary for their role and enhances system security by compartmentalizing administrative functions and sensitive operations.

The integration of a streaming database feature significantly enhances the post-mission analyses, evaluates operational procedures and outcomes, identifies trends or patterns in UAV operations over time, and uses historical data for training purposes or operational improvements.

It is worth noting that the web application was unable to incorporate the object detection results from DeepStream with the real-time position of NB-IoT tags. Despite the inability to complete the DeepStream integration on the day this document was delivered, the interface can access the camera of the platform that initiates the app.

## 7 Conclusions

UAVs are integrated into hierarchical computational models, where they process computational tasks from IoT devices, optimize resource allocation, and ensure task offloading to edge and cloud servers, enhancing system performance and scalability. UAVs play a crucial role in enhancing the capabilities and effectiveness of IoT technologies by integrating with sensors and communication networks. The deployment of UAVs is important due to their capacity to access challenging or hazardous locations, ensuring comprehensive monitoring without compromising human safety. Moreover, the use of advanced imaging technologies and sensors on UAVs ensures the acquisition of accurate and high-quality data, which is superior to the capabilities of human inspectors and facilitates more effective decision-making processes.

This project sets out to create a prototype for a UAV-based inspection and surveillance system, with a strong emphasis on using open-source tools to streamline implementation in UAS. The objective was to limit the development of entirely innovative technologies or methods where possible, opting instead for readily available solutions to reduce costs and complexity while maintaining a flexible approach that allowed for some degree of innovation.

A key deliverable for the project was a user interface web application that would facilitate streaming, people detection, and geolocation of potential intruders. The intention was to develop a platform that could effectively support the UAV's real-time monitoring and detection capabilities while remaining accessible and easy to use.

To meet these objectives, an extensive research and selection phase was conducted to choose appropriate components for the UAV. Performance, cost, and compatibility were the primary factors in selecting each hardware and software, as these were crucial to the project's objectives. This process led to adopting components such as the NVIDIA Jetson Orin Nano Developer Kit as the UAV's onboard computer and the SIYI ZR10 gimbal camera for imaging.

However, challenges arose during the integration phase, particularly when attempting to connect the SIYI ZR10 camera with the Jetson Orin Nano Developer Kit. Due to compatibility issues, it was not possible to establish a reliable connection between these two components, which hindered the original plan. As an alternative, a USB webcam was used in place of the ZR10 camera, which allowed the project to proceed but at the cost of certain functionalities and image capabilities features that the original gimbal camera would have provided. Despite this substitution, the integration of DeepStream with the new camera setup allowed for partial testing of the object detection system.

The integration of the geolocation component proved to be a significant challenge. Despite the restriction of the DeepStream framework and the unavailability to modify restricted parameters within restricted files, it was possible to overcome this hurdle, and the integration of the geolocation calculation with DeepStream's metadata structures was achieved. However, the calculations were not executed correctly, resulting in a significant discrepancy in the determination of GPS coordinates for the detected objects.

The project successfully developed a web application for the user interface using Flask. However, it fell short of fully integrating all planned components. Specifically, the mapping of NB-IoT tags could not be completed within the project period. The inability to fully integrate the NB-IoT tags map was a key objective that remained unachieved at the time of writing.

Despite these challenges, the project made considerable progress in developing individual components and establishing a foundation for future work. The experience gained and the partial successes achieved provide valuable insights for further development and refinement of the system. Future iterations of this project can build upon these foundations to fully realize the intended integrated UAS user interface with advanced detection and geolocation capabilities.

The project has demonstrated its practical potential by establishing a foundation for applications in emergency response, infrastructure monitoring, and surveillance.

The real-time video streaming functionality of the UAV camera is made possible by the availability of a fully operational Flask web application. The user-friendly interface and incorporation of essential surveillance capabilities illustrate the potential for an accessible control platform with advanced computer vision and UAV technology.

This study illustrates the potential of pre-trained models for real-time operation on embedded devices, such as the Jetson Orin Nano, by demonstrating the feasibility of deploying the NVIDIA DeepStream SDK for human detection. This illustrates that DeepStream may be employed for UAV surveillance tasks without necessitating extensive model training. The utilization of alternative components, exemplified by a USB webcam, illustrates the adaptability of prototyping techniques when the integration of the SIYI ZR10 camera is not feasible. This approach demonstrates the importance of validating crucial system functionalities with alternative hardware arrangements.

In this theoretical model, the project integrates several components, including object identification, geolocation computation, UAV camera streaming, and a web interface, into a unified system. The fundamental structure has been established, although the objective of comprehensive integration with NB-IoT and its mapping has not yet been attained. Furthermore, the geolocation component has not yet been fully developed, as it currently exhibits inaccuracies in calculating the GPS coordinates of objects.

This provides a clear trajectory for further development and revisions, offering a foundation for enhancing the system or even the UAV's real-time tracking and mapping capabilities.

At the Capstone Project level, and on the date of delivery of this document, the UAV did not undergo any testing. Consequently, it was not feasible to conduct a comprehensive analysis of the system or to perform any tests, such as evaluating the battery capacity of the UAV with and without running computer vision algorithms. The financial resources available are insufficient considering the proof of concept developed, which did not permit the procurement of replacement components. Consequently, in the event of an error or accident during flight, there are no replacement parts to continue the project.

As future work, the highlight is the completion of what was designed in this work: the geolocation algorithm and its integration with DeepStream. In this study, it was attempted to calculate the GPS position through reference transformations. However, it is proposed that the use of Universal Transverse Mercator (UTM) coordinates be considered to facilitate the calculation of the geolocation of objects.

With this functional system, it then remains to develop a comprehensive geolocation system that integrates NB-IoT tags with accurate mapping capabilities. This could involve refining algorithms to calculate GPS coordinates more precisely based on detected objects, enabling better tracking and monitoring in real-time and sorting, among the located people, who are authorized people or intruders.

It is also an unrealized aspect that is the realization of these systems with a gimbal, whether the ZR10 or another. It can be an option for the investigation of alternative camera options or configurations that maintain the stability and functionality of the SIYI ZR10 camera, focusing on enhancing image quality and object detection capabilities. Another aspect that was not fulfilled was the infrastructure inspection, so in this development, it is necessary to allow the UAV to perform both scenarios that were studied.

Still, there are more possibilities like extending the object detection capabilities to include multi object tracking. This would allow for simultaneous tracking of multiple individuals or objects within the UAV's FOV, enhancing situational awareness during surveillance operations. This or incorporating other more advanced features, such as interactive maps, alerts for detected intruders, or enhanced data visualization tools, can improve the web application's user interface. This would make the application more intuitive and effective for operators. It may also be interesting to integrate Mission Planner GUI or study another way to access UAV performance and real-time data analytics.

## References

Abdalla, A. S., Powell, K., Marojevic, V., & Geraci, G. (2020, August). UAV-Assisted Attack Prevention, Detection, and Recovery of 5G Networks. *IEEE Wireless Communications*, 27(4), 40-47. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9170267>. doi: 10.1109/MWC.01.1900545

- Abdulghafoor, N., & Abdullah, H. (2021, February). REAL-TIME MOVING OBJECTS DETECTION AND TRACKING USING DEEP-STREAM TECHNOLOGY. *Journal of Engineering Science and Technology*, 16, 194-208. <https://jestec.taylors.edu.my/Vol%2016%20issue%201%20February%202021/16.1.14.pdf>.
- Beekman, J. (2017, May). What Is Internet of Drones (IoD)? *IoT Marketing*. <https://iotmktg.com/what-is-internet-of-drones-iod/>.
- Derhab, A., Cheikhrouhou, O., Allouch, A., Koubaa, A., Qureshi, B., Ferrag, M. A., ... Khan, F. A. (2023, February). Internet of drones security: Taxonomies, open issues, and future directions. *Vehicular Communications*, 39, 100552. <https://www.sciencedirect.com/science/article/pii/S2214209622000997?via%3Dihub>. doi: 10.1016/j.vehcom.2022.100552
- Fraga-Lamas, P., Lopes, S. I., & Fernández-Caramés, T. M. (2021, August). Green IoT and Edge AI as Key Technological Enablers for a Sustainable Digital Transition towards a Smart Circular Economy: An Industry 5.0 Use Case. *Sensors*, 21(17), 5745. <https://www.mdpi.com/1424-8220/21/17/5745>. doi: 10.3390/s21175745
- Hartley, R., & Zisserman, A. (Eds.). (2004). *Multiple View Geometry in Computer Vision* (2nd ed.). New York: Cambridge University Press. [https://www.r-5.org/files/books/computers/algo-list/image-processing/vision/Richard\\_Hartley\\_Andrew\\_Zisserman-Multiple\\_View\\_Geometry\\_in\\_Computer\\_Vision-EN.pdf](https://www.r-5.org/files/books/computers/algo-list/image-processing/vision/Richard_Hartley_Andrew_Zisserman-Multiple_View_Geometry_in_Computer_Vision-EN.pdf).
- Li, S., & Yoon, H. (2023, March). Vehicle Localization in 3D World Coordinates Using Single Camera at Traffic Intersection. *Sensors*, 23(7), 3661. <https://www.mdpi.com/1424-8220/23/7/3661>. doi: 10.3390/s23073661
- NVIDIA. (2023a, January). *Deepstream Overview*. <https://docs.nvidia.com/launchpad/ai/tao-ds/latest/tao-ds-deepstream-overview.html>.
- NVIDIA. (2023b, February). *NVIDIA Jetson Orin Nano Developer Kit - Accelerating entry-level edge AI applications*. <https://files.seeedstudio.com/wiki/Jetson-Orin-Nano-DevKit/jetson-orin-nano-developer-kit-datasheet.pdf>.
- Sanyal, S., Bhushan, S., & Sivayazi, K. (2020, January). Detection and Location Estimation of Object in Unmanned Aerial Vehicle using Single Camera and GPS. In *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*. Raipur, India. <https://ieeexplore.ieee.org/document/9071439>. doi: 10.1109/ICPC2T48082.2020.9071439
- Silver, B., Mazur, M., Wiśniewski, A., & Babicz, A. (2017, July). Welcome to the era of drone-powered solutions: a valuable source of new revenue streams for telecoms operators. *PWC Communications Review*. <https://www.pwc.pl/pl/pdf/communications-review-dps-drones-pwc-2017.pdf>.
- Trust.com. (2024). *Exis Webcam – black/silver*. <https://www.trust.com/pt/product/17003-exis-webcam>.
- Vera-Cruz, T. (2024). *User Interface for UAV Object Detection and Geolocation* (MSc Thesis in Aerospace Engineering). Instituto Superior Técnico, Lisboa, Portugal.



# **Criação de um simulador de voo híbrido, de realidade virtual e mista, e proposta de adaptação do Syllabus de instrução de voo**



**Autor: ALF/PILAV Francisco José Lopes Félix**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

**Orientador: TCOR/TMAEQ Paula Alexandra Veiga Gonçalves**  
Academia da Força Aérea, Sintra

**Coorientador: CAP/ENGEL Diogo Alexandre Oliveira Silva**  
Academia da Força Aérea, Sintra

## **Resumo**

Até 2043, será necessário formar 674.000 pilotos, em termos globais, e apenas na aviação civil. Deste modo, é necessário tornar o treino dos pilotos mais produtivo, efetivo, diminuindo os custos, sem descuidar a segurança. Assim, tecnologias emergentes como a realidade virtual e mista, aplicadas à instrução de voo, serão capazes de alterar o paradigma atual.

O desenvolvimento de sistemas de simulação de voo que empreguem tecnologias de *extended reality*, é crucial, por forma a integrar os novos desenvolvimentos tecnológicos na instrução de voo, servindo de *proof of concept* para outros sistemas de armas da Força Aérea (FA). Esta dissertação propõe-se a delinear a metodologia subjacente ao desenvolvimento de um simulador de voo que empregue tanto realidade virtual (VR) como realidade mista (MR) e a sua respetiva implementação. Para tal, será necessária a criação de duas plataformas diferentes: uma recuperando a base já existente na Esquadra 802 que empregará tecnologia MR, permitindo a visualização real do interior do cockpit e a visualização virtual do exterior; e outra criada de origem para simulação VR de qualquer aeronave de asa fixa ou de asa rotativa.

A criação dos simuladores preconizados nesta dissertação foi feita através da integração de produtos *commercial off the shelf* (COTS), tanto em *hardware* como em *software*, e de manufatura aditiva, sendo que estes apresentam globalmente um custo substancialmente inferior a simuladores especializados de fabricantes aeronáuticos. Ademais, este estudo também pretende expor a metodologia aplicável a um qualquer sistema de armas na execução de um simulador de voo VR/MR.

Finalmente, foi delineada uma proposta de adaptação do *syllabus* de instrução de voo da Esquadra 802, passando pela integração dos simuladores na mesma.

**Palavras-chave:** Aviação militar; Instrução de voo; Realidade mista; Realidade virtual; Simulador de voo.

## **1 Introdução**

Até 2043 serão necessários formar 674.000 pilotos em termos globais, apenas na aviação civil (Boeing, 2024). Este facto irá causar desafios tremendos aos operadores e às companhias aéreas, a não ser que algumas medidas sejam tomadas, com o objetivo de aumentar o número de pilotos qualificados a operar. Com esta necessidade, as companhias aéreas têm procurado no meio militar um alívio, forçando a saída de pilotos militares, atraídos por melhores condições e remunerações (Sérgio A. Vitorino, 2025). Assim, as organizações militares vêem diminuir o número de pilotos nas suas fileiras, motivada quer pela saída destes para o mundo civil, quer pelo aumento das missões confiadas aos militares, podendo, caso algumas medidas não sejam tomadas, afetar a segurança e a soberania nacional. Deste modo, de forma a que a aviação possa continuar a florescer, é necessário tornar o treino dos pilotos mais eficiente, diminuindo os custos, sem descuidar a segurança, garantindo o mesmo nível de proficiência. Daqui surgiu a investigação de tecnologias emergentes aplicadas à instrução de voo, como a realidade virtual e mista, as quais serão capazes de alterar o paradigma atual em menos de uma década, segundo estudos recentes (Varjo, 2021).

A 31 de maio de 2019, a Força Aérea dos Estados Unidos da América (USAF) implementou experimentalmente o *Pilot Training-Next* (PTN) (Hawkins, 2019), com vista a substituir o *Undergraduate Pilot Training* (UPT). Esta atualização do programa de instrução de voo depende da utilização de tecnologia de Realidade Virtual (VR), por forma a reduzir os custos, aumentar a flexibilidade e diminuir o tempo de instrução dos pilotos (Pope, 2019). Um estudo de 2019 apresenta que os alunos do PTN qualificam-se 6 meses mais cedo, quando comparados com os alunos do UPT, com a mesma utilização da aeronave de treino T-6 em termos de horas de voo. Em 10 anos, conseguiria treinar 14.444 alunos, com uma poupança de 8,96 milhões de dólares em custos fixos e de 3,96 milhões de dólares em custos variáveis (Pope, 2019).

Não existe, à data, a utilização de tecnologias de *Extended Reality* (xR), isto é, qualquer tipo de tecnologia que altere a realidade por meio de adição de elementos digitais ao mundo físico (Tremosa, 2023), na instrução de voo dos pilotos da FA. O desenvolvimento de sistemas de simulação de voo que empreguem este tipo de tecnologias é crucial, com o fim de integrar os novos desenvolvimentos tecnológicos na instrução de voo, servindo de *proof of concept* para outros sistemas de armas da FA. As vantagens e desvantagens da tecnologia VR serão abordadas num próximo tópico, mas uma mais-valia da tecnologia VR e de realidade mista (MR) é a flexibilidade, a qual permite que seja adaptada a um qualquer sistema de armas com alterações mínimas. Assim, é garantido que a tecnologia não se tornará, num futuro próximo, obsoleta, tendo um grande potencial de desenvolvimento e adaptação.

## 2 Revisão de literatura

### 2.1 *Extended reality*

*Extended Reality* (xR) é um termo que se refere a tecnologias que permitem combinar o mundo físico com o mundo virtual, providenciando imersão e realismo a experiências virtuais. Assim, congrega principalmente 3 tipos de tecnologias: realidade aumentada (AR), realidade virtual (VR) e realidade mista (MR). Na prática, a tecnologia VR espelha o mundo físico de forma totalmente virtual, a tecnologia AR sobrepõe elementos virtuais no mundo real e a tecnologia MR combina elementos virtuais com elementos reais (Wang, 2023).

### 2.2 Validade e fidelidade

A validade, aplicada aos simuladores de voo, representa o grau de precisão com que a simulação representa a realidade. Isto significa que uma alta validade implica uma representação mais fidedigna da tarefa a realizar no simulador. Assim, um sistema de simulação apenas poderá ser válido caso este sistema tenha a capacidade de recriar a realidade e não apenas os pormenores visuais desta (Wang, 2023).

A fidelidade representa a capacidade do simulador replicar o ambiente real, podendo ser subdividida em diversos subtipos. Os mais importantes para a simulação são a fidelidade cognitiva e ergonômica, uma vez que são os que têm a maior influência na transferência de treino. De ressaltar que a fidelidade depende inteiramente do objetivo da simulação, por exemplo: um simulador de voo que tenha como objetivo o treino de manobras não tem a necessidade de replicar de forma fiel todo o *cockpit* (Myers, Starr, & Mullins, 2018; Vincenzi, Liu, & Macchiarella, 2008; Wang, 2023).

### 2.3 Transferência de treino

A *Federal Aviation Administration* (FAA) define transferência de treino como a habilidade para aplicar conhecimento ou procedimentos aprendidos num contexto, em novos contextos (FAA, 2020). Desta forma, a transferência de treino refere-se à forma como o conhecimento aprendido anteriormente influencia uma situação futura. Assim, a transferência de treino pode ser positiva, neutra ou negativa, dependendo se o conhecimento aprendido anteriormente afeta de forma positiva, neutra ou negativa a performance numa situação real (Borgvall, Castor, Svensson, Nählinder, & Oscarsson, 2008).

Um estudo desenvolvido por Thomas Carretta e Ronald Dunlap constituiu dois grupos, sendo que um dos grupos, antes da instrução em voo, iria receber duas sessões em simulador, de forma a praticarem aterragens, e o outro grupo iria servir de grupo de controlo, sem qualquer exposição aos simuladores. Os resultados apontaram que o grupo que tinha sido sujeito a sessões no simulador precisou, em média, de menos 1.5 horas antes de reunir condições para o voo solo (Thomas R. Carretta & Ronald D. Dunlap, 1998).

### 2.4 Uso de realidade virtual e mista

A tecnologia VR é referente à tecnologia que permite gerar um ambiente totalmente virtual, com vista a simular um ambiente real físico, sendo que normalmente assenta na utilização de *Head-mounted Displays* (HMD) por forma a servir de ponte visual. A tecnologia MR é a tecnologia que permite a combinação de elementos visuais virtuais e físicos, onde o ambiente real não é totalmente substituído pelo ambiente virtual, mas sim integrado. Num sistema MR, os objetos reais e virtuais são integrados utilizando um HMD que contenha sensores eletro-óticos e software que permita a mistura dos objetos reais com o ambiente virtual (Wang, 2023).

A tecnologia VR já existe desde a década de 60, mas devido aos seus elevados custos à época e limitações tecnológicas em termos de *hardware* e *software*, apenas viu o seu desenvolvimento inicial por instituições com recursos financeiros suficientes para o alimentar. Um estudo de 1993 (Regian, Shebilske, & Monk, 1993) revelou que alunos que treinavam com VR para tarefas espaciais tinham uma maior retenção de conhecimento, uma mais rápida familiarização com as tarefas e uma performance superior, quando comparados com uma amostra obtida estatisticamente. Em 2002, um outro estudo (Seymour et al., 2002) apontou que alunos treinados com o uso de tecnologia VR executavam as suas tarefas 29% mais rápido, faziam 6 vezes menos erros e 9 vezes menos falhas de progressão, do que um aluno sem qualquer treino em VR (Pope, 2019). Um estudo de 2001 (Estrada & Adam, 2001) verificou que a instrução em VR contribuiu para um aumento da



produtividade e da experiência de voo em áreas onde são requeridas capacidades de decisão, de *situational awareness* e de *crew coordination*. Um outro estudo de 2007 (Bowman & McMahan, 2007) concluiu que a retenção de habilidades adquiridas em instrução VR dependia principalmente da fidelidade do simulador, do realismo e da integração dos diversos componentes tecnológicos. Este tipo de tecnologia poderá complementar ou até substituir os métodos de treino atuais, trazendo benefícios claros no que toca à segurança e custos, bem como maximizando a eficiência e a efetividade do treino (Wang, 2023).

#### **2.4.1 Capacidade de replicar as sensações do voo**

Dentro da capacidade de replicar estas sensações, importa referir os conceitos de imersão e de presença. A imersão refere-se à utilização de equipamento imersivo, como o HMD, tendo como objetivo a correlação dos inputs sensoriais reais com inputs sensoriais gerados digitalmente, como imagens ou sons (Freina & Ott, 2015). A presença refere-se à capacidade do utilizador reagir a estar imerso num ambiente virtual, onde o seu sistema cerebral e nervoso reage aos estímulos simulados de forma semelhante à reação que teria a um estímulo real. Assim, os simuladores de voo podem ser distinguidos entre simuladores imersivos, os quais geram um ambiente simulado, e simuladores não imersivos, normalmente os gerados utilizando um monitor ou um projetor (Jensen & Konradsen, 2018).

Para a replicação das sensações de voo, não é necessário replicar o movimento, através de plataformas de movimento, uma vez que a presença desta capacidade não apresenta uma visível melhoria na transferência da simulação de voo para o voo real. Isto acontece porque o *feedback* fornecido pela plataforma de movimento é muito inferior à sensação visual, competindo a esta o papel principal na imersão e na presença na simulação de voo. Isto aplica-se sempre que o ambiente da simulação permita que o utilizador do simulador de voo acredite que a sensação fornecida, de forma simulada, replica as sensações reais do voo (Guthridge & Clinton-Lisell, 2023).

#### **2.4.2 Diferença entre VR e MR na instrução de voo**

Importa, assim, definir qual das plataformas exerce maior efeito no utilizador: a plataforma VR ou a plataforma MR. Um estudo de 2023 (Wang, 2023) investigou qual implementação, VR ou MR, proporciona maior fidelidade cognitiva e ergonómica na instrução de voo. Os resultados do estudo apontaram que a tecnologia MR apresenta melhor adaptabilidade à instrução de voo quando comparada com a tecnologia VR, evidenciando índices mais elevados de fidelidade cognitiva e ergonómica. Entre os fatores que justificam esta superioridade, está a capacidade da tecnologia MR oferecer um ambiente mais intuitivo, preciso e natural aos utilizadores.

Além disso, a tecnologia MR demonstrou maior aceitação por parte dos utilizadores, ao encorajá-los a interagir de forma mais eficiente com o simulador. Este envolvimento facilitou a concentração dos alunos nas tarefas e contribuiu para uma melhor transferência de treino. O estudo também indicou que os participantes concluíram as tarefas com menos erros e de forma mais rápida em ambientes MR do que em VR, mesmo sob condições semelhantes. Tal facto deve-se à capacidade da tecnologia MR promover o uso natural de ambas as mãos pelos participantes, em contraste com a tecnologia VR, que leva a movimentos rígidos, devido à ausência de uma representação real das mãos.

### **2.5 Requisitos de hardware**

Um simulador de voo deve ter a capacidade de refletir em tempo real os *inputs* fornecidos nos controlos de voo nos instrumentos de bordo, sem que o utilizador perceciona qualquer atraso entre a ação e a reação (Guthridge & Clinton-Lisell, 2023). De acordo com um estudo de 1995 (Bradley & Abelson, 1995), um simulador deve garantir uma taxa de *frames* entre 15 e 30 *frames* por segundo (FPS) para replicar adequadamente o movimento de uma aeronave. Esta especificação apresenta um dos principais desafios dos sistemas de simulação: assegurar que o computador tem capacidade suficiente para correr o *software* de simulação acima desse limite de FPS. Quando o *software* apresenta cenários altamente detalhados, a taxa de *frames* pode diminuir drasticamente, chegando a 4 ou 6 FPS. Isso implica que cerca de 75% da informação visual da simulação está a ser omitida (Guthridge & Clinton-Lisell, 2023). Tal situação é particularmente problemática em fases críticas do voo, como voos a baixa altitude, aterragem ou descolagem, onde as alterações no ambiente são rápidas e os utilizadores necessitam de realizar ajustes precisos.

A *European Union Aviation Safety Agency* (EASA), agência europeia responsável pela segurança e certificação de simuladores de voo, introduziu em 2023 uma norma que requer que os sistemas de realidade virtual utilizados em simulação de voo, incluindo os *head-mounted displays*, mantenham uma taxa mínima de 60 FPS, por forma a assegurar a fluidez e precisão das simulações (EASA, 2023; Guthridge & Clinton-Lisell, 2023).

## 2.6 Pilot training next – USAF

A 31 de maio de 2019, a Força Aérea dos Estados Unidos da América (USAF) implementou, de forma experimental, o *Pilot Training-Next* (PTN) (Hawkins, 2019), com o objetivo de substituir o *Undergraduate Pilot Training* (UPT). Esta atualização no programa de instrução de voo baseia-se na utilização intensiva da tecnologia de Realidade Virtual (VR), visando reduzir custos, aumentar a flexibilidade e diminuir o tempo necessário para a formação de pilotos (Pope, 2019).

A vertente VR do PTN congrega um *head-mounted display* VR com equipamentos físicos de alta-fidelidade, providenciando sensação tátil, auditiva e háptica que, combinado com os instrutores de voo, tem a capacidade de providenciar cenários de treino extremamente fiéis à dinâmica de voo real. Assim, para além de reduzir o *workload* dos instrutores de voo, a instrução de voo deixa de estar totalmente dependente da prontidão das aeronaves, levando a que os alunos mantenham a continuidade de voo (Holmes, 2019). A utilização intensiva da realidade virtual no PTN permite que os pilotos adquiram habilidades essenciais e pratiquem manobras num ambiente simulado antes de as realizar em voos reais. Em termos de custos, o PTN apresenta uma redução significativa em comparação com o seu antecessor, o UPT.

Um estudo de 2019 (Pope, 2019) revelou que os alunos do PTN qualificam-se 6 meses mais cedo em relação aos alunos do UPT, com a mesma utilização da aeronave de treino T-6 em termos de horas de voo. Em 10 anos, o PTN conseguiria formar 14.444 alunos, com uma poupança de 8,96 mil milhões de dólares em custos fixos e 3,96 mil milhões de dólares em custos variáveis.

## 3 Metodologia

A seguinte figura ilustra os softwares necessários para um sistema de simulação MR, sendo que a única alteração para um sistema de simulação VR é a não utilização de um *software* misturador de realidade.

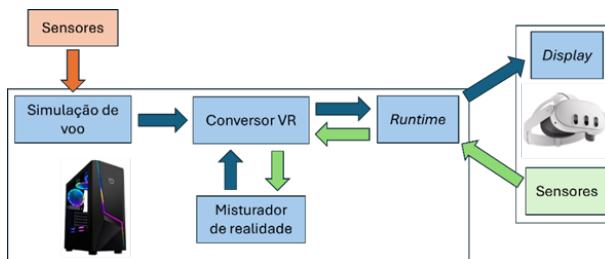


Figura 1: Esquema de *softwares* necessários (Elaboração própria)

### 3.1 Software

O *software* de simulação será responsável por simular o ambiente exterior e por replicar as condições de voo, através dos dados dos sensores dos controlos de voo. Assim, um *software* de simulação de voo deve ser analisado tendo em conta os seguintes fatores: performance, capacidade gráfica, dinâmica de voo e simulação de condições meteorológicas.

O *software* conversor para VR é aquele que permite a ligação entre o computador e o HMD, permitindo que este interaja diretamente com o computador, como se tratasse de um *display*. Assim, este permite espelhar o computador em VR, exibindo o *display* do computador como se de uma janela tridimensional se tratasse. Este *software* também é o responsável pela receção dos dados dos sensores do HMD, sendo necessário este, para a simulação MR, ter a capacidade

de *passthrough*, isto é, permitir que o *hardware* do HMD (câmaras e sensores espaciais) seja utilizado diretamente pelo *software* misturador de realidade.

O *software* misturador de realidade é aquele que permite a interligação das imagens reais com as imagens virtuais, permitindo o HMD ser de MR. Desta forma, utiliza as imagens reais captadas pelas câmaras do HMD e integra-as com as imagens geradas do ambiente virtual.

O *runtime* é um protocolo que permite transmitir a informação do computador para o HMD (Murray, 2017). Assim, este serve como tradutor entre o computador e o HMD, permitindo a integração de ambos. Este cria uma base entre os diversos tipos de HMD, permitindo unificar as diferentes interfaces.

## **3.2 Hardware**

Os sensores têm como objetivo a medição das posições dos controlos de voo, bem como a replicação dos periféricos das aeronaves. Assim, serão necessários sensores para as superfícies principais e secundárias da aeronave (por exemplo, *ailerons* e *flaps*), para os sistemas de controlo do motor (por exemplo, manetes de potência, da mistura e do passo), para os travões, entre outros sistemas e periféricos da aeronave que se ache pertinente aplicar.

Um *human interface device* (HID) é o dispositivo responsável por extrair os *inputs* dos sensores e modificá-los em valores que irão equivaler a um movimento no *software* de simulação (Li, You, Zhou, Chen, & Tang, 2018). Assim, este permite a programação dos sensores, por forma a transformar os valores dos sensores em dados digitais que o *software* de simulação reconheça. Este processa dados quer analógicos (sensores dos controlos de voo), quer digitais (interruptores).

É necessário um computador com os requisitos adequados para assegurar uma simulação fluida, permitindo aos softwares operarem sem latência. Para tal, um computador deve ser avaliado principalmente tendo em conta o seu processador (CPU), placa gráfica (GPU) e memória de acesso aleatório (RAM), entre outros parâmetros.

O HMD deve ter a capacidade de reproduzir imagens, quer VR, quer MR, para além de conseguir utilizar os diversos softwares necessários para a simulação de voo e conectar-se ao computador. Os critérios a adotar para a escolha do HMD deverão incluir o preço, a facilidade de integração e a abrangência de utilização, o que permitirá uma resolução mais fácil de problemas.

# **4 Desenvolvimento dos sistemas de simulação**

## **4.1 Software**

### **4.1.1 Software de simulação de voo**

Foram identificados dois *softwares* de simulação de voo, COTS e de fácil aquisição, que proporcionam simulação de voo VR: o *Microsoft Flight Simulator* (MSFS), atualmente na versão MSFS2020; e o *X-Plane*, atualmente na versão *X-Plane 12*. No momento da escrita, não foram encontrados estudos científicos que comprovem a superioridade de um, face ao outro, para os objetivos propostos.

Com base numa análise global aos seguintes fatores: performance, capacidade gráfica, dinâmica de voo e simulação de condições meteorológicas, o MSFS2020 destaca-se como superior em relação ao *X-Plane 12*, sendo recomendado para o desenvolvimento dos sistemas de simulação.

### **4.1.2 Software conversor para VR**

Este tipo de *software* age como um *desktop virtual*, isto é, um posto de trabalho virtual que utiliza os dados de um computador físico separado deste, separando o *desktop* do *hardware* (IBM, 2023). Para tal, foram considerados dois softwares: o *Virtual Desktop* e o *Meta Link*.

A aplicação *Virtual Desktop* é uma aplicação que permite visionar o computador em VR. Esta aplicação é altamente otimizada, tendo sido desenvolvida para transmitir as imagens com baixa latência e com uma alta qualidade (Meta, n.d.). Esta aplicação apresenta-se como alternativa à aplicação nativa do HMD Meta Quest 3, o *software Meta Link*, uma vez que o *software Virtual Desktop* permite *passthrough*. Assim, o *software* escolhido foi o *Virtual Desktop*.

### 4.1.3 Software misturador de realidade

Existem dois *softwares* misturadores de realidade que permitem a integração com o MSFS2020: o *Reality Mixer* da *Reality Inside* e o *SimXR* da *rkApps*.

O *software Reality Mixer* é um *software* que permite utilizar as imagens geradas nas câmaras do HMD em combinação com as imagens virtuais do ambiente VR. Permite a sua utilização em paralelo com outros *softwares* e jogos VR. Este *software* cria "caixas de realidade" que podem ser alteradas, por forma a inserirem-se num qualquer ambiente (Inside, 2022).

O *software SimXR* é, à semelhança do *Reality Mixer*, um *software* que permite utilizar as imagens geradas nas câmaras do HMD em combinação com as imagens virtuais do ambiente VR. No entanto, em vez de funcionar com "caixas de realidade", funciona através de formas (2D) totalmente personalizáveis, permitindo ao utilizador selecionar a forma geométrica que pretende captar das imagens reais. Porém, dado fazer a captura através de formas 2D, não permite a captura de um *cockpit* físico completo (rkApps, n.d.).

Assim, apesar de ambas serem semelhantes no objetivo, o *Reality Mixer* mostra-se como superior ao *SimXR*, dado poderem ser selecionadas janelas 3D, as quais permitem fazer um recorte de um *cockpit* físico completo. Ainda assim, o *SimXR* permite uma integração mais rápida com o MSFS2020, podendo no futuro próximo apresentar-se como um substituto ao *Reality Mixer*.

### 4.1.4 Runtime

Podem ser utilizados dois tipos de *runtime* com o *Virtual Desktop*: VDXR ou *SteamVR*. No entanto, devido ao *Reality Mixer* ser um *software* nativo da *Steam*, apenas pode ser utilizado o *runtime SteamVR*.

O *SteamVR* é um *runtime* incluído na aplicação *Steam* que possibilita experiências VR (Steamworks, n.d.). A figura seguinte apresenta o esquema da integração dos *softwares* utilizados.

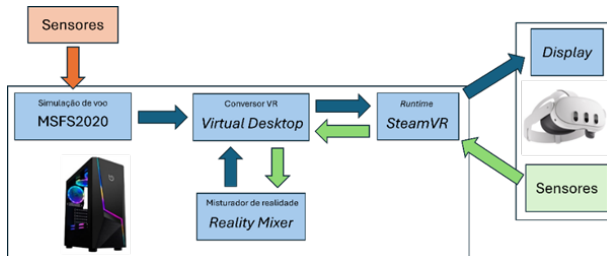


Figura 2: Esquema da integração dos *softwares* utilizados

## 4.2 Hardware

### 4.2.1 Human Interface Device

Um tipo de dispositivo que permite ser um HID é o *Arduíno*, sendo que este poderá existir em vários tamanhos e potencialidades. Assim, é necessário programar os *Arduínos* de forma a transformarem os valores dos sensores em dados digitais que o *software* de simulação reconheça. Para isto, foi utilizada a biblioteca de *Arduíno Joystick Arduino Library*, onde esta permite que, com código simples, sejam convertidos os sinais dos sensores analógicos e digitais num protocolo possível de ser entendido pelo *Windows* e pelo *software* de simulação de voo. No entanto, esta biblioteca apenas pode ser utilizada com *Arduíno* baseado no processador ATmega32u4 (por exemplo, *Arduíno Micro*) (Heironimus, 2024). Após a implementação do código, estes devem ser calibrados para a sua amplitude de movimento. Para tal, estes devem ser calibrados utilizando o *software* nativo do *Windows* "Calibrar controladores de jogo USB", sendo que esta calibração deve ser sempre feita antes de inicializar o *software* de simulação.

#### 4.2.2 Computador

Os requisitos ideais em termos de CPU para o MSFS2020 são uma CPU *Intel i7-9800X* com 4.50 GHz de velocidade de relógio e *cache* de 16.5 MB (Intel, n.d.; Microsoft, 2024). Os requisitos ideais em termos de GPU para o MSFS2020 são uma GPU *NVIDIA RTX2080* com 2944 cores com velocidade de relógio de 1710 MHz e VRAM de 8 GB (Microsoft, 2024; TechPowerUp, 2024). Os requisitos ideais em termos de RAM para o MSFS2020 são 32 GB de memória (Microsoft, 2024).

As características do computador adquirido são:

- CPU: *Intel i7-14700F*;
- GPU: *NVIDIA GeForce RTX 4070 Ti SUPER*;
- RAM: 64GB;

#### 4.2.3 Head-Mounted Display

Existem diversos HMD comerciais, no entanto, apenas alguns permitem a integração para MR. Aquele que apresenta a melhor relação de custo/benefício é o *Meta Quest 3*. Os requisitos recomendados para a utilização deste são no sentido dos apresentados anteriormente: processador da gama *Intel i7*, placa gráfica *NVIDIA RTX* da série 40 e 16 GB de RAM (Meta, 2024).

A seguinte figura retrata o esquema de *hardware* utilizado.

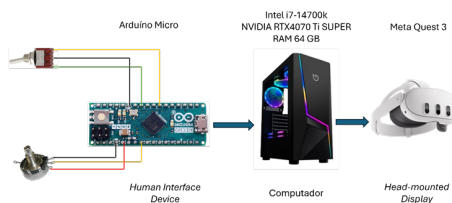


Figura 3: Esquema de *hardware* utilizado (Elaboração própria)

### 4.3 Sistema de simulação *Chipmunk*

A plataforma de simulação foi desenhada para ser um simulador de manobras, isto é, uma plataforma que permite a replicação das manobras em voo, desde a decolagem até à aterragem, passando pelas manobras básicas, acrobacia e navegação, e não um simulador totalmente integrado que permita a aplicação de procedimentos de solo e a resolução de emergências, uma vez que este último iria requerer a quase integral recuperação da plataforma já existente. Assim, a recuperação passou pelos controlos de voo (eixos longitudinal, transversal e vertical, manete de potência, manete da mistura, *flaps*, compensador, travões) e de alguns periféricos (motor de arranque, magnetos).

#### 4.3.1 Controlos de voo

A recuperação dos controlos de voo (manche e pedais) passou por uma ideologia que permitisse uma fácil substituição das peças no caso de falha das mesmas, uma vez que foi um dos grandes obstáculos às tentativas anteriores de recuperação desta plataforma. Assim, a aplicação dos sensores nos controlos de voo foi feita através de manufatura aditiva, permitindo esta que a substituição de componentes seja rápida, eficaz e adaptável.

O manche controla os eixos longitudinal e transversal da aeronave. Desta forma, foi necessário avaliar quais os 2 melhores locais onde iriam ser ligados os sensores. Uma vez que esta plataforma ainda funciona mecanicamente de forma semelhante à aeronave real, utilizaram-se os eixos ligados ao manche. Os 2 tipos de sensores analógicos que poderiam ser empregues seriam potenciômetros ou sensores de efeito de *Hall*. Dada a amplitude de movimento do manche, empregou-se potenciômetros para a extração dos dados, tendo sido aplicado um sistema de engrenagens para aumentar a resolução de leitura do mesmo. O potenciômetro rotacional mede valores numa amplitude de 300 graus. Foi aferido que a deslocação do manche é de 40 graus e 90 graus para o eixo transversal e longitudinal, respetivamente. Uma

vez que a relação entre o diâmetro das engrenagens e os ângulos de rotação é linear, implica que: para o eixo transversal, um diâmetro de 13 milímetros da engrenagem do potenciômetro pode ser utilizada uma engrenagem de 79 milímetros, dando um valor inferior a 300 graus, por forma a garantir uma margem de segurança; para o eixo longitudinal, um diâmetro de 13 milímetros da engrenagem do potenciômetro pode ser utilizada uma engrenagem de 39 milímetros, dando de igual forma um valor inferior a 300 graus, por forma a garantir uma margem de segurança. A figura seguinte mostra a modelação 3D das peças e a respetiva aplicação real.



Figura 4: Modelação 3D e aplicação real do manche

Os pedais controlam o eixo vertical da aeronave. A mesma metodologia empregue no manche foi aplicada nos pedais, sendo que, neste caso, não foi utilizado um sistema de engrenagens, uma vez que, dada a sensibilidade real destes, não se torna imperativo uma boa resolução do potenciômetro. Assim, a figura seguinte mostra a modelação 3D das peças e a respetiva aplicação real.

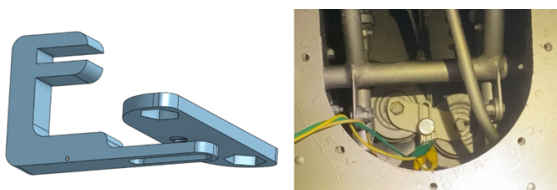


Figura 5: Modelação 3D e aplicação real dos pedais

Uma vez que este simulador apenas tem como objetivo simular manobras, foi desconsiderada a recuperação dos periféricos. No entanto, por forma a servir de exemplo para desenvolvimento futuro, foram recuperados os interruptores dos magnetos e do motor de arranque.

### 4.3.2 Calibração dos sensores

Para além da calibração feita através do *software* nativo do *Windows* "Calibrar controladores de jogo USB", foi também necessário proceder ao ajuste da curva de sensibilidade dos sensores, por forma a replicar da forma mais precisa o comportamento da aeronave real. De igual forma, foi inserido no código do *Arduino* um filtro, por forma a diminuir o *jitter*, isto é, o movimento errático de baixa amplitude, dos potenciômetros. Isto tornou-se evidente no caso do potenciômetro inserido nos pedais, uma vez que este foi calibrado para uma amplitude de rotação inferior. O tipo de filtro inserido foi do tipo *infinite impulse-response filter* (IIR) de primeira ordem, o qual pondera o valor antigo com o valor atualizado do potenciômetro, por forma a diminuir as pequenas flutuações de valores (*jitter*) (Nounou & Bakshi, 2000).

### 4.3.3 Display digital

Uma solução encontrada para possibilitar a leitura dos instrumentos da aeronave passa pela inclusão de um monitor na zona dos instrumentos e a replicação destes. Assim, é necessário um *software* que colete as informações do voo do *software* de simulação e que as transforme em instrumentos digitais, que repliquem os da aeronave real.

Um *software* com estas características é o *Air Manager*, que, para além de outras capacidades, permite selecionar o formato pretendido para o instrumento de voo e permite também inseri-lo numa posição fixa. Para além de permitir

esta personalização, também contém painéis de instrumentos que replicam os de aeronaves reais, garantindo uma grande flexibilidade. Também permite a calibração destes instrumentos, podendo ser personalizada a resposta do instrumento, a amplitude de valores deste e o tempo de resposta (Sim Innovations, n.d.).

### 4.3.4 Integração da realidade mista

O *software* misturador de realidade utilizado foi o *Reality Mixer*, referido anteriormente. Este *software* não tem acesso direto às imagens captadas pelo HMD. Para isto, necessita de um *desktop* virtual que permita *passthrough*. O *software* de *desktop* virtual utilizado foi o *Virtual Desktop*. Após o acesso do *Reality Mixer* às imagens captadas pelo HMD, foi necessário definir o espaço afeto às imagens reais. Este *software* gera “caixas de realidade”, ou seja, paralelepípedos possíveis de editar lateralmente, longitudinalmente e em altura.

Uma limitação identificada neste processo nas potencialidades do computador, nomeadamente da placa gráfica, uma vez que esta processa cada um destes paralelepípedos individualmente, mesmo que estejam sobrepostos. Este fator tornou-se significativo nas tentativas de fazer uma junção de “caixas” que permitissem um recorte mais arredondado do *cockpit* da aeronave, uma vez que resultou numa perda muito significativa de desempenho do computador. Assim, optou-se pela utilização de 2 “caixas”: uma com vista para a parte inferior do *cockpit* e outra para o painel de instrumentos. O *Reality Mixer* é inicializado juntamente com o *SteamVR*, funcionando em paralelo com o MSFS2020, e permite ser ajustado diretamente neste. A figura seguinte mostra como estas “caixas” foram desenhadas para realizar o recorte e a disposição do *cockpit* em MR.



Figura 6: Disposição do *cockpit* MR

Uma dificuldade sentida durante a utilização do simulador em modo MR foi a latência, ou seja, uma decadência no tempo de resposta do simulador, que é perceptível por paragens momentâneas da simulação (Chen, Yang, & Chen, 2013). Esta latência não é constante ao longo de toda a simulação, manifestando-se apenas de forma momentânea, geralmente relacionada com mudanças repentinas de fase de voo e movimentos rápidos da cabeça do utilizador. Apesar de não representar um problema grave para a simulação, a sua ocorrência pode levar a lapsos de imersão e até a enjoos, uma vez que perturba a percepção visual do movimento. Esta latência pode ser causada por vários fatores, sendo um dos principais o funcionamento simultâneo do *software* de simulação com o *software* misturador de realidade. Esta degradação está diretamente relacionada com a capacidade da GPU, mais especificamente com o tamanho da VRAM, que é responsável por processar ambos os programas em simultâneo e em paralelo.

Assim, uma potencial solução poderá passar pela substituição da GPU por uma de características superiores ou pela instalação de duas placas gráficas, sendo que uma destas seria dedicada exclusivamente ao *software* misturador de realidade. Dessa forma, reduzir-se-ia a carga de trabalho da GPU principal e aumentaria a fluidez da simulação.

## 4.4 Sistema de simulação multiplataformas

Este sistema permite a replicação, quer de aeronaves de asa fixa, quer de asa rotativa. Uma vez que cada aeronave tem as suas especificidades e dado que esta plataforma servirá apenas como introdução básica às mesmas, os sistemas construídos são transversais a todas: no caso de aeronaves de asa fixa, um quadrante com manetes de potência, de passo e de mistura e um compensador do eixo lateral; no caso de aeronaves de asa rotativa, um coletivo; comum a ambos, um manche/cíclico e pedais.

#### 4.4.1 Controlos de voo

A execução dos controlos de voo passou por manufatura aditiva, o que permite uma manutenção mais fácil e direta. Assim, foi necessário modelar os controlos de voo e integrá-los como HOTAS, uma vez que estes serão utilizados em VR, o que não permite a visualização dos mesmos. Para além disto, por forma a incrementar a realidade dos controlos de voo, foram utilizados componentes de um helicóptero *Alouette III*, nomeadamente o cíclico, o coletivo e os pedais, tendo sido necessário criar um sistema para a implementação destes. No caso do manche/cíclico, foi necessário desenvolver um sistema que permitisse a rotação e a medição dos dois eixos, bem como um sistema de *centering* e de *force feedback*. Para além disto, apesar de ser utilizado o cíclico do *Alouette III*, foi desenvolvido um sistema que permite acoplar e desacoplar este. Foi também desenvolvido um sistema de engrenagens de forma a aumentar a resolução dos potenciômetros.

A figura seguinte mostra a modulação 3D e o respetivo sistema real, sem e com sistema de *centering* e *force feedback*.

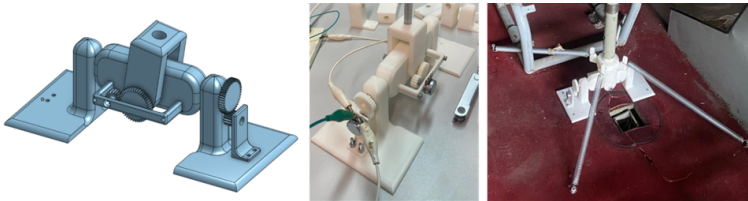


Figura 7: Modelação 3D e sistema real, sem e com o sistema de *centering* e *force feedback*, do manche/cíclico

No caso dos pedais e do coletivo, a aplicação nestes passou por uma adaptação dos eixos já presentes na base do simulador de *Alouette III*. Assim, foi necessário criar um sistema de engrenagens, por forma a aumentar a amplitude de leitura dos potenciômetros. A figura seguinte mostra as modelações 3D e a aplicação real dos sensores.

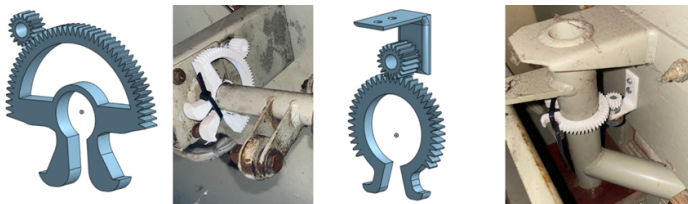


Figura 8: Modelação 3D e aplicação real dos pedais e do coletivo

No caso do quadrante, foi desenvolvido um sistema de engrenagens, quer para as manetes, quer para o compensador. A figura seguinte ilustra a modelação 3D e o respetivo sistema real.

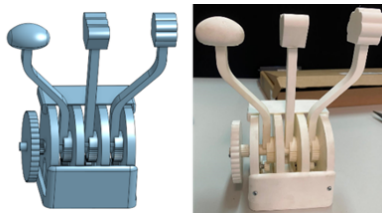


Figura 9: Modelação 3D e sistema real do quadrante

A calibração destes sensores seguiu o mesmo padrão do apresentado anteriormente para os controlos de voo do



sistema de simulação *Chipmunk*.

## 4.5 Circuitos eletrônicos

Para ambos os sistemas de simulação foram produzidos circuitos eletrônicos, com vista a facilitar o processo de manutenção/substituição dos componentes. Isto reveste-se de valor, uma vez que permite a sustentação dos sistemas, dado que, caso algum componente avarie, este possa ser diretamente substituído, sem que coloque em causa a totalidade do sistema. Para cada um dos sistemas, foram soldados a uma placa de circuito perfurada conectores *Stackable*, os quais permitem que os *Arduínos* não sejam diretamente soldados ao circuito e que estes possam ser colocados ou retirados. Para além disto, foram também soldados às placas de circuito perfuradas conectores terminais com parafuso, que permitem colocar e retirar os fios elétricos conectados aos potenciômetros e aos interruptores. Por fim, foram modeladas e produzidas, através de manufatura aditiva, caixas de proteção para estes circuitos, por forma a que estivessem protegidas e acomodadas, permitindo, no entanto, serem facilmente acessíveis a ações de manutenção/substituição. A figura seguinte mostra o circuito produzido para o sistema de simulação *Chipmunk* e a respetiva caixa de proteção.

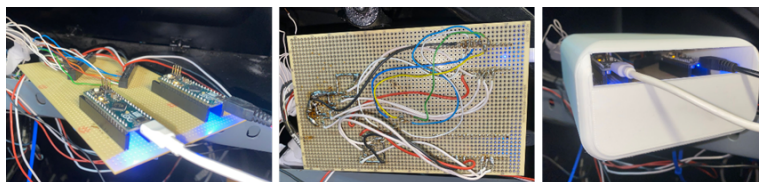


Figura 10: Circuito eletrónico do sistema de simulação *Chipmunk*

O código e os desenhos de modelação 3D desenvolvidos para este projeto estão disponíveis no repositório *GitHub* no endereço [https://ciafa.academiafa.edu.pt/sim\\_mr/](https://ciafa.academiafa.edu.pt/sim_mr/).

## 5 Discussão dos resultados e proposta de alteração

Foi possível realizar um inquérito aos alunos e instrutores de voo, onde foram questionados relativamente à fidelidade aparente do simulador em relação à replicação de manobras e parâmetros do mesmo. Para além disso, também foram feitas questões relativas aos efeitos esperados da implementação deste simulador na instrução de voo e em que fases poderia ter um impacto mais significativo. Assim, tendo por base as respostas dos alunos e instrutores de voo, e com referência ao que já é praticado na Esquadra 101, onde os simuladores estão inseridos na instrução de voo, foi possível elaborar uma proposta de integração deste no atual *Syllabus* da Esquadra 802.

As manobras consideradas foram: descolagem, compensação, perdas, acrobacia (a qual compreende *looping*, trevo e *tonneaux* barrilado), circuito de aterragem, simulação de aterragem forçada e aterragem. Praticamente todos os parâmetros avaliados tiveram uma classificação igual ou superior a 4, numa escala de 1 a 5. As manobras que ficaram aquém foram a descolagem e a aterragem, tendo contribuído para isto a excessiva sensibilidade dos pedais no solo e a falha na replicação correta do efeito de torque na aeronave. A sensibilidade não pode ser alterada, uma vez que não é possível diferenciá-la entre o voo e o solo, tendo sido dada primazia ao voo. Relativamente ao efeito de torque, este poderá ser um problema de fidelidade do modelo de voo utilizado. Foi possível também aferir que estes simuladores poderão ter um papel muito positivo, principalmente nas modalidades de voo de contacto (C1 e C2). No entanto, também poderão desempenhar um papel relevante na modalidade de voo de navegação (N1).

Assim, uma proposta de alteração possível do *Syllabus* da Esquadra 802 passaria, não pela redução das horas de voo previstas, mas pela inclusão dos simuladores como forma de consolidação e preparação para o voo real. A inclusão dos simuladores seria, à semelhança do que é feito na Esquadra 101 e corroborado pelo inquérito, integrada na preparação da fase de voo. Assim, seria requisito para os alunos iniciarem uma nova fase, a conclusão de um número determinado de missões em simulador, onde seriam demonstradas e praticadas as manobras específicas de cada fase, bem como

situações anômalas. Esta proposta visa a implementação dos simuladores, não como forma de reduzir as horas de voo de instrução, mas para as complementar, contribuindo para uma instrução fluida e sem perdas de continuidade de voo.

## 6 Conclusão

Este trabalho procurou comprovar que, com softwares COTS, de fácil aquisição e amplamente disponíveis, é possível, de forma acessível e com custos reduzidos, transformar um simulador de voo num simulador de voo VR/MR.

A possibilidade de ver o interior do *cockpit* permite que a simulação possibilite uma transferência de treino mais elevada, quando comparada com a simulação VR, uma vez que permite a criação de automatismos e a formação de memória muscular, o que reduzirá a carga mental no voo real. Foi possível desenvolver uma metodologia geral a adotar na criação de um simulador MR/VR, bem como os *softwares* necessários para este e a sua interligação, além da aplicação desta metodologia num caso concreto. Com isto, foi possível desenvolver um sistema de simulação MR e um sistema de simulação VR, sendo este último capaz de simular o voo de uma qualquer aeronave de asa fixa ou de asa rotativa. Para além disto, foi possível avaliar qual seria a melhor forma de implementar o sistema de simulação *Chipmunk* no atual *Syllabus* de instrução de voo da Esquadra 802, seguindo a implementação já existente de simuladores na instrução de voo na Esquadra 101.

Além disso, os componentes que relacionam os sensores com os eixos de controlo foram produzidos através de manufatura aditiva. Este facto garante a sustentação do simulador, uma vez que permite uma fácil manutenção, substituição de peças degradadas pelo uso e adaptação/melhoria do simulador. De igual forma, uma vez que foram produzidos circuitos eletrónicos que permitem a remoção dos *Arduínos* e das ligações aos potenciômetros e interruptores, permitirá também uma fácil substituição destes, devido à degradação natural relacionada com o uso. Concomitantemente, a utilização de produtos COTS e *open-source* permite que a manutenção e a operação destes não precisem ser realizadas por técnicos especializados.

Finalmente, este trabalho permitiu a criação de uma plataforma que tem o potencial de tornar mais eficiente e eficaz a instrução de voo, visto que contribui para a diminuição da descontinuidade de voo dos alunos. Permite também um contacto mais regular com o ambiente aeronáutico, servindo como motivador para estes.

São de destacar três caminhos que poderão ser seguidos após este trabalho: atualização do simulador, com a substituição da GPU ou a utilização de duas GPUs, com a criação de um segundo simulador e a ligação destes de forma a ser possível fazer voo de formação, com a criação de um simulador dedicado apenas ao voo por instrumentos, com o desenvolvimento de ferramentas para assistir o *debriefing* de missões, simuladas ou reais, ou com a criação de um sistema que permita a replicação de comunicações, entre outras; a tentativa de aplicação num outro sistema de armas (SA); e a aferição da validade, fidelidade e transferência de treino do simulador e os seus efeitos nos Alunos Pilotos (AP).

## Referências Bibliográficas

- Boeing. (2024, July). *Boeing Projects Global Need for Nearly 2.4 Million New Commercial Pilots, Technicians, Cabin Crew*. Retrieved 2024-10-16, from <https://www.prnewswire.com/news-releases/boeing-projects-global-need-for-nearly-2-4-million-new-commercial-pilots-technicians-cabin-crew-302201002.html>
- Borgvall, J., Castor, M., Svensson, E., Nählinder, S., & Oscarsson, P.-A. (2008). *Transfer of Training in Military Aviation*. doi: 10.13140/RG.2.1.5087.7280
- Bowman, D. A., & McMahan, R. P. (2007, July). Virtual Reality: How Much Immersion Is Enough? *Computer*, 40(7), 36–43. Retrieved 2024-01-10, from <https://ieeexplore.ieee.org/document/4287241> (Conference Name: Computer) doi: 10.1109/MC.2007.257
- Bradley, D. R., & Abelson, S. B. (1995, June). Desktop flight simulators: Simulation fidelity and pilot performance. *Behavior Research Methods, Instruments, & Computers*, 27(2), 152–159. Retrieved 2024-01-12, from <https://doi.org/10.3758/BF03204721> doi: 10.3758/BF03204721
- Chen, D. Y., Yang, H. T., & Chen, K. T. (2013). Dude, the source of lags is on your computer. In *2013 12th Annual Workshop on Network and Systems Support for Games, NetGames 2013* (p. 6820608). IEEE Computer Society. Retrieved 2024-11-15, from <https://scholars.ncu.edu.tw/en/publications/dude-the-source-of-lags-is-on-your-computer> doi: 10.1109/NetGames.2013.6820608

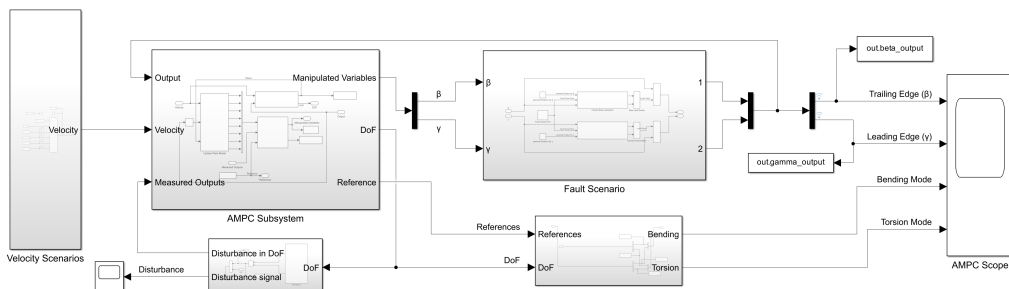
- EASA. (2023, March). *FSTD Special Conditions for the use of Head Mounted Displays (HMD) combined with a motion platform with reduced envelope*. Retrieved from <https://www.easa.europa.eu/en/downloads/137723/en>
- Estrada, A., & Adam, G. E. (2001, July). *Use of Simulator Spatial Disorientation Awareness Training Scenarios by the U.S. Army and National Guard*: (Tech. Rep.). Fort Belvoir, VA: Defense Technical Information Center. Retrieved 2024-01-10, from <http://www.dtic.mil/docs/citations/ADA395900> doi: 10.21236/ADA395900
- FAA. (2020). *Aviation Instructor's Handbook (FAA-H-8083-9)*. Retrieved 2024-06-09, from [https://www.faa.gov/regulations\\_policies/handbooks\\_manuals/aviation/aviation\\_instructors\\_handbook](https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/aviation_instructors_handbook)
- Freina, L., & Ott, M. (2015, April). A Literature Review on Immersive Virtual Reality in Education: State Of The Art and Perspectives.. doi: 10.12753/2066-026X-15-020
- Guthridge, R., & Clinton-Lisell, V. (2023, July). Evaluating the Efficacy of Virtual Reality (VR) Training Devices for Pilot Training. *Journal of Aviation Technology and Engineering*, 12(2). Retrieved from <https://docs.lib.purdue.edu/jate/vol12/iss2/1> doi: 10.7771/2159-6670.1286
- Hawkins. (2019). *PTN innovations move to undergraduate pilot training*. Retrieved 2024-01-10, from <https://www.af.mil/News/Article-Display/Article/1784958/ptn-innovations-move-to-undergraduate-pilot-training/>  
<https%3A%2F%2Fwww.af.mil%2FNews%2FArticle-Display%2FArticle%2F1784958%2Fptn-innovations-move-to-undergraduate-pilot-training%2F>
- Heironimus, M. (2024, October). *Arduino Joystick Library*. Retrieved 2024-10-18, from <https://github.com/MHeironimus/ArduinoJoystickLibrary> (original-date: 2015-11-19T14:10:12Z)
- Holmes, M. (2019). *Rebuilding the Forge: Reshaping How the Air Force Trains Fighter Aviators*. Retrieved 2024-01-10, from <https://warontherocks.com/2019/01/rebuilding-the-forge-reshaping-how-the-air-force-trains-fighter-aviators/>
- IBM. (2023, October). *What Is a Virtual Desktop?* Retrieved 2024-06-16, from <https://www.ibm.com/topics/virtual-desktop>
- Inside, R. (2022, January). *Reality Mixer - Mixed Reality for VR headsets no Steam*. Retrieved 2024-01-26, from [https://store.steampowered.com/app/1844610/Reality\\_Mixer\\_Mixed\\_Reality\\_for\\_VR\\_headsets/](https://store.steampowered.com/app/1844610/Reality_Mixer_Mixed_Reality_for_VR_headsets/)
- Intel. (n.d.). *Intel® Core™ i7 processor 14700KF*. Retrieved 2024-06-04, from <https://www.intel.com/content/www/us/en/products/sku/236789/intel-core-i7-processor-14700kf-33m-cache-up-to-5-60-ghz/specifications.html>
- Jensen, L., & Konradsen, F. (2018, July). A review of the use of virtual reality head-mounted displays in education and training. *Education and Information Technologies*, 23(4), 1515–1529. Retrieved 2024-01-25, from <https://doi.org/10.1007/s10639-017-9676-0> doi: 10.1007/s10639-017-9676-0
- Li, X., You, J., Zhou, X., Chen, R., & Tang, C. (2018, January). *HID Architecture and Design in Embedded USB Host System*.. doi: 10.2991/nccse-18.2018.72
- Meta. (n.d.). *Meta Quest 3: New Mixed Reality VR Headset - Shop Now | Meta Store | Meta Store*. Retrieved 2024-06-07, from <https://www.meta.com/quest/quest-3/#tour>
- Meta. (2024, May). *Requirements to use Meta Quest Link*. Retrieved 2024-06-04, from <https://www.meta.com/pt-pt/help/quest/articles/headsets-and-accessories/oculus-link/requirements-quest-link/>
- Microsoft. (2024, March). *Minimum, Recommended, and Ideal PC requirements for Microsoft Flight Simulator*. Retrieved 2024-04-16, from <https://flightsimulator.zendesk.com/hc/en-us/articles/360013463459-Minimum-Recommended-and-Ideal-PC-requirements-for-Microsoft-Flight-Simulator>
- Murray, J. W. (2017). *Building Virtual Reality with Unity and Steam VR*. CRC Press, Taylor & Francis Group. (Google-Books-ID: WZ8.MQAACAAJ)
- Myers, P., Starr, A., & Mullins, K. (2018). Flight Simulator Fidelity, Training Transfer, and the Role of Instructors in Optimizing Learning. *International Journal of Aviation, Aeronautics, and Aerospace*. Retrieved 2024-01-05, from <https://commons.erau.edu/ijaaa/vol5/iss1/6/> doi: 10.15394/ijaaa.2018.1203
- Nounou, M. N., & Bakshi, B. R. (2000, January). Chapter 5 - Multiscale Methods for Denoising and Compression. In B. Walczak (Ed.), *Data Handling in Science and Technology* (Vol. 22, pp. 119–150). Elsevier. Retrieved 2024-11-13, from <https://www.sciencedirect.com/science/article/pii/S0922348700800301> doi: 10.1016/S0922-3487(00)80030-1
- Pope, T. (2019, March). *A Cost-Benefit Analysis of Pilot Training Next*. *Theses and Dissertations*. Retrieved from <https://>

scholar.afit.edu/etd/2314

- Regian, Shebilske, & Monk. (1993). *A Preliminary Empirical Evaluation of Virtual Reality as a Training Tool for Visual-Spatial Tasks*. Retrieved 2024-01-10, from [https://www.researchgate.net/publication/235198099\\_A\\_Preliminary\\_Empirical\\_Evaluation\\_of\\_Virtual\\_Reality\\_as\\_a\\_Training\\_Tool\\_for\\_Visual-Spatial\\_Tasks](https://www.researchgate.net/publication/235198099_A_Preliminary_Empirical_Evaluation_of_Virtual_Reality_as_a_Training_Tool_for_Visual-Spatial_Tasks)
- rkApps. (n.d.). *SimXR*. Retrieved 2024-06-09, from <https://www.rkapps.shop/products/simxr>
- Seymour, N. E., Gallagher, A. G., Roman, S. A., O'Brien, M. K., Bansal, V. K., Andersen, D. K., & Satava, R. M. (2002, October). Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of Surgery*, 236(4), 458–463; discussion 463–464. doi: 10.1097/00000658-200210000-00008
- Sim Innovations. (n.d.). *Air Manager 4 Desktop*. Retrieved 2024-11-15, from <https://www.siminnovations.com/shop/air-manager-desktop/>
- Steamworks. (n.d.). *SteamVR para utilização por empresas e entidades governamentais (Documentação do Steamworks)*. Retrieved 2024-06-09, from <https://partner.steamgames.com/doc/features/steamvr/enterprise>
- Sérgio A. Vitorino. (2025, February). Dois pilotos por semana 'voam' da Força Aérea. *Correio da Manhã*. Retrieved 2025-02-18, from <https://www.cmjornal.pt/portugal/detalhe/dois-pilotos-por-semana-voam-da-forca-aerea>
- TechPowerUp. (2024, June). *NVIDIA GeForce RTX 2080 Specs*. Retrieved 2024-06-04, from <https://www.techpowerup.com/gpu-specs/geforce-rtx-2080.c3224>
- Thomas R. Carretta, & Ronald D. Dunlap. (1998, September). *Transfer of Training Effectiveness in Flight Simulation: 1986 to 1997*. (Tech. Rep.). Air Force Research Laboratory. Retrieved 2024-06-08, from <https://apps.dtic.mil/sti/citations/ADA362818>
- Tremosa, L. (2023, July). *Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR?* Retrieved 2024-04-09, from <https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr>
- Varjo. (2021, July). *Mixed and Virtual Reality for Pilot Training and Simulation* (Whitepaper). Retrieved 2024-06-09, from <https://varjo.com/e-book/white-paper-virtual-and-mixed-reality-for-pilot-training-and-simulation/>
- Vincenzi, D., Liu, D., & Macchiarella, N. (2008, December). Simulation Fidelity. In *Human factors in simulation and training* (pp. 61–73). (Journal Abbreviation: Human factors in simulation and training) doi: 10.1201/9781420072846.ch4
- Wang, K. (2023). *Comparing MR/VR implementations in flight training simulation* (Doctoral dissertation, Stockholm University). Retrieved 2024-12-13, from <https://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-219624>



# Design of a Fault Tolerant Active Flutter Suppression System using Model Predictive Control



Autor: **ALF/ENGEL Francisco Manuel Costa Ribeiro**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

Orientador: **Prof. Afzal Suleman**  
Instituto Superior Técnico, Lisboa

Coorientador: **Dr. Mário Brás**  
University of Victoria, Canada

## Abstract

The development of adaptive control systems for flexible aeroelastic wing structures addresses the challenge of maintaining stability in high-aspect ratio wings. While offering efficiency benefits, these wings exhibit complex aeroelastic effects such as flutter. Although effective under stable and pre trained conditions, traditional Model Predictive Control (MPC) systems lack the flexibility to respond to fast dynamic changes and control surface faults. An Adaptive Model Predictive Control (AMPC) system was developed to dynamically adapt to structural and aerodynamic changes, thereby enhancing fault tolerance and stability. The performance of the AMPC system was evaluated in scenarios involving flutter and control surface failures. Results indicate that the AMPC not only mitigates aeroelastic instabilities efficiently but also demonstrates potential for extending the operational limits of the wing, providing a robust fault-tolerant performance and enhancing safety margins. These findings highlight the AMPC potential as a solution for managing the rapidly changing dynamics associated with flight in the vicinity of flutter conditions.

**Keywords:** Active Flutter Suppression, Adaptive Model Predictive Control, Aeroelasticity, Fault-Tolerant Control.

## 1 Introduction

The drive to improve fuel economy and meet environmental targets is leading the aerospace sector toward lightweight structures and high-aspect-ratio wings (Afonso, Vale, Oliveira, Lau, & Suleman, 2017). These designs, however, introduce aeroelastic challenges that affect stability and safety, such as flutter and divergence. As structural flexibility increases, advanced control solutions are needed to maintain steady and predictable flight.

Traditional control systems, including standard Model Predictive Control (MPC), work well in stable, predictable conditions but adapt poorly to the rapidly changing dynamics and possible control surface faults in flight. The ability of standard MPC to manage variability and uncertainty in flexible wing structures is restricted by fixed model parameters (Virgilio Pereira, Kolmanovsky, Cesnik, & Vetrano, 2019).

This work develops an Adaptive Model Predictive Control (AMPC) framework for flexible aeroelastic systems. The objectives are to analyse the aeroelastic behaviour of a flexible wing under increasing airspeed, which can trigger flutter; to design an AMPC capable of maintaining stability where MPC may fail; and to evaluate the robustness of AMPC in flutter and control surface fault scenarios.

A MATLAB/Simulink simulation environment was built to test AMPC under these conditions, providing insight into its fault tolerance and adaptability. The results indicate that AMPC can extend operational safety limits for flexible aeroelastic systems in aerospace applications.

## 2 Background

Flexible, lightweight structures such as high-aspect-ratio wings improve fuel efficiency but are more susceptible to aeroelastic phenomena like flutter, caused by coupling between bending and torsion modes. Aeroelasticity studies the interaction between aerodynamic forces and structural flexibility in aircraft wings. Ensuring sufficient damping is essential to maintain safe and reliable flight, as inadequate damping can severely affect structural integrity (I. Garrick, 1976; The Grainger College of Engineering, University of Illinois Urbana-Champaign, n.d.).

Aeroservoelasticity combines aeroelastic knowledge with control systems to counteract aerodynamic forces. By adjusting control surfaces, these systems actively mitigate oscillations from flight disturbances, improving stability and extending safety limits for flexible wing designs (Livne, 1999).

Flutter is an undesirable phenomenon that can lead to structural failure through self-sustained oscillations. Higher airspeeds increase the likelihood and severity of flutter (Banerjee, Liu, & Kassem, 2014; Peter Katz, n.d.). While passive solutions such as structural reinforcement can mitigate flutter, they increase weight and cost. Moreover, flutter onset depends directly on stiffness and indirectly on strength (I. E. Garrick & Reed, 1981). Active control is often a more efficient approach (Livne, 2018).

Model Predictive Control (MPC), widely used in mechanical and chemical industries, is also applicable in aerospace. Its ability to predict system behaviour and optimise control actions via a cost function distinguishes it from other controllers (Qin & Badgwell, 2003). Once limited by computational demands, MPC is now practical due to advances in processing power. However, fixed model parameters reduce effectiveness in highly dynamic environments, creating the need for adaptive strategies.

Adaptive Model Predictive Control (AMPC) extends MPC by updating the model as environmental conditions change (Köhler, Kötting, Soloperto, Allgöwer, & Müller, 2020). This adaptability enables robust performance under varying flight conditions, including high airspeeds and control surface failures, addressing MPC limitations.

Operational safety is a key requirement in aviation, making Fault Tolerant Control (FTC) an important consideration. FTC strategies ensure operation despite failures in critical components such as actuators. Passive FTC can handle predefined faults within a set range, while active FTC responds dynamically to faults in real time, making it suitable for unpredictable environments (Jiang, 2005). Active FTC is not addressed in this work.

### 3 Methodology

#### 3.1 Binary Aeroelastic Model

A binary aeroelastic model, which constitutes bending and torsion modes, is used in this research to analyze the response of the flexible wing structure. While bending consists in the vertical displacement of the wing due to lift, torsion is the twisting motion about the elastic axis caused by aerodynamic moments, as shown by Figure 1.

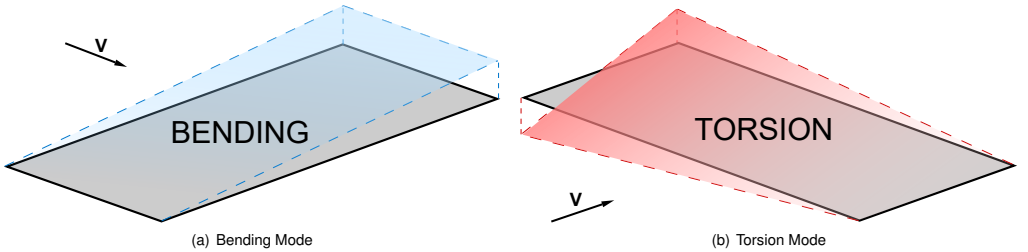


Figure 1: Schematic representation of the two primary aeroelastic modes.

Bending significantly impacts the load on the wing, inducing compressive stresses on one side and tensile stresses on the other. Torsion is critical in flutter analysis, as it modifies the angle of attack of the wing and, consequently, the aerodynamic forces acting on it. Torsion also induces shear stresses in the cross-section, which can become problematic when excessive, leading to instability.

#### 3.2 Equations of Motion

To represent the aeroelastic behavior of the system, the equations of motion were derived from the Lagrangian equation as

$$\mathcal{L} \equiv T - U, \quad (1)$$

where  $T$  is the kinetic energy and  $U$  is the potential energy.

In a classical problem involving a mass attached to a spring, the kinetic and potential energies can be expressed as

$$T = \frac{1}{2}m\dot{x}^2, \quad U = \frac{1}{2}kx^2, \quad (2)$$

where  $m$  represents the mass,  $x$  and  $\dot{x}$  the position and velocity respectively.



Thus, the Lagrangian becomes

$$\mathcal{L} = \frac{1}{2}m\dot{x}^2 - \frac{1}{2}kx^2. \quad (3)$$

The Euler-Lagrange equation is then used to derive the equations of motion from the Lagrangian. It is expressed as

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{x}} \right) = \frac{\partial \mathcal{L}}{\partial x}. \quad (4)$$

This equation relates the rate of change of the kinetic energy of the system (with respect to velocity) with its potential energy (with respect to position). For the mass-spring system, the Lagrangian yields

$$\frac{\partial \mathcal{L}}{\partial \dot{x}} = m\dot{x}, \quad \frac{\partial \mathcal{L}}{\partial x} = -kx. \quad (5)$$

Substituting into the Euler-Lagrange equation gives

$$m\ddot{x} = -kx. \quad (6)$$

This is the second-order differential equation of motion for a harmonic oscillator, which dictates the behavior of the system. It is also analogous to Newton's second law,  $F = ma$ , where the restoring force is proportional to the displacement (Morin, n.d.).

In this specific case, to represent the binary aeroelastic model through its equations of motion, it is necessary to describe the surface motion of the wing in terms of torsional and bending

$$z(x, y, t) = \left( \frac{y}{s} \right)^2 q_b(t) + \left( \frac{y}{s} \right) (x - x_f) q_t(t), \quad (7)$$

where the displacement  $z$  represents the transverse displacement of a point on the wing, where positive displacements are defined as downwards. The generalized coordinates  $q_b(t)$  and  $q_t(t)$  represent the bending and torsional motions of the wing, respectively.

The kinetic energy of the system is expressed as

$$\begin{aligned} T &= \int_{\text{Wing}} \frac{1}{2} dm \dot{z}^2 \\ &= \frac{m}{2} \int_0^s \int_0^c \left( \left( \frac{y}{s} \right)^2 \dot{q}_b + \left( \frac{y}{s} \right) (x - x_f) \dot{q}_t \right)^2 dx dy \end{aligned} \quad (8)$$

where  $\dot{z}$  represents the vertical velocity of a general point on the wing (which incorporates contributions from both bending and torsional motions). The term  $dm$  is an infinitesimal mass element of the wing, while  $m$  is the total mass of the wing. Respectively, the parameters  $s$  and  $c$  represent the span and chord length of the wing. The variables  $y$  and  $x$  represent the spanwise and chordwise coordinates of the wing, respectively, while  $x_f$  represents the distance of the aeroelastic axis from the leading edge. The generalized coordinates  $\dot{q}_b$  and  $\dot{q}_t$  represent the time derivatives of the bending and torsional motions, respectively.

Similarly, the potential energy of the system is given by

$$U = \frac{1}{2} \int_0^s EI \left( \frac{2q_b}{s^2} \right)^2 dy + \frac{1}{2} \int_0^s GJ \left( \frac{q_t}{s} \right)^2 dy. \quad (9)$$

where the term  $EI$  represents the bending stiffness,  $E$  is Young's modulus, and  $I$  is the second moment of area. Likewise,  $GJ$  represents the torsional stiffness, where  $G$  is the shear modulus, and  $J$  is the torsional constant (which depends on the wing's cross-sectional shape). The generalized coordinates  $q_b$  and  $q_t$  represent the bending and torsional displacements, respectively, while  $s$  is the span of the wing over which the integration is performed.

Combined, both  $T$  and  $U$  portray the dynamic behavior of the wing in aeroelastic conditions. The total energy of the system is crucial to derive the equations of motion using the Euler-Lagrange formulation.

The partial derivatives of the kinetic energy with respect to the generalized velocities are given by

$$\frac{\partial T}{\partial \dot{q}_b} = m \int_0^s \int_0^c \left( \left( \frac{y}{s} \right)^4 \dot{q}_b + \left( \frac{y}{s} \right)^3 (x - x_f) \dot{q}_t \right) dx dy. \quad (10)$$

Taking the time derivative of this expression yields

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_b} \right) = m \left[ \frac{sc}{5} \ddot{q}_b + \frac{s}{4} \left( \frac{c^2}{2} - cx_f \right) \ddot{q}_t \right]. \quad (11)$$

Similarly, the partial derivative of the kinetic energy with respect to  $\dot{q}_t$  is

$$\frac{\partial T}{\partial \dot{q}_t} = m \int_0^s \int_0^c \left( \left( \frac{y}{s} \right)^3 (x - x_f) \dot{q}_b + \left( \frac{y}{s} \right)^2 (x - x_f)^2 \dot{q}_t \right) dx dy. \quad (12)$$

Taking the time derivative of this expression gives

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_t} \right) = m \left[ \frac{s}{4} \left( \frac{c^2}{2} - cx_f \right) \ddot{q}_b + \frac{s}{3} \left( \frac{c^3}{3} - c^2 x_f + x_f^2 \right) \ddot{q}_t \right]. \quad (13)$$

For the potential energy, the partial derivatives with respect to the generalized coordinates are given by

$$\frac{\partial U}{\partial q_b} = \int_0^s EI \left( \frac{4q_b}{s^4} \right) dy = \frac{4EI}{s^3} q_b, \quad (14)$$

$$\frac{\partial U}{\partial q_t} = \int_0^s GJ \left( \frac{q_t}{s^2} \right) dy = \frac{GJ}{s} q_t. \quad (15)$$

Since the potential energy does not depend on the velocities, the derivative of the potential energy with respect to the generalized velocities is zero

$$\frac{\partial U}{\partial \dot{q}_k} = 0 \quad (16)$$

The generalized forces  $Q_b$  and  $Q_t$  manifest as unsteady aerodynamic forces, which, for an oscillatory motion can be written through their derivatives in terms of the reduced frequency  $k = \omega c / (2V)$ . Although these forces are complex, they can be expressed as displacements and velocities in the real plane. Applying strip theory along the simplified unsteady aerodynamics representation, results in the lift and pitching moment for each elemental strip  $dy$

$$dL = \frac{1}{2} \rho V^2 c dy a_w \left( \frac{y^2}{s^2 V} \dot{q}_b + \frac{y}{s} \dot{q}_t \right), \quad (17)$$

$$dM = \frac{1}{2} \rho V^2 c^2 dy \left[ e a_w \left( \frac{y^2}{s^2 V} \dot{q}_b + \frac{y}{s} \dot{q}_t \right) + M_{\theta} c \left( \frac{y}{4sV} \right) \dot{q}_t \right]. \quad (18)$$

The incremental work done on the wing due to the incremental deflections  $\delta q_b, \delta q_t$ , is

$$\delta W = \int_{\text{Wing}} \left[ dL \left( - \left( \frac{y}{s} \right)^2 \delta q_b \right) + dM \left( \left( \frac{y}{s} \right) \delta q_t \right) \right]. \quad (19)$$

Keeping in mind that  $z$  is positive downwards, the generalized forces are

$$Q_{q_b} = \frac{\partial(\delta W)}{\partial(\delta q_b)} \quad (20)$$

$$= - \int_0^s \left( \frac{y}{s} \right)^2 dL \quad (21)$$

$$= - \frac{1}{2} \rho V^2 c a_w \int_0^s \left( \frac{y^4 \dot{q}_b}{s^4 V} + \frac{y^3}{s^3} \dot{q}_t \right) dy \quad (22)$$

$$= - \frac{1}{2} \rho V^2 c a_w \left( \frac{s}{5V} \dot{q}_b + \frac{s}{4} \dot{q}_t \right) \quad (23)$$

And in a similar manner for  $Q_{q_t}$

$$Q_{q_t} = \frac{\partial(\delta W)}{\partial(\delta q_t)} \quad (24)$$

$$= \int_0^s \frac{y}{s} dM \quad (25)$$

$$= \frac{1}{2} \rho V^2 c^2 \int_0^s \left[ e a_w \left( \frac{y^3 \dot{q}_b}{s^3 V} + \frac{y^2}{s^2} \dot{q}_t \right) + M_{\theta c} \left( \frac{y^2 \dot{q}_t}{4 s^2 V} \right) \right] dy \quad (26)$$

$$= \frac{1}{2} \rho V^2 c^2 \left[ e a_w \left( \frac{s}{4V} \dot{q}_b + \frac{s}{3} \dot{q}_t \right) + M_{\theta c} \left( \frac{s}{12V} \dot{q}_t \right) \right] \quad (27)$$

The aeroelastic equations are written in the classical second order form for  $N$  Degrees of Freedom (DoF) encapsulating the dynamic behavior of the system by considering the contributions of mass, aerodynamic forces and structural properties:

$$\mathbf{A} \ddot{\mathbf{q}} + (\rho V \mathbf{B} + \mathbf{D}) \dot{\mathbf{q}} + (\rho V^2 \mathbf{C} + \mathbf{E}) \mathbf{q} = 0. \quad (28)$$

In this equation,  $\mathbf{A}$  is the mass matrix, while  $\mathbf{B}$  and  $\mathbf{C}$  represent the aerodynamic damping and stiffness, respectively. Moreover,  $\mathbf{D}$  and  $\mathbf{E}$  represent the structural damping and stiffness. The variables  $\rho$  and  $V$  correspond to air density and freestream velocity, respectively, and the generalized coordinates  $q$  portray the bending and torsional displacements of the wing.

The full aeroelastic equations of motion describe the wing's dynamic response with the structural and aerodynamic effects accounted for. These are written in the matrix form as

$$m \left[ \begin{array}{cc} \frac{sc}{5} & \frac{s}{4} \left( \frac{c^2}{2} - c x_f \right) \\ \frac{s}{4} \left( \frac{c^2}{2} - c x_f \right) & \frac{s}{3} \left( \frac{c^3}{3} - c^2 x_f + x_f^2 \right) \end{array} \right] \begin{Bmatrix} \ddot{q}_b \\ \ddot{q}_t \end{Bmatrix} + \rho V \left[ \begin{array}{cc} \frac{cs}{10} a_w & 0 \\ -\frac{c^2 s}{8} e a_w & -\frac{c^3 s}{24} M_{\theta} \end{array} \right] \begin{Bmatrix} \dot{q}_b \\ \dot{q}_t \end{Bmatrix} + \left( \rho V^2 \left[ \begin{array}{cc} 0 & \frac{cs}{8} a_w \\ 0 & -\frac{c^2 s}{6} e a_w \end{array} \right] + \left[ \begin{array}{cc} \frac{4EI}{s^3} & 0 \\ 0 & \frac{GJ}{s} \end{array} \right] \right) \begin{Bmatrix} q_b \\ q_t \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}. \quad (29)$$

where the first matrix is the mass matrix (representing the inertial effects of the wing), the second matrix is the aerodynamic damping matrix (accounting for velocity-dependent aerodynamic forces). The third matrix is the aerodynamic stiffness matrix, representing the aerodynamic forces that are dependent on the displacement. Lastly, the resistance to deformation is represented by the structural stiffness matrix.

### 3.3 Control Surfaces: Trailing and Leading Edge

A Leading Edge (LE) and a Trailing Edge (TE) were implemented as the two control surfaces shown, respectively,

The  $c_w$  is the chord of the wing,  $s_w$  is the semi span of the wing, and  $x_f$  represents the location of the aeroelastic axis from the leading edge.

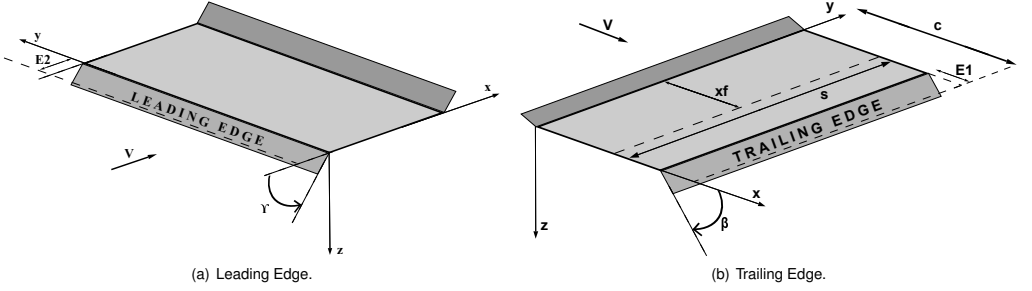


Figure 2: Schematic representation of both edges of the wing.

Additional terms were implemented in the lift and pitching moment equations. These control surfaces, represented by their deflection angles  $\beta$  and  $\gamma$ , contribute through the terms  $a_{C_1}\beta$ ,  $a_{C_2}\gamma$ ,  $b_{C_1}\beta$ , and  $b_{C_2}\gamma$ , which are added to the corresponding equations for lift and moment, as shown in (17) and (18).

$$dL = \frac{1}{2}\rho V^2 c dy \left[ a_w \left( \frac{y^2}{s^2 V} \dot{q}_b + \frac{y}{s} q_t \right) + a_{C_1}\beta + a_{C_2}\gamma \right] \quad (30)$$

$$dM = \frac{1}{2}\rho V^2 c^2 dy \left[ e a_w \left( \frac{y^2}{s^2 V} \dot{q}_b + \frac{y}{s} q_t \right) + M_{\theta} \frac{c y}{4 s V} \dot{q}_t + b_{C_1}\beta + b_{C_2}\gamma \right], \quad (31)$$

where  $a_{C_1}$  and  $a_{C_2}$  represent the control surfaces that influence lift, while  $b_{C_1}$  and  $b_{C_2}$  account for their effect on the pitching moment. The control surface deflection angles  $\beta$  and  $\gamma$  refer to the trailing edge and leading edge control surfaces, respectively. These terms enable the model to account for the aerodynamic influence of both control surfaces, therefore improving the control over the wing's dynamics.

As a consequence, the full aeroelastic equations of motion now becomes

$$\begin{aligned} m \begin{bmatrix} \frac{sc}{5} & \frac{s}{4} \left( \frac{c^2}{2} - c x_f \right) \\ \frac{s}{4} \left( \frac{c^2}{2} - c x_f \right) & \frac{s}{3} \left( \frac{c^3}{3} - c^2 x_f + x_f^2 \right) \end{bmatrix} \begin{Bmatrix} \dot{q}_b \\ \dot{q}_t \end{Bmatrix} + \rho V \begin{bmatrix} \frac{cs}{10} a_w & 0 \\ -\frac{c^2 s}{8} e a_w & -\frac{c^3 s}{24} M_{\theta} \end{bmatrix} \begin{Bmatrix} \dot{q}_b \\ \dot{q}_t \end{Bmatrix} + \left( \rho V^2 \begin{bmatrix} 0 & \frac{cs}{8} a_w \\ 0 & -\frac{c^2 s}{6} e a_w \end{bmatrix} + \begin{bmatrix} \frac{4EI}{s^3} & 0 \\ 0 & \frac{GJ}{s} \end{bmatrix} \right) \begin{Bmatrix} q_b \\ q_t \end{Bmatrix} = \\ = \rho V^2 \begin{bmatrix} -\frac{c a_{C_1} s}{6} + \frac{c^2 b_{C_1} s}{4} \\ \frac{c a_{C_2} s}{6} + \frac{c^2 b_{C_2} s}{4} \end{bmatrix} \begin{Bmatrix} \beta \\ \gamma \end{Bmatrix} \end{aligned} \quad (32)$$

This formulation includes the effect of two control surfaces in the right-hand side of the equation.

### 3.4 MPC and AMPC Model Design

The MPC and AMPC controllers were integrated into the aeroelastic model to suppress oscillations and improve overall stability. The design and simulation process was carried out in MATLAB/Simulink.

For the MPC implementation, the built-in controller block was configured by tuning the prediction horizon, control horizon, and cost function parameters to achieve optimal stability. The performance of the controller was evaluated under both nominal operating conditions and beyond-flutter scenarios, with particular focus on the ability to reject disturbances.

The AMPC architecture employed the *Update Plant Model* block to modify the plant discrete-time state-space matrices

at each simulation step. This enabled real-time adaptation of the control law to changing aerodynamic and structural conditions, enhancing robustness and maintaining stability under varying parameters.

3.5 Robustness Testing with Control Surface Failures

To assess AMPC robustness, control surface failures were simulated by jamming the LE and TE at fixed positions. This tested the controller’s ability to reallocate efforts and maintain stability under failure conditions, demonstrating potential resilience to further control surface faults.

4 Results and Discussion

Figure 3 shows the eigenvalue locus of the system and the interaction between bending and torsion modes. At low airspeeds, eigenvalues for both modes lie in the left half of the complex plane, indicating stability. With increasing airspeed, the eigenvalues approach the imaginary axis, showing reduced stability. At the flutter threshold, torsion mode eigenvalues cross into the right half-plane. This intersection point corresponds to the flutter speed,  $V_{flutter} = 90.90\text{ m/s}$ .

In order to design an effective controller capable of mitigating flutter, determining the flutter threshold is essential. Therefore, the interaction between bending and torsion modes was analyzed, as shown in Figure 3.

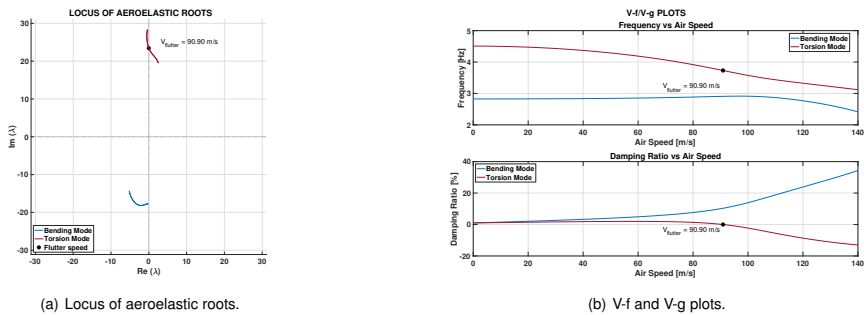


Figure 3: Wing aeroelastic behavior

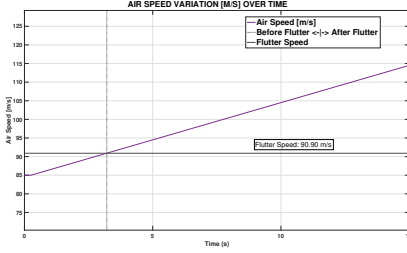
The top plot presents natural frequencies of the bending and torsion modes as a function of airspeed. At low velocities, the bending mode (blue) and torsion mode (red) remain distinct. As airspeed increases, the frequencies converge, indicating mode coupling. When they intersect, aerodynamic energy enters the structure, making oscillations self-sustaining. The bottom plot shows damping ratios for both modes as airspeed increases. At low speeds, positive damping indicates stable behavior with energy dissipation. As the flutter threshold approaches, the torsion mode damping decreases and becomes negative. Instability occurs when damping crosses zero, as energy accumulates in oscillations. Beyond flutter speed, the system cannot remain stable without active control.

4.1 Comparative Performance of Standard MPC and AMPC

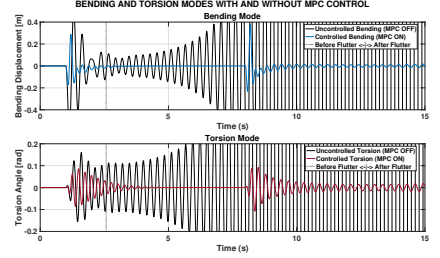
Both controllers were tested with disturbances applied before and after reaching the flutter threshold. Airspeed varied from 85 m/s (below flutter) to values above the threshold, allowing a direct comparison under identical conditions (Figure 4).

4.2 MPC Performance

The MPC was defined with a Prediction Horizon  $P = 20$  and a Control Horizon  $M = 10$ . Figure ?? shows that without control (OFF), both bending and torsion modes develop uncontrolled oscillations. With control (ON), oscillations are miti-



(a) Airspeed variation over time, showing the flutter threshold at 90.90 m/s.



(b) Wing aeroelastic behavior with MPC ON and OFF.

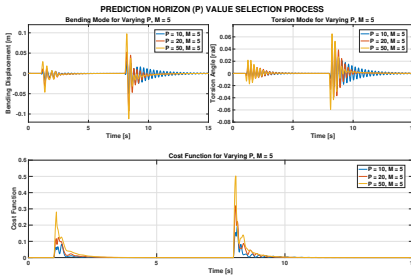
Figure 4: MPC performance below and beyond flutter speed.

gated, but beyond flutter, the MPC fails to fully suppress instability. This is because the MPC was not designed for such large speed variations.

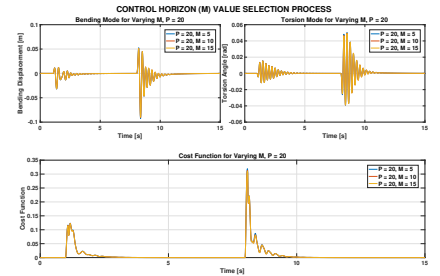
## 4.3 AMPC Tuning Process

In order to be able to compare the AMPC with the previous controller, a tuning process is required. Tuning is an essential part of a controller build up process since it extracts the best performance available taking into account the needs of the control engineer.

### 4.3.1 Prediction Horizon ( $P$ ) and Control Horizon ( $M$ ) Selection



(a) Effect of varying Prediction Horizon ( $P$ ) with fixed  $M = 5$ .



(b) Effect of varying Control Horizon ( $M$ ) with fixed  $P = 20$ .

Figure 5: Prediction Horizon ( $P$ ) and Control Horizon ( $M$ ) choosing value process.

First, the AMPC was set to have a fixed Control Horizon ( $M = 5$ ) and vary the Prediction Horizon ( $P$ ). The values chosen were 10, 20, and 50 steps.

This tuning plot suggests that  $P = 20$  offers the best balance of response speed, stability, and efficiency for the AMPC when  $M = 5$ . Although  $P = 10$  and  $P = 50$  offer sufficient stabilization, they show higher residual oscillations and higher cost, respectively. For further AMPC tuning, the ideal  $P = 20$  is thus selected.

Then, with Prediction Horizon set to  $P = 20$  the values selected to evaluate the Control Horizon were 5, 10 and 15 steps.

Different values for the Control Horizon do not have a significant impact on the overall performance of the AMPC, as demonstrated by the Figure 5. The bending and torsion modes, in addition to the Cost Function, continue to be relatively consistent with one another over the course of time. However, both modes are mitigated, despite the fact that there are

significant oscillations on both modes. The value that was chosen was a balanced one, which was  $M = 10$ . This was the result of the selection process.

4.3.2 Effect of Output Weight Variation on Stability and Cost Function

Figure 6 shows the effect of varying output weights on the bending and torsion modes for the AMPC as well as the cost function. The configurations of values chosen to give a first input of the effect of the Output Weights were  $[10, 1]$   $[1, 1]$  and  $[1, 10]$ .

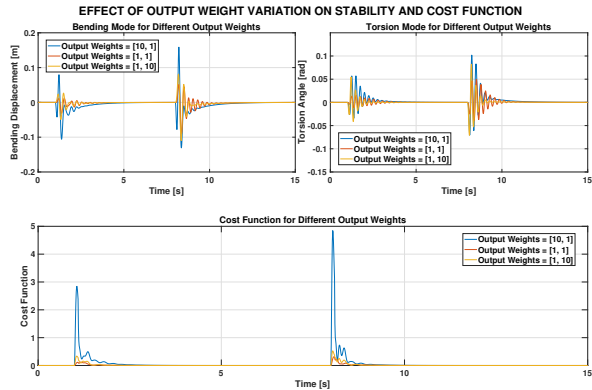


Figure 6: Effect of output weights on stability and cost function.

This study indicates that, particularly close to the flutter threshold, the  $[1, 10]$  configuration, which focuses on torsional stability, is ideal for uses when torsional control is given top priority. Conversely, the  $[10, 1]$  and  $[1, 1]$  configurations ought not to be chosen since they offer no appreciable benefit.

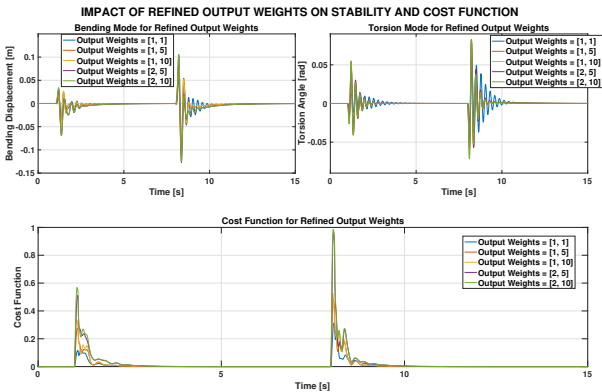


Figure 7: Refined output weight analysis.

4.3.3 Impact of Refined Output Weights on Stability and Cost Function

Testing-refined output weights  $[1, 1]$   $[1, 5]$   $[1, 10]$   $[2, 5]$  and  $[2, 10]$  provides a more thorough understanding of how these configurations affect system stability, especially around the flutter threshold.

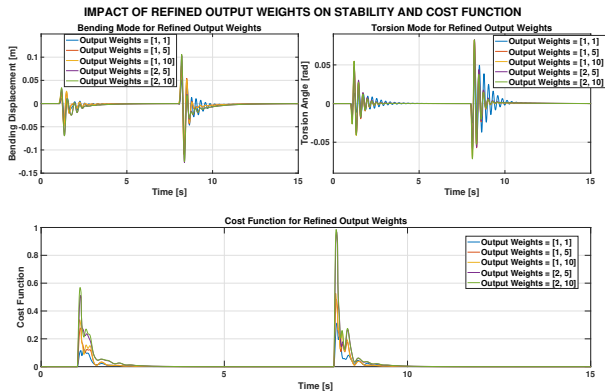


Figure 8: Impact of Refined Output Weights on Bending and Torsion Modes Stability and Cost Function.

Although  $[1, 1]$  has the lowest cost, the oscillations and slow response show that the refined output weights  $[1, 5]$  and  $[2, 5]$  provide a more effective damping across both bending and torsion modes, obtaining stable performance with minimum residual oscillations. Following result analysis, the configuration  $[1, 5]$  was chosen as the best one since its cost function is lower.

4.3.4 Effect of Manipulated Variable Weights on Stability and Cost Function

Figure 9 shows the effects of several manipulated variable weight configurations on the stability of the bending and torsion modes as well as on the Cost Function, for testing values of  $[1, 1]$   $[5, 5]$  and  $[10, 10]$ .

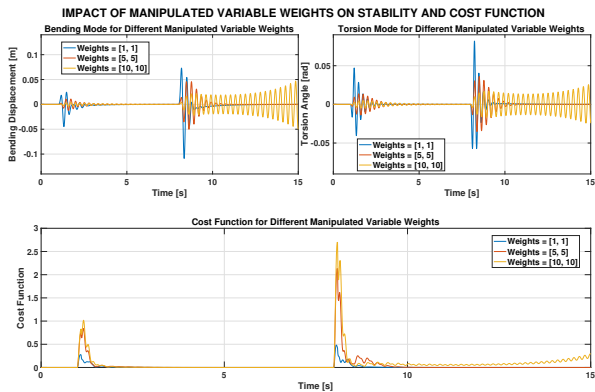


Figure 9: Bending and Torsion Modes Stability and Cost Function analysis for varying Manipulated Variable Weights.



The manipulated variable weights [1, 1] were selected as the optimal setting. This configuration minimizes control effort while enabling the AMPC to respond dynamically.

4.3.5 Stability and Cost Function Analysis for Different Rate Weights

Figure 10 investigates the impact of different rate weights on the stability of bending and torsion modes. Important for management of the aggressiveness of control actions to ensure smooth transitions and minimize oscillations, testing rate weight values [1, 1] [5, 5] and [10, 10] help analyze the impact of penalizing the rate of change of control inputs.

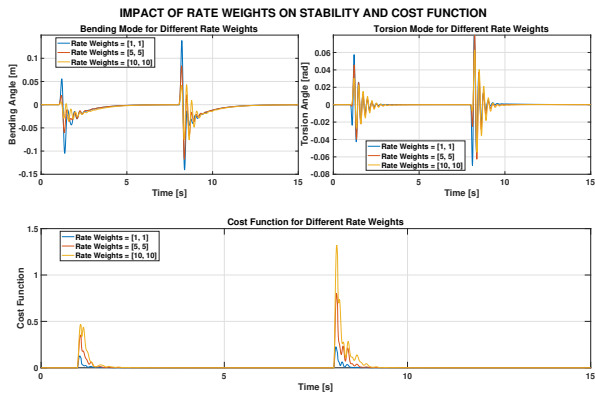


Figure 10: Bending and Torsion Modes Stability and Cost Function analysis for Different Weight Rate Conditions.

Overall, the ideal setting was chosen to be [1, 1] which offers minimal cost on the rate of control actions and smooth transitions.

Table 1 resumes the chosen values for the AMPC.

Table 1: Chosen values for the AMPC

| Parameter                    | Optimal Value | Reason                                                       |
|------------------------------|---------------|--------------------------------------------------------------|
| Prediction Horizon ( $P$ )   | 20            | Balances response speed and stability                        |
| Control Horizon ( $M$ )      | 10            | Provides efficient control with minimal oscillations         |
| Output Weights               | [1, 5]        | Prioritizes torsion stability with balanced bending response |
| Manipulated Variable Weights | [1, 1]        | Penalizes control usage for stability and flexibility        |
| Rate Weights                 | [1, 1]        | Controls aggressiveness of rate changes                      |

4.3.6 Flight Data Analysis for MPC and AMPC

Emphasizing both the bending and torsion modes as well as the TE and LE control surface deflections (defined respectively as  $\beta$  and  $\gamma$  for all Figures) in Figure 11, this section contrasts the flight data for the MPC and AMPC under an increasing airspeed profile. As airspeed exceeds the flutter threshold, this comparison helps evaluate each controller's performance in handling stability concerns.

As it can be observed, although both controllers can keep stability below the flutter threshold, when airspeed rises above this limit the AMPC beats the standard MPC. The AMPC achieves smoother control surface deflections and greater stability in bending and torsion modes.

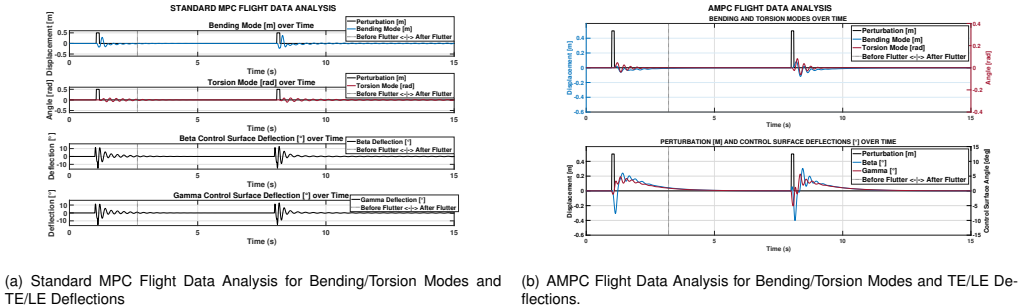


Figure 11: Flight Data Analysis for MPC and AMPC

#### 4.4 Cost Function Analysis for MPC and AMPC

Figure 12 shows the cost functions over time. This demonstrates the efficiency of each control surface to stabilize the system.

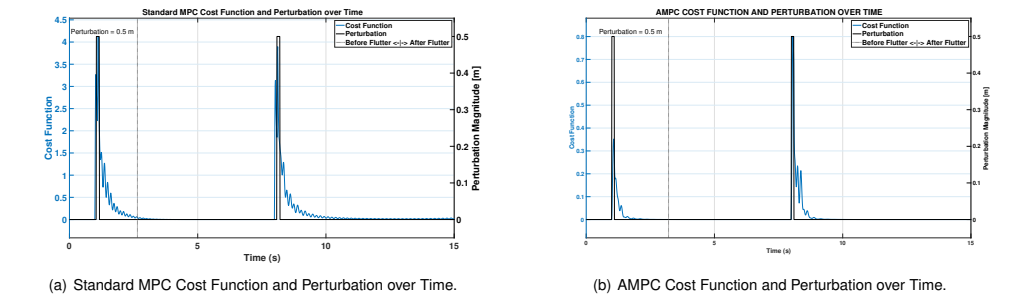


Figure 12: Cost Function Analysis for MPC and AMPC

Although both controllers can react to perturbations, the AMPC showcases a better performance than the MPC, given that it leads to a faster stabilization, keeping a lower cost, especially as airspeed increases. This performance advantage emphasizes the efficiency of the AMPC in responding to dynamic conditions, guaranteeing system stability with minimum control effort.

### 5 System Behavior under Control Surface Failure

This section examines the AMPC fault tolerance under control surface failures. Due to the high stability requirements beyond the flutter threshold, only the AMPC was used for these fault scenarios. Previous tests showed that the standard MPC struggled to maintain stability at airspeeds above the flutter threshold, while the AMPC adapted to dynamic conditions and effectively mitigated flutter.

#### 5.1 TE Control Surface Jammed at Zero Degrees

The TE control surface of the first fault scenario is jammed at zero degrees, reflecting a typical actuator failure mode.

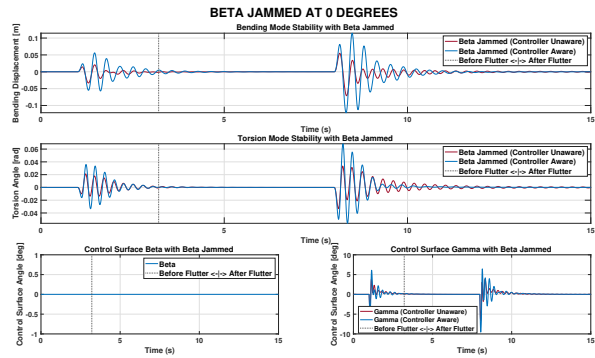


Figure 13: AMPC Performance with TE Control Surface Jammed at 0 Degrees: Bending/Torsion Mode Stability and Control Surface Deflections.

The top two plots of Figure 13 show the bending and torsion mode responses. In a beyond-flutter condition, the AMPC adapts its control strategy when aware of the jammed TE control surface, damping oscillations in both modes. In contrast, when unaware of the fault, both modes experience greater oscillations and longer instability periods as the controller continues to attempt activation of the jammed surface.

5.2 LE Control Surface Jammed at Zero Degrees

Under the second fault scenario, the LE control surface gets jammed at zero degrees.

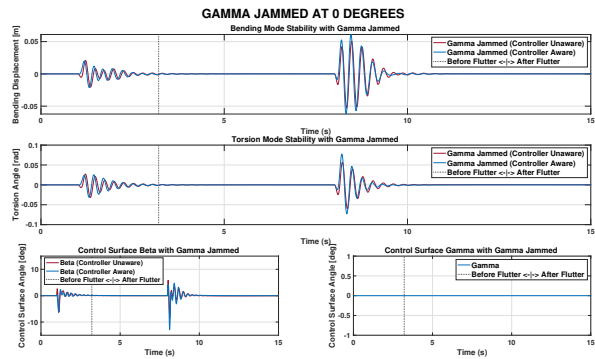


Figure 14: AMPC Performance with LE Control Surface Jammed at 0 Degrees: Bending/Torsion Mode Stability and Control Surface Deflections.

This is shown in the top two plots of Figure 14, when aware of the jammed LE surface, the AMPC redirects control actions to the TE surface, stabilizing the system and mitigating oscillations in both bending and torsion modes. When unaware, it continues attempting control changes on the jammed LE surface, making stabilization more difficult.

## 6 Conclusions

This work presented an AMPC made to tackle significant aeroelastic problems for flexible wing structures including flutter and control surface faults. By controlling changes in aerodynamic forces and structural dynamics, AMPC's adaptability improves stability over several flight conditions and surpasses MPC.

Key achievements include verifying AMPC's performance in fault conditions and investigating aeroelastic behavior. By reallocating control when faults arise, AMPC showed strong fault tolerance, which suited high-aspect-ratio wings needing consistent performance under dynamic conditions.

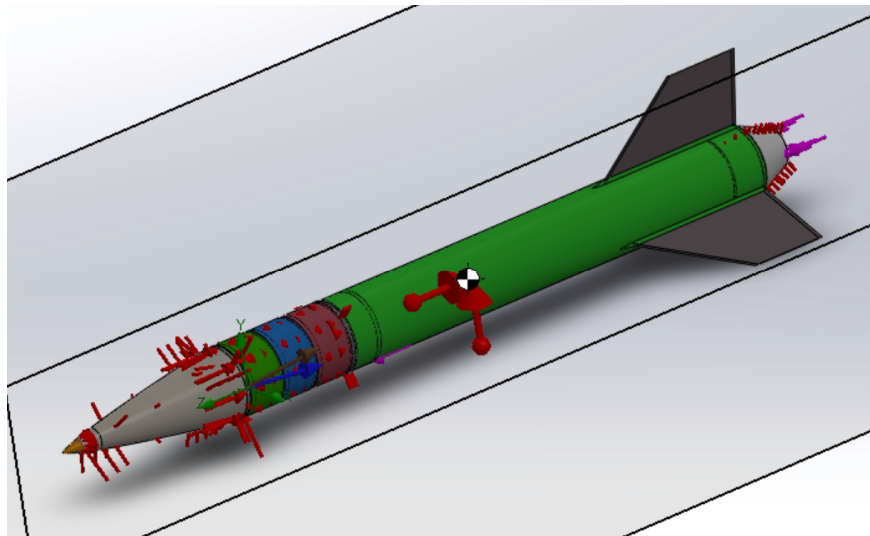
This work emphasizes AMPC's ability to improve safety and resilience in contemporary aircraft, thus supporting adaptive aerospace control. Expanding its degrees of freedom, integrating better fault detection, isolation and recovery mechanisms, and optimizing computational efficiency to support more general aerospace uses could be the main priorities of future developments.

## References

- Afonso, F., Vale, J., Oliveira, J., Lau, F., & Suleman, A. (2017, February). A review on non-linear aeroelasticity of high aspect-ratio wings. *Progress in Aerospace Sciences*, 89, 40–57. Retrieved from <http://dx.doi.org/10.1016/j.paerosci.2016.12.004> doi: 10.1016/j.paerosci.2016.12.004
- Banerjee, J., Liu, X., & Kassem, H. (2014). Aeroelastic stability analysis of high aspect ratio aircraft wings. *Journal of Applied Nonlinear Dynamics*, 3(4), 413–422.
- Garrick, I. (1976, September). Aeroelasticity-frontiers and beyond. *Journal of Aircraft*, 13(9), 641–657. Retrieved from <http://dx.doi.org/10.2514/3.58696> doi: 10.2514/3.58696
- Garrick, I. E., & Reed, W. H. (1981, November). Historical development of aircraft flutter. *Journal of Aircraft*, 18(11), 897–912. Retrieved from <http://dx.doi.org/10.2514/3.57579> doi: 10.2514/3.57579
- Jiang, J. (2005). Fault-tolerant control systems-an introductory overview. *Acta Automatica Sinica*, 31(1), 161–174.
- Köhler, J., Kötting, P., Soloperto, R., Allgöwer, F., & Müller, M. A. (2020, August). A robust adaptive model predictive control framework for nonlinear uncertain systems. *International Journal of Robust and Nonlinear Control*, 31(18), 8725–8749. Retrieved from <http://dx.doi.org/10.1002/rnc.5147> doi: 10.1002/rnc.5147
- Livne, E. (1999, January). Integrated aeroservoelastic optimization: Status and direction. *Journal of Aircraft*, 36(1), 122–145. Retrieved from <http://dx.doi.org/10.2514/2.2419> doi: 10.2514/2.2419
- Livne, E. (2018, January). Aircraft active flutter suppression: State of the art and technology maturation needs. *Journal of Aircraft*, 55(1), 410–452. Retrieved from <http://dx.doi.org/10.2514/1.C034442> doi: 10.2514/1.C034442
- Morin, D. (n.d.). *Chapter 6: Circular motion*. <https://scholar.harvard.edu/files/david-morin/files/cmchap6.pdf>. Retrieved from <https://scholar.harvard.edu/files/david-morin/files/cmchap6.pdf> (PDF file)
- Peter Katz. (n.d.). *Fixing Flutter Is Nothing New*. <https://www.planeandpilotmag.com/article/ntsb-debriefer-fixing-flutter-is-nothing-new/>. (Accessed on: 2023-12-21)
- Qin, S., & Badgwell, T. A. (2003, July). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. Retrieved from [http://dx.doi.org/10.1016/S0967-0661\(02\)00186-7](http://dx.doi.org/10.1016/S0967-0661(02)00186-7) doi: 10.1016/S0967-0661(02)00186-7
- The Grainger College of Engineering, University of Illinois Urbana-Champaign. (n.d.). *Aeroelasticity*. <https://aerospace.illinois.edu/research/research-areas/aeroelasticity>. (Accessed on: 2023-12-20)
- Virgilio Pereira, M. d., Kolmanovsky, I., Cesnik, C. E., & Vetrano, F. (2019, January). Model predictive control architectures for maneuver load alleviation in very flexible aircraft. In *Aiaa scitech 2019 forum*. American Institute of Aeronautics and Astronautics. Retrieved from <http://dx.doi.org/10.2514/6.2019-1591> doi: 10.2514/6.2019-1591



# Design and Analysis of the Structure of a Sounding Rocket



Autor: **ALF/ENGAER Henrique Brás do Carmo Fernandes**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

Orientador: **Prof. Paulo Jorge Soares Gil**  
Instituto Superior Técnico, Lisboa

Coorientador: **Prof. Alain de Souza**  
Instituto Superior Técnico, Lisboa

### Abstract

This work aims to analyse and design the structure of a single-stage sounding rocket using solid propellant, capable of reaching an altitude of 100 km while carrying a 49 kg payload. The structural design of the rocket was conducted, with the creation of a parameterised CAD model. A mass and sizing model was developed, linked to the software SOLIDWORKS®, assessing structural strength and optimising thicknesses. This methodology, which employs Finite Element Analyses (FEA), fulfils the objective of minimising rocket mass. The analyses consider aerodynamic forces and internal loads, such as the combustion chamber pressure, and the generated thrust force. The model is also integrated into a Multidisciplinary Design Optimization (MDO) Software, performing an iterative process to reach the optimised design point, while incorporating trajectory data and engine characteristics. The structural response is evaluated for variations in parameters, such as the rocket's diameter, angle of attack, and flow incidence, while keeping the other parameters constant. The main contributions include the integration of a mass model, capable of performing an automated iterative design process in SOLIDWORKS®, within an MDO Software, to minimise rocket mass while ensuring structural strength, as well as the assessment of the impact of the parameter changes, such as the rocket's diameter, angle of attack, and flow incidence, on the structure.

**Keywords:** Sounding Rocket, Finite Element Method, Iterative Design, Buckling, Structural Design, Optimization.

## 1 Introduction

Rocket structural analysis is a broad field, tied to numerous other areas. The accumulation of works and research in rocket science and structural analysis is driving the evolution of the field, with the surge in knowledge it has undergone over the past decades (Martinez-Val & Perez, 2009). Several entities are investing in the development of this field, strategically identifying areas with potential for growth, such as the advancement of rockets and small satellite launchers (European Investment Bank, J. Toth and A. Concini, 2018). This trend is also emerging in Portugal, reflecting a forward-thinking approach, with projects such as the Santa Maria spaceport and investments in agencies dedicated to space exploration (European Space Agency, 2024; International Trade Administration, n.d.). This trend is expected to continue, developing more advanced technology with the increasing popularity of low Earth orbit rockets (Space Foundation, 2023).

### 1.1 Objectives

The objective of this study is to define and analyse the structure of a small single stage solid fuel sounding rocket capable of reaching an altitude of 100 km while carrying a payload of 49 kg (Edberg & Costa, 2022; Palmeiro, 2017). The rocket has a cylindrical shape with a roughly conical nose, and its structure must withstand lift-off and flight forces, including aerodynamic forces, internal pressure caused by the burning of solid fuel, and the thrust generated.

To achieve these objectives, a parameterised structural analysis will be performed using specialised software, taking into account the various phases of the trajectory and the forces acting on the rocket. The analysis will also consider buckling and natural vibration modes. The goal is to minimise the mass of the rocket, iteratively modifying its dimensions, while ensuring that the structure can withstand various loads. The purpose is also to make an appropriate choice of the materials that will make up the various parts of the rocket structure.

A mass and sizing model was developed, along with an automated simulation procedure using SOLIDWORKS® and python. The CAD models are parameterised, so that they can be integrated into a Multidisciplinary Design Optimisation (MDO) software.

### 1.2 State of the Art

Previous studies on rocket structural design highlight the importance of combining the structural analysis with the material selection, loads tests, and design optimisation. The Finite Element Method (FEM) is the common used tool to evaluate structural response to the environment. This method is being used to study the evaluation of stresses, vibration and buckling, as well as testing the most suitable materials. P. Silva (2021) analysed a sounding rocket using the software NX® SIEMENS testing materials such as aluminium and CFRP, concluding that composites offer good mechanical properties

and low density. S. Ahamed et al. (2014) compared aluminium and composite materials for a Saturn V-like rocket, finding composites to behave in a superior way. S. Chiesa et al. (1999) makes a bridge between a conceptual design with mass estimation and verification with a FEM code, for a vehicle with very specific characteristics, laying the foundations for further studies. The FEM allows to perform thermal and fracture analysis, as D. Kumar et al. (2016) did for the engine casing of a Solid Rocket Motor (SRM), with estimations for dimensions from the literature, proving the materials and estimations reliability. A. Nosseir (2019) focused on the design of propellant tanks for a micro-satellite launcher, and used advanced structural analysis methods, using carbon composites and stainless steel, conducting buckling and modal analyses. J. Yoo and S. Jeon (2020) studied the natural vibration of a Thrust Vector Control actuator, showing that the applied loads changed the natural frequencies.

## 2 Sounding Rocket Structural Design Overview

### 2.1 Requirements and Essential Parameters

Rocket design begins with the definition of mission requirements and constraints. These are based on payload's (PL) target altitude and energy requirements. The necessary  $\Delta v$  for a specific mission is influenced by factors such as gravity losses and atmospheric drag, and is determined alongside with parameters like thrust, specific impulse, and mass ratios, including the structural factor  $\sigma$  and the mass ratio  $\mu = m_0/m_f$ .

### 2.2 Main Structural Components

The main components of a rocket are (Edberg & Costa, 2022): the motor, which generates propulsion forces, and is composed of the casing, the insulation, the propellant, and a pressure vessel; the nozzle, which is responsible for accelerating the gases produced in the chamber; the fins, which are aerodynamic stabilising surfaces; the nosecone, which houses the PL; and the Instrument Unit, which contains crucial systems.

### 2.3 Materials Overview

It is important to select materials which balance the strength, weight and thermal resistance required for each of the rocket component necessities (NASA, 1970; Silva, 2021). The materials selected are: Carbon Fibre-Reinforced Polymer (CFRP), due to its high strength-to-weight ratio and being a leading-edge technology (Edberg & Costa, 2022; Silva, 2021); fibreglass for the nosecone to prevent radio frequency shielding (Peng, Wu, Cherry, & Walton, 2008); maraging steel for high temperature and pressure components (Kumar, Nayana, & Shree, 2016); and aluminium for modules, fins and connectors (Palmeiro, 2017).

### 2.4 Factors of Safety

Factors of Safety (FoS) are coefficients used to multiply the design loads to account for deviations from analysis, manufacturing processes, material properties, among others (ECSS, 2000; NASA, 2014), reducing failure probability. The selected FoS are (ECSS, 2008; NASA, 2014; NASA Technical Standard, 2022): for the internal pressure of the combustion chamber, a FoS of 1.25, while for the composite over-wrapped pressure vessel, a FoS of 1.4; for the nosecone, a FoS of 1.4; for the Payload Attach Fitting (PAF) and the metallic modules, a FoS of 1.25; for the nozzle, a FoS of 1.5; for the fins, a FoS of 1.4.

### 2.5 Structural Loads

Rockets are subjected to a variety of loads during their mission. These loads can be static or dynamic, internal or external, and can also be axial or bending and shear forces (Edberg & Costa, 2022; Nosseir, 2019). Loads influence the stability of the rocket, leading to the concept of the Static Margin (SM), which depends on the rocket Center of Gravity (CG) and Center of Pressure (CP). Loads with interest for the analyses are thrust, drag, chamber pressure and gravity.



## 2.6 Finite Element Theory

The Finite Element Method is selected to study the structural response, due to its versatility in handling various materials, shapes and boundary conditions, as well as having great accuracy and efficiency (Madier, 2020), allied with its usefulness for studying iterative design processes. It is composed of three main steps (Madier, 2020): pre-processing, where the mesh, boundary conditions and loads are defined; solving, which is computing the solution; and post-processing, where results are analysed and validated.

This method is based on Eq. 1, which defines the equilibrium forces in a structure (Madier, 2020).

$$[M]\{\ddot{u}\} + [B]\{\dot{u}\} + [K]\{u\} = P(t), \quad (1)$$

where,  $[M]$  is the mass matrix,  $[B]$  is the damping matrix,  $[K]$  is the stiffness matrix,  $\{u\}$  is the response and  $P(t)$  is the excitation.

The quality and accuracy of the results depend on the mesh quality. A poor quality mesh will lead to completely different results, when compared to the real solution (Madier, 2020). Thus, a convergence study must be performed to determine the optimal mesh that balances accuracy and computational effort.

Boundary conditions must also be set. These are the values of displacements, constraints, or specifications applied to the model, in this case the rocket structure, to define the problem and obtain a unique solution. The analysis utilised (static, buckling and modal) derive from Eq. 1.

## 3 Structural Definition and Analysis Description

In this chapter, the whole structure and its components will be defined and the methodology for the structural analysis will be described. Also, the sizing and convergence process are explained.

### 3.1 Mass Model and Rocket Sizing

The key parameters for the developed mass and sizing model are the mass ratio  $\mu$ , the structural factor  $\sigma$ , the structural mass  $m_s$  and the propellant mass  $m_p$ , related by (Edberg & Costa, 2022)

$$\mu = e^{\frac{\Delta v}{g_0 I_{sp}}}, \quad (2)$$

and

$$\sigma = \frac{m_s}{m_s + m_p}, \quad (3)$$

where,  $g_0$  is the gravitational constant and  $I_{sp}$  is the specific impulse, and is equal to 260 s. The initial value for the structural factor is given by the literature (Edberg & Costa, 2022). These equations are the foundation for further sizing and design.

The created mass model uses MERs, comparisons with other rockets, and fundamental equations that relate essential parameters and provide dimensions for the components.

The equations of the mass model are connected to the SOLIDWORKS® model through parameterisation. Their output change reflects a change in the design of the model in terms of dimensions, ratios, or masses. To achieve this linking between the equations of the mass model and the geometry of the CAD model, a relationship between dimensions, masses, volumes, material properties and geometry is necessary. The developed model ultimately calculates CAD dimensions, inserting them into a text file that is read by SOLIDWORKS®, and the latter updates the overall model to meet the dimensions calculated.

The model receives motor characteristics and a trajectory from Palaio (2024), after an optimisation process for a certain rocket aiming to reach the altitude of 100 km. The essential equations that guide this model are presented below.

**Pressure Vessel and Bodytube:** It is possible to calculate the shell thickness for a cylindrical shape using the equation (ASME, 2023; Kumar et al., 2016):

$$t = \frac{P \cdot D \cdot \text{mismatch factor}}{\text{biaxial gain} \cdot 2 \cdot (SE - 0.6P)}, \quad (4)$$

where,  $P$  is the maximum operating pressure,  $D$  is the diameter, the *mismatch factor* is a factor that accounts for differences in thickness or materials, the *biaxial gain* is a factor that accounts for forces acting in various directions,  $S$  is the strength of the materials used and  $E$  is the weld efficiency.

**Insulation:** For insulation, an estimation can be made based on volumes, as (Edberg & Costa, 2022).

$$V_{\text{insulation}} = 0.02 \cdot V_{\text{propellant}}, \quad (5)$$

where, the propellant volume  $V_{\text{propellant}}$  is calculated using the initial estimate of its mass and propellant density.

**Nozzle:** The nozzle sizing starts with the throat area calculation. The expression for this is (Edberg & Costa, 2022).

$$A_{\text{throat}} = c_{\text{del}} \cdot \frac{T_{\text{vac avg}}}{P_{\text{c avg}} \cdot g_0 \cdot I_{\text{sp}}}, \quad (6)$$

where,  $c_{\text{del}}$  is the delivered specific exhaust velocity,  $T_{\text{vac avg}}$  is the average thrust in vacuum,  $P_{\text{c avg}}$  is the average pressure in vacuum,  $g_0$  is the acceleration of gravity at the surface of Earth and  $I_{\text{sp}}$  is the specific impulse.

The nozzle exit radius is given by (Edberg & Costa, 2022).

$$R_{\text{nozzle exit}} = \sqrt{\epsilon} \times R_{\text{throat}}, \quad (7)$$

where,  $\epsilon$  is the nozzle expansion ratio.

Then, the nozzle length  $L_{\text{nozzle}}$  is obtained using:

$$L_{\text{nozzle}} = \frac{(2R_{\text{nozzle exit}} - 2R_{\text{throat}}) \times 0.8}{2 \tan(15^\circ)}. \quad (8)$$

For the nozzle divergent section thickness  $t_{\text{nozzle}}$ , the equation is (ASME, 2023; Kumar et al., 2016):

$$t = \frac{P \cdot d}{2 \cos \alpha \left( \frac{UTS}{F.S} \right) \eta_{\text{weld}} - 0.6P}, \quad (9)$$

where,  $P$  is the pressure at the throat,  $d$  is the exit diameter,  $\alpha$  is the convergent angle,  $UTS$  is the ultimate tensile strength of the nozzle material and  $\eta_{\text{weld}}$  is the weld efficiency.

**Nosecone:** The thickness and mass of the nosecone are estimated using a MER for composite payload fairings. Since it is made of fibreglass, an *areal density* = 9.89 kg/m<sup>2</sup> (Edberg & Costa, 2022). Hence, the nosecone thickness is obtained dividing the *areal density* by the density of the material, as

$$t_{\text{nosecone}} = \text{areal density} / \rho_{\text{nosecone}}. \quad (10)$$

The nosecone volume is obtained by multiplying this estimated thickness by the nosecone surface area  $A_{\text{surface nosecone}}$ . Its mass is obtained by multiplying the nosecone volume by its density.

**Modules:** The length of the modules of the rocket is based on the length of the modules of the REXUS rocket (Lesch & Völk, 2017), as well as its initial thickness, that is  $t_{\text{module}} = 4$  mm. With this thickness and their geometry, a mass for each of them can be calculated. Then, the masses of the internal components are added.

**Payload Attach Fitting, Payload and Connector:** The mass of components that are related to the PL arrangement in the nosecone can be estimated by the MER (Morgado, 2019):

$$m_{\text{PAF}} = 0.0477536 \cdot (m_{\text{PL}})^{1.01317}. \quad (11)$$

**Fins:** The mass of the fins is determined using their design and an initial thickness estimate of  $t_{\text{fin}} = 5$  mm, which aligns with typical values. Other relations are used to define them:

$$\text{semispan}_{\text{fin}} = 1.1 \cdot (2 \cdot R_{\text{rocket}}), \quad (12)$$

$$c_{\text{root}} = 2.5 \cdot (2 \cdot R_{\text{rocket}}), \quad (13)$$

$$c_{\text{tip}} = 0.5 \cdot (c_{\text{root}}), \quad (14)$$

where,  $\text{semispan}_{\text{fin}}$  is the length of the fins semispan,  $c_{\text{root}}$  is the fins root chord and  $c_{\text{tip}}$  is the fins tip chord. Thus, with the sharp edges design taken into account and the material choice, fin mass can be estimated by calculating its volume.

**Boattail:** The boattail has the geometry of a truncated cone, whose thickness  $t_{\text{boattail}}$  is defined as  $t_{\text{boattail}} = 1.5 \text{ mm}$ .

**Wiring:** The wiring mass can be estimated as (Edberg & Costa, 2022)

$$m_{\text{wiring}} = 1.43 \cdot L_{\text{rocket}}. \quad (15)$$

### 3.2 Methodology for the Iterative Process

The process starts with an educated guess on the  $\Delta v$  for the trajectory, which is imported from an MDO Software developed by Palaio (2024), and an initial structural factor  $\sigma$ . Both of these parameters are intended to converge to a final value, when the thicknesses stabilize and cannot be optimised more. Not only a  $\Delta v$  is obtained from the MDO Software but also other necessary inputs for the analyses, such as the rocket velocity or the characteristics of the atmosphere.

This methodology consists of the following steps:

- Estimating a preliminary rocket diameter;
- Estimating an initial  $\Delta v$  for the trajectory;
- Using a system of equations to relate and estimate propellant mass  $m_p$ , structural mass  $m_s$ ,  $\Delta v$  and structural factor  $\sigma$ ;
- Using the mass model, linked to the CAD model, to calculate rocket dimensions;
- If the structural factor  $\sigma$  and the structural mass  $m_s$  converge, the process continues; otherwise, the last two items are repeated with the current values;
- Calculating a new trajectory based on the current values for the parameters, obtaining a new  $\Delta v$  required;
- If this new  $\Delta v$  has converged when compared to its previous value, the process continues; otherwise, values return to the mass model to converge the structural factor  $\sigma$  and the remaining parameters again, continuing then from this step;
- Performing Flow and Structural analyses in SOLIDWORKS®;
- Adjusting thicknesses or dimensions based on the stresses on each component;
- If the thicknesses converge, the process stops; otherwise, the new dimensions are input in the mass model to redesign the rocket and converge all the parameters, starting the process again.

The developed tool, when the mass model parameters convergence is achieved, interacts with the software interface, inserting all the input variables that are necessary, meshes, and runs the analysis. Then, after the maximum stresses input is given, the developed tool automatically makes an adjustment to the dimensions based on the maximum stress allowed. This adjustment leads to a new rocket, with new dimensions and masses, restarting the process, inputting all the new values into the mass model to achieve a newer convergence.

### 3.3 Design Definition

The models were essentially based on studies available in the literature or typical designs. The components of the rocket CAD model are: the Nosecone Tip; the Nosecone; the Payload Attach Fitting; the Payload (as a 2:1 cylinder); the Modules for GNC, experimental devices, and recovery; a connector between the Modules and the Payload Attach Fitting; the Hatches for accessing the interior of the Modules; the Insulation; the Propellant cylinder; a Pressure Vessel to contain the Propellant and the Insulation, as well as to resist burning pressure; the Bodytube, which is the over-wrapped composite in the Pressure Vessel; the Fins; the Nozzle; and the Boattail.

Figure 1 presents a section view of the developed rocket CAD model, showing its components.

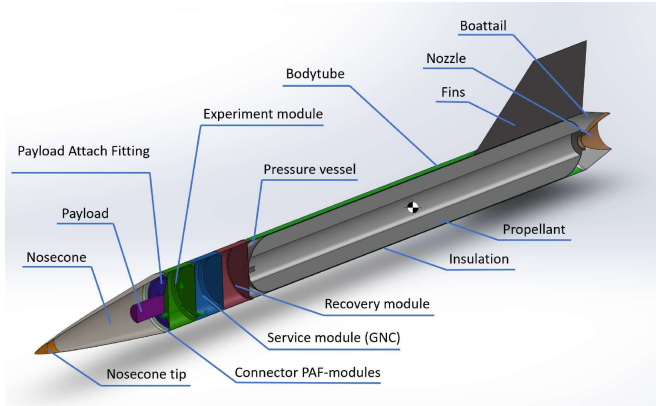


Figure 1: Rocket components section view.

### 3.4 Material Properties

The nosecone tip, the connecting parts, the PAF, the fins and the modules are made of aluminium; the nosecone is made of fibreglass; the bodytube is made of CFRP; the pressure vessel and the nozzle are made of maraging steel; the propellant is HTPB/AP/AI; and the insulation is EPDM.

Table 1 summarises the properties of the selected materials. It contains the material, its density, Young's modulus and strength.

Table 1: Properties of the selected materials.

| Material        | Density<br>(kg/m <sup>3</sup> ) | Young's<br>Modulus (GPa) | Strength<br>(MPa) |
|-----------------|---------------------------------|--------------------------|-------------------|
| CFRP            | 1480                            | 91.8                     | 829               |
| Fibreglass      | 2490                            | 89.5                     | 4500              |
| Maraging Steel  | 8000                            | 186                      | 1710              |
| Aluminium Alloy | 2800                            | 71.7                     | 503               |
| EPDM            | 962                             | 0.064                    | 4.88              |
| HTPB/AP/AI      | 1758                            | 0.0075                   | 1.03              |

### 3.5 Material Failure Criteria

Some of the selected materials are ductile, undergoing significant deformation before failure, and typically fail due to yielding; while others are brittle, where failure occurs with little or no plastic deformation (Christensen, 2013). For ductile

materials, the applied stress is compared to the yield strength, while for brittle materials, the applied stress is compared to the ultimate strength. The used failure criterias are the Von Mises criterion and the Maximum Normal Stress criterion (Christensen, 2013).

The Von Mises criterion should be used for ductile materials and isotropic materials (Christensen, 2013). It is defined as (Christensen, 2013; Palmeiro, 2017).

$$\frac{1}{2} [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2] \leq T^2, \quad (16)$$

However, since the developed model includes ductile and brittle materials, as well as isotropic and orthotropic materials, the Von Mises stress criterion is not sufficient to evaluate all the stresses experienced by the components.

Hence, another failure criterion is used, the Maximum Normal Stress criterion. According to this theory, failure occurs when the principal stress of the brittle material reaches its ultimate strength (Dassault Systèmes, 2021), being defined as

$$\sigma_1 \geq \sigma_{\text{limit}}, \quad (17)$$

where,  $\sigma_1$  is the maximum principal stress and  $\sigma_{\text{limit}}$  is the maximum allowed stress.

### 3.6 Flow Simulation

Aerodynamic forces are usually present in the conducted studies. To acquire them, it was chosen to use the software SOLIDWORKS® FLOW SIMULATION. It is a Computational Fluid Dynamics (CFD) solution embedded within CAD modelling, enabling the simulation of fluids through and around designs to assess models (SOLIDWORKS, 2024).

### 3.7 Analysis Definition

A static analysis was performed with the forces acting on the rocket's external surface, the thrust generated by the engine, and the internal pressure in the combustion chamber. This analysis makes it possible to verify the distribution of stresses on the structure due to these loads, and evaluate if they remain within the safety factors. A buckling analysis was performed, to obtain the buckling modes, and a modal analysis, in which the vibration modes of the whole structure were obtained. These analyses include performing a *flow simulation*, defining the *loads* and the *distributed masses* in the software, creating the *materials*, set the *connections* and, for static analyses, activate the *inertial relief*. The results of all the analyses give stresses, buckling factors of safety (BFS) and mode shapes.

## 4 Results

### 4.1 Model Design

Since rocket design itself is also an iterative process, the same occurred here, with the design not only being enhanced by optimisation processes but also undergoing significant modifications. The Figure 2 presents the final rocket design.

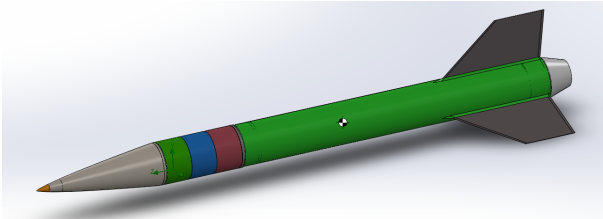


Figure 2: Rocket assembly.

## 4.2 Finite Element Analysis setup

The analyses start with a *flow simulation* to obtain pressure along the rocket surface. The process of converging the results of the *flow simulation* was to iteratively run this type of simulation, registering and tracking the results of some parameters, such as the static pressure, dynamic pressure, total pressure, friction force, and normal force.

Then, the convergence of the FEM mesh was performed. It is also an iterative process, where it is tracked the evolution of the interest parameter, in this case the maximum stress in each component, along with parameters that are related to the mesh quality, such as the aspect ratio, the element size, the number of elements, the number of nodes, the number of Degrees of Freedom (DoF), the Average Percentage Error (APE) and the simulation time (Hawk Ridge Systems, 2024; Madier, 2020). The final mesh had most of its elements between 2 mm and 4 mm, with around  $1 \times 10^7$  elements,  $5 \times 10^6$  nodes, and  $1.5 \times 10^7$  DoFs. Figure 3 presents the evolution of the number of mesh elements along the convergence process.

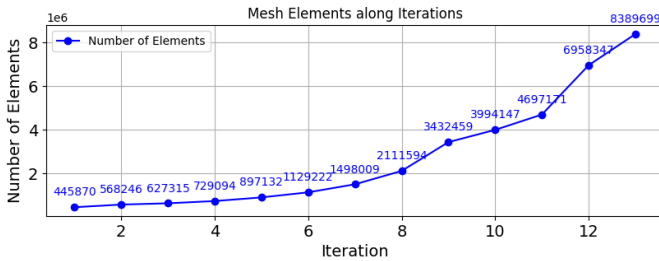


Figure 3: Number of mesh elements along the iterative process.

## 4.3 Static Analysis

For the analyses, it was defined that only theoretical critical points of the trajectory would be studied (Edberg & Costa, 2022; Newlands, Heywood, & Lee, 2016). The points of interest are: maximum dynamic pressure, maximum air speed, maximum chamber pressure, and maximum thrust. These points are the same, as a result of the thrust profile.

### Verification of Model Simplification Accuracy

As the complexity of the developed model leads to large simulation times, two simplifications were performed: the removal of the insulation and propellant CAD models, and substitution of *contact* by *bonded* interactions. Tests were performed to validate these assumptions, whose impact was negligible, thus being able to proceed with the analyses.

### Mass Model Integration within an MDO Software

The integration of the mass model into an inner loop of an MDO software leads to the possibility of iterating the design of the rocket within the mass model, while assessing energy needs in a trajectory. Also, with a trajectory, it is possible to acquire the critical points where loads are significant the most. It was observed that the intended parameters were converging along the process: the  $\Delta v$  and the structural factor  $\sigma$ . However, the analyses process required some time, considering analyses and data exchange.

Figure 4 shows the evolution of the various masses of the rocket. A reduction of 18.89 kg was achieved in relation to the structural mass, which corresponds to an overall reduction in the structural mass of 16.76 %, and a reduction of 10.67 kg was achieved in relation to the propellant mass, a reduction of 5.60 %. This means an overall reduction of the initial mass of 29.55 kg, corresponding to a reduction of 8.36 % of the rocket's overall initial mass. The components that reduced their thickness, when compared to its initial value are: the nosecone, the PAF, the bodytube, the nozzle, and the fins. The components that maintain their thickness are: the modules, and the cylindrical part of the vessel. The domes of the vessel increased their thickness.

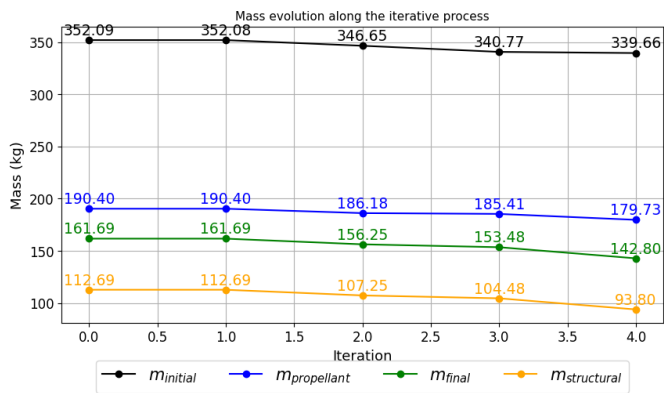


Figure 4: Mass evolution along the iterative process for the initial mass, the propellant mass, the final mass and the structural mass.

Angle of Attack Impact on Component Maximum Stress

This study aimed to evaluate the impact of the AoA in the component maximum stress. An AoA of  $0^\circ$  was set as the reference value. Simulations were performed varying the angle between  $-3^\circ$  and  $9^\circ$  in increments of  $1.5^\circ$ . It can be observed that the percentage impact decreases along the rocket axis, from the tip to the nozzle. The upper body of the rocket was more affected by the AoA than the lower part. The components most affected, in percentage deviation, comparing the maximum stress at an AoA of  $9^\circ$  with the reference, being the nosecone tip (33.68%), the nosecone (15.77%), the PAF (10.40%), the connector of the PAF to the modules (9.29%). For the remaining components, the percentage deviation is less than 1%. Figure 5 shows the evolution of the maximum stress on the nosecone with the increase of the AoA.

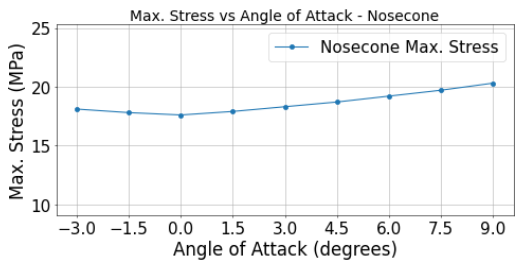


Figure 5: Maximum Stress evolution with the increase in the AoA, for angles between  $-3^\circ$  and  $9^\circ$ , in the Nosecone.

Fin Angle in Relation to the Airflow Impact on Component Maximum Stress

This study evaluated the impact of the fins' angle relative to the airflow on component stress, rotating them around the longitudinal axis of the rocket.

Simulations were performed, considering that the flow came from the front and the top, for the angles shown in Fig. 6. From left to right, the first will be called the  $0^\circ$  angle, the second  $30^\circ$  angle, and the third  $60^\circ$ , with the AoA of  $0^\circ$  as the reference.

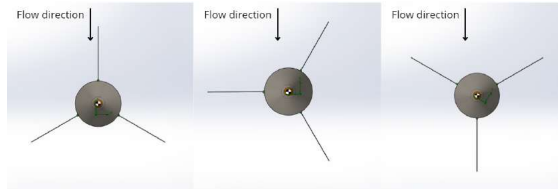


Figure 6: Fin angles in relation to the airflow.

The maximum deviation on the maximum stress happens for the  $60^\circ$  and is  $-2.1\%$ . This value is considered low and not significant.

### Rocket Radius Impact on Component Maximum Stress

This study evaluated the impact of rocket radius variation on component stress. It was observed that changing the rocket radius may decrease or increase the maximum stress, depending on the component. The rocket radius has a significant impact on the maximum stress of the components, when compared to the previous analyses. Rocket radius ranged from 0.100 m to 0.250 m, in increments of 0.025 m. The component with the smallest percentage change in maximum stress is the nozzle, with a reduction of  $-10.3\%$  ( $-62.0\text{ MPa}$ ) when comparing the larger with the smaller radius, whereas the components most affected are the modules, with a variation of  $355.9\%$  ( $402.9\text{ MPa}$ ) when comparing the larger with the smaller radius. The pressure vessel is the most affected component in absolute terms ( $1041.4\text{ MPa}$ ), as the radius increases for the same chamber pressure. Figure 7 shows the impact of rocket radius variation on maximum stress of the pressure vessel.

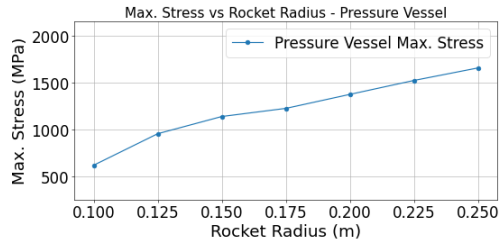


Figure 7: Maximum Stress evolution with the increase in the rocket radius, for radius between 0.100 m and 0.250 m, in the Pressure Vessel.

## 4.4 Buckling Analysis

The Buckling Analysis results are the BFS and the displacement amplitudes for the modes. The first two buckling modes were obtained for before lift-off, during lift-off and flight.

Table 2: Buckling Factors of Safety. Adapted: (SOLIDWORKS, 2021).

| BFS                     | Buckling Status        |
|-------------------------|------------------------|
| $\text{BFS} < 0$        | Buckling not predicted |
| $0 < \text{BFS} \leq 1$ | Buckling predicted     |
| $1 < \text{BFS}$        | Buckling not predicted |

**Before lift-off case:** There are no aerodynamic loads, chamber pressure, or thrust. The structure is fixed as if it were on a ground support. The BFS obtained were  $-20.04$  and  $-10.68$ , leading to maximum displacements of  $15.5\text{ mm}$  and  $23.7\text{ mm}$ .



From the data of Tab. 2, it can be inferred that buckling is *not predicted* to occur. By analysing the deformed shape of the model, it can be concluded that these displacements occur in the module closer to the PL. The PAF supports the PL mass, being joined to the connector and the experiment module, thus causing loads in that region. Although buckling is not likely to occur, an increase in the thickness of the ends of the modules, may be considered to better handle a buckling situation.

**During lift-off case:** There are no aerodynamic loads, but there is chamber and nozzle pressures, and thrust. The BFS obtained were  $-3.797 \times 10^{-5}$  and  $-1.197 \times 10^{-6}$ , leading to maximum displacements of 3.0 mm and 0.6 mm. From the data of Tab. 2, it can be inferred that buckling is *not predicted* to occur. Regarding the displacements, they are not significant. If the structure buckles, it tends to deviate from its original direction, leading to the need for trajectory corrections. Figure 8 shows the buckling mode 2 for during lift-off case.

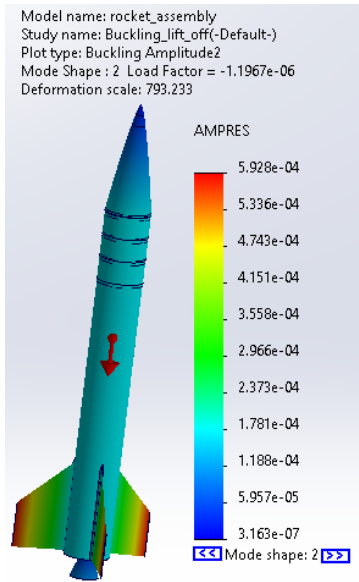


Figure 8: Buckling mode 2 for during lift-off case.

**Flight case:** There are aerodynamic loads, chamber and nozzle pressures, and thrust. The BFS obtained were  $-1.708 \times 10^{-5}$  and  $-5.376 \times 10^{-7}$ , leading to maximum displacements of 2.0 mm and 0.4 mm. From the data of Tab. 2, it can be inferred that buckling is *not predicted*. Regarding the displacements, they are not significant. If the structure buckles, it tends to deviate from its original direction, leading to the need for trajectory corrections.

4.5 Modal Analysis

The Modal Analysis outputs are the Mode Shapes and the displacement amplitudes. Load conditions are similar to the ones in the Buckling studies, with the inclusion of the scenario in which the rocket is in a free-free condition. The first twelve modes were obtained for each case. Local modes that affect the fins, contributing little to the structural response (Chen, Abbas, Xiaoting, & Guoping, 2017).

**Free-free:** The rocket is in a free-free condition, without loads applied. The first six modes are rigid-body motion modes. The remaining modes are bending, torsional and local. Figure 9 and Figure 10 show the Mode Shapes 10 and 11 for the free-free condition.

Table 3: Relevant vibration Modes Frequency and Max. Displacement for free-free case.

| Mode | Frequency (Hz) | Max. Displac. (m) |
|------|----------------|-------------------|
| 7    | 21.049         | 1.355             |
| 8    | 21.069         | 1.261             |
| 9    | 21.961         | 0.970             |
| 10   | 40.510         | 0.399             |
| 11   | 40.535         | 0.461             |
| 12   | 43.044         | 1.903             |

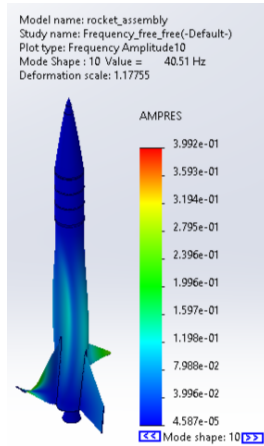


Figure 9: Mode shape 10 for free-free.

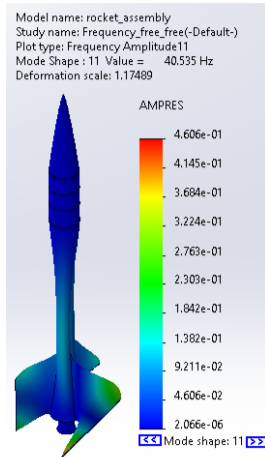


Figure 10: Mode shape 11 for free-free.

**Before lift-off case:** The rocket's base is fixed as if the rocket is on a ground support (*clamped*), thus leading to the absence of rigid-body motion modes. The modes are bending, torsional and local.

Table 4: Relevant vibration Modes Frequency and Max. Displacement for before lift-off case.

| Mode | Frequency (Hz) | Max. Displac. (m) |
|------|----------------|-------------------|
| 1    | 10.968         | 0.1401            |
| 2    | 10.970         | 1.010             |
| 9    | 46.391         | 1.897             |
| 10   | 47.009         | 0.383             |
| 11   | 47.016         | 0.349             |
| 12   | 53.408         | 0.170             |

**During lift-off case:** The rocket is in a free-free condition, but loads are applied. The first three modes are rigid-body motion modes. The remaining modes are bending, torsional and local.

Table 5: Relevant vibration Modes Frequency and Max. Displacement for during lift-off case.

| Mode | Frequency (Hz) | Max. Displac. (m) |
|------|----------------|-------------------|
| 4    | 7.682          | 0.136             |
| 5    | 7.687          | 0.137             |
| 12   | 57.066         | 0.197             |

**Flight case:** The rocket is in a free-free condition, but loads are applied. The first three modes are rigid-body motion modes. The remaining modes are bending and local.

Table 6: Relevant vibration Modes Frequency and Max. Displacement for flight case.

| Mode | Frequency (Hz) | Max. Displac. (m) |
|------|----------------|-------------------|
| 8    | 38.735         | 1.655             |
| 9    | 39.402         | 1.899             |
| 10   | 39.516         | 1.377             |
| 11   | 39.516         | 1.377             |
| 12   | 68.190         | 2.423             |

5 Conclusions

The structure of a sounding rocket was designed, optimised, and tested under various conditions.

In terms of the design itself, various components were redesigned and discarded due to poor performance, such as the fins and the motor.

Automating analyses in SOLIDWORKS® was challenging due to limitations of the API, resulting in a code developed to interact with the software interface itself. However, automation was required since the goal was to integrate the model into an MDO Software.

The developed mass model linked to the parameterised SOLIDWORKS® CAD model, allowed for easy iteration and dimension change. The iterative process performed well by converging the structural factor  $\sigma$  from an obtained energy requirement  $\Delta v$ . Both parameters converged within the mass model, along with the structural mass  $m_s$  and propellant mass  $m_p$ .

Regarding the MDO integration, the objective of minimising the rocket mass was achieved. However, as the thicknesses could have decreased further after the finish of the four iterations, some additional iterations would be necessary to reach thickness convergence and set a minimum thickness concerning manufacturing feasibility of the materials.

Regarding the other studies performed, including the AoA impact, the radius change impact, and flow incidence impact, these were evaluated for the final calculated trajectory in the MDO integration. The study that showed a greater impact was the change in rocket radius. It affected the maximum stress in every component, with emphasis (in absolute terms) on the pressure vessel, since the chamber pressure was kept constant simultaneously with all the other variables. The chamber is the component that experiences the highest stresses, requiring the most attention in terms of strength. The chamber pressure is more critical to structural stress than aerodynamic forces, which are more relevant for energy requirements.

In terms of Finite Element Analysis, it was verified that studying the entire rocket structure requires significant computational power, leading to the implementation of some simplifications to free machine resources while maintaining an adequate mesh refinement.

The rocket performed well under buckling conditions, being acquired the buckling factors of safety and deformation.

The vibration modes were also obtained, allowing to avoid certain undesirable phenomena to be avoided, such as resonance, as well as dynamic loads related to the vibration condition of the rocket.

References

ASME. (2023). *Asme boiler and pressure vessel code: Rules for construction of pressure vessels, division 1, section viii*. American Society of Mechanical Engineers. Retrieved from <https://www.asme.org>

Chen, D., Abbas, L. K., Xiaoting, R., & Guoping, W. (2017). Aerodynamic and static aeroelastic computations of a slender rocket with all-movable canard surface. *Journal of Aerospace Engineering*, 0, 1–17. doi: 10.1177/0954410017705910

Christensen, R. M. (2013). *The theory of materials failure*. Oxford University Press.

- Dassault Systèmes. (2021). Maximum normal stress criterion - 2021 - solidworks help [Computer software manual]. Retrieved from [https://help.solidworks.com/2021/english/SolidWorks/cworks/r\\_Maximum\\_Normal\\_Stress\\_Criterion.htm](https://help.solidworks.com/2021/english/SolidWorks/cworks/r_Maximum_Normal_Stress_Criterion.htm) (Accessed: 2024-10-10)
- ECSS. (2000, apr). *Space engineering - mechanical - part 2: Structural* (Tech. Rep. No. ECSS-E-30 Part 2A). Author.
- ECSS. (2008, November). *Space engineering: Structural design and verification of pressurized hardware* (Tech. Rep. No. ECSS-E-ST-32-02C Rev. 1). ECSS Secretariat, ESA-ESTEC, Requirements & Standards Division.
- Edberg, D., & Costa, W. (2022). *Design of rockets and space launch vehicles* (2nd ed.). American Institute of Aeronautics and Astronautics, Inc.
- European Investment Bank, J. Toth and A. Concini. (2018). *The future of the european space sector: How to leverage europe's technological leadership and boost investments for space ventures* [Executive Summary]. Luxembourg.
- European Space Agency. (2024). *Santa maria station*. Retrieved from [https://www.esa.int/Enabling\\_Support/Operations/ESA\\_Ground\\_Stations/Santa\\_Maria\\_station](https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Santa_Maria_station) (Accessed: 2023-11-21)
- Hawk Ridge Systems. (2024). *Curvature based mesh advantages*. Retrieved from <https://hawkridgesys.com/blog/curvature-based-mesh-advantages> (Accessed: 2024-10-20)
- International Trade Administration. (n.d.). *Portugal - aerospace and defense*. Retrieved from <https://www.trade.gov/country-commercial-guides/portugal-aerospace-and-defense> (Accessed: 2024-01-25)
- Kumar, B., Nayana, B., & Shree, D. (2016, 01). Design and structural analysis of solid rocket motor casing hardware used in aerospace applications. *Journal of Aeronautics & Aerospace Engineering*, 5. doi: 10.4172/2168-9792.1000166
- Lesch, K., & Völk, S. (2017, November). *Rexus user manual* (Tech. Rep.). EuroLaunch. Retrieved from [https://rexusbexus.net/wp-content/uploads/2021/12/RX\\_UserManual.v7-17\\_26Nov21.pdf](https://rexusbexus.net/wp-content/uploads/2021/12/RX_UserManual.v7-17_26Nov21.pdf)
- Madier, D. (2020). *Practical finite element analysis for mechanical engineers* (1st ed.). FEA Academy.
- Martinez-Val, R., & Perez, E. (2009). Aeronautics and astronautics: Recent progress and future trends. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(12), 2767–2820. doi: 10.1243/09544062jmes1546
- Morgado, F. M. P. (2019). *Coupled preliminary design and trajectory optimization of rockets using a multidisciplinary approach* (Unpublished master's thesis). Instituto Superior Técnico, Universidade de Lisboa.
- NASA. (1970). *Solid rocket motor metal cases* (No. NASA SP-8025). National Aeronautics and Space Administration.
- NASA. (2014, July). *Structural design requirements and factors of safety for spaceflight hardware* (Tech. Rep. Nos. JSC 65828 Rev. B, change 1). Lyndon B. Johnson Space Center.
- NASA Technical Standard. (2022, October). *Structural design and test factors of safety for spaceflight hardware* (Tech. Rep. No. NASA-STD-5001B). Office of the NASA Chief Engineer. Retrieved from <https://ntrs.nasa.gov/>
- Newlands, R., Heywood, M., & Lee, A. (2016, June). *Rocket vehicle loads and airframe design* (Tech. Rep.). Aspirespace.
- Nosseir, A. (2019). *Rocket design and structural modeling and analysis of micro-satellite launch vehicle: Inspired by vector-r* (Unpublished master's thesis). Università di Pisa.
- Palmeiro, A. F. (2017). *Introdução à tecnologia de foguetes* (2nd ed.). Author.
- Peng, B. J., Wu, Y., Cherry, S. R., & Walton, J. H. (2008). Effective rf shielding with carbon fiber composites for simultaneous pet / mri. *Proceedings of the International Society for Magnetic Resonance in Medicine*(17). Retrieved from <https://api.semanticscholar.org/CorpusID:35277465>
- Silva, P. (2021). *Concept design of a sounding rocket main structure using high performance materials* (Unpublished master's thesis). Instituto Superior Técnico.
- SOLIDWORKS. (2021). *Buckling load factor*. Retrieved from [https://help.solidworks.com/2021/english/SolidWorks/cworks/c\\_Buckling\\_Load\\_Factor.htm](https://help.solidworks.com/2021/english/SolidWorks/cworks/c_Buckling_Load_Factor.htm) (Accessed: 2024-10-20)
- SOLIDWORKS. (2024). *SOLIDWORKS Flow Simulation*. Retrieved from <https://www.solidworks.com/product/solidworks-flow-simulation> (Accessed: 2024-10-10)
- Space Foundation. (2023). *The space report 2023 q2* (Tech. Rep.). Space Foundation. Retrieved from <https://www.spacefoundation.org/2023/07/25/the-space-report-2023-q2/> (Accessed: 30-09-2023)



# Study of the trajectory of the stages of a rocket launched from the Santa Maria spaceport



Autor: **ALF/ENGAER José Miguel Alves Gonçalves Mota**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

Orientador: **Professor Paulo Gil**  
Instituto Superior Técnico, Lisboa

Coorientador: **CAP/ENGAER Marta Santos**  
Academia da Força Aérea, Sintra

## Abstract

This work explores the optimal trajectory for a rocket launched from the Santa Maria spaceport and the potential impact zones of the falling stages. Calculating the optimal trajectory involves several steps, starting with vertical flight. The time of this phase is used as an optimisation variable. After reaching a safe altitude, the rocket is inclined to begin a gravity turn, reducing velocity losses caused by gravity. This manoeuvre ends when the atmospheric effects are no longer significant, allowing the rocket to shed its fairing and control its angle of attack. The control is guided by an indirect optimisation method based on Pontryagin's Minimum Principle. This method transforms the problem into a "two-point boundary value problem" where initial conditions are adjusted to achieve the desired orbit. Particle Swarm Optimisation, inspired by birds' behaviour, iterates the system until an optimal solution is reached. Once the optimal trajectory is calculated, the rocket's stages are simulated to determine their fall points, with and without drag, comparing them with analytical models. If the stages risk falling into inhabited areas, the trajectory is re-optimised to reduce the horizontal distance travelled. The performance impact of these changes is assessed by evaluating the fuel left in the final stage, which could otherwise increase payload capacity. This methodology concluded that Santa Maria provides the initial conditions necessary for launching rockets of varying sizes and for different types of mission No paragraph breaks.

**Keywords:** Impact points, Optimisation, Pontryagin's Minimum Principle, Santa Maria Spaceport, Trajectory

## 1 Introduction

The main objective of this work is to determine the performance cost of decreasing the downrange of the first stages in terms of payload mass to Low Earth Orbit (LEO). This has application to the Santa Maria spaceport when the stages downrange reach Europe, Africa, or high-intensity maritime traffic areas. For this purpose, small and medium-sized rockets will be used as case studies, using reasonable assumptions to approximate the rocket's real trajectory. Different Santa Maria launch scenarios will be simulated.

At the beginning of the flight, the atmosphere affects the trajectory, so instead of relying on its control, a less computationally heavy problem is used: the gravity turn manoeuvre (Sotto & Teofilatto, 2002). When the rocket leaves the atmosphere, it begins optimising the trajectory. The calculus of the optimal trajectory is usually divided into two main types: direct and indirect optimisation, each with its advantages and disadvantages (Betts, 1998).

A direct method does not require an analytical expression for the necessary conditions and usually does not require initial estimates for the additional variables. Instead, the variables are adjusted to optimise the objective function directly (Betts, 1998). The main disadvantage of this method is the lack of accuracy in the optimisation.

The explicit solution of the optimal and associated boundary conditions characterises indirect methods. It is necessary to derive analytical expressions for the conditions and non-linear dynamics (Trélat, 2012). This work used an indirect method due to the high accuracy, in this case, the PMP. The problem is converted to a two-point boundary value problem (TPBVP) and resolved with a shooting method coupled with a PSO, a metaheuristic method based on natural phenomena behaviour.

## 2 Rocket trajectories

### 2.1 Rocket Dynamics

The trajectory is divided into 3 phases: Vertical, gravity turn, and free flight. The vehicle starts vertically until a kick angle is given, initiating the gravity turn. When the rocket enters exo-atmospheric conditions, the free flight phase and trajectory optimisation begin with the PSO algorithm employed to explore the solution space and find the optimal trajectory (Pontani & Teofilatto, 2014).

The system is represented by a state vector that describes the conditions of the launcher at every instant. The state vector for the vertical phase and gravity turn phase is (Pontani & Teofilatto, 2014)

$$y = [x \quad h \quad V \quad \gamma]^T, \quad (1)$$

where  $x$  is the horizontal distance,  $h$  is the altitude,  $V$  is the horizontal velocity,  $\gamma$  is the flight path angle, and it is the angle between the local horizon and the velocity axis. It is assumed that Earth is a regular spherical body without rotation and the launcher moves only in this orbit plane, and the equations of motion are 2D.

The equations of motion are (Sforza, 2016):

$$\begin{aligned}\dot{x} &= \frac{R_e}{R_e + h} V \cos(\gamma), \\ \dot{h} &= V \sin(\gamma), \\ \dot{V} &= \frac{T}{m} \cos(\alpha) - g \sin(\gamma) - \frac{D}{m}, \\ \dot{\gamma} &= \frac{\frac{T}{m} \sin(\alpha) - \left(g - \frac{V^2}{R_e + h}\right) \cos(\gamma)}{V},\end{aligned}\tag{2}$$

where  $R_e$  is the radius of the Earth,  $m$  is the mass,  $T$  is the thrust,  $D$  is the drag,  $I_{sp}$  is the specific impulse,  $g$  is the gravity of the Earth at sea level,  $\alpha$  is the angle of attack of the rocket.

The initial vertical take-off phase of rockets involves the launch and ascent of the rocket from the launchpad to a pre-determined altitude. In this case, in 2,  $\dot{x}$  and  $\dot{\gamma}$  are zero. The vertical ascent ends up at time  $t_b$  with the states  $[0, h_b, V_b, 90^\circ]$ .

The gravity turn starts with these initial conditions with one difference, the momentary kick angle given to the launcher:  $[0, h_b, V_b, 90^\circ - \Delta\gamma]$  where  $\Delta\gamma$  represents the kick angle. With gravity-turn guidance, the thrust vector parallels the instantaneous velocity vector during the phase. The equations of motion for this case are in 2 where  $\alpha$  is zero.

The gravity turn ends at time  $t_g$  with the states  $[x_g, h_g, V_g, \gamma_g]$ . If data is available, it is considered the end of the gravity turn when the rocket drops its fairing. This occurs when aerodynamic pressure and external temperatures are no longer a concern for damaging the payload. If information is not available, the gravity turn ends when the aerothermal flux( $\phi$ ) goes below certain values (Balesdent, 2011).

$$\phi = \frac{1}{2} \rho V^3.\tag{3}$$

As the vehicle exits the denser layers of the atmosphere, it begins the trajectory optimisation. The first step is to define the boundary conditions for the states of the problem in this optimisation. The initial conditions are

$$x_i = x_g, \quad h_i = h_g, \quad V_i = V_g, \quad \gamma_i = \gamma_g,\tag{4}$$

and the final conditions are

$$x_f = x, \quad h_f = h_c, \quad V_f = V_c, \quad \gamma_f = 0,\tag{5}$$

where  $x_f$  is unspecified,  $h_c$  is the altitude of the orbit of the specific mission chosen and  $V_c$  is the velocity needed to achieve the orbit, in this case, the orbits being calculated are all circular, so  $V_c$  is calculated by (Edberg & Costa, 2022):

$$V_{\text{circ}} = \sqrt{\frac{g_0 R_e^2}{R_e + h_c}}.\tag{6}$$

The main objective of the optimisation is to maximise the payload, which is equal to minimising the fuel consumed in the last stage. So, the objective function is

$$J = t_f - t_i,\tag{7}$$

where  $t_f$  is the final burn time of the last stage, and  $t_i$  is the last stage ignition time. For the free flight phase, the equations used are the ones defined in 2. The control vector is represented by the vector  $u$  equal to  $[\alpha]$ .

To optimise the trajectory, a Hamiltonian (H) is needed, which is a function that helps derive conditions for optimal solutions to control problems. This function is calculated as (Hull, 2003):



$$\begin{aligned}
 H &= \lambda_n V \sin(\gamma) \\
 &+ \lambda_v \left( \frac{T}{m} \cos(\alpha) - g \sin(\gamma) \right) \\
 &+ \lambda_\gamma \left( \frac{\frac{T}{m} \sin(\alpha) - \left( g - \frac{V^2}{R_e + h} \right) \cos(\gamma)}{V} \right).
 \end{aligned} \tag{8}$$

Applying the Weierstrass-Erdmann corner conditions helps to ensure that, in the case of a discontinuity in the optimisation the optimal solutions are maintained (Hull, 2003). The optimal control is then written using the Pontryagin minimum principle

$$u = \arg \min_u H. \tag{9}$$

By solving this equation, the alpha for the optimal control is obtained

$$\alpha = \arctan \left( \frac{\lambda_\gamma}{\lambda_v V} \right). \tag{10}$$

With this equation, the problem is wholly formulated, so the PSO method generates  $N$  particles, each with the initial optimisation parameters (initial values of the co-estates, final burn time, and coast duration if the launcher does a coast phase), with positions and velocities distributed along the search space defined by the bounds of the parameters which depend on the specific mission. Each particle does a shooting method and is attributed a score based on its proximity to the solution. The objective function is then adapted to

$$J' = J + \sum_{n=1}^3 s_n \|y_{n,f} - y'_n\| + s_4 \|H_f^{\text{last stage}} + 1\|. \tag{11}$$

In this equation, the  $s$  terms are penalising factors, and  $\|y_{n,f} - y'_n\|$  is the difference between the final state vector obtained by the particle and the required one for the mission. The PSO method simulates until the limit number of iterations is reached.

## 2.2 External Forces

Launcher trajectories consider three main forces: rocket thrust, aerodynamic forces, and the Earth's gravitational force.

The gravitational force is given by (Edberg & Costa, 2022):

$$F = G \frac{Mm}{r^2}, \tag{12}$$

where  $G$  is the gravitational constant with a value of  $6.67430 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$ ;  $M$  is the mass of Earth equal to  $5.972 \times 10^{24} \text{ kg}$ ;  $m$  is the launcher mass, in this case is negligible;  $r$  is the distance between the launcher and the centre of the Earth.

To compute the aerodynamic loads on a vehicle, it is necessary to determine the atmospheric density, temperature, and pressure as functions of altitude. Two models were used: The exponential model and the U.S. Standard Atmosphere 1976 (US76).

The exponential model is a simplified assumption of the ideal gas law, constant temperature, constant molecular mass, and hydrostatic equilibrium in the atmosphere. The equations are given by

$$\frac{\rho}{\rho_0} = e^{-\beta h} = e^{-\frac{h}{H_s}}. \tag{13}$$

For Earthly applications are  $\rho_0 = 1.225 \text{ kg/m}^3$  and  $H_s = 7640$ , resulting in a constant temperature of  $T_c = 240 \text{ K}$  (Edberg & Costa, 2022).

The aerodynamic forces can be divided into drag, lift, and side force. The drag force opposes the direction of motion. In contrast, the lift force acts perpendicular to the direction of motion, and the side force is mutually perpendicular to the lift and the drag. (Coşkun, 2014)

For this work, the lift and side forces have been disregarded as rockets cannot withstand large angles of attack as the transverse forces would break the rocket. It is a generally valid assumption since the trajectories of launch vehicles are designed to keep the aerodynamic angles small during atmospheric flight. The expression for the drag force is

$$D(t) = -\frac{1}{2}\rho(t)SC_Dv^2(t), \quad (14)$$

where  $\rho$  is the atmospheric density calculated with the atmospheric model;  $S$  is the reference area of the launcher;  $C_D$  is the drag coefficient, which depends on the Mach number and rocket;  $v$  is the velocity vector.

## 3 The launchers

### 3.1 VEGA

The VEGA is a four-stage launch vehicle developed by the European Space Agency (ESA) program. Solid rocket motors power the first three stages, while the fourth stage, AVUM (Altitude and Vernier Upper Module), is a liquid-fuelled, restartable upper stage. (Arianespace, 2014)

| VEGA                 | Stage 1 (P80) | Stage 2 (Z23) | Stage 3 (Z9) | Stage 4 (AVUM) |
|----------------------|---------------|---------------|--------------|----------------|
| Stage Type           | solid         | solid         | solid        | liquid         |
| Diameter [m]         | 3             | 1.9           | 1.9          | 2.18           |
| Total Mass [kg]      | 96243         | 26300         | 12000        | 688            |
| Propellant Mass [kg] | 87710         | 23814         | 10567        | 381            |
| Thrust [N]           | 3015000       | 1120000       | 317000       | 2450           |
| Specific Impulse [s] | 280           | 287.5         | 295.9        | 314.6          |
| Payload Mass [kg]    |               |               | 1500         |                |
| Fairing Mass [kg]    |               |               | 540          |                |

Figure 1: VEGA characteristics (Arianespace, 2014)

The  $C_d$  Vs Mach curve used in this work is based on a work which had the partnership of the prime contractor in developing and producing the VEGA launcher, the ELV (de Volo, 2017).

More recently, ESA upgraded the VEGA rocket to a more upgraded and powerful version, the VEGA-C. This launcher offers an improved performance of about 60%, sending 2300kg to a 700km circular orbit. Just like the VEGA, solid rocket motors power the first three stages, and the AVUM is liquid-fuelled. The launcher characteristics are:

| VEGA-C               | Stage 1 (P120C) | Stage 2 (Z40) | Stage 3 (Z9) | Stage 4 (AVUM) |
|----------------------|-----------------|---------------|--------------|----------------|
| Stage Type           | solid           | solid         | solid        | liquid         |
| Diameter [m]         | 3.4             | 2.4           | 1.9          | 2.18           |
| Total Mass [kg]      | 155027          | 40477         | 12000        | 698            |
| Propellant Mass [kg] | 141634          | 36239         | 10567        | 492            |
| Thrust [N]           | 4323000         | 1304000       | 317000       | 2450           |
| Specific Impulse [s] | 279             | 293.5         | 295.9        | 315.8          |
| Payload Mass [kg]    |                 |               | 2300         |                |
| Fairing Mass [kg]    |                 |               | 860          |                |

Figure 2: VEGA-C characteristics (Arianespace, 2018)

### 3.2 Electron

The Electron is a two-stage light launch vehicle developed by Rocket Lab. The first stage has nine Rutherford sea-level engines, while the second has one Rutherford Vacuum engine. The rocket characteristics are:

| Electron             | Stage 1 | Stage 2 |
|----------------------|---------|---------|
| Diameter [m]         | 1.2     | 1.2     |
| Total Mass [kg]      | 10200   | 2300    |
| Propellant Mass [kg] | 9250    | 2050    |
| Thrust [N]           | 216000  | 22000   |
| Specific Impulse [s] | 311     | 341     |
| Payload Mass [kg]    |         | 200     |
| Fairing Mass [kg]    |         | 860     |

Figure 3: Electron characteristics (Rocket Lab USA, Inc., 2022)

The  $C_D$  versus Mach number curve for the Electron rocket was obtained using a methodology similar to that used for the VEGA launcher. These parameters were input into Missile DATCOM using structural and dimensional data from Rocket Lab, and the corresponding aerodynamic coefficients were computed across different Mach numbers. (Hoefsloot, 2022)

3.3 MURALM

The final rocket considered in this study is the MURALM (Multi-Role Air Launch Missile). MURALM benefits from being released from an aircraft at high altitudes, reducing the impact of the atmospheric and aerodynamical models during the flight. The characteristics of this rocket are:

| MURALM               | Stage 1 | Stage 2 | Stage 3            |
|----------------------|---------|---------|--------------------|
| Total Mass [kg]      | 3608    | 822     | 406 (payload mass) |
| Propellant Mass [kg] | 2480    | 370     | 406 (payload mass) |
| Thrust [N]           | 235180  | 23472   | 7500               |
| Specific Impulse [s] | 290     | 291     | 320                |
| Payload Mass [kg]    |         |         | 406                |
| Fairing Mass [kg]    |         |         | 860                |

Figure 4: MURALM characteristics (Pontani & Teofilatto, 2014)

For this rocket it was considered a constant  $C_D = 0.2$ . The surface area is also considered constant and equal to  $S = 0.588 \text{ m}^2$

4 Algorithm Overview and Validation

4.1 Algorithm Overview

The inputs for the ascent phase include the mass, thrust,  $I_{sp}$ , burn time, reference area, and  $C_D$  plus the beginning of the ascent phase, kick angle, initial conditions, and the time the fairing is dropped.

The inputs of the PSO method are the co-states( $\lambda_h$ ,  $\lambda_v$ ,  $\lambda_\gamma$ ), the duration of the coast time and the final burn time. The system outputs all the flight data for trajectory plots: acceleration, velocity, altitude, downrange, flight path angle, thrust vectoring, dynamic pressure, and mass.

The inputs for the descent phase are the same as those for the ascent phase and the value obtained from the PSO. The output consists of the data necessary to plot the fall of the stages of the rocket of the problem. The Python libraries employed were Pyswarm, Scipy, and Matplotlib:

4.2 Algorithm Validation

The optimisation validation compares the values obtained by optimising the trajectory and the values presented in Pontani and Teofilatto (2014). The descent phase validation compares the numerical results for the downrange obtained with this system with the analytical solution for this problem.

The rocket used for the comparison is the MURALM. The selected orbit is a circular LEO with an altitude of 280 km.

The control and payload obtained may differ because of differences in the models. One of the differences is that the orbit in the article is a 3D orbit. However, the article's second control angle (the  $\beta$ ) never exceeds 0.46 degrees, meaning the thrust direction can be considered co-planar. One other difference is that the atmospheric model used in the article is not specified while in this system it was used the US76.

In this system and the article, the rocket is launched from an aircraft with an initial altitude of 6.7 km and an initial velocity of 280 m/s. The  $h_f$  is 280 km, the  $\gamma_f = 0$  degrees, and the  $V_f = 7736$  m/s. The parameters to be optimised are the co-states, initial gamma, coast time and final burn time. It is possible to see both systems give similar optimisation results.

| Parameter                 | Units | Model      | Article    | $\Delta$ | Relative Error (%) |
|---------------------------|-------|------------|------------|----------|--------------------|
| Final mass                | [kg]  | 118.76     | 138.2      | 19.44    | 14.07              |
| Final altitude            | [m]   | 280000.001 | 280000.000 | 0.001    | 0.00000357         |
| Coast duration            | [s]   | 104.736    | 110.5      | 5.764    | 5.22               |
| Final burn time           | [s]   | 120.184    | 112.0      | 8.184    | 7.31               |
| Initial flight path angle | [rad] | 0.53       | 0.6468     | 0.1168   | 18.06              |
| Final velocity            | [m/s] | 7736.187   | 7736       | 0.187    | 0.0024             |
| Final flight path angle   | [rad] | 0          | 0          | 0        | 0                  |

Figure 5: Comparison of results

The graphics that are adequate to compare are the variables of the state vector:

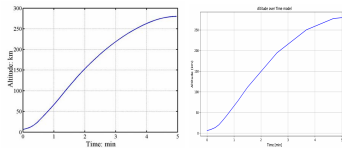


Figure 6: Comparison of altitude graphic

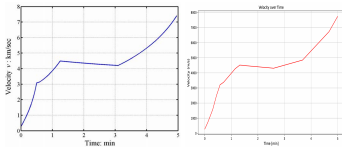


Figure 7: Comparison of velocity graphic

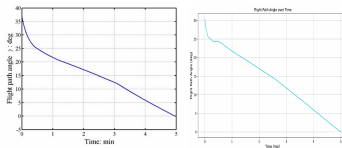


Figure 8: Comparison of the flight path angle

These results validate the trajectory's optimisation.

To validate the whole system, a mission with a vehicle launched from the ground was chosen. This mission simulated all atmospheric flight, optimisation, and descent phases.

For the simulation, the VEGA was selected to achieve a circular target orbit of 700 km with a fixed payload of 1500 kg. The atmospheric model considered was the exponential atmosphere. The mission objectives are:

| Mission Parameter       | Value    |
|-------------------------|----------|
| Altitude [m]            | 700      |
| Velocity [m/s]          | 7502.872 |
| Flight path angle [rad] | 0        |
| Payload mass [kg]       | 1500     |

Figure 9: VEGA mission parameters

For this simulation, the rocket starts its trajectory with  $x = 0$ ,  $h = 0$ ,  $V = 0$ , and  $\gamma = 0$ . The gravity turn begins at 5 seconds and ends when the goes below  $900W/m^2$ , starting the free flight phase and optimisation. The parameters to be optimised are the co-states, the coast time and the final burn time. The state vector of each stage is calculated until  $h = 0$ . The comparison between the optimal solution final values and the objective mission parameters is as follows:

| Parameter               | Mission objective | System values | Delta   | Relative Error (%) |
|-------------------------|-------------------|---------------|---------|--------------------|
| Altitude [m]            | 700000            | 700000.000    | 0.0000  | 0.0000             |
| Velocity [m/s]          | 7502.8720         | 7502.8271     | 0.0449  | 0.0006             |
| Flight path angle [rad] | 0                 | 0             | 0.0000  | N/A                |
| Payload mass [kg]       | 1500              | 1572.709      | 72.7090 | 4.85               |

Figure 10: VEGA final state

The system optimised the mass while satisfying its objectives. The state vector is close to the aim. The payload mass is higher than expected, but it is normal for a launcher to have left-over mass. The graphics of the states of the system are:

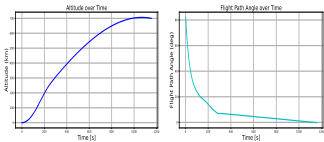


Figure 11: Altitude and flight path angle VEGA

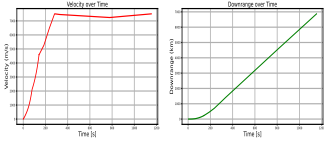


Figure 12: Velocity and downrange VEGA

To conclude the validation, the descent phase of the VEGA stages needs to be calculated. To calculate the descent trajectory, first, the optimal ascent trajectory is calculated using the values given by the PSO method. This values are:

| Parameter of the PSO | Value from the PSO |
|----------------------|--------------------|
| $\lambda_h$          | 0.00307            |
| $\lambda_V$          | 0.54867            |
| $\lambda_\gamma$     | 0.12805            |
| Coast duration [s]   | 292.18             |
| Final burn time [s]  | 604.70             |

Figure 13: VEGA optimisation parameters

It is considered the stage fall of the first two stages, while the last two burn in the atmosphere.

Each stage begins the descent phase only with the structural mass. It is possible to compare the analytical existing solution for a ballistic fall with the numerical solution obtained with the system with some simplifications. The gravity was considered constant while the air density and drag were disregarded throughout the descent phase. The term of the centrifugal force in the  $\dot{\gamma}$  is not considered.

The following graphics show the comparison between the analytical and numerical solution:

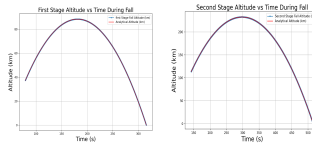


Figure 14: Comparison between analytical and numerical solutions without drag: First stage and second stage

It is possible to see that the curves overlap. This means that the numerical solutions of the system are coherent with the analytical expressions for the ballistic descent problem.

The simplifications made to compare with the analytical solutions are considered too simplistic. For this reason, the way to validate the descent phase consists of comparing the simplified analytical solution with the numerical solution obtained from the system and concluding whether they are logical.

The equations used for the numerical solution were the ascent equations of motion without thrust, with the same gravity and atmospheric models used in the ascent phase.

The following graphics show the comparison between the analytical and numerical solutions in these conditions:

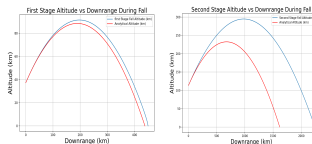


Figure 15: Comparison between analytical and numerical solutions with drag: First stage and second stage

The graphics show a disparity between the downrange of the analytical and numerical solutions, which gets bigger with each stage. As expected, the numerical solutions always reach a higher altitude and downrange because the gravity and centrifugal forces have been simplified. Gravity pulls the rocket in the direction of the ground, so if gravity is considered constant, the magnitude of the force will be higher than it is because it decreases with altitude. In addition, not considering the centrifugal force, an extra force that pulls the rocket away from the surface of the area, contributed to the rocket not gaining as much altitude as it is supposed to in the analytical solution case.

There is also a slight difference between the graphics when the rocket is close to the ground. This is because of the terminal velocity off the stage, which means the stage stops accelerating. The downrange it reaches is inferior to what it would be without drag.

In conclusion, the variations are reasonable and justifiable. The numerical results demonstrate high consistency with established aerodynamic and trajectory principles, thus validating the overall methodology.

## Santa Maria Case Study

The selected launchers for the case study are the Electron and VEGA-C, which allow for studying small and medium-sized launched vehicles. The chosen orbit for the Electron mission is a low-inclination, 90-degree azimuth LEO of 400 km altitude. The target orbit is also a low-inclination, 90-degree azimuth LEO of 700 km altitude for the VEGA-C. The 90-degree

azimuth allows the trajectory to take advantage of the Earth's rotation. The rocket, In the case of Santa Maria, benefits from the Earth's rotation velocity with  $V_{\text{rotation}} = 465 \cos(\text{latitude})$ , which for the latitude of Santa Maria ( $36^{\circ}58'$ ), is 372 m/s.

**5.1 Electron Mission**

The mission objectives are:

| Parameter               | Mission Objective |
|-------------------------|-------------------|
| Altitude [m]            | 400000            |
| Velocity [m/s]          | 7667.283*         |
| Flight path angle [rad] | 0                 |
| Payload mass [kg]       | 200               |

\* Actually 7295.283 m/s because of the Earth's rotation.

Figure 16: Electron mission objectives

For this simulation, the rocket starts its trajectory at the ground with  $V = 0$  ,  $h = 0$  ,  $x = 0$  and  $\gamma = 0$ . The vertical phase lasts 5 seconds when the gravity turn begins with a kick angle of 1 degree. The fairing is dropped when the altitude reaches 126 km. The parameters to be optimised are the co-states and the final burn time;

The optimal solution parameters and the ascent final states obtained for this mission were:

| Parameter               | Mission Objective | System Final Value |
|-------------------------|-------------------|--------------------|
| Altitude [m]            | 400000            | 399962.7           |
| Velocity [m/s]          | 7295.283          | 7295.282           |
| Flight path angle [rad] | 0                 | -0.00021           |
| Payload mass [kg]       | 200               | 217.0509           |

Figure 17: Electron final state

The graphics for the different states for the ascent trajectory are:

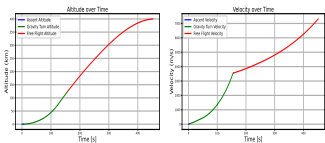


Figure 18: Altitude and velocity plot Electron

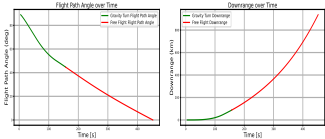


Figure 19: Flight path angle and downrange plot Electron

With the optimisation parameters the descent phase of the first stage of the Electron is studied. The second stage of the Electron re-enters the atmosphere and burns when jettisoned. (Rocket Lab USA, 2020).

The descent trajectory of the first stage is given by:

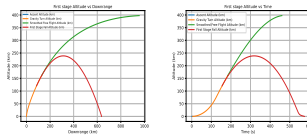


Figure 20: Descent trajectory first stage Electron

The final downrange of the first stage of the Electron rocket is 636.194 km. The main objective of this study is to evaluate whether the Electron launched from the Santa Maria Spaceport poses any safety risks.

The first step in this risk analysis is to use the calculated downrange distance to identify the possible impact points for the first stage. Variations in azimuth could potentially direct the impact toward areas that may not be safe, such as shipping lanes or the vicinity of other islands in the Azores archipelago. The radius of the possible points of impact is:

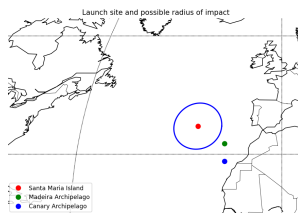


Figure 21: Possible impact points first stage Electron

Based on the system's downrange results, it is highly likely that a launcher with the characteristics of the Electron's first stage will fall into the sea, posing no significant risk to people or infrastructure. This is a positive indication of the feasibility of launching from the Santa Maria Spaceport.

Santa Maria's position in the eastern group of the archipelago offers flexibility in terms of azimuth selection, with launch angles between 0 and 180 degrees being particularly viable.



Figure 22: Santa Maria localisation in Azores. Adapted from (Açores, 2024).

Another critical factor is the maritime traffic close to the impact points.

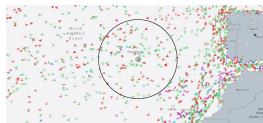


Figure 23: Maritime traffic near Santa Maria (MarineTraffic, 2024)

The maritime traffic patterns around Santa Maria are focused along European and African coasts. The region around Santa Maria is relatively free of major shipping lanes. This suggests a lower risk of interference with shipping lanes.

Due to the geography of Santa Maria, the location of the spaceport on Santa Maria island, and the maritime traffic concerns, based on the Electron first stage impact points, the possible azimuths will be:





Figure 24: Possible launch azimuths ELECTRON, adapted from (Ávila & de Frias Martins, 2007)

In conclusion, launching small vehicles like the Electron from Santa Maria spaceport offers significant advantages, more precisely, azimuths between  $90^\circ$  and  $180^\circ$ . This range supports a variety of orbital inclinations, making it possible for different mission profiles, including sun-synchronous and low-inclination orbits.

The DEIMOS study, which evaluated the Orbex rocket—a vehicle similar in dimensions to the Electron, supports this conclusion. Their findings, paired with the results from this case study, indicate that Santa Maria's location and the range of azimuths make it an optimal launch site for small-sized rockets (Elecnor Deimos, 2018).

5.2 VEGA-C Mission

The mission objectives are:

| Parameter               | Mission Objective |
|-------------------------|-------------------|
| Altitude [m]            | 700000            |
| Velocity [m/s]          | 7502.871*         |
| Flight path angle [rad] | 0                 |
| Payload mass [kg]       | 2300              |

\* Actually 7130.871 m/s because of the Earth's rotation.

Figure 25: VEGA-C mission objectives

For this simulation, the rocket starts its trajectory at the ground with  $V = 0$ ,  $h = 0$ ,  $x = 0$  and  $\gamma = 0$ . The vertical phase lasts 5 seconds. Then, the gravity turn begins with a kick angle of 1 degree. The fairing is dropped when the aerothermal flux:  $\phi < 800 \text{ W/m}^2$ . There is a coast phase during the free flight phase starting at the end of the third stage burn time. The parameters to be optimised are the co-states, the coast duration and the final burn time.

The optimal solution parameters obtained were:

| Parameter                     | Mission Objective | Calculated Value | Delta   |
|-------------------------------|-------------------|------------------|---------|
| Coast Duration [s]            | N/A               | 271.04           | N/A     |
| Final Burn Time [s]           | N/A               | 505.55           | N/A     |
| Final Altitude [m]            | 700000.0          | 699976.7         | -23.3   |
| Final Velocity [m/s]          | 7130.871          | 7100.232         | -30.639 |
| Final Flight Path Angle [rad] | 0.000             | 0.0100           | 0.0100  |
| Payload Mass [kg]             | 2300.000          | 2307.9218        | 7.9218  |

Figure 26: VEGA-C final state

The graphics for the ascent trajectory are:

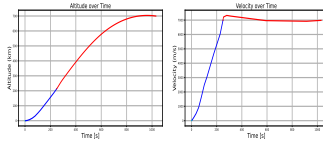


Figure 27: Altitude and velocity plot VEGA-C

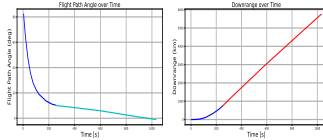


Figure 28: Flight path angle and downrange plot VEGA-C

For the VEGA-C, only the first two stages were calculated. The third and fourth stage operate at higher altitudes where several factors introduce considerable unpredictability and the stages may not remain intact. The descent trajectories are:

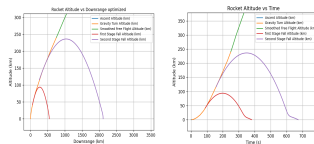


Figure 29: Descent trajectory VEGA-C

The downrange for the first and second was 545.582 km and 2111.612 km, respectively.

As in the Electron risk analysis, the calculated downrange distance is used to identify the possible impact points. The two radii of the possible points of impact are:

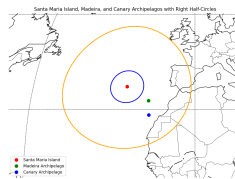


Figure 30: Possible impact points for the VEGA-C stages

The first stage of the VEGA-C follows a similar trajectory to the Electron rocket's first stage, which has already been described. Launching with an azimuth between 90 and 180 degrees would result in the first stage, most likely falling into uninhabited ocean areas.

The second stage presents a different challenge. As currently calculated, the optimal trajectory for the second stage would result in a downrange that extends toward Europe or Africa, which violates the safety concerns.

One solution to the problem is to reduce the payload. Based on safety considerations, the maximum allowable downrange for the second stage is 1300 km. This is calculated by accounting for the width of the maritime traffic which width approximately 90 to 180 km, while avoid any populated areas (European Marine Observation and Data Network (EMODnet), 2024).

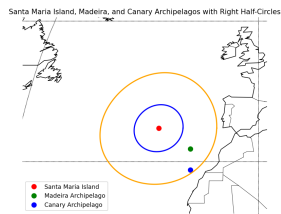


Figure 31: Maximum downrange for the second stage of the VEGA-C

Reducing the payload mass on the VEGA-C is a possible solution to reduce the downrange to respect the safety requirements. The trajectory is re-optimised for each new payload mass, and the resulting downrange distance is calculated. Reducing the payload reduces the rocket's overall mass, reducing the total energy needed for the flight. This leads to a shorter downrange distance upon second-stage separation.

| Payload Mass [kg] | Downrange second stage [km] |
|-------------------|-----------------------------|
| 2300              | 2111.612                    |
| 2100              | 2102.809                    |
| 1900              | 2100.868                    |
| 1700              | 2090.80                     |

Figure 32: Downrange of the second stage for different payloads

It is possible to see that although the payload is being reduced, the downrange of the stage is not decreasing enough to allow the azimuths that direct the rocket towards the direction of Europe and Africa. One possibility for this event is that the second stage is jettisoned very closely to the beginning of the control phase.

To mitigate this fact, a possibility would be to add the kick angle or the duration of the vertical phase to the optimisation variables.

For these reasons, the VEGA-C azimuths from Santa Maria are limited to those that do not reach the European or African coastlines nor fly over Santa Maria island or the other islands in the Azores.



Figure 33: Possible azimuths for the VEGA-C

In conclusion, launching medium-sized launch vehicles like the VEGA-C from Santa Maria spaceport is possible but is limited to azimuths between 170° and 270°. These azimuths allow for sun-synchronous orbits.

## 6 Conclusions

This work developed a simulation framework to optimize rocket launch trajectories from the Santa Maria Spaceport. It used models of the VEGA-C and Electron rockets coupled with generic trajectory models to calculate optimal ascent trajectories and predict impact zones for separated stages.

The results demonstrated the effectiveness of this approach. The obtained solutions approximated the rockets' real capacity (mass to orbit) and correctly predicted the impact zones of the falling stages.

The case study on Santa Maria's feasibility as a launch site demonstrated favourable conditions for small launch vehicles. While medium-sized rockets face some limitations, Santa Maria shows strong potential for future commercial and scientific missions.

Implementing a 3DOF trajectory model is highly recommended in future work. Also, if the study extended to more types of rockets, it would be useful to understand the rocket size limit possible to launch from Santa Maria.

## References

- Arianespace. (2014, April). Vega User's Manual: Issue 4 – Revision 0 [Computer software manual]. (Available at: <https://www.arianespace.com/mission/vega/>)
- Arianespace. (2018, May). Vega C User's Manual, Issue 0 Revision 0 [Computer software manual]. Arianespace. (Issued and approved by Roland LAGIER, Senior Vice President, Chief Technical Officer)
- Açores. (2024). *Mapa dos açores*. Retrieved from <https://byacores.com/mapa-dos-acores/> (Accessed: 2024-10-19)
- Balesdent, M. (2011). *Multidisciplinary Design Optimization of Launch Vehicles* (Unpublished doctoral dissertation). Ecole Centrale de Nantes (ECN). (HAL Id: tel-00659362, <https://theses.hal.science/tel-00659362v1>)
- Betts, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance Control and Dynamics*, 21, 193-207.
- Coşkun, E. C. (2014). *Multistage launch vehicle design with thrust profile and trajectory optimization* (Ph.D. Thesis). Middle East Technical University, Graduate School of Natural and Applied Sciences. (In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mechanical Engineering)
- de Volo, G. D. C. B. (2017). *Vega launchers' trajectory optimization using a pseudospectral transcription* (Unpublished master's thesis). Delft University of Technology. (Master of Science Thesis Research, ELV - Mission Design & Guidance Team)
- Edberg, D. L., & Costa, G. (2022). *Design of rockets and space launch vehicles* (Second ed.). Reston, Virginia: The American Institute of Aeronautics and Astronautics, Inc. Retrieved from <https://lccn.loc.gov/2022012407>
- Elecnor Deimos. (2018). *Azores spaceport study*. Retrieved from <https://elecnor-deimos.com/azores-spaceport-study/> (Accessed: October 12, 2024)
- European Marine Observation and Data Network (EMODnet). (2024). *Emodnet human activities - maritime transport density*. Retrieved from <https://www.emodnet-humanactivities.eu/view-data.php> (Accessed on October 21, 2024)
- Hoefsloot, S. D. (2022). *A feasibility study on the recovery of electron's first stage using a vertical landing approach* (Unpublished master's thesis). Delft University of Technology.
- Hull, D. (2003). *Optimal control theory for applications*. New York, NY: Springer.
- MarineTraffic. (2024). *Live map - marinetrffic*. [urlhttps://www.marinetraffic.com/en/ais/home/centerx:-29.5/centery:34.3/zoom:5](https://www.marinetraffic.com/en/ais/home/centerx:-29.5/centery:34.3/zoom:5). (Accessed: 2024-10-21)
- Pontani, M., & Teofilatto, P. (2014). Simple method for performance evaluation of multistage rockets. *Acta Astronautica*, 94, 434-445.
- Rocket Lab USA, Inc. (2022, November). Electron payload user's guide [Computer software manual]. Retrieved from <https://www.rocketlabusa.com> (Accessed on: Oct 12, 2024)
- Rocket Lab USA, I. (2020). *The kick stage: Responsible orbital deployment*. Retrieved from <https://www.rocketlabusa.com/updates/the-kick-stage-responsible-orbital-deployment/> (Accessed: 2024-10-18)
- Sforza, P. M. (2016). *Manned spacecraft design principles*. Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo: Butterworth-Heinemann, Elsevier.
- Sotto, E. D., & Teofilatto, P. (2002). Semi-analytical formulas for launcher performance evaluation. *Journal of Guidance Control and Dynamics*, 25, 538-545.
- Trélat, E. (2012). Optimal control and applications to aerospace: Some results and challenges. *Journal of Optimization Theory and Applications*, 154, 713-758.
- Ávila, S. P., & de Frias Martins, A. M. (Eds.). (2007). *Palaeontology in atlantic islands. proceedings of the "1st atlantic islands neogene", international congress* (Açoreana, Supplement 5 ed.). Açoreana.



# Military Boot Vibration Energy Harvesting



Autor: **ALF/ENGEL Eduardo Filipe Rocha Pinto**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra / Instituto Superior Técnico, Lisboa

Orientador: **Prof. Luís Filipe Soldado Granadeiro Rosado**  
Instituto Superior Técnico, Lisboa

Coorientador: **CAP/ENGEL Hugo Costa**  
Força Aérea Portuguesa

## Abstract

The work done investigates the design, implementation, and validation of a vibration-based energy harvesting system integrated into a military boot. The goal is to develop a wearable energy system capable of converting human motion into electrical energy, providing a sustainable solution for reducing reliance on external batteries and minimizing the carried load of soldiers during missions. By utilizing electromagnetic transducer technology, the system captures vibrations generated by heel strikes and converts them into usable electrical power. The research focuses on optimizing vibration energy conversion through careful analysis of key parameters, including resonance frequency, magnet spacing, coil design, and load resistance. Laboratory experiments revealed that the system performs optimally at a resonance frequency of 13.96 Hz, achieving a peak output power of 83.06  $\mu$ W under controlled conditions. Maximum power transfer was observed with a load resistance of 1000 Ohms, while structural configurations, such as the positioning of fixed and core magnets, significantly influenced energy output and overall system efficiency. Despite its limited power generation capacity, the system demonstrates the feasibility of harnessing vibration energy for small-scale applications, such as powering low-demand devices. The findings underscore the importance of optimizing resonance conditions and load matching to enhance energy harvesting efficiency. This work highlights the potential of wearable energy systems for military power solutions, contributing to the advancement of portable and sustainable energy technologies for field operations and other remote applications.

**Keywords:** Energy Harvesting, Electromagnetic Transducer, Portable Energy System, Resonance Frequency, Military Energy Solutions.

## 1 Introduction

Over the last decade, soldiers have become more dependent on electronics to complete their missions. Devices such as the Global Positioning System, radios, Night Vision Goggles, cellphones, and optical sights are used on short and long missions. (Pinto & Dias, 2023), (Pinto & Pires, 2024). This amount of devices creates weight challenges due to the extra batteries the soldiers are required to complete their duties. According to Bhatnagar Lang, in 2006, a lot of weight carried by a soldier on a given mission is "parasitic until it is needed". This is where the batteries for the radios are located, and for a 72-hour mission without resupply, the loaded weight varies between 7 and 9.5 kg (Bhatnagar & Lang, 2006), (Easley, 2022). The power load requirements on the infantry have increased seriously over the last years and will increase even more when electronic warfare becomes a global type of gun (Mittal, 2020).

Currently, on long missions, there are some limitations regarding sources of electric power and weight loads carried by soldiers in isolated zones. These factors aggravate the military's performance and, therefore, the chances of a successful mission. By extending the studies on energy consumption and renewable and self-generated energy sources, it's possible to have a better understanding of the impact of energy harvesting system models and their real application and efficiency.

Having this in mind, it's crucial to analyze similar systems implemented to ensure it's possible to harvest energy from the vibrations capped on the soldier's boots.

According to the material used in standard missions by the Portuguese Joint Terminal Attack Controller (JTAC) Forces given by Lieutenant Patrick Pires, there are some crucial devices in case of complete isolation of the team. Most of these devices required AA or AAA batteries.

## 2 State-of-the-Art

Energy harvesting refers to the process of extracting energy from the environment around and converting it into usable electrical energy. Energy can be extracted from a variety of sources (Ali, Yahaya, Nallagownden, & Zakariya, 2014), which are separated into two categories based on their properties. First, there are the naturally occurring resources that come straight from the surroundings. Examples of these include radio-frequency (RF), solar power, temperature gradients (particularly geothermal heat), and mechanical vibration. Second, artificial sources, are the result of human or system behaviors and are not produced by nature, such as breathing, body motion, blood pressure, and system vibrations while in operation (Priya, 2007).

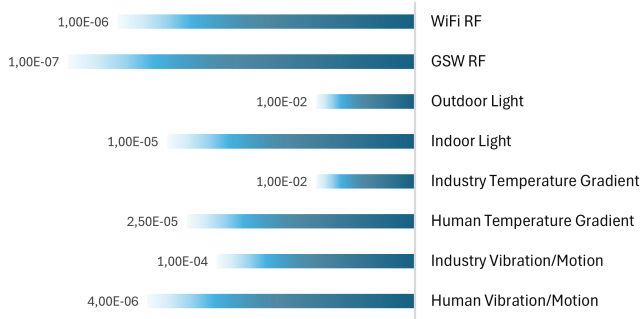


Figure 1: Approximate amount of energy per unit available from four micro-harvesting sources (Raju, 2008).

## 2.1 Overview of Energy Harvesting From Vibrations

The vibration energy harvesters can be separated into three types of transducers: piezoelectric, electrostatic, and electromagnetic.

The term piezoelectric effect refers to how piezoelectric materials deform when exposed to external stress. This occurs when the material's elasticity and dielectric properties combine to generate opposite charges on the material's surface, which results in the polarization phenomenon (Bera & Sarkar, 2016). The direct and the converse piezoelectric effects are shown by the Equations 1 and 2, respectively:

$$D_i = e_{ij}^\sigma E_j + d_{im}^d \sigma_m \quad (1)$$

$$\varepsilon_k = d_{jk}^c E_j + S_{km}^E \sigma_m, \quad (2)$$

where vector  $D_i$  is the dielectric displacement in N/mV or C/m<sup>2</sup>,  $\varepsilon_k$  is the strain vector,  $E_j$  is the applied electric field vector in volts/meter and  $\sigma_m$  is the stress vector in N/m<sup>2</sup>. The piezoelectric constants are the piezoelectric coefficients  $d_{im}^d$  and  $d_{jk}^c$  in m/V or C/N, the dielectric permittivity  $\varepsilon_{ij}^\sigma$  in N/V<sup>2</sup> or F/m, and  $S_{km}^E$  is the elastic compliance matrix in m<sup>2</sup>/N. The superscripts  $\sigma$  and  $E$  denote that the quantity is measured under constant stress and constant electric field, respectively, and the superscripts  $c$  and  $d$  stand for the converse and direct effects, respectively (Kim, Kim, & Kim, 2011).

Michael Faraday, in 1831, found that when a magnet and coil move, relative to one another, the cutting of the magnetic flux creates a current induced in the wire, creating voltage as a result. This amount of voltage created differs on the number of loops in the coil and the rate of change in the magnetic flux. This principle is described by Faraday's law:

$$\varepsilon = -N \frac{\Delta \phi}{\Delta t}, \quad (3)$$

where  $\varepsilon$  is the voltage produced in terms of Electromotive Force (EMF),  $N$  is the number of loops of the coil, and  $\phi$  is the magnetic flux across the coil section. A negative sign arises due to Lenz's law (which says that the (EMF) and the rate of change in magnetic flux have opposite signs). By investigating the rate of change in the magnetic flux, Equation 3 can be broken down further like showed in Equation 4, where,  $\beta$  is the strength of the magnetic field,  $l$  is the length of the wire and  $v$  is the relative velocity between the magnet and the wire (Kraus, 1991).

$$\varepsilon = \beta l v \quad (4)$$

An Electrostatic Energy Harvester (EEH) is a kind of vibration-based harvester that consists of capacitive structures comprising of two conductive plates separated by an insulating layer, which could be air or a dielectric medium. When two conductive plates experience relative motion, the capacitance between them changes. This variation in capacitance, when paired with an external voltage source or a pre-charged capacitor, causes a movement of charge, resulting in the generation of electrical energy (Khan & Qadir, 2016).



2.2 Electromagnetic Transducers

The electromagnetic transducers work with bases on Faraday's Law as referred to previously. Figure 2 illustrates the scheme of a typical electromagnetic vibration energy harvester. This mechanism exhibits vibration motion, forced by the base that induces the vibration on the coil. When the coil vibrates, the coil cuts through the magnetic field produced by the two permanent magnets. This relative motion will result in generating an induced current within the coil producing power, according to Faraday's law of induction. (Foong, Thein, Ooi, & Aziz, 2018), (Kraus, 1991)

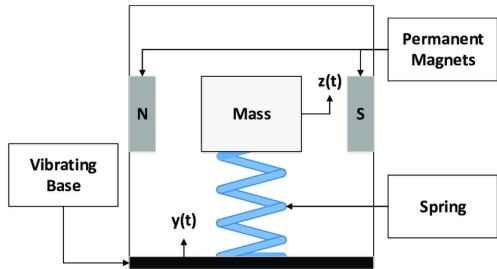


Figure 2: Model of an Electromagnetic Vibration Energy Harvester (Foong et al., 2018).

2.2.1 One Magnet Spring-Based

Harvesting electromagnetic energy from one magnet spring-based design brings improved results. It consists of a non-metallic cylindrical structure featuring a single magnet and two spiral springs, one linked to the magnet's top and the structure's top (prevents the appearance of gravitational forces onto the magnet), and the other to the magnet's bottom and the structure's bottom. Additionally, a coil around the magnet outside the structure ensures the energy can be harvested. The coil experiences vibration from an external source, which the helical springs absorb. This magnet movement through the coil modifies its magnetic flux, resulting in the coil experiencing a voltage (Min-Chie, Chang, Long-Yi, & Chung, 2015).

This design has a lower damping effect on its oscillations, but the usage of the springs generates a buckling phenomenon creating deformations in the spring shape under load, partially absorbing the vibrational energy and consequently diminishing the overall system performance.

2.2.2 Flux-Guided Magnet Stacks

This mechanism consists of a movable, non-magnetic ball inserted between two front-facing, frequency-upconverted electromagnetic generators. A coil is wrapped around the magnet stack over a cylindrical housing, and each generator consists of a flux-guided magnet stack with a cylindrical shape that is supported by a helical compression spring (Halim, Cho, Salauddin, & Park, 2016). The primary limitation of this electromagnetic energy harvester is the requirement for the motion source to be highly efficient and directionally precise (aligned with the designated arrows). Only under these conditions can the mechanical force exerted on the magnet be optimized to achieve maximum electrical energy output. Over time, the impact of the inertial mass on the magnets can lead to degradation of both the mass and the magnets, similar to the wear observed in one-magnet, spring-based electromagnetic energy harvesters (Halim et al., 2016).

2.2.3 Magnet Levitation Vibration

A Magnet Levitation Vibration Energy Harvester converts ambient vibrations into electrical energy using a levitated magnet by repulsive magnetic forces. This allows the magnet to oscillate freely in response to vibrations, with these oscillations converted into electricity. The main advantages of this harvester include high efficiency at low frequencies, reduced mechanical wear due to the lack of physical contact, and the ability to operate across a wide range of vibration

frequencies. However, the system's complexity, sensitivity to environmental conditions, limited power output, and potential for magnet demagnetization are notable drawbacks (Kecik, Mitura, Lenci, & Warminski, 2017).

There are several possible combinations of the number of magnets levitating and coil arrangements that can be used to build an electromagnetic energy harvester using this type of design.

Chunfang, Shuai, Patrick, Min, and Zongxia (Li, Wu, Luk, Gu, & Jiao, 2019), in 2019, proposed an electromagnetic energy harvester shown in Figure 3 using two fixed magnets at both ends of the container, four levitating magnets spaced, and an arrangement of three coils of rings wrapped over the outer surface. The levitating magnets are in a cylindrical configuration, in opposing orientations, allowing the stack to move together and with a smoother motion.

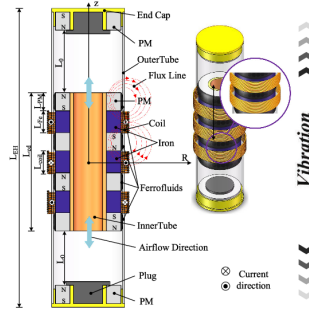


Figure 3: Multiple magnets and multiple coils design proposed in (Li et al., 2019).

The friction between the vibrator and the tube can consume most of the energy of the harvester. To improve the efficiency of the harvester, A ferrofluid is utilized around the curved edges of the moving magnets to concentrate in the areas with the strongest magnetic fields. Using the ferrofluid a hydro-static forms making the stack of magnets make no contact with the inner wall of the container, thus resulting in smaller friction forces.

When the harvester is subjected to external vertical forces, the stack will vibrate along the tube, causing the magnetic flux to be cut by the coils generating electrical energy through electromagnetic induction. As referred before the transducer of an electromagnetic energy harvester is based on Faraday's law of electromagnetic induction. The theory states that the voltage generated  $U$  is related to the rate of change of net magnetic flux. The voltage generated  $U_m$  across a winding can be given by

$$U_m = -n \frac{d\phi}{dt} = -n \frac{d \left( \int \vec{B} \cdot d\vec{A} \right)}{dt} = -nBl\dot{z}, \quad (5)$$

where  $n$  is the number of winding turns,  $\dot{z}$  is the relative velocity between the magnet stack and the winding,  $\phi$  is the magnetic flux,  $B$  is the magnetic flux density,  $dA$  is the differential element area of a magnet-winding assembly, and  $l$  is the total length of coils windings.

## 3 Methodology

### 3.1 Development and Architecture

The project approach starts with the data acquired utilizing the system built in the 1st Cycle Integrated Project by E. Pinto, H. Lima, and R. Sampaio in 2022. The developed system is a data collection module acceleration data that features a microcontroller and a tri-axial analog accelerometer powered by a rechargeable battery, thus allowing remote data acquisition (Pinto, Lima, & Sampaio, 2022). This work had good results when testing vibrations from human motion at three speeds: slow walking, fast walking, and running. The device was attached to a person's feet, and then performed the respective movements at the same distance, approximately 100 m. The data from this work was analyzed in MATLAB software to understand the frequencies generated by the impact of the boot when walking.

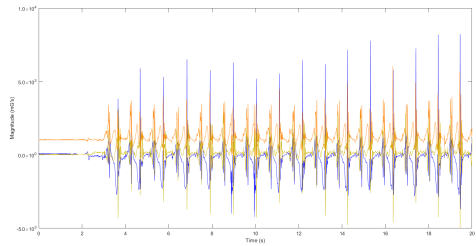


Figure 4: Normal pace walking data (Pinto et al., 2022).

After the data analyses, it's crucial to adjust the work done by H. Cunha in 2021. The developed system in this thesis was an induction-based vibration energy harvesting system that was tested with a vibration generation as shown in Figure 5. (Cunha, 2021).

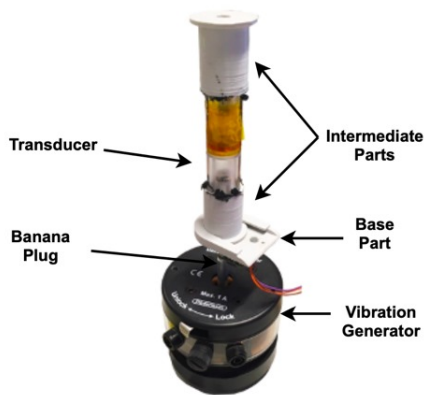


Figure 5: 3D Printed induction-based vibration energy harvesting system and Vibration generator used by H. Cunha(Cunha, 2021).

This work on laboratory vibration generation equipment will enable the creation of conditions in the laboratory that mimic the vibrations in a military boot, making the transducers' design easier. With the data and information obtained and the design of the transducer, the system will be implemented on a boot used by the Portuguese Air Force to validate the data acquired in the laboratory.

3.2 Energy Harvester Architecture

As referred to in Sub Subsection 2.2.3 there are several types of architectures of energy harvesters with different designs. For this project, the architecture tested was an adaption of the design used by the authors in (Li et al., 2019). The design had a fully closed capsule that wouldn't allow the airflow inside giving some restrictions on the friction of the levitation magnets and spacers. To reduce the amount of friction between the magnets and the inner wall and create airflow the structure that connects the top and base fixed magnets is composed of four pillars. The 3D model used for this architecture is shown in Figure 6.

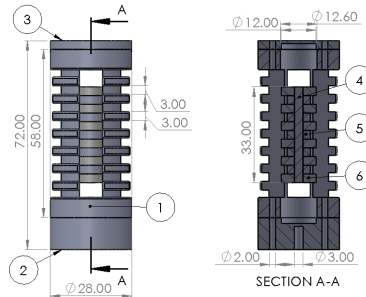


Figure 6: Design of the transducer structure using six levitating magnets adapted from (Li et al., 2019).

Table 1: Description of Components in Figure 6.

| Nº | Component          | Subsection |
|----|--------------------|------------|
| 1  | Main Structure     | 3.2.1      |
| 2  | Bottom Structure   | 3.2.2      |
| 3  | Top Structure      | 3.2.3      |
| 4  | Magnets Core Shaft | 3.2.4      |
| 5  | Magnets Spacer     | 3.2.4      |
| 6  | Magnet             | 3.2.5      |

### 3.2.1 Main Structure

The Main Structure was designed to hold the other components for a perfect alignment of the transducer. On the outside wall of the container were created spaces to wrap the coil (with dimensions of  $3mm * 3mm$ ). To create a better airflow and allow visualization of the core movement the container was cut in four sides and has an inner zone where the core can move freely in vertical movements. In addition, the structure has eight holes, four on top and four on the bottom to connect to the bottom and top structures.

To allow movement without the rotation of the magnets inside the container a core was designed that connects the magnets giving the projected space between them. For this, it was design a shaft and also spacers.

### 3.2.2 Bottom Structure

The Bottom Structure was designed to accomplish three requirements: (1) to connect the transducer to the fixation on the vibration generator, (2) to connect and adjust the distance between the fixed magnet and the main structure, and (3) to hold one of the fixed magnets.

### 3.2.3 Top Structure

The Top Structure has a similar design to the bottom structure except for the hole that connects to the vibration generator, thus it's smaller in terms of height.

### 3.2.4 Magnets Core Shaft

The planned shaft design was a cylindrical piece, that covered the complete length of the stack, with perforations on both ends to lock the magnets stack with a screw. The shaft diameter was designed to fit inside the magnet and the spacer

holes. Due to the small dimensions of the design and the amount of imperfections with the 3D printed design, the shaft was replaced by a metal threaded shaft with one hex nut on each end. The spacers have structures identical to the magnets in use, differing in diameter. The design also has holes to be connected to the magnets and to the shaft.

3.2.5 Magnet

The transducer has eight magnets, one in the top structure, one in the bottom structure, and six in the core connected by the shaft and spaced with the spacers as referred previously. The magnets were bought in (Magnetpt, 2024). These magnets have the following dimensions:

- Diameter- 12 mm;
- Hole Diameter- 3 mm;
- Height- 3 mm.

4 Experimental Results

4.1 Setup

To be able to analyze system comportment in a laboratory, to optimize it, was built a setup that permitted to control of the vibrations. Thus was necessary to generate vertical vibrations to the transducer and read the output voltage of it. The results given were then analyzed in MATLAB as shown in sections 4.2, 4.3, and 4.4.

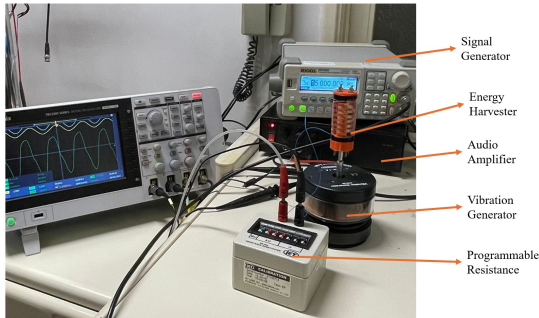


Figure 7: Complete setup.

The vibration source was selected according to the previous work done by (Cunha, 2021). Cunha selected the Vibration Generator 2185.00 which is a versatile vibration generator that allows different waveform inputs and the operating frequency ranges from 0.1 to 1.5 KHz. The generator also allows us to test the transducer sideways or with an inclined angle but since the objective of this work is to analyze the impacts of a foot when it enters in contact with the floor, it was only used to generate vertical forces.

The concept of the Vibration Generator 2185.00 is essentially a loudspeaker. A bobbin wound on a thin aluminum tube moves within a permanent magnetic field when alternating current is applied to it. The bobbin is suspended between two membranes for optimal guidance. The generator has a banana plug connector that was used to connect the transducer as said in subsection 3.2.2.

The signal generator was introduced in the setup to select the type of wave to provide input to the vibration generator selected. The signal generator selection allows us to give a specific type of wave with the frequency and input voltage to do the necessary tests for this work. When tested the signal generator couldn't produce enough current to emulate the

vibration generator as planned. So an audio amplifier was introduced to the system connected to the output of the signal generator.

A simple programmable resistance was inserted into the setup to measure the energy harvested design's generated power. With this new introduction, the load values could be varied to understand the point of maximum power.

## 4.2 Wave/Vibration Characterization Results

To get the most accurate results of the amount of power possible to harvest from the system, it was necessary to do some tests on the data collected by the work done by H. Lima, and R. Sampaio, and (Pinto et al., 2022). To understand which frequencies are generated by the impact of the heel the data acquired from the data logger on the work done by H. Lima, and R. Sampaio, and (Pinto et al., 2022) was studied. By passing the data through a Fast Fourier Transform, the signal converts from time into frequency domain. When transformed into the frequency domain the display shows how the amplitude varies with the frequency. By doing the simulation it's possible to identify harmonic frequencies in mechanical structures, as shown in Figure 8.

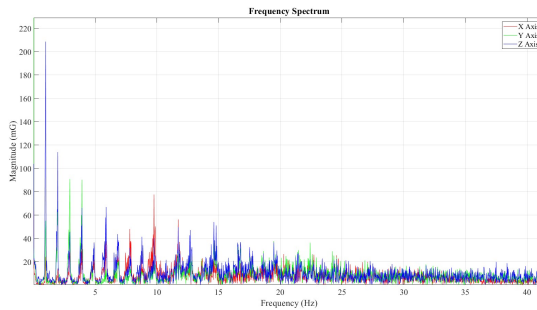


Figure 8: Fast Fourier Transform of acquired data from H. Lima, and R. Sampaio, and (Pinto et al., 2022).

The data analyzed was gathered on a test conducted of the acceleration on a foot while walking at a normal pace for approximately 100 meters using the Portuguese Air Force boots. In Figure 8 it's possible to understand that the frequencies with higher magnitudes range from 1 Hz to 20 Hz, so this range of frequencies was tested in the transducer designed to understand the amount of power generated with the energy harvested built.

## 4.3 Distance of Structures Results

As referred before the energy harvester structure was designed to allow the user to change the space between the main structure and the top ends structure. By this, it's possible to analyze the effect of the distances between the fixed magnets and the magnets on the core.

Initially, the coil was made out of a copper wire with 0.20 mm of diameter. This wire was wrapped around the structure with 900 turns in wire coils. By this, it was possible to validate the concept without the difficulties of wrapping the structure of the energy harvester with the smaller wire, such as the high number of laps and the lower strength.

After adjusting the structure of the system so it could be tested at different distances, and after changing the wire the tests were redone. This time the wire selected had 0.08 mm of diameter and was wrapped around the structure for 4200 laps, 700 laps in each section. Although reducing the diameter effects negatively on the resistance losses the values obtained from this diameter were far more convincing.

The tests conducted were done using the setup previously shown in section 4.1. The bottom structure distance to the main structure was tested at three different distances: 0 mm (Figure 9), 1 mm (Figure 10), and 1.5 mm (Figure 11). At distances higher than 1.5 mm was observed a phenomenon where the fixed magnet had almost no effect on the magnet's

core due to the screws (that connect the two structures) material. The top structure was tested at four different positions: 0 mm, 2 mm, 3 mm, and without top structure.

Figures 9, 10, and 11 show the RMS voltage values for the different distances of the fixed magnets to the structure of the transducer.

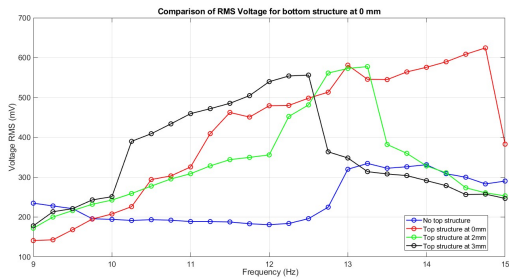


Figure 9: Comparison of the RMS voltage when the bottom structure distance is 0 mm.

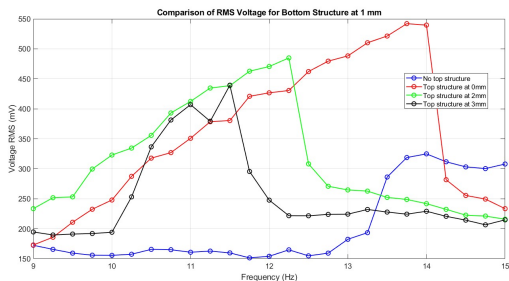


Figure 10: Comparison of the RMS voltage when the bottom structure distance is 1 mm.

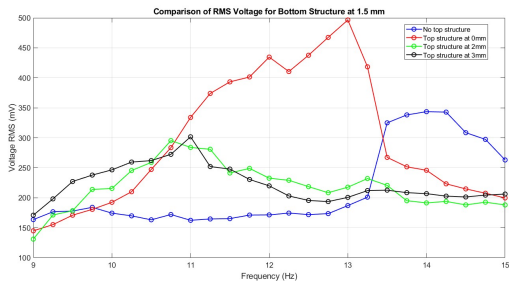


Figure 11: Comparison of the RMS voltage when the bottom structure distance is 1.5 mm.

By analyzing the three figures it's noticed that the top structure has a big effect on the behavior of the system and the output RMS voltage. Higher values of RMS voltage, across all the tested frequencies, are shown when the top structure is at 0 mm (red lines) of the main structure. Also, the output voltage is consistently lower without the top structure (blue lines).

According to the tests is possible to notice the frequency dependence and the damping at higher frequencies. Firstly, the voltage peaks for most configurations in the 12 Hz to 13, 5/14 Hz range, suggesting the energy harvester has a resonance frequency in this range. Secondly, for frequencies below 10 Hz, the system voltage is relatively lower for all distances, meaning that the system is less efficient for lower frequencies. Finally, the figures show us that for higher frequencies (beyond 13, 5/14 Hz) the voltage drops off sharply, suggesting the system is less efficient and less responsive.

According to the results, it can be concluded that the original design, where the fixed magnets are at 0 mm of the main structure, is the best configuration for this electromagnetic energy harvester since it has consistent performance and is effective.

#### 4.4 Load Results

Knowing that the transducer behavior had the best results when it had the fixed magnets in contact with the main structure ( distance of 0 mm at both structures to the main structure) tests were conducted to understand its manners when subjected to a load. For these tests, the programmable resistance shown in section 4.1 was used to simulate a load. The programmable resistance sweep values tested were used in a logarithmic scale from 1 Ohm to 5000000 Ohm, and the frequency sweep was linear between 10 and 20 Hz with a 1 Hz step. These frequencies, as shown in Figures 9, 10, and 11 were where the system had more response. Figure 12 shows the plot comparing the generated power vs load resistance at different frequencies.

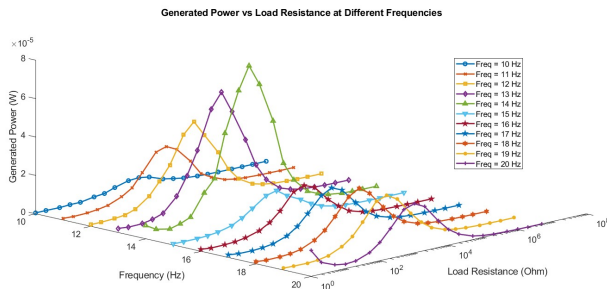


Figure 12: Generated power vs load resistance at different frequencies.

During these tests, the acceleration of the vibration generator was analyzed while operating at 13.96 Hz. The recorded movement of 4 mm at this frequency resulted in  $30774.47 \text{ mm/s}^2$  of acceleration.

A simple test was conducted with the developed energy harvester tied to the heel of the author, Figure 13. As shown in Figure 12 the maximum output generated power is seen when the programmable resistance is set to 1000 Ohm and has high values in the range of 200 to 5000 Ohms. The results of this test are shown in Table 2. The test conducted was made at a slow march pace around 1 second of repetition.



Figure 13: Energy harvesting results in the target embodiment.



Table 2: Results of heel test.

| Output Vpp (mV) | Output RMS (mV) | Resistance (Ohm) | P( $\mu$ W) |
|-----------------|-----------------|------------------|-------------|
| 157.5           | 10.310          | 200              | 0.5315      |
| 372.5           | 25.270          | 500              | 1.2771      |
| 550             | 43.150          | 1000             | 1.8619      |
| 590             | 40.890          | 2000             | 0.8359      |
| 630             | 47.090          | 5000             | 0.4435      |

When tested, the system had a maximum output power of  $83.06 \mu W$  at a frequency of 13.96 Hz. Comparing the results of the load tests, in the heel and the vibration generator, it's evident the big difference in the peak values, suggesting the system is highly resonant, and the stimulus ends up being wideband. For a better analysis of the results shown in Figure 12, the analysis is divided into four different points: **Peak Power Generation at Different Frequencies**, **Frequency Dependence**, **Effect of Load Resistance**, and **Other Considerations**.

4.4.1 Peak Power Generation at Different Frequencies

As shown in Figure 12 the power reaches a peak before declining for each frequency proving the existence of an optimal load resistance for each frequency where maximum power is harvested. The maximum generated power is observed in each frequency when the load resistance reaches the value of 1000 Ohm, and it's notable in load resistances of 500 Ohm and 2000 Ohm which are the values before and after the peak. The behavior is proved in every frequency tested and in the frequency of 13, 96 Hz (the frequency that has the best value) the energy harvested is  $83, 06 \mu W$

4.4.2 Frequency Dependence

Analyzing the dependence of the system on the frequency it's easy to confirm that the frequency of 13.96 Hz is the system's optimal frequency or near it.

- Examining the frequencies above the optimal one, 15 Hz and beyond, the generated power decreases sharply, showing reduced efficiency.
- When looking at the frequencies below 13.96 Hz is notable that the generated power increases significantly, the opposite of higher frequencies, showing that the system has a better behavior and efficiency at lower frequencies and a reduced efficiency at frequencies higher than 14 Hz.

4.4.3 Effect of Load Resistance

The behavior observed in the effect of load resistance is influenced by the variation in voltage and current due to changes in the load resistance. This can be explained based on Ohm's Law and power relationships. As the load resistance changes, it directly affects both the current flowing through the circuit and the voltage across the load.

As the resistance load varies along the test is notable the presence of three different states:

- Short Circuit: At very low resistance (close to 0 Ohms), the system behaves almost like a short circuit. The current is very high, but the voltage across the load is minimal, resulting in low power output. In this case, most of the energy is dissipated as heat in the source or the wires.
- Open Circuit: At very high resistance (approaching infinity), the system behaves like an open circuit. The current is nearly zero because the load impedance is too high, although the voltage across the load is at its maximum. However, since power depends on both voltage and current, and the current is almost zero, the power output becomes negligible.
- Optimal Power Output: Between these two extremes, there is an optimal point where the resistance is neither too low nor too high, and the system delivers maximum power. This point often corresponds to an impedance match, where the load resistance is close to the internal impedance of the source. In energy harvesting systems, this is the key operating region, as the system transfers energy most efficiently at this point.

In summary, the variation in power is a result of changes in voltage and current with the load. At low resistance values, the system behaves like a short circuit, and at high resistance values, it behaves like an open circuit. The optimal operating point occurs when the load resistance allows for maximum power transfer.

#### 4.4.4 Other Considerations

Power generation is highly dependent on the frequency of the system. Frequencies around 13 – 14 Hz tend to generate higher power compared to others, as seen by the sharp peaks in generated power for these frequencies. This could indicate resonance or near-resonant behavior in the system, where the structure is vibrating more efficiently at these frequencies.

For lower resistances (near 1 Ohm), power generation is relatively low across all frequencies. However, as resistance increases, certain frequencies lead to much higher power generation. The decline in power generation at very high resistances may suggest that the circuit impedance is becoming too high, limiting the current flow and thus reducing power.

At very high load resistances (beyond  $10^4$  Ohms), the generated power drops significantly for all frequencies, indicating that the system is not able to drive a significant amount of current at such high resistances.

The best range for generating maximum power appears to be in the mid-range of load resistances (between 100 and 5000 Ohm), especially for frequencies in the range of 12 – 14 Hz.

As shown in the last subsections It can be concluded that the system has satisfactory behavior at frequencies in the range of 12 – 14 Hz, maximizing the output power at 13.96 Hz. Analyzing load resistance values indicates that the harvester performs adequately with load resistances between 100 and 5000 Ohms, maximizing output power at 1000 Ohms.

The values shown may have some errors due to the use of the oscilloscope during the tests, the calculation of the RMS voltage on the oscilloscope is not the best way to do it due to the lack of useful time it was the only setup tested. The values also could have been influenced by the electromagnetic field created by the vibration generator used. Although this possible influences, the results obtained are positive.

## 5 Conclusions

The study's primary goal was to design, implement, and validate a vibration-based energy harvesting system embedded within military footwear. The system aimed to provide a portable power solution for small electronics without the added weight of batteries. The project demonstrated substantial findings on the power generation capabilities of electromagnetic harvesters in response to vibrational frequencies typical of human movement.

Through the integration of electromagnetic transducers within the footwear, the system captured vibrational energy most effectively in the 12–14 Hz frequency range, with an identified peak at approximately 13.96 Hz. This frequency matches common vibrational patterns associated with military personnel's regular motion, such as walking or jogging. The system achieved its highest power output at an optimal load resistance of around 1000 Ohms, with promising results for various load resistances between 100 and 5000 Ohms. These findings indicate that the system is well-suited for wearable applications, where frequency and load variations due to different movement types are inevitable.

However, it was observed that the power output declines significantly when frequencies diverge from the optimal range. This limitation points to a key area for improvement: expanding the system's frequency responsiveness. Future designs that can harness energy from a broader range of frequencies would enhance the harvester's practicality in diverse movement patterns, such as high-speed running or jumping, which occur during military missions.

Compared to traditional piezoelectric or electrostatic energy harvesters, which have been used in similar military applications, the electromagnetic approach in this study demonstrated several advantages. It provided relatively high energy density and robust performance within the identified frequency range without significant degradation from mechanical wear. Piezoelectric systems, though efficient, often require more precise material configurations and tend to degrade over time due to mechanical strain. In contrast, the electromagnetic harvester design, with its magnetic repulsion and limited physical contact, showed promise for durability under repeated use.

The design also performed comparably to recent work on micro-turbine energy harvesters and non-resonant electromagnetic systems, which are generally limited by their requirement for more specific environmental conditions to function effectively. By focusing on vibrations specific to human motion, the current system provides a tailored solution that leverages soldiers' natural movements, enhancing both practicality and accessibility for portable military use.

This study successfully demonstrates the potential of harnessing energy from vibrations, indicating applications beyond military use. Electromagnetic energy harvesting could serve as a portable power source in various civilian industries where reliance on batteries is problematic, such as outdoor sports, disaster response, and remote fieldwork. In these situations, vibration energy harvesters embedded in footwear or backpacks could provide power for communication devices or even GPS devices.

Furthermore, by decreasing reliance on disposable batteries, this technology supports sustainable energy objectives and offers an eco-friendly option for portable energy solutions. Its potential for integration with low-power electronics enhances its value in areas that require self-sustaining energy sources.

In summary, this discussion highlights the effectiveness of the proposed electromagnetic vibration harvester as a lightweight and portable power solution for military applications, while also identifying areas for design improvement. This technology shows promise in reducing reliance on traditional power sources during mission-critical operations, representing a significant advancement in wearable energy-harvesting technology. By addressing the identified limitations, future versions of this harvester could become an invaluable resource for both military and civilian applications, promoting energy autonomy and sustainability in the field.

This research has demonstrated the feasibility of a vibration-based energy harvesting system integrated into military footwear. However, further advancements and explorations could enhance the design and applicability of the system. Future research directions include:

- **Miniaturization:** To facilitate practical deployment, it is essential to minimize the size and weight of the harvester. This will help avoid negatively affecting the user's comfort and mobility;
- **Real-World Testing and Environmental Adaptation:** Future versions should undergo extensive testing in a variety of environmental conditions, including extreme temperatures and high-humidity areas, to evaluate their resilience. Enhancing the design for environmental durability, such as encapsulating components to protect against water and dust ingress, would increase the system's reliability in field operations;
- **Application in Broader Military Equipment:** Beyond wearable systems, this energy harvesting technology could be applied to other military gear that is susceptible to vibrational forces, such as backpacks and vehicles. By harnessing these consistent energy sources, the technology could offer auxiliary power to various operational tools and equipment. This would reduce reliance on traditional power sources and enhance energy autonomy in remote locations.

## References

- Ali, E., Yahaya, N. Z., Nallagownden, P., & Zakariya, M. (2014, 01). Design of rf to dc rectifier at gsm band for energy harvesting applications. *platform - A Journal of Engineering, Science and Society*, 10.
- Bera, B., & Sarkar, M. D. (2016). Piezoelectric effect, piezotronics and piezophototronics: a review. *Imperial Journal of Interdisciplinary Research (IJIR)*, 2(11), 1407–1410.
- Bhatnagar, A., & Lang, D. (2006). Military and law enforcement applications of lightweight ballistic materials. *Lightweight Ballistic Composites*. doi: 10.1201/9781439824436.ch13
- Cunha, H. (2021). *Vibration energy harvesting for lpwan connected devices* (Unpublished doctoral dissertation). Universidade de Lisboa, Instituto Superior Técnico.
- Easley, M. (2022, May). *Army exploring new tech to charge up troops on the go*. Retrieved from <https://www.nationaldefensemagazine.org/articles/2022/5/27/army-exploring-new-tech-to-charge-up-troops-on-the-go>
- Foong, F., Thein, C. K., Ooi, B., & Aziz, A. (2018, 11). A novel design to increase the power output of an electromagnetic vibration energy harvester using finite element analysis. *Journal of Physics: Conference Series*, 1123, 012009. doi: 10.1088/1742-6596/1123/1/012009
- Halim, M. A., Cho, H., Salauddin, M., & Park, J. Y. (2016). A miniaturized electromagnetic vibration energy harvester using flux-guided magnet stacks for human-body-induced motion. *Sensors and Actuators A: Physical*, 249, 23-31. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0924424716303831> doi: <https://doi.org/10.1016/j.sna.2016.08.008>

- Kecik, K., Mitura, A., Lenci, S., & Warminski, J. (2017). Energy harvesting from a magnetic levitation system. *International Journal of Non-Linear Mechanics*, 94, 200-206. Retrieved from <https://www.sciencedirect.com/science/article/pii/S002074621730241X> (A Conspectus of Nonlinear Mechanics: A Tribute to the Oeuvres of Professors G. Rega and F. Vestroni) doi: <https://doi.org/10.1016/j.ijnonlinmec.2017.03.021>
- Khan, F. U., & Qadir, M. U. (2016, sep). State-of-the-art in vibration-based electrostatic energy harvesting. *Journal of Micromechanics and Microengineering*, 26(10), 103001. Retrieved from <https://dx.doi.org/10.1088/0960-1317/26/10/103001> doi: 10.1088/0960-1317/26/10/103001
- Kim, H. S., Kim, J.-H., & Kim, J. (2011, 12). A review of piezoelectric energy harvesting based on vibration. *International Journal of Precision Engineering and Manufacturing*, 12. doi: 10.1007/s12541-011-0151-3
- Kraus, J. (1991). *Electromagnetics*. McGraw-Hill.
- Li, C., Wu, S., Luk, P. C. K., Gu, M., & Jiao, Z. (2019). Enhanced bandwidth nonlinear resonance electromagnetic human motion energy harvester using magnetic springs and ferrofluid. *IEEE/ASME Transactions on Mechatronics*, 24(2), 710-717. doi: 10.1109/TMECH.2019.2898405
- Magnetpt. (2024). *Íman disco 12x3mm com furo 3mm*. <https://magnetpt.pt/iman-disco-12x3mm-com-furo-3mm/>.
- Min-Chie, C., Chang, Y.-C., Long-Jyi, Y., & Chung, C.-H. (2015). Numerical assessment of a one-mass spring-based electromagnetic energy harvester on a vibrating object. *Archives of Acoustics*, 41(1), 119–131.
- Mittal, V. (2020, Oct). *U.s. soldiers' burden of power: More electronics means lugging more batteries*. Forbes Magazine. Retrieved from <https://www.forbes.com/sites/vikrammittal/2020/10/26/energy-management-a-deciding-factor-of-future-battles/?sh=1c1684b12b1a>
- Pinto, E., & Dias, J. (2023, Dec). *Eletronic devices used by portuguese soldiers in missions and the difficulties an energy harvesting system creates to soldiers*.
- Pinto, E., Lima, H., & Sampaio, R. (2022, July). *Desenvolvimento e aplicação de módulo de recolha de dados de aceleração*.
- Pinto, E., & Pires, P. (2024, Apr). *Electronic devices used by portuguese tacp/jtac operators*.
- Priya, S. (2007, 10). Advances in energy harvesting using low profile piezoelectric transducers. *Journal of Electroceramics*, 19, 167-184. doi: 10.1007/s10832-007-9043-4
- Raju, M. (2008). Energy harvesting. ulp meets energy harvesting: A game-changing combination for design engineers. *MCU Strategic Marketing*, Texas Instruments.



# Benchmarking Real-Time Fire and Smoke Segmentation in RGB Images for Wildfire Monitoring



## FIREFRONT BENCHMARK

Autor: **ALF/ENGEL Hugo Lima**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

Orientador: **Prof. Dr. Alexandre Bernardino**  
Instituto Superior Técnico, Lisboa

Coorientador: **MAJ/ENGEL Gonalo Cruz**  
Academia da Força Aérea, Sintra

# Abstract

Forest fires pose a significant threat to human life, property, and the environment, with global annual fire-related mortality reaching 339,000 deaths. This thesis addresses the critical challenge of fire and smoke segmentation through two main contributions: first, we develop an efficient semi-automated annotation pipeline that combines YOLOv9 for object detection and SAM (Segment Anything Model) for segmentation. This pipeline achieves significant improvements over the YOLOv9-seg baseline, with a 4.42% increase in fire segmentation and 31.78% increase in smoke segmentation performance. Through an iterative process of automated labeling and human verification, we successfully annotated a complete private test dataset of aerial imagery and processed 21.04% of the public training dataset, establishing a strong foundation for future expansion; second, we introduce the FIREFRONT benchmark suite, a comprehensive evaluation platform that extends beyond traditional accuracy metrics to assess practical considerations crucial for real-world deployment. The suite features a robust client-server architecture with secure authentication and leverages the ONNX format to ensure broad cross-framework compatibility. We conduct extensive evaluations of five state-of-the-art models (DeepLabV3, DeepLabV3+, HRNet, Swin Transformer, U-Net) across diverse hardware platforms, measuring critical performance metrics including inference speed, memory usage, energy consumption, and segmentation accuracy. By utilizing both high-performance hardware (NVIDIA RTX 4090) and embedded systems (Jetson TX2), the benchmark provides valuable insights into model performance across the full spectrum of deployment scenarios.

**Keywords:** Benchmarking Suite, Dataset Creation, Segmentation Challenge, Fire Segmentation, Smoke Segmentation.

## 1 Introduction

Forest fires pose significant threats to life, property, and the environment, with National Burned Area Composite (NBAC) reporting an average of 2.26 megahectares burned annually in Canada from 2004 to 2016, Hall et al. (2020), and global annual deaths reaching 339,000 due to smoke exposure, Johnston et al. (2012). While traditional fire detection relies on sensors with limited coverage and slow response times, Gaur et al. (2019), video-based detection systems offer faster detection compared to conventional sensors, Cetin et al. (2013). These vision-based models provide advantages in accuracy, error resilience, cost-effectiveness, and coverage, Gaur et al. (2019), particularly when integrated with Unmanned Aerial Vehicle (UAV)s for monitoring vast areas.

Current benchmarks for fire detection models focus primarily on accuracy metrics, overlooking crucial aspects such as memory usage and inference speed—factors essential for real-time applications, especially in embedded systems deployments.

The research objectives focus on consolidating and enhancing existing datasets from FIREFRONT, Ribeiro et al. (2023) and UAV imagery taken by the Portuguese Air Force Squadron "991 - Hárpias". The project will develop both a classical benchmark challenge for algorithmic accuracy and an innovative framework evaluating memory consumption, inference speed, real-time performance on embedded systems.

The primary contribution will be a comprehensive Benchmark Suite enabling evaluation of fire and smoke segmentation models, considering both accuracy and efficiency metrics, alongside an unified, open-source dataset for wildfire monitoring applications. This will establish a more reliable benchmarking tool for real-world applications, including baseline results from state-of-the-art (SOTA) models.

## 2 Background

Image segmentation, particularly for real-time fire and smoke domain, requires a solid understanding of both traditional and modern approaches. This section presents the essential theoretical background needed for comprehending image segmentation techniques and their evaluation.

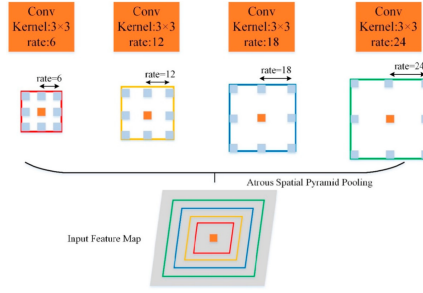


Figure 1: Atrous Spatial Pyramid Pooling (ASPP) architecture. Multiple parallel atrous convolutions with different rates capture multi-scale information. Adapted from Huang and Shen (2023).

## 2.1 Image Segmentation and Techniques

Image segmentation involves partitioning images into multiple segments or objects at pixel level, Bu and Gharajeh (2019), Szeliski (2022). While traditional methods like fuzzy systems and histogram-based approaches laid the ground-work, deep learning, particularly CNNs, has revolutionized the field. Modern architectures like Vision Transformers (ViT), Dosovitskiy et al. (2021) and Masked Autoencoders (MAE), He et al. (2022) have shown remarkable capabilities in feature extraction and image understanding. Of particular interest for real-time applications is the Atrous Spatial Pyramid Pooling (ASPP) technique, which enables multi-scale feature extraction without increasing computational complexity, Chen, Papandreou, Schroff, and Adam (2017).

## 2.2 General Challenges and Problems of Image Segmentation Techniques

Real-time fire and smoke segmentation faces several critical challenges. Image variability, including factors like illumination, contrast, and noise, significantly impacts segmentation quality. This is particularly relevant for wildfire monitoring, where environmental conditions can vary drastically. Parameter selection remains a crucial challenge, as the optimal configuration of segmentation models often requires careful tuning based on the specific application context. The choice of method itself presents another challenge, as different approaches offer varying trade-offs between accuracy, speed, and computational requirements.

## 2.3 Metrics for Segmentation Models

For evaluating segmentation performance, several metrics are essential. The fundamental metrics include Accuracy, Precision, Recall, and F1 Score, based on True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) classifications. For pixel-wise segmentation tasks, Intersection over Union (IoU) and Mean-IoU provide more robust evaluation measures:

$$\text{IoU} = J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^N \text{IoU}_i. \quad (2)$$

The Dice coefficient, particularly relevant for binary segmentation tasks like fire and smoke detection, is defined as:

$$\text{Dice} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} = \text{F1} \quad (3)$$



These metrics provide a comprehensive framework for evaluating the performance of real-time fire and smoke segmentation models.

### 3 State of the Art

Recently, deep learning-based forest fire segmentation methods have shown promising results in delineating fire boundaries. Several studies have demonstrated significant achievements, such as the work by G. Wang, Zhang, Qu, Chen, and Maqsood (2019), achieving 94.2% accuracy for early-stage forest fire identification, and Akhloufi, Tokime, and Ellassady (2018) reaching an F1-Score of 97.09% using a U-Net encoder-decoder architecture.

#### 3.1 General Segmentation Models

The state-of-the-art in segmentation includes several powerful architectures. The Segment Anything Model (SAM) Kirillov et al. (2023) represents a breakthrough in universal image segmentation, achieving 72.8% AP in zero-shot performance on COCO and introducing a massive dataset of 1.1B masks across 11M images. U-Net, Ronneberger, Fischer, and Brox (2015) introduces a symmetric encoder-decoder architecture with skip connections, particularly effective for precise segmentation tasks. DeepLabv3, Chen et al. (2017) and its successor DeepLabv3+ ,Chen, Zhu, Papandreou, Schroff, and Adam (2018) leverage atrous convolutions and encoder-decoder structures, achieving 89.0% mIOU on PASCAL VOC. HR-Net, Sun et al. (2019) maintains high-resolution representations throughout the network by connecting parallel streams of different resolutions. The Swin Transformer, Liu et al. (2021) introduces a hierarchical vision transformer using shifted windows, demonstrating strong performance across various vision tasks while maintaining linear computational complexity.

#### 3.2 Fire and Smoke Datasets

A significant challenge in fire detection research is the limited availability of comprehensive datasets. Notable datasets include FLAME, Shamsoshoara et al. (2021) with 48,010 images, CorsicanFire, Toulouse, T.; Rossi, L.; Campana, A.; Celik, T.; Akhloufi, M.A. (n.d.) containing 1,135 images with binary masks, and FiSmo, Cazzolato et al. (2017) comprising 9,448 images and 158 videos. The FIREFRONT project has developed several specialized datasets, as detailed in Table 1.

Table 1: FIREFRONT datasets summary.

| Name                           | Dataset Composition                                                                  | Labeled | % Label |
|--------------------------------|--------------------------------------------------------------------------------------|---------|---------|
| BA Fire-Smoke Multilabel       | 1807 total images, 322 negatives, 217 smoke only, 233 fire only, 1035 fire and smoke | yes     | 33.20   |
| GP Fire Segmentation Webimages | 180 fire images and 358 hard negatives                                               | yes     | 100     |
| HK Gestosa Fire Segmentation   | 238 fire                                                                             | yes     | 100     |
| LK Fire+Smoke Dataset          | 429 fire and smoke                                                                   | no      | 0       |
| LK Smoke Only Dataset          | 93 smoke only images                                                                 | no      | 0       |
| LK.Smoke Segmentation Dataset  | 96 labeled smoke images                                                              | yes     | 100     |
| SF Bombeiros Fire Dataset      | 1746 fire + 1685 non-fire                                                            | no      | 0       |

#### 3.3 Semantic Segmentation Datasets

The Cityscapes dataset, Cordts et al. (2016) represents a comprehensive benchmark for semantic segmentation, featuring both fine and coarse annotations across 50 different cities. The dataset employs rigorous evaluation metrics, including Intersection over Union (IoU) and instance-level IoU (iloU), providing a standardized framework for comparing segmentation model performance. The PASCAL VOC dataset, Everingham, Van Gool, Williams, Winn, and Zisserman (n.d.) is another influential benchmark, containing 20 object categories across roughly 11,000 images with pixel-level annotations,

and has been instrumental in advancing semantic segmentation research. While not specific to fire detection, these benchmarks' evaluation methodologies and quality control measures offer valuable insights for developing robust segmentation benchmarks.

## 4 Annotation

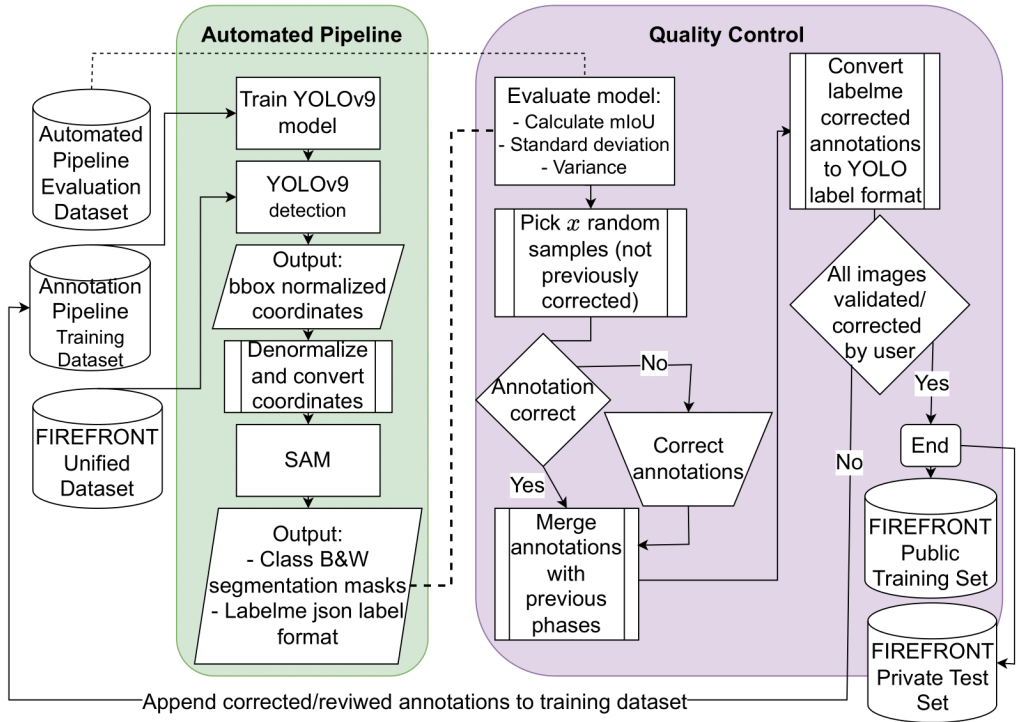


Figure 2: Overview of the annotation pipeline workflow, showing the automated and quality control phases. The pipeline iteratively improves through feedback from human annotators.

### 4.1 Annotation Pipeline Overview

The annotation pipeline consists of two main phases: an Automated Pipeline and a Quality Control phase. The Automated Pipeline utilizes YOLOv9 for initial object detection followed by the Segment Anything Model (SAM) for precise segmentation. The process begins with training YOLOv9 with the *Annotation Pipeline Dataset*, and then performing predictions in the unlabeled *FIREFRONT Unified Dataset*. These predictions, in the form of bounding boxes are prompted into SAM to generate detailed instance masks. The masks are then processed and organized in convenient formats. The output includes both class-specific black and white segmentation masks and annotations in LabelMe JSON format and YOLO format. The Quality Control phase evaluates the quality of these automated annotations through statistical metrics (mIoU, standard deviation). A random selection of unevaluated samples undergoes human review, where incorrect annotations are manually corrected. These corrections are merged with previous annotations in *Annotation Pipeline Dataset*. The pipeline is iterative, with corrected annotations being fed back into the *Annotation Pipeline Dataset* to improve future automated

annotations. The final output is divided into two sets: the FIREFRONT Public Training Set and the FIREFRONT Private Test Set, ensuring a robust evaluation framework for the benchmark suite. A detailed overview of the pipeline is shown in Fig. 2.

Table 2: Datasets Used in the Annotation Pipeline.

| Dataset                               | Description                                                                                                                                                                  |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Automated Pipeline Evaluation Dataset | Combined LK.Smoke.Segmentation and GP.Fire.Segmentation.Webimages datasets. Used to evaluate smoke and fire segmentation performance against human annotations.              |
| Annotation Pipeline Training Dataset  | Training dataset for YOLOv9. Initially used VOCfinaldataset ( $v_0$ ), then switched to quality-controlled annotations from previous pipeline iterations. ( $v_1$ to $v_4$ ) |
| FIREFRONT Unified Dataset             | Unified dataset combining all FIREFRONT datasets (excluding evaluation sets) used as input for YOLOv9 object detection.                                                      |

4.2 Automated Pipeline Phase

The Automated Pipeline phase employs two key models: YOLOv9-E for object detection and SAM for segmentation. YOLOv9-E was selected as our object detector due to its superior performance on the COCO dataset, achieving 54.3% AP. C.-Y. Wang and Liao (2024). Its architectural innovations, including the Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Networks (GELAN), enable more efficient training and better feature extraction capabilities, crucial for detecting fire and smoke in both aerial and ground-level images.

SAM was chosen for the segmentation task due to its zero-shot capabilities and promptable inputs. Having been trained on over 1 billion masks, SAM can effectively segment new objects without specific training, while accepting various prompt types (including bounding boxes) to guide its segmentation process. This flexibility allows us to use YOLOv9’s detections as prompts for precise segmentation.

The pipeline underwent multiple iterations, starting with  $v_0$  which utilized the VOCfinaldataset, CatargiuConstantinVOC-Dataset (2024) for initial training. While this iteration provided a starting point for annotating initial images for the *Annotation Pipeline Training Dataset*, the model struggled with domain-specific challenges as the dataset included non-wildfire related images. Subsequent iterations ( $v_1$ - $v_4$ ) transitioned to using only the corrected annotations from our Quality Control phase, leading to significant improvements in both fire and smoke segmentation. This iterative refinement process allowed the pipeline to become increasingly specialized in wildfire-specific scenarios, with each iteration incorporating more domain-specific knowledge through human-verified annotations.

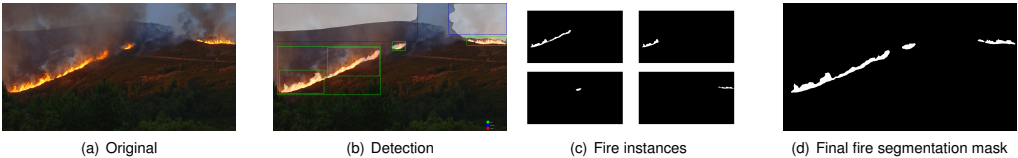


Figure 3: Pipeline visualization: (a) Input image, (b) YOLOv9 detection with Segment Anything Model (SAM) segmentation, (c) Individual fire instance masks, (d) Merged result.

The pipeline generates annotations in both LabelMe JSON format and binary segmentation masks. The LabelMe JSON format was chosen for its compatibility with various annotation tools and its ability to store detailed polygon information for each instance. However, to optimize LabelMe software performance and ease the review process, we implemented

polygon simplification using the Ramer–Douglas–Peucker algorithm with an adaptive epsilon value based on the polygon's perimeter. This optimization reduced the number of points in each polygon while maintaining segmentation accuracy.

### 4.3 Quality Control Phase

The quality control phase implements three distinct benchmarks to evaluate annotation quality. The GP Benchmark evaluates fire segmentation performance using the GP\_Fire\_Segmentation\_Webimages.v1 dataset, while the GP+HH Benchmark combines this with the HH\_Gestosa\_Fire\_Segmentation dataset to assess fire segmentation across different capture scenarios, including aerial footage. The LK Benchmark, utilizing the LK\_Smoke\_Segmentation dataset, Kuhlmann, Niknejad, Barata, Bernardino, and Zhang (2022), specifically evaluates smoke segmentation capabilities. This multi-benchmark approach allows us to evaluate the quality of both fire and smoke annotations separately, while also assessing the model's ability to accurately segment fire from different perspectives, including both ground-level and aerial views.

Following the benchmarking, a random selection of unevaluated samples undergoes human review using the LabelMe annotation software. The random sampling strategy prevents model bias towards specific subdatasets within the FIREFRONT unified dataset. Human annotators review the predictions focusing on polygon boundary accuracy, instance completeness, and correct classification. For smoke instances, following the approach in, Kuhlmann et al. (2022), only unambiguous regions are labeled due to the inherent challenges in smoke segmentation. The corrected annotations are then merged with previous, iterations in the *Annotation Pipeline Dataset* and converted to YOLO format using Labelme2YOLO GreatV (2024), creating an improved training dataset for subsequent iterations.

### 4.4 Results and Discussion

The pipeline's performance enhanced from iteration  $v_1$ , which achieved mIoU values of 0.514 for GP, 0.559 for GP+HH, and 0.500 for LK benchmarks. By iteration  $v_2$ , with nearly doubled annotations (937 vs 482), the performance improved substantially, reaching mIoU values of 0.614, 0.567, and 0.690 respectively. The most significant enhancement was observed in smoke segmentation (LK benchmark), with a 38% increase in mIoU from  $v_1$  to  $v_2$ .

Table 3: Performance metrics across annotation iterations.

| Iteration | #annotations | GP<br>mIoU   | GP+HH<br>mIoU | LK<br>mIoU   |
|-----------|--------------|--------------|---------------|--------------|
| v1        | 482          | 0.514        | 0.559         | 0.500        |
| v2        | 937          | <b>0.614</b> | <b>0.567</b>  | <b>0.690</b> |
| v3        | 1475         | 0.606        | 0.564         | 0.673        |
| v4        | 1675         | <b>0.614</b> | 0.554         | 0.680        |

To validate our approach, we compared the pipeline's performance with the state-of-the-art YOLOv9-seg model trained on the same dataset (1675 images from iteration v4). The results presented in Table 4 demonstrated the superiority of our pipeline.

The superior performance of our pipeline can be attributed to its architectural advantage over YOLOv9-seg. While YOLOv9-seg constrains segmentation within detected bounding boxes, our pipeline uses these boxes only as prompts for SAM, allowing more flexible and accurate segmentation. This is particularly evident in smoke segmentation, where our pipeline achieved a 31.78% higher mIoU compared to YOLOv9-seg. The lower standard deviation in the LK benchmark (0.198 vs 0.283) also indicates more consistent performance in smoke segmentation.

Table 4: Comparison between YOLOv9-seg and Annotation Pipeline.

| Model        | GP mIoU      | GP $\sigma$  | LK mIoU      | LK $\sigma$  |
|--------------|--------------|--------------|--------------|--------------|
| YOLOv9-seg   | 0.588        | <b>0.191</b> | 0.516        | 0.283        |
| Our Pipeline | <b>0.614</b> | 0.195        | <b>0.680</b> | <b>0.198</b> |

## 5 FIREFRONT Benchmark Suite

### 5.1 Open Neural Network Exchange (ONNX)

The FIREFRONT benchmark suite adopts the ONNX format to ensure seamless interoperability across deep learning frameworks and hardware platforms. ONNX enables framework-agnostic model submissions while providing hardware-specific optimizations through ONNX Runtime, Ultralytics (2024). This standardization streamlines model validation and comparison processes, maintaining consistency across submissions and enhancing the benchmark's robustness and adaptability to future developments. All submitted models must be provided in ONNX format to ensure compatibility. Models should accept input tensors of shape  $(1, 3, H, W)$ , where  $H$  and  $W$  represent the input image height and width respectively, with 3 channels for RGB input. The expected output is a tensor of shape  $(1, C, H, W)$ , where  $C$  denotes the number of classes, with each pixel value corresponding to a specific class label. The benchmark validates these specifications to ensure model compatibility prior to evaluation.

### 5.2 System Level Architecture

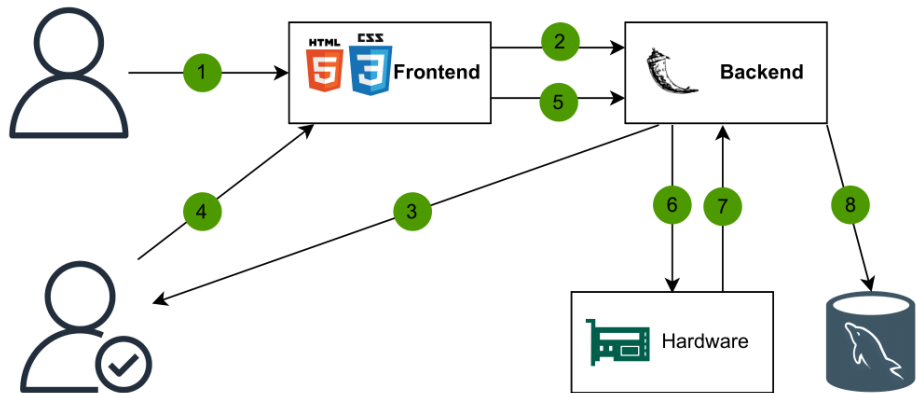


Figure 4: FIREFRONT benchmark system architecture and data flow.

The benchmark employs a client-server architecture with a web frontend and Flask backend. The system utilizes Keycloak for authentication, enabling both traditional and social login methods (Google, GitHub) for enhanced user accessibility. Database management is handled through MySQL, ensuring robust data persistence and efficient query handling. The benchmark workflow, represented in Fig. 4, begins with user authentication through the frontend interface. The frontend then forwards login credentials to the backend, where Keycloak manages secure authentication. Upon successful verification, the backend establishes a secure session and renders the appropriate authenticated user pages. Authenticated users can then submit ONNX models through the frontend interface. Once submitted, the backend validates the model to ensure compatibility with benchmark requirements. If validation succeeds, the backend allocates the model to appropriate hardware (either RTX4090 or Jetson device) for benchmarking. The designated hardware executes the benchmark suite on the model and generates performance results. Finally, the backend stores these results in a MySQL database for persistent storage and future analysis. The architecture implements a state machine that manages model submissions by tracking their status (QUEUED, PROCESSING, RUNNING, COMPLETED, FAILED), ensuring submissions are processed in a First-In-First-Out order and only one model is evaluated at a time. The frontend dynamically informs the user of his submission state. Communication with hardware devices is handled through SSH connections, enabling secure remote command execution and file transfers. Our benchmark platform is composed of a ground station and an embedded system, represented

by the hardware shown in Table 5. This dual-platform approach enables comprehensive evaluation of models under both high-performance and embedded scenarios, crucial for real-world wildfire monitoring applications. The benchmark results are presented through an intuitive web interface, illustrated in Fig. 5, which provides comprehensive visualization of model performance metrics and enables easy comparison across different submissions.

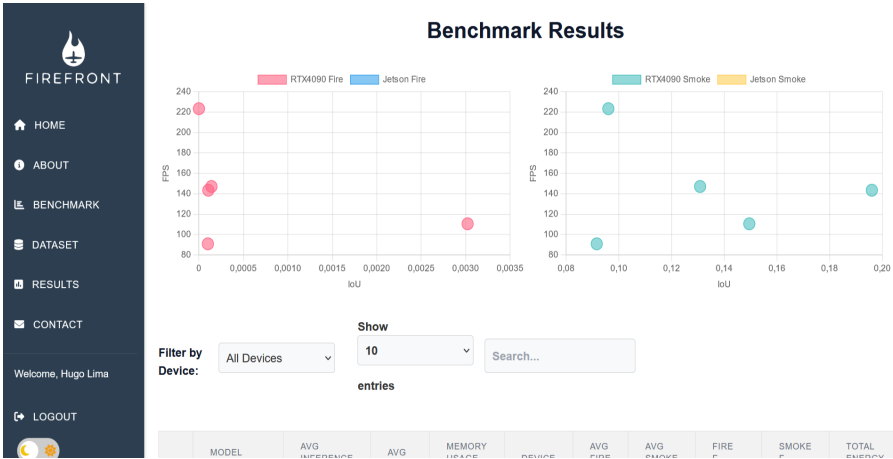


Figure 5: FIREFRONT benchmark results page.

The Jetson TX2 uses two power modes for benchmarking: MAXN and MAXQ. MAXN mode uses maximum computational resources for inference during fire monitoring. MAXQ mode reduces power usage for scenarios like drone operations and battery-powered systems. Testing models in both modes measures the relationship between inference speed and power consumption for deployment planning. The power modes are detailed in Table 6.

Table 5: Especificações de Hardware para as Plataformas de Benchmark.

| Specification | Ground Station  | Embedded System                           |
|---------------|-----------------|-------------------------------------------|
| Platform      | Desktop PC      | NVIDIA Jetson TX2                         |
| GPU           | RTX 4090        | Tegra Parker GPU                          |
| CUDA Cores    | 16,384          | 256                                       |
| Memory        | 24GB GDDR6X     | 8GB LPDDR4x                               |
| Power Modes   | N/A             | MAXN, MAXQ                                |
| CPU Model     | Intel i9-13900K | ARM Cortex-A57 (Quad)<br>+ Denver2 (Dual) |
| CPU Cores     | 24 (8P + 16E)   | 6 (4 + 2)                                 |

Table 6: Modos de Energia da Jetson TX2. \*TDP for MAXN mode is not limited to 15W.

| Mode | A57 Cores | A57 Max (MHz) | GPU Max (MHz) | EMC Max (MHz) | TDP (W) |
|------|-----------|---------------|---------------|---------------|---------|
| MAXN | 4         | 2000          | 1300          | 1866          | 15.0*   |
| MAXQ | 3         | 1200          | 850           | 1331          | 7.5     |

The FIREFRONT benchmark system implements a structured data flow process for model evaluation. The process begins with model processing, where uploaded ONNX models undergo compatibility validation. The system then handles dataset preparation, resizing to match model requirements, normalizing pixel values, and transposing to CHW format. Label processing follows, converting LabelMe JSON annotations to pixel-wise segmentation masks for ground truth comparison. After model inference generates segmentation predictions, the system calculates comprehensive performance metrics including inference time, FPS, IoU scores, precision, recall, F-score, balanced accuracy, FLOPS count, energy consumption, and memory usage. Finally, these results are aggregated and formatted as JSON for backend communication. Fig. 6 illustrates this systematic approach to model evaluation.

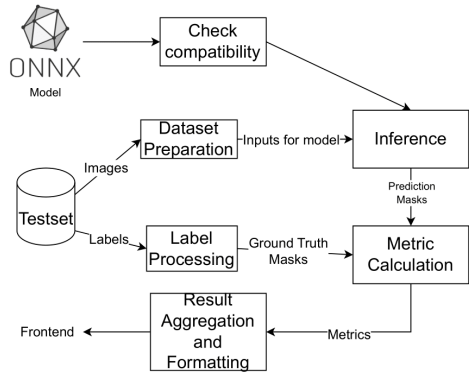


Figure 6: Data flow in the FIREFRONT benchmark system.

### 5.3 Evaluation Metrics

The benchmark evaluates models using comprehensive metrics including inference time, FPS, IoU scores, F-score, balanced accuracy, memory usage, and energy consumption. For performance assessment, we measure average inference time across  $n$  inferences and calculate the corresponding FPS. Prediction quality is evaluated through IoU scores for both fire and smoke classes, F-score (combining precision and recall), and balanced accuracy to account for class imbalance.

Power consumption measurement methodology differs between platforms. For the ground station, we utilize the NVIDIA System Management Interface to sample GPU power consumption at 10-millisecond intervals during model inference. Similarly, on the Jetson TX2 platform, we employ the `tegrastats` utility to collect power measurements at the same sampling rate. The total energy consumption per inference is then calculated by integrating the collected power measurements across the complete inference duration. We apply an analogous approach to monitor and analyze memory utilization on both platforms.

These metrics provide a comprehensive evaluation of model performance, considering both computational efficiency and prediction accuracy, crucial for real-time wildfire monitoring applications. The power measurement methodology ensures accurate energy profiling across different hardware platforms, essential for understanding deployment feasibility in resource-constrained scenarios.

### 5.4 Datasets

The benchmark suite incorporates two complementary datasets: the FIREFRONT Private Dataset for benchmarking purposes and the FIREFRONT Public Dataset for model development. The FIREFRONT Private Dataset serves as the test set, containing 736 aerial images from the FIREFRONT project and the Portuguese Air Force Squadron 991, with all images labeled and verified by two human annotators. These test images range from 800x600 to 4288x2848 pixels, with smoke being the predominant class due to the aerial perspective, making it ideal for standardized performance evaluation. The FIREFRONT Public Dataset comprises 6,206 images, with 1,306 containing annotations, available in both LabelMe and YOLO formats. In contrast to the private dataset, this public dataset contains more instances of fire than smoke, primarily due to its inclusion of more ground-based images. The FIREFRONT Public Dataset is freely available for download through the benchmark suite's frontend by authenticated users, providing researchers with a standardized dataset for developing and evaluating wildfire segmentation models. Additionally, sensitive operational information such as mission coordinates present in the images from the Portuguese Air Force Squadron 991 was redacted and replaced with black rectangles to maintain operational security while preserving the relevant fire and smoke content, as shown in Fig. 7.



Figure 7: Sample images from the FIREFRONT dataset test set. First sample showing the censoring of operational information.

Table 7: Benchmark results for baseline models on NVIDIA RTX 4090 and NVIDIA Jetson TX2.

Table 8: Benchmark results for baseline models on NVIDIA RTX 4090.

| Metric     | DL3           | DL3+          | HR     | Sw            | UN     |
|------------|---------------|---------------|--------|---------------|--------|
| FPS        | 110.53        | 143.42        | 90.91  | <b>223.33</b> | 147.08 |
| Fire IoU   | <b>0.0030</b> | 0.0001        | 0.0001 | 0.0000        | 0.0001 |
| Smoke IoU  | 0.1495        | <b>0.1960</b> | 0.0916 | 0.0960        | 0.1308 |
| Mem (MB)   | <b>1066</b>   | <b>1066</b>   | 1324   | 1318          | 1322   |
| Energy (J) | 0.8131        | 0.4859        | 1.0392 | <b>0.2987</b> | 0.8442 |
| Power (W)  | 89.87         | 69.69         | 94.48  | <b>66.71</b>  | 124.16 |

Table 9: Benchmark results for baseline models on NVIDIA Jetson TX2.

| Metric       | DL3           | DL3+          | HR     | Sw     | UN     |
|--------------|---------------|---------------|--------|--------|--------|
| FPS-N        | 3.39          | <b>9.19</b>   | 6.05   | 4.81   | 4.97   |
| FPS-Q        | 2.30          | <b>6.40</b>   | 4.23   | 3.61   | 3.51   |
| Fire IoU     | <b>0.0030</b> | 0.0001        | 0.0001 | 0.0000 | 0.0002 |
| Smoke IoU    | 0.1495        | <b>0.1959</b> | 0.0918 | 0.0958 | 0.1310 |
| Energy-Q (J) | 1.9904        | <b>0.5897</b> | 0.9680 | 1.0741 | 1.2327 |
| Power-N (W)  | 8.22          | <b>6.46</b>   | 7.04   | 6.49   | 7.68   |
| Power-Q (W)  | 4.58          | <b>3.78</b>   | 4.10   | 3.88   | 4.33   |

## 5.5 Experiments with Baseline Models

The benchmark established baseline performance metrics using five state-of-the-art models: Swin Transformer Liu et al. (2021), HRNet J. Wang et al. (2020), DeepLabv3+ Chen et al. (2018), DeepLabv3 Chen et al. (2017), and U-Net Ronneberger et al. (2015). All models were trained on the FIREFRONT Public Dataset using consistent hyperparameters: learning rate of 0.0001, batch size of 16, and Adam optimizer with ReduceLROnPlateau scheduler. The models were evaluated on both an NVIDIA RTX 4090 GPU and NVIDIA Jetson TX2 embedded system, with the latter tested in both MAXN and MAXQ power modes. Results showed that DeepLabV3+ emerged as the most balanced performer across platforms. On the RTX 4090, it achieved 143.42 FPS with the highest segmentation accuracy (Smoke IoU: 0.1960, Smoke F-score: 0.3278). On the Jetson TX2, it maintained superior performance with 9.19 FPS in MAXN mode and 6.40 FPS in power-efficient MAXQ mode, while requiring the least energy per inference (0.5897J in MAXQ mode). The Swin Transformer excelled in raw processing speed on the RTX 4090, achieving 223.33 FPS with the lowest energy consumption per inference (0.2987J), making it suitable for ground station deployment where processing speed is paramount. These baseline results establish reference points for future model comparisons and highlight the trade-offs between inference speed, accuracy, and energy efficiency across different hardware platforms.

## 6 Conclusions

This thesis presents a comprehensive approach to fire and smoke detection and segmentation through two main contributions: a semi-automated annotation pipeline and a benchmark framework. The annotation pipeline combines YOLOv9 for detection with SAM for segmentation, demonstrating higher mIoU than YOLOv9-seg baseline in both GP and LK bench-



marks. Through this pipeline, we fully annotated the private test set and processed 21.04% of the public training set, as shown in Table 10.

Table 10: Labeling status of FIREFRONT datasets.

| Dataset  | Total | Reviewed | %     |
|----------|-------|----------|-------|
| Training | 6206  | 1305     | 21.04 |
| Test     | 736   | 736      | 100   |
| Total    | 6942  | 2041     | 29.41 |

The research presents two significant contributions to wildfire monitoring: a semi-automated annotation pipeline and a comprehensive benchmark suite. The annotation pipeline, combining YOLOv9 and SAM, successfully processed the entire private test set and 21.04% of the public training set, demonstrating superior mIoU compared to YOLOv9-seg baselines.

The benchmark suite implements a client-server architecture with Keycloak authentication and ONNX compatibility, enabling cross-framework evaluation of five state-of-the-art models: DeepLabV3, DeepLabV3+, HRNet, Swin Transformer, and U-Net. Testing across different hardware platforms revealed DeepLabV3+ as the top performer, achieving a Smoke IoU of 0.1960 and Smoke F-score of 0.3278 while maintaining 143.42 FPS and 69.69W power consumption on the RTX 4090. The Swin Transformer exhibited the highest speed at 223.33 FPS with minimal energy consumption of 0.2987J per inference. On the resource-constrained Jetson TX2, DeepLabV3+ maintained reliable performance at 9.19 FPS in MAXN mode.

The poor fire class performance (best model achieving only 0.003 mIoU) can be attributed to the training set not being representative of the test set - while the training set contains ground-level images, the test set consists of aerial views. Despite this data distribution difference, the benchmark successfully provides comprehensive insights into model performance, balancing accuracy with computational efficiency metrics. These findings establish a robust foundation for evaluating fire and smoke segmentation models, enabling participants to make informed decisions based on their specific accuracy, speed, and energy efficiency requirements in real-time wildfire monitoring applications.

References

Akhloufi, M. A., Tokime, R. B., & Ellassady, H. (2018). Wildland fires detection and segmentation using deep learning. In *Pattern recognition and tracking xxix* (Vol. 10649, pp. 86–97).

Bu, F., & Gharajeh, M. S. (2019). Intelligent and vision-based fire detection systems: A survey. *Image Vis. Comput.*, 91.

CatargiuConstantinVOCDataset. (2024, jan). *Vocfinaldataset dataset* [Open Source Dataset]. <https://universe.roboflow.com/catargiuconstantinvocdataset/vocfinaldataset>. Roboflow. Retrieved from <https://universe.roboflow.com/catargiuconstantinvocdataset/vocfinaldataset> (visited on 2024-05-16)

Cazzolato, M. T., Avalhais, L., Chino, D., Ramos, J. S., de Souza, J. A., Rodrigues-Jr, J. F., & Traina, A. (2017). Fismo: A compilation of datasets from emergency situations for fire and smoke analysis. In *Brazilian symposium on databases-sbdb* (pp. 213–223).

Cetin, A. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboglu, Y. H., ... Verstockt, S. (2013). Video fire detection - review. *Digit. Signal Process.*, 23, 1827-1843.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the european conference on computer vision (eccv)* (pp. 801–818).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3213–3223).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale*.

- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (n.d.). The pascal visual object classes (VOC) challenge. *88*(2), 303–338. Retrieved 2023-11-21, from <http://link.springer.com/10.1007/s11263-009-0275-4> doi: 10.1007/s11263-009-0275-4
- Gaur, A., Singh, A., Kumar, A., Kulkarni, K. S., Lala, S., Kapoor, K., ... Mukhopadhyay, S. C. (2019). Fire sensing technologies: A review. *IEEE Sensors Journal*, *19*(9), 3191–3202. doi: 10.1109/JSEN.2019.2894665
- GreatV. (2024). *Labelme2yolo*. <https://github.com/GreatV/labelme2yolo>. GitHub.
- Hall, R. J., Skakun, R. S., Metsaranta, J. M., Landry, R., Fraser, R. H., Raymond, D., ... Little, J. M. (2020). Generating annual estimates of forest fire disturbance in canada: the national burned area composite. *International Journal of Wildland Fire*, *29*, 878–891.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16000–16009).
- Huang, Y.-F., & Shen, G.-T. (2023). Dual-pyramid wide residual network for semantic segmentation on cross-style datasets. *Preprints*. doi: 10.20944/preprints202310.1610.v1
- Johnston, F. H., Henderson, S. B., Chen, Y., Randerson, J. T., Marlier, M. E., DeFries, R. S., ... Brauer, M. (2012). Estimated global mortality attributable to smoke from landscape fires. *Environmental Health Perspectives*, *120*, 695 – 701.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... Girshick, R. (2023). *Segment anything*.
- Kuhlmann, L., Niknejad, M., Barata, C., Bernardino, A., & Zhang, G. (2022). An adversarial method for semi-supervised segmentation of smoke and fire in images. In *Workshop on image analysis for forest environmental monitoring, icpr 2022*.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).
- Ribeiro, R., Bernardino, A., Cruz, G., Silva, D., Felix, L., Caetano, J., ... Nascimento, J. (2023). Towards the Automation of Wildfire Monitoring with Aerial Vehicles: The FIREFRONT Project. In J.-J. Rousseau & B. Kapralos (Eds.), *Pattern recognition, computer vision, and image processing. icpr 2022 international workshops and challenges* (pp. 183–193). Cham: Springer Nature Switzerland.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention – miccai 2015* (pp. 234–241). Cham: Springer International Publishing.
- Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fulé, P. Z., & Blasch, E. (2021). Aerial imagery pile burn detection using deep learning: The flame dataset. *Computer Networks*, *193*, 108001.
- Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., ... Wang, J. (2019). High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*.
- Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature.
- Toulouse, T.; Rossi, L.; Campana, A.; Celik, T.; Akhloufi, M.A. (n.d.). *CorsicanFire Dataset*. [https://feuxdeforet.universita.corsica/article.php?id\\_art=2133&id\\_rub=572&id\\_menu=0&id\\_cat=0&id\\_site=33&lang=en](https://feuxdeforet.universita.corsica/article.php?id_art=2133&id_rub=572&id_menu=0&id_cat=0&id_site=33&lang=en). ([Accessed 08-01-2024])
- Ultralytics. (2024). *Onnx export for yolo models*. <https://docs.ultralytics.com/integrations/onnx/#onnx-and-onnx-runtime>. (Accessed: 2024-08-06)
- Wang, C.-Y., & Liao, H.-Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*.
- Wang, G., Zhang, Y., Qu, Y., Chen, Y., & Maqsood, H. (2019). Early forest fire region segmentation based on deep learning. In *2019 chinese control and decision conference (ccdc)* (pp. 6237–6241).
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., ... others (2020). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *43*(10), 3349–3364.



# The use of UAV based photogrammetry and BIM in construction supervision. Application to a practical case



Autor: **ALF/ENGAED Maria Catarina Sousa Paulino Libano Monteiro Wemans**  
Mestrado Integrado em Aeronáutica Militar  
Academia da Força Aérea, Sintra

Orientador: **Professor Ana Paula Falcão**  
Instituto Superior Técnico, Lisboa

Coorientador: **Professor João Paulo Janeiro Gomes Ferreira**  
Instituto Superior Técnico, Lisboa

Coorientador: **TCOR/ENGAED João Pedro Teodoro Martins**  
Força Aérea - Direção de Infraestruturas, Alfragide

## **Abstract**

BIM integration during construction and maintenance phases grow at a slower pace than in project design. Specifically, the Portuguese Air Force (PrtAF) could enhance construction supervision and maintenance by adopting advanced technologies for daily supervision. Considering the existing gap between the design and construction phases, this thesis focuses on developing a methodology on worksite monitoring including Unmanned Aerial Vehicle (UAV) photogrammetry and Building Information Modeling (BIM).

The construction phase was explored using a quadcopter drone to follow a construction site during 6 months. Flights were planned exploring Google earth .kml files and the collected images went through photogrammetric software to produce the point clouds. A volume measurement of a stockpile was executed and the point clouds were inserted in the existing BIM model. The structural elements were categorized according to the respective construction phase and non-geometric information was also inserted.

A UAV point cloud was compared with another one from a Terrestrial Laser Scanner (TLS) using specific software and with on-site measurements for validation. For the former, two common stripes between the two point clouds on the top slab were compared, the results being a mean distance between point clouds of 0,049m and 0,091m, a standard deviation of 0,032m and 0,043m and a Root Mean Square Error (RMSE) of 0,058m and 0,101m, for stripes 1 and 2 respectively. For the latter, the top slab perimeter totalled 109,51m for the on-site and 109,43m for the UAV point cloud, corresponding to an absolute error of 8cm and a relative error of 0,073%.

**Keywords:** Building Information Modeling (BIM), Construction Monitoring Automation, Photogrammetry, Point Cloud, Unmanned Aerial Vehicle (UAV)

## **1 Introduction**

Starting from 2030, all architectural projects in Portugal have to be delivered in Building Information Modeling (BIM) methodology (DR, 2024). Despite the current studies for advanced construction supervision (Deliry & Avdan, 2023; El Kas-sis, Ayer, & El Asmar, 2023; Lawani, Sadeghineko, Tong, & Bayraktar, 2024; Ojeda et al., 2024; Yoon & Lee, 2023; Zhang et al., 2023), BIM and process automation are not being implemented as rapidly as in project. This means that while the conception phase is being renewed, worksite supervision continues to rely on traditional methods (Feng & Chen, 2019; Lin, Chang, & Su, 2016; Ochoa, Bravo, Pino, & Rodríguez-Covili, 2011).

By integrating new technologies into its reality, Portuguese Air Force (PrtAF) can benefit from new ways to complement construction supervision and future maintenance in a day-to-day supervision. Even though it is already being implemented in project, BIM tool will not be taking the full advantage of its capabilities if not introduced in other aspects of the infrastructure lifecycle. In addition to that, almost every week there is the need to travel several kilometres to supervise constructing vertical and horizontal structures in air bases located throughout the country. This means spending not only physical and financial resources but also close to 16 hours per week commuting. The use of a UAV with a proficient methodology is the opportunity to work at a distance. It is important to note that, while this makes it simpler to improve the timing or to reduce unnecessary physical visits at times, it does not eliminate essential ones. In that regard, there is a gap in the modernization of construction supervision and its association with BIM tool, as no such correlation exists in that phase within the PrtAF. More specifically, the collected data during supervision is not inserted in BIM models and no UAV is being used to help monitoring or for documentation purposes. This presented an opportunity to develop a methodology where all these potentials for improvement were addressed for better and easier future implementation.

The aim of this work was to develop a methodology to enhance the automation degree for the control and execution of civil engineering works using UAV technology and BIM. It was intended for this study to have a practical approach so, for that purpose, an on-going construction for lodging was used as a case study. Partial objectives were established to achieve the main goal: Optimization and automation of the data acquisition in photogrammetry captured with UAV, namely the flight parameters; Generate appropriate quality point clouds with minimal computational and human effort, facilitating post-processing after each flight; Incorporate point clouds into the BIM model, including the addition of non-geometric information and applying timelines to each construction phase; Supervise the worksite to understand key information for the BIM model and verify point cloud quality for accurate monitoring and documentation.

To achieve that, the construction phase was explored by employing the capabilities of a UAV DJI Mavic 2 Enterprise Advanced to follow a construction site using photogrammetry. A series of flight plans were conducted without an available

flight planning software and the collected images went through the photogrammetric software Autodesk ReCap Student, to produce the point clouds. A volume measurement of a stockpile using a point cloud was conducted for demonstration purposes using the infrastructure design software Autodesk Civil 3D. The point clouds were inserted in the existing Autodesk Revit model according to the phase they were in, along with some basic information such as the concreting date or any important documents. The last resulting point cloud was then compared with the point cloud obtained with a Terrestrial Laser Scanner (TLS) using CloudCompare software and also with on-site measurements for validation.

## 2 Background and Literature Review

### 2.1 Background on worksite supervision, BIM and photogrammetry

Worksite supervision consists in keeping track of the tasks execution, while ensuring the quality throughout the construction process (Halder, Afsari, Chiou, Patrick, & Hamed, 2023). This supervision is done according to the technical specifications document and legal procedures. If one does not meet those requirements, the contractors are informed and corrective measures take action (Feng & Chen, 2019; Halder et al., 2023; Lin et al., 2016). The exchange of information among various participants in construction projects increases significantly when the main contractor outsources services like electrical facilities and plumbing, often relying on paper-based documents such as blueprints and technical sheets for supervision and information storage (Feng & Chen, 2019; Ochoa et al., 2011). This traditional approach presents challenges due to the high number of printed documents that overlap rather than replace each other, complicating access to the information when needed (Eastman, Fisher, & Lafue, 1974). Despite these issues, paper documents remain the primary source of information during the construction phase (Disney, Roupé, Johansson, & Domenico Leto, 2024).

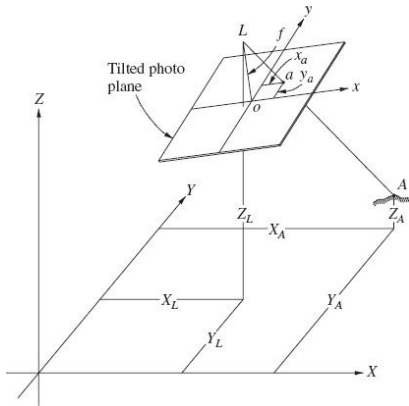
Meanwhile, in 2030 all architectural projects in Portugal are required to be delivered in BIM methodology (DR, 2024). As Autodesk describes it, "BIM integrates structured, multi-disciplinary data to produce a digital representation of an asset across its lifecycle, from planning and design to construction and operations." (Autodesk, 2023). Each BIM model can be categorized into two important groups: geometric and non-geometric (Nguyen & Nguyen, 2024). While the former is related to what people usually refer to as 3D model, including the geometric positioning of construction elements such as slabs, beams, columns, water/sewage/electric network, the latter consists of data that complements these, such as material type, manufacturer, maintenance time, cost, supplier, among others (Nguyen & Nguyen, 2024). BIM also provides interoperability for different teams in the same working environment, combining all types of information that were once scattered all around (Aung, 2022; Autodesk, 2023).

On another note, photogrammetry can be defined as the capability of reconstructing 3D objects, including its location and orientation, by combining pictures and analysing various information through mathematical conditions (DeWitt & Wolf, 2000; Gonçalves, 2005; Kraus, 2011). Depth perception, which can also be described as the missing axis from a plane image, is achieved by two separate vision capable objects that capture the same point from different perspectives. These might be a pair of pictures or even our own set of eyes. The viewed information is then communicated to the program that's processing the images (in our case, it is the brain)(DeWitt & Wolf, 2000; Gonçalves, 2005). The mathematical process to bring 2D images into a 3D world is based on either collinearity or coplanarity equations. Their objective is to align the pictures between one another (DeWitt & Wolf, 2000; Gonçalves, 2005; Temba, 2000). In other words, computing the model without worrying about its logical harmony with the rest of the digital space. That connection is usually done with specific auxiliary equipment either fixed to the imaging device, for instance a real time kinematic (RTK) module, or spread around the imaging site, such as ground control points (GCP), thus allowing for common points to exist between the model and the external coordinate system (DeWitt & Wolf, 2000; Gonçalves, 2005).

Figure 1 is a visual aid to better understand the collinearity equations (1). These are based on the collinear condition, which requires for the object ( $A$ ), its image ( $a$ ) on the photographic coordinate system defined by  $(x, y)$  axis and the principal point (PP) - defined by the lighting sensor of the imaging device - ( $L$ ) to be in the same, straight line. It is important to notice that the object is in a different reference system  $(X, Y, Z)$  from the photo plane.

In the photographic system,  $(x_a, y_a)$  are the coordinates of the object's image.  $O$  is its origin and  $f$  is the focal distance. In the object's system,  $(X_L, Y_L, Z_L)$  are the PP coordinates while the  $(X_A, Y_A, Z_A)$  are the real object A coordinates.

With that in mind, the collinearity equations (1) were developed, where  $(m_{i,j})$  is the rotation matrix. This matrix allows the rotation of one coordinate system to match the other (DeWitt & Wolf, 2000; Gonçalves, 2005). There are three different



$$\begin{cases} x_a = x_0 - f \left( \frac{m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \right) \\ y = y_0 - f \left( \frac{m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \right) \end{cases}$$

(1)

Figure 1: Collinearity condition (Adapted from (DeWitt & Wolf, 2000))

types of transformation from one coordinate system to another, that expand into seven different parameters: Rotation around the three axis (3), translation along the three axis (3) and a scale factor to adjust the model (1) (Gonçalves, 2005).

## 2.2 Literature Review

In this section, recent studies are reviewed to understand how point clouds acquired by photogrammetry and BIM, both individually and combined, are being used worldwide, with an emphasis in construction supervision.

Jones and Church (2020) compared two photogrammetric software programs: Agisoft PhotoScan (now Metashape) and Autodesk ReCap Photo. They analysed Structure-from-Motion (SfM) data captured by a DJI Phantom 4 Pro UAV and compared it to terrestrial photography using a Nikon D5200 DSLR. The study focused on two archaeological sites: Tyler House, photographed solely by the UAV, and Eyrie House, captured only by the DSLR. Multiple datasets were obtained by varying the UAV's flight height and photographing from different positions. Three comparisons were made: Different photosets using the same software; The same photosets with different software and different photosets with different software. Results indicated no significant advantage in using either software when conditions were equal. ReCap processed the models in the cloud, requiring less powerful hardware, while Agisoft Metashape allowed for parameter customization but demanded more sophisticated hardware. Ultimately, the choice of software depended on user experience, comfort, optimization needs, and available resources (Jones & Church, 2020).

The aim of the study Deliry and Avdan (2023) was to evaluate the accuracy of volumetric measurements using photogrammetry compared to traditional methods and LiDAR. The workflow included data acquisition, SfM-based image processing, and accuracy analysis comparison with a TLS. Two UAVs were used for data collection: a fixed-wing Sense Fly eBee Plus for large areas and a rotary-wing DJI Phantom 4 Pro for close-range imaging. Flight parameters included 70-80% forward overlap and 65-70% side overlap for the fixed-wing at altitudes of 100m and 170m AGL, while the rotary-wing had 80% overlap and a height of 16m AGL (Deliry & Avdan, 2023). Photogrammetry software utilized included 3Dsury, Agisoft Metashape, and Pix4Dmapper. Four Ground Control Points were used in a 424,868 m<sup>2</sup> area. Results showed that Metashape produced the best point cloud, with a volume difference of 0.66 m<sup>3</sup> from a reference volume of 2480.77 m<sup>3</sup>, meaning a relative error of 0.027%. The RMSE(xy) and RMSE(z) values were 0.024m and 0.019m, respectively (Deliry & Avdan, 2023).

Pishdad and Onungwa (2024) addressed the challenge of implementing 5D BIM (which integrates cost into the model), in daily applications. To improve efficiency, existing systems for cost estimating, monitoring, control, and cash flow were used. Artificial intelligence (AI) was employed to automatically update costs based on material codes. The study also highlighted that the "Integration of point-cloud with 5D BIM programs will increase cost control during construction and aid

the automation of 5D BIM through the project lifecycle.” (Pishdad & Onungwa, 2024). Combining the as-built point cloud with the 3D model enabled tracking of work progress and payments (Pishdad & Onungwa, 2024).

The authors Ojeda et al. (2024) presented a physical follow-up of construction works using UAV and BIM. The used methodology started with the capture of high-quality images using a DJI Phantom 4 RTK drone. The photographs were then processed by Agisoft Metashape where the point cloud was created. After that, Revit was used to link the point cloud with the phase elements and to Power BI to estimate the physical progress through its dashboards. The construction area in this study is divided in 3 blocks with,  $614.35 \text{ m}^2$ ,  $1056.55 \text{ m}^2$  and  $1315.30 \text{ m}^2$  respectively. The flight plan endured a series of phases from making sure the equipment was properly mounted and batteries were charged to uploading the flight polygon in .KML, defining flying settings such as camera tilt, overlap and waiting for proper satellite connection. All these helped to ensure a safe and efficient 12 minute flight where 848 images were taken (Ojeda et al., 2024). After image processing in Agisoft Metashape, the point cloud consisting of 37.786,411 points was exported to ReCap in .LAS format. Additional quality control was made by eliminating anomalous points and only then it was exported to Revit. That step was also necessary since there was no common format between Metashape and Revit (Ojeda et al., 2024). The object was then modelled and the phase parameters applied, extending it to the 4th BIM dimension.

## 3 Material and Methods

### 3.1 Introduction

The case study is an on-going construction of a lodging building, located in air base nº1, Sintra. It is close to  $750 \text{ m}^2$  of implantation area with around 8 meters high, with 2 floors and 16 rooms per floor, making a total of 32 rooms. It will have a living room and a laundry area, as well as a technical area for the equipment. The building structure and local topography had already been modelled in Revit in the design phase. The structural model was provided by the projects division of the infrastructure department of the PrtAF. It can be seen in figure 2. The used coordinate system was PT-TM06/ETRS89.

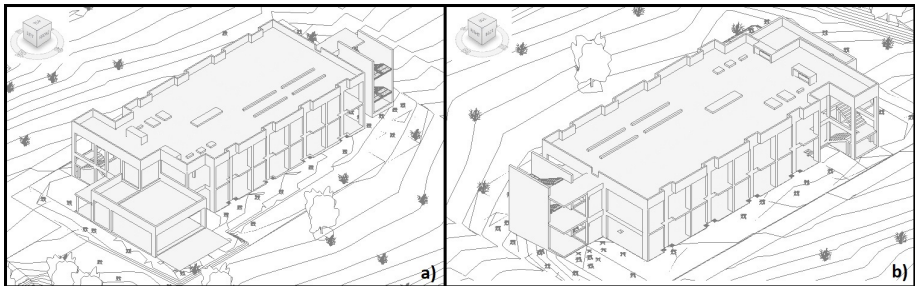


Figure 2: Isometric views from the REVIT model

The crane's location was also delivered in .kml format (used by Google Earth). This is of high importance since the crane's height, surrounding buildings and trees are obstacles to the drone's flight plans.

The implemented methodology was as follows in the flowchart of figure 3.

### 3.2 Data acquisition

The used UAV, provided by *Centro de Investigação da Academia da Força Aérea* (CIAFA), is a quadcopter DJI Mavic 2 Enterprise Advanced. Besides the basic software available by the RC (DJI Pilot), there was no other flight planning tool available. This would later present a problem to the flight planning that was partially solved by a developed python program with the help of ChatGPT and Google Earth's capacity to deal with .kml files. The acceptance of artificial intelligence (AI) in this work seemed only natural since the lack of knowledge in a programming language was standing between a solution



## THE USE OF UAV BASED PHOTOGRAMMETRY AND BIM IN CONSTRUCTION SUPERVISION: APPLICATION TO A PRACTICAL CASE

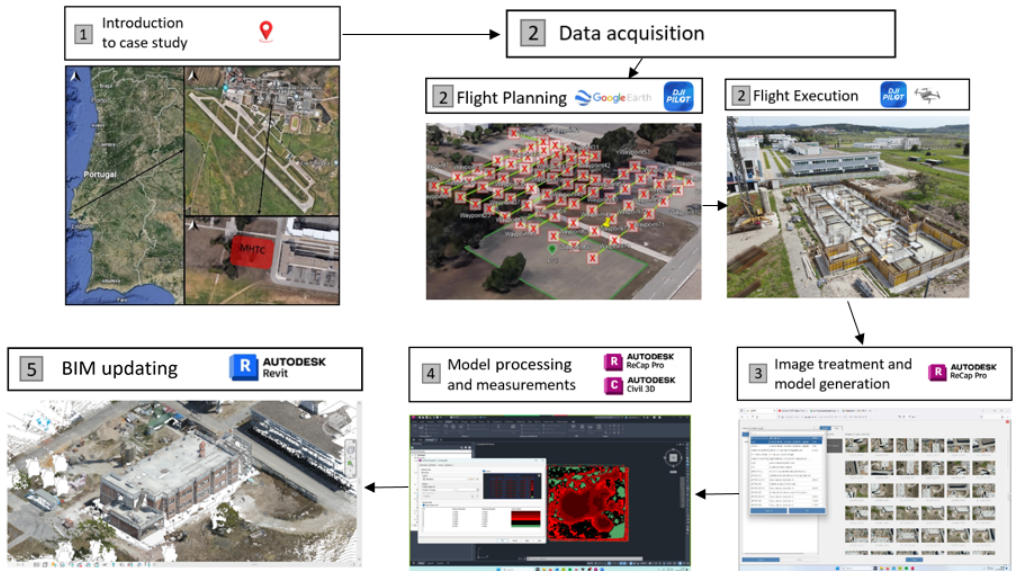


Figure 3: Implemented methodology

and a problem. For the Ground Control Points (GCP) acquisition, both GPS station Leica GS18 and a total station Leica TCSP1201 were used.

The flight planning was an iterative process since most of the improvements were based on experience gained from previous flights. Each flight plan was composed by four settings that could vary from flight to flight:

- Flight path - Designed in Google Earth and responsible for placing the waypoints;
- Altitude - The height in meters AGL of the flight;
- Yaw direction - Referring to the yaw axis (vertical), it could be either "Along the Route" with the drone facing the direction of movement, or "Point of Interest (POI)" if it meant always facing a chosen point while it moved in a different direction;
- GP angle - It could be pointing to the POI, changing its angle at each waypoint, or it could be given a specific, constant angle, such as  $-70^\circ$ , throughout all the waypoints. Its range went from  $30^\circ$  to  $-90^\circ$  (DJI, 2022).

Since no free and compatible solution was available for the duration of the scheduled flights, it was noted that the drone's remote controller (RC) could import and export .kml files. The same format is created by Google Earth when defining a polygon, waypoints or a POI in a given area. That tool was explored and a series of tests were evaluated, importing and exporting the files to and from Google Earth and the controller, to better understand the structure of the .kml format and to make sure that the .kml file would not be corrupted and worked as expected when importing it to the RC. It was then used to plan the flight. Firstly, the safe-to-fly zone was drawn, along with the building's limits. Then, the flight path, considering that each point would later be used to add an action, such as taking a photograph. When the desired path was completed, the final .kml was imported to the RC and the remaining parameters, such as POI, altitude, yaw direction and GP angle were defined there. However, those had to be manually inputted for each point so, for easier implementation of this process, a python program was developed with the assistance of CHATGPT and CIAFA members.

GCP are reference points with accurately known coordinates used in most topographic surveys to improve model accuracy, as shown in Deliry and Avdan (2023) and Jones and Church (2020). With that in mind, four points were surveyed,

spread around the worksite. Appropriate terrain and exposure to daily construction activities were taken into account in its placement.

A total of 10 flights were carried out during the construction period. They were mostly flown during the rising of the structure, since it is when the significant visual change occurs. Table 1 exhibits the flights' summary, along with the number of points of each processed point cloud. It's important to note that this was an iterative process and the results obtained after each flight and respective model generation, impacted the next flight's parameters.

Table 1: Flights' summary

| Flight Nº | Date       | Construction Phase                  | RGB (un.) | Points (un.) |
|-----------|------------|-------------------------------------|-----------|--------------|
| 1         | 2024-03-05 | Foundation pre-concreting           | 111       | 1312265      |
| 2         | 2024-03-06 | Foundation post-concreting (part1)  | 105       | 2282024      |
| 3         | 2024-03-12 | Foundation post-concreting (part2)  | 140       | 6937380      |
| 4         | 2024-03-19 | Semi-basement walls post-concreting | 140       | 7110220      |
| 5         | 2024-04-03 | Slab 0 post-concreting              | 296       | 9233426      |
| 6         | 2024-04-11 | Slab 0 Columns post-concreting      | 296       | 9994424      |
| 7         | 2024-04-23 | Slab 1 post-concreting (part1)      | 149       | 9399125      |
| 8         | 2024-05-03 | Slab 1 post-concreting (part2)      | 166       | 9117035      |
| 9         | 2024-06-27 | Cover slab post-concreting          | 165       | 8297412      |
| 10        | 2024-09-08 | Post-bricklaying                    | 130       | 2991043      |

### 3.3 Image processing and model generation

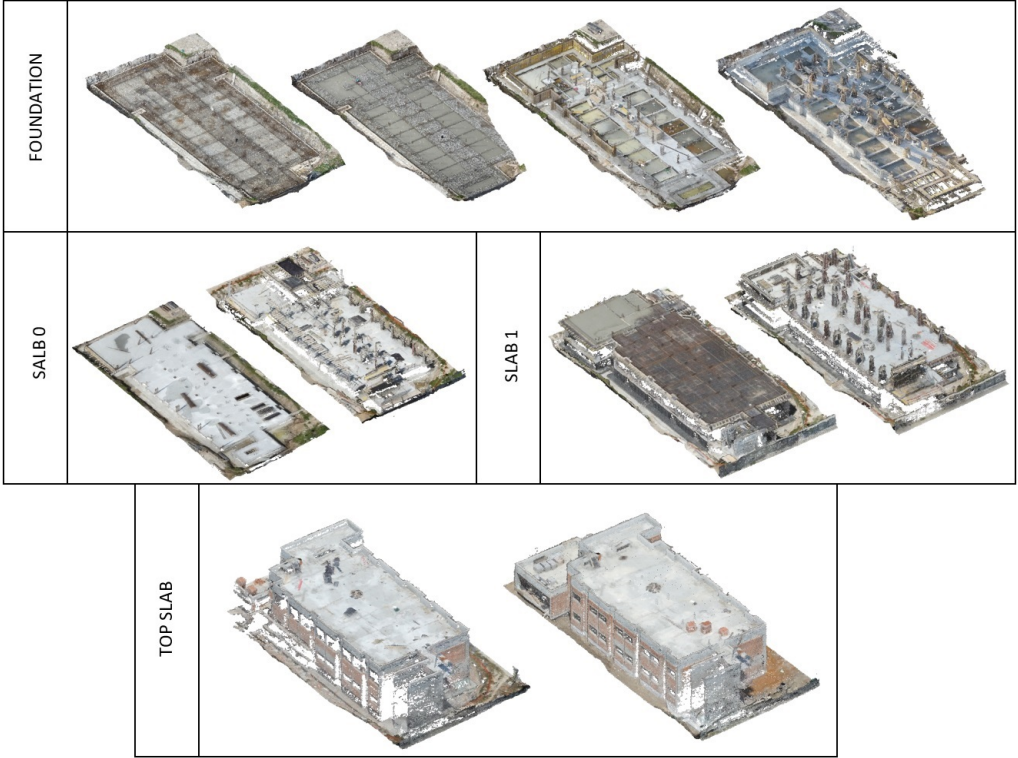
For the image processing, ReCap Pro v.24.1 was chosen for the model generation. Since it was the student version, there was the limitation of 100 photographs per model. So, the first step was to remove pictures that did not catch the building nor the surroundings, or only partially did so, and the ones that looked alike and taken from very slightly different perspectives. The photographs were then imported to the program. The targets were carefully marked in every photograph that had captured them and once the minimum images required by the program were achieved for each GCP, the project was created. The final point cloud was exported in .RCS, with the same coordinate system of the existing Revit model (PT-TM06/ETRS89).

The building area evolution can be seen through the progress of the generated point clouds in figure 4.

The original point clouds shown in figure 4 are larger, as they include the terrain surrounding the building. Capturing this surrounding area is essential for documenting the entire worksite, including nearby stockpiles, which are often important to quantify for land-filling purposes, as shown below.

### 3.4 Model processing and measurements

The stockpiles in the northeast facade were used for land filling purposes. A prior analysis of the volume before its application could be convenient in some cases to estimate how much land filling soil there is and how much more is needed, or for soil transportation cost, for example. For point cloud shaping and volume measurements, ReCap and Civil 3D 2025.0.2 were used, respectively.



For demonstration effects, the stockpile volume located on figure 5a was measured. Firstly, the point cloud was opened in ReCap and the wanted area was selected. The point number was reduced to almost 10% to reduce hardware effort. Then, it was imported to Civil 3D as a .PTS file and the volumes dashboard tool was used for comparing both surfaces. A schedule was created with a color code for a better visual representation of what and where was the excavation/filling, as shown in figure 5b.

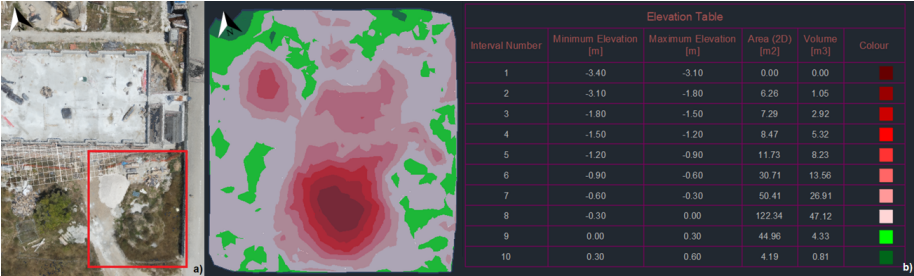


Figure 5: a) Location of the stockpile in the worksite; b) Elevation color code and schedule

The red spectrum represents excavation while the greens represent the opposite. The total excavation volume calculated by the program was  $105,12m^3$ .

### 3.5 BIM updating

In this section, not only the elements were categorized according to the time they were constructed but also non-geometric information was inserted in the model. The BIM software used was Revit 2024.2.0 due to its availability and to the compatibility with the already used Autodesk software.

Element categorization consists in assigning each designed element (slabs, columns, beams, rebars, among others) a specific phase, associated to a timeline, thus making it possible to see when each element was constructed, which elements were constructed at the same time and what elements are still to be constructed. Phasing does not only makes it clearer to see the construction progress, but it is also useful for documentation and, specially, for payment purposes. Work progress records - documents elaborated by the construction supervisor that provide information on the present condition of the worksite - outline the development in terms of achieved tasks (from the constructed concrete elements to inserted water and sewage pipes for instance) and encountered setbacks, also referring to the used equipment. These take into account each individual material, and correspondent quantity increment, from month to month. It is based on these that the payment is processed, when agreed by the interested parties (Vergo, n.d.). One way to automate processes can be to use the point cloud of the as-is worksite status to sort the elements by construction phase, automatically adding up the total for that month's work progress record.

The first step for element categorization was to overlap the point cloud with the Revit model, the final point cloud was inserted into Revit with its positioning being "Auto - By Shared Coordinates". Then, the phases corresponding to each flight were created. Finally, each element was selected and put in the phase they were constructed. To ensure everything was associated with the correct phase, they were carefully examined and verified for compatibility with the actual photographs and point cloud, as shown in figure 6. With phasing process completed, the fourth dimension of BIM was accomplished.

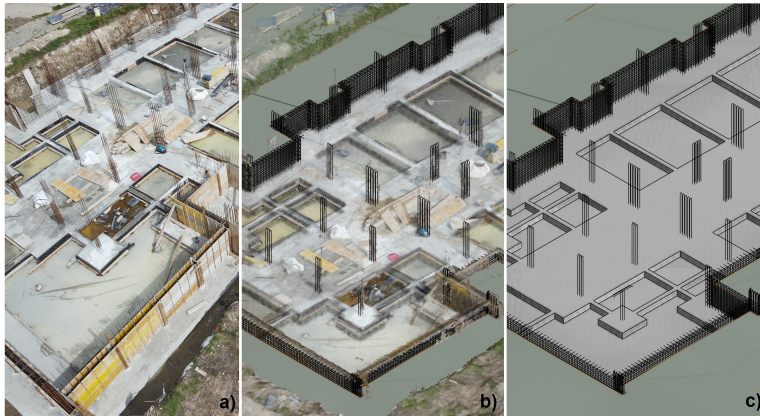


Figure 6: Elements' comparison: a) UAV photograph; b) Revit model overlapped with the point cloud; c) Revit model with the elements in the respective phase. (Flight N°3)

The addition of non-geometric information has proven to be relevant for documentation and maintenance purposes (Morgenstern & Raupach, 2022). The first step was to understand which day-to-day information could be beneficial to include in the model by analysing the worksite supervisors routine. In this regard, there were more than 100 printed documents such as delivery slips, the rebars' labels and the concrete quality sheet (developed by the supervising team), among others. In the construction's end, all these documents are combined in a technical file, which comprises all the technical information for that worksite. So, with the aim of later implementing this process on a daily basis and joining all

information in one database only, for easier and more efficient finding and reading, some of these printed documents and information were scanned and inserted into the model for demonstration purposes.

This was done by creating two attributes for the respective elements: Concreting Date (text format) and Technical Compilation (URL format). The content of the last being the combination of the delivery slips, the concrete quality sheet or any other important document for that particular element such as the FAPs or the technical sheets. As for the attributes' filling, the date was confirmed with the technical sheet and entered by hand. Both technical documents - delivery slips and concrete quality sheet - were scanned to a .pdf format. They were then inserted in a drive that integrates the general repository for the worksite, using this URL ([https://drive.google.com/file/d/1MyOtNiqIKLNWqhO\\_qrHDQc4bUacAFESu/view](https://drive.google.com/file/d/1MyOtNiqIKLNWqhO_qrHDQc4bUacAFESu/view)), which was used to fill the Technical Compilation attribute.

## 4 Results

In this chapter, the obtained point clouds are analysed and tested for accuracy using CloudCompare software. The reference point cloud was obtained through a TLS scan, surveyed on the 24-07-2024. On-site measurements were also executed and compared with the UAV photogrammetric point cloud.

The terrestrial laser scan was conducted in order to have a valid point cloud to compare with the UAV photogrammetry. The equipment used was a Faro Focus S70 laser scanner.

The buildings' last floor and the top slab were captured with four scans. Using Autodesk ReCap software, they were then combined into a single point cloud. No GCP were used for this survey and they proved to be a part of the puzzle missing when comparing to the UAV point cloud. Since the TLS point cloud had no associated coordinate system, the geographic placement of the UAV point cloud could not be evaluated. Nonetheless, the relative placement between the two was assessed.

The chosen UAV point cloud was from flight N°10 because it was considered to be the one that better captured the same elements captured in the TLS scan, as they were not captured in the same day. After reducing the number of points in each point cloud to ease hardware efforts, common ground between the two was selected. A vertical and horizontal stripe on the top slab were then cut, aligned and compared using the cloud-to-cloud distance tool. For stripe 1, the calculated mean distance was 0,049m, the standard deviation was 0,032m, as shown in figure 7. The RMSE value was 0.058m. Stripe 2 developed a mean distance of 0,091m, a standard deviation of 0,043m and an RMSE value of 0.101m.

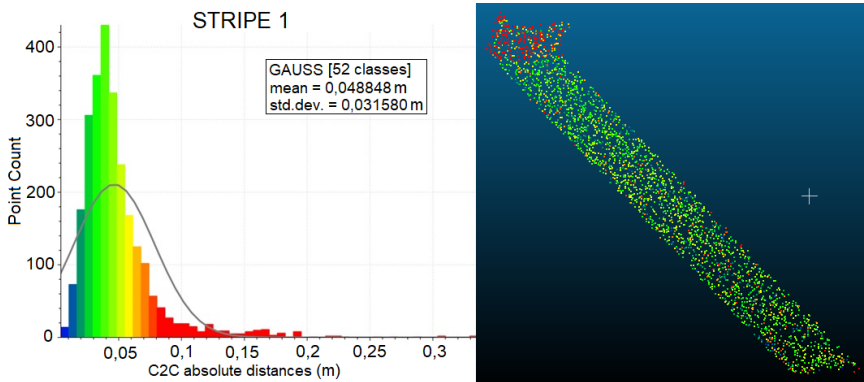


Figure 7: Point cloud alignment statistics for stripe 1

These results were obtained through the scalar field tools and are comparable with the ones obtained in (Deliry & Avdan, 2023) where the highest and lowest mean distance were 0,035m and 0,014m, respectively and the highest and lowest standard deviation factor were 0,011m and 0,005m respectively. It's essential to recognize that the small comparison area and the point-picking alignment between scans can be enhancers for this outcome.

For the on-site measurement comparison, the top slab perimeter was measured with a BOSCH laser tape, totaling 109,51m. The UAV point cloud was sectioned to the top slab and a polyline was then drawn along the inside of the platband, picking points by sight. The total polyline length added up to 109,43m. This translates into an absolute error of 8cm and a relative error of 0,073%.

Along 6 months of the construction, 10 flights were executed using 17 different flight plan variations (flight path, height, GP angle and yaw direction) and 1698 RGB photographs were captured. 10 point clouds were generated and overlapped with the Revit model, phasing it down into 6 different timings for element categorization. In the whole process, 5 different software were manipulated - DJI Pilot, Autodesk ReCap, Autodesk Civil 3D, Autodesk Revit and CloudCompare - and 4 different hardware were used - DJI Mavic 2 Enterprise Advanced, GPS Leica GS18, Total Station Leica TCRP1201, TLS Faro Focus S70. It is also worth mentioning the developed Python program employing AI (ChatGPT) as part of the flight planning solution.

This methodology involved several stakeholders, since it connected a series of thematics, such as flight planning and execution, image processing, model processing, BIM analysis and model, among others. This meant dealing with different software and hardware and exploring interoperability between them.

## 5 Conclusions

In order to conclude, this dissertation allowed to understand the not only the advantages but also the challenges in implementing the proposed methodology. In the data acquisition, not having a flight planning software presented to be a time-consuming challenge. The way it was resolved can be enough for amateur use at no additional cost. However, it is not recommendable for a more professional use as the entire process may require considerable time in comparison to flight planner solutions. The potential of photogrammetry and point cloud documentation are not limited to BIM, as they can also be valuable for on-site monitoring of a long distance worksite or for volume measurement of stockpiles for land filling or transport purposes. In terms of BIM implementation, the 4th dimension was achieved by associating the designed elements to their respective construction phases. This allowed for work documentation and can also be useful for payment purposes by automatically generating cost schedules of the increment in work progress reports. Regarding non-geometric information in the BIM model, it was realized that for an easier implementation a gradual process, i.e., scanning the documents as they appear, would have been the best option. An online folder per element or material allows all the information to be organized in one place only and available for anyone, anywhere at anytime. Finally, in the comparison of the UAV and TLS point cloud, the common ground between both was explored and two stripes of the top slab were evaluated. The results were a mean distance of 0,049mm and 0,091mm for stripe 1 and 2, respectively and a standard deviation of 0,032mm and 0,043mm, respectively. For the on-site measurement, the measured top slab perimeter was 109,51m while the one in the UAV point cloud was 109,43m, meaning an absolute error of 8cm and a relative error of 0,073%.

Future works include filling all the remaining attributes in the model to make it valuable for management and maintenance purposes, to obtain a flight planning software and apply the same methodology exploring the cost schedules for payment purposes using work progress reports, thus exploring the 5th BIM dimension, among others.

## References

- Aung, T. H. S. (2022). *Dimensions in bim*. (URL: <https://www.novatr.com/blog/dimensions-in-bim>)
- Autodesk. (2023). *Bim solutions*. (URL: <https://www.autodesk.com/eu/solutions/bim/benefits-of-bim>)
- Deliry, S. I., & Avdan, U. (2023). Accuracy evaluation of uas photogrammetry and structure from motion in 3d modeling and volumetric calculations. *Journal of Applied Remote Sensing*, 17(2), 024515–024515. (DOI: 10.1117/1.JRS.17.024515)
- DeWitt, B. A., & Wolf, P. R. (2000). *Elements of photogrammetry (with applications in gis)*. McGraw-Hill Higher Education.
- Disney, O., Roupé, M., Johansson, M., & Domenico Leto, A. (2024). Embracing bim in its totality: a total bim case study. *Smart and Sustainable Built Environment*, 13(3), 512–531. (DOI: 10.1108/SASBE-06-2022-0124)
- DJI. (2022). *Specifications for mavic 2 enterprise advanced*. (URL: <https://enterprise.dji.com/mavic-2-enterprise-advanced/specs>)



- DR. (2024). *Decreto-lei n.º 10/2024, artigo 17, 1.* Diário da República. (URL: <https://diariodarepublica.pt/dr/detalhe/decreto-lei/10-2024-836222484>)
- Eastman, C., Fisher, D., & Lafue, G. (1974). *An outline of the building description system.* (URL: <https://files.eric.ed.gov/fulltext/ED113833.pdf>)
- El Kassis, R., Ayer, S. K., & El Asmar, M. (2023). Augmented reality applications for synchronized communication in construction: a review of challenges and opportunities. *Applied Sciences*, 13(13), 7614. (DOI: 10.3390/app13137614)
- Feng, C. W., & Chen, C. W. (2019). Using bim and mr to improve the process of job site construction and inspection. *WIT Transactions on the Built Environment*, 192, 21–32. (DOI: 10.2495/BIM190031)
- Gonçalves, G. R. (2005). Elementos de fotogrametria analítica. *Departamento de Matemática.* (URL: <https://www.mat.uc.pt/gil/downloads/fotogrametria0506.pdf>)
- Halder, S., Afsari, K., Chiou, E., Patrick, R., & Hamed, K. A. (2023). Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study. *Journal of Building Engineering*, 65, 105814. (DOI: 10.1016/j.jobte.2022.105814)
- Jones, C. A., & Church, E. (2020). Photogrammetry is for everyone: Structure-from-motion software user experiences in archaeology. *Journal of Archaeological Science: Reports*, 30, 102261. (DOI: 10.1016/j.jasrep.2020.102261)
- Kraus, K. (2011). *Photogrammetry: geometry from images and laser scans.* Walter de Gruyter.
- Lawani, K., Sadeghineko, F., Tong, M., & Bayraktar, M. (2024). Methodology for retrospectively developing a bim model from point cloud scans using ongoing building project as case study. *Journal of Engineering, Design and Technology.* (DOI: 10.1108/JEDT-07-2023-0325)
- Lin, Y. C., Chang, J. X., & Su, Y. C. (2016). Developing construction defect management system using bim technology in quality inspection. *Journal of civil engineering and management*, 22(7), 903–914. (DOI: 10.3846/13923730.2014.928362)
- Morgenstern, H., & Raupach, M. (2022). Bim-centred building diagnoses as a decision support tool for maintenance and repair. *e-Journal of Nondestructive Testing*, 10, 27287. (DOI: 10.58286/27287)
- Nguyen, T. A., & Nguyen, T. A. (2024). Building information modeling (bim) for construction project schedule management: A review. *Engineering, Technology & Applied Science Research*, 14(2), 13133–13142. (DOI: 10.48084/etasr.6834)
- Ochoa, S. F., Bravo, G., Pino, J. A., & Rodríguez-Covili, J. (2011). Coordinating loosely-coupled work in construction inspection activities. *Group Decision and Negotiation*, 20, 39–56. (DOI: 10.1007/s10726-010-9203-9)
- Ojeda, J. M. P., Huatangari, L. Q., Calderon, B. A. C., Tineo, J. L. P., Panca, C. Z. A., & Pino, M. E. M. (2024). Estimation of the physical progress of work using uav and bim in construction projects. *Civil Engineering Journal*, 10(02). (DOI: 10.28991/CEJ-2024-010-02-02)
- Pishdad, P., & Onungwa, I. O. (2024). Analysis of 5d bim for cost estimation, cost control, and payments. *Journal of Information Technology in Construction (ITcon)*, 29(24), 525–548. (DOI: 10.36680/j.itcon.2024.024)
- Temba, P. (2000). Fundamentos da fotogrametria. *Belo Horizonte: UFMG*, 22–31. (URL: [https://www.faed.udesc.br/arquivos/id\\_submenu/815/fotogrametria\\_material.01.pdf](https://www.faed.udesc.br/arquivos/id_submenu/815/fotogrametria_material.01.pdf))
- Vergo. (n.d.). *Work progress report.* (URL: <https://www.getvergo.com/define/work-progress-report>)
- Yoon, J. W., & Lee, S. H. (2023). Development of a construction-site work support system using bim-marker-based augmented reality. *Sustainability*, 15(4), 3222. (DOI: 10.3390/su15043222)
- Zhang, J., Yang, X., Wang, W., Guan, J., Ding, L., & Lee, V. C. (2023). Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering. *Automation in Construction*, 146, 104699. (DOI: 10.1016/j.autcon.2022.104699)



Grupo do Mercado - 2715-0211 Pílo Pílo  
Tel: 215 678 562