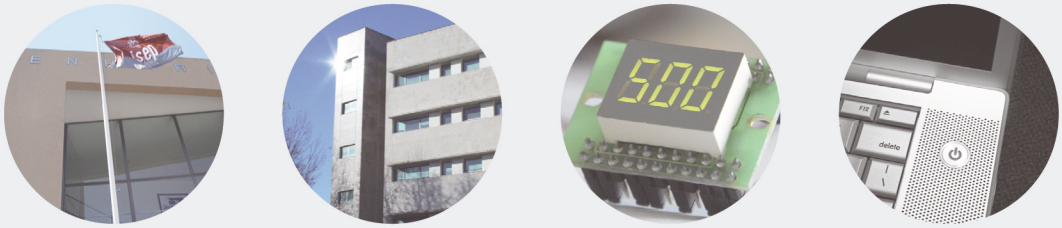
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



Desenvolvimento de métodos de inspeção de ativos elétricos recorrendo a um UAV VTOL

RUI FILIPE LOUREIRO MENDES
novembro de 2020

POLITÉCNICO
DO PORTO

Desenvolvimento de métodos de inspeção de ativos elétricos recorrendo a um UAV VTOL

Rui Filipe Loureiro Mendes
Nº 1150779

Master in Electrical and Computer Engineering
Branch of Autonomous Systems
Electrical Engineering Department

Instituto Superior de Engenharia do Porto

2020

Dissertation for partial satisfaction of the requirements of the Master in Electrical and Computer Engineering

Candidate: Rui Filipe Loureiro Mendes
Nº 1150779

Supervisor: Dr. André Miguel Pinheiro Dias
apd@isep.ipp.pt

Master in Electrical and Computer Engineering
Branch of Autonomous Systems
Electrical Engineering Department

Instituto Superior de Engenharia do Porto

17 de Novembro de 2020

Agradecimentos

Para a conclusão do presente projeto foi necessário muito esforço e dedicação, de modo a chegar a resultados aceitáveis. Muitas teorias foram avaliadas e por vezes a frustração veio ao de cima. Assim sendo, gostaria de agradecer a todos os abaixo mencionados, por de uma maneira ou outra, terem-me ajudado a superar as adversidades que foram encontradas ao longo deste projeto.

Ao Professor Doutor André Miguel Pinheiro Dias, pelo desafio proposto, por todas as dicas e propostas de solução que me deu ao longo do projeto e de todo o meu percurso do mestrado. Foi com este incentivo todo que considero que cresci como aluno e como indivíduo.

À Diana Salgado, que desde o início do mestrado foi sempre a minha colega de grupo. Fazemos uma grande equipa e estamos juntos no desenvolvimento do *Vertical take-off and landing aircraft* (VTOL) desde o início. A Diana ajudou-me em vários momentos do projeto, maioritariamente nos momentos difíceis do mesmo, dando-me motivação e algumas possíveis causas dos meus problemas do projeto.

A todos os que fazem ou já fizeram parte do LSA, estou grato por tudo o que me ensinaram, pela ajuda que me forneceram e pelos momentos de convívio, no entanto, cabe um especial agradecimento ao Tiago Santos e Alexandre Oliveira, pelas ajudas em problemas pontuais no desenvolvimento e por terem facilitado os testes efetuados.

Por último, a toda a minha família pelo carinho e incentivo que me deu e em especial à minha tia Julieta Loureiro, que desde sempre me incentivou e investiu nos meus estudos académicos, inequivocamente se não fosse ela, eu nunca teria chegado ao presente nível académico.

Obrigado.

Rui Filipe Loureiro Mendes

Esta página foi intencionalmente deixada em branco.

Abstract

The Human Being increasingly depends on electricity for everyday life. The availability of electricity therefore becomes a key factor for any population development. For this, it is necessary that the network is constantly maintained in good condition, through constant inspections, ensuring less failures at any point in the energy transmission chain.

Using *Unmanned Aerial Vehicle*s (UAVs) it is possible to mitigate current inspection problems, which are the cost of inspection and its associated risk. Equipping a multirotor with relevant sensors and designing inspection maneuvers can solve many of these problems. However, the problem of low aircraft autonomy is raised.

A *Vertical take-off and landing aircraft* (VTOL) consists of a vehicle with characteristics of multirotor and fixed wing, it seeks to exploit the advantages of each, thus mitigating the problem of flight autonomy. A VTOL solution was developed to respond to these problems, potentially capable of inspecting large extensions of high and medium voltage power lines. This solution was also designed to operate autonomously, reducing the need for human intervention.

In addition to the developed aircraft, new inspection maneuvers were also investigated and applied to transmission lines. These maneuvers were designed to take advantage of the benefits of a VTOL. Using both conventional multirotor inspection methods and fixed wing inspection maneuvers, it was possible to obtain promising results for future applications.

The final result consists of several simulations of the implemented maneuvers and data collection that justify the use of a VTOL for large-scale inspections, all in an autonomous way. Finally, the prototype of an aircraft developed with the characteristics of a VTOL is also presented.

Keywords: UAV, VTOL, Power lines, Path planning

Esta página foi intencionalmente deixada em branco.

Resumo

O Ser Humano está cada vez mais depende de eletricidade para o seu dia a dia. A abrangência da eletricidade torna-se um fator fulcral para qualquer desenvolvimento populacional. Para isto, é necessário que a rede seja mantida em boas condições operacionais, através de inspeções constantes, não podendo existir falhas em qualquer ponto da cadeia de transmissão de energia.

Com recurso a UAVs é possível mitigar os problemas de inspeção atuais, sendo estes o custo de inspeção ou o risco da mesma. Equipando um *multirotor* com sensores relevantes e projetando manobras de inspeção pode resolver muitos dos problemas das mesmas. No entanto, é levantado o problema da baixa autonomia das aeronaves.

Um VTOL híbrido consiste num veículo com características de *multirotor* e *fixed wing*, e procura explorar vantagens de cada um, mitigando assim o problema da autonomia de voo. Foi desenvolvida uma solução VTOL para responder a estes problemas, potencialmente capaz de inspecionar grandes extensões de linhas de alta e média tensão. Esta solução também foi pensada para operar de forma autónoma, reduzindo a necessidade de intervenção Humana.

A par da aeronave desenvolvida, foram também investigadas e aplicadas novas manobras de inspeção a linhas de transmissão. Estas manobras foram projetadas de modo a tirar partido das vantagens de um VTOL. Recorrendo tanto a métodos convencionais de inspeção com *multirotor* e a manobras de inspeção de *fixed wing*, foi possível obter resultados promissores para aplicações futuras.

O resultado final consiste na simulação de possíveis manobras implementadas e no levantamento de dados que justificam o uso de um VTOL para inspeções em grande escala, tudo de forma autónoma. Por fim, também é apresentado o protótipo de uma aeronave desenvolvida com as características de um VTOL.

Palavras-Chave: UAV, VTOL, Linhas elétricas, planeamento de trajetórias

Esta página foi intencionalmente deixada em branco.

Conteúdo

Lista de Figuras

Esta página foi intencionalmente deixada em branco.

Lista de Tabelas

Esta página foi intencionalmente deixada em branco.

Lista de Algoritmos

Esta página foi intencionalmente deixada em branco.

Lista de Acrónimos

2D Bidimensional

3D Tridimensional

BVLOS *Beyond Visual Line Of Sight*

BVS *Boundary Value Solver*

CRAS *Centre for Robotics and Autonomous Systems*

ENU *East-North-Up*

ESC *Electronic Speed Controller*

ETH *Swiss Federal Institute of Technology in Zurich*

FOV *Field-Of-View*

GNSS *Global Navigation Satellite System*

IMU *Inertial Measurement Unit*

INESC TEC Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência

ISEP Instituto Superior de Engenharia do Porto

LASMU Laboratório de Sistemas Multirobóticos

LiDAR *Light Detection And Ranging*

LiPo *Lithium polymer battery*

LKH *Lin-Kernighan-Helsgaun Heuristic*

LSA Laboratório de Sistemas Autónomos

MTOW *Maximum Takeoff Weight*

P2P *Peer-To-Peer*

PID *Proportional Integral Derivative*

RRT *Rapidly-exploring Random Tree*

ROS *Robotic Operating System*

RWR Rolling on Wires Robot

SITL *Software In The Loop*

STL *Standard Triangle Language*

TSP *Travelling salesman problem*

UAV *Unmanned Aerial Vehicle*

VTOL *Vertical take-off and landing aircraft*

Capítulo 1

Introdução

Com o passar dos tempos, o Ser Humano está cada vez mais está dependente da tecnologia, desde equipamentos considerados imprescindíveis no dia a dia, como os telemóveis, à simples electricidade presente nas suas casas. Foi através dos avanços tecnológicos que tarefas rotineiras passaram a ser automatizadas, bem como as tarefas perigosas, que seguiram o mesmo caminho.

Nas últimas décadas, parte do desenvolvimento tem sido focado no uso de *Unmanned Aerial Vehicle*s (UAVs) para várias tarefas. A utilização dos UAVs, também vulgarmente denominados de *drones*, tem cada vez mais entrado no quotidiano do Ser Humano. Atualmente, os UAVs já executam tarefas de busca e salvamento [25], auxiliando bombeiros na procura de vítimas com necessidade de auxílio. Os médicos recorrem aos UAVs para transporte de equipamentos médicos com urgência de tempo [26]. O uso da tecnologia é portanto essencial no quotidiano humano, existe uma necessidade constante de manter o seu desenvolvimento e de o dotar de capacidade de planeamento de missões de forma autónoma, tendo por base os objetivos pretendidos e constrangimentos inerentes ao ambiente como obstáculos.

Apesar de toda a tecnologia a ser desenvolvida, algumas tarefas ainda requerem grande intervenção humana, por vezes com tarefas perigosas, como o caso de inspeções de postes de alta tensão [27]. Estas inspeções são feitas ou a pé, percorrendo vários quilómetros de distância entre postes e subindo aos mesmos, ou de helicóptero, sobrevoando a baixa altitude as extensões de linhas de alta tensão. Ambas as abordagens são bastante perigosas, dispendiosas e levam bastante tempo a realizar.

Atualmente existem soluções de inspeção autónomas [28] que abrigam soluções a alguns dos problemas, como por exemplo os Rolling on Wires Robots (RWRs). Estes são robôs que são colocados nas linhas de alta tensão e percorrem a distância das mesmas,

enfrentando os problemas de passar obstáculos como bolas de sinalização e junções de isoladores. Os RWRs também necessitam de intervenção humana para os colocar nas linhas, não deixando assim de ser uma tarefa que acarreta riscos.

Figura 1.1: Exemplo de um RWR [1]

Outro tipo de robôs vulgarmente utilizados neste tipo de inspeção são os *multirotors*. Dadas as suas características de voo holonómicas, ou seja, a capacidade de mudar de velocidade em qualquer direção instantaneamente, estes tornam-se muito versáteis no processo de inspeção. Alguns problemas que estes robôs enfrentam são a deteção de obstáculos, pelo que requer a utilização de sensores como câmaras de espectro visível ou laser, como é o caso dos *Light Detection And Ranging* (LiDAR), que permitem obter a reconstrução Tridimensional (3D). Estes métodos são bastante vulneráveis a ruído de fundo, uma vez que a maioria dos ambientes onde estão colocadas as linhas de transmissão são perto de vegetação. Além disso, a autonomia dos aparelhos é reduzida, comprometendo assim a inspeção, uma vez que se pretende inspecionar vários postes.

Figura 1.2: *Multirotor* Byrd II num ambiente de inspeção com vegetação [2]

As inspeções das linhas de transmissão devem ser efetuadas de forma contínua, rigorosa e eficiente, de modo a evitar acidentes. A falta de inspeção dos limites de vegetação, estado das linhas e outros fatores, podem culminar em incidentes tais como o incêndio de Pedrogão [29, 30].

Figura 1.3: Exemplo de um possível acidente resultante da falta de inspeção [3]

Tendo por base os desafios anteriormente enumerados em relação ao processo de inspeção de ativos elétricos, a dissertação irá abordar o desenvolvimento de um UAV híbrido entre *multirotor* e avião (Figura 1.4). Este deve ser capaz de autonomamente inspecionar um ativo elétrico, em particular postes de média e alta tensão. A dissertação irá incidir na otimização do processo de inspeção e na avaliação de diferentes estratégias de inspeção, no sentido de procurar maximizar as potencialidades associadas a um *Vertical take-off and landing aircraft* (VTOL).

(a) *Multirotor* STORK UAV [31] (b) *Fixed wing* (FALCOS) [32]

Figura 1.4: Exemplos de *multirotores* (esquerda) e aviões (direita)

Um UAV híbrido, VTOL, consiste na mistura entre um *multirotor* e um avião, aproveitando as vantagens de cada um. Um avião tem a vantagem de atingir longos tempos de voo comparativamente a um *multirotor*, tendo como desvantagem a falta de mobilidade, uma vez que é um veículo não holonómico. Já o *multirotor* apresenta uma elevada taxa de manobra, no entanto, os seus tempos de voo são muito reduzidos. O conceito de híbrido consiste na existência de uma aeronave capaz de voar tanto como um *multirotor* como um *fixed wing*. A acoplação de motores de *multirotor* numa *frame* de avião é um exemplo clássico de um VTOL híbrido, podendo assim voar com as duas configurações ou numa mistura das duas. O controlo do modo híbrido é um tópico ainda pouco explorado na comunidade científica, apresentando grandes desafios de controlo.

Figura 1.5: Exemplo de um UAV híbrido, retirado de [4]

1.1 Enquadramento e motivação

No decorrer dos últimos anos, o Laboratório de Sistemas Autónomos (LSA)[1] e o Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC)[2] têm trabalhado conjuntamente, apresentando soluções robóticas inovadoras, com o intuito de responder a problemas existentes na comunidade. Entre as soluções anteriores surgiram algumas iterações de UAVs que são capazes de efetuar uma inspeção de um poste de alta tensão [33] [34].

[1] `http://lsa.isep.ipp.pt/`
[2] `https://www.inesctec.pt/en`

Apesar da elevada complexidade da manobra existente, esta ainda não está otimizada, sendo que a sua inspeção consiste numa série de pontos predefinidos de forma a ter ângulos de visão diferentes para um ponto de interesse. Além disso, atualmente a manobra é feita com um *multirotor*.

Com a elevada procura no mercado de distribuição de energia elétrica, a inspeção torna-se vital, sendo bastante rentável efetuar a mesma com um *multirotor*. No entanto, tendo em conta o elevado número de postes que requerem inspeção, surge a necessidade de melhorar a eficiência da inspeção, assim foi introduzido o conceito de UAV híbrido que será abordado no decorrer da dissertação.

1.2 Objetivos

Esta dissertação aborda a inspeção de linhas de alta tensão, recorrendo a um VTOL. Com o objetivo de melhorar a eficiência da inspeção será utilizado um VTOL, retirando as vantagens dos *multirotor* e dos *fixed wing*. O objetivo desta dissertação é desenvolver uma solução que efetue um *takeoff* vertical, percorra as linhas em modo avião até ao ativo elétrico, transite para híbrido ou *multirotor* e efetue a inspeção, tudo isto de forma autónoma.

Para alcançar estes objetivos, algumas metas intermediárias terão de ser alcançadas:

- Estudo de sistemas semelhantes e as suas aplicações;
- Análise de soluções de planeamento de trajetórias aplicáveis no caso;
- Implementação de manobras de controlo para inspeção com um VTOL;
- Avaliação e aplicação de um *firmware* para *Autopilots*: Ardupilot [35] ou PX4 [36];
- Implementação de voo autónomo;
- Efetuar a implementação de diferentes estratégias de inspeção com um VTOL e avaliar os seus resultados.

1.3 Estrutura do relatório

No capítulo 1 é elaborada uma pequena introdução ao projeto, recorrendo a uma breve contextualização, levantamento dos motivos pelos quais o projeto foi realizado e uma descrição dos objetivos definidos para o projeto.

No capítulo seguinte, 2, é apresentado o estado de arte do projeto e efetuado um levantamento de ideais e soluções similares ao projeto em desenvolvimento com o objetivo de ajudar na especificação dos requisitos a impor na fase de projeto.

De seguida, no capítulo 3 serão abordados conceitos e fundamentos necessários para a compreensão desta dissertação.

No capítulo 4 é abordada em pormenor uma *framework* de inspeção de estruturas, desenvolvida pela universidade de *Swiss Federal Institute of Technology in Zurich* (ETH). É também discutida a possibilidade de aplicação ao problema da dissertação.

O capítulo seguinte, 5, apresenta as diferentes soluções encontradas para o projeto, o material escolhido para a construção do protótipo do VTOL e a arquitetura detalhada do sistema. Também são apresentadas e detalhadas algumas manobras de inspeção de linhas de transmissão de energia.

No capítulo 6, é detalhada toda a implementação dos aspetos abordados no capítulo anterior.

Os resultados obtidos são apresentados no capítulo 7, onde o algoritmo de inspeção e de controlo do voo é testado em simulação. São evidenciados dados relevantes de comparação das várias manobras e influência do vento e peso na aeronave. Também é justificado o uso de um VTOL para a aplicação.

Por último, no capítulo 8 são apresentadas algumas conclusões sobre o trabalho desenvolvido, bem como o trabalho futuro a realizar.

Capítulo 2

Estado de arte

Neste capítulo serão abordados alguns conceitos para contextualização do uso de VTOL na indústria do consumo. Posteriormente serão apresentados alguns desenvolvimentos da comunidade cientifica da área de planeamento de trajetórias para inspeção e controlo de veículos híbridos. Por fim, serão discutidos alguns tópicos relevantes abordados no capítulo.

As aplicações de um VTOL podem ser bastante diversas, desde operações de inspeção, missões de busca e salvamento, a monitorização de grandes áreas florestais. O facto de o VTOL não necessitar de uma pista para levantar e aterrar, podendo voar tanto em asa fixa como *multirotor*, tirando vantagens de ambos os modos de funcionamento, são factores cruciais para o seu uso. Para melhor compreensão do que é um VTOL também se deve evidenciar o que é um sistema de UAV autónomo. A operação de um UAV requer a combinação de 3 partes:

- O veículo em si, podendo ser *multirotor*, *fixed wing*, helicóptero ou VTOL
- Uma estação de controlo terrestre, vulgarmente chamada de *ground station*
- Um meio de comunicação entre a *ground station* e o UAV, denominado de telemetria bidirecional.

Um VTOL pode também assumir várias configurações, (Figura 2.1), estudos como os de [15] evidenciam esses aspetos, concluindo que podem existir pelo menos 4 tipos diferentes de VTOL:

- *Standard VTOL*, composto pela adaptação convencional de motores de um *multirotor* a uma estrutura de asa fixa, resultando em pelo menos 4 motores a ser

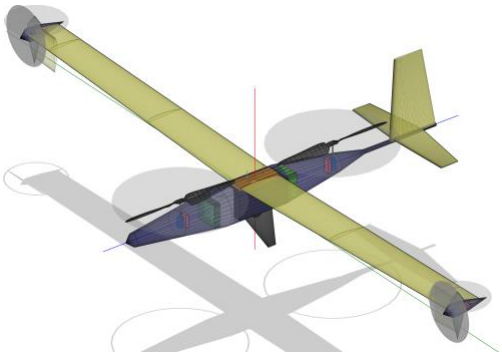
utilizados.

- *Tilt Rotor*, que consiste no uso de apenas 4 motores de *multirotor*, rodando os motores para voo em asa fixa.
- *Tail sitter*, consistindo numa estrutura convencional de asa fixa bi-motora. O lançamento desta estrutura é feito na vertical, sendo esta apoiada pela "cauda"no solo.
- *5TOL*, que consiste numa mistura entre *Standard VTOL* e *Tilt rotor*, esta estrutura foi alvo de investigação em [15].

(a) *Standard VTOL*

(b) *Tilt Rotor*, retirado de [37]

(c) *Tail sitter*, adaptado de [38]



(d) *5TOL*, retirado de [15]

Figura 2.1: Tipos de estruturas de VTOL

2.1 Diabetes Drone - *WingcopterUAV*

O *Diabetes Drone* [26,39,40] foi um UAV desenvolvido para a entrega de medicamentos, nomeadamente insulina, a doentes crónicos situados em zonas rurais.

Este foi um projeto desenvolvido por um professor irlandês, Derek O'Keefe, preocupado com a dificuldade da obtenção de medicação de doentes crónicos localizados nas ilhas mais afastadas da Irlanda, em caso de tempestade. Nestas situações, seria necessária a existência de um plano que conseguisse garantir a chegada da medicação necessária a estes doentes, evitando assim a espera de diversos dias para que fosse possível a chegada de ajuda a essas mesmas ilhas.

Para dar resposta a este problema, com a ajuda de especialistas na área da robótica e na área de medicina, foi criado o primeiro protótipo, com voo *Beyond Visual Line Of Sight* (BVLOS), para a entrega de insulina em Inis Mór, a maior ilha irlandesa. No seu retorno, deveria trazer consigo amostras sanguíneas dos pacientes à qual a medicação era destinada.

De modo a garantir a segurança do procedimento, e tendo em conta o possível embate do aparelho com aviões, foi colocado a bordo uma câmara, onde era possível ver as imagens dos locais por onde o drone se encontraria a passar. Para além disso, foi colocado um sistema de telemetria que permitia o *tracking* da aeronave durante todo o voo.

A aeronave utilizada neste projeto foi um Wingcopter 178 Heavy Lift, com 178 cm de envergadura e *payload* de 6 kg. Para o transporte da insulina foi desenhada uma caixa específica para tal, acoplada na parte inferior da aeronave.

Figura 2.2: Diabetes drone - *WingcopterUAV*

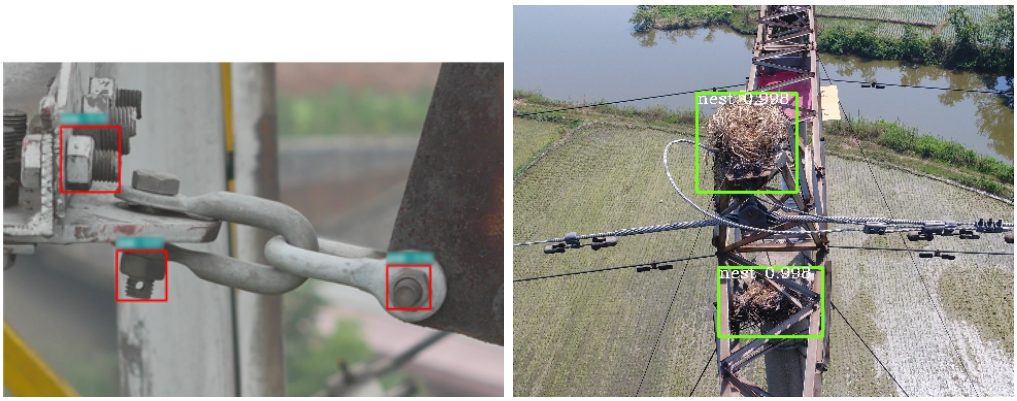
Em relação ao seu voo, foi efetuado com sucesso. A aeronave descolou perto do aeroporto de Connemara com direção a Inis Mór, num voo pré programado recorrendo à aplicação *QGroundControl* [41] para monitorização. A aeronave voou aproximadamente 21.7 km entre estes dois pontos, só na viagem de ida, em 32 minutos. Uma bateria carregada foi suficiente para a viagem de ida e volta desta aeronave entre os pontos considerados.

2.2 Métodos de inspeção com recurso a UAVs

A inspeção de estruturas com recurso a UAVs não é algo inovador na industria. Já existem várias empresas com projectos para o efeito, bem como grandes desenvolvimentos na comunidade cientifica. As inspeções autónomas podem ser efectuadas de várias formas, desde recurso a deteção de *features* [42], recorrendo a câmaras, a planeamento de trajetórias pré-definidas. Diversos veículos aéreos podem ser utilizados, tais como um helicóptero autónomo [43], *multirotors* [14] e ainda *fixed wing*. Tal como evidenciado anteriormente, os VTOL possuem características interessantes para a inspeção, sendo que já começam a existir estudos para os requisitos de inspeção com esses veículos [44].

2.2.1 Inspeção de ativos elétricos, Terra Drone

A *Terra Drone* [5] é uma empresa que se foca no desenvolvimento de soluções industriais com recurso a UAVs. Esta desenvolveu algoritmos com deteção de *features* de pilares das linhas de alta tensão, identificando componentes cruciais para a inspeção.



(a) Deteção de corrosão (b) Deteção de elementos naturais

Figura 2.3: Deteção de features com um UAV [5, 6]

Através de técnicas de *deep learning* é possível identificar pontos relevantes à inspeção, podendo ser em seguida processados algoritmos de planeamento de trajetórias, de modo a inspecionar os pontos de interesse.

2.2.2 Inspeção com recurso a *multirotors* em Portugal

A EDP Labelec é a empresa responsável pela inspeção dos ativos elétricos em Portugal. Desde 1994 que a empresa inspeciona ativos elétricos com recurso a helicóptero, no entanto, com o desenvolver da tecnologia, adoptou-se o uso de *multirotors* [45].

Atualmente em Portugal existem métodos de inspeção das linhas de alta e média tensão com recurso a *multirotors*. As aeronaves de inspeção estão dotadas de câmaras termográficas, para a deteção de pontos quentes nos isoladores e de um LiDAR para vários propósitos, entre eles a construção de uma nuvem de pontos. Além destes métodos, os *multirotors* ainda são capazes de efetuar uma inspeção ultravioleta, para detetar o efeito coroa e de efetuar uma inspeção visual, de modo a detetar anomalias nos órgãos elétricos.

Figura 2.4: *Multirotor* utilizado para inspeção de ativos elétricos em Portugal

2.2.3 Planeamento do caminho de inspeção

Os algoritmos de planeamento de caminho consistem na obtenção de um caminho entre dois pontos, passando por pontos intermédios ou não, sem colisões com objetos no meio. No que toca a inspeções, existem várias abordagens de planeamento de caminho.

Algumas abordagens mais conservadoras defendem que apenas se deve sobrevoar uma linha de transmissão, não efetuando aproximações inferiores a uma determinada

distância. Outras abordagens [46], ainda que simples, consistem em fazer ziguezague sobre uma área, recolhendo dados importantes com recurso a sensores.

Porém, com o aumento da complexidade de cada torre, estes métodos tornam-se insuficientes, resultando assim na necessidade de planeamentos mais complexos. Alguns estudos [7] sugerem uma manobra que passa pelas laterais e pela parte superior de uma torre de transmissão, evitando os cabos de tensão, tal como sugere a figura 2.5. No capítulo 4 é ainda abordado um algoritmo de planeamento cujo o objetivo é essencialmente cobertura de área de forma eficiente.

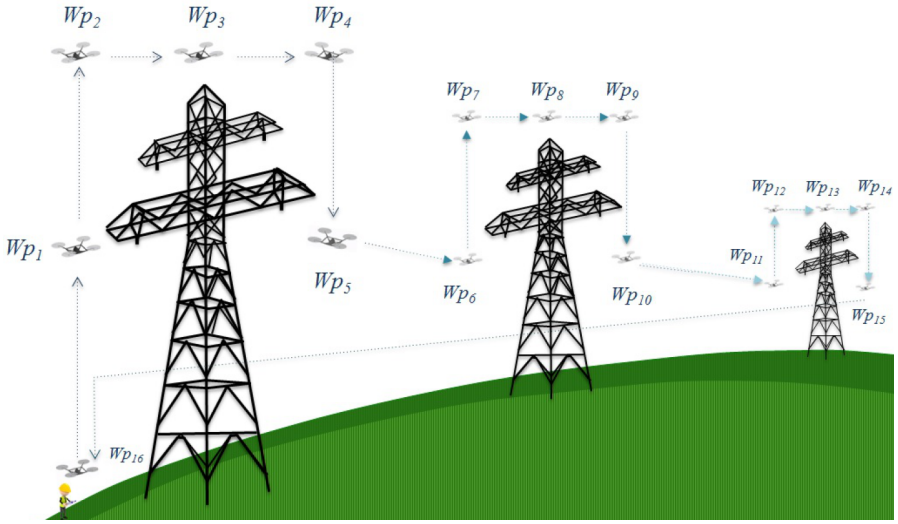


Figura 2.5: Exemplo de um caminho de inspeção, retirado de [7]

2.3 Modos de operação de um VTOL

O tema de estudo dos modos de operação de um VTOL é algo que a comunidade cientifica já aborda [47–49]. Dada a natureza de um VTOL, existem vários modos de voo possíveis, sendo estes:

- Modo de *multirotor*, no qual é aplicado o controlo convencional para o mesmo.
- Modo de *fixed wing*, de igual forma, aplicado o controlo convencional para asa fixa.
- Modo de transição de *multirotor* para *fixed wing*, também denominado de *front transition*
- Modo de transição de *fixed wing* para *multirotor*, ou *back transition*

Entre estes, os métodos de controlo críticos são os que respeitam às operações de transição uma vez que é nestes estados em que a estabilidade do sistema pode ficar comprometida. O controlo nas transições segue uma combinação entre o controlo de *multirotor* e *fixed wing*. Segundo o descrito em [47], a fase mais instável é a que diz respeito às transições. Nesse estudo foram feitos alguns testes nos quais se efetua alterações do peso relativo de cada controlador (*multirotor* e *fixed wing*) de modo a evidenciar o método mais estável para esta fase.

Um dos principais problemas apresentados no artigo foi a manutenção da atitude do UAV. Com o aumento da velocidade, a contribuição das superfícies de controlo de asa fixa também aumenta, sendo que esta terá de ser compensada, reduzindo o efeito do *multirotor*.

Em [47] são apresentados também os resultados referentes ao controlo em *yaw*, este é o mais complexo uma vez que o eixo é afetado também pela rotação do UAV. Foram efetuados 2 testes, um deles consistiu em manter o controlo integral de *multirotor* e *fixed wing* simultâneamente na fase de transição. O segundo teste consistiu em aplicar, na mesma fase, apenas o controlo de asa fixa. Os resultados foram de um erro de 7° para os dois controladores em simultâneo e 3° aquando da utilização apenas do controlador de asa fixa. Perante estes resultados, constatou-se que manter apenas o controlador de *fixed wing* nas transições era mais vantajoso por introduzir menos erro.

Por último foram aplicados os resultados anteriores em testes de transição. Os controladores de *roll* e *pitch* mantém-se com um peso entre *multirotor* e *fixed wing* mediante a velocidade, e o controlo de *yaw* é feito através do controlador de *fixed wing* a partir do momento de transição.

Os testes de voo iniciais mostraram-se promissores, apenas com um desvio da atitude em *pitch*. Após um ajuste nos controladores de *pitch* o voo mostrou-se estável, mesmo nas transições, concluindo assim o melhor método de controlo nas transições.

2.4 Discussão do estado de arte

De todos os tipos de VTOL apresentados, o *standard VTOL* é o mais indicado para o processo de inspeção de ativos elétricos. No caso de um *tilt rotor*, a aplicação de um *propeller* que sirva para os dois modos de voo principais é complexa, sendo que os mecanismos de controlo de inclinação dos motores também poderiam resultar em perda de estabilidade em caso de falha. Para além dos aspetos mencionados, uma vantagem do *standard VTOL* é que em caso de falha de um dos motores, pode entrar no modo de voo complementar, sendo possível na maioria das vezes efetuar uma aterragem segura.

O mesmo não se reflete no *tilt rotor*, uma vez que em caso de falha dos motores, o mais provável é uma queda do UAV.

No que toca ao desenho da estrutura, a solução *Standard VTOL* permite um processo de transporte e de assemblagem rápido. Também é de salientar que no caso de necessidade de reparação a mesma poderá ser efetuada sem grandes requisitos de desmontagem. O *payload* apresentado pelo *Diabetes Drone* é relevante para a aplicação, uma vez que pode ser aplicado a sensores como câmaras ou LiDAR.

Quanto às inspeções, o conceito de obtenção de pontos de interesse através da deteção de features é muito relevante para a aplicação. No entanto, dados os avanços tecnológicos da industria de distribuição energética, os mecanismos de planeamento de caminhos devem ser efetuados de forma cuidadosa, uma vez que as estruturas atuais apresentam um elevado grau de complexidade. É de salientar a importância da conjugação dos diferentes modos de voo, ou seja, aquando a inspeção, utilizar o modo de multirotor, mas em viagens entre torres é importante o uso do *fixed wing*.

Por fim, os métodos de controlo apresentados são considerados válidos, no entanto podem diferenciar conforme as estruturas em que foram implementadas, sendo que será necessário realizar testes adicionais com a estrutura a utilizar para as inspeções.

Capítulo 3

Fundamentos teóricos

Neste capítulo serão abordados alguns conceitos teóricos necessários para o planeamento e desenvolvimento do projeto. Os tópicos abordados contém informação sobre o funcionamento de *frameworks*, explicações de sistemas de coordenadas e esclarecimentos de algoritmos relacionados com o projeto.

3.1 *Robotic Operating System* (ROS)

O ROS [9, 50, 51] é uma *framework open-source* que tem como propósito aplicações robótica. Uma das principais vantagens do ROS é a facilidade com que se faz a partilha de pacotes com facilidade de integração em sistemas individuais. Cada pacote pode conter vários nós. Esta ferramenta foi rapidamente adotada pela comunidade cientifica de robótica pela sua facilidade de uso e vantagens que propõe.

Figura 3.1: Logótipo do ROS, retirado de [8]

A partilha de pacotes pode ser feita através de repositórios online, como o *github* ou por qualquer outro método de transferência de ficheiros. A comunicação em ROS funciona pelo principio de *Peer-To-Peer* (P2P), através de publicadores, subscritores e serviços. Como demonstra a Figura 3.2, um publicador em ROS emite um tópico que pode ser obtido por qualquer subscritor que aceda a sua informação. Um tópico

pode ser acedido por vários subscritores ao mesmo tempo e pode ser atualizado por vários publicadores. Os serviços são funções que são utilizadas recorrentemente, sendo que podem ser chamadas por qualquer nó. Um nó ao enviar informação a um serviço também espera pela resposta do mesmo. A estrutura conta também com mensagens personalizadas, predefinidas ou criadas por utilizadores.

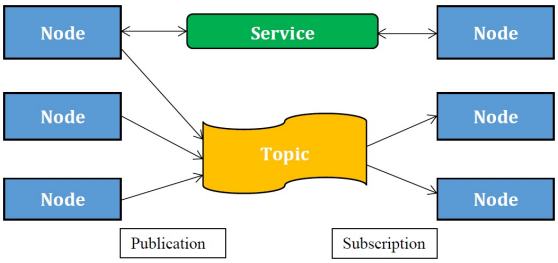


Figura 3.2: Funcionamento dos nós em ROS, retirado de [9]

Todos os elementos anteriores são controlados pelo *master*, tal como demonstra a Figura 3.3. Este é um nó especial, inicializado cada vez que o ROS é aberto num sistema. É responsável pelo controlo de registos dos restantes nós, pelo controlo de subscrições e ligações entre nós. Sem o *master* a rede de ligações não funcionaria, fazendo com que cada mensagem enviada não chegasse ao seu destino, uma vez que as mesmas não chegariam ao respetivo nó por não existir registo.

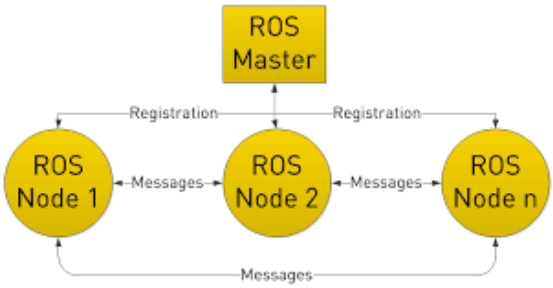


Figura 3.3: Função do nó *master*, retirado de [9]

Para além de suportar várias linguagens de programação, tais como *C++* e *Python*, o ROS também suporta várias ferramentas de simulação robótica, visualização e de *debug*. Os pacotes de ROS podem interagir diretamente com o *hardware* presente num sistema ou podem conter informação *high level*. Um pacote pode conter *drivers* para leitura de um sensor específico, conectando diretamente o sensor ao dispositivo com ROS. O

nó pode também processar informação de vários sensores, aplicar filtros matemáticos e debitar resultados relevantes.

3.2 Conceitos gerais de robótica aérea

3.2.1 *Mavlink*

O *MAVLink* é um protocolo de comunicação com estruturas de mensagens específicas entre o controlador de voo (*autopilot*) e a *Ground station* (computador do utilizador). Através destas mensagens é possível definir *waypoints*, modos de voo, verificar estados de sensores e respetivos dados e por fim, verificar o estado geral da aeronave. A Figura 3.4 demonstra um exemplo de mensagem do *Mavlink*, correspondente ao controlo de *waypoints*.

MAV_CMD_NAV_WAYPOINT (16)

[Command] Navigate to waypoint.

Param (:Label)	Description	Values	Units
1: Hold	Hold time. (ignored by fixed wing, time to stay at waypoint for rotary wing)	*min*:0	s
2: Accept Radius	Acceptance radius (if the sphere with this radius is hit, the waypoint counts as reached)	*min*:0	m
3: Pass Radius	0 to pass through the WP, if > 0 radius to pass by WP. Positive value for clockwise orbit, negative value for counter-clockwise orbit. Allows trajectory control.		m
4: Yaw	Desired yaw angle at waypoint (rotary wing). NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.).		deg
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

Figura 3.4: Exemplo de mensagem de *mavlink*, retirado de [10]

3.2.2 *Mavros*

O *mavros* consiste na integração do protocolo *mavlink* em ROS. Esta integração é efetuada pela mesma equipa que desenvolve o *mavlink*. A implementação do *mavros* recorre a:

- Publicadores ROS, por exemplo para declarar o modo de voo que o veículo se encontra;
- Subscritores ROS, para receber um comando de *mavlink*, nomeadamente o comando de *waypoints*;
- Serviços ROS, para proceder a mudança do modo de voo.

3.2.3 *Ground station*

Uma *ground station* geralmente é constituída por um computador com uma aplicação específica instalada. É através desta que o utilizador monitoriza o estado da aeronave e a controla.

Existem alguns programas para o efeito, sendo alguns exemplos o *QGroundControl* (Figura 3.5) e o *Mission Planner*.

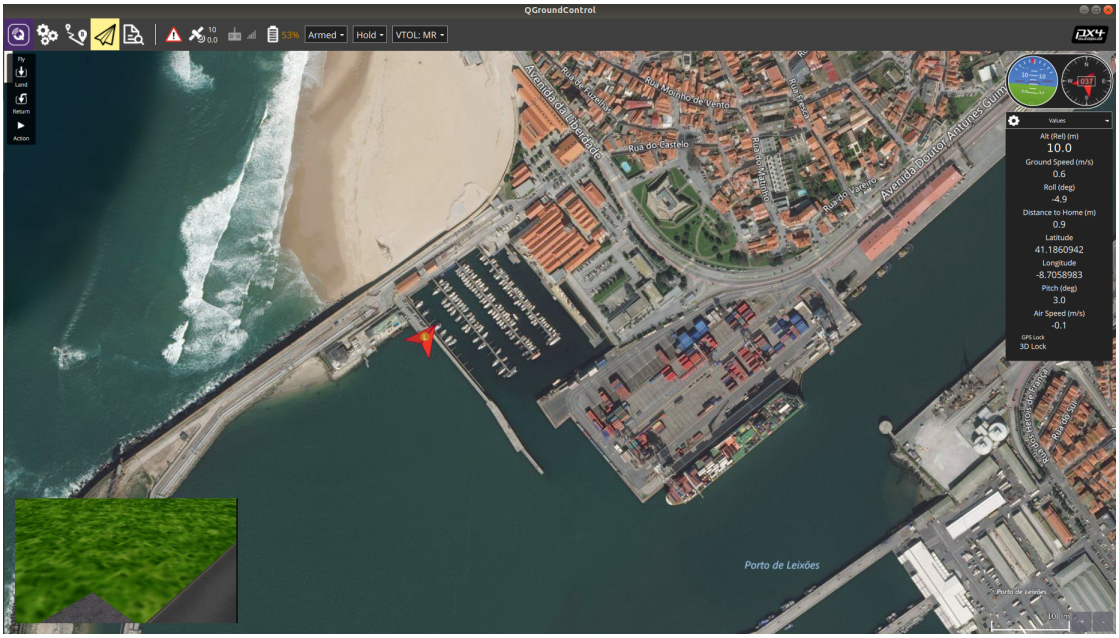


Figura 3.5: Ambiente da aplicação *QGroundControl*

Esta aplicação interage com o *autopilot* através do protocolo *mavlink* e permite o envio de qualquer mensagem de controlo, bem como a respetiva resposta e estado do sistema.

3.3 Relações 3D de referenciais

3.3.1 Referenciais globais e locais

O uso de referenciais é muito comum no mundo da robótica, seja para localizar um robô, seja para ditar a posição de um sensor em relação ao robô. É portanto importante a existência de referenciais, entre os quais globais e locais, e a sua relação de transformação.

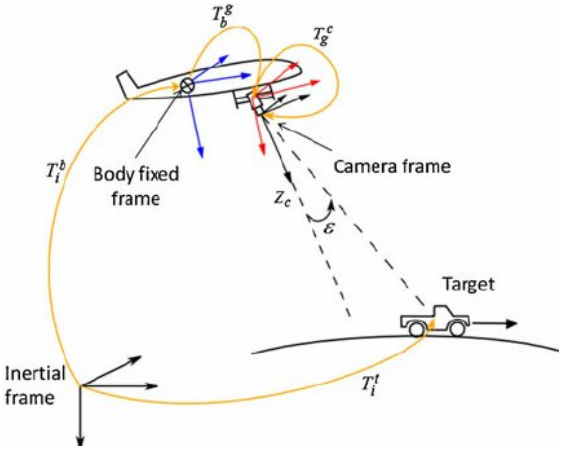


Figura 3.6: Importância de referenciais e respetivas relações, retirado de [11]

A referência geral de todos os robôs é o referencial de navegação global, constituído, por exemplo, por coordenadas de latitude, longitude e altitude. De modo geral, também pode ser gerada uma coordenada de referência local, através de um ponto de referência. A relação entre os dois sistemas de coordenadas pode ser representada por uma matriz T, tal como demonstra a equação 3.1. Nesta, estão presentes as rotações R entre os eixos de referência e as translações t dos mesmos. A matriz de rotação pode ser obtida através dos ângulos de *Euler* [52].

$$\mathbf{T} = \begin{bmatrix} & \mathbf{R_{3\times 3}} & & \mathbf{t_{3\times 1}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

3.3.2 Ângulos de *Euler*

Tal como referido anteriormente, a matriz de rotação pode ser obtida através dos ângulos de *Euler* e vice versa [52]. Estes permitem descrever a atitude de um corpo, recorrendo a ângulos, tal como demonstra a Figura 3.7. Os ângulos são descritos por:

- Roll (ϕ): rotação em torno do eixo X (*east*);
- Pitch (θ): rotação em torno do eixo Y (*north*);
- Yaw (ψ): rotação em torno do eixo Z (*down*).

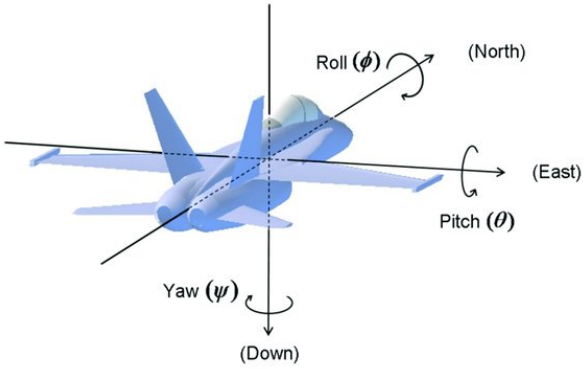


Figura 3.7: Representação dos ângulos de *Euler*, retirado de [12]

Os ângulos de *Euler* são calculados respeitando as seguintes equações [53]:

$$\mathbf{R}_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \tag{3.2}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{3.3}$$

$$\mathbf{R}_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

3.4 *Rapidly-exploring Random Tree* (RRT)

O RRT [54–56] é um algoritmo de *path planning*, cujo seu propósito é encontrar um caminho viável, dado um ponto inicial e final, sem colisões. Este algoritmo, comparativamente a outros com o mesmo propósito, apresenta uma maior eficiência computacional, no entanto contém um elevado caráter de aleatoriedade.

O RRT é um algoritmo iterativo, gerando desde o momento inicial uma série de pontos aleatórios. Para cada conjunto de pontos é verificada a existência de colisões com objetos do meio, retirando-os do conjunto de pontos aceitáveis em caso de colisão. O conjunto de pontos aceitáveis forma possíveis caminhos ou ramos, com origem no

ponto anterior. Cada ramo pode originar novos ramos, de forma iterativa, até chegar ao ponto final. A Figura 3.8 demonstra o conceito abordado e o algoritmo 1 demonstra o processo de implementação.

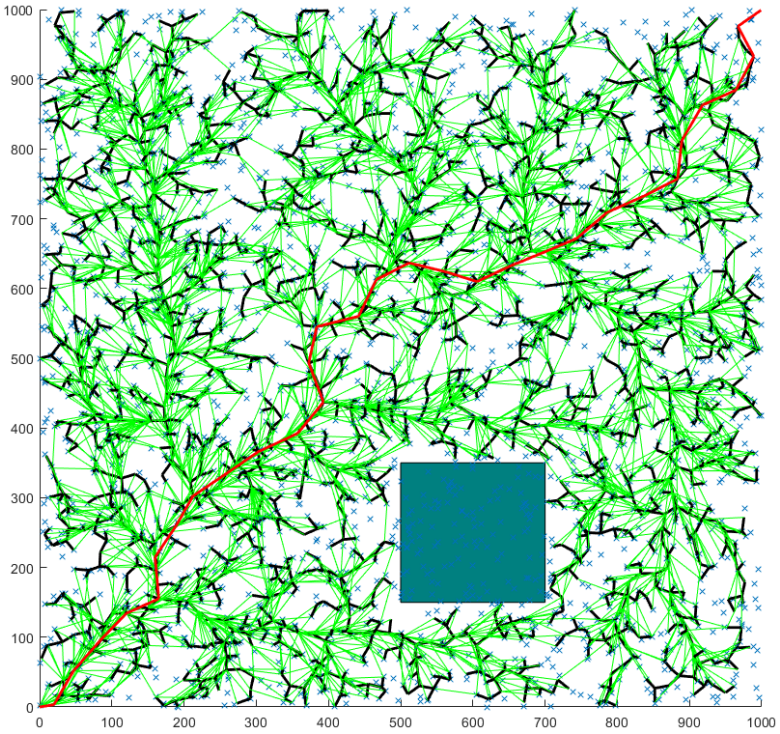


Figura 3.8: Representação visual do algoritmo de RRT

Algoritmo 1 Rapidly-exploring random tree [54]

Input: x_{init}, $K_{vertices}$, Δt
Output: T

T.init(x_{init});
for k=1 **TO** K **do**
  $x_{rand} \leftarrow$ Random_State();
  $x_{near} \leftarrow$ Nearest_Neighbor(x_{rand}, T);
  $u \leftarrow$ Select_Input(x_{rand}, x_{near});
  $x_{new} \leftarrow$ New_State(x_{near}, u, Δt);
  T.add_vertex(x_{new});
  T.add_edge(x_{near}, x_{new}, u);
end for

3.5 *Lin-Kernighan-Helsgaun Heuristic* (LKH)

O algoritmo LKH [13, 57, 58] faz parte de uma família de algoritmos que procura dar resposta a um problema de *Travelling salesman problem* (TSP). O TSP consiste no conceito de um vendedor ter de viajar entre vários pontos, passando uma única vez por cada um e regressando a origem. O objetivo do problema é fazer com que o vendedor efetue esse percurso com o menor custo possível. O custo pode implicar distâncias, tempo ou dinheiro.

Os problemas do tipo TSP podem ter algumas classificações, considerando C_{AB} o custo do ponto A ao ponto B, surgem as seguintes:

- Problema simétrico, quando o custo C_{AB} é igual a C_{BA};
- Problema assimétrico, quando a condição anterior não se verifica;
- Problema métrico, se $C_{AC} \leq C_{AB} + C_{BC}$;
- Problema euclidiano, caso o custo C_{AB} seja uma distância num plano.

O algoritmo LKH é uma implementação generalizada para algoritmos de busca local *k-opt*. O *k-opt* inclui todos os pontos disponíveis do problema de TSP, removendo k sub-caminhos do caminho inicial T e adicionando sub-caminhos diferentes. É de notar que avaliar todas as permutações entre os pontos do TSP resulta num elevado peso computacional, sendo que são apenas analisados conjuntos com *2-opt* ou *3-opt*.

De forma semelhante, o LKH remove e adiciona sub-caminhos ao caminho total, mas apenas explora os mais promissores para uma solução otima. Este algoritmo supõe a existência de um caminho prévio com vários pontos para otimizar. Em cada iteração, um sub-caminho é dividido e metade é invertido e recuperado posteriormente, (Figura 3.9), após este processo estar concluído é calculado o ganho deste caminho através da equação 3.5, sendo g o ganho do respetivo caminho.

$$g = weight(x \rightarrow y) - weight(e \rightarrow x) \tag{3.5}$$

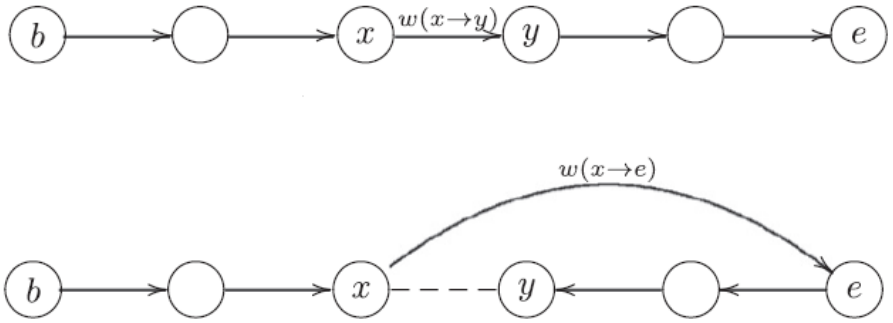


Figura 3.9: Exemplo de separação de caminhos no LKH, adaptado de [13]

Após o cálculo da equação 3.5, é selecionado o caminho com maior ganho positivo. O algoritmo 2 representa uma simplificação do algoritmo LKH.

Algoritmo 2 Algoritmo LKH simplificado, adaptado de [13]

```
1:  Function: ImprovePath (T, depth, R)
2:
3:  while Running do
4:    if depth < α then
5:      for every edge x → y ∈ P such that x ∉ R do
6:        Calculate g = weight(x → y) − weight(e → x)
7:        if g > 0 then
8:          if newTour is an improvement (g is bigger) then
9:            Accept the new tour and Terminate
10:         else
11:           ImprovePath(newTour, depth + 1, R ∪ {x})
12:         end if
13:       end if
14:     end for
15:   else
16:     Find the edge x → y which maximizes g = weight(x → y) − weight(e → x)
17:     if g > 0 then
18:       if newTour is an improvement (g is bigger) then
19:         Accept the new tour and Terminate
20:       else
21:         ImprovePath(newTour, depth + 1, R ∪ {x})
22:       end if
23:     end if
24:   end if
25: end while
```

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Exploração do método "*Structural Inspection Path Planning*"aplicado ao processo de inspeção de Ativos Elétricos

A *Structural inspection path planning* é uma *framework* desenvolvida por investigadores da universidade ETH na Suíça, e está disponível na página de *github*[1], sendo possível a sua utilização em ROS.

Esta *framework* foi estudada para identificar a viabilidade de aplicação ao problema. Foi analisada a sua implementação e retirados conceitos importantes para o desenvolvimento da dissertação. Este capítulo aborda a implementação do algoritmo, a demonstração da utilização do mesmo e, no final, apresenta vantagens e desvantagens do mesmo como método de análise crítica à aplicação nesta dissertação.

Tal como o nome indica, esta aborda a inspeção de estruturas, com recurso a um UAV, nomeadamente o problema de cobertura de área da mesma. Dado o caminho a seguir pelo equipamento, a inspeção fica reduzida à utilização de sensores de percepção (ex. LiDAR ou câmaras). Contudo, o objetivo desta *framework* é obter o caminho com o menor custo possível, em termos de distância percorrida ou tempo, tendo em conta as restrições físicas dos sensores e do equipamento de inspeção. Esta otimização é resolvida com recurso a problemas do tipo TSP. Os parâmetros de entrada desta *framework* que podem ser controlados pelo utilizador são:

[1] `https://github.com/ethz-asl/StructuralInspectionPlanner`

- Um ficheiro .*Standard Triangle Language* (STL) que contenha uma *mesh* composta por vários triângulos da estrutura a inspecionar

- A *bounding box* que consiste nos limites máximos permitidos de inspeção, esta tem de ser retangular.

- As distâncias máximas e mínimas de inspeção ao objeto

- Os parâmetros de *Field-Of-View* (FOV) da câmara e o seu respetivo ângulo vertical da mesma em relação ao veículo

- Velocidade máxima de translação e rotação

Os dados de saída do algoritmo são um ficheiro .m que contém os dados da *mesh*, das posições a seguir e a respetiva orientação, e ainda o custo da inspeção. Paralelamente, são também publicados tópicos em ROS com os mesmos dados.

4.1 Algoritmo

O algoritmo 3 representa a estrutura da *framework* utilizada. Este possui uma característica iterativa uma vez que, após a solução inicial, este itera N soluções de menor custo. O número de soluções pode ser definido pelo utilizador, sendo o número por defeito de 20 iterações. Nos tópicos seguintes são explicados os processos utilizados em cada secção do algoritmo.

Algoritmo 3 Structural Inspection Path Planning [14] [59]

```
1: Initialize variables
2: Load STL mesh
3: Sample Viewpoints using BVS algorithm
4: Compute collision-free path for all points
5: Fill cost matrix and find optimal solution
6: while iteration < n_of_iterations do
7:     Re-Sample Viewpoints
8:     Re-Compute collision-free path for all points
9:     Re-Fill cost matrix and update bestPath and cost
10: end while
11: return bestPath, cost
```

4.1.1 Criação e *downsampling* da *mesh*

Um dos principais requisitos da *framework* é uma *mesh* composta por triângulos. A divisão destes influencia o detalhe da inspeção num determinado local, isto é, quantos mais triângulos estiverem contidos numa secção do objeto, mais detalhada será a inspeção nessa área.

4.1.2 Obtenção de *viewpoints*

A obtenção dos *viewpoints* é feita a partir de cada triângulo, por um método denominado de *convex optimization method*. Este consiste na optimização de um *viewpoint* tendo em conta a distância ao seu vizinho, minimizando-a. De forma a garantir que todos os pontos são admissíveis, garantindo a sua visibilidade, são impostas restrições na geração de cada ponto através do algoritmo BVS.

4.1.2.1 Restrição do ângulo de incidência

A restrição do ângulo de incidência é imposta de modo a garantir a visão total do triângulo. Segue as restrições na equação 4.1, sendo g a posição [x, y, z], x_i o vértice i do triângulo, n_i a normal dos hiperplanos, a_N a normal do triângulo e por fim, d_{min} e d_{max} as distâncias mínimas e máximas de inspeção, respectivamente. A figura 4.1 demonstra a restrição do ângulo de incidência.

$$\begin{bmatrix} (g-x_i)^T n_i \\ (g-x_1)^T a_N \\ -(g-x_1)^T a_N \end{bmatrix} \geq \begin{bmatrix} 0 \\ d_{min} \\ -d_{max} \end{bmatrix}, i = \{1,2,3\} \tag{4.1}$$

4.1.2.2 Restrição dos parâmetros da câmara

Tendo em conta o FOV da câmara, a abertura vertical impõe uma restrição de um cone Bidimensional (2D) de visão não convexo, dada a distorção da lente. De modo a fazer uma aproximação convexa do problema são seguidas as restrições da equação em 4.2. O triângulo é dividido em N_c partes, de acordo com a figura 4.2, de forma a compensar a convexidade da lente da câmara, reduzindo assim o erro do cone de visão, aproximando o triângulo a uma figura convexa.

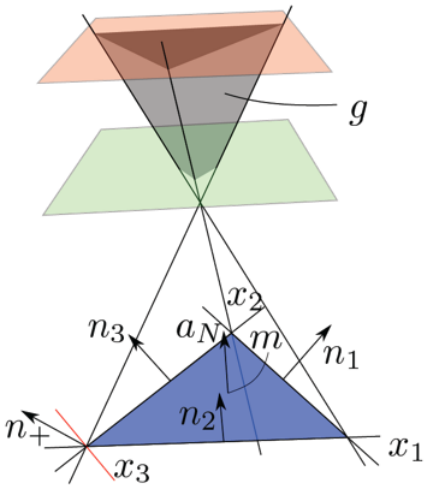


Figura 4.1: Restrição do ângulo de incidência, adaptado de [14]

$$\begin{bmatrix} (g - x^{rel}_{lower})^T n^{cam}_{lower} \\ (g - x^{rel}_{upper})^T n^{cam}_{upper} \\ (g - m)^T n_{right} \\ (g - m)^T n_{left} \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.2}$$

Na equação, x^{rel}_{lower} e x^{rel}_{upper} são os vertices relevantes do triângulo, m é o centro do triângulo, n^{cam}_{lower},n^{cam}_{upper},n_{right} e n_{left} representam os respetivos hiperplanos.

4.1.3 Obtenção do caminho a seguir, passando por todos os *viewpoints*

De modo a resolver o problema de chegar a todos os pontos de forma óptima, são aplicados dois algoritmos genéricos de planeamento de trajetórias. Para encontrar uma solução entre dois *viewpoints*, é aplicado um algoritmo de *Boundary Value Solver* (BVS), caso não existam obstáculos entre os dois pontos. Se existirem obstáculos, é aplicado um algoritmo de RRT, de modo a efetuar a exploração de caminhos possíveis sem colisões.

É importante salientar que para ambos os algoritmos, o modelo a ser utilizado apenas tem em conta a posição e a orientação do veículo, considerando os eixos de *roll* e *pitch* nulos.

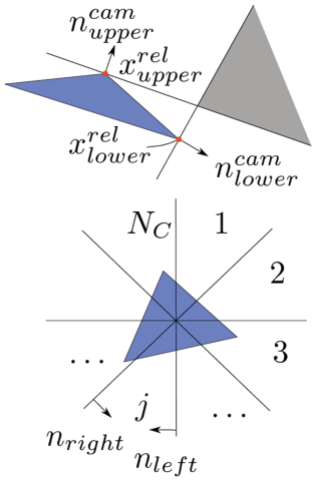


Figura 4.2: Restrições da câmara, adaptado de [14]

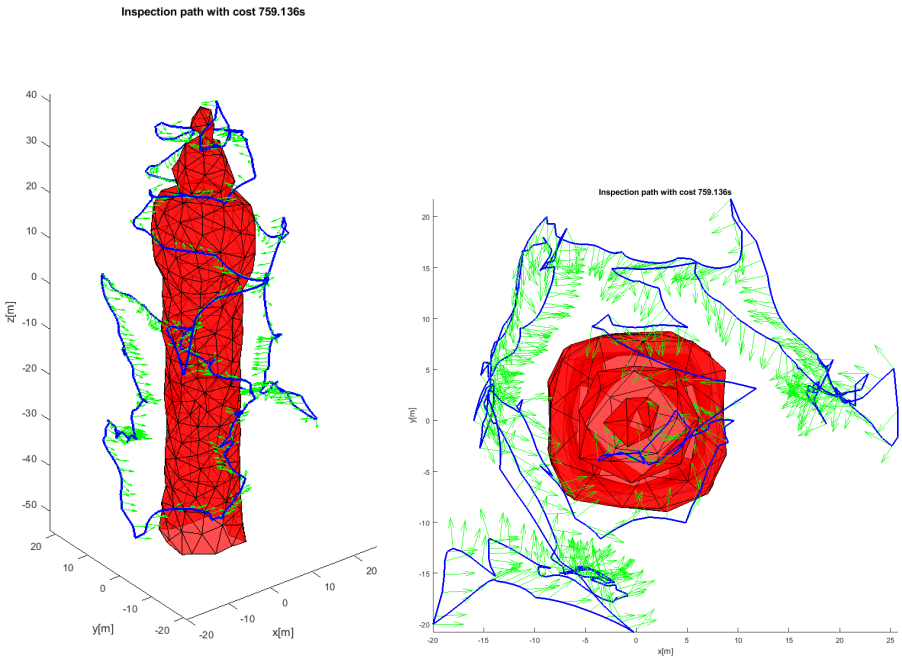
4.1.4 Optimização e preenchimento da matriz de custo

Tal como referido anteriormente, a otimização do problema é efetuada através de um problema do tipo TSP. Através do método LKH, é calculado o caminho com menor custo (seja custo temporal ou de distância) e, consequentemente, calculada a matriz de custo. O custo corresponde ao somatório dos tempos de execução entre todos os pontos. A equação 4.3 demonstra como é calculado o custo de tempo entre dois pontos, sendo d a distancia euclidiana entre os dois pontos, v_{max} e ψ_{max} as velocidades translacionais e rotacionais máximas, respectivamente, e ψ_i a orientação do ponto i .

$$t_{ex} = max(d/v_{max}, ||\psi_1 - \psi_0||/\dot{\psi}_{max}) \tag{4.3}$$

4.2 Resultados da framework

A *framework* já disponibiliza alguns exemplos para o utilizador executar. A figura 4.3 representa um exemplo de uma simulação de inspeção da estrutura do *Big Ben* em Londres. Na imagem, a vermelho estão os triângulos a inspecionar, a azul o caminho que o veículo deve seguir e a verde a representação da orientação que o veículo deve respeitar.



(a) *Big Ben* (vista geral) (b) *Big Ben* (vista de cima)

Figura 4.3: Exemplo fornecido do *Big Ben*

Para além dos exemplos, foi efetuada a implementação de uma *mesh* de um poste de alta tensão de forma a validar a sua utilização. Contudo, a configuração da *mesh* não foi simples, tendo sido efetuadas várias adaptações. A figura 4.4 demonstra a *mesh* antes de qualquer processamento.

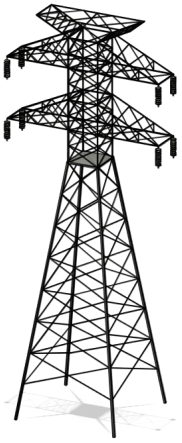
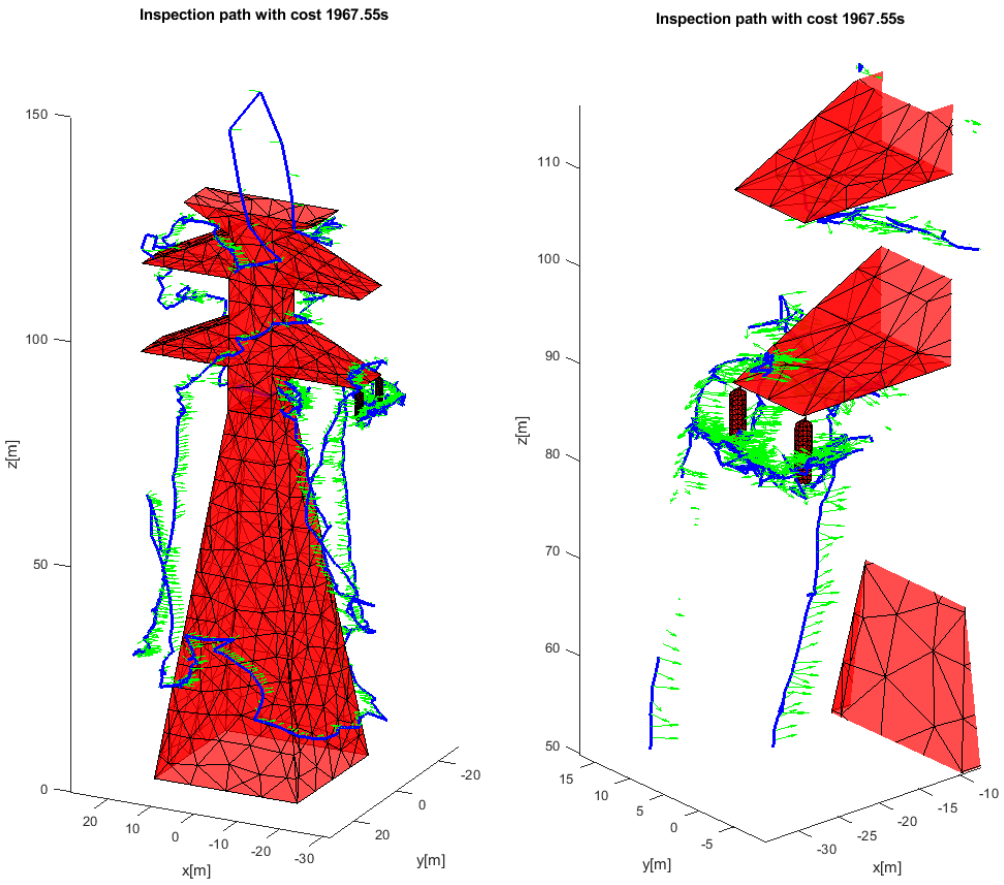


Figura 4.4: Modelo original utilizado para validação do algoritmo.

Como é possível constatar na figura 4.5, que demonstra os resultados da tentativa de implementação, não estão representados todos os isoladores. A opção de limitar o número de isoladores para esta demonstração deve-se ao facto de, para além de aumentar o número de triângulos em cerca de 300 unidades por isolador, a sua inserção induz em mais colisões contra o objeto.



(a) Torre (vista geral) (b) Torre (aproximação aos isoladores)

Figura 4.5: Inspeção de um poste de alta tensão recorrendo ao algoritmo

Contudo, no exemplo da torre, é possível verificar na figura 4.6 que o algoritmo não está a funcionar no que toca ao aspeto de *collision avoidance*, sendo que este apresenta um caminho que atravessa a estrutura. Apesar de o exemplo de falha de deteção de colisões apenas ter sido evidenciado no exemplo da torre, este fenómeno ocorre também nos exemplos fornecidos, não sendo uma falha do utilizador. A falta de consistência reside na estimação de parâmetros iniciais do algoritmo.

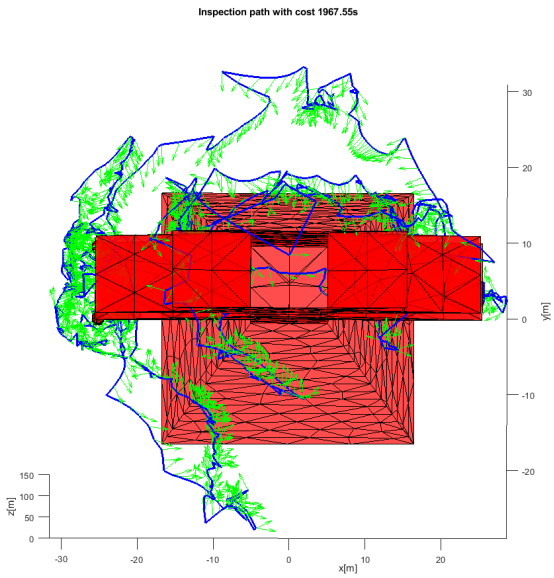


Figura 4.6: Inspeção de um poste de alta tensão (vista de cima)

Para ambos os cenários exemplificados foram levantados alguns dados relevantes, a tabela 4.1 mostra os resultados. Os parâmetros de entrada de ambos os ensaios são os mesmos, com excepção da *mesh*.

Tabela 4.1: Comparação do desempenho do algoritmo entre os exemplos

	Big Ben	Poste de alta tensão
Nr. de triângulos	526	1588
Nr. de pontos obtidos	1054	3178
Custo de inspeção (s)	759	1967
Tempo de cálculo (ms)	38635	182298
Tempo do LKH (ms)	31712	156844
Tempo de RRT* (ms)	9	26
Tempo de cálculo de distância (ms)	1417	8794
Tempo de cálculo dos viewpoints (ms)	4931	13588

4.3 Discussão da *framework*

Esta framework foi publicada em 2015, sendo que já existem versões mais atualizadas, mas não públicas, sendo que foram aplicadas algumas melhorias. Para o caso específico de inspeção de um poste de alta tensão, esta abordagem não é viável uma vez que a

torre não é uma estrutura opaca. A inspeção da torre não é essencialmente estrutural, mas sim dos isoladores e junções da torre.

É importante evidenciar que para uma *mesh* poder ser utilizada na *framework*, esta tem de ser pré-processada com muito rigor. No exemplo da torre, tiveram de ser retirados não só os triângulos da base (uma vez que a inspeção da base no mundo real não é aplicável), mas também do topo da estrutura, já que o algoritmo não conseguia computacionar um ângulo de visão viável para essa zona.

Alguns aspetos a melhorar na *framework*, com um dos objetivos a inspeção de um poste de alta tensão são:

- Implementação de ângulos variáveis para a câmara em relação à estrutura para inspeção do mesmo ponto de vários ângulos (preparação para inspeção termográfica);
- A forma como são identificados os pontos de interesse, não é viável a divisão de triângulos;
- Caso existam pontos cuja inspeção não é possível, devido a restrições físicas, a *framework* não funciona;
- A possibilidade de inspeção de um ponto com rotação de *roll* e *pitch* diferentes de 0;
- A falta de viabilidade no *collision avoidance*.

Apesar dos problemas identificados, a *framework* foi publicada na comunidade científica e recebida de forma positiva, uma vez que contém conceitos inovadores e relevantes para a inspeção, nomeadamente:

- A utilização de algoritmos de BVS e RRT para encontrar um caminho entre os pontos.
- A utilização do algoritmo de LKH para a otimização dos pontos.
- O facto de a partir de um ficheiro STL ser possível gerar a manobra de inspeção.
- O baixo peso computacional.

Esta página foi intencionalmente deixada em branco.

Capítulo 5

Projeto

Neste capítulo irá ser apresentada uma arquitectura geral do sistema que no final será detalhada com mais rigor. Serão também abordados alguns aspetos cruciais na montagem do VTOL, nomeadamente a escolha de componentes mecânicos e *frame*, bem como algumas peças de montagem do mesmo. De seguida serão apresentadas duas propostas de manobra de inspeção, com o respetivo fundamento matemático. Nesta secção consta também uma comparação entre os dois *Firmwares* mais utilizados nos controladores de voo do ramo de robótica aérea. Por fim, serão expostas algumas decisões de projeto, fundamentando as mesmas.

5.1 Arquitetura de alto nível do sistema

O projeto de desenvolvimento de um VTOL surgiu da disciplina de mestrado de Sistemas Autónomos de Laboratório de Sistemas Multirobóticos (LASMU). Dada a grande dimensão do projeto, este foi desenvolvido em grupo, com a aluna Diana Salgado, sendo que cada um dos elementos se focou numa parte especifica do projeto. A Figura 5.1 representa a arquitetura de algo nível do sistema, sendo que a secção de interesse está representada a azul. Na figura não estão representados os elementos de controlo do sistema, apenas a arquitetura do VTOL. Para além dos elementos representados, existe uma *ground station*, que consiste num computador com o ROS instalado e os respetivos nós necessários ao voo.

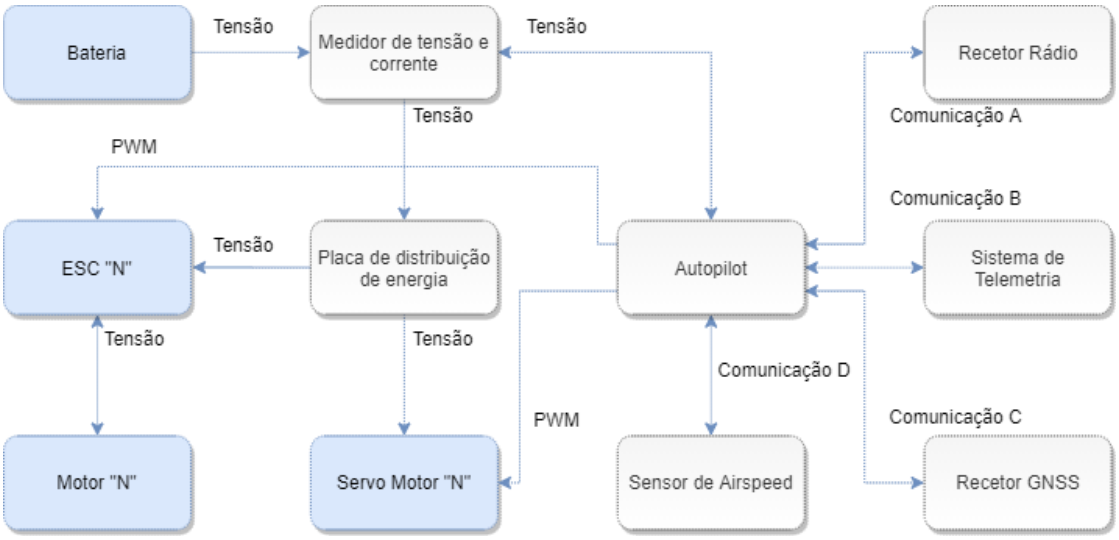


Figura 5.1: Arquitetura de alto nível

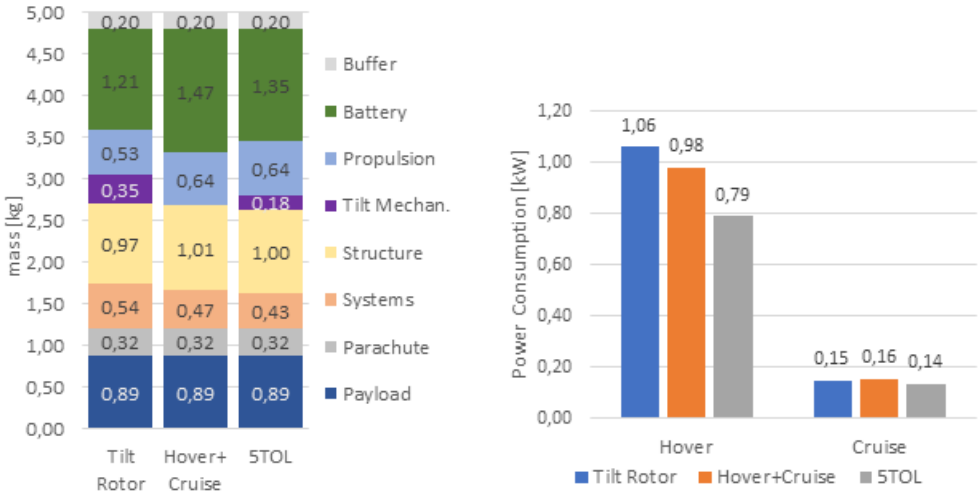
5.2 Montagem do VTOL

5.2.1 Estudo dos tipos de *frames*

Tal como referido no capítulo 2, existem vários tipos de VTOL. Entre os apresentados, o *Standard VTOL* e o *Tilt rotor* são as duas hipóteses mais promissoras para o cenário de aplicação.

No trabalho em [15], são analisadas as configurações de aeronave anteriormente enumeradas e acrescenta uma opção nova, a *frame 5TOL*. Para efeitos de análise, apenas serão consideradas as *frames* tradicionais, uma vez que a *5TOL* foi apresentada apenas como um conceito e a referência de *Hover + Cruise* corresponde ao *Standard VTOL*.

A Figura 5.2(a) e 5.2(b) representam a distribuição de massa e consumos energéticos dos diferentes tipos de *frame*, respetivamente. Em análise dos gráficos é possível concluir que a distribuição de peso no *Tilt rotor* no que toca ao sistema de propulsão e mecanismos respetivos, representam uma maior porção relativamente ao *Standard VTOL*. Para além disso, o *Tilt rotor* acomoda uma bateria com menor capacidade.



(a) Estudo da distribuição de massa

(b) Estudo dos consumos energéticos

Figura 5.2: Estudo dos tipos de *frames* de VTOL, retirado de [15]

Em análise ao gráfico da Figura 5.2(b) é possível concluir que para além dos aspetos referidos anteriormente, o *Tilt rotor* também apresenta um maior consumo de energia em modo *multicopter* e menor em modo *fixed wing (cruise)*. Em termos percentuais, o *Tiltrotor* em modo *multirotor* gasta aproximadamente mais 7.55 % de energia e em modo *cruise* gasta aproximadamente 6.25 % de energia quando comparado com o *Standard VTOL*.

Por fim, também é importante referir algumas vantagens e desvantagens intrínsecas ao tipo de *frame* usada, sendo estas:

- *Standard VTOL*
  - Vantagens
    - Fácil implementação mecânica;
    - Transição mais segura e suave entre modos de voo;
    - Maior segurança total do sistema;
  - Desvantagens
    - O peso dos motores de *multirotor* em *forward-flight (fixed wing)* ficam como "peso morto"e o inverso (motores de avião) em modo *multirotor* também se verifica;
    - Maior consumo de energia, inerente ao número de motores;

- *Tilt rotor*
  - Vantagens
    - Otimização do uso da eletrónica, não tendo tanto " peso morto";
    - Possível utilização de mais motores para voo *fixed wing*;
  - Desvantagens
    - Manutenção e implementação mecânica difícil;
    - A escolha dos *propellers* é muito complexa, uma vez que é necessário que o mesmo *propeller* opere nos dois modos de voon, não sendo a solução otimizada para nenhum;
    - Em caso de falha, a segurança fica comprometida.

5.2.1.1 Decisão do tipo de *frame*

Após o estudo dos tipos de *frames*, foi concluído que a solução ideal para a aplicação é a *frame* de um *Standard VTOL*. O principal fator de peso na decisão foi a segurança do voo, uma vez que, em caso de falha de um dos motores de um *Tilt rotor*, pode resultar a queda do aparelho. Uma queda em cima de uma linha de transmissão pode ser catastrófica.

Para além disso, a implementação e manutenção do *Tilt rotor* é mais complexa, não se justificando assim a vantagem de consumo energético que este possui.

5.2.2 *Frames* para VTOL

A *frame* do veículo é um dos elementos mais importantes, é através da mesma que são efetuadas grande parte das restrições, como por exemplo o *payload* que a aeronave é capaz de suportar. De modo a garantir que o *payload* é respeitado, é necessário garantir que tanto os motores como o tamanho da asa são adequados ao peso que se pretende colocar na aeronave. Assim, de seguida, são apresentadas as *frames* consideradas interessantes para a elaboração do seguinte projeto.

5.2.2.1 Skywalker X8

A primeira *frame* considerada interessante é a *Skywalker X8* (Figura 5.3). Esta *frame* tem o formato de uma asa delta, uma envergadura de 2.2 *metros* e *Maximum Takeoff Weight* (MTOW) de 4 *kg*.

Figura 5.3: *Frame* Skywalker

Dada a grande área das asas da *frame* seria fácil utilizar a mesma para a adaptação a VTOL (utilizando tubos de carbono acoplados nas asas). Já a área da parte central, podia facilmente ser aproveitada para inserir a eletrónica necessária para o voo, tal como demonstram as seguintes imagens:

(a) *Frame* Skywalker 1 (Parte central) (b) *Frame* Skywalker 2 (Parte central)

Figura 5.4: Exemplo de aplicação de eletronica na frame *skywalker*

O resultado da transformação seria algo semelhante a uma solução já projetada pela equipa do PX4, cuja a informação está presente online em [16]. A Figura 5.5 demonstra o resultado final da montagem.

Figura 5.5: *Frame* Skywalker versão VTOL, retirado de [16]

5.2.2.2 Dragon VTOL

A segunda *frame* considerada interessante é a *Dragon VTOL* [17]. Em comparação com a anterior, esta *frame* tem a vantagem de já trazer na sua constituição as barras de carbono que sustentam os motores de drone. Esta *frame* tem uma estrutura de avião convencional, com 2.2 m de envergadura e um MTOW de 9 Kg.

Figura 5.6: *Frame* Dragon VTOL, retirado de [17]

Tal como a *frame* anterior, possui três compartimentos na parte central, onde pode ser acoplada a bateria e toda a eletrónica necessária para o bom funcionamento da aeronave. Para além disso, possui a vantagem de ter as asas removíveis, existindo conectores para a passagem de energia e informação entre as asas e controlador de voo.

Figura 5.7: Componentes removíveis da *frame*, retirado de [17]

É também importante salientar que o mecanismo utilizado nas asas é também utilizado nas barras de carbono. Isto confere facilidade no transporte da aeronave e facilidade de reparação dos componentes.

Figura 5.8: Conecção das barras de carbono à *frame*, retirado de [17]

5.2.2.3 Decisão da *frame*

Após fazer uma análise cuidada das vantagens e desvantagens de cada uma das *frames*, foi escolhida a frame *Dragon VTOL*. No entanto, foram analisadas possíveis distribuições de pesos dos componentes, representados pelas tabelas 5.1 e 5.2.

Tabela 5.1: Distribuição de peso da frame Skywalker X8

Componente	**Peso (g)**
Frame Skywalker X8	880
Motores Multirotor	450
Motor Fixed-Wing	270
Bateria	850
Trem de aterragem	150
Servo motores	40
Suportes motores Multirotor (carbono)	200
Eletrónica	200
Total	**3040**

Tabela 5.2: Distribuição de peso da frame Dragon VTOL

Componente	**Peso (g)**
Frame Dragon VTOL (com motores para voo)	4000
Bateria	1610
Trem de aterragem	150
Eletrónica	200
Total	**5 960**

Para além de já possuir toda a estrutura de VTOL preparada para instalação, nomeadamente conectores para as asas e tubos de carbono dos motores, esta também apresenta a melhor relação de peso da *frame* versus MTOW. Assim, todos os componentes selecionados nas secções seguintes serão projetados para esta solução.

5.2.3 Propulsão *multirotor*

Dada a escolha da *frame*, os motores a escolher devem seguir as especificações da mesma, tendo em conta a sua dimensão, peso, e *thrust* fornecido. Para a escolha dos motores de drone foram pesquisados dois modelos:

- DJI E800, uma solução de baixo peso, para a propulsão do veiculo em modo drone;
- DJI E1200, uma solução alternativa aos anteriores, com um maior peso, mas também com maior propulsão.

A DJI [60] é das marcas mais conhecidas e conceituadas do mercado, sendo que foram escolhidos os motores da marca para a propulsão em modo drone. Ao longo do planeamento do projeto foi sempre tida em consideração a capacidade de voo em *fixed-wing*, sendo esta o principal factor de restrição, limitada pelo peso. Os motores utilizados são do tipo *brushless* uma vez que sofrem menos desgaste e atingem maiores velocidades, sendo assim ideais para o uso em drones.

5.2.3.1 DJI E800

Os motores DJI E800 [18] asseguram a restrição de peso da *frame*, uma vez que são desenvolvidos de modo a ser o mais leves e eficientes possível. Cada motor tem um *thrust* máximo de 2100 gramas o que seria suficiente para a propulsão, no entanto ficaria muito perto do limite mínimo de *thrust* de $6.5\,kg$ (peso total da aeronave). Estes motores operam a uma tensão nominal de 26 V e um máximo de 20 A, sendo a marca comercializa kits com os motores e os *Electronic Speed Controller*s (ESCs) com o modelo 620S.

Figura 5.9: Motor DJI E800, retirado de [18]

5.2.3.2 DJI E1200 PRO

Os motores DJI E1200 PRO [19] são a gama seguinte aos E800. Estes seguem igualmente todas as especificações necessárias para o voo, no entanto aproximam-se do limite máximo de peso que o avião poderá suportar em modo *fixed-wing* ($8\,kg$). Uma vantagem em relação aos E800 é que conseguem um *thrust* máximo por rotor de 3900 gramas, o que se tornou o fator decisivo, uma vez que seria mais do que suficiente para todas as manobras a ser realizadas. Estes motores operam a uma tensão nominal de 26 V e um máximo de 40 A e contém o ESC incluído na mesma estrutura.

Figura 5.10: Motor DJI E1200, retirado de [19]

5.2.4 Propulsão *Fixed-wing*

5.2.4.1 Motor T-Motor AT3520-550

O motor da T-Motor AT3520-550 [20] é o motor ideal para as duas *frames* propotas, tendo em conta as suas características em termos de dimensão e *thrust*. É um motor popular na área dos aviões de rádio controlo, uma vez que tem uma capacidade de *thrust* de 4116 gramas. A T-Motor também é reconhecida na área de motores como um dos melhores fabricantes, sendo que esta empresa produz dos motores mais eficientes e com melhor qualidade. Como o motor nunca vai suportar o peso total do avião mas sim projetar para voo *fixed wing*, considera-se esta a opção ideal para o projeto. Este motor opera a uma tensão nominal de 26 V e um máximo de 50 A.

Figura 5.11: T-Motor AT3520-550, retirado de [20]

5.2.4.2 ESC HobbyKing YEP 80 A

Contrariamente aos motores de propulsão da DJI, a T-Motor não recomenda nenhum ESC em específico. Foi necessário efetuar uma pesquisa e determinar qual é que seria o mais apropriado a ser utilizado. Assim, após analisadas as especificações do motor, foi possível comprovar que a corrente máxima consumida pelo motor é de 50 A. No entanto, e para evitar eventuais picos de corrente é boa prática a escolha de um ESC com capacidade de corrente superior, sendo o modelo escolhido o HobbyKing YEP de 80 A [21].

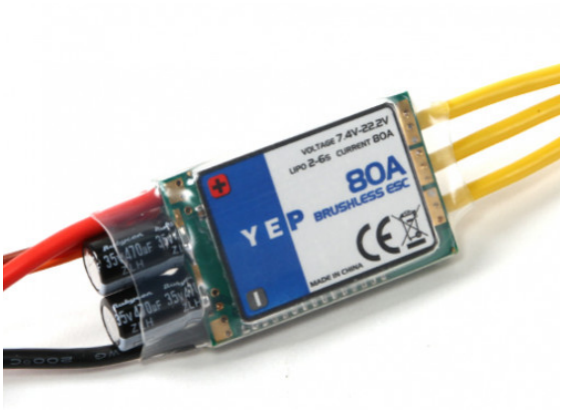


Figura 5.12: ESC apropriado ao motor T-Motor [21]

5.2.5 Servo Motores

Os servo motores são necessários para atuar nas superfícies de controlo do avião. O fabricante da *frame* recomenda servo motores 17g, sendo que os selecionados são o modelo Emax-ES3004 [22]. Estes motores são *full metal*, uma vez que tem as suas engrenagens em metal, possibilitando assim suportar até 3.5 Kg.

Figura 5.13: Servo motor selecionado, retirado de [22]

5.2.6 Bateria

Como os motores de propulsão selecionados possuem uma tensão nominal de 26 V, a bateria selecionada terá de respeitar esse requisito. Assim, foi selecionada uma bateria de 6 células *Lithium polymer battery* (LiPo) com 12 Ah, da gama *Turnigy Graphene Professional*. A tecnologia LiPo permite descargas de corrente instantâneas de elevada quantidade, sendo este um requisito obrigatório para um *multirotor*. Esta bateria pesa 1610 g, o que representa 27 % do peso total da aeronave.

Figura 5.14: Bateria selecionada, retirado de [23]

5.2.7 Trem de aterragem

Apesar de a *frame* selecionada já ter incluído os suportes de VTOL, esta não possui um trem de aterragem funcional. Dada a dimensão do motor de *fixed-wing* e do respetivo *propeller*, no momento de aterragem, o último iria bater no chão, ficando danificado.

Assim, foi selecionado um mecanismo (Figura 5.15) capaz de elevar a *frame* toda, que mais tarde irá ser acoplado às barras de carbono que suportam os motores. Para além da elevação, este mecanismo também proporciona amortecimento no momento de aterragem.

Figura 5.15: Mecanismo de aterragem selecionado, retirado de [24]

5.3 Manobra de inspeção

Tal como referido anteriormente, um VTOL possui características relevantes tanto de *multicopter* como de *fixed-wing*. A principal vantagem dos *multicopters* é a sua manobralidade, ideal para inspeções detalhadas a qualquer objeto, no entanto gastam muita bateria, devido aos seus motores. No que toca aos *fixed-wing* estes apresentam restrições nos seus movimentos. A sua autonomia de voo é uma característica importante para inspeções de grande números de estruturas.

Assim, foram planeadas duas manobras, tirando partido das vantagens de cada um dos tipos de veículo. É importante referenciar também que independentemente da manobra escolhida, a viagem entre os apoios pode ser efetuada sempre em *fixed-wing*, graças à capacidade do VTOL de comutação a meio de um voo.

5.3.1 Manobra em modo drone

Esta manobra é ideal para inspeções que tenham de ser obrigatoriamente detalhadas, como por exemplo, a inspeção visual da deterioração de um isolador ou deteção de elementos externos à estrutura.

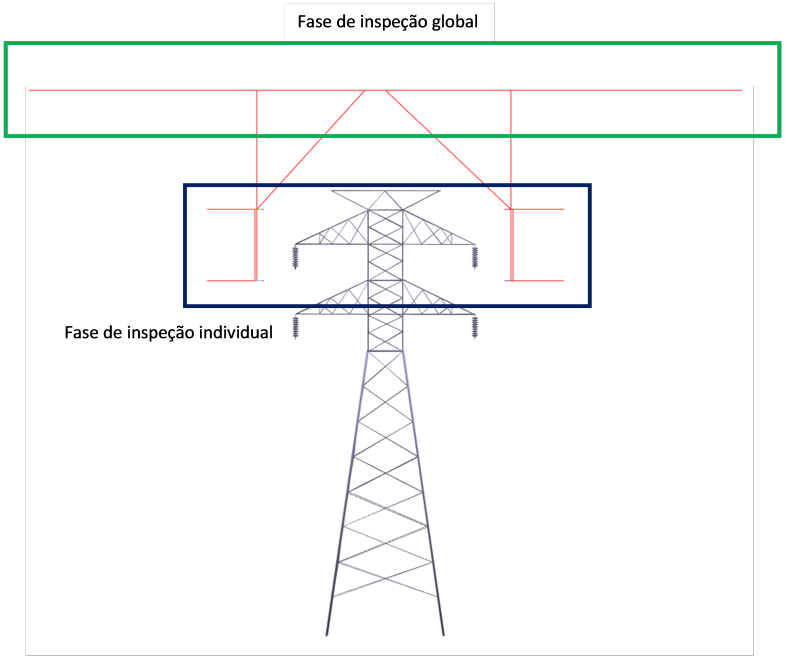


Figura 5.16: Fases de inspeção em modo de drone

A Figura 5.16 representa as diferentes fases de inspeção:

- Inspeção geral, através de um círculo acima da torre, com a câmara a apontar para a zona de interesse de inspeção (os isoladores);
- Inspeção individual de cada isolador, mais detalhada que a anterior, em semi-circulo, mantendo uma distância segura a cada elemento da linha de transmissão.

Assim, foi assumido que a câmara se encontra posicionada na parte inferior do veiculo, com um *pitch* (θ) de 30° e rotação de 0° em *roll* e *yaw*. A câmara têm um FOV de 60°, tanto horizontal como vertical. As equações 5.1 e 5.2 permitem calcular parâmetros restritivos da inspeção, representados nas figuras 5.17 e 5.18.

$$H = \frac{raio}{tan(90 - \theta)} \tag{5.1}$$

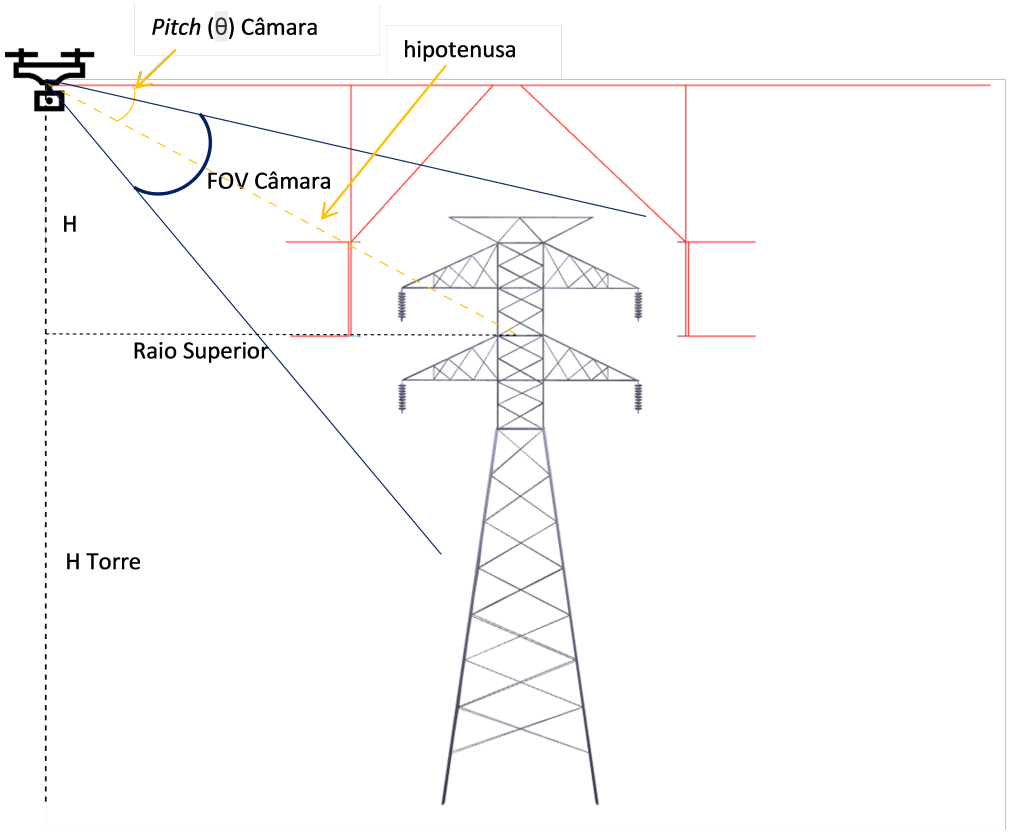


Figura 5.17: Planificação de inspeção com drone, circulo superior

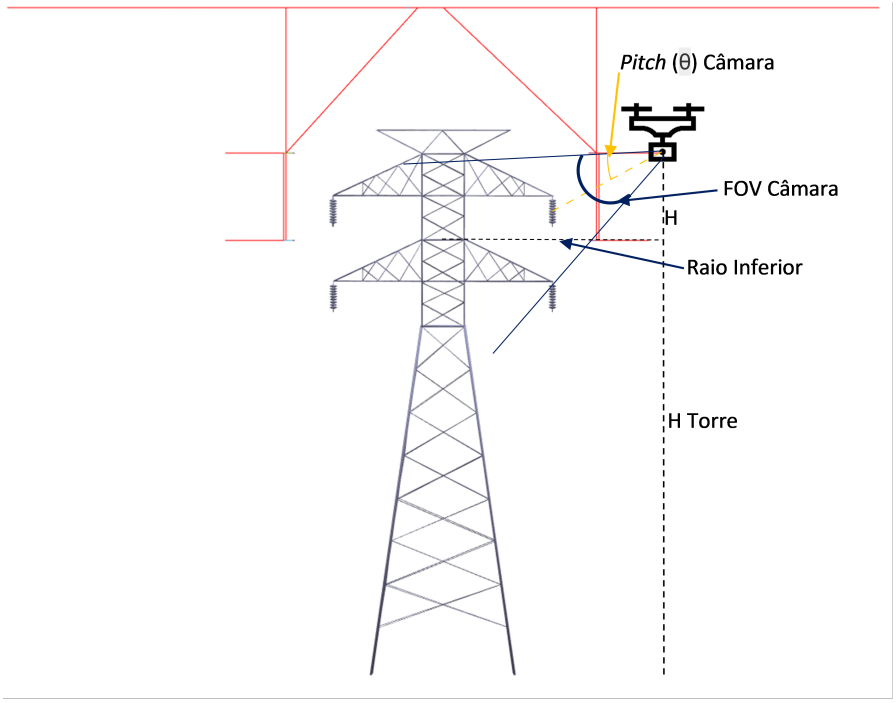


Figura 5.18: Planificação de inspeção com drone, circulo inferior

Através da equação 5.2 é possível calcular se toda a área de inspeção desejada está contida na imagem vista pela câmara.

$$\begin{cases} H\_FOV_{cam} = 2 * hipotenusa * \tan(\frac{FOV}{2}) \\ V\_FOV_{cam} = H\_FOV_{cam} \end{cases} \tag{5.2}$$

Caso as áreas de interesse estejam contidas na imagem da câmara, serão aplicadas as equações descritas em 5.3 que ditam a posição X, Y, Z que o veículo deve assumir para efetuar a manobra de inspeção.

$$\begin{cases} 0 \leq t \leq 10 \\ X_{local} = raio * cos(t) \\ Y_{local} = raio * sen(t) \\ Z_{local} = H + H_{torre} \end{cases} \tag{5.3}$$

5.3.2 Manobra em modo avião (*Loiter*)

Esta manobra tira partido da maior vantagem de um *fixed-wing*, a sua autonomia de voo. Um *Loiter* consiste numa manobra circular, tal como demonstra a Figura 5.19, em torno de um ponto, sendo que podem ser parametrizados o raio do circulo e a velocidade de voo.

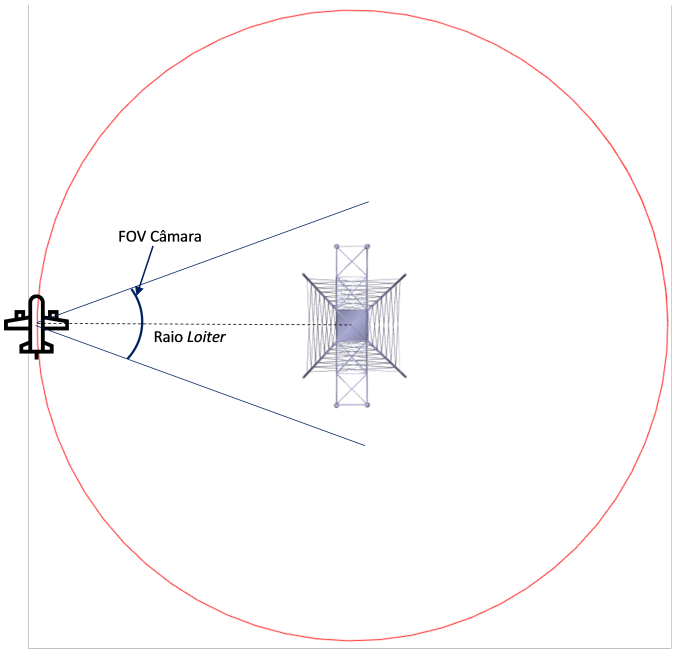


Figura 5.19: Planificação de inspeção avião em *Loiter*, vista superior

A equação 5.4 descreve a relação entre as variáveis, sendo que g corresponde à aceleração gravítica, v corresponde a velocidade do avião, $raio$ corresponde ao raio do circulo e ϕ corresponde ao *roll* a ser aplicado ao avião para satisfazer as condições.

$$tan(\phi) = \frac{v^2}{g * raio} \tag{5.4}$$

Após o cálculo do *roll*, a equação em 5.5 pode ser aplicada, de modo a calcular a altitude a que o *Loiter* deve ser feito para que a imagem da câmara seja projectada na zona de interesse. É assumido que a câmara se encontra fixa ao avião, na mesma posição que o exemplo anterior, mas com rotação de 0° em *roll* e *pitch* e com −90° em *yaw*, para que esta aponte para a lateral do avião.

$$H = \frac{raio}{tan(90 - \phi)} \tag{5.5}$$

A Figura 5.20 consiste numa representação visual dos parâmetros anteriormente referidos.

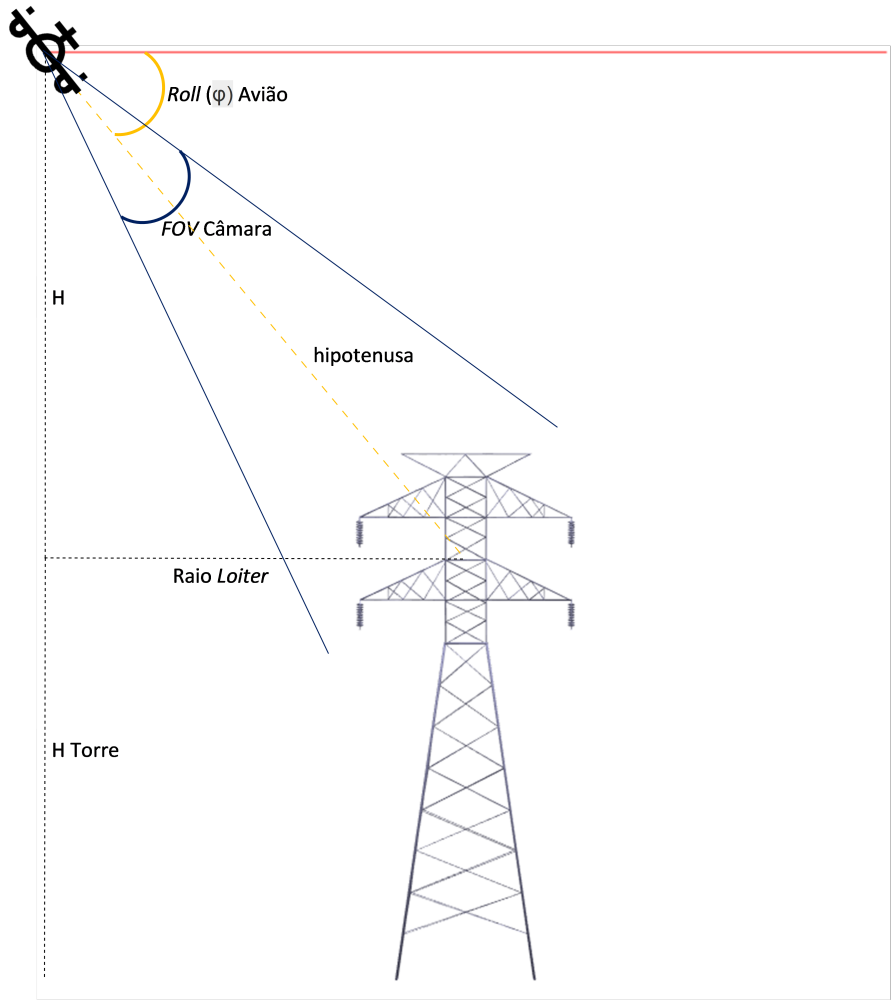


Figura 5.20: Planificação de inspeção avião em *Loiter*, vista lateral

Por fim, pode ser verificado se a área de interesse de inspeção está contida na imagem da câmara, através da equação 5.2, assumindo um FOV vertical e horizontal igual.

5.4 *Firmwares* da controladora de voo

O controlador de voo (*autopilot*) é constituído por um micro controlador, sensores e atuadores cruciais para o voo. Este apresenta um controlo de baixo nível, estando diretamente relacionada com o *output* dos motores, com o acelerómetro, magnetómetro, sensor *airspeed*, recetor de *Global Navigation Satellite System* (GNSS), entre outros.

Na área de robótica aérea, existem dois *firmwares* muito comuns para o *autopilot*, o *Ardupilot* e o *PX4*. Ambos apresentam vantagens e desvantagens em relação ao outro, no entanto, tem em comum a integração com o ROS. De seguida serão abordados alguns dos aspetos relevantes de ambos os *firmwares* de modo a instaurar um elemento comparativo entre os mesmos.

5.4.1 Ardupilot

O *Ardupilot* é um *firmware open-source*, que surgiu de uma prova de robótica aérea na Austrália, a prova "*Outback challenge*". Desde então, o *Ardupilot* foi desenvolvido continuamente, sendo que atualmente se encontra dividido em várias subcategorias:

- *Arducopter* - *firmware* específico para *multicotores*;
- *Arduplane* - específico para *fixed-wing* e VTOL;
- *Audurover* - específico para veículos terrestres;
- *Ardusub* - indicado para veículos subaquáticos.

As subcategorias com maior desenvolvimento são as de *Arducopter* e *Arduplane*, uma vez que a origem do *firmware* foi a prova de robótica aérea, sendo o *subfirmware* indicado para VTOL o *Arduplane*. Estes *subfirmwares* apresentam *features* muito relevantes para o desenvolvimento de um robô aereo, entre as quais o *autotune* e o modo *assisted flight*.

O *autotune* consiste na calibração automática do controlador *Proportional Integral Derivative* (PID). O procedimento é efetuado em voo, sendo selecionado este modo de voo. É possível selecionar o eixo a calibrar, entre *roll*, *pitch* e *yaw*, podendo também ajustar a agressividade do procedimento. Já a calibração em si, consiste na variação propositada em cada um dos eixos, medindo o resultado através do acelerómetro, verificando assim a resposta do controlador. No final da calibração o utilizador pode efetuar um voo de teste dos novos parâmetros de PID e selecionar se pretende guardar os mesmos ou não.

O modo *assisted flight* é específico a VTOL e consiste na utilização dos motores de *multirotor* para auxiliar o modo de voo em *fixed wing*, aumentanto a sustentação da aeronave. Este modo não pode ser forçado, mas sim parametrizadas as situações nas quais este é ativo. Entre os parâmetros estão a altitude mínima a que o VTOL se pode deslocar e velocidade mínima.

Para além das características referenciadas, o *Ardupilot* também integra um modo de *SITL* para simulação. Este é geralmente utilizado a par com simuladores externos visuais, como por exemplo o *Gazebo*. Apesar de todo o desenvolvimento de funcionalidades, a integração oficial do simulador *Gazebo* com o *Ardupilot* não existe, sendo que esta foi feita por um utilizador alheio e disponibilizada em *open source*. Esta implementação apenas está feita para *multirotor* e *fixed wing*, não existindo o caso do VTOL.

5.4.2 PX4

O *firmware* PX4 é uma alternativa ao *Ardupilot*. Apesar de não ter funcionalidades de *autotune*, este já apresenta uma integração oficial de simuladores robóticos para todos os tipos de veículo que suporta. Tem integração total com o *Gazebo* e ROS, podendo ser criado um mundo de simulação dedicado ao contexto de aplicação.

O PX4 é também o *firmware* utilizado no LSA pela equipa de drones. Já existem várias *drivers* de sensores específicos utilizados no laboratório, que podem ser usadas também para VTOL.

5.4.3 Comparação entre Ardupilot e PX4

Para ser possível efetuar uma comparação justa, no decurso do ano letivo, foram estudados e utilizados os dois *firmwares*. Após a experiência "*hands on*"em ambos, foi possível efetuar a comparação relatada na tabela 5.3. Dada a dimensão deste projeto e a origem do mesmo (disciplina de LASMU), este estudo e comparação foi feita em conjunto com a aluna Diana Salgado.

O modo híbrido consiste na utilização dos motores de *multirotor* e *fixed wing* em simultâneo, de modo a auxiliar a manobrilidade do veículo. Na disciplina de LASMU foi desenvolvido um protótipo para o voo híbrido, no entanto nunca foi testado em campo.

No caso da implementação de comandos Mavros, o PX4 é consideravelmente superior, uma vez que suporta a maioria dos comandos de mavlink em VTOL. Contrariamente o *Ardupilot* suporta poucos comandos, como é possivel constatar em [61].

Tabela 5.3: Comparação entre firmwares de PX4 e Ardupilot

PX4	**Ardupilot**
Não possui modo híbrido implementado	Modo híbrido implementado pelo grupo em LASMU (não testado em cenário real)
Com implementação de simulador Gazebo nativa	Sem implementação oficial, não existindo uma implementação de VTOL totalmente funcional
Controlo em posição e velocidade através do ROS implementado	Apenas implementado controlo em posição
Com desenvolvimento ativo	Sem desenvolvimento ativo, apenas adaptadas funcionalidades de avião
Sem Autotune	Com Autotune
Drivers de sensores desenvolvidas no laboratório, já existindo conhecimento sobre o tópico	Sem drivers para os sensores do laboratório e com bugs em alguns sensores genéricos
Suporta vários comandos de Mavros em Offboard	Suporta comandos de Mavros limitados em modo Guided
Interface de calibração de sensores simples e intuitiva	Interface confusa, com pouca documentação

5.4.4 Decisão do *Firmware*

Após o estudo, utilização e implementação dos dois *firmwares*, foi concluído que o *firmware* mais relevante para o contexto deste projeto é o PX4.

O fator com mais peso nesta escolha foi o facto do PX4 conter implementado nativamente o simulador *Gazebo*. A componente de simulação é imprescindível, sendo uma das partes mais relevantes, uma vez que é um tipo de veículo (VTOL) pouco estudado. A simulação de vários comandos, comportamentos de transição, interferência do vento e mais factores relevantes ao voo, são cruciais para o desenvolvimento do projeto final.

Para além da implementação em *Gazebo*, o facto de já existir "*know how*"no laboratório, *drivers* desenvolvidas, e a maioria dos comandos *Mavros* suportados nativamente, são fatores muito importantes para o projeto e consecutivamente para a decisão tomada.

5.5 Arquitetura detalhada do sistema

Após o estudo detalhado das soluções apresentadas anteriormente e tomadas as respetivas decisões, surgiu um novo esquema para a arquitetura do projeto, representado na Figura 5.21.

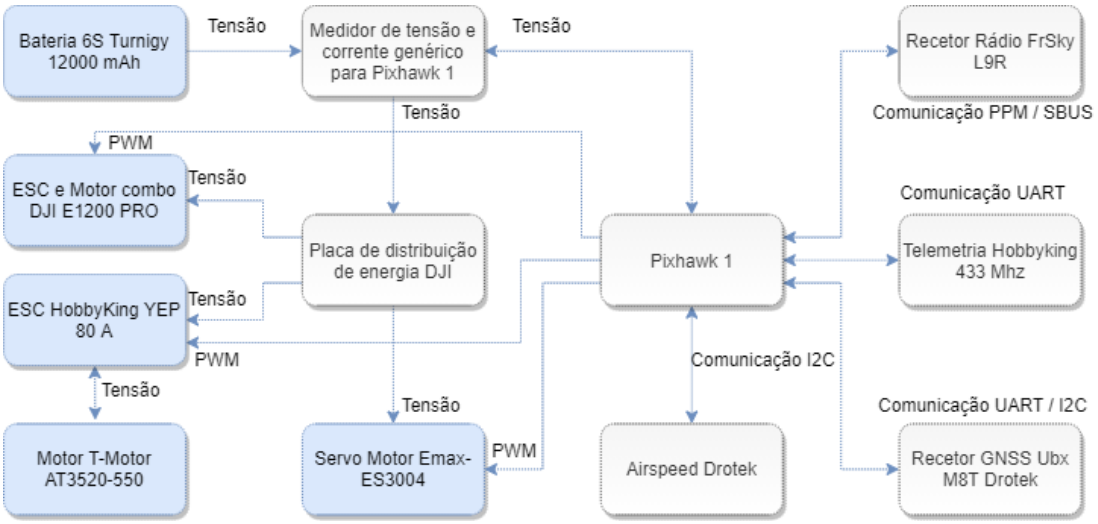


Figura 5.21: Arquitetura detalhada do sistema

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Implementação

Neste capítulo irão ser demonstrados os resultados obtidos através do desenvolvimento dos tópicos abordados no capítulo 5. Serão também abordados alguns tópicos desenvolvidos no projeto que não constam no capítulo 5, uma vez que foram desenvolvidos sem base de decisão. Estes tópicos serão a implementação da simulação em *Software In The Loop* (SITL) da aeronave e a adição de uma câmara ao modelo. Além disso será abordada a criação do mundo *Gazebo* e implementação de vento no mesmo. Como tal, serão clarificados os processos de implementação dos vários elementos do projeto.

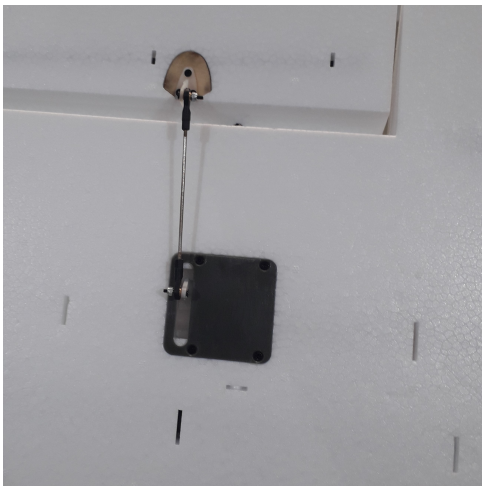
6.1 Assemblagem do VTOL

O processo de desenvolvimento e assemblagem do VTOL foi um trabalho partilhado com a colega Diana Salgado. Ao longo do projeto a divisão foi sempre clara, sendo o foco desta metade do projeto a montagem dos componentes mecânicos e propulsão.

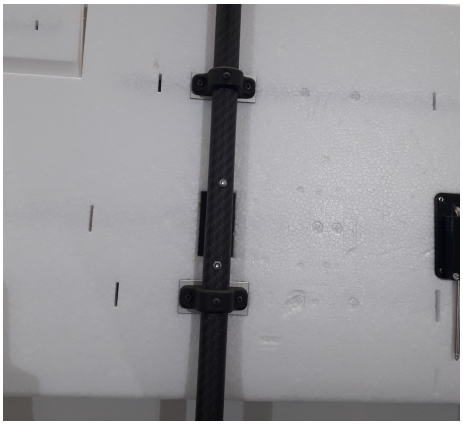
6.1.1 Estrutura mecânica do VTOL

A Figura 6.1 representa diferentes etapas na montagem do VTOL. A montagem da estrutura consistiu na colagem de várias peças com uma cola específica para espuma, na montagem dos componentes mecânicos e na ligação elétrica dos mesmos. A Figura 6.1(a) demonstra a montagem do motor DJI E1200 PRO na barra de carbono da estrutura enquando que a Figura 6.1(b) representa a montagem do servo motor Emax-ES3004 na asa da *frame*. A Figura 6.1(c) demonstra a ligação da barra de carbono à asa e o conector da mesma e a Figura 6.1(d) demonstra a ligação da asa ao corpo principal do avião.

(a) Montagem do motor de *multirotor* na barra de carbono



(b) Montagem do servo motor na asa



(c) Ligação das barras de carbono à asa

(d) Ligações da asa à parte central da aeronave

Figura 6.1: Montagem do VTOL, parte mecânica

6.1.2 Peças 3D desenvolvidas

Aquando do desenvolvimento do VTOL foi necessário desenhar algumas peças 3D para as mesmas pudessem ser aplicadas na aeronave. Entre as peças desenvolvidas destaca-se o trem de aterragem, que sofreu várias iterações, e suporte da bateria.

6.1.2.1 Suporte da bateria

O suporte de bateria foi projetado para que, em caso de queda, esta ficasse sempre segura à estrutura. Foi então projetada uma peça que cola à parte interior da estrutura central,

e que contém locais apropriados para passar fitas de *velcro* para prender a bateria a essa peça. A Figura 6.2 representa a projeção da peça e a Figura 6.3 representa a aplicação na estrutura.

Figura 6.2: Desenho tridimensional do suporte da bateria

Figura 6.3: Aplicação do suporte da bateria à estrutura

6.1.2.2 Trem de aterragem

O desenvolvimento do trem de aterragem foi um processo contínuo que sofreu várias iterações. O objetivo seria elevar a aeronave em relação ao chão uma vez que, na aterragem, o *propeller* de *fixed wing* poderia colidir com o chão.

A primeira iteração, demonstrada na Figura 6.4, consistiu no desenho de uma peça que aparafusasse no suporte de motores das barras de carbono e incluísse uma extensão para um tubo de carbono. Esta peça seria produzida numa impressora 3D, daí a sua grossura e aparente robustez. Apesar dos esforços para criar robustez na peça, tal não se verificou, sendo que a zona onde aparafusava às barras de carbono partiu na primeira aterragem.

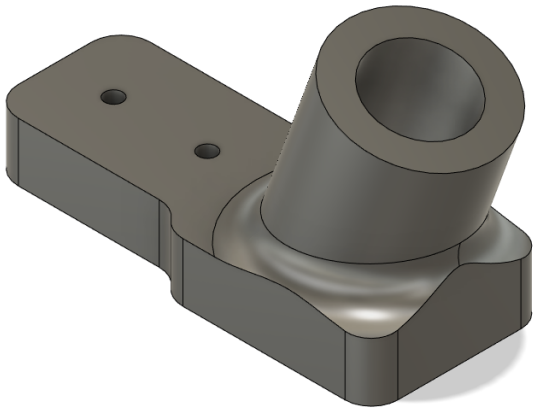


Figura 6.4: Trem de aterragem, primeira iteração

A segunda iteração, representada na Figura 6.5, foi um sucesso na resolução dos problemas da primeira. De forma a evitar a rotura na zona onde aparafusava, a peça foi desenhada de modo a ser parcialmente produzida em alumínio. A zona que prendia às barras de carbono passou a abraçar as mesmas a ser maquinada numa CNC. No entanto, esta peça partiu na zona onde acoplava o tubo de carbono, que ainda era feita de plástico.

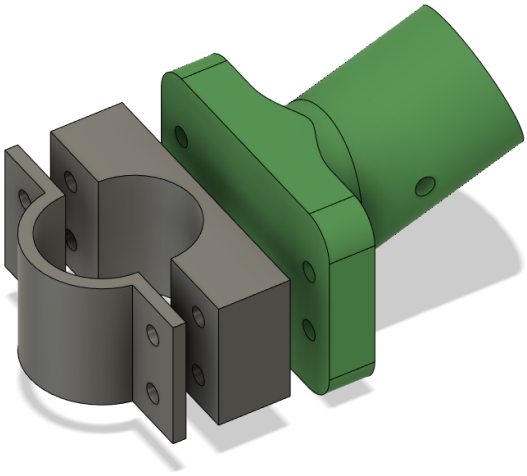
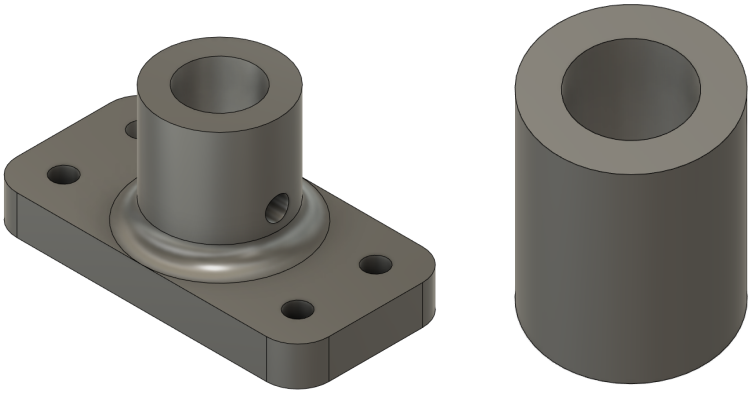


Figura 6.5: Trem de aterragem, segunda iteração

A terceira iteração adota o desenho da primeira, no que toca à junção com as barras de carbono, no entanto é toda feita em alumínio. Foi também adicionado um sistema de amortecimento, recorrendo as rodas e respetiva estrutura apresentadas no capítulo 5. Para além disso, os tubos passaram para uma grossura menor de modo a criar um ponto de rotura propositado, sendo que seria o ponto de falha em caso de queda. A Figura 6.6(a) representa o desenho da nova peça em alumínio e a Figura 6.6(b) representa a junção entre o tubo de alumínio e as rodas de aterragem.



(a) Peça de junção as barras de carbono (b) Peça de junção entre o tubo de alumínio e as rodas

Figura 6.6: Trem de aterragem, terceira iteração

6.2 Ambiente de simulação

Um ambiente de simulação é uma das ferramentas mais importantes aquando do desenvolvimento de um robô. No caso de um veículo aéreo, é ainda mais relevante, uma vez que a queda do mesmo pode resultar em danos materiais de elevada escala. Assim, é importante detalhar o desenvolvimento do ambiente de simulação utilizado neste projeto. O simulador escolhido foi o *Gazebo*, uma vez que é muito utilizado em robótica e possui simulação de dinâmica e cinemática. Além disso, existe uma integração direta no *firmware* utilizado com o Gazebo, recorrendo a *plugins* para cálculo de parâmetros importantes, tais como a sustentação das asas.

6.2.1 Ambiente de simulação - Mundo *Gazebo*

Considerando o objetivo de validar diferentes abordagens de inspeção de ativos elétricos, em particular postes de média e alta tensão, procedeu-se ao desenvolvimento de um cenário de simulação que integra esses mesmos ativos. Foram criados modelos de torres de transmissão de média tensão, para posterior inserção no mundo, tal como demonstra a Figura 6.7.

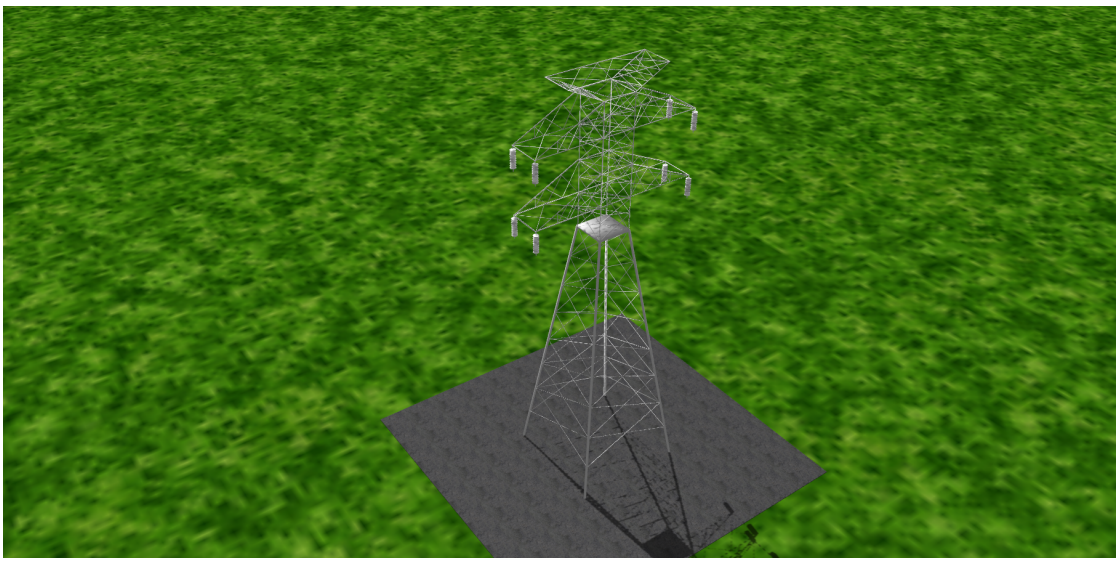


Figura 6.7: Modelo *Gazebo* da torre de média tensão

O cenário de simulação consiste na existência de 6 torres de transmissão de linhas de média tensão, todas espaçadas 200 m entre si. As coordenadas $[X, Y, Z]$ das torres estão

descritas na tabela 6.1. O resultado final do mundo está ilustrado na Figura 6.8.

Tabela 6.1: Coordenadas das torres de transmissão de média tensão no mundo do Gazebo

Modelo	**X**	**Y**	**Z**
Torre 0	0	-50	0
Torre 1	200	-50	0
Torre 2	400	-50	0
Torre 3	600	-50	0
Torre 4	800	-50	0
Torre 5	800	-250	0

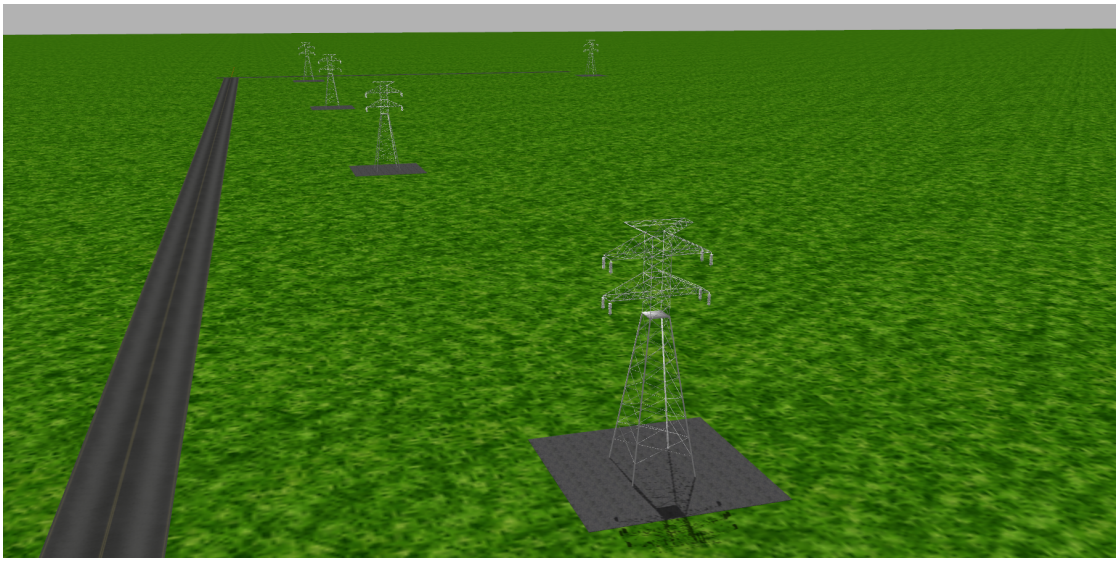


Figura 6.8: Cenário de simulação com os ativos elétricos em *Gazebo*

6.2.2 Integração de vento no ambiente simulado

No sentido da simulação ser o mais realista possível, efetuou-se a integração do *plugin* de vento, com o objetivo de avaliar o desempenho do VTOL para diferentes valores e direções de vento. Assim, foi inserido um *plugin* do *gazebo* de vento que interage com as superfícies de controlo e com o modelo. O *plugin* em questão é o "*libgazebo_wind_plugin.so*"e interage na *frame* como uma *força* e deve ser inserido no ficheiro do mundo.

Através deste *plugin* pode ser controlada a direção do vento, velocidade do vento constante, velocidade de rajadas temporárias e a respetiva duração. A parametrização

é feita tal como evidenciada abaixo:

```
<plugin name='wind_plugin' filename='libgazebo_wind_plugin.so'>
  <frameId>base_link</frameId>
  <robotNamespace/>
  <xyzOffset>1 0 0</xyzOffset>
  <windDirectionMean>0 1 0</windDirectionMean>
  <windVelocityMean>4.0</windVelocityMean>
  <windGustDirection>0 0 0</windGustDirection>
  <windGustDuration>0</windGustDuration>
  <windGustStart>0</windGustStart>
  <windGustVelocityMean>0</windGustVelocityMean>
  <windPubTopic>world_wind</windPubTopic>
</plugin>
```

O parâmetro *windDirectionMean* corresponde ao vetor $[X, Y, Z]$ de direção do vento, o *windVelocityMean* corresponde à velocidade média constante do vento, em metros por segundo. O resultado da implementação é possível ser verificado através de um teste de voo, com um simples comando para manter a posição no ar (*hover*). O comando *hover* consiste na manutenção da aeronave na posição atual sendo que com vento, é possível verificar uma inclinação da aeronave para manutenção da posição. A manutenção da posição com vento é possível ser verificada através da Figura 6.9.

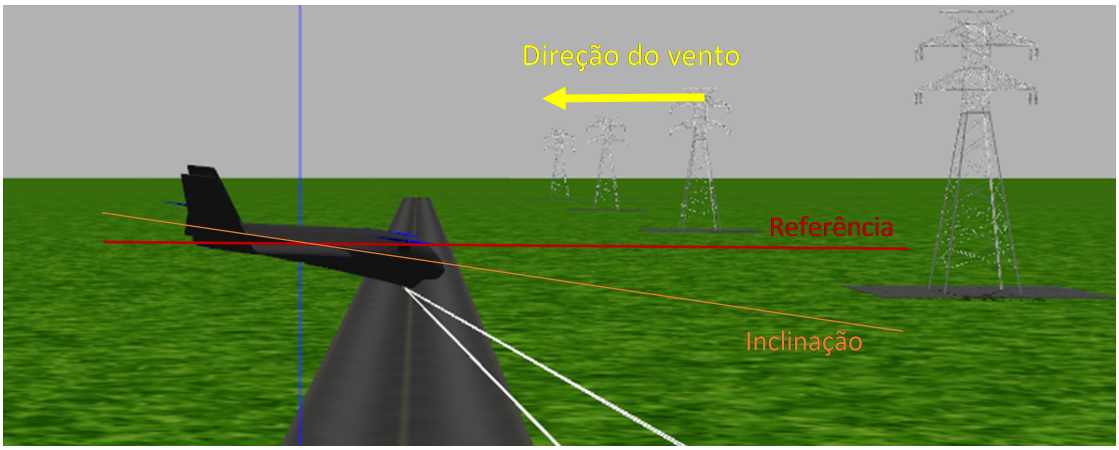


Figura 6.9: *Plugin* de vento aplicado ao VTOL em *Gazebo*

6.2.3 Modelo de simulação da aeronave

O modelo de simulação utilizado para a aeronave foi o "*Standard VTOL*"disponibilizado pela equipa do PX4, implementado em *Gazebo*. Este segue uma estrutura semelhante à *frame SkywalkerX8* uma vez que esta era bastante utilizada quando o desenvolvimento dos VTOL começou em PX4. Apesar de ser asa delta e o modelo real ser asa convencional, o modelo possui as mesmas capacidades, com excepção de ter um coeficiente de sustentação maior, intrínseco da asa delta. Para efeitos de comparação, é também possível atribuir massas diferentes aos diversos componentes da *frame* e consequentemente da estrutura completa. A Figura 6.10 ilustra o modelo utilizado.

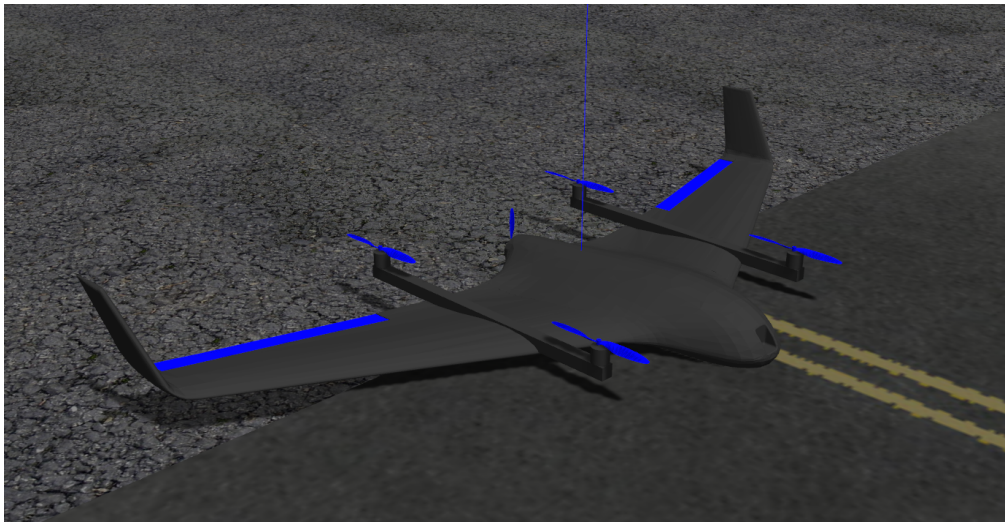


Figura 6.10: Modelo da aeronave utilizado no *Gazebo*

6.2.4 Integração de uma câmara no UAV simulado

Para efetuar a simulação da inspeção é necessário a integração de um sistema de visão. Assim, foi estudado e inserido um modelo de uma câmara no modelo da aeronave "*Standard VTOL*". O modelo em questão permite a configuração do FOV vertical e horizontal, o *frame rate* da câmara e a sua resolução. Os parâmetros utilizados foram definidos de modo a corresponder com os inseridos na planificação das missões e correspondem aos valores da tabela 6.2. O modelo tira partido da integração do *gazebo* do PX4 e envia as imagens para a *ground station* e publica também as imagens para um tópico de ROS.

Tabela 6.2: Parâmetros da câmara introduzidos no modelo

Parâmetro	Valor
FOV Vertical	60^{o}
FOV Horizontal	60^{o}
Resolução	1920 x 1080 pixeis
Frame rate	5 FPS

6.3 Manobra de inspeção autónoma de ativos elétricos

A manobra desenvolvida para inspeção de ativos elétricos respeita o descrito no capítulo 5 na secção 5.3. O algoritmo tem como parâmetros de entrada o número de isoladores, as coordenadas dos mesmos, o ângulo entre a torre e a linha de transmissão de cada lado, a altura da torre, o *pitch* e o FOV da câmara, a distância mínima de segurança em relação ao ativo e por fim o raio desejado. É também possível selecionar se pretende uma inspeção detalhada que consiste na inspeção local individual de um só isolador ou uma manobra combinada que consiste na inspeção agrupada de dois isoladores. Nos tópicos seguintes apenas será explicado o procedimento para a manobra detalhada. A manobra desenvolvida segue a lógica do algoritmo 4.

Algoritmo 4 Algoritmo da manobra de inspeção de drone desenvolvida

```
Get data inputs
Calculate_restricted_area()
Calculate_local_inspection_points()
Calculate_global_inspection_points()
InspectionPoints = local_inspection_points + global_inspection_points
for all InspectionPoints do
    if InspectionPoint ⊂ restricted_area then
        Delete_Point()
    end if
    if Point is invalid (camera properties) then
        Delete_Point()
    end if
end for
Sort_Points()
Plot_path()
Write_to_file()
```

6.3.1 Cálculo da área restrita

Após a introdução de todas as variáveis, o algoritmo começa por calcular a área restrita. Esta área consiste na zona que está preenchida por linhas de transmissão, logo a aeronave não se pode aproximar. A área é calculada através das coordenadas dos isoladores e da distancia de segurança, sendo computados pontos extremos, Ext_i, com recurso à equação 6.1, e em seguida aplicada a distância de segurança desse ponto. A junção de todos os pontos computados gera um polígono correspondente à área restrita.

$$\begin{cases} Isol_i = [A, B, C] \\ Ext_{ix} = \frac{cos(angulo\_linha)}{L} + A \\ Ext_{iy} = \frac{sen(angulo\_linha)}{L} + B \\ Ext_{iz} = C \end{cases} \tag{6.1}$$

As Figuras 6.11, 6.12(a) e 6.12(b) demonstram o resultado do cálculo das áreas restritas para vários ângulos das linhas de transmissão.

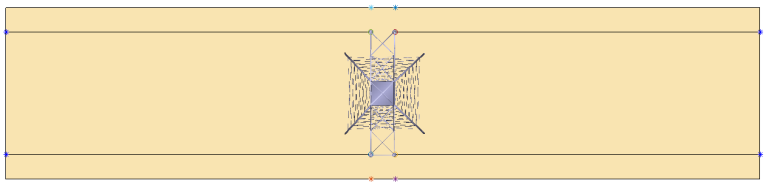
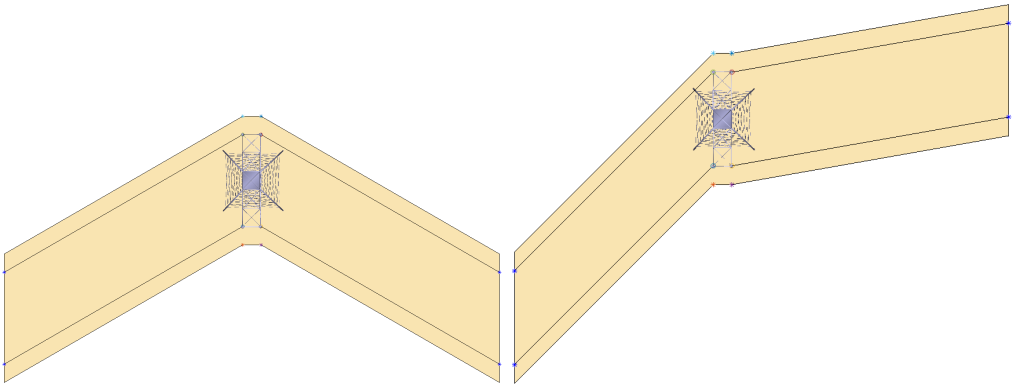


Figura 6.11: Área restrita, com ângulos nulos, vista de cima



(a) Área restrita, com ângulos de 30° e −30° (b) Área restrita, com ângulos de 45° e 10°

Figura 6.12: Área restrita, com vários ângulos, vista de cima

6.3.2 Cálculo dos pontos de inspeção local e global

Os pontos de inspeção são calculados de forma a efetuarem um círculo em torno do ponto de interesse. No caso dos pontos locais o circulo i é calculado em torno do isolador i, enquanto que os pontos globais tem como ponto de interesse o centro da torre.

A par do cálculo dos pontos é efetuado também o cálculo do *yaw* que a aeronave deve adotar, de modo a ficar com a câmara a apontar para o ponto de interesse. Todo este procedimento é relatado pela equação 6.2.

$$\begin{cases} Ponto\_de\_interesse_i = [A, B, C] \\ 0 \leq t \leq 10 \\ X_i = raio\_juncoes * cos(t) + A \\ Y_i = raio\_juncoes * sin(t) + B \\ Z_i = C + \frac{raio\_juncoes}{tan(cam\_pitch)} \\ Yaw_i = atan2(\frac{B-Y_i}{A-X_i}) \end{cases} \tag{6.2}$$

6.3.3 Validação do ponto

A validação do ponto implica duas fases, numa primeira é verificado se o ponto está contido na área restrita, caso esteja, o ponto é apagado. Numa segunda fase é verificado se o ponto é válido através da projeção da imagem da câmara. Para o ponto ser válido é necessário que a imagem da câmara contenha o ponto de interesse no seu campo de visão vertical e horizontal. A equação 6.3 representa o método de cálculo utilizado para pontos locais e a equação 6.4 para pontos globais.

$$\begin{cases} D_{x\_min} = Comprimento\_isolador \\ D_{z\_min} = Altura\_isolador \\ H\_FOV_{cam} = 2 * hipotenusa * \frac{FOV}{2} \\ V\_FOV_{cam} = H\_FOV_{cam} \\ H\_FOV_{cam} > D_{x\_min} \\ V\_FOV_{cam} > D_{z\_min} \end{cases} \tag{6.3}$$

$$\begin{cases} D_{x\_min} = |Isol_{ix} - Isol_{i+1x}| \\ D_{z\_min} = |Isol_{iz} - Isol_{i+2z}| \\ H\_FOV_{cam} = 2 * hipotenusa * \frac{FOV}{2} \\ V\_FOV_{cam} = H\_FOV_{cam} \\ H\_FOV_{cam} > D_{x\_min} \\ V\_FOV_{cam} > D_{z\_min} \end{cases} \tag{6.4}$$

6.3.4 Ordenação dos pontos

Após a verificação de todos os pontos, é iniciada a ordenação dos mesmos. Todos os pontos contém informação $[X, Y, Z, YAW, ZONA]$ onde $ZONA$ corresponde ao lado em que o ponto se encontra. A manobra é organizada de modo a nunca passar pelos locais da área restrita, mesmo na transição entre pontos de inspeção local e global, através do cálculo de pontos de fuga. Os pontos são organizados de modo a inspecionar pela seguinte ordem:

- Inspeção global lado direito;
- Inspeção local de todos os isoladores do lado direito;
- Inspeção global lado esquerdo;
- Inspeção local de todos os isoladores do lado esquerdo.

Os pontos de fuga consistem no momento em que ocorre uma transição entre o lado direito e o esquerdo da torre e no momento final de inspeção, para garantir que não existe colisão com a torre. Por fim, a ordem dos pontos entre as mesmas zonas é calculada através da menor distância entre pontos, de modo a assegurar o menor custo possível para a manobra em, questão.

6.3.5 Resultado final

No resultado final, o algoritmo gera os *plots* para que o utilizador consiga visualizar o planeamento de trajetória que o VTOL poderá executar, bem como a orientação que deve ter num dado ponto. É também exportado para um ficheiro de texto, *.txt*, todos os pontos ordenados e a sua respetiva orientação em *yaw*. A Figura 6.13 demonstra o *plot* final da manobra detalhada e a Figura 6.14 demonstra o exemplo da manobra combinada descrita no inicio deste tópico.

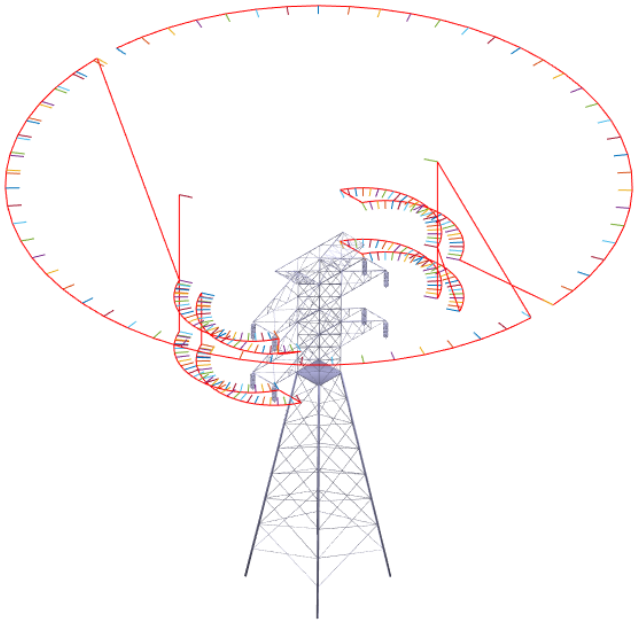


Figura 6.13: Resultado da inspeção detalhada

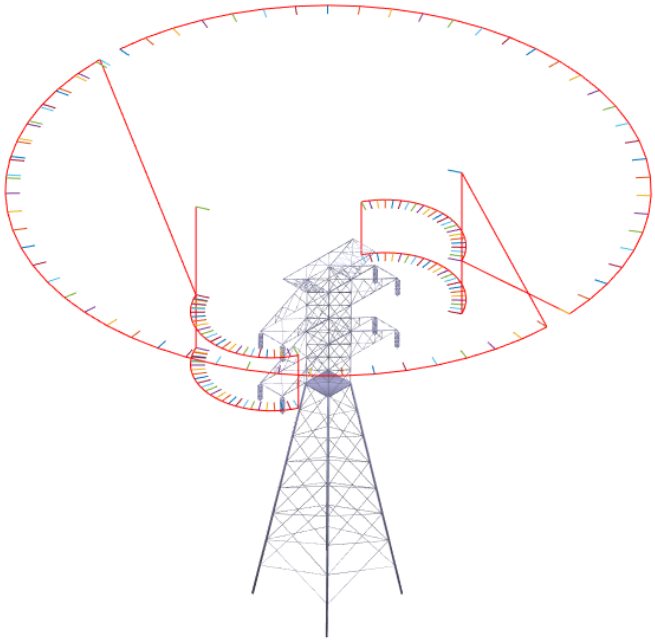


Figura 6.14: Resultado da inspeção combinada

6.4 Planeamento de missões

Na sequência do desenvolvimento do projeto foi necessário criar um *standard* para o planeamento de uma missão. Para qualquer tipo de missão, a lógica deveria ser idêntica, sendo que o utilizador apenas teria de inserir os parâmetros num ficheiro de texto para o algoritmo de inspeção gerar a trajetória planeada. Assim, foi desenvolvido um protocolo de criação de missão, que segue a estrutura representada na Figura 6.15.

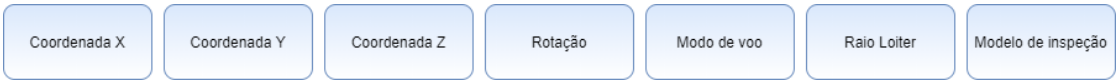


Figura 6.15: Estrutura de preenchimento do ficheiro de texto

- Coordenadas $[X, Y, Z]$- corresponde às coordenadas locais desejadas (*waypoint*) ou, como função alternativa, às coordenadas locais do objeto de inspeção;
- Rotação - corresponde ao ângulo em *yaw* desejado, medido em radianos $[-\pi, \pi]$, para o *waypoint* em questão. Também pode corresponder, como função alternativa, à rotação de um objeto de inspeção, medido em graus [-360,360]. Como função terciária, é usado para indicar o *pitch* máximo para uma manobra de parafuso;
- Modo de voo - corresponde ao modo de voo desejado, 0 para *multirotor* e 1 para *fixed wing* quando chega ao *waypoint*. Pode também ser 2, indicando um pedido de uma manobra de parafuso;
- Raio *loiter* - caso seja diferente de 0, corresponde ao raio desejado para o *loiter*. Caso o valor seja 0 indica que é apenas um *waypoint* e não deve ser efetuado um *loiter*;
- Modelo de inspeção - Corresponde ao nome de um ficheiro de texto externo que contém as coordenadas de inspeção de um modelo. Caso este parâmetro esteja ativo, (diferente de 0), serão ativadas as funções alternativas dos campos anteriores.

6.4.1 Exemplos de aplicação do protocolo de missões

6.4.1.1 Waypoint em *multirotor*

Caso o utilizador deseje efetuar um *waypoint* em modo de *multirotor* deve introduzir os parâmetros representados na Figura 6.16. Os parâmetros $[X, Y, Z]$ correspondem às coordenadas locais do *waypoint*, rotação corresponde ao *yaw* desejado nesse ponto. O modo de voo deve ser 0, indicando ao planeador que quando chegar ao ponto, a aeronave deve estar em modo *multirotor*. Os restantes parâmetros devem ser 0.

Coordenada X	Coordenada Y	Coordenada Z	Rotação	Modo de voo	Raio Loiter	Modelo de inspeção
200	0	50	1.5708	0	0	0

Figura 6.16: Exemplo de aplicação do protocolo para um *waypoint* em *multirotor*

6.4.1.2 Waypoint em *fixed wing*

Para efetuar um *waypoint* em *fixed wing* o utilizador deve inserir os parâmetros representados na Figura 6.17. As únicas diferenças em relação à mensagem de *multirotor* são que o parâmetro de rotação é ignorado, uma vez que está em voo de *fixed wing* e o modo de voo é 1, indicando o voo em *fixed wing* na chegada ao *waypoint*.

Coordenada X	Coordenada Y	Coordenada Z	Rotação	Modo de voo	Raio Loiter	Modelo de inspeção
200	0	50	0	1	0	0

Figura 6.17: Exemplo de aplicação do protocolo para um *waypoint* em *fixed wing*

6.4.1.3 *Loiter* em torno de um ponto

A lógica inserção de um comando de *loiter* é apenas uma adição ao comando de *fixed wing*. Com excepção do parâmetro *RaioLoiter*, que deve ser igual ao raio desejado, em metros, em torno do ponto especificado $[X, Y, Z]$, os restantes parâmetros seguem a mesma lógica de um *waypoint* em *fixed wing*. Este procedimento está clarificado na Figura 6.18.

Coordenada X	Coordenada Y	Coordenada Z	Rotação	Modo de voo	Raio Loiter	Modelo de inspeção
200	0	50	0	1	50	0

Figura 6.18: Exemplo de aplicação do protocolo para um *Loiter*

6.4.1.4 Manobra de parafuso em torno de um ponto

A manobra de parafuso consiste num voo em *loiter* em torno de um ponto, mas com variação de altitude. Como tal, é necessário inicialmente enviar ao planeador um comando de *loiter*, com a altitude inicial desejada do parafuso. De seguida deve ser enviada uma instrução semelhante, com as mesmas coordenadas $[X, Y]$ mas com um Z diferente, correspondente à altitude final do parafuso. De forma a indicar que a manobra é um parafuso controlado e não uma simples alteração de altitude, é necessário indicar um modo de voo 2. Mais ainda, o parâmetro rotação passa a indicar o *pitch* máximo que a aeronave pode assumir, controlando assim o passo do parafuso. Este procedimento está clarificado na Figura 6.19.

Coordenada X	Coordenada Y	Coordenada Z	Rotação	Modo de voo	Raio Loiter	Modelo de inspeção
200	0	100	0	1	50	0
200	0	50	5	2	50	0

Figura 6.19: Exemplo de aplicação do protocolo para um parafuso

6.4.1.5 Manobra de inspeção de um modelo externo

No caso da manobra de inspeção de um modelo externo, ou seja, uma manobra específica criada num ficheiro de texto à parte, contendo os pontos de inspeção, a lógica de inserção é completamente diferente. Esta opção foi criada de forma a que pudessem ser introduzidas várias manobras de inspeção independentes na mesma missão global. Assim, é possível inserir uma manobra de inspeção de uma torre com as linhas de transmissão dispostas com diferentes ângulos, vários modelos de torres e vários tipos de manobras.

O modelo de inspeção indica o nome do ficheiro onde está contida a manobra de

inspeção do modelo em específico. Os parâmetros de modo de voo e raio *loiter* são ignorados, no entanto os parâmetros $[X, Y, Z]$ passam a ser as coordenadas onde o modelo está centrado no mundo. Tal como o nome indica, rotação é o parâmetro que indica a rotação do modelo em *yaw* no mundo. Este procedimento está clarificado na Figura 6.20.

Coordenada X	Coordenada Y	Coordenada Z	Rotação	Modo de voo	Raio Loiter	Modelo de inspeção
200	200	0	45	0	0	torre_ABC

Figura 6.20: Exemplo de aplicação do protocolo para um modelo de inspeção

6.5 Módulos de software desenvolvidos

Para efetuar o controlo da aeronave de forma autónoma, recorreu-se à utilização da *framework* ROS e *Mavros*. Foram criados 3 nós em ROS para para efetuar a gestão da missão, sendo apenas necessário a introdução da mesma num ficheiro de texto. Esta secção explica o conceito de cada nó e as suas funcionalidades. A Figura 6.21 demonstra a comunicação entre elementos do sistema implementado.

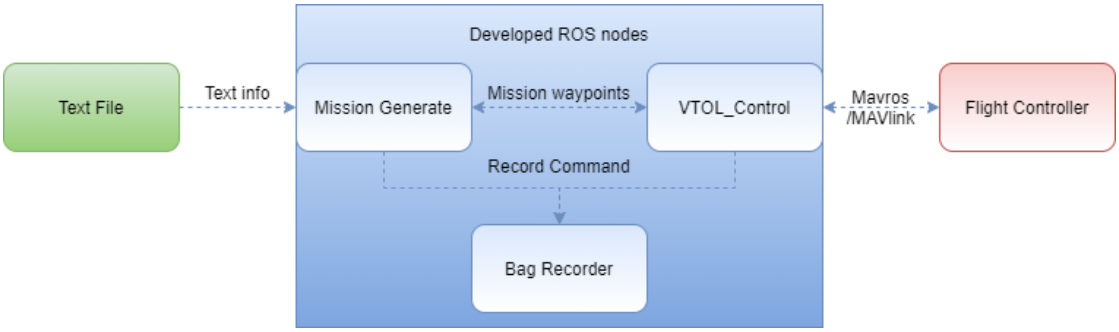


Figura 6.21: Comunicação entre elementos do sistema implementado

6.5.1 *VTOL control*

O nó principal do projeto é o *VTOL control*. Este é responsável pelo controlo da aeronave através do *mavros*. A aeronave apenas deve estar ligada e existir uma ligação entre o PC com o nó e o *autopilot*. Este nó baseia-se no modo de voo *offboard* do PX4, que permite

a que toda a informação de navegação de alto nível seja controlada pelo PC. Em modo *offboard* o *autopilot* apenas calcula o *output* dos motores e lê os sensores.

Este nó é portanto o mais complexo dos desenvolvidos, sendo que garante que todos os *waypoints* introduzidos pelo utilizador são concluídos. É também capaz de comandar transições entre *multirotor* e *fixed wing*, armar e desarmar os motores e fazer *takeoff*.

As funções implementadas neste nó são diversas, estando divididas por subscritores, publicadores, serviços, funções de uso geral e as funções de controlo de navegação. Assim, de seguida, serão apresentadas as funções mais relevantes, bem como uma breve descrição do seu objetivo.

- Funções dos subscritores
  - **state_cb(const mavros_msgs::State::ConstPtr& msg)**: responsável pelo *update* do estado da aeronave (ex. estado da conexão, *armed*/*disarmed*, modo de voo);
  - **void inspection_path_rec(const vtol_ctrl::vtol::ConstPtr& msg)**: responsável pela recepção dos *waypoints*, proveniente do nó *mission generate*;
  - **curr_local_pos_update(const geometry_msgs::PoseStamped::ConstPtr& msg)**: responsável pelo *update* da posição local indicada pelo *autopilot*;
  - **curr_global_pos_update(const sensor_msgs::NavSatFix::ConstPtr& msg)**: responsável pelo update da posição global indicada pelo recetor de GNSS do *autopilot*;
- Funções dos publicadores
  - **nav_local_pos_setpoint(geometry_msgs::Point point, float yaw)**: função que publica um *waypoint* local e a respetiva orientação em quaterniões. Utilizada para controlo em posição local;
  - **nav_global_pos_setpoint(geographic_msgs::GeoPoint ponto_global, float yaw)**: responsável por publicar um *waypoint* global e a respetiva orientação em quaterniões. Utilizada para controlo em posição global;
  - **nav_local_vel_setpoint(geometry_msgs::Point point, double theta_ref)**: responsável por publicar para o controlo em velocidade do PX4. Publica velocidade linear nos eixos X, Y, Z e angular no eixo Z;
  - **local_pos_loiter_target(geometry_msgs::Point point)**: responsável pelo envio dos parâmetros para a manobra de *loiter* ao PX4;

  - **nav_attitude_setpoint(float thrust, float pitch)**: responsável pelo envio de *setpoints* de atitude da aeronave. Atualmente apenas controla o *thrust* e *pitch*;

- Funções dos serviços

  - **request_arm(bool state)**: responsável por enviar o pedido de *arm/disarm* ao *autopilot* e receber resposta de sucesso ou insucesso;
  - **request_set_mode(std::string string_rec)**: responsável por enviar o pedido de alteração do modo de voo e receber resposta de sucesso ou insucesso;
  - **request_transition(uint8_t received)**: função capaz de enviar um pedido de transição entre *fixed wing* e *multirotor* e vice-versa;
  - **change_param(std::string string_rec, float value)**: função capaz de alterar qualquer parâmetro de configuração do voo (ex. raio do loiter, *roll* máximo, valores de PID);

- Funções de uso geral

  - **mavlink_quaternion_to_euler(const float quaternion[4], float* roll, float* pitch, float* yaw)**: função de conversão entre quaterniões e ângulos de *euler*;
  - **float get_distance(double x1, double y1, double z1, double x2, double y2, double z2)**: função para cálculo de distância aritmética entre dois pontos;
  - **calculate_setpoints(float des_vel, float des_alt, float* thrust_out, float* pitch_out)**: função de cálculo de pontos de atitude desejados;

- Funções de controlo de navegação

  - **go_to_position_pid(float x, float y, float z,float yaw_des)**: função que aplica controlo em velocidade. Recorre a um controlador PID para atingir a posição desejada. Apenas está preparada para pontos locais, no entanto a aplicação a pontos globais não é muito exigente;
  - **check_navigation_loiter(geometry_msgs::Point point_check, float raio_check)**: função responsável por garantir que a manobra de loiter é realizada corretamente e de forma controlada. Também é utilizada para a manobra de parafuso. Quando atingido o objetivo da função, esta faz o pedido de um novo *waypoint*;

  - **check_navigation_FW(geometry_msgs::Point point_check)**: responsável pela navegação de *waypoints* em *fixed wing*. Quando atingido o objetivo da função, esta faz o pedido de um novo *waypoint*;
  - **check_navigation_FW_MC(geometry_msgs::Point point_check,float yaw)**: responsável pela navegação em *fixed wing* até ao ponto e consequente transição para *multirotor* na aproximação ao *waypoint*. Quando atingido o objetivo da função, esta faz o pedido de um novo *waypoint*;
  - **check_navigation_MC(geometry_msgs::Point point_check,float yaw)**: função responsável pela navegação em *multirotor*. Geralmente recorre à função *go_to_position_pid*. De igual forma, concluído o objetivo da função, esta faz o pedido de um novo *waypoint*;

- Função de controlo de missão

  - **mission()**: Função principal do nó. Recebe o *waypoint* enviado pelo gerador de missões e delega o controlador a ser utilizado. Atualiza parâmetros de voo, decide a necessidade de efetuar transições e mantém o controlo *offboard* ativo.

Todas as funções anteriormente descritas foram implementadas e testadas em simulação. Tal como referido anteriormente, a função *mission()* é a função responsável por gerir a missão de inspeção e encontra-se clarificada no algoritmo 5.

Algoritmo 5 Algoritmo da função *mission()* do nó *VTOL control*

```
1: Request new waypoint
2: while flying do
3:    Wait for new waypoint
4:    if Flight mode = Fixed Wing OR Flight mode = Screw then
5:       Request Fixed Wing transition
6:       Restore default pitch and roll maximum values
7:       if Loiter radius > 0 then //Loiter or Screw flight modes
8:          if Flight mode = Loiter then
9:             Set loiter radius parameter
10:            check_navigation_loiter()
11:         else
12:            Set max Pitch parameter
13:            Set loiter radius parameter
14:            check_navigation_loiter()
15:         end if
16:      else
17:         check_navigation_FW() //Regular fixed wing waypoint
18:      end if
19:   else
20:      Calculate distance to waypoint
21:      if Fixed wing transition feasable then
22:         Request fixed wing transition
23:         check_navigation_FW_MC() //Fly to waypoint in FW and transition to MC
    when close
24:      else
25:         Request transition to multirotor
26:         check_navigation_MC() //Regular multirotor waypoint
27:      end if
28:   end if
29: end while
30: Terminate ROS node
```

6.5.2 *Mission Generate*

O segundo nó desenvolvido foi o *Mission Generate*. O nó tem o objetivo de efetuar a leitura do ficheiro de texto que contém a informação da missão e a traduzir para tópicos ROS. Este depende de um outro nó (*VTOL control*) que publique um pedido de *waypoint*. O algoritmo 6 descreve o processo do *mission create*.

Algoritmo 6 Algoritmo do nó *Mission Generate*

```
1: Initialize ROS publishers and subscribers
2: Open text file (mission waypoints)
3: while file not fully read (mission waypoints) do
4:    Read text line
5:    if line contains a model name then
6:       Open text file (model specific)
7:       while file not fully read (model specific) do
8:          Read text line
9:          Apply translation point and rotation to yaw
10:         Wait for request
11:         Send Waypoint
12:         Increment text line
13:      end while
14:      Close model file
15:   end if
16:   Wait for request
17:   Send Waypoint
18:   Increment text line
19: end while
20: Close mission waypoints file
21: Terminate ROS node
```

Inicialmente o algoritmo começa por ler o ficheiro de texto principal, preenchido pelo utilizador com a lógica descrita na secção anterior. O nó lê o ficheiro e uma linha, esperando pelo pedido de um novo *waypoint* por parte de outro nó. Depois de receber um pedido, este envia o *waypoint* e lê a linha seguinte.

Caso a linha contenha a informação de inspeção de um modelo externo (ex. gerado pelo algoritmo de *matlab*), então um novo ficheiro (com o nome do modelo a inspecionar) será aberto. A lógica de funcionamento é idêntica a um ponto introduzido pelo utilizador, no entanto após a leitura do ponto é aplicada a translação dos pontos, tal como descrito na secção anterior. Para o *yaw* é também aplicada uma rotação de acordo com a rotação da torre. Estes procedimentos são descritos pela equação 6.5

Após a leitura de todos os pontos, os ficheiros são encerrados e o nó ROS é terminado.

$$\begin{cases} WP_x = XC_{modelo} + X * cos(YAW_{modelo}) - Y * sin(YAW_{modelo}) \\ WP_y = YC_{modelo} + Y * cos(YAW_{modelo}) + X * sin(YAW_{modelo}) \\ WP_Z = ZC_{modelo} + Z \\ WP_{yaw} = YAW_{modelo} + YAW \end{cases} \tag{6.5}$$

6.5.3 Registo de dados da missão - *Bag recorder*

O último nó a ser desenvolvido foi o *Bag recorder*. Este tem como objetivo principal gravar num ficheiro *.bag* todos os dados que o utilizador desejar. Para o caso, este nó apenas está a gravar as imagens provenientes da câmara do gazebo e o respetivo *time stamp*. Toda a informação do *bag* gravado pode ser posteriormente utilizada para analisar resultados. O algoritmo 7 descreve o conceito do nó implementado.

Algoritmo 7 Algoritmo do nó *bag recorder*

```
1: Initialize ROS subscribers
2: Create bag file
3: Wait for start record command
4: while true do
5:    Update ROS subscribers
6:    if new image available then
7:       Write image and info to bag
8:    end if
9:    if stop record received then
10:      Close bag
11:      Break loop
12:   end if
13: end while
14: Terminate ROS node
```

O nó de ROS em questão subscreve ao tópico publicado pela câmara do *Gazebo* e a um tópico que indica quando este deve começar ou parar de gravar as imagens. Desta forma, pode ser iniciada a captação de imagem em momentos estratégicos para comparação de manobras ou outras situações relevantes.

Capítulo 7

Resultados

Neste capítulo serão demonstrados os resultados obtidos ao longo do desenvolvimento deste projeto. Inicialmente serão apresentadas alguns resultados com as relações de peso e vento, aplicados em voos da aeronave. Em seguida serão apresentadas comparações dos diferentes tipos de manobras. É importante salientar que os elementos anteriormente referidos foram resultados de simulação. Por fim, também serão demonstrados resultados do VTOL construído em voo.

7.1 Testes de vento e peso

Em condições reais as aeronaves são afetadas pelo vento a que estão expostas. Para efeitos de comparação, uma aterragem de um avião comercial de passageiros é considerada arriscada a partir dos 35 kn (knots) de vento horizontal, o que equivale a cerca de 65 km/h.

De modo a tornar a simulação o mais realista possível, foi aplicado vento no cenário de simulação e testados os efeitos no VTOL. Assim, foram efetuados testes de voo horizontais, que correspondem a vento lateral, testes de voo longitudinais, que correspondem a um voo a favor do vento na ida e contra o vento na volta e a testes de voo com o vento na diagonal. As Figuras 7.1, 7.2 e 7.3 representam exemplos dos voos horizontais, longitudinais e diagonais efetuados.

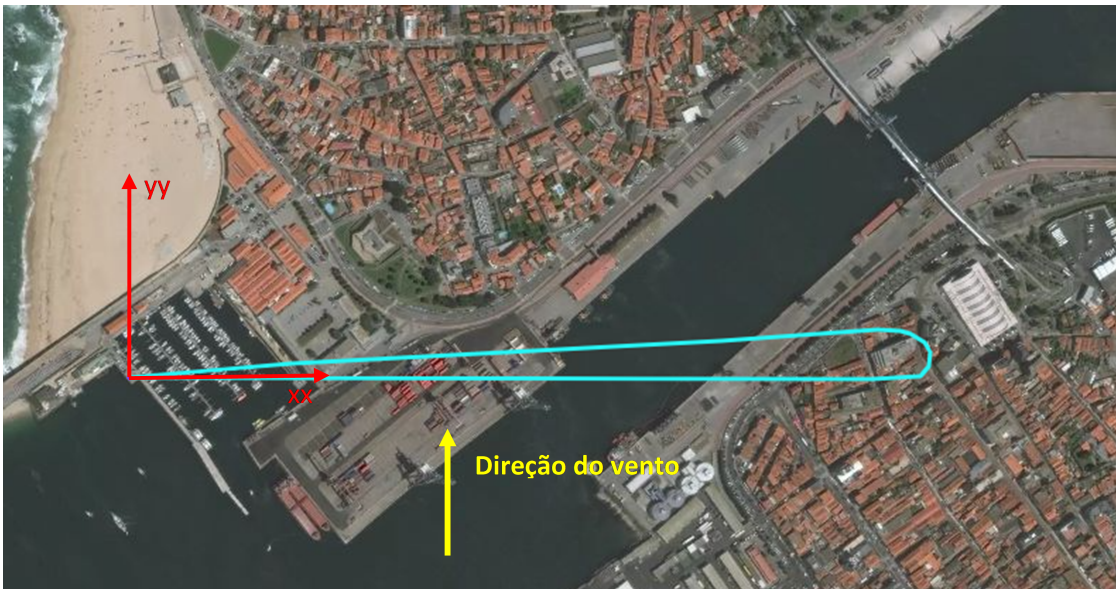


Figura 7.1: Exemplo de um voo horizontal efetuado

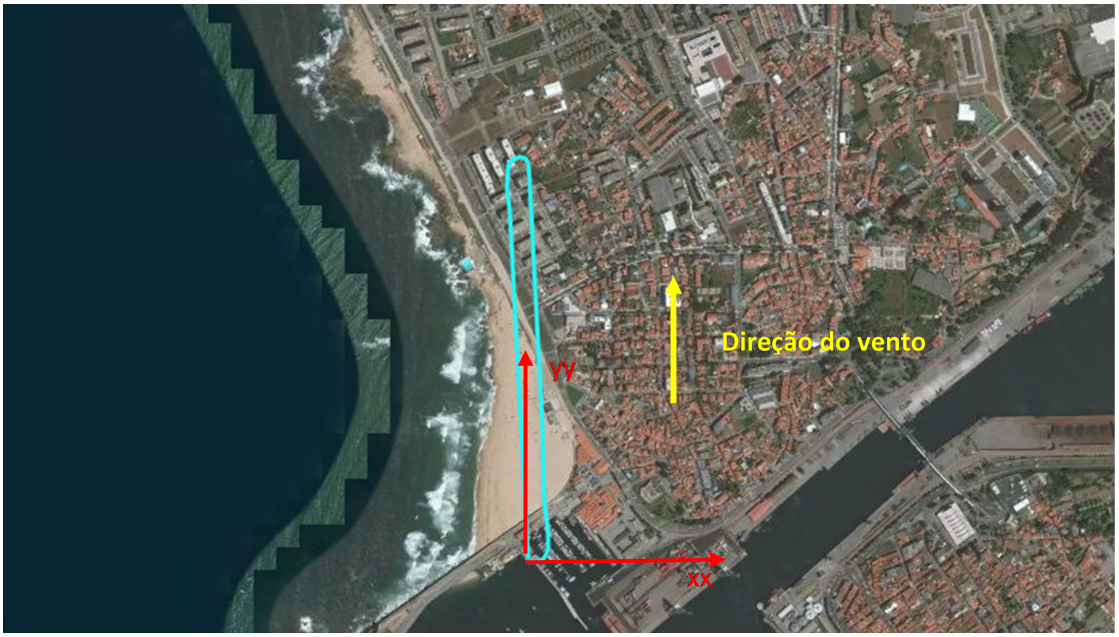


Figura 7.2: Exemplo de um voo longitudinal efetuado

Figura 7.3: Exemplo de um voo diagonal efetuado

Todos os testes de voo efetuados seguiram os mesmos pontos de controlo. Todos os voos horizontais foram desde a origem, subiram a 50 metros de altitude e em seguida até à posição $[X, Y, Z]$ de $[1000, 0, 50]$, em seguida fizeram a viagem de regresso. De igual forma, os voos horizontais e diagonais seguiram a mesma ordem de pontos, no entanto os pontos foram $[0, 1000, 50]$ e $[1000, 1000, 50]$, respetivamente. Para o primeiro teste o peso da aeronave foi constante e de um valor igual a 5 kg.

A variação de vento foi incremental até resultar na impossibilidade de realizar o voo com sucesso. O incremento entre voos foi de 5 m/s em cada iteração. Após a análise dos *logs* de voo, foram levantados os dados mais relevantes, estando estes representados pela Tabela 7.1. Apesar de o voo longitudinal ter sido concluído, este esteve a 1 m de bater no chão.

De modo a verificar a influência do peso da aeronave, foram também feitos testes para 10 kg de peso. Todos os restantes parâmetros foram variados conforme os testes anteriores. Os resultados para os testes efetuados com a aeronave de 10 kg estão representados na Tabela 7.2. É importante referenciar que esta não apresenta resultados para voos longitudinais acima de 36 km/h uma vez que a aeronave não conseguiu completar esses voos.

Tabela 7.1: Testes de voo, aeronave de 5 kg

Tipo de voo	Vel. do vento	Dist. perc.	Vel. máx.	T. de voo	Dif. de alt.
Horizontal	0 km/h	2,17 km	58,6 km/h	211,81 s	6.4 m
Horizontal	18 km/h	2,17 km	63,7 km/h	218,53 s	7.5 m
Horizontal	36 km/h	2,16 km	55,5 km/h	219,90 s	6.3 m
Horizontal	54 km/h	2,15 km	54,3 km/h	229,19 s	6.5 m
Horizontal	72 km/h	2,15 km	50,5 km/h	238,83 s	10.4 m
Longitudinal	0 km/h	2,21 km	58,6 km/h	220,82 s	6.2 m
Longitudinal	18 km/h	2,24 km	72,1 km/h	229,57 s	7.5 m
Longitudinal	36 km/h	2,28 km	82,8 km/h	222,92 s	27.2 m
Longitudinal	54 km/h	2,31 km	83,0 km/h	227,73 s	33.5 m
Longitudinal	72 km/h	2,58 km	87,4 km/h	244,91 s	54.2 m
Diagonal	0 km/h	2,99 km	60,7 km/h	264,99 s	5.8 m
Diagonal	18 km/h	3,02 km	68,2 km/h	290,57 s	7.3 m
Diagonal	36 km/h	3,05 km	78,9 km/h	272,72 s	14.1 m
Diagonal	54 km/h	3,06 km	82,8 km/h	271,65 s	38.4 m
Diagonal	72 km/h	3,07 km	83,0 km/h	266,53 s	41.4 m

Tabela 7.2: Testes de voo, aeronave de 10 kg

Tipo de voo	Vel. do vento	Dist. perc.	Vel. máx.	T. de voo	Dif. de alt.
Horizontal	0 km/h	2,22 km	75,0 km/h	213,45 s	20.2 m
Horizontal	18 km/h	2,23 km	75,5 km/h	211,75 s	17.0 m
Horizontal	36 km/h	2,24 km	76,0 km/h	215,66 s	25.0 m
Horizontal	54 km/h	2,24 km	75,8 km/h	212,68 s	27.9 m
Horizontal	72 km/h	2,24 km	76,0 km/h	218,94 s	29.6 m
Longitudinal	0 km/h	2,27 km	75,5 km/h	220,11 s	21.8 m
Longitudinal	18 km/h	2,30 km	75,3 km/h	216,33 s	17.7 m
Longitudinal	36 km/h	2,52 km	79,3 km/h	247,09 s	36.4 m

Através destes resultados é possível transpor esta informação para gráficos. Desta forma é possível comparar a interferência do vento e do peso no voo da aeronave. A primeira comparação (Figura 7.4) consiste em todos os testes, comparados com vento aplicado e tempos de voo. Para o voo diagonal é possível constatar que o vento não interfere muito nos tempos de voo, no entanto o mesmo não se verifica com os restantes.

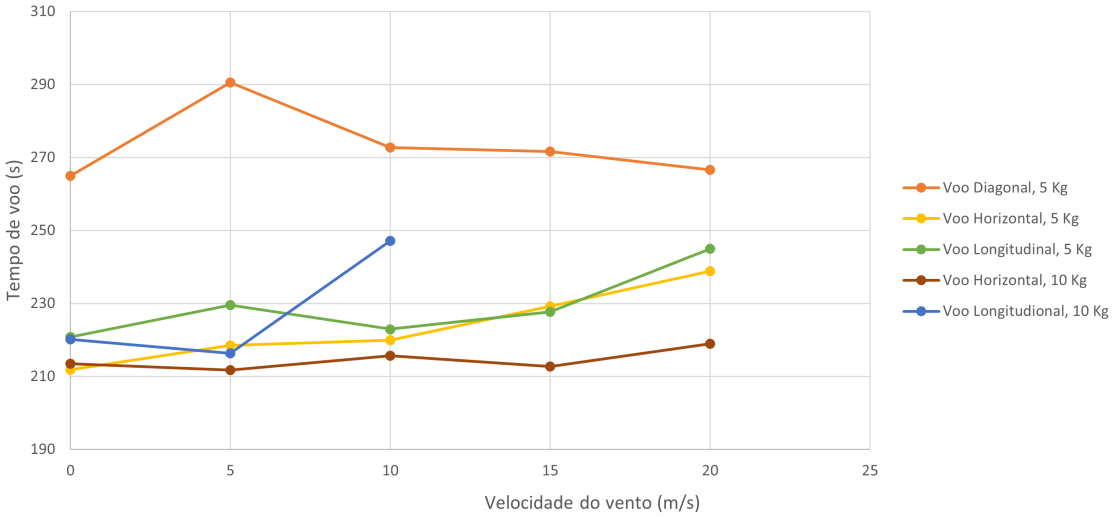


Figura 7.4: Comparação da interferência do vento e peso nos tempos de voo

Nas comparações das Figuras 7.5 e 7.6 são comparados a influência do vento e peso na distância de voo e diferenças máximas de altitude em *fixed wing*. Estes resultados, (Figura 7.5), são importantes para que se avalie o *overshoot* dos controladores de posição em relação a diferentes incidências de vento e peso. Os resultados da Figura 7.6 permitem avaliar a estabilidade do veículo nas condições apresentadas, no entanto também podem influenciar a distância percorrida.

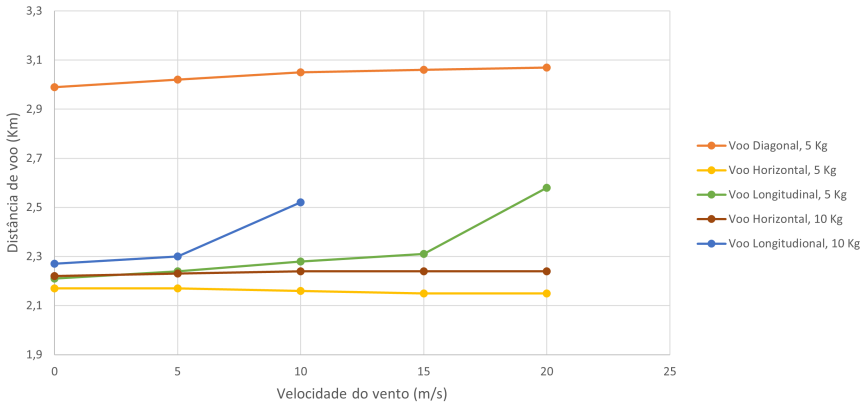


Figura 7.5: Comparação da interferência do vento e peso na distância percorrida de voo

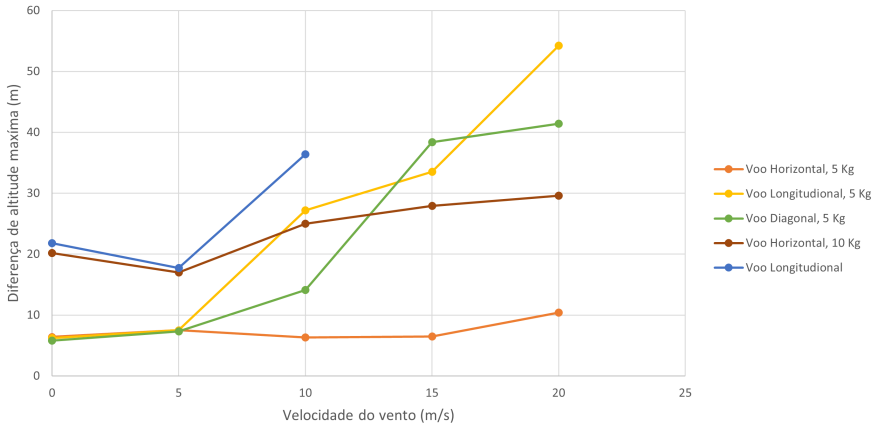


Figura 7.6: Comparação da interferência do vento e peso na diferença de altitude

É possível constatar novamente constatar que a situação menos influenciada pelo vento é o voo em diagonal. No entanto, também é possível concluir que quanto maior o peso da aeronave, menos influência o vento tem sobre esta em voo horizontal. No que toca a voo longitudinal, quanto maior o peso maior a dificuldade de voo, acabando mesmo por comprometer a segurança do voo. Concluindo, em análise a todos os elementos anteriormente apresentados, a situação crítica de voo é a longitudinal, sendo este o primeiro ponto de saturação para um voo estável. Esta é a configuração mais afetada (longitudinal) porque está totalmente relacionada com o ângulo de ataque das asas e a direção de propulsão. Não tendo propulsão suficiente, não consegue contrariar os ventos.

7.2 Comparação das inspeções e manobras

Após o desenvolvimento integral de todos módulos de software na *framework* ROS, a correta implementação de todos os elementos em *Gazebo* e por fim das manobras de inspeção, foram comparadas diferentes situações de voo. Para todas as situações foi utilizado o peso *default* da *frame* de simulação, 5 kg.

Para efeitos de comparação das manobras foi imposto um critério de forma a validar ou rejeitar uma determinada imagem. Como tal, para uma imagem ser aceite, esta deve conter dentro de uma área específica todo o objeto de interesse para a inspeção. Para o caso específico da inspeção geral e *loiter* o critério é aplicado aos isoladores da torre, sendo que pelo menos 7 dos 8 devem estar dentro da área. No caso da inspeção local, apenas o isolador ou o par de isoladores em inspeção deve estar contido na área designada. É importante referir que a câmara está fixa à estrutura não tem qualquer

estabilização. A Figura 7.7 representa uma imagem aceite enquanto que a Figura 7.8 representa uma imagem rejeitada.

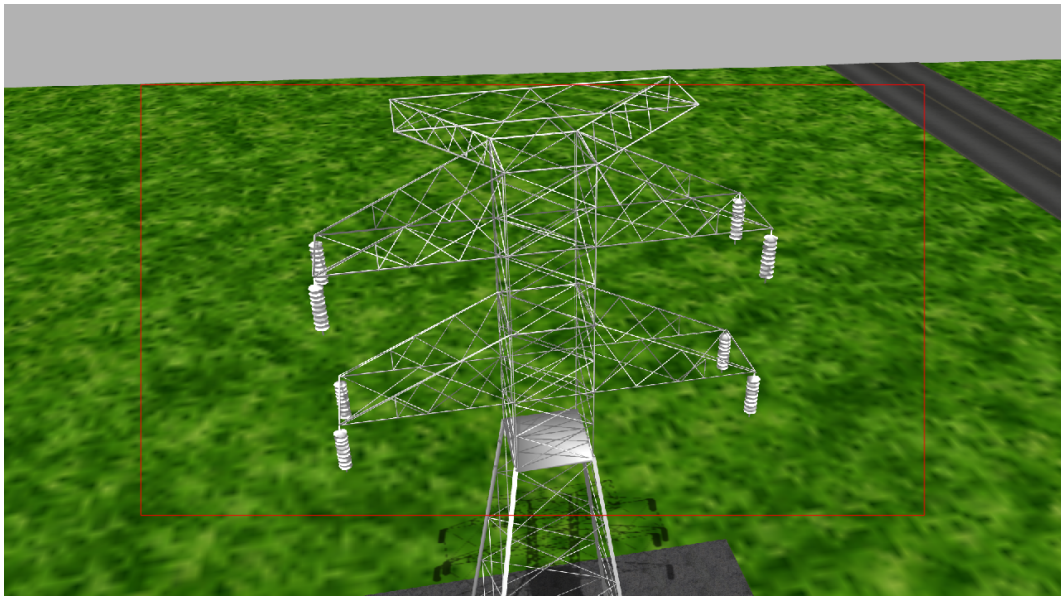


Figura 7.7: Exemplo de uma imagem aceite

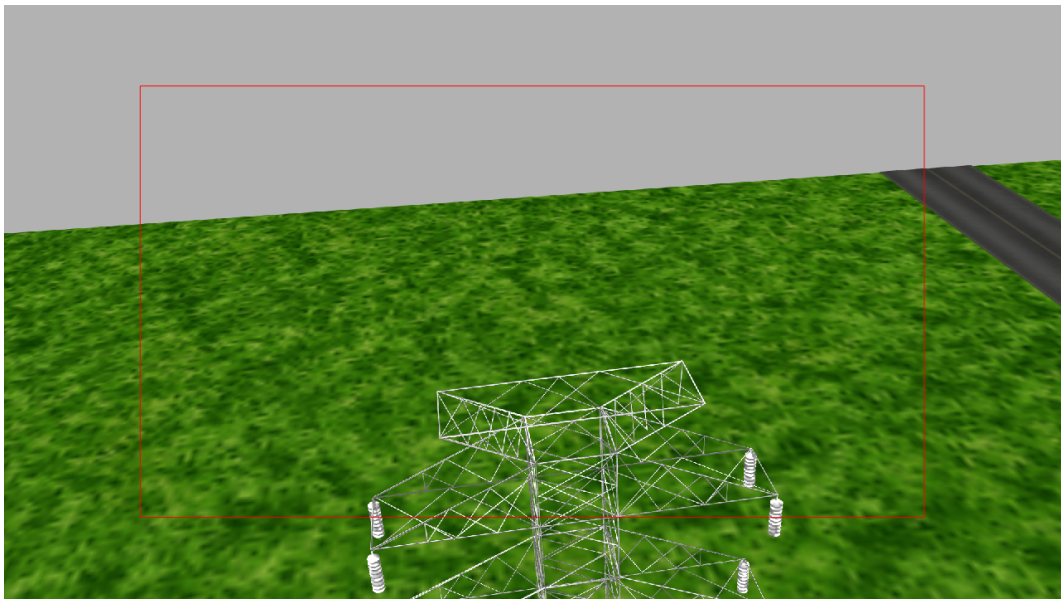


Figura 7.8: Exemplo de uma imagem rejeitada

7.2.1 Manobra de inspeção proposta

O primeiro ponto de comparação foi a manobra proposta. Tal como referido no capítulo 6, foram desenvolvidas duas manobras, uma combinada que agrupa os isoladores em pares na inspeção local. A outra manobra inspeciona os isoladores individualmente.

De modo a obter resultados comparativos foram gravados *bags* com imagens das inspeções, sendo que foram efetuados 8 *bags*. Os *bags* foram feitos com uma combinação de parâmetros alterados, entre eles o recurso às capacidades de VTOL ou apenas voo em modo *multirotor*, a manobra de inspeção e a simulação de vento a 36 km/h ou sem aplicação de vento. A Tabela 7.3 demonstra os resultados para as manobras que recorreram ao uso das capacidades totais de VTOL. O método de aceitação ou rejeição de imagens é o exemplificado nas Figuras 7.7 e 7.8.

Tabela 7.3: Resultado da manobra de inspeção proposta, com recurso a VTOL

Tipo de voo	Nr. de imagens aceites	Nr. total de imagens	Percentagem de sucesso de imagens	Tempo total de inspeção	Tempo total de voo
Insp. combinada sem vento	448	716	62.6 %	149.66 s	334.86 s
Insp. detalhada sem vento	684	925	73.9 %	193.24 s	397.19 s
Insp. combinada com vento	386	713	54,1 %	147.09 s	394.94 s
Insp. detalhada com vento	540	887	60.9 %	186.51 s	387.16 s

Através da análise dos resultados obtidos é possível constatar que a manobra complicada tem uma percentagem de sucesso maior. Isto deve-se ao facto de apenas necessitar de conter um isolador na área de interesse, enquanto que a manobra simples necessita de ter os dois. Apesar deste facto, a diferença percentual média não excede os 8 %.

7.2.1.1 Avaliação do VTOL como sistema de inspeção de ativos elétricos

De modo a comparar o uso de um VTOL com o uso de um simples *multirotor*, tal como referido anteriormente, foram efetuados testes das manobras de inspeção desenvolvidas em *matlab*, recorrendo apenas ao voo em *multirotor*. Os resultados dos testes apenas em modo *multirotor* estão representados na Tabela 7.4, os resultados da utilização do VTOL são os da Tabela 7.3.

Tabela 7.4: Resultado da manobra de inspeção desenvolvida em matlab, apenas em multirotor

Tipo de voo	Nr. de imagens aceites	Nr. total de imagens	Percentagem de sucesso de imagens	Tempo total de inspeção	Tempo total de voo
Insp. combinada sem vento	468	764	61.3 %	158.20 s	434.95 s
Insp. detalhada sem vento	618	945	65.4 %	200.39 s	503.81 s
Insp. combinada com vento	416	768	54.2 %	158.14 s	454.62 s
Insp. detalhada com vento	612	959	63.8 %	201.69 s	493.27 s

Tendo em conta os resultados das tabelas, comparando o uso de VTOL com o *multirotor* apenas, o tempo das manobras de inspeção apenas difere em média 5.9 % enquanto que o tempo total de missão difere em média 19.7 %. A Figura 7.9 ilustra a comparação efetuada.

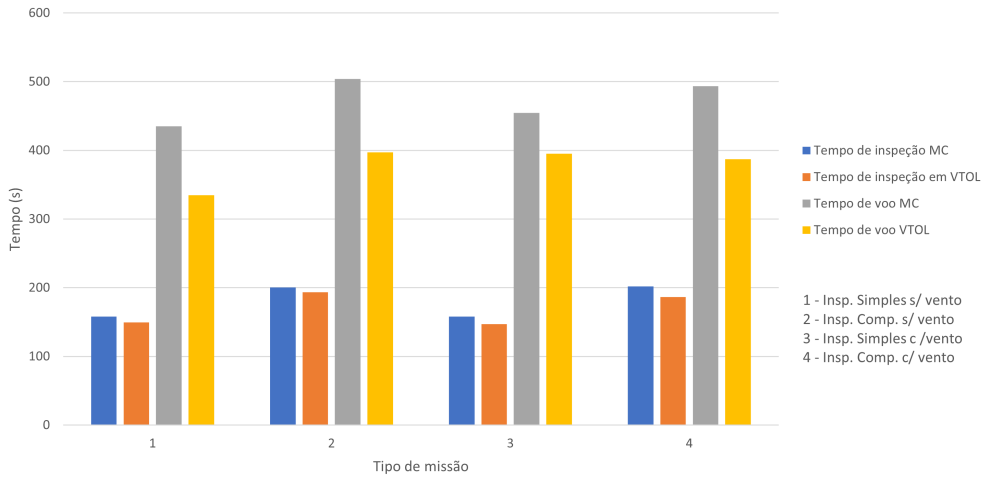


Figura 7.9: Comparação de tempos de voo, modos de voo e missões

Tendo em consideração que os resultados apresentados apenas indicam a inspeção de uma torre, sendo que a diferença nos tempos de inspeção é apenas 5.9 %, é possível afirmar que o uso de VTOL não afeta a inspeção. No entanto, comparando o tempo de voo total, o uso de VTOL é muito benéfico uma vez que para inspeções de maior número de ativos e com maiores distâncias, o tempo de voo total diminuiu cada vez mais.

Considerando a velocidade média de cruzeiro do avião na simulação, 15.4 m/s e a velocidade média de cruzeiro na simulação do *multirotor*, 4 m/s, é possível concluir que entre dois pontos distantes, modo avião atinge esse ponto cerca de 3.75 vezes mais rápido

que o *multirotor*.

7.2.2 Manobra de inspeção em *loiter*

O segundo ponto de comparação foi a manobra de *loiter* em avião. Foram efetuados dois testes, sem vento (condições ideais) e com vento de 36 km/h. A Figura 7.10 representa um exemplo de uma imagem aceite.

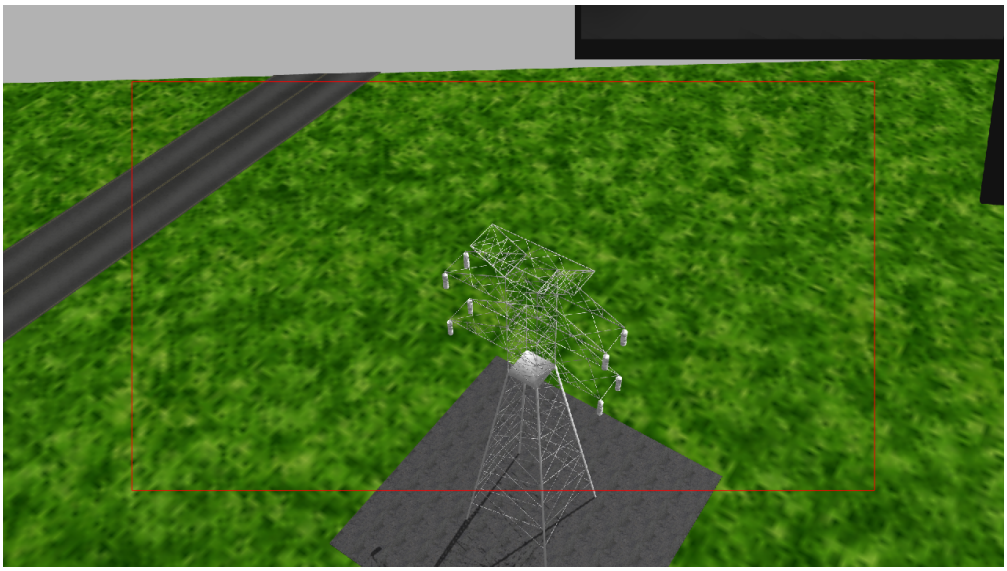


Figura 7.10: Exemplo de uma imagem aceite

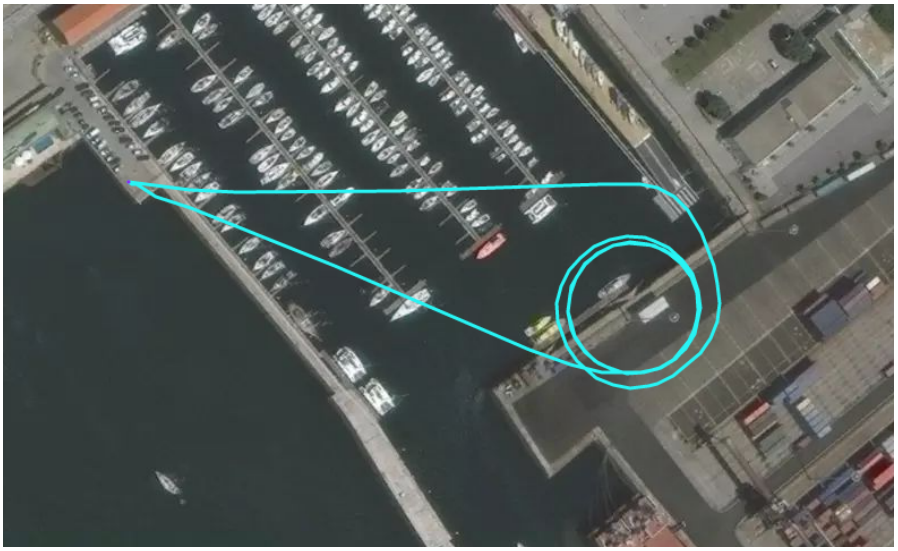
Tal como o exemplo anterior, foram analisados o número de frames em que o ponto de interesse se encontra no centro da imagem, o tempo de inspeção da manobra e o tempo de voo. A Tabela 7.5 demonstra os resultados obtidos.

Tabela 7.5: Resultado da manobra de inspeção em loiter

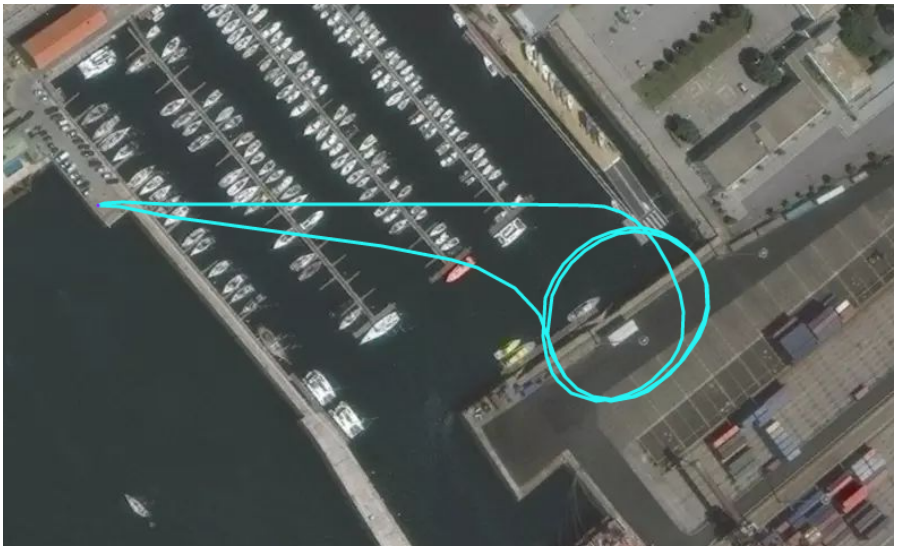
Tipo de voo	Nr. de imagens aceites	Nr. total de imagens	Percentagem de sucesso de imagens	Tempo total de inspeção	Tempo total de voo
Inspeção loiter sem vento	156	171	91.2%	35.78 s	153.83 s
Inspeção loiter com vento	48	170	28.2%	34.98 s	147.38 s

Como é possível constatar, em condições ideais a manobra de *loiter* apresenta excelente resultados, com uma taxa de sucesso de imagens de 91.2 %. No entanto, após a inserção de vento na simulação a taxa baixa para os 28.2 %. Este fenómeno deve-se

ao facto de a aeronave não conseguir manter um círculo estável em torno do objeto de inspeção, tornando-se uma elipse, influenciada pelo vento. De modo a contrariar estes efeitos, a introdução de um *gimbal*, para estabilizar a câmara, é importante, podendo aumentar novamente a percentagem de imagens aceites. A Figura 7.11 demonstra a influência do vento na inspeção.



(a) Inspeção sem vento



(b) Inspeção com vento

Figura 7.11: Trajetória de inspeção em *loiter*

7.2.3 Reconstrução 3D do modelo da torre

Após o levantamento das imagens válidas, procedeu-se à reconstrução tridimensional do objeto inspecionado através das imagens aceites da inspeção. Esta reconstrução foi feita através de um *software* semelhante ao *meshroom*, *Regard3D*. Uma vez que o ambiente é pouco complexo, a deteção de *features* para associação de imagens torna-se complicada.

O uso do *Regard3D* foi motivado pela utilização da informação da posição global de cada imagem para a reconstrução. Esta informação auxilia a associação entre imagens, possibilitando assim a reconstrução do modelo. Esta demonstração foi feita com o intuito de avaliar visualmente as manobras de inspeção projetadas. A Figura 7.12 representa a reconstrução da manobra efetuada em *loiter* e a Figura 7.13 a manobra de inspeção detalhada em *multirotor*.

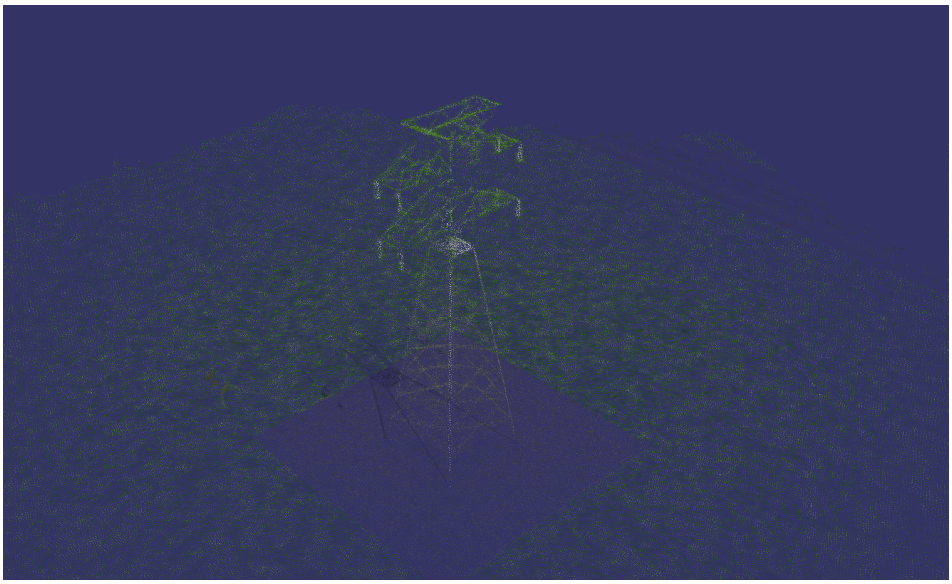


Figura 7.12: Reconstrução 3D a partir das imagens obtidas na manobra de *loiter*

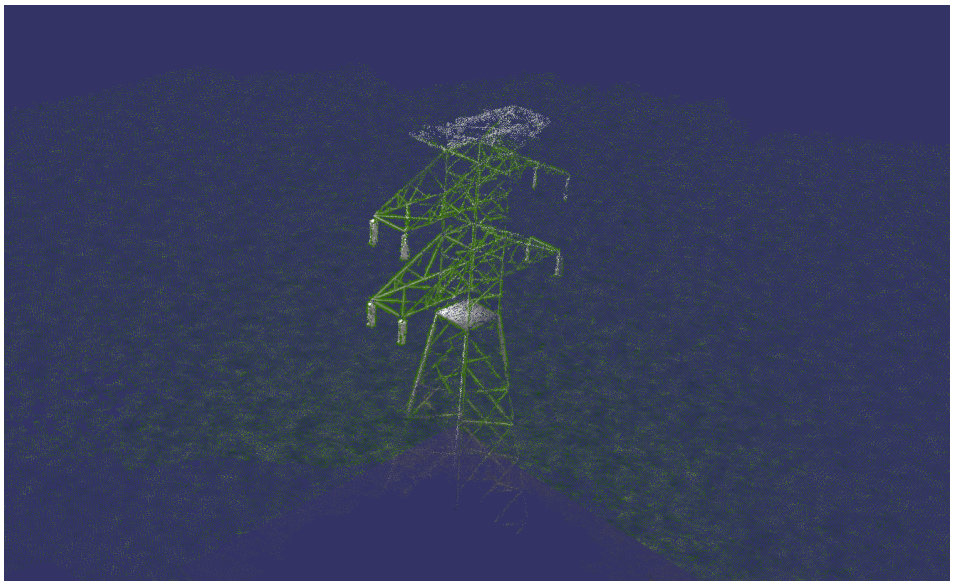


Figura 7.13: Reconstrução 3D a partir das imagens obtidas na manobra de inspeção em *multirotor*

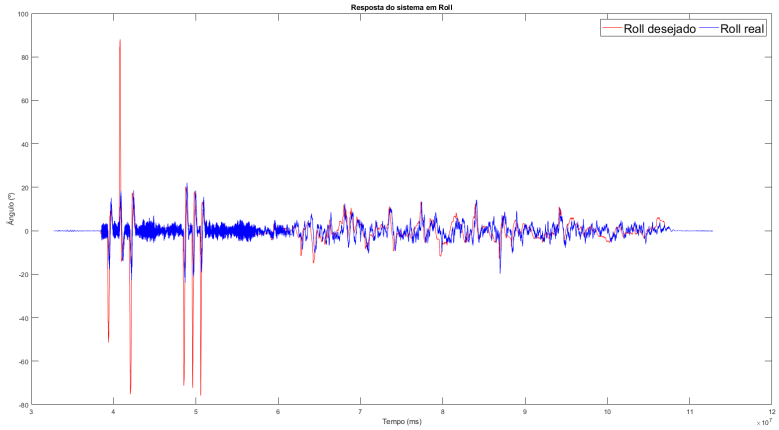
7.3 Testes experimentais efetuados com a plataforma VTOL

Os últimos resultados deste documento consistem na demonstração de um voo da aeronave desenvolvida. Dada a situação da pandemia COVID-19, o acesso ao equipamento físico foi restrito, o que afetou o desenvolvimento e liberdade de realizar mais testes de voo. Porém, ainda foi possível efetuar testes de voo no Instituto Superior de Engenharia do Porto (ISEP), mas apenas em *multirotor*.

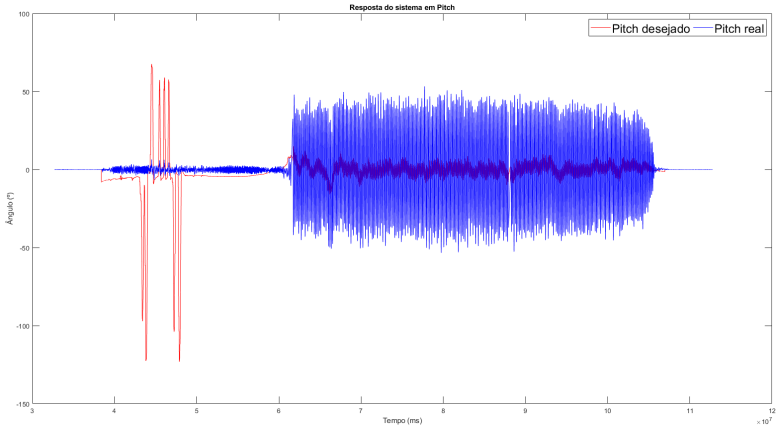
Após a montagem do veículo, e consequente realização dos testes da integração de todos os elementos, foram efetuados alguns voos. Foi constatado que com os valores *default* do *firmware*, a aeronave não voava corretamente, apresentando muita oscilação no eixo de *pitch*. O voo pré calibração pode ser visualizado em [1] e o voo pós calibração pode ser vizualizado em [2].

Como é possível constatar no vídeo, o voo foi curto, devido à instabilidade do VTOL. Após alguma análise, concluiu-se que o controlador PID estava mal calibrado, procedendo-se assim à sua calibração. Na Figura 7.14 é possível constatar o *log* da resposta da aeronave com o PID *default*. Já a Figura 7.15 contém a resposta do controlador PID após a calibração.

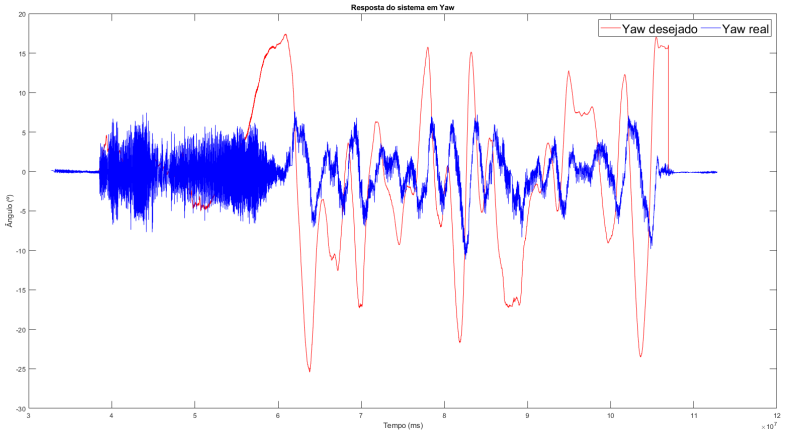
[1] `https://youtu.be/9moH8nvL68s`
[2] `https://youtu.be/px3XkDsV84k`



(a) Resposta em *Roll*

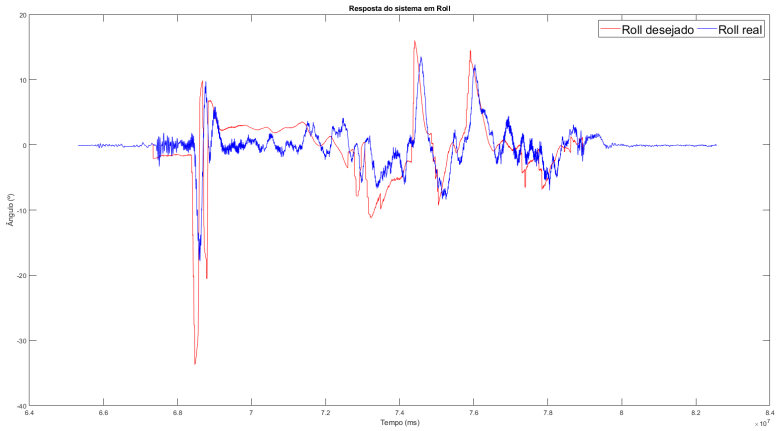


(b) Resposta em *Pitch*

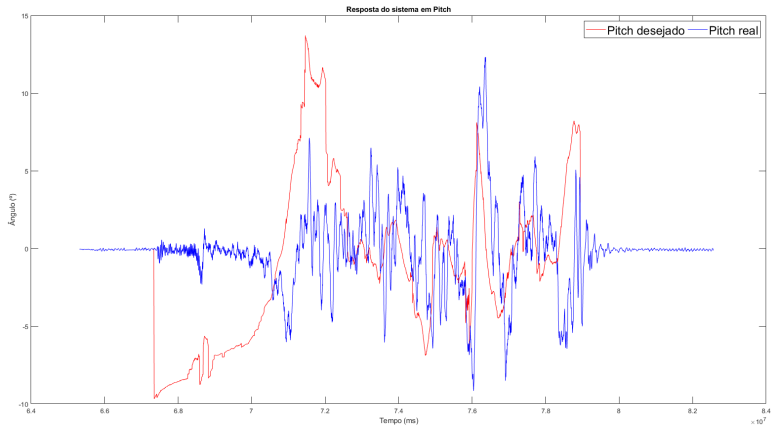


(c) Resposta em *Yaw*

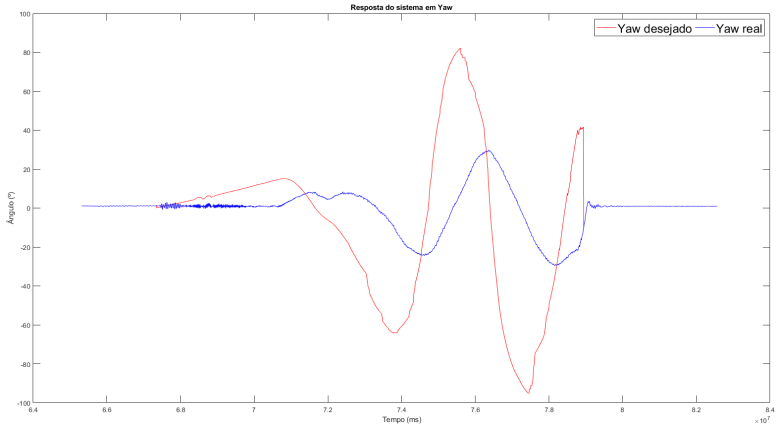
Figura 7.14: Resposta do controlador PID, pré calibração



(a) Resposta em *Roll*



(b) Resposta em *Pitch*



(c) Resposta em *Yaw*

Figura 7.15: Resposta do controlador PID, pós calibração

Esta página foi intencionalmente deixada em branco.

Capítulo 8

Conclusões e trabalho futuro

Com o desenvolvimento deste projeto foi possível constatar que o desenvolvimento de um VTOL é um processo complexo e requer muito tempo despendido. Contudo, considerados os objectivos iniciais do trabalho, o projeto foi considerado como um sucesso, apesar de ainda carecer de algumas melhorias.

Considera-se que o VTOL é uma solução viável para que no futuro sejam efetuadas inspeções a ativos eléctricos em segurança, de forma autónoma, sem recurso a helicópteros. A flexibilidade que o VTOL introduz para diferentes tipos de manobras é um fator muito importante, sendo o fator autonomia de voo o mais importante comercialmente. No entanto, a solução tal como apresentada ainda requer testes reais de inspeção e possivelmente algumas otimizações.

No que toca às manobra de inspeção, caso o objetivo seja apenas verificar a integridade da linha de transmissão, a manobra em *loiter* considera-se a mais indicada, mesmo com a presença de vento. No entanto, caso o objetivo da inspeção seja detalhar o estado dos isoladores (ex. quanto à sua deterioração), considera-se muito relevante as manobras desenvolvidas em *matlab*, tanto a simples como a complicada. A adição de um *gimbal* para fazer estabilização de imagem é essencial e sem dúvida melhoraria a percentagem de imagens aceites. Uma vez que permite alterações a vários parâmetros por parte do utilizador, o algoritmo desenvolvido é bastante versátil para a aplicação a diferentes modelos de apoios.

A montagem da aeronave foi concluída com sucesso. Apesar de não ter sido realizado nenhum voo em *fixed-wing*, todos os componentes foram selecionados com recurso às especificações do fabricante da *frame*, de modo a garantir o seu funcionamento. A Figura 8.1 representa um teste de voo real, feito com a estrutura desenvolvida, em modo de *multirotor*.

Figura 8.1: Solução final desenvolvida, em voo

8.1 Trabalho futuro

Dada a dimensão do projeto e às restrições temporais, alguns aspetos do desenvolvimento do mesmo foram considerados prioritários em relação a outros. Assim, em seguida são apresentados alguns aspetos considerados relevantes para trabalho futuro, caso este projeto se mantenha em desenvolvimento:

- Testes extensivos da estabilidade de voo do VTOL desenvolvido, tanto em *multirotor* como em *fixed wing*;
- Estudo e implementação de um modo de voo híbrido entre *fixed wing* e *multirotor*, de modo a tentar auxiliar as manobras de *fixed wing* com os motores de *multirotor*;
- Validação dos nós de controlo desenvolvidos em ambientes reais;
- Validação em ambiente real das manobras de voo;
- Integração de um *gimbal* para adicionar estabilidade à câmara;
- Integração de novos sensores e métodos de *obstacle avoidance*;
- Desenvolvimento de placas electrónicas específicas para a integração no VTOL;

- Integração do modulo de missão gerada com a informação que iremos receber de percepção a bordo do VTOL (informação proveniente de LiDAR ou de câmaras de espectro visível).

Esta página foi intencionalmente deixada em branco.

Bibliografia

[1] HiBot. "Inspection of high-voltage power lines is costly, difficult, and dangerous. It's the perfect job for a robot". [Online]. Available: https://spectrum.ieee.org/image/1783012, urldate = 2011-02-04, note = [Accessed: 28-May-2020],.

[2] Autonomous Systems Laboratory. "Byrd II UAV".

[3] Nicole Walton. "Wind cuts power to hundreds across the U.P.". [Online]. Available: https://www.wnmufm.org/sites/wnmu/files/styles/x_large/public/201209/tree%20on%20line.jpg, urldate = 2012-09-24, note = [Accessed: 02-June-2020],.

[4] Paul Ridden. "Hydrogen-powered VTOL launches with 15-hour flight time". [Online]. Available: https://newatlas.com/drones/hydrogen-vtol-griflion-h-15-hour-flight-time/, note = [Accessed: 31-October-2020],.

[5] Terra Drone. "Terra Drone launches UAV AI-based solution for power asset inspection developed after inspecting over 90,000 km power lines by BVLOS". [Online]. Available: https://www.terra-drone.net/global/2019/10/03/terra-drone-launches-uav-ai-based-solution-for-power-asset-inspection\-developed-after-inspecting-over-90000-km-power-lines-by-bvlos/, urldate = 2019-10-03, note = [Accessed: 02-June-2020],.

[6] Terra Drone. [Online]. Available: https://www.terra-drone.net/global/solution-electric/, note = [Accessed: 15-June-2020],.

[7] L. F. Luque-Vega, B. Castillo-Toledo, A. Loukianov, and L. E. Gonzalez-Jimenez. Power line inspection via an unmanned aerial system based on the quadrotor helicopter. In *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, pages 393–397, 2014.

[8] [Online]. Available: `https://www.ros.org/`, note = [Accessed: 15-July-2020],.

[9] Harsh Pandya. Mobile manipulator based framework for dataset generation and algorithm testing. 01 2015.

[10] Mavlink DEV Team. [Online]. Available: `https://mavlink.io/en/messages/common.html`, note = [Accessed: 21-October-2020],.

[11] Ashraf Qadir, William Semke, and Jeremiah Neubert. Vision based neuro-fuzzy controller for a two axes gimbal system with small uav. *Journal of Intelligent and Robotic Systems*, 74, 08 2013.

[12] Andrea Civita, Simone Fiori, and Giuseppe Romani. A mobile acquisition system and a method for hips sway fluency assessment. *Information*, 9:321, 12 2018.

[13] G. Gutin D. Karapetyan. Lin–kernighan heuristic adaptations for the generalized traveling salesman problem. *Information*, 208:221–232, 02 2011.

[14] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6423–6430, 2015.

[15] Philipp Stahl, Thomas Seren, Christian Roessler, and Mirko Hornung. Development and performance comparison of optimized electric fixed-wing vtol uav configurations. 11 2018.

[16] PX4 2017 DEV Team. "Deltaquad VTOL". [Online]. Available: `http://dronecode.diyrobocars.com/portfolio/deltaquad-vtol/`, urldate = 2017, note = [Accessed: 23-October-2020],.

[17] UnmannedRC. "Dragon VTOL". [Online]. Available: `https://unmannedrc.com/products/dragon-vtol-plane-uav`, note = [Accessed: 23-October-2020],.

[18] DJI. "DJI E800 - Tuned propulsion system". [Online]. Available: `https://www.dji.com/pt/e800`, note = [Accessed: 23-October-2020].

[19] DJI. "DJI E1200 Pro - Tuned propulsion system". [Online]. Available: `https://www.dji.com/pt/e1200`, note = [Accessed: 23-October-2020],.

[20] T-Motor. "T-Motor AT3520 Long Shaft". [Online]. Available: `https://store-en.tmotor.com/goods.php?id=794`, note = [Accessed: 23-October-2020],.

[21] Hobbyking. "HobbyKing YEP 80A". [Online]. Available: `https://hobbyking.com/pt_pt/yep-80a-2-6s-sbec-brushless-speed-controller.html`, note = [Accessed: 23-October-2020],.

[22] Emax. "EMAX ES3004". [Online]. Available: `https://emaxmodel.com/products/emax-es3004-17g-3-5kg-0-13sec-23t-metal-gear-analog-servo-for-rc-airplane-es3104-upgra` note = [Accessed: 23-October-2020],.

[23] Hobbyking. "Turnigy Graphene Professional 12000mAh". [Online]. Available: `https://hobbyking.com/pt_pt/turnigy-graphene-professional-12000mah-6s-15c-lipo-pack-w-xt90.html?fbclid=IwAR3HIDoek0mHXuTUxFkMdLWwi3yZbnsU_USnWgG2Kys7DlOWVnNG4fRuZ9E`, note = [Accessed: 23-October-2020],.

[24] Hobbyking. "Plane landing gear". [Online]. Available: `https://hobbyking.com/pt_pt/alloy-oleo-strut-set-with-wheels-and-rubber-tires-104mm-length-4mm-mounting-pin.html`, note = [Accessed: 23-October-2020],.

[25] J. Roberts, D. Frousheger, B. Williams, D. Campbell, and R. Walker. How the outback challenge was won: The motivation for the uav challenge outback rescue, the competition mission, and a summary of the six events. *IEEE Robotics Automation Magazine*, 23(4):54–62, 2016.

[26] Wingcopter, Vodafone Ireland. "World's first diabetes drone completes maiden flight". [Online]. Available: `https://www.vodafone.com/perspectives/blog/world-first-diabetes-drone-completes-maiden-flight`, urldate = 2020-05-29, note = [Accessed: 29-May-2020],.

[27] A Pagnano, M Höpf, and R Teti. A Roadmap for Automated Power Line Inspection. Maintenance and Repair. *Procedia CIRP*, 12:234–239, 2013.

[28] Leena Matikainen, Matti Lehtomäki, Eero Ahokas, Juha Hyyppä, Mika Karjalainen, Anttoni Jaakkola, Antero Kukko, and Tero Heinonen. Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119:10–31, 2016.

[29] Jornal de Notícias, Ana Gaspar. "Fogo de Pedrógão começou com contacto entre vegetação e linha da EDP". [Online]. Available: `https://www.jn.pt/nacional/fogo-de-pedrogao-comecou-com-falha-eletrica-diz-novo-relatorio-8848733.html`, urldate = 2017-10-16, note = [Accessed: 02-June-2020],.

[30] Diário de Notícias, Paula Sofia Luz. "Pedrógão. Dois anos depois, a região é um barril de pólvora". [Online]. Available: https://www.dn.pt/edicao-do-dia/15-jun-2019/pedrogao-grande-dois-anos-depois-a-regiao-e-um-barril-de-polvora-11011420.html, urldate = 2019-06-15, note = [Accessed: 02-June-2020],.

[31] Autonomous Systems Laboratory. "STORK UAV".

[32] Autonomous Systems Laboratory. "FALCOS". [Online]. Available: https://archive.osvaldosousa.com/lsa/falcos/index.html, urldate = 2020-05-28, note = [Accessed: 28-May-2020],.

[33] F. Azevedo, A. Dias, J. Almeida, A. Oliveira, A. Ferreira, T. Santos, A. Martins, and E. Silva. Real- time lidar-based power lines detection for unmanned aerial vehicles. In *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 1–8, 2019.

[34] T. Santos, M. Moreira, J. Almeida, A. Dias, A. Martins, J. Dinis, J. Formiga, and E. Silva. Plined: Vision-based power lines detection for unmanned aerial vehicles. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 253–259, 2017.

[35] ArduPilot Dev Team. "ArduPilot". [Online]. Available: https://ardupilot.org/ardupilot/index.html, note = [Accessed: 03-June-2020],.

[36] EDP Labelec. "A era do drone". [Online]. Available: https://www.edp.com/pt-pt/a-era-do-drone, note = [Accessed: 01-November-2020],.

[37] "Panther tilt-rotor UAV". [Online]. Available: http://www.deagel.com/library/Panther-tilt-rotor-UAV_m02010102600002.aspx, urldate = 2017-04-08, note = [Accessed: 15-June-2020],.

[38] Drone know-how. "Tailsitters vs. quadplanes – why a VTOL tailsitter is the best surveying drone for your mapping missions". [Online]. Available: https://wingtra.com/tailsitters-vs-quadplanes-why-a-vtol-tailsitter-is-the-best-surveying-drone-for-
urldate = 2018-11-29, note = [Accessed: 15-June-2020],.

[39] Steven Flynn. "Ireland's First BVLOS Delivery of Insulin". [Online]. Available: https://skytango.com/

irelands-first-bvlos-insulin-delivery-for-diabetes-via-drone-to-inis-mor/, urldate = 2019-09-16, note = [Accessed: 09-June-2020],.

[40] Wingcopter. [Online]. Available: https://wingcopter.com/, note = [Accessed: 09-June-2020],.

[41] QGroundControl dev team. "Intuitive and Powerful Ground Control Station for the MAVLink protocol.". [Online]. Available: http://qgroundcontrol.com/, note = [Accessed: 09-June-2020],.

[42] S. Omari, P. Gohl, M. Burri, M. Achtelik, and R. Siegwart. Visual industrial inspection using aerial robots. In *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*, pages 1–5, 2014.

[43] S. Hrabar, T. Merz, and D. Frousheger. Development of an autonomous helicopter for aerial powerline inspections. In *2010 1st International Conference on Applied Robotics for the Power Industry*, pages 1–6, 2010.

[44] S. Montambault, J. Beaudry, K. Toussaint, and N. Pouliot. On the application of vtol uavs to the inspection of power utility assets. In *2010 1st International Conference on Applied Robotics for the Power Industry*, pages 1–7, 2010.

[45] PX4 Dev Team. "PX4". [Online]. Available: https://px4.io/software/software-overview/, note = [Accessed: 19-October-2020],.

[46] X. Luo, X. Li, Q. Yang, F. Wu, D. Zhang, W. Yan, and Z. Xi. Optimal path planning for uav based inspection system of large-scale photovoltaic farm. In *2017 Chinese Automation Congress (CAC)*, pages 4495–4500, 2017.

[47] J. Zhang, Z. Guo, and L. Wu. Research on control scheme of vertical take-off and landing fixed-wing uav. In *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 200–204, 2017.

[48] H. Gu, X. Lyu, Z. Li, S. Shen, and F. Zhang. Development and experimental verification of a hybrid vertical take-off and landing (vtol) unmanned aerial vehicle(uav). In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 160–169, 2017.

[49] J. k. Gunarathna and R. Munasinghe. Development of a quad-rotor fixed-wing hybrid unmanned aerial vehicle. In *2018 Moratuwa Engineering Research Conference (MERCon)*, pages 72–77, 2018.

[50] W Curran, T Thornton, B Arvey, and W D Smart. Evaluating impact in the ROS ecosystem. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6213–6219, 2015.

[51] Morgan Quigley, Ken Conley, Brian P Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[52] Satya Mallick. "Rotation Matrix To Euler Angles". [Online]. Available: `https://www.learnopencv.com/rotation-matrix-to-euler-angles/`, urldate = 2016-06-04, note = [Accessed: 16-July-2020],.

[53] Gregory G. Slabaugh. Computing Euler angles from a rotation matrix.

[54] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[55] Y. Wang and Y. Huang. Mobile robot path planning algorithm based on rapidly-exploring random tree. In *2019 IEEE International Conferences on Ubiquitous Computing Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pages 555–560, 2019.

[56] Q. Sun, M. Li, T. Wang, and C. Zhao. Uav path planning based on improved rapidly-exploring random tree. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 6420–6424, 2018.

[57] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. Technical report, Department of Computer Science, Roskilde University, Building 42.1, Universitetsvej 1, Box 260, DK-4000 Roskilde, Denmark, 1999.

[58] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

[59] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes. Uniform coverage structural inspection path–planning for micro aerial vehicles. In *2015 IEEE International Symposium on Intelligent Control (ISIC)*, pages 59–64, 2015.

[60] DJI. "DJI - Who we are". [Online]. Available: `https://www.dji.com/pt/company?site=brandsite&from=footer`, note = [Accessed: 23-October-2020],.

[61] Ardupilot DEV Team. "Plane Commands in Guided Mode". [Online]. Available: `https://ardupilot.org/dev/docs/plane-commands-in-guided-mode.html`, note = [Accessed: 23-October-2020],.