DOI: 10.13009/EUCASS2023-625
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

MINI-APTERROS: A GNC demonstrator for Vertical Landing and Thrust Vector Control systems

Hadrien AFSA [1], Gwendal COLLET [2], David TCHOU-KIEN [3]*

[1] MCA Ingénierie, 38-48 Rue Victor Hugo, 92300 Levallois-Perret, France
[2] University of Rennes, Rue du Thabor, 35000 Rennes, France
[3] Centre National d'Etudes Spatiales (CNES), 2 Place Maurice Quentin, 75001 Paris, France
** Corresponding author, hadrien.afsa@mca-groupe.com*

Abstract

Designing the guidance and control laws of a Vertical Take-off Vertical Landing (VTVL) rocket with a Thrust Vector Control (TVC) system often proves to be a challenge, as it requires complex simulation platforms to experiment and fine-tune a large number of parameters to achieve the desired performance. To overcome this issue and facilitate the process of development of such systems, PERSEUS is developing a small-scale and low-cost rocket as a test platform for implementing Guidance, Navigation and Control (GNC) algorithms for TVC and VTVL. This demonstrator, called Mini-Apterros, not only allows quick testing, tuning and validation of the control laws and navigation solutions with real flights but also to experiment with exploratory GNC technologies such as artificial intelligence. Such technologies could drastically change the way GNC systems are developed.

DOI: 10.13009/EUCASS2023-625

1 Introduction

The PERSEUS (Projet Étudiant de Recherche Spatiale Européen Universitaire et Scientifique) programme from the French Space Agency (CNES) offers students and future engineers the opportunity to develop their interest and vocation towards the space sector by participating in ambitious and collaborative projects aimed at developing reusable rockets through the incremental development of Guidance, Navigation and Control (GNC) demonstrators.

Reusable rockets require overcoming and mastering several challenging technologies, such as Thrust Vector Control (TVC), Reaction Control Systems and Vertical Take-off and Vertical Landing (VTVL) capabilities. Implementing such technologies demand extensive developments, testing and validation through time-consuming and expensive processes. Therefore, the concept of the Mini-Apterros (Mini-Advanced Propulsion Technology for Reusable Rocket and Operating System) has been introduced as a testbed to ease the incremental development toward reusability and to experiment with innovative technologies.

Initiated as a student challenge, Mini-Apterros is a small-scaled, electric-propelled rocket that aims to perform VTVL flights with TVC. The demonstrative objective of this project is to perform a vertical take-off one meter above the ground, a stationary flight, a horizontal translation of three meters, and a safe vertical landing. Three versions of Mini-Apterros are being developed by distinct student teams with the same project constraints: Each version has an electric propulsion system with four synchronized electric turbines, a maximum weight of 10 kg and four legs to stand on.

Each version also focuses on different approaches. Mini-Apterros from the University of Rennes uses a simple Proportional, Integral, Derivative (PID) controller with an air deflection nozzle for TVC and cold gas thrusters for roll control, while Mini-Apterros from ENSEA Cergy has a similar algorithmic approach but uses smaller electric turbines for roll control and implements the TVC by orienting the propulsion system.

Finally, with regard to the disruptive and exploratory ambitions of the PERSEUS project, ENSAM Bordeaux follows the same mechanical approach but uses artificial intelligence with Deep Reinforcement Learning (DRL) to train a neural network through a multi-physics simulation to control the demonstrator.

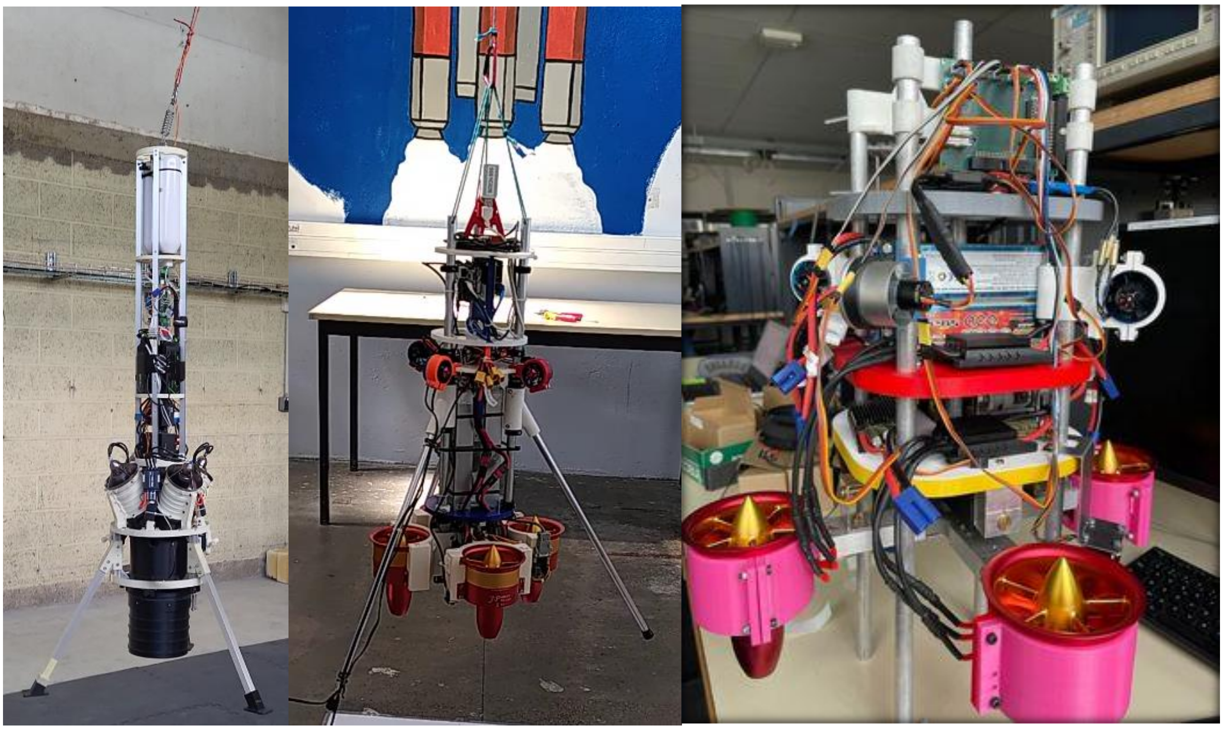

Figure 1 – The three versions of Mini-Apterros:
a) University of Rennes; b) ENSAM Bordeaux; c) ENSEA Cergy

This article focuses on the University of Rennes and ENSAM Bordeaux versions, as they are the most advanced to this date. A technical description and the development processes of each demonstrator will be covered, as well as the first results obtained and the future implementations to come.

DOI: 10.13009/EUCASS2023-625

2 Mini-Apterros from the University of Rennes

2.1 General description

The Mini-Apterros of the University of Rennes is a model of a reusable stage with vectored thrust. Unlike the other versions, the four electric turbines are fixed on the vehicle, and the TVC is performed by orienting the airflow with a controllable 'nozzle'. The demonstrator weighs 12.5 kg and is 153 cm tall. The mass is currently higher than the authorised limit but will be eventually reduced to satisfy this constraint.

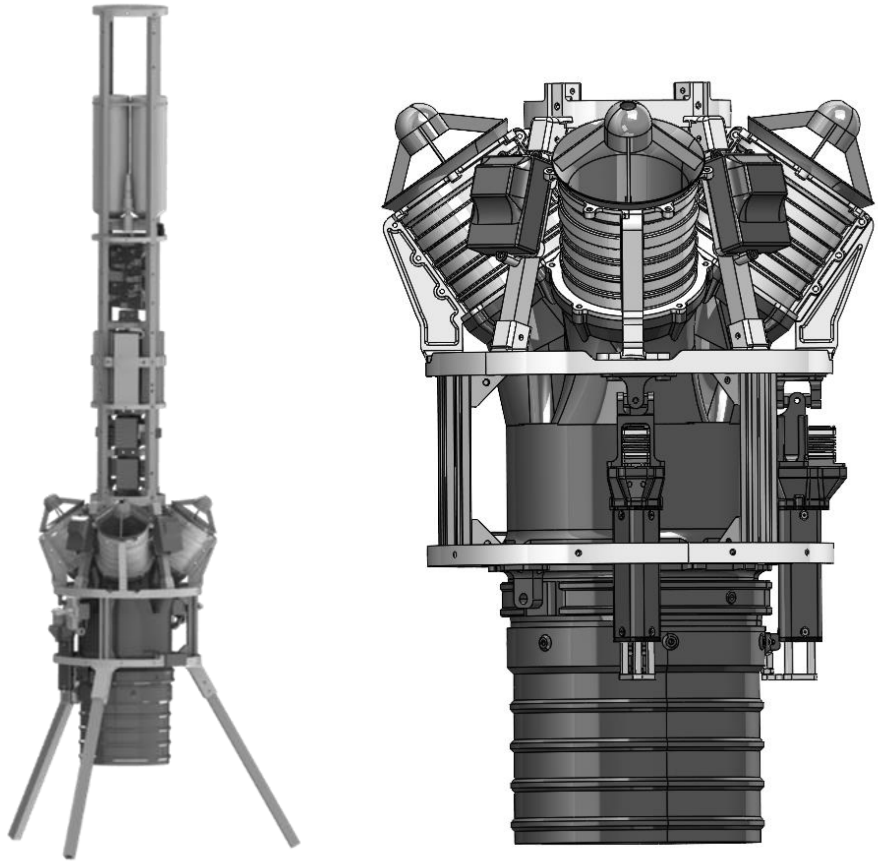

Figure 2 – a) CAD view of Mini-Apterros from the University of Rennes; b) close-up view of the propulsion system

2.1.1 Propulsion system

Each EDF EPF Hobby Mercury II 90mm electric turbine is controlled by a 150 A electronic speed controller and can generate up to 47 N of thrust. For space requirements, they are mounted at a 35-degree angle, and a merger redirects and merges the airflows towards the ground to maximise the thrust (Figure 3). This total airflow is then redirected by the nozzle mounted with a gimbal ring and oriented by two linear electric actuators.

DOI: 10.13009/EUCASS2023-625

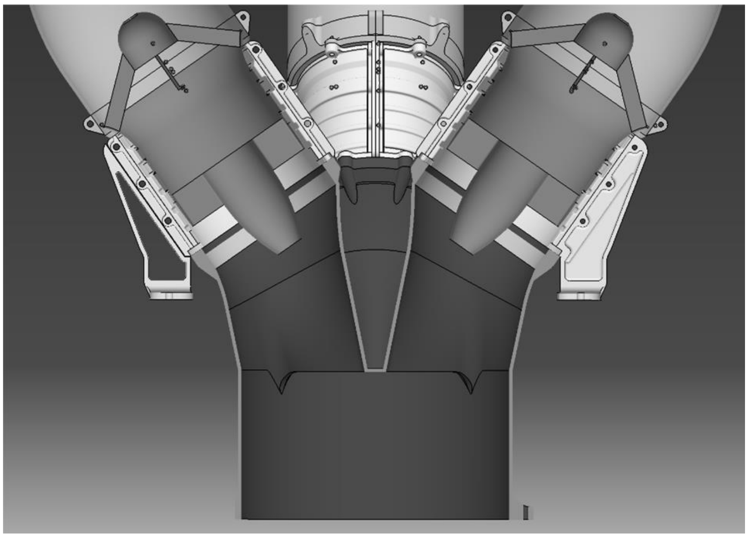

Figure 3 - Sectional view of the airflow merger

However, the current design of the airflow merger significantly reduces the thrust generated by the airflow as it causes a pressure drop, mainly due to friction and the reduction of the cross-sectional area. This part will be redesigned and improved to maximise the thrust provided by the system.

2.1.2 Roll control system

As the propulsion system is not strictly symmetric partly because the turbines do not have the same speed for a given thrust and different latency times, an induced torque is observed on the main axis, resulting in an increasing roll rate during the flight. In order to stabilise the flight and compensate for this parasite torque, a cold gas roll control system is being implemented on the demonstrator. It is using a 1.1 L paintball tank pressurised at 200 bar with air. The compressed air is expanded to 6 bar by two regulators, and the output is controlled by 4 solenoid valves arranged in pairs on opposite sides to avoid generating a torque around other axes. The gas expulsion can generate around 0.3 Nm of torque, which is sufficient to compensate for the parasite torque that was estimated at around 0.2 Nm.

As this system is not fully implemented yet, a temporary solution was adopted to reduce it. It consists in controlling each turbine individually to correct for the slight asymmetrical behaviour. This solution is not compliant with the project constraints and will be removed when the roll control system is ready for use.

2.1.3 Avionics definition

The avionics of a reusable rocket includes all the electronic and computer systems that ensure the control, monitoring and navigation of the demonstrator, allowing it to orient itself, communicate and carry out precise manoeuvres during the flight.

These tasks are performed by the different components that the demonstrator embarks:

- An on-board computer (OBC), responsible for collecting, processing and managing the data needed for precise navigation, system control and strategic decision-making throughout the flight, thus guaranteeing mission success. The OBC chosen for this application is a BeagleBone Black computer development card. It is equipped with 512 MB of RAM, an Ethernet port, USB ports, an HDMI connector, a micro-SD card slot and a 2x46-pin expansion interface.
- A 9-degree of freedom (DoF) Inertial Measurement Unit (IMU), including accelerometers, gyroscopes and magnetometers. The unit chosen is the Ellipse-E from SBG Systems.

2.2 Flight software

The flight software, implemented within the OBC, carries all the communication and data processing during the flight. It is responsible for executing the algorithms and command sequences that precisely control the various stages of flight, from initial launch to soft landing, thus ensuring the rocket's stability, safety and optimal performance. Among other tasks, it receives the data from the IMU, computes and controls the nozzle's actuators and stores the relevant data for exploitation.

DOI: 10.13009/EUCASS2023-625

The flight software is developed in embedded C and uses *RobotControl*, a robotics-focused library for embedded Linux computers, as well as the sbgECom library for communication with the IMU. The software is structured in distinct and independent threads, each dedicated to a specific part of the overall algorithm. It allows modularity and to isolate each block to test them individually. The flight-software structure is shown in Figure 4. Four threads are used: one dedicated to the communication with the IMU and three threads dedicated to each of the GNC components. They are controlled by a fifth main thread.

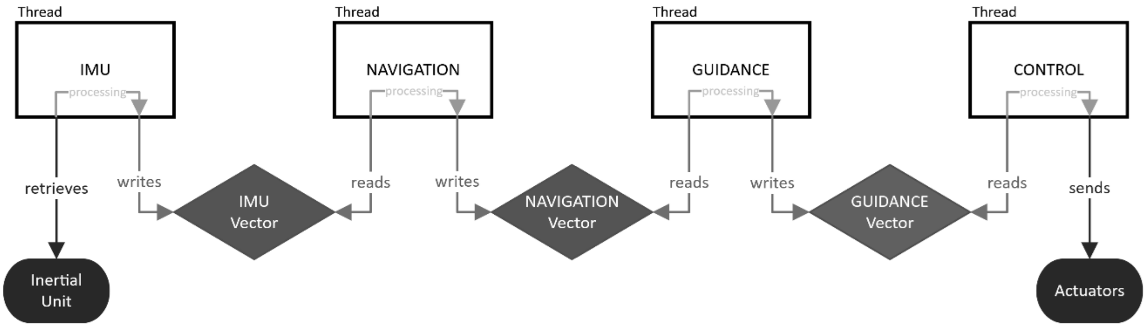

Figure 4 – Flight Software block diagram (within the main thread)

The Navigation, Guidance and Control blocks are further detailed in the following sections.

2.2.1 Navigation

The navigation thread gets the raw IMU data and computes the speed and position of the rocket in real time, by using the rotation matrix given by the IMU and integrating the accelerations using a simple Riemann sum method. Thanks to the high accuracy of the embedded sensors, the precision obtained for the position is sufficient for guidance and control. Prior to the flight, a calibration mode is used to average the raw data and estimate the bias of the sensors, therefore increasing the accuracy of the calculation performed in the thread.

The data obtained is sent to the control thread.

2.2.2 Guidance

The guidance thread contains the flight plan, including the desired list of positions and speeds that the demonstrator has to follow during its flight. It consists of a list of waypoints (i.e. coordinates) that are reached one after the other. No attitude constraint is given, except that the roll angle (around the main axis) should stay near zero.

When a waypoint is reached, i.e. that the current position of the demonstrator is close to the waypoint within a fixed tolerated margin, the guidance thread sends the coordinates of the next one to the control thread.

Additional instructions were added for specific sequences such as the landing phase, especially to take account of the touchdown. During this sequence, the horizontal speed is controlled accordingly to the altitude: the vehicle must slow down when it gets close to the ground. This behaviour is illustrated in Figure 5. When the touchdown is detected from the IMU data, the turbines are slowly stopped.

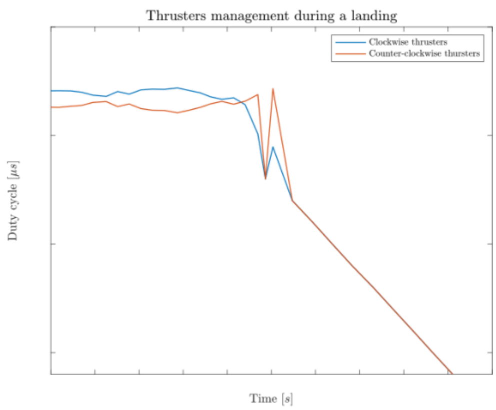

Figure 5 – Thrusters management during a landing sequence

DOI: 10.13009/EUCASS2023-625

2.2.3 Control

The control thread calculates in real-time the corrections to be applied to the orientation of the demonstrator by orienting the thrust of its engines to follow the planned trajectory and desired parameters.

The six state variables must be controlled for the demonstrator to fulfil its mission: position (x, y), altitude (z), and attitude. Depending on the variable, proportional and integral correctors are used to determine the commands to send to the actuators.

In the current state of development, the thrust and orientation control algorithms have been set and tuned for the system. As the roll control system is not implemented yet, the roll angle (around the vertical axis) is not controlled. These algorithms are responsible for lifting the rocket at the desired speed, maintaining it at the desired altitude, and stabilising it to perform a stationary flight. The thrust is also controlled to lower the rocket's altitude with a controlled speed for safe landing. For instance, Figure 6 schematises the thrust control algorithm for altitude control. The entrance of this algorithm, i.e. the commanded altitude z_c, is provided by the guidance thread.

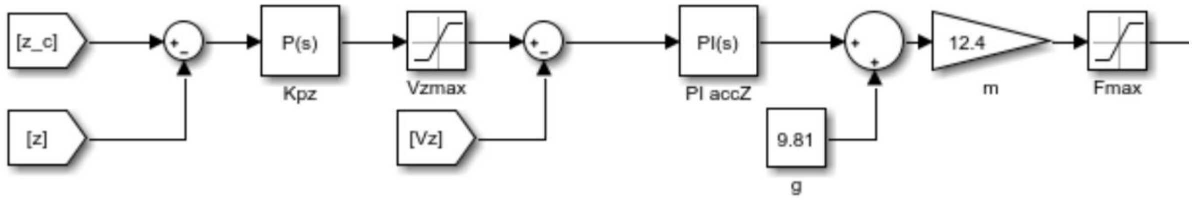

Figure 6 – Thrust control algorithm

The tuning of the correctors' gains was performed empirically, using the Ziegler-Nichols method and fine-tuning with real flights to attain the desired behaviour.

The control thread is also responsible for controlling the actuators of the demonstrator. It sends the commands obtained from the previously mentioned algorithm as duty cycles to the two linear actuators of the nozzle and to the electronic speed controllers of the turbines, controlling the overall thrust of the propulsion system.

A response characterisation of the propulsion system was performed to correlate the effective thrust obtained from a duty cycle. This was done by weighing the demonstrator down so that the total mass of the system was greater than the maximum thrust, ensuring that it would not lift with full power. The demonstrator was then hanged to a dynamometer connected to an Arduino to record, after calibration, the value of the weight at a given time. It was then possible to correlate the thrust values as a function of the duty cycle sent to the turbines, and establish a linear regression (Figure 7) in order to obtain the equation to calculate the duty cycle (in μs) to be sent to the turbines as a function of the thrust calculated by the guidance thread.

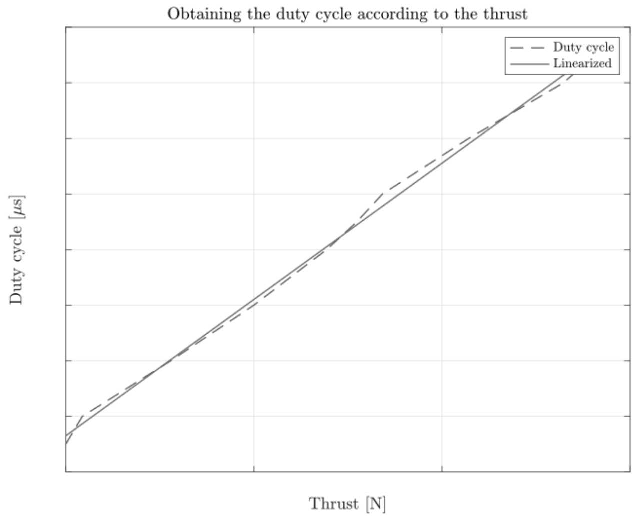

Figure 7 – Duty cycle curve as a function of thrust and its linearisation

DOI: 10.13009/EUCASS2023-625

2.3 Flight tests and results

Several tests were run to validate the different systems of the demonstrator. The test area consists of a 3x3x5 meters gantry with a safety cable that holds the demonstrator in case of failure (the security cable can be seen in Figure 1 a). The flight starts above the ground so that the cable is not taut under normal conditions.

Several take-offs and stationary flight tests were successfully run, with the demonstrator effectively reaching the desired altitude and maintaining it with accuracy for the whole duration of the flight, approximately 10 seconds (for preserving the batteries for several tests, the autonomy of the demonstrator is around 30 seconds). The roll rate is sufficiently low and the position does not drift significantly. Most of the position error observed is mainly due to the drifting position calculated by the Navigation system and can be reduced with an absolute positioning system that still has to be implemented.

2.4 Future work

To prepare for the full mission flight, including the horizontal translation and soft landing, the roll control system must be implemented and tested. To improve the navigation system, additional sensors can be implemented and coupled with the IMU for lower drift.

Besides, as the power autonomy of the demonstrator might be insufficient for the full flight, it will be increased by optimising the structure and reducing the overall mass of the vehicle. It also serves the purpose of respecting the project constraint. The redesign and replacement of the airflow merger will also help improve its performance.

3 Mini-Apterros from ENSAM Bordeaux: Using artificial intelligence for guidance and control

3.1 General description

The Mini-Apterros version of ENSAM Bordeaux follows the same project requirements but uses a slightly different approach for the TVC system. The four electric propulsion turbines are held on a platform mounted on a gimbal and oriented by two linear actuators. This design does not require a merger or a nozzle for redirecting the airflow, therefore leading to a much more compact and lightweight demonstrator. It has a height of 87 cm and weighs 9.1 kg. It is propelled by four EDJ JP Hobby 90 mm electric turbines capable of up to 4.3 kg of thrust each. The propulsion system is powered by a 6S 22 Ah 25C LiPo battery, allowing for up to 4 minutes of full flight autonomy.

Another difference with the other versions of Mini-Apterros lies in the guidance and control system, which uses Artificial Intelligence rather than PID controllers to explore the use of such technology in space vehicles.

Finally, a roll control system is implemented to stabilise the demonstrator around its vertical axis.

3.1.1 Structure and mass optimisation

A mass optimisation of the structure was made to reduce it and to gain performance and autonomy. Some structural parts were redesigned using topology optimisation in order to reduce their mass while keeping their mechanical strength. For instance, the clamps that hold the main turbines were optimised using the Solid Isotropic Material with Penalization method in Altair Hypermesh and OptiStruct (Figure 8). The obtained design was then 3D printed to be mounted on the demonstrator. This optimisation allowed to reduce the mass by ~30%, or around 151 grams. This technique can be applied to other structural parts for further reductions.

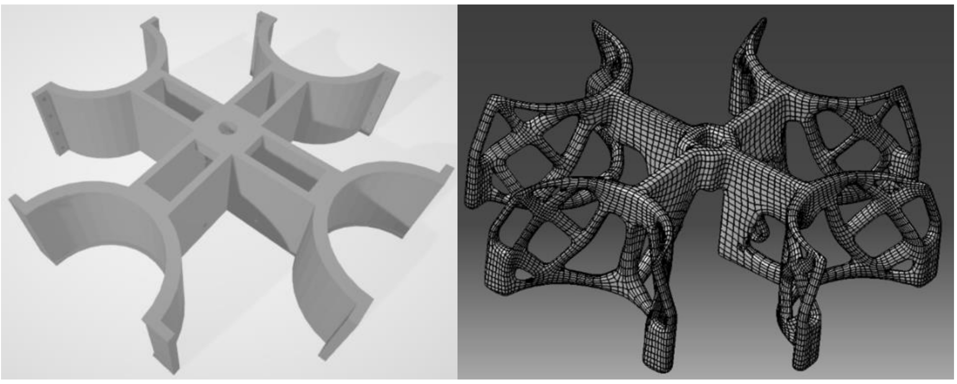

Figure 8 – Turbines plateau before (left) and after (right) its topological optimisation.

DOI: 10.13009/EUCASS2023-625

Overall, the mass of the demonstrator was reduced by nearly 15%, reaching slightly less than 9 kg, therefore respecting the constraint imposed by the project.

3.1.2 Navigation system

Like the University of Rennes' version, the navigation used for this version consists of one 9-DoF Xsens MTi-30 IMU that provides the attitude of the demonstrator, and through some computation of the position. To anticipate the potential limitations of this solution and the drifting of the position, a one-dimension VL53L0X LIDAR oriented downwards is used for precise altitude calculation. Regarding the two other positions, an indoor positioning system is being implemented. It consists of four IIDRE Ultra Wide Band (UWB) beacons positioned statically around the flight area of Mini-Apterros, and an embedded anchor that computes the position based on the received signals with a sufficient frequency for the required precision.

In the first place, these different sensors are used independently for simplicity, but a data fusion algorithm can be implemented to significantly improve the performances. For instance, the UWB anchor can be placed with a known offset from the main axis of the rocket (the roll axis) to determine its heading, together with the magnetometer and the gyroscope.

3.1.3 Roll control system

Similar to the other versions, parasite torques can be observed, mainly due to the uncontrolled asymmetrical behaviour of the turbines. Roll control is therefore required, and a different approach is made for this demonstrator. Instead of cold gas thrusters, it uses four low-thrust EDF THRUST 35 mm electric turbines that are capable of up to 170 g of thrust each. They are paired in opposite directions to generate torque only on the desired axis and limit any parasite torque. The system is powered by an additional 1350 mAh 2S Li-Po batteries.

For testing purposes, the roll control is first implemented using a basic, straightforward PID control. Eventually, the roll speed and angle will be controlled by the same neural network with the same training technique as shown earlier.

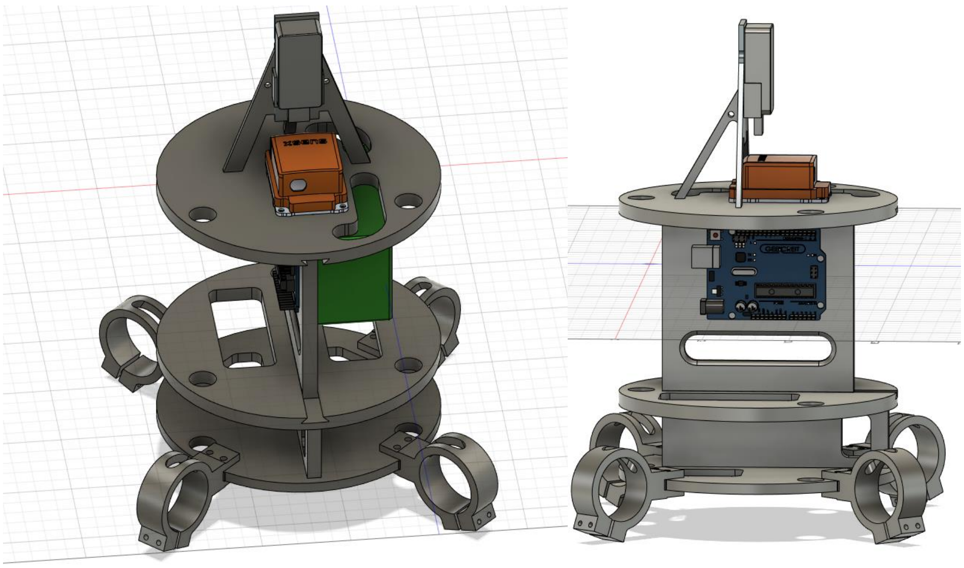

Figure 9 – CAD view of the main chassis of the demonstrator

3.1.4 Avionics

The avionics of this demonstrator is rather straightforward. It consists of a Raspberry Pi 4 Model B as the main OBC that contains the neural network and receives the data provided by the different navigation sensors (IMU, LIDAR and UWB anchor) through Robot Operating System (ROS). It sends the commands to the different actuators (TVC actuators, main electric turbines and roll control turbines) through an Arduino R3 that converts them into Pulse Width Modulation (PWM), as the Raspberry Pi 4 does not provide enough hardware PWM pins.

DOI: 10.13009/EUCASS2023-625

3.2 Neural Network Guidance and Control

The main specificity of this demonstrator is that it uses a DRL neural network instead of classical control laws, to compute the input of the different actuators in accordance with the flight plan provided beforehand. This neural network is implemented with the PyTorch open-source library in Python 3 and the Stable Baseline 3 implementations. It uses the Proximal Policy Optimization (PPO) method for training the neural network.

The data required to train the neural network is obtained through repetitive flight simulation using CoppeliaSim, an advanced physics engine with a Python Application Programming Interface (API) for real-time communication. The neural network sends flight commands to the virtual demonstrator, and a set of criteria rates its behaviour and returns it as rewards to the training algorithm. This process is illustrated in Figure 10.

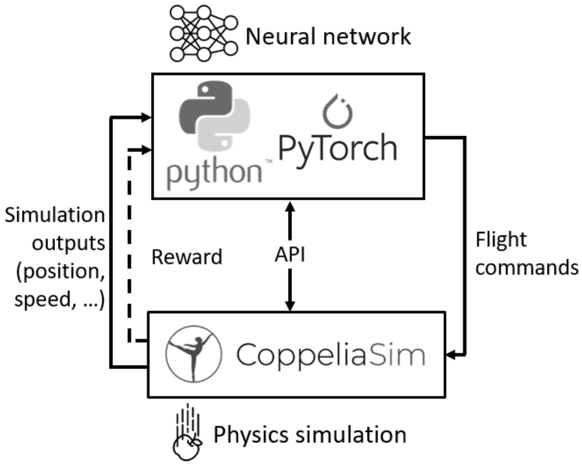

Figure 10 – Representation of the flight simulation and neural network training

The reward algorithm trains the demonstrator to have the desired flight behaviour in the different phases. For instance, the reward algorithm for the take-off phase gives higher rewards if:

- the PWM signal controlling the turbines does not exceed 85%;
- the PWM signal does not change abruptly;
- the height of the demonstrator increases slowly;
- the distance with the desired altitude is small.

Figure 11 shows a 1D simulated take-off flight and the corresponding reward values attributed.

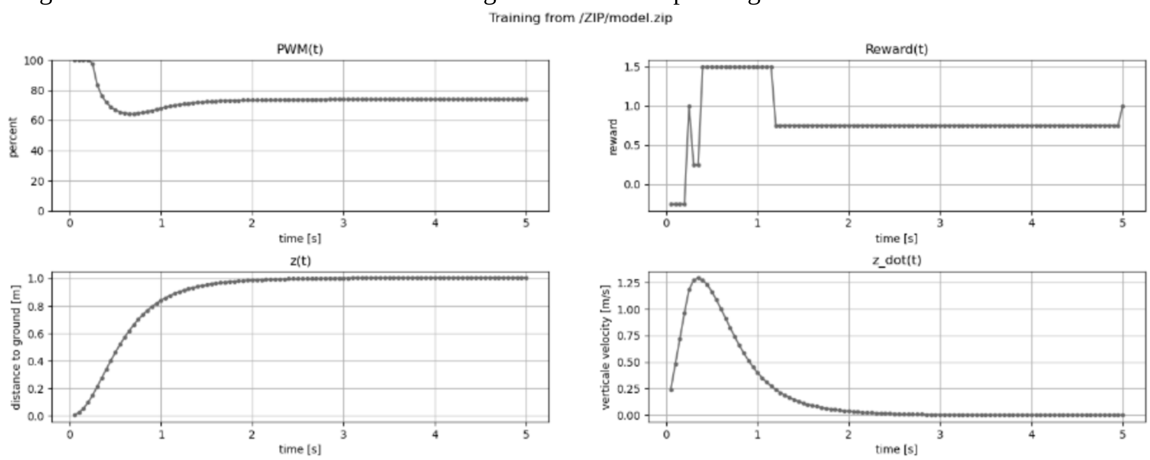

Figure 11 – Simulated take-off flight

Once the network is trained and achieves the expected results, it is deployed on the demonstrator's OBC for actual test flight.

3.3 Flight tests and results

Several constrained test flights were performed. Two security cables passing through the demonstrator's structure prevented it to have any rotation or horizontal translation, therefore leaving only one DoF. These tests aimed to validate

DOI: 10.13009/EUCASS2023-625

the training sequence of the neural network for altitude control, as well as the new mechanical structures and avionics. Other constrained flights validated the implementation of the roll control system.

3.4 Future work

In order to prepare unconstrained flights while securing the demonstrator, a safety gantry similar to the solution used by the University of Rennes is being designed and built. This gantry also adds horizontal cables that also prevent the demonstrator to fly outside of a defined volume. These cables leave a sufficient clearance and do not affect the flight (except for the weight of the cables that can be taken into account in the flight software), as long as the roll angle does not increase significantly. This solution is used for the first test flights and will be eventually removed, leaving only the vertical cable holding the rocket for security.

To prepare for a 6 DoF flight trial, the next milestone is to achieve a stationary flight controlled by the neural network trained in a 6 DoF simulation and with effective roll control. The desired trajectory mission can then be implemented.

4 Conclusion

Each demonstrator is at a different stage of development. University of Rennes' version achieved its first 10 seconds 6 DoF unconstrained stationary flight with a temporary roll control system. ENSAM Bordeaux's version is still flying with 1 DoF in constrained flights but has implemented the DRL and has more advanced navigation and roll control subsystems. Both demonstrators are pursuing their development towards the desired mission flight.

It is therefore premature to conclude which technology is the most efficient, both in terms of control performances and implementation time, and if neural networks and DRL are suitable for such applications. However, the first results obtained are promising and the development of both demonstrators will continue to reach the same stage of development, where more comparisons can be made between the two approaches.

5 Acknowledgements

This paper and the research behind it would not have been possible without the support of the University of Rennes and ENSAM Bordeaux supervisors who provided a significant help and guidance to their respective students. These acknowledgements also extend to the CNES, which provided funding for this project.