UNIVERSIDADE FEDERAL DO PARANÁ

EDUARDO VUDALA SENOSKI

THERMAL EXPLOITATION BY UAVS, ON A SIMULATED ENVIRONMENT

CURITIBA PR

2024

EDUARDO VUDALA SENOSKI

THERMAL EXPLOITATION BY UAVS, ON A SIMULATED ENVIRONMENT

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Todt.

CURITIBA PR

2024

Ficha catalográfica

Substituir o arquivo `0-iniciais/catalografica.pdf` pela ficha catalográfica fornecida pela Biblioteca da UFPR (PDF em formato A4).

Instruções para obter a ficha catalográfica e fazer o depósito legal da tese/dissertação (contribuição de André Hochuli, abril 2019. Links atualizados Wellton Costa, Nov 2022):

1. Estas instruções se aplicam a dissertações de mestrado e teses de doutorado. Trabalhos de conclusão de curso de graduação e textos de qualificação não precisam segui-las.
2. Verificar se está usando a versão mais recente do modelo do PPGInf e atualizar, se for necessário (`https://gitlab.c3sl.ufpr.br/maziero/tese`).
3. conferir o *checklist* de formato do Sistema de Bibliotecas da UFPR, em `https://bibliotecas.ufpr.br/servicos/normalizacao/`
4. Enviar e-mail para "`referencia.bct@ufpr.br`" com o arquivo PDF da dissertação/tese, solicitando a respectiva ficha catalográfica.
5. Ao receber a ficha, inseri-la em seu documento (substituir o arquivo `0-iniciais/catalografica.pdf` do diretório do modelo).
6. Emitir a Certidão Negativa (CND) de débito junto a biblioteca, em `https://bibliotecas.ufpr.br/servicos/certidao-negativa/`
7. Avisar a secretaria do PPGInf que você está pronto para o depósito. Eles irão mudar sua titulação no SIGA, o que irá liberar uma opção no SIGA pra você fazer o depósito legal.
8. Acesse o SIGA (`http://www.prppg.ufpr.br/siga`) e preencha com cuidado os dados solicitados para o depósito da tese.
9. Aguarde a confirmação da Biblioteca.
10. Após a aprovação do pedido, informe a secretaria do PPGInf que a dissertação/tese foi depositada pela biblioteca. Será então liberado no SIGA um link para a confirmação dos dados para a emissão do diploma.

Ficha de aprovação

Substituir o arquivo 0-iniciais/aprovacao.pdf pela ficha de aprovação fornecida pela secretaria do programa, em formato PDF A4.

Nothing is perfect, the world's not perfect, but it's there for us, trying the best it can. That's what makes it so damn beautiful.

ACKNOWLEDGEMENTS

Fico feliz em poder dizer com convicção que tive as melhores pessoas ao meu lado ao longo de toda a minha jornada dentro da universidade, e algumas durante minha vida toda. Me alegra saber que esses anos vão deixar saudades em mim, que vou lembrá-los pelas coisas boas, que mesmo os erros e tempestades me realizam como o ser humano que sou e que serei.

Minha gratidão e amor maior são aos meus pais, Sirlene e Enéas, que me deram a vida. Por confiarem em mim quando necessitei de independência, por me educarem, por me apoiarem incondicionalmente e me deixarem seguir meu próprio caminho. Por me mostrarem que a vida não é uma corrida, por me fazerem uma pessoa mais empática. Vocês são meu porto seguro. Eu amo vocês.

Aos meus amigos de infância, da universidade, da vida, obrigado por me ouvirem, obrigado pela companhia, obrigado por estarem ao meu lado. Sou grato por todas as madrugadas que passamos ébrios.

Sou grato aos professores e mestres que tive durante esse trajeto, lembrarei para sempre do valor da educação e como ela muda a percepção do mundo. Também ao meu orientador neste trabalho, que confiou em mim, e me deixou agir como um pesquisador responsável.

Aos pesquisadores da ÉTS que me guiaram durante a execução deste projeto, obrigado pela paciência e disponibilidade.

RESUMO

De acordo com a DRONEII, o mercado global de drones deve crescer de 26,3 bilhões de dólares americanos em 2021 para 54,6 bilhões de dólares americanos até 2030. O mercado de drones é composto por diversos segmentos, incluindo defesa, empresas, consumidores, segurança pública, logística e transporte de passageiros.

À medida que as aplicações dos VANTs para problemas do dia a dia ampliam seus horizontes, também aumentam os desafios que precisam ser superados. Um fator crítico que afeta essas aeronaves é o consumo de energia, que limita o tempo de voo e reduz as possibilidades desses robôs, tanto em voos autônomos quanto manuais.

Para mitigar esse problema, este trabalho desenvolve uma plataforma de simulação capaz de recriar voos de VANT de alta fidelidade e em tempo real, com simulação gráfica incluída. Esses voos estão sujeitos à presença de térmicas simuladas, que podem influenciar o comportamento da aeronave. Utilizando a sustentação proporcionada por esses fenômenos atmosféricos, são desenvolvidos algoritmos que consideram as térmicas como pontos de interesse durante o voo autônomo, com o objetivo de otimizar o consumo de energia da aeronave, implementando técnicas de voo planado e, consequentemente, economizando energia. Testes utilizando estes algoritmos foram feitos para uma pequena aeronave asa-fixa, resultando em uma redução da aceleração necessária para orbitar uma região, de 50% para 31% explorando algumas térmicas.

Palavras-chave: VANT. Drone. Simulação. Térmicas. Planejamento de missão.

ABSTRACT

According to DRONEII, the global drone market is expected to grow from 26.3 billion US dollars in 2021 to 54.6 billion U.S. dollars by 2030. The drone market is made up of various segments including defense, enterprise, consumer, public safety, logistics, and passenger.

As the applications of UAVs to day-to-day problems expand their horizons, so do the challenges that need to be overcome. One critical factor that affects aircraft is the energy consumption, which limits their flight time, narrowing the possibilities of such robots both on autonomous and manual flights.

To mitigate that issue, this work develops a simulation platform that is able to recreate high-fidelity and real-time UAV flights, with graphical simulation as well. These flights are subject to the presence of simulated thermals that can affect the flight of the aircraft. Using the lift of these atmospheric phenomena, algorithms that consider thermals as a point of interest during autonomous flight are developed, in order to optimize the energy consumption of the aircraft, implementing soaring, and consequently saving energy. Tests using these algorithms were performed for a small fixed-wing aircraft, resulting in a reduction of the throttle needed to orbit a region, from 50% to 31% while exploiting some thermals.

Keywords: UAV. Drone. Simulation. Thermals. Mission planning.

LIST OF FIGURES

LIST OF ACRONYMS

ADSB	Automatic Dependent Surveillance–Broadcast
ESC	Electronic Speed Control
DINF	Departamento de Informática
GCS	Ground Control Station
GPS	Global Positioning System
HITL	Hardware-in-the-loop
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
OS	Operational System
PPGINF	Programa de Pós-Graduação em Informática
PWM	Pulse Width Modulation
RC	Remote Controller
SITL	Software-in-the-loop
UAV	Unmaned Aerial Vehicle
UFPR	Universidade Federal do Paraná
uORB	micro Object Request Broker

CONTENTS

1 INTRODUCTION

An unmanned aerial vehicle (UAV), also known as a drone, is an aircraft that operates without a pilot, crew, or passanger on board. They were originally developed for military purposes and have become essential assets for most military forces around the world. As aviation technologies evolved, their costs fell and became more accessible, expanding their usage to non-military applications.

Due to the rising demand for automated tasks involving drones - such as vigilance, delivery, agriculture, and many others - mission planning for UAVs has become a critical field of research.

Among various aspects of mission planning, one that limits the possibilities of it is the maximum flight time of the aircraft. This factor is capped mainly by the battery capacity and energy consumption of the UAV.

Thermals are columns of rising air that are formed on the ground through the warming of the surface by sunlight. Using soaring on thermal lifts that could be present in a given environment, it is possible to reduce the energy consumption of a drone (Akos et al., 2010).

Many works that tange the soaring of thermals. Most of them discuss flight techniques and thermodynamics characteristics between the aircraft and the air. And to validate the presented conjectures, the tests are performed using real aircrafts on physical environments. Given the costs of equipment, maintenance, and the labor of engineers and pilots, the financial and human capital to perform these activities is high.

Therefore, building a simulation environment is valuable, for it would reduce the costs of tests, reduce the demanded time to perform them, it would allow their parallelization and would also allow one to replicate missions on scenarios that are not at all time available, like cases where thermals are not present in the environment.

1.1 OBJECTIVES

The main objective of this project is to create a high fidelity simulated environment that can replicate in real time the effects of thermals over autonomous UAVs that are executing a mission. Moreover, to use the simulation to test algorithms that consider the presence of thermals during flight planning.

1.1.1 Specific goals

As a set of goals to achieve our objective, the developed environment should:

- Develop an architecture that highlights the interaction between the simulated objects and simulators
- Replicate high fidelity flight scenarios in real time
- Plan and execute missions on environments with thermals present
- Plan and execute missions with thermals as point of interest

1.2 STUDY OUTLINE

Chapter 2 - Background provides a brief overview of the definition of UAVs and thermals. It explores the architecture of a standard UAV, its components, and the role each one plays. This chapter also presents the definition of thermals and how they naturally occur in the environment.

Chapter 3 - Related works explores existing research that has one or more aspects to do with the work presented in this research.

Chapter 4 - Simulation aims to simulate the defined objects and environment of Chapter 2. First, the autopilot that will be executed by the UAV is defined, then the simulators used and how they will interact with each other if in favor of performing the proposed work. Then, a way to simulate a thermal in this setup is also presented.

Chapter 5 - Flight assembles the ideas presented in previous chapters and uses them as a platform to develop algorithms that aim to use thermals during flight, to save energy. Tests are performed and results are shown to determine whether thermals can actually be used in advantage of UAV missions.

Chapter 6 - Conclusion summarizes the study's proposal and achievements and offers an analysis of what was created within the defined objectives. In addition, it suggests opportunities for future research and possible improvements to expand the simulator and algorithms' capabilities.

2 BACKGROUND

In this chapter, the objects involved in the pretend work and it's environment are defined.

2.1 THE UAV

On this scenario, the object that will be soaring around a thermal will be an UAV classified as a fixed wing. It is very similar to a single piston plane in terms of design and maneuvers, but it works on a very different scope.

Figure 2.1 displays the moves that these types of aircraft are capable of performing. All of these moves are controlled by the ailerons and propellers of the aircraft.

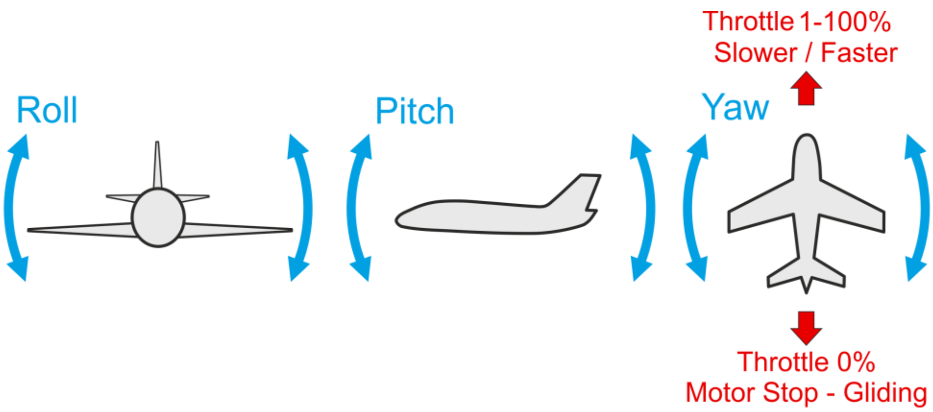

Figure 2.1: Movements that can be performed by a fixed-wing UAV (PX4, 2024a)

The UAV model chosen for this work is called RC Cessna. It weighs approximately 1.64 kilograms and has a wingspan of 1120 milimeters.

Figure 2.2 displays a screenshot of a simulator called Gazebo simulating the described aircraft.

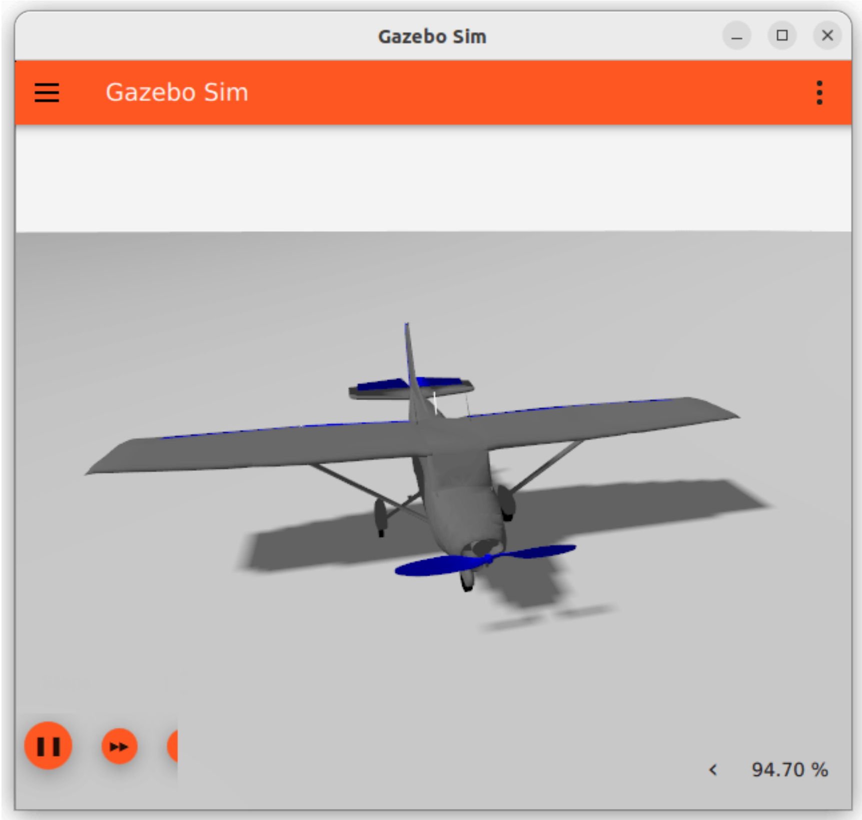

Figure 2.2: Image of a simulated RC Cessna on Gazebo (PX4, 2024b)

The UAV is small compared to a common aircraft, for it does not need a human pilot. Instead, it uses an autopilot to monitor the flight and perform the actions necessary to finish it's tasks.

2.1.1 Autopilot

An autopilot or flight controller is a hardware system that interacts with sensors and payloads of the aircraft, its propellers and ailerons. This interaction is managed by a flight controller software, with the intent of performing a task. Be it to execute the moves issued by a remote controller, to follow a mission that has been flashed into the hardware.

The hardware includes a flight controller, which houses a microprocessor or microcontroller for executing real-time control algorithms, and sensors like accelerometers, gyroscopes, magnetometers, GPS modules, barometers, and cameras for situational awareness. These components are complemented by actuators such as motors and servos, a communication module for remote commands and telemetry, and an optional payload interface for mission-specific equipment like cameras or delivery mechanisms. The flight controller processes sensor data to maintain stability and execute commands, while redundant systems ensure reliability in critical operations.

The software layer of the autopilot system is built on a modular framework that handles tasks like flight stabilization, path planning, obstacle avoidance, and mission execution. Low-level control loops manage real-time stabilization, while higher-level modules use algorithms for waypoint navigation, adaptive control, and object recognition. These systems communicate via middleware, enabling scalability and integration with external ground control stations or onboard AI systems. Safety protocols, including fail-safes and pre-programmed return-to-home functions, are embedded to handle anomalies like signal loss or low battery. Together, these components enable UAVs to operate autonomously in dynamic environments, balancing performance, safety, and versatility for applications ranging from aerial photography to search-and-rescue missions.

2.1.2 Architecture

Figure 2.3 represents a diagram that displays the architecture of a drone, specifically one that runs on the Pixhawk platform. On the rightmost part of the figure, the hardware components of a drone is illustrated. And on the leftmost part of the figure, the software stack corresponding to each component is described.

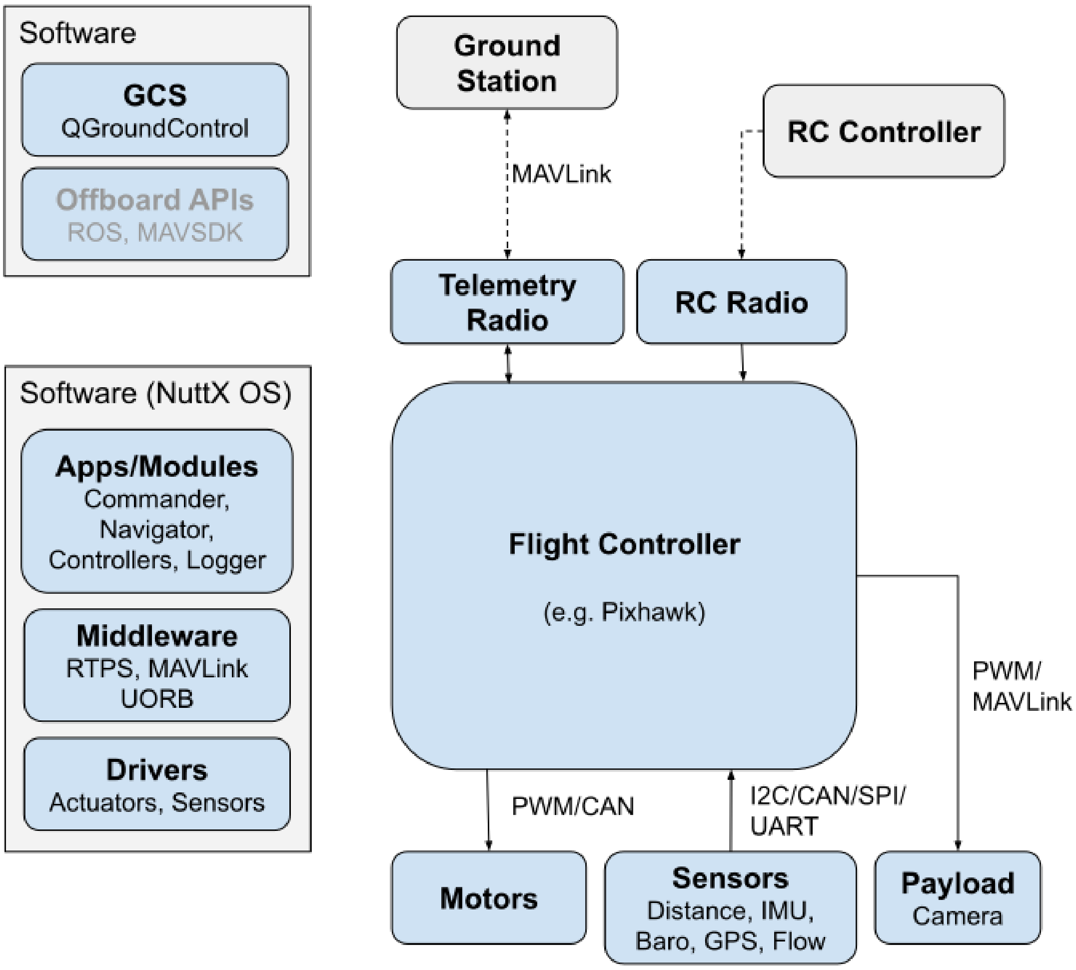

Figure 2.3: Architecture of PX4 Autopilot (PX4, 2024c)

2.1.3 Hardware

- Flight controller, includes internal IMUs, compass and barometer.
- Motor ESCs connected to PWM outputs, DroneCAN (DroneCAN allows two-way communication, not single direction, as shown) or some other bus.
- Sensors (GPS, compass, distance sensors, barometers, optical flow, barometers, ADSB transponders, etc.).
- Camera or other payload. Cameras can be connected to PWM outputs or via MAVLink.

- Telemetry radios for connecting to a ground station computer/software.
- RC Control System for manual control.

2.1.4 Software

The left-hand side of the diagram shows the software stack, which is horizontally aligned (approximately) with the hardware parts of the diagram.

- The ground station computer typically runs QGroundControl (or some other ground station software). It may also run robotics software like MAVSDK or ROS.
- The PX4 flight stack running on the flight controller includes drivers, comms modules, controllers, estimators, and other middleware and system modules.

2.2 THERMALS

Since the goal of this work is to simulate the flight with thermals as objects of interest, a clear landscape space will suffice. Adding obstacles would add an unnecessary layer of complexity for the purpose of this work. A world with an unobstructed landing and taking off path will be considered, as well as the resistance of the air present in the world. Scattered across this landscape, thermals will be present.

(Sparrow et al., 1970) defines thermals as masses of relatively hot fluid which ascend through the environment above a heated horizontal surface. A thermal has a mushroom-like appearance with a blunted nearly hemispherical cap. At a given heating rate, thermals are generated at fixed sites that are spaced more or less regularly along the length of the heated surface.

Figure 2.4 illustrates a thermal. The cloud (A) is above the ground. The sun increases the temperature of the ground which will then warm the air above it (1). The bubble of hot air starts to rise (2) until a certain point . Due to its lower temperature, the mass condenses and moves downward (3).

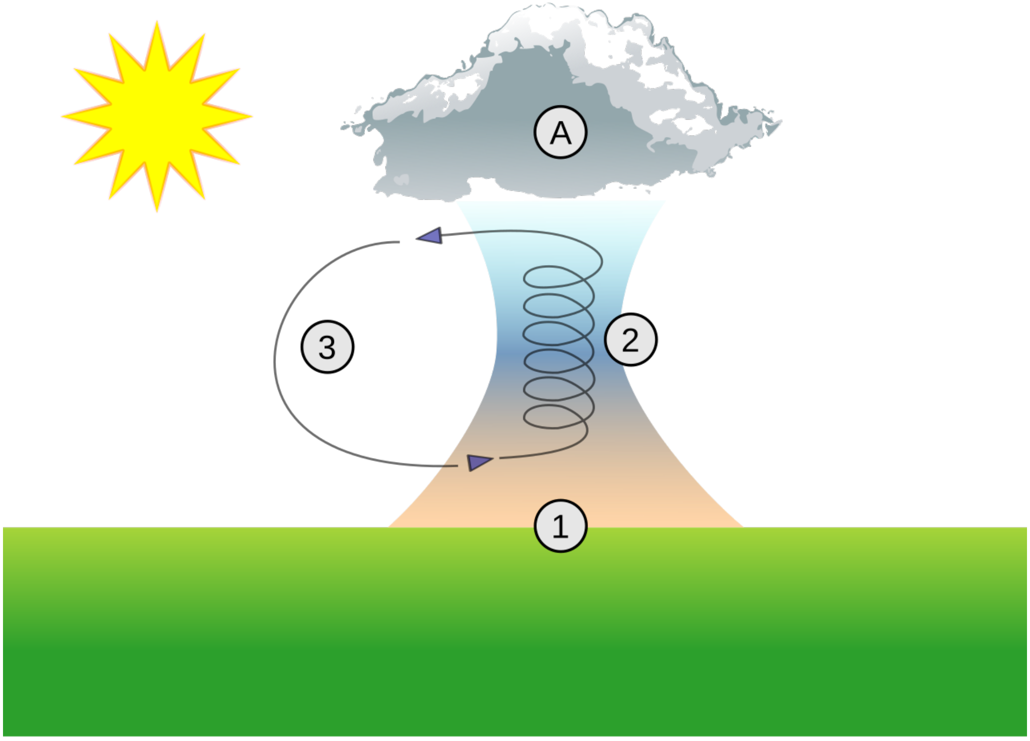

Figure 2.4: Illustration of a thermal (Dake, 2006)

As they are irregular masses of air, its lift would also vary depending on the UAV position relative to it, so the established force that a thermal will apply must vary accordingly.

Thermals can exert forces that depend on a wide variety of factors of the environment, mainly related to the heat difference of its actors (ground surface, surrounding air, etc.). In order to simplify the definition, all thermals present in the environment will exert upward forces only, which range from 0 meters per second up to 12 meters per second. The lower limit is established so the effect can have a sufficient significance on the flight, and the upper limit is defined so the thermal's effect does not disturb the flight up to point that the flight controller isn't able to steer the aircraft properly.

2.3 DETERMINING THE POSITION OF A THERMAL

If a UAV is to exploit a thermal's lift, it needs to determine the position and area of the phenomenon. Some existing works present techniques on how this could be done. For example, (Cobano et al., 2013) presents an algorithm that considers the change of altitude gain to determine the possible positions of these thermals.

In this work's scenario, the thermals must have their presence recognized before they can be considered as a point of interest; therefore, the present thermals positions and areas will be known *a priori*.

3 RELATED WORKS

As the usage of UAVs in industrial and commercial areas grows, so does the interest in them in research. In recent years, there has been significant growth in research using simulation as a tool to leverage their work. These works touch on the most various aspects of drone usage and design. Here are some of them that are related to the objective of this project.

(Akos et al., 2010) presents an overview of the soaring flight and strategy of birds, also an historical review, followed by a discussion of control strategies that have been developed for soaring UAVs both in simulations and applications on real platforms. They state how birds and aircraft can save energy through this exploitation. To improve the accuracy of simulation of thermal exploitation strategies they propose a method to take into account the effect of turbulence.

(Lu and Geng, 2011) develops the control law of an autopilot, designs a modular simulation using Matlab/Simulink, then introduces a HITL (Hardware in the Loop) to make the simulation more accurate. Reducing the costs of validating and modeling algorithms for autopilots.

(Wen et al., 2019) designs a UAV fault injection simulation system, in order to train the ability of unmanned aerial vehicle operators to deal with the fault and verify the effectiveness of the fault detection and processing scheme. Based on the distributed architecture, the system performs real-time simulations and has good scalability.

(Zhang et al., 2012) aims to find an effective way to design and test the control law for the UAV, using simulation as a tool to validate the model developed.

The HPO-RRT* algorithm (Guo et al., 2023) focuses on real-time path planning for UAVs, ensuring path homotopy optimality and high planning efficiency. It features a hierarchical architecture that includes a UAV perception system, path planner, and path optimizer, which work together to predict threats, update world information, and derive heuristic paths.

An improved RRT*-based real-time path planning algorithm suitable for UAVs is discussed in (Xu et al., 2021). This version of the algorithm incorporates a multi-optimized RRT* that redefines the cost function by considering both Euclidean distance and the turning angle, and modifies the sampling method to reduce the sampling space and hasten convergence.

The authors in (Fan et al., 2023) conducted a study on UAV trajectory planning using a bidirectional APF-RRT* algorithm. This algorithm aims to reduce convergence time and implements improved strategies to decrease redundancy points, combined with a goal-biased strategy to achieve higher quality sampling and smoother path generation optimized by cubic spline.

Upon analysis of all the work up to this point, path-planning with the integration of soaring has not been investigated so far in the context of UAVs. In addition, no simulations were built to perform path planning of UAV missions where thermals were considered a factor.

4 SIMULATION

With the definition of the objects that will be simulated, now what remains is how to simulate.

4.1 FLIGHT CONTROLLER

Before delving into the physical aspects, the definition of the autopilot that will be used is crucial, as its choice will determine which softwares can be used along with it, including the available physical simulators.

(Ebeid et al., 2018) describes and compares features of flight controller hardware, flight controller software, and physical simulators. PX4 Autopilot (PX4 for short) is an open-source autopilot for drones; this is one of the flight controller platforms explored by their work.

PX4 was created in 2009; since then, it has evolved and been widely adopted by the industry, while also being used in academic work. It has support for a wide variety of simulators to interact with and has excellent performance on small UAVs.

Due to these reasons, PX4 is the autopilot used to perform the tasks presented in the following chapters.

4.1.1 PX4 Software in the Loop

PX4 supports both Software In the Loop (SITL) simulation, where the flight stack runs on a computer (either the same computer or another computer on the same network) and Hardware In the Loop (HITL) simulation using a simulation firmware on a real flight controller board. Since this work will be performed on simulated hardware, SITL will be used to simulate autopilot software.

4.2 PHYSICAL SIMULATOR

The physical simulation refers to the simulation of the drone's hardware and the landscape where it will act.

Figure 2.3 highlights the hardware of a drone, but in this scenario, the payload will not be needed. The physical simulator must simulate the hardware of the flight controller, the motors, sensors, radios, and the frame of the aircraft. And it also has to provide ways to interact with the hardware to apply forces on it, simulating the effect of a thermal.

Gazebo is a set of libraries capable of simulating high-fidelity scenarios of the real world, leveraging various robotics applications. It has modules for diverse cases, from aerodynamics applications to LiDARs (Light Detection and Ranging). The PX4 community has integrated some UAVs to be simulated with Gazebo. Among them is the aircraft that was previously defined in 2.1. Therefore, it is convenient to use it to develop the following activities.

4.3 SIMULATION ARCHITECTURE

The following figure illustrates how the software of a drone would interact with Gazebo.

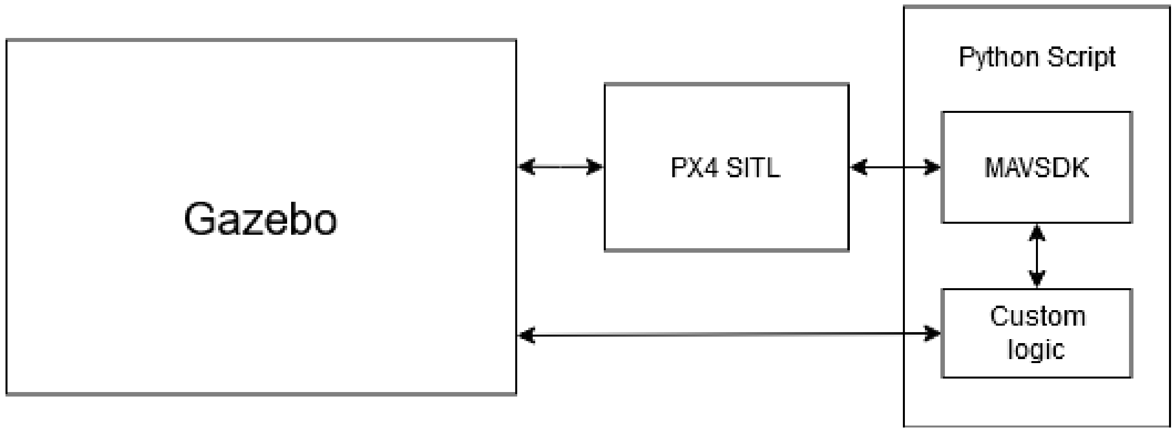

Figure 4.1: Simulation setup architctecture overview

Gazebo will be responsible for simulating the real world: the wind resistance, the three-dimensional space, objects, and the drone hardware.

PX4 SITL is the autopilot that controls the UAV, and the Python script will act as a ground control station that is responsible for managing the drone's activities. It is also possible to connect another ground control station to monitor flight progress in real time.

4.3.1 Communicating with the flight controller via MAVLINK

Communication between the Python script and the autopilot takes place through the MAVLink protocol. MAVLink is a lightweight messaging protocol for communicating with drones, built over UDP.

The interface of communication is abstracted via MAVSDK, a Software Development Kit that abstracts the application of the MAVLink protocol, making it easier to interact with the UAV.

And the custom logic highlighted in the diagram in Python Script, will implement the functionalities described in chapter 5.2. Mainly related to monitoring and

4.3.2 How PX4 SITL interacts with Gazebo

The uORB (micro Object Request Broker) topics play a crucial role in communication between the PX4 autopilot and the Gazebo simulator. This communication is essential to simulate sensor data and actuator responses during flight simulation. This communication happens in both ways.

Gazebo simulates sensors such as GPS, IMU, barometers, cameras, etc. The simulation sensor data are published by Gazebo plugins on uORB topics via the MAVLink interface or a dedicated PX4-Gazebo plugin. PX4 modules (e.g. estimators and controllers) subscribe to these uORB topics to process the incoming sensor data.

PX4 generates actuator outputs (e.g., motor speed, servo positions) based on its control algorithms. These actuator commands are published as uORB topics and sent to Gazebo via the plugin interface. Gazebo then uses this data to simulate the physical response of the UAV.

4.3.3 Interaction between scripts and Gazebo

The interaction that occurs between the Python script and Gazebo in this scenario is described in 4.5.

4.4 RUNNING THE SIMULATION

This section describes which dependencies and how they should be run in order to implement the architectured simulation.

4.4.1 Dependencies

- OS: Ubuntu 22.04.04 Jammy Jellyfish
- Physical/Graphical Simulator: Gazebo Harmonic
- Autopilot: PX4 SITL
- Ground Control Station: Python Script

The ground control station will be the Python Script established in figure 4.1. It was developed using MAVSDK for the specific case of this project; it monitors the drone activities, flashes missions into it, and, furthermore, it is used to apply thermal force on Gazebo. It can be found in a GitHub repository referenced in appendix A.1.

4.4.2 Start the simulation

Gazebo simulator must be started along with PX4 SITL. This can be done through a Makefile target present in the PX4 SITL project, through the following command:

```
cd PX4-Autopilot
make px4_sitl gz_rc_cessna
```

When the simulator is properly running, the Python Script must be pre-configured via the configuration file, and started using a Bash command:

```
cd drone-thermal
python3 maestro.py
```

4.5 SIMULATING A THERMAL

The forces exerted by a thermal are not uniform; their powers vary across its area of effect, just like its direction. Therefore, the simulator must provide a way to apply an arbitrary force to an entity.

Gazebo makes some of it's features available through a publisher/subscriber module that exposes the functionalities of other Gazebo components.

One of it's features allows the application of a force over the center of mass of an entity, and it is accessible through a topic denominated /world/[WORLD_NAME]/wrench/persistent. The force is abstracted as a three-dimensional vector, where each axis represents the acceleration on that direction using meters per second as the unit. Using this topic, we can simulate the effect of a thermal on a body.

Whenever the drone enters one of the thermals in the environment, the ground control station will detect it and trigger the appropriate routine to apply the effects of a thermal on the UAV.

Figure 4.2 highlights the force that acts upon an aircraft during flight. The Lift L is the vector that will receive the push from the thermal in the simulation.

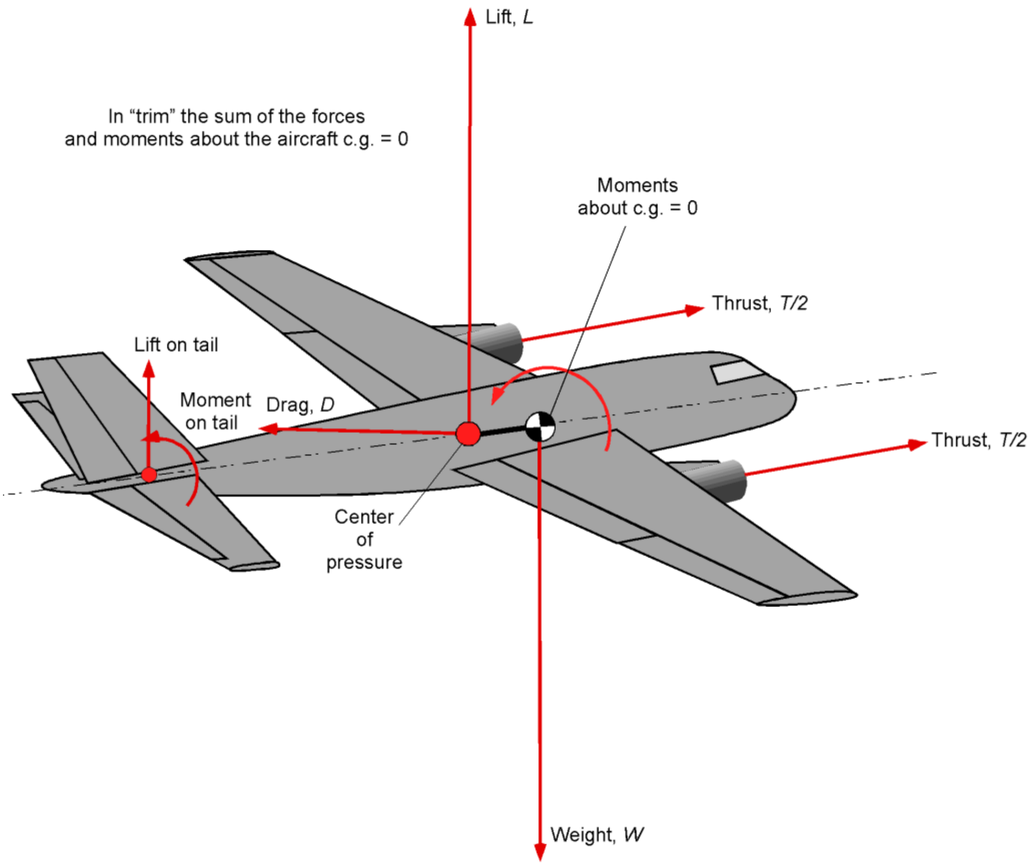

Figure 4.2: Forces distribution on an aircraft (AirspaceHK, 2024)

4.5.1 Applying persistent force

As stated before, Gazebo topics can be used to apply forces over entities. Here is an example of how this can be done. The following shell command is used to apply a force of 5 meters per second on the Z axis, over an entity named 'rc_cessna_0', on a world named 'default':

```
gz topic -t /world/default/wrench/persistent \
-m gz.msgs.EntityWrench -p \
'entity: {
    type: 2,
    name: "rc_cessna_0"
},
wrench: {
    force: {x: 0, y: 0, z: 5}
}'
```

4.5.2 Clearing the force of an entity

And on the moment the aircraft leaves the area of the thermal, its force must have its effect cleared. The topic /world/[WORLD_NAME]/wrench/clear can be accessed through a bash command, and it serves to clear the force being applied over an entity named 'rc_cessna_0':

```
gz topic -t /world/default/wrench/clear -m gz.msgs.Entity -p\
'type: 2, name: "rc_cessna_0"'
```

5 FLIGHT

After defining the actors and scenario of the proposed simulation, the actions that the UAV will perform must be defined. Also, how the thermals would be exploited by it.

Drones move through space performing complex movements that are executed by the autopilot. But who controls the autopilot? The flight controller can receive commands from various sources depending on its flight mode.

The available flight modes of drones depend on each flight controller stack, as well as on the UAV. These flight modes can be manual or autonomous. Manual being a mode where flight is commanded by a human pilot, and autonomous being a mode where the whole flight is performed by the autopilot with no human interference.

In this setup, the drone will initially fly in mission mode, a form of autonomous flight, and in chapter 5.4.3 another flight mode will come into play.

5.1 MISSION MODE

In this mode, the vehicle executes a predefined flight plan, which typically includes items to control taking off, flying a sequence of waypoints, capturing images and/or video, deploying cargo, and landing. This mission is then flashed into the drone's hardware, and then the autopilot can run it.

5.2 PLANNING MISSIONS

Missions can be very complex in their construction, the missions that will be executed on this set-up are a set of waypoints that a UAV needs to follow one after the other, from the take-off stage to the last waypoint. These waypoints are a list of XYZ coordinates, where the 0,0,0 coordinate is the starting point of the aircraft. The unit of each axis is meters. Through this representation, we can use relative distances and positions to build the mission that will be executed, rather than using absolute latitude and longitude. The mission will be planned by the ground control station and flashed into the UAV before its take-off stage.

In figure 5.1 there is a screenshot of a ground control station that can be used to plan and flash missions. It is also capable of configuring the autopilot parameters and monitoring the flight of multiple aircraft at the same time.

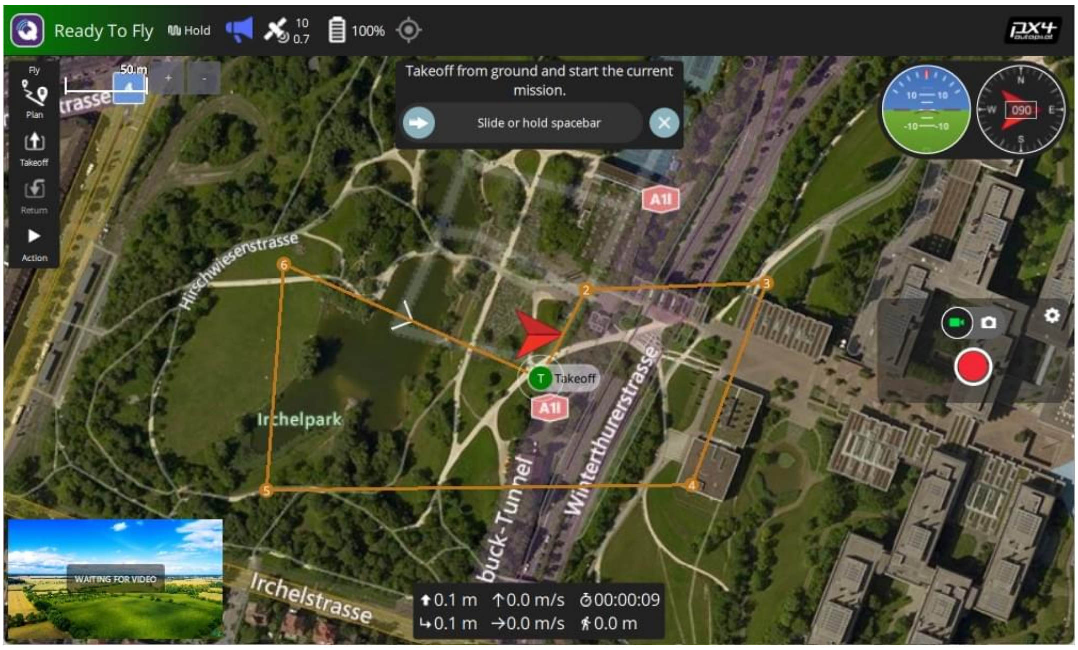

Figure 5.1: Screenshot of QGroundControl ground control station (QGroundControl, 2024)

In the simulation architecture, Python will flash the mission using MAVSDK, acting as the ground control station.

5.3 MISSION WITH THERMALS

During any point of the mission, a thermal that is present in the environment could come in contact with the aircraft and affect its flight. This exerted force is then compensated for by the autopilot in order to steer the drone towards mission completion.

The UAV will be tracked by the ground control station while flying at all times. Its position, speed, and attitude are fetched by the mission commander and can be used not only to monitor its activity, but also as input for other routines.

Knowing the status of the UAV in real time and information about thermals around the flight area, is then possible to utilize these data as inputs for a routine that triggers different behaviors on the drone.

5.4 EXPLOITING THERMALS

The force created by a thermal lift can be used by birds in complex ways during their flight. They do it for a series of reasons:

- Energy Conservation: Large birds, especially those with broad wings, such as eagles, hawks, vultures, and storks, use thermals to soar without flapping their wings, saving energy during long flights.
- Migration: Many birds use thermals during migration to travel great distances. By "hopping" between thermals, they can maintain altitude while gliding forward.
- Search for Food: Birds of prey use thermals to ascend and then glide to scan large areas for prey from above.

- Traveling Over Land: Thermals are more common on land (where the ground heats the air) than over water, so birds often follow land-based routes during migration.

They achieve those objectives using two techniques: soaring and gliding. While soaring, birds enter these thermals in order to gain height, circling in it while gaining altitude, lifted by the warm air flow. And once they reach sufficient altitude, they leave these hot currents and begin gliding forward, gradually descending until they reach the next thermal. Figure 5.2 illustrates how these techniques are applied by a bird and what role each variable plays.

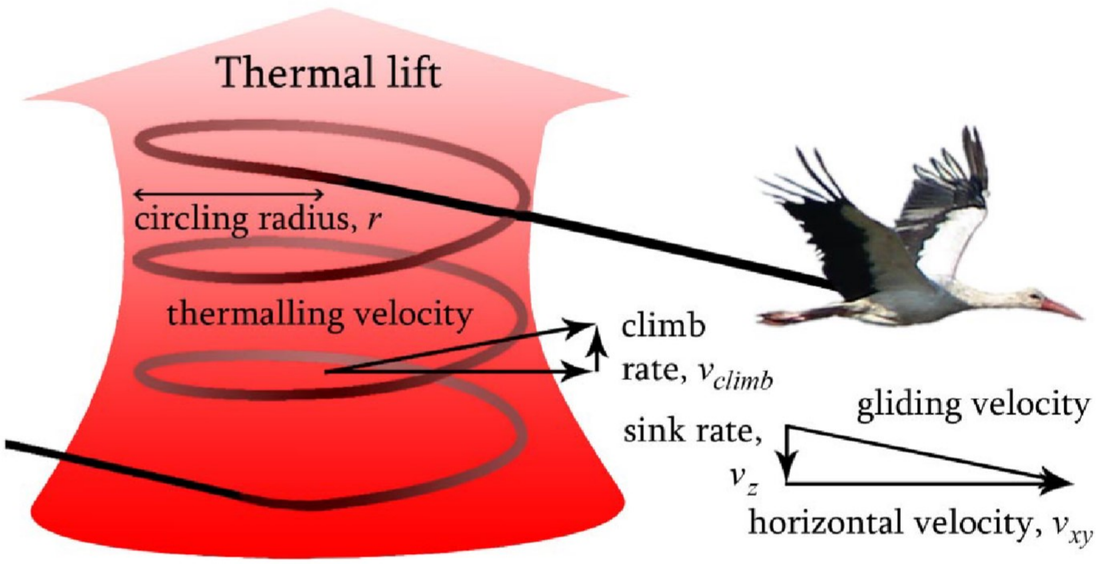

Figure 5.2: Bird soaring a thermal in order to gain altitude (Akos et al., 2010)

UAVs will change their flight altitude frequently. Sometimes they need to gain altitude, which demands energy, and sometimes they need to descend to lower altitudes, by gliding, which requires less energy. They are capable of modifying their altitude by controlling its ailerons, which stabilize the drone in the required position, and the rotation of the propellers. The action of spinning the propellers consumes the most energy during flight, without considering payloads and other specific cases such as drones with companion computers[1] in their architecture.

While maintaining altitude during flight, in the default configuration, a RC Cessna spins its propellers, generating a throttle of 33% of the maximum throttle of the aircraft. And, while climbing, this throttle percentage increases to 50%. Increasing the throttle means increasing energy consumption and reducing battery life. Therefore, minimizing the throttle needed to perform movements will save energy for the aircraft.

5.4.1 Decision loop

Using the behavior of birds as inspiration, the algorithm 1 is defined. Overall, the idea is to constantly compare the drone's current position with the position of each thermal that it is aware of. If one of these thermals is worth exploiting, then it goes inside the thermals area, exploits it, and resumes the mission.

[1]Companion computers ("mission computers") are separate on-vehicle computers that are connected to the flight controller, and which enable computationally expensive features like collision prevention.

Algorithm 1 Routine to scan and exploit thermals

```
Procedure ScanForThermals(Drone, Thermals)
while Drone is flying do
    for all T ∈ Thermals do
        if WorthExploiting(Drone, T) then
            ExploitThermal(Drone, T)
        end if
    end for
end while
```

5.4.2 Heuristic

The process of deciding if is worth exploiting is related to the decision between simply going for the next waypoint, or expending more energy to gor for a thermal and then exploiting it to save energy in the long run. Would less energy be spent if the drone simply went to its next waypoint, or would it save more energy if it exploited a thermal and then went to the next waypoint?

This is not a question that can be simply answered and requires a thorough analysis of all the factors involved. This choice depends on the aircraft, the environment, the thermals, the mission, etc. Heuristically, this is a decision problem and can be optimized for each case.

If the heuristic is too broad, the UAV will try to exploit more thermals than it should, wasting energy and time. And if its too narrow, the aircraft will not take advantage of the air currents when it would be advantageous.

Algorithm 2 presents an approach for this decision based on the next mission waypoint altitude and the distance between a thermal and the drone. The assumption here is that if the drone is too far from the thermal, it will possibly expend more energy going for that thermal than it would save by exploiting it. To delimit this, a threshold for that is represented by the 'Max worth altitude' on line 8. Increasing the threshold will allow the drone to go for farther thermals, and the inverse affects it accordingly. The distance between the drone and the thermal is calculated by the Euclidean distance between the positions of these two actors.

Algorithm 2 Deciding wether or not a thermal is worth exploiting

```
1: Function WorthExploiting(Drone, Thermal)
2: drone_pos ← Drone global position
3: thermal_pos ← Thermal global position
4: next_waypoint ← Drone mission next waypoint
5: alt_diff ← next_waypoint altitude − Drone altitude
6: if alt_diff > Minimal worth altitude then
7:     dist ← EuclideanDistance(drone_pos, thermal_pos)
8:     if dist < Max worth distance then
9:         return True
10:    end if
11: end if
12: return False
```

In addition to it, on line 6 a 'Minimal worth altitude' value is presented. This is related to the logic that if the drone does not need to gain enough altitude to reach the next waypoint, it will not exploit the given thermal. Again, it might be more expensive to exploit the thermal than to follow the mission without additional maneuvers.

5.4.3 Riding a thermal

After the decision is made, the process of going to the thermal and exploiting it begins. Algorithm 3 presents a way to stop the mission, exploit the thermal, and then resume the mission.

Algorithm 3 Exploiting a thermal

Procedure ExploitThermal($Drone, Thermal$)
$Drone$ stop mission mode
$Drone$ trigger offboard mode
$desired\_altitude \leftarrow Drone$ mission next waypoint altitude
$Drone$ enter $Thermal$
while $Drone$ altitude $<$ $desired\_altitude$ **do**
    $Drone$ loiter inside the thermal while gaining altitude
end while
$Drone$ stop offboard mode
$Drone$ resume mission mode

In the beginning of this chapter, the existence of different flight modes is mentioned. These flight modes alter the behavior of the drone, and how it's controlled.

In order to make the drone go to a specific position in its flight space - the thermal area in this case - the drone must stall the mission issued to it and adopt some other behavior. Offboard mode of the PX4 autopilot is a mode in which the flight controller accepts commands from external sources during flight. The vehicle obeys position, velocity, acceleration, attitude, attitude rates, or thrust/torque setpoints provided by some source that is external to the flight stack, such as a companion computer or ground control station. Commands issued to the autopilot are processed regardless of the conclusion of the previous command issued. In this case, the external source is the script that acts as a ground control station.

After the script detects that a thermal is worth exploiting, it orders the UAV to stop mission mode and switch to offboard mode. The aircraft is then commanded to enter the thermal. When within range, the thermal starts to push the drone upward and then the UAV can start loitering until it reaches the desired altitude. The thermal force is cleared and the mission is resumed; only then the drone start following the next waypoint.

5.4.4 Result

After implementing the algorithm established on this chapter, multiple simulations were ran using the same mission, with a thermal worth exploiting on a part of the flight, but gradually increasing the thermalling velocity of that thermal.

Analyzing figure 5.3, it is possible to conclude that the increase in thermalling velocity did have a positive implication on the reduction of the throttle needed to climb while loitering. When a null thermal force is applied, the throttle rate is around 50%, as expected. From this point on, increasing the force linearly decreases the throttle needed to perform the same climb. Exploiting a thermal that applies 8 m/s of thermalling velocity resulted in a decrease of 15% in the necessary throttle.

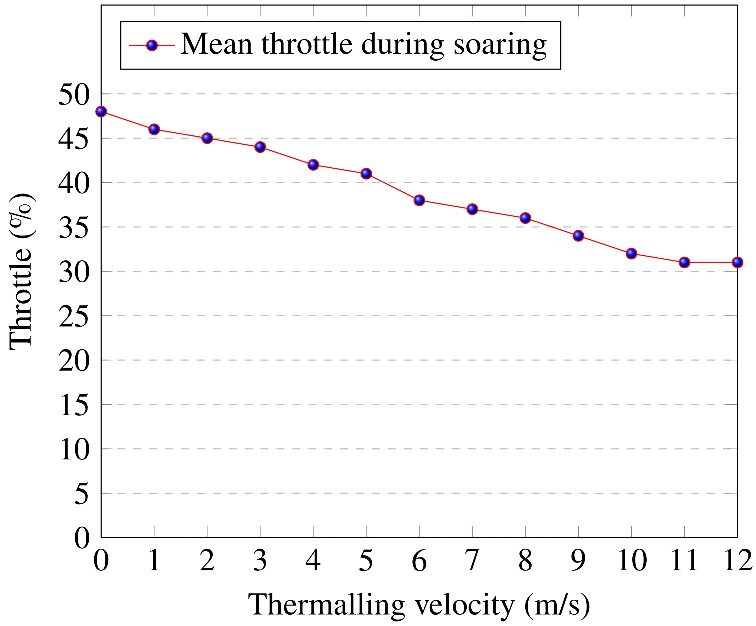

Figure 5.3: Necessary throttle to climb, in respect of thermal speed

Apart from saving energy, exploiting a thermal has also presented deviation on the path established by the drone. Figures 5.4 and 5.5 present the paths traveled by two drones that execute the very same mission, but one is exploiting a thermal of 5 m/s and the other of 10 m/s, respectively. These thermals have a radius of 40 meters which is delimited by the red circle in the figures.

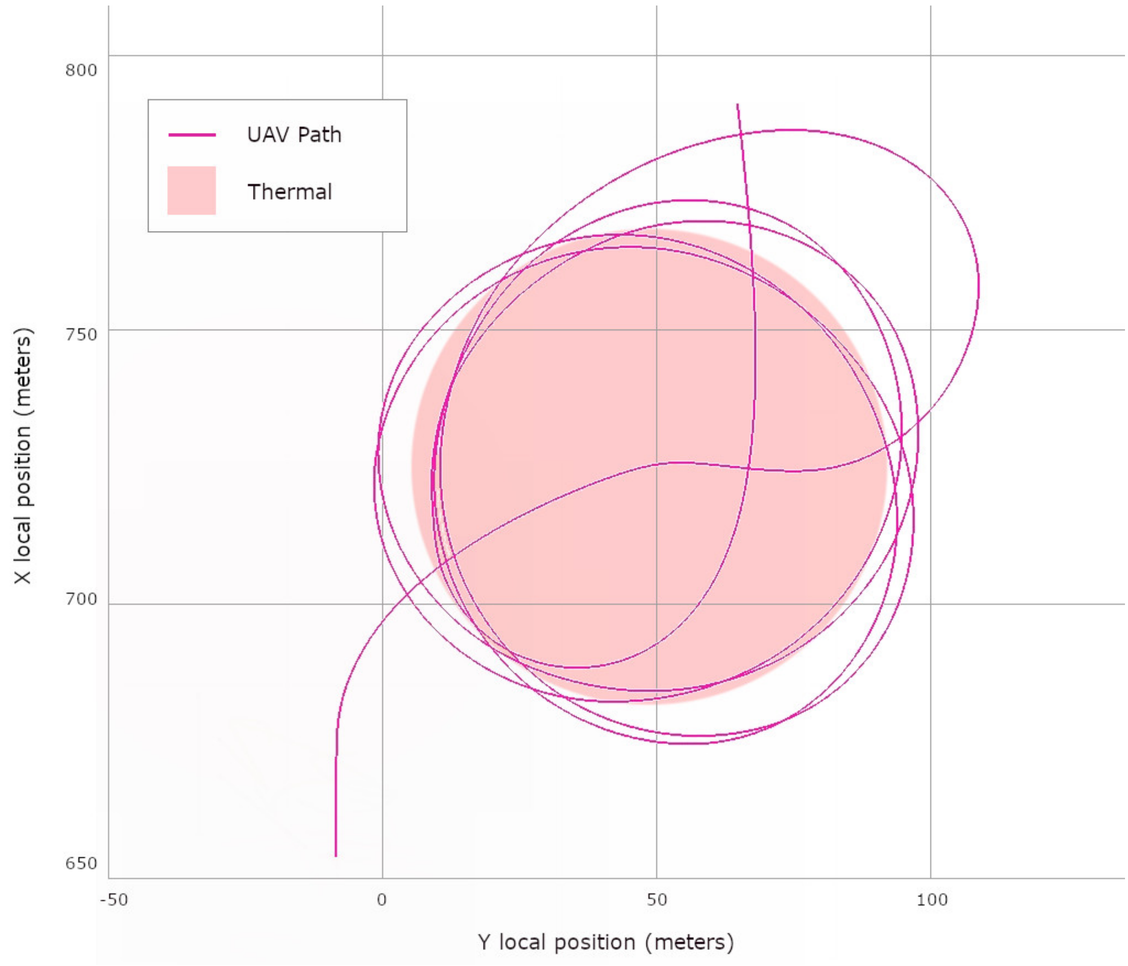

Figure 5.4: Path trace of an UAV exploiting a thermal of 5 m/s thermalling speed

When comparing the trajectory of figure 5.4 with the set point, the UAV had a smoother flight, able to more easily steer to stay within the thermal area, climbing without much deviation from the established setpoint.

On the other hand, the path traveled by the craft in figure 5.5 was much steeper. Every time the UAV came into contact with the thermal flow, the aircraft received a huge impact, forcing the autopilot to perform longer maneuvers to counteract the effect of the airflow.

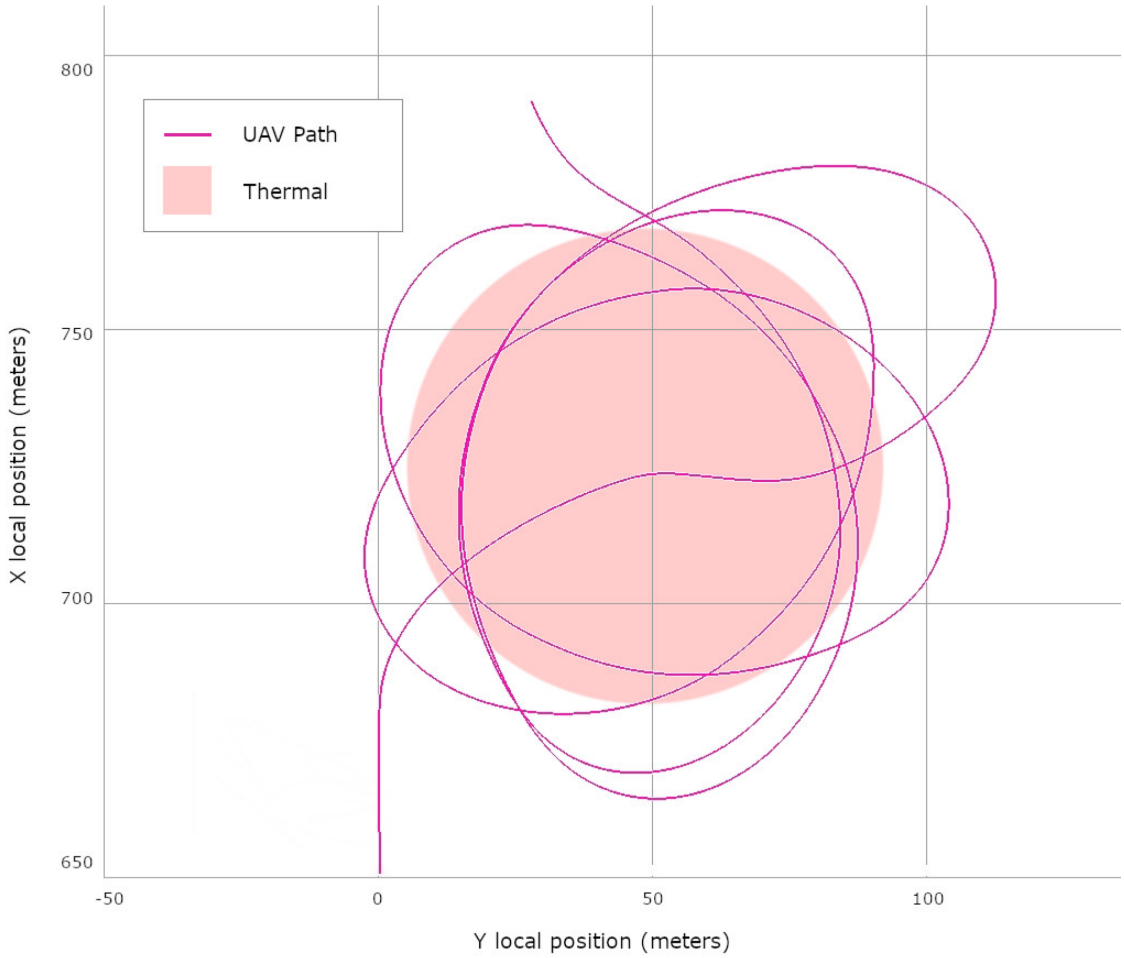

Figure 5.5: Path trace of an UAV exploiting a thermal of 10 m/s thermalling speed

Observe that the paths the drone followed while exploiting these different thermals were very distinct. This difference in maneuvers performed due to different thermal speeds is also reflected in the throttle along the flight of this phase.

Figures 5.6 and 5.7 display the throttles in the time slice in which the drones of Figure 5.4 and 5.5 were flying, respectively.

Analyzing the throttle during the climb on figure 5.6, it is possible to see that there is a low point right at the beginning of the activation of offboard mode; this is the point where the drone has decided to go for that thermal. Once it reaches the thermal, the climb starts and the throttle goes up to 50% again, then the throttle starts oscillating, ranging from 44% to 48%. That oscillation is due to the UAV leaving and entering the thermal area, causing the throttle to diminish when it is in and increase when it is out of the thermal and needs to perform maneuvers to enter it once again.

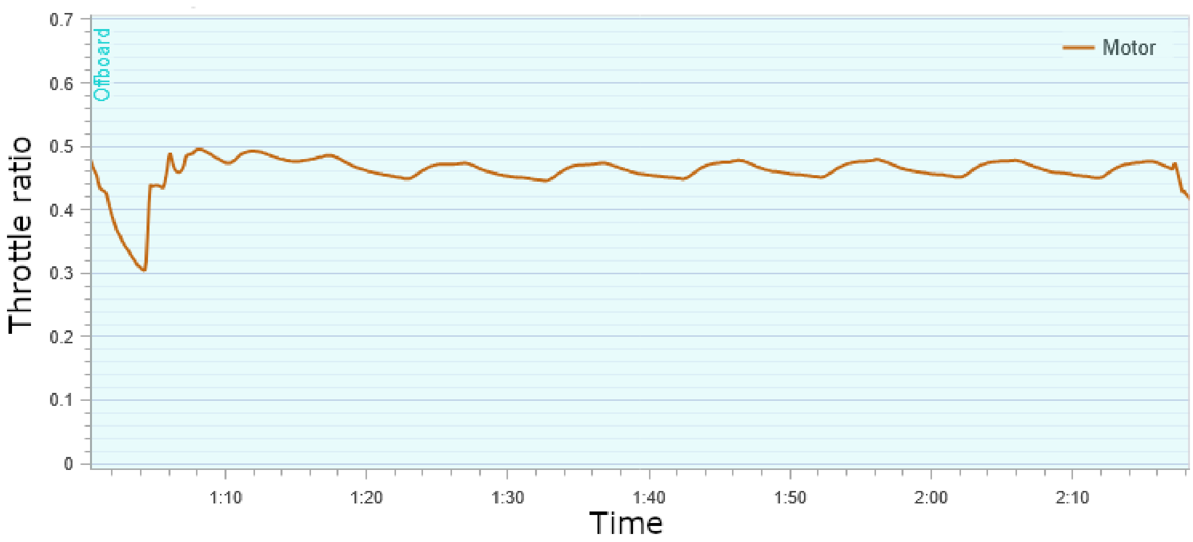

Figure 5.6: Throttle ratio of a UAV during offboard mode while exploiting a thermal of 5 m/s thermalling speed

In contrast, the throttle from 5.7 behaves almost exactly the same in the beginning of the thermal exploitation. However, after it reaches the thermal point, the oscillation is more acute compared to the flight of the figure 5.4. When the UAV enters the thermal, the throttle is reduced to 37% due to the greater thermalling velocity of that airflow. And because of that force, it ends up having to execute longer maneuvers to stay in course. These longer maneuvers lead the drone out of the thermal, and having to increase its throttle again to maintain its setpoint.

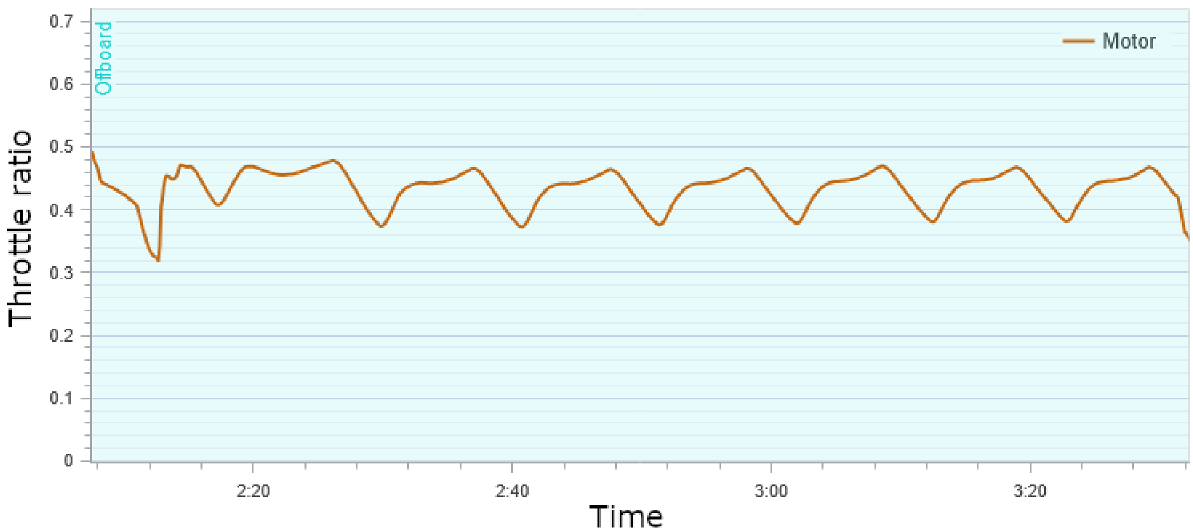

Figure 5.7: Throttle ratio of a UAV during offboard mode while exploiting a thermal of 10 m/s thermalling speed

Comparing the throttle in both these figures to 5.3, it is possible to notice that the throttles never decreased as low as described on it for the thermal velocities tested. That is because the autopilot has not enough time to adapt its propellers to the ever-changing stimulus caused by leaving and entering the thermal.

With further analysis, we can conclude that exploiting a thermal of greater force will proportionally save more energy of the aircraft, and also make the UAV perform more aggressive movements in order to stay in contact with the thermal. In both of these cases, exploiting a thermal was worth the effort as they were not costly to reach, and exploiting them always led to less energy expenditure to climb.

6 CONCLUSION

This chapter concludes our study by summarizing and analyzing what we achieved. Finally, future research that may be done to expand the work on simulations of UAV missions with thermals will be presented.

6.1 SUMMARY

This work has successfully developed a high-fidelity simulation platform to explore UAV flights utilizing thermals as a tool. The study has demonstrated how exploiting these atmospheric phenomena can significantly reduce energy consumption, therefore extending flight duration.

In addition to achieving its primary objectives, this research uncovers the potential to improve UAV operational strategies through advanced simulation tools. The results underscore the value of integrating thermals into UAV mission planning, not only for energy savings but also to improve UAV autonomy and efficiency.

Future research directions include refining thermal modeling for greater realism, optimizing UAV maneuverability to harness thermal lift effectively, and extending the study to diverse UAV configurations and real-world scenarios. This work contributes to the fields of UAV mission planning, paving the way for innovative applications in both research and industry.

6.2 FUTURE WORK

This work leaves great room for expansion on its tangents, due to its interdisciplinary nature and relatively few explorations done by the academic community.

6.2.1 Improve movement to exploit thermal

The drones attitude during the exploitation of a thermal affects the aircraft intimately and therefore its energy consumption. The way the UAV exploited it in this work was by using the default mechanisms of the offboard mode from PX4, being ordered to loiter inside the thermal area, while receiving its lift. This behavior could be further improved by changing the attitude of the drone while doing so. Changing the maximum pitch, yaw, and roll, increasing or decreasing maximum airspeed, could decrease the need for abrupt maneuvers and lead to a better use of the thermals power.

6.2.2 Optimize thermal selection

Selecting the thermals worth it depends on multiple factors, and data on these thermals are crucial to make these decisions. Better definition and representation in thermals help to make better informed selections. All of this could be used to avoid poor thermal selection, which could lead to energy savings. Optimizing this process can be of great value in future research.

6.2.3 Improve thermal simulation

Thermals simulated on this work were not as faithful to their real world counterparts, for they are irregular masses of air, both in shape, direction, and force. Modeling these phenomena with more accuracy will lead to more precise results, affecting all aspects of this work.

6.2.4 Test other aircrafts

RC Cessna is the most standard fixed wing aircraft on the market. It is widely used by amateurs and non commercial applications. While on industrial and military solutions, other aircrafts can also be used.

The results presented by this work encompass many of the general aspects of the fixed wing UAVs, but these results will vary according to the model of each aircraft. From weight to shape, it affects all the components of it. Changing aspects of energy consumption, autopilot algorithms, throttle, etc.

Building tests on thermals using different aircrafts is also a point that could be explored to further improve the simulation and algorithms presented by this research.

REFERENCES

AirspaceHK (2024). Pof. 04 introduction to aircraft stability. https://airspacehk.com/courses/pof-06-introduction-to-aircraft-stability/. Accessed: 18/12/2024.

Akos, Z., Nagy, M., Leven, S., and Vicsek, T. (2010). Thermal soaring flight of birds and unmanned aerial vehicles. *Bioinspiration & biomimetics*, 5(4):045003.

Cobano, J. A., Alejo, D., Sukkarieh, S., Heredia, G., and Ollero, A. (2013). Thermal detection and generation of collision-free trajectories for cooperative soaring uavs. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2948–2954. IEEE.

Dake (2006). Thermal column. https://en.wikipedia.org/wiki/File:Thermal_column.svg. Accessed: 05/12/2024.

Ebeid, E., Skriver, M., Terkildsen, K. H., Jensen, K., and Schultz, U. P. (2018). A survey of open-source uav flight controllers and flight simulators. *Microprocessors and Microsystems*, 61:11–20.

Fan, J., Chen, X., and Liang, X. (2023). Uav trajectory planning based on bi-directional apf-rrt* algorithm with goal-biased. *Expert Systems with Applications*, 213:119137.

Guo, Y., Liu, X., Jia, Q., Xuhang, L., and Zhang, W. (2023). Hpo-rrt*: a sampling-based algorithm for uav real-time path planning in a dynamic environment. *Complex Intelligent Systems*, 9:7133–7153.

Lu, P. and Geng, Q. (2011). Real-time simulation system for uav based on matlab/simulink. In *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering*, volume 1, pages 399–404.

PX4 (2024a). Basic flying (fixed wing). https://docs.px4.io/main/en/flying/basic_flying_fw.html. Accessed: 05/12/2024.

PX4 (2024b). Gazebo vehicles. https://docs.px4.io/main/en/sim_gazebo_gz/vehicles.html. Accessed: 05/12/2024.

PX4 (2024c). Px4 system architecture. https://docs.px4.io/main/en/concept/px4_systems_architecture.html. Accessed: 05/12/2024.

QGroundControl (2024). Qgroundcontrol user guide. https://docs.qgroundcontrol.com/master/en/qgc-user-guide/index.html. Accessed: 05/12/2024.

Sparrow, E., Husar, R., and Goldstein, R. (1970). Observations and other characteristics of thermals. *Journal of Fluid Mechanics*, 41(4):793–800.

Wen, J., Ji, H., Wang, H., Zhang, M., Li, D., and Wu, J. (2019). Design of a real-time uav fault injection simulation system. In *2019 IEEE International Conference on Unmanned Systems (ICUS)*, pages 767–772.

Xu, D., Qian, H., and Zhang, S. (2021). An improved rrt*-based real-time path planning algorithm for uav. pages 883–888.

Zhang, J., Geng, Q., and Fei, Q. (2012). Uav flight control system modeling and simulation based on flightgear. In *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*, pages 2231–2234.

APPENDIX A – ASSETS FOR SIMULATION

A.1 GITHUB REPOSITORY FOR PYTHON SCRIPT

`https://github.com/vudala/drone-thermal`