

Article

# Comparison of Flight Parameters in SIL Simulation Using Commercial Autopilots and X-Plane Simulator for Multi-Rotor Models

Michał Welcer , Nezar Sahbon  and Albert Zajdel 

Lukasiewicz Research Network—Institute of Aviation, Al. Krakowska 110/114, 02-256 Warszawa, Poland; nezar.sahbon@ilot.lukasiewicz.gov.pl (N.S.); albert.zajdel@ilot.lukasiewicz.gov.pl (A.Z.)

\* Correspondence: michal.welcer@ilot.lukasiewicz.gov.pl

**Abstract:** Modern aviation technology development heavily relies on computer simulations. SIL (Software-In-The-Loop) simulations are essential for evaluating autopilots and control algorithms for multi-rotors, including drones and other UAVs (Unmanned Aerial Vehicle). In such simulations, it is possible to compare the flight parameters achieved by flying platforms using various commercial autopilots widely used in the UAV sector. This research aims to provide objective and comprehensive insights into the effectiveness of different autopilot systems. This article examines the simulated flight test results of a drone performing the same mission using different autopilot systems. The X-Plane software was used as an environment to simulate the dynamics of the drone and its surroundings. Matlab/Simulink r2023a provided the interface between autopilot software and X-Plane models. Those methods allowed us to obtain an appropriate comparison of the autopilot systems and indicate the main differences between them. This research focused on analyzing UAV flight characteristics such as stability, trajectory tracking, response time to control changes, and the overall effectiveness of autopilots. Various flight scenarios including take-off, landing, flight at a constant altitude, dynamic manoeuvres and, flight along a planned trajectory were also examined. In order to obtain the most accurate and realistic results, the tests were carried out in various weather conditions. The aim of this research is to provide objective data and analysis to compare the performance of commercial autopilots. This method offers several advantages, including cost-effective testing, the ability to test in diverse environmental conditions, and the evaluation of autopilot algorithms without the need for real hardware. The findings of this study may have a considerable impact on how autopilot designers and developers choose the best platforms and technologies for their projects. Future research on this topic will compare the obtained data with flight test data.

**Keywords:** UAV; autopilot; SIL; flight simulation; control; multi-rotor models



**Citation:** Welcer, M.; Sahbon, N.; Zajdel, A. Comparison of Flight Parameters in SIL Simulation Using Commercial Autopilots and X-Plane Simulator for Multi-Rotor Models.

*Aerospace* **2024**, *11*, 205. <https://doi.org/10.3390/aerospace11030205>

Academic Editor: Andreas Strohmayer

Received: 25 January 2024

Revised: 1 March 2024

Accepted: 1 March 2024

Published: 5 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The use of commercial autopilots in Unmanned Aerial Vehicles (UAVs) has become a very common phenomenon. This is due, among other things, to their high availability, popularity, and broad community of users. These types of autopilots are widely used in many areas, such as agriculture, photography, surveillance, firefighting, personal use, infrastructure inspection, and much more [1]. Commercial autopilots are easy to use and install on the drone, regardless of its type. The more challenging aspects involve fine-tuning the PID (Proportional-Integral-Derivative) controllers for a particular platform, designing the mission, and mastering the art of drone piloting [2,3]. Several pieces of literature have described and investigated the use of commercial autopilots for Unmanned Aerial Vehicles. Studies such as [4] described an approach for the implementation of controllers with COTS (Commercial Off-The-Shelf) autopilot for a simple kinematic model of aircraft dynamics. The open-loop controller was implemented by using output tracking in the study, with the COTS autopilot serving as the closed-loop interface. Typically, commercial autopilots

are designed to provide only guidance loops for the purpose of waypoint navigation. However, the study performed by Kaminer, I. [5] presented a three-dimensional path following a control algorithm for off-the-shelf autopilots. The aim of implementing such an algorithm was to expand the possible range of UAV applications. Moreover, integrating commercial autopilots with UAV airframes provides several advantages and benefits, including enhancing safety by reducing human error. They aid in performing stable and precise manoeuvres, especially in dangerous and complex situations. In addition, autopilot integration provides an extension for UAV capabilities and data collection to enhance decision making, mission-planning improvements, and performance analysis [6].

In the development and testing of Unmanned Aerial Vehicles, integrating commercial autopilots with Software-In-The-Loop (SIL) simulations provides significant advantages. Such an approach was presented by [7], where the flight stabilisation system was designed and modelled by using Software-In-The-Loop (SIL) simulations [8]. The simulation environment provides a controlled environment for identifying and resolving issues early in the development cycle, minimising the need for expensive equipment and resources. Article [9] shows the usage of simulating an environment for testing the vision algorithms to determine the position of the drone. This method also enables risk mitigation by simulating various flight scenarios and potential failures in a simulated environment. In a related study focusing on UAV-based wireless sensor networks (WSNs), an improved k-means clustering algorithm was proposed to improve mission area decomposition and optimise UAV flight paths [10]. Identifying and resolving problems during tests helps minimise the risks associated with unpredictable real-world conditions, ensuring a safer and more reliable autopilot system. It is possible to create diverse and complex scenarios that may be difficult to test in the real world. Autopilots can be exposed to a wide range of environmental conditions, allowing developers to evaluate system performance and ensure robustness and adaptability. Detecting and fixing problems early reduces the likelihood of costly modifications later in the development cycle, leading to a more efficient and timely project delivery. This method allows developers to monitor and analyse system responses and quickly implement improvements. The existing literature has researched the integration of commercial autopilots with SIL simulations, such as the work presented in [11–14]. This approach was utilised by [15] to design a novel control structure that permits the explicit creation of tracking techniques for a system with an unchangeable autopilot in the loop. In real-life conditions, the presence of wind presents a significant disturbance in flight conditions; these disturbances must be taken into account during the development of the UAV system. Research taking into account wind disturbances for the design of a guidance controller was presented in [16].

This study aimed to test commercial autopilots in SIL simulations for an octocopter drone. A special environment was created to enable the comparison of two different autopilots. The configuration used allowed for a detailed comparison of the operation of different systems. Two commercial autopilots, Ardupilot and Veronte, were taken into account in this research. The strengths and weaknesses are described and the main differences between them are indicated. This study incorporates aerodynamics, flight dynamics, and motor models from the Xplane simulator, which offers a powerful and effective approach for the validation and testing of the autopilots under highly realistic flight conditions. A diverse range of environmental conditions are available for testing in the Xplane, along with sophisticated visualisations and the ability to record a wide array of flight data [17–21]. The navigation algorithms can be compared with different flight modes or the behaviour of failures during the mission. The errors of sensors can be simulated, such as signal blackouts, sensor failures [22], actuator failures, and navigation system failures. Performing such a study by integrating Ardupilot and Veronte autopilots with SIL simulations using the Xplane capability may aid in the selection of commercial autopilots to increase the effectiveness of UAV development and improve mission design. This in turn leads to a reduction in the necessary resources and development time. The usage of the

X-Plane simulator for the visualisation of the mission and Matlab/Simulink for autopilot algorithms was presented in [23].

Regarding the organisation of this work, the following Section 2 of this study presents the multi-rotor octocopter test bed. This is followed by Section 3, which extensively discusses the methods used in this research with the selected commercial autopilots and the corresponding software used for planning the missions before the real flights or simulations. Then, Section 4 presents the concept of the SIL simulation. Finally, conclusions with remarks and future suggestions are highlighted in Section 6. In this study, our main contribution is a comprehensive evaluation and comparison of two widely used commercial autopilots, Ardupilot and Veronte, in Software-In-The-Loop (SIL) simulations for an octocopter drone. By carefully examining their strengths, weaknesses, and main differences, our goal is to provide valuable information on the selection of commercial autopilots in UAV development. The research used advanced models from the Xplane simulator, providing a realistic and robust testing environment. Additionally, we study the impact of environmental conditions and simulate various scenarios to assess the effectiveness of autopilots in various challenges. Through this extensive analysis, we aim to contribute to the development of UAV development practices and improve the decision-making process regarding autopilot selection.

## 2. Test Vehicle

A coaxial octocopter UAV was selected as the test bed for this research. The model consists of eight propellers with a diameter of 51.3 cm, and the arrangement of the engines is symmetrical with a wheelbase of 86.3 cm. The model of the drone is presented in Figure 1, with its inertial parameters listed in Table 1.

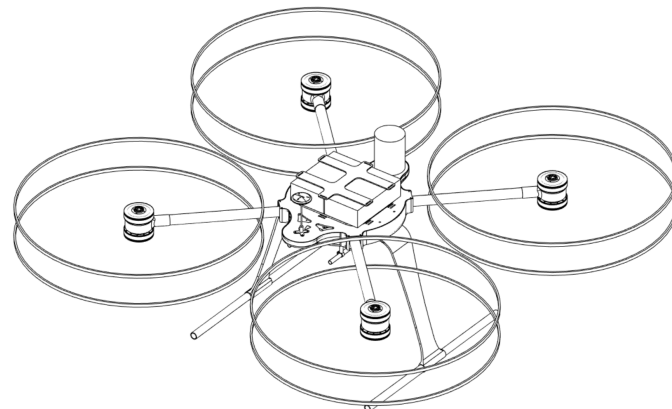


Figure 1. CAD model of octocopter.

Table 1. Mass and inertia parameters.

Parameter	Value	Units
$m$	16.2	[kg]
$I_{xx}$	0.873	[kg·m <sup>2</sup> ]
$I_{yy}$	0.909	[kg·m <sup>2</sup> ]
$I_{zz}$	1.263	[kg·m <sup>2</sup> ]

The UAV multi-rotors were configured to rotate in opposite directions to balance the torque generated by each rotor; the rotation direction and notation of each propeller are presented in Figure 2. The rotation speed of the rotors was based on commanded orientation signals in the form of the Euler angles  $\phi$ ,  $\theta$ , and  $\psi$ . The commanded orientation signals were then converted to motor commands with the aid of a mixer matrix (See Equation (1)).

$$\begin{bmatrix} \text{Motor1} \\ \text{Motor2} \\ \text{Motor3} \\ \text{Motor4} \\ \text{Motor5} \\ \text{Motor6} \\ \text{Motor7} \\ \text{Motor8} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T \cdot \begin{bmatrix} \text{Roll} \\ \text{Pitch} \\ \text{Yaw} \\ \text{Thrust} \end{bmatrix} \quad (1)$$

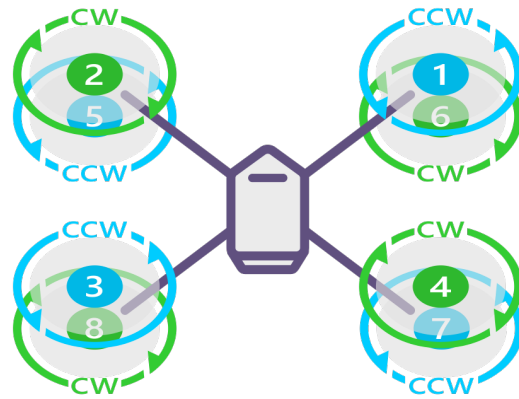


Figure 2. Motor notation and propeller rotation direction [24].

### 3. Research Methods

For the purpose of this research, the curve of the earth and its rotation were neglected. Therefore, a flat earth model was used in this study as the global coordinate system. The x-axis of the global coordinate system is directed to the north, the y-axis is directed to the east, and the z-axis points down; those axes create the North East Down (NED) coordinate system. The body-fixed coordinate system and orientation Euler angles  $\phi$ ,  $\theta$ , and  $\psi$  are shown in Figure 3.

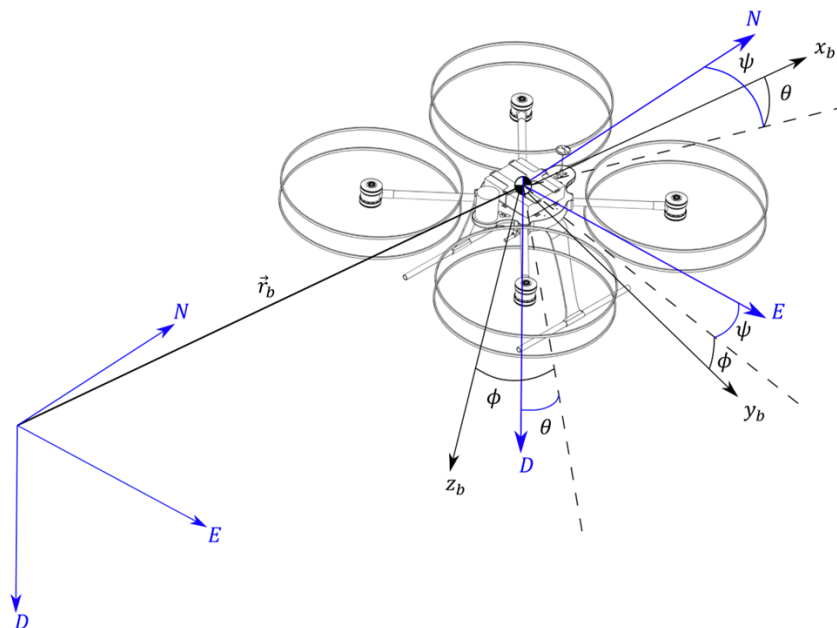


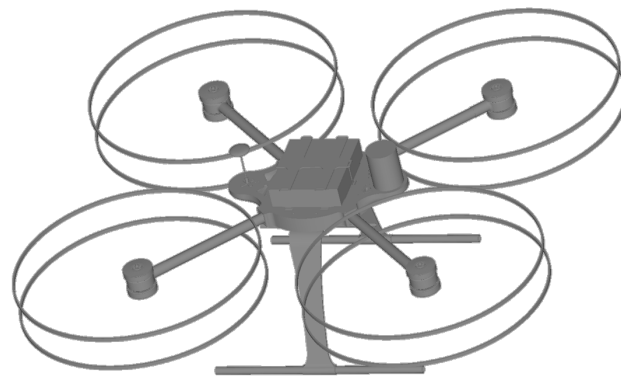
Figure 3. Coordinate systems.

The roll angle, denoted by the symbol  $\phi$ , represents the rotation about the longitudinal axis of the object. A positive roll corresponds to a counterclockwise rotation when viewing



along the positive x-axis toward the origin and negative to a clockwise rotation. The pitch angle, denoted by the symbol  $\theta$ , represents the rotation about the lateral axis of the object. The positive pitch corresponds to the rotation that points the object's nose upward, and the negative angle points the nose downward. The yaw angle, denoted by the symbol  $\psi$ , represents the rotation about the vertical axis of an object. A positive yaw angle corresponds to a counterclockwise rotation when viewing along the positive z-axis toward the origin, while the negative sign corresponds to clockwise rotation. It is essential to be aware of the coordinate system used in Figure 3. The Euler angle conventions 'ZYX' define the sequence of rotations and the axis order. Different conventions may lead to different interpretations of positive and negative rotations.

The octocopter model was developed within the PlaneMaker 11.55 program, a tool used to design flying models in the Xplane simulator. It takes into account dimensions, mass, moments of inertia, and aerodynamic coefficients. Unlike traditional simulators, the X-Plane uses blade element theory in its aerodynamic model. Blade element theory enhances this simulation approach by modelling the forces and moments acting on the aircraft and individually evaluating its component parts [25]. This segmentation was carried out in accordance with design assumptions aimed at making the dynamic model as accurate as possible to the real object. Figure 4 shows the model prepared in the PlaneMaker 11.55 software used for simulations in the X-Plane.



**Figure 4.** Model of octocopter in X-Plane simulator.

For this study, two commercial autopilots were used: the Ardupilot [26], an open-source autopilot, and the second one, Veronte [27]. Both of them are easy-to-use autopilots that can be deployed on many flying and moving platforms. Operators can install additional devices such as external transponders, LIDAR, radars, or gimbals. Veronte was developed in accordance with aviation standards such as DO178C/ED-12, DO254, and DO160, while Ardupilot is an open-source project with a large community of users and more customisation options.

For both of the autopilots, the limitations such as the Euler angles and angular velocities were set. For the pitch and roll angles, the maximum desired value is 30 degrees and the maximum angular velocities were set to 90 deg/s. When it comes to the time constant of a drone, it is largely defined by the propulsion system dynamic. The time constant of the single motor is equal to 0.1 s, and this amount of time is needed by the motor to reach 63.2% of its final value. This characteristic influences the dynamic behaviour of the octocopter. Figure 5 illustrates the motor's response to a step command. The maximum thrust of individual power units is approximately 60 Newtons.

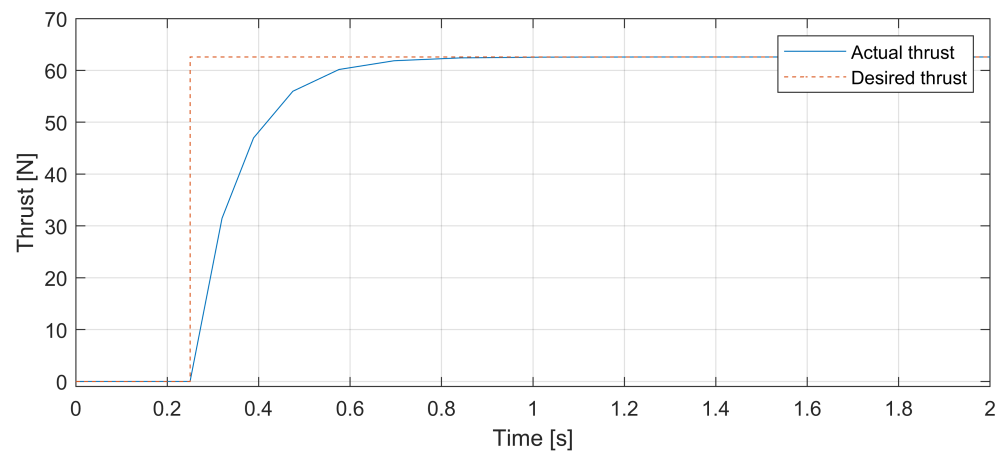


Figure 5. Motor dynamic.

Each of those autopilots has dedicated software for tuning the controllers and planning the mission. Both of them are intuitive and easy to use. Veronte provides Veronte Pipe 6.4.52 software, in which the operator has to design an autopilot structure with control loops and PID regulators. The design process involves arranging blocks in a designated area and connecting them to various signals, providing users with a detailed view inside the control loops. While this approach may be more time consuming compared to Ardupilot, it provides operators with a thorough understanding of the inner workings of the autopilot. An example of such an autopilot design is shown in Figure 6.

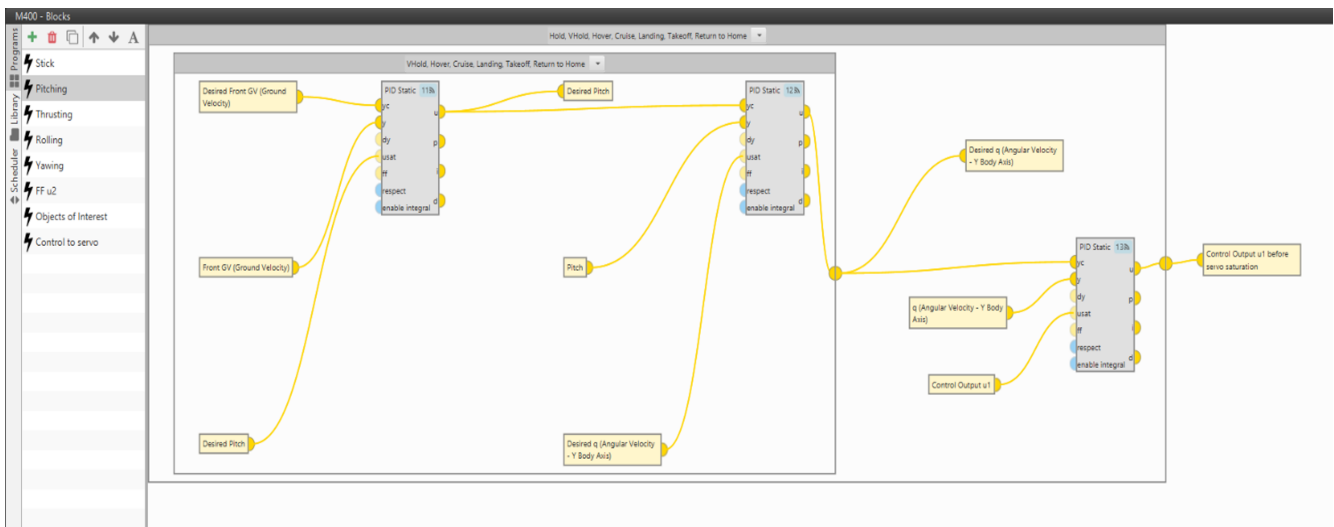


Figure 6. Example of autopilot structure in Veronte Pipe software version 4.5.

The grey blocks presented in the figure above represent PID controllers whose parameters are set by the user. Depending on the operator, it may be a simple PID controller or a more complex one. Figure 7 shows the content of this block.

Figure 8 represents the section of Veronte Pipe 6.4.52 software dedicated to planning the mission.

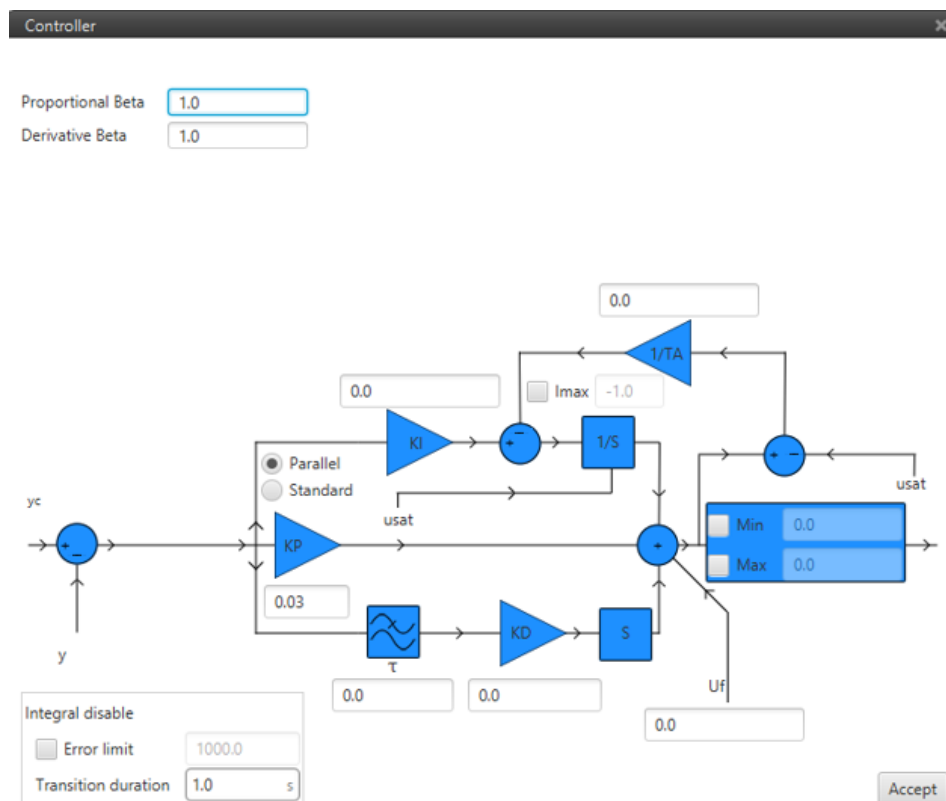


Figure 7. Content of Veronte PID block.

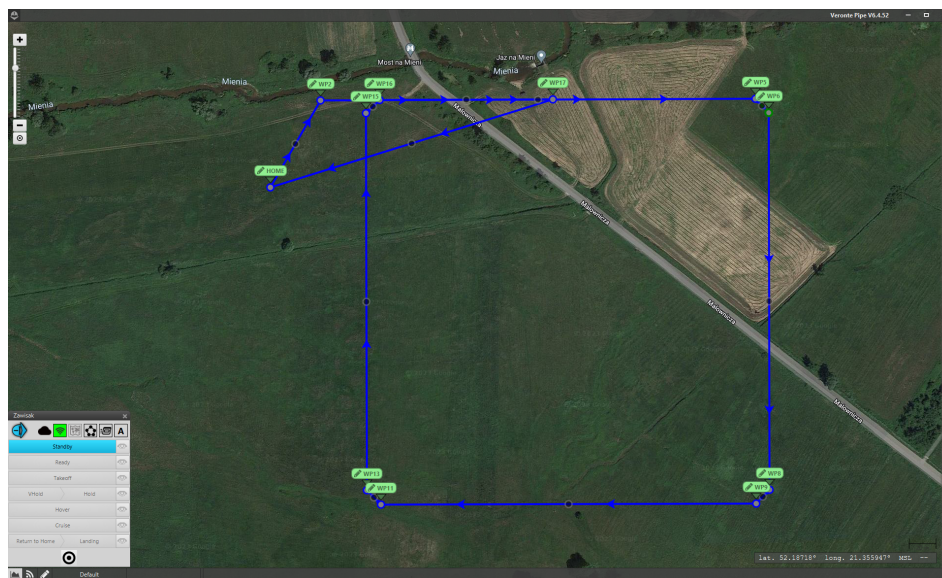


Figure 8. Ground control station of Veronte Pipe.

For Ardupilot autopilots, the MissionPlanner 1.3.80 software was used. This is ground control station software dedicated to planning, monitoring, and controlling UAV missions. It also provides tools for the real-time visualisation of flight data and the configuration of vehicle parameters. Figure 9 shows the part of the MissionPlanner responsible for planning the UAV mission. This is a user-friendly tool where waypoints can be placed on a specific point on the map by the operator or imported from a csv file.

Figure 10 presents the section where the operator can change and tune the parameters of the autopilot. Those parameters are responsible for flight and mission performance.

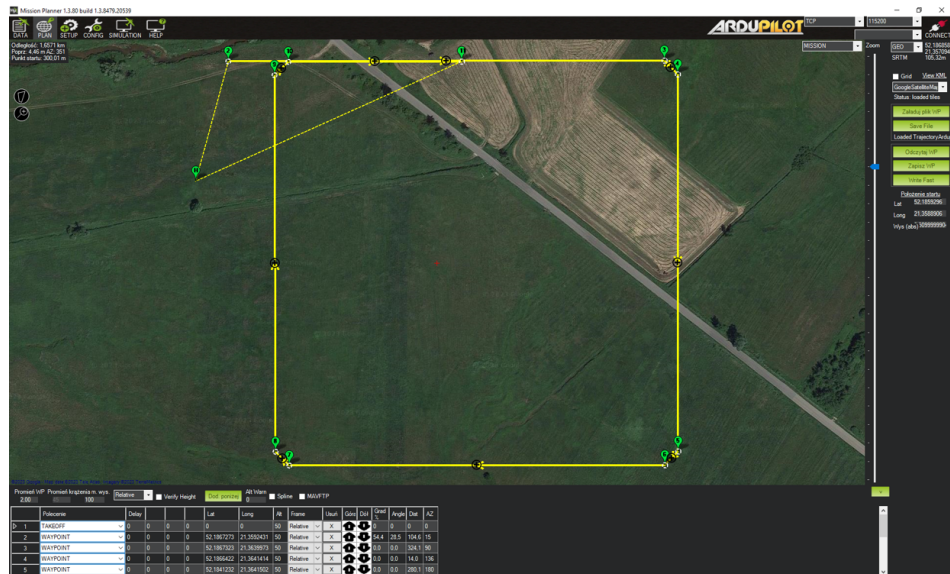


Figure 9. Ground control station section of MissionPlanner.



Figure 10. Tuning of PID and autopilot parameters in MissionPlanner.

#### 4. Simulation Environment

To replicate the environment of the mission, the X-Plane simulator was chosen. It was chosen because of the customisation options of the model and environment. The aerodynamic model of the octocopter was designed in PlaneMaker 11.55, so the aerodynamic forces can be calculated by the simulator. Parameters such as the mass and moments of inertia were taken into account during the design process. One of the drawbacks of this approach is that it is impossible to import the model from any of the CAD files; the user is obligated to redraw the model. Figure 11 presents the PlaneMaker tool.



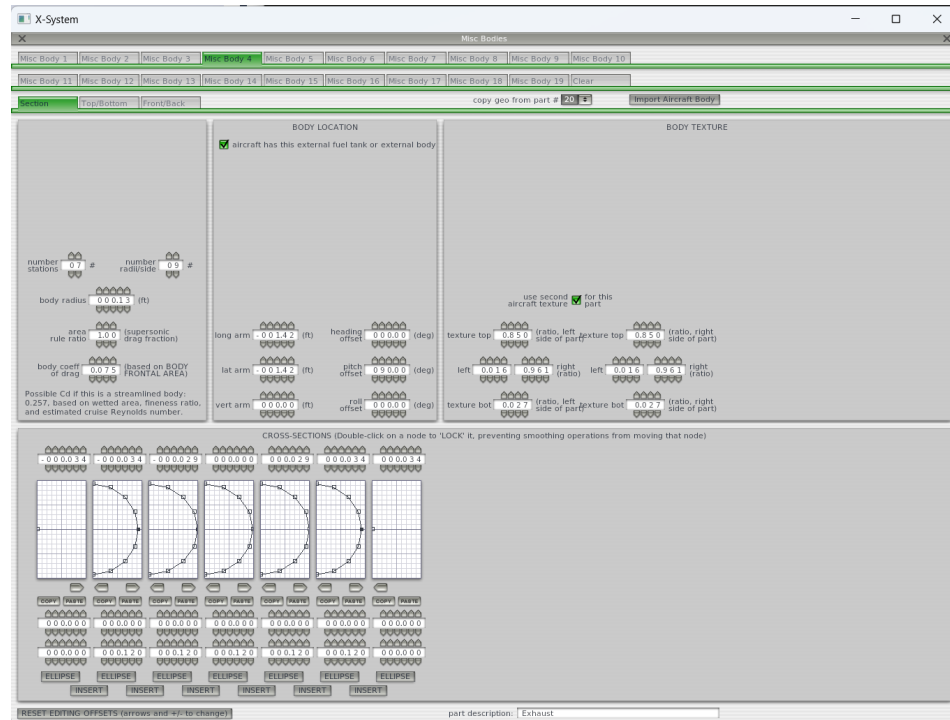


Figure 11. Drone design in PlaneMaker tool.

The X-Plane simulator enables the application of weather disturbances such as wind, rain, and fog. The same environmental conditions can be replicated in many simulations. The graphics of the simulator are also an advantage; the real world can be accurately visualised, which is a big plus. Figure 12 shows the mission visualisation in the X-Plane simulator. For this purpose, version 11.55 of X-Plane was used.



Figure 12. X-Plane simulator.

Software-In-The-Loop simulations consist of autopilot software, ground control station software, a dynamic model of the UAV, and an environment simulator. The simple diagram of SIL is presented in Figure 13.

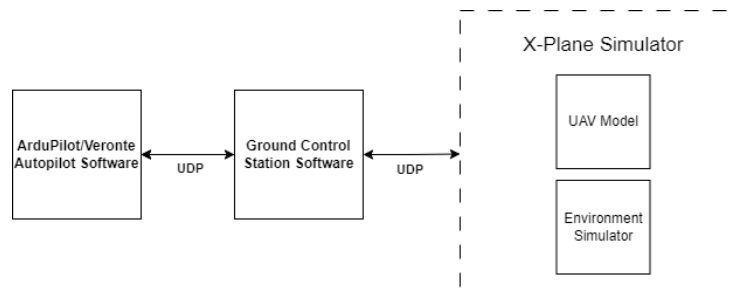


Figure 13. Software-In-The-Loop structure.

The communication between the components is conducted by using the UDP protocol. The frequency of the data coming from the X-Plane to the autopilot is set to the maximum possible value, equal to 100 Hz. The X-Plane sends data such as the acceleration of the UAV in the body frame, angular velocities, orientation, global coordinates, and position of the object. Those data are simulating the IMU, AHRS, and GPS measurements. The interface between the X-Plane and autopilots software was implemented in the Simulink Model. Figure 14 presents its structure; it consists of five blocks that exchange data between them. A block called 'XPlane Data receive' is responsible for receiving and parsing the data that the X-Plane sends during the simulation, and then those data go to the block called 'Environment Data', where the gravity acceleration and air density are computed, and 'Veronte PWM Receiver' and 'ArduPilot PWM Receiver', where the signals with states are sent to the autopilots software. The outputs from those two blocks are the PWM signals, which were calculated based on the simulation data. In the 'APIInterface block,' those signals are translated and transmitted to the X-Plane simulator, where they are directed to the motors of the octocopter model.

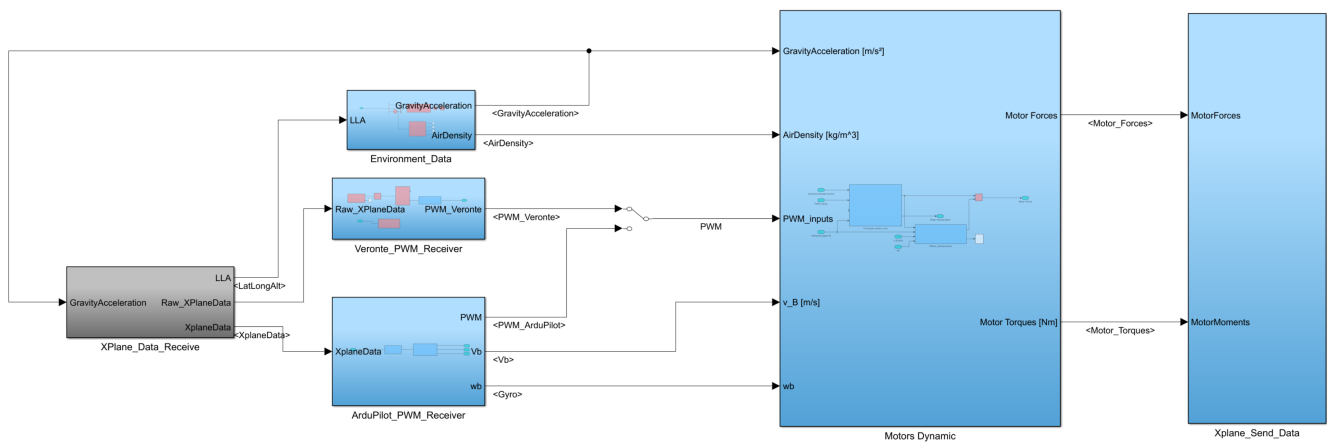
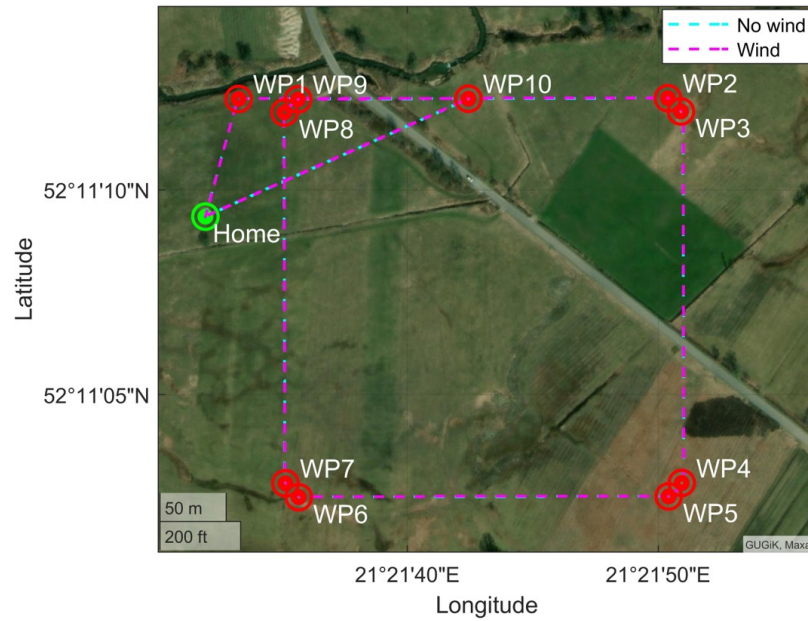


Figure 14. Software-In-The-Loop structure.

### 5. Results

The comparison of the autopilots was conducted in three distinct mission scenarios. The first mission is a flight along a predefined trajectory, which is presented in Figure 15. The starting point and other initial conditions are the same for each mission. Those trajectories were imported into dedicated ground control software to perform the predefined missions. In the second mission scenario, the UAV follows the same trajectory as in the first scenario, with the additional inclusion of a weather disturbance, which is a northward wind with a magnitude of 5 m/s. The third scenario is a rapid deceleration from 25 to 0 m/s.



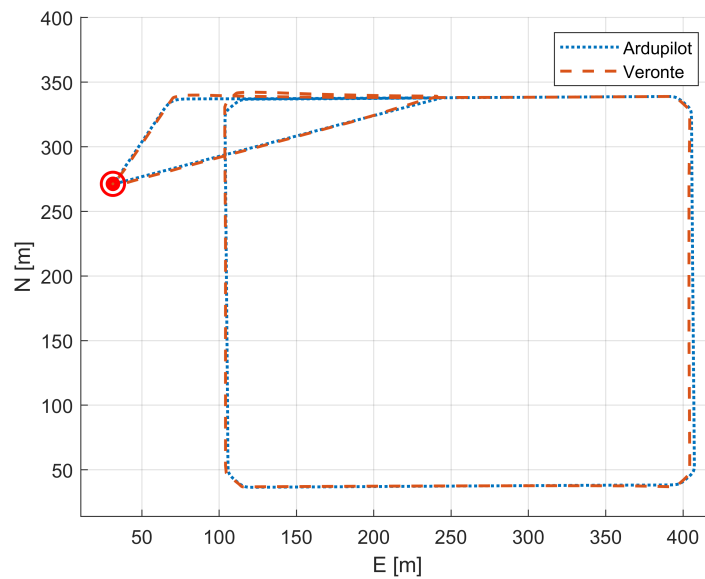


**Figure 15.** Predefined waypoint trajectory.

The results obtained from this research are presented in the graphs within the relevant subsections below.

*5.1. First Scenario Results*

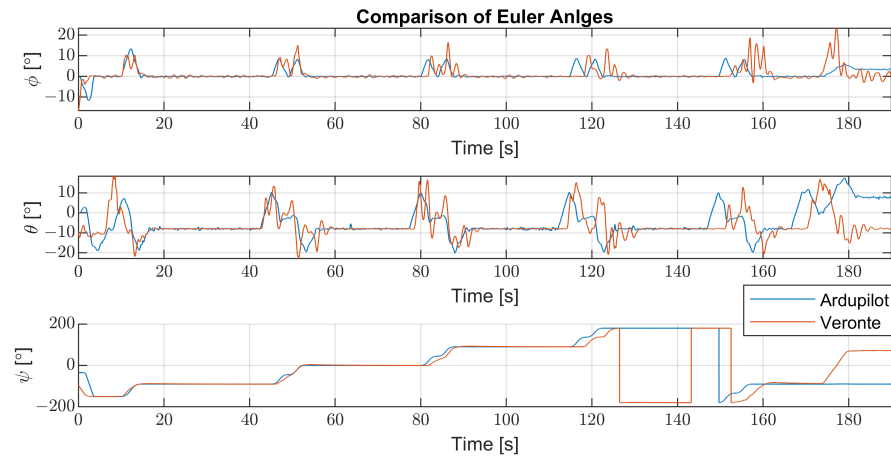
In this section, the comparison of two sets of autopilot flight data from the first scenario is presented. Data such as the Euler angles, accelerations, and flight trajectory will be compared. The trajectory resembles a square, with each side measuring 300 m; in total, there are 10 waypoints. Figure 16 shows the comparison of two autopilot trajectories in the NED coordinate system; the axis is marked N points north and E points east.



**Figure 16.** Comparison of trajectories.

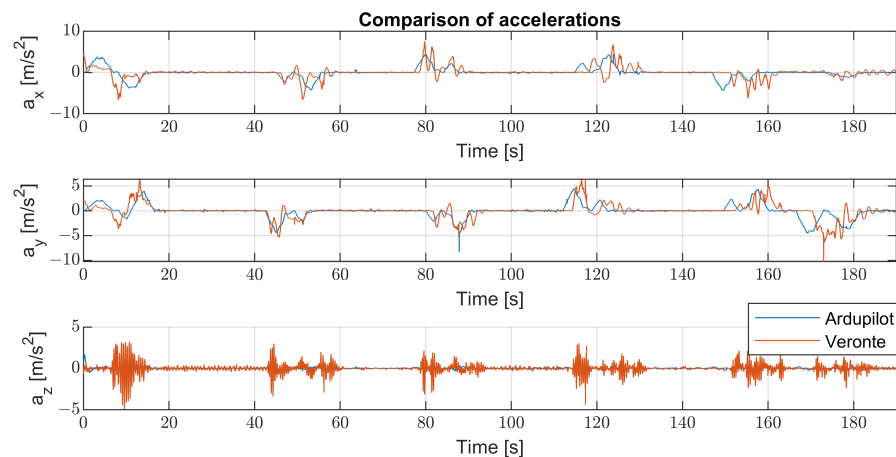
The trajectory in blue corresponds to the mission test conducted with Ardupilot, while the red colour corresponds to Veronte. Circular symbols on the graph indicate “home” positions for the first two mission scenarios.

Figure 17 presents the orientation of the simulated object during the missions, containing data obtained from both missions using the Veronte and Ardupilot autopilot. The figure contains graphs on which the change in the roll, pitch, and yaw angles during the mission is demonstrated. The data in blue represent the Ardupilot data, and the data in red correspond to the Veronte data.



**Figure 17.** Comparison of orientation angles.

Figure 18 presents the accelerations relative to the body frame of the model that were obtained during the missions. It is the dynamic acceleration due to motion or external forces (including the gravitational components). This focus on variations in acceleration offers critical insights for analyzing motion, estimating orientation, and supporting various dynamic applications. The set of three figures is demonstrated; on the first graph, the acceleration along the x-axis in the body frame is presented. The second figure shows the acceleration along the y-axis and the acceleration along the z-axis in the third figure. On each of them, the data derived from the simulations using Ardupilot and Veronte autopilots are compared.



**Figure 18.** Comparison of accelerations.

Figure 16 does not indicate any significant differences in the operation of both autopilots. The trajectories remain very close to each other. The Ardupilot autopilot faithfully reproduced the given trajectory. However, when performing missions with the Veronte autopilot, there were overshoots in turns, with errors not exceeding 3 m. The next two graphs show the difference in the mission completion time; Ardupilot completed the mission approximately 5 s faster than Veronte. This distinction becomes evident after the

third turn. The values of the orientation angles and accelerations do not show significant differences in the operation of the two autopilots.

### 5.2. Second Scenario Results

This section includes the results from the second scenario tests in which the wind with a constant velocity was added. The trajectory planned for this mission is the same as in scenario 1, where the magnitude of wind is constant, equal to 5 m/s and in a north-to-south direction. Figure 19 presents the comparison of two trajectories; the blue colour corresponds to the trajectory of the object with the Ardupilot autopilot, while the red colour presents the data obtained from the simulation where the Veronte autopilot was used.

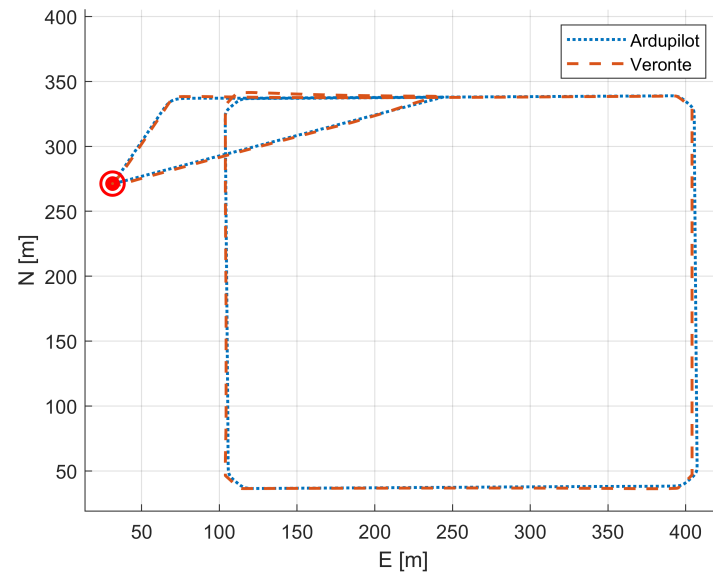


Figure 19. Comparison of flight trajectories.

Figure 20 presents the orientation of the object during the mission; data from the simulations where the Veronte and Ardupilot were used is demonstrated on the same figures. The changes in the roll, pitch, and yaw angles are presented. In both cases, the differences in the orientation angles compared to the first scenario are evident. It is caused by the appearance of a constant wind velocity that either counteracts or facilitates the drone's flight.

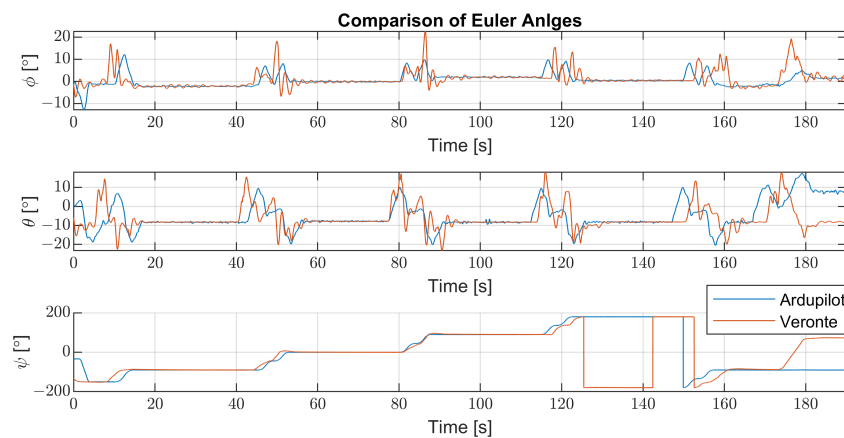
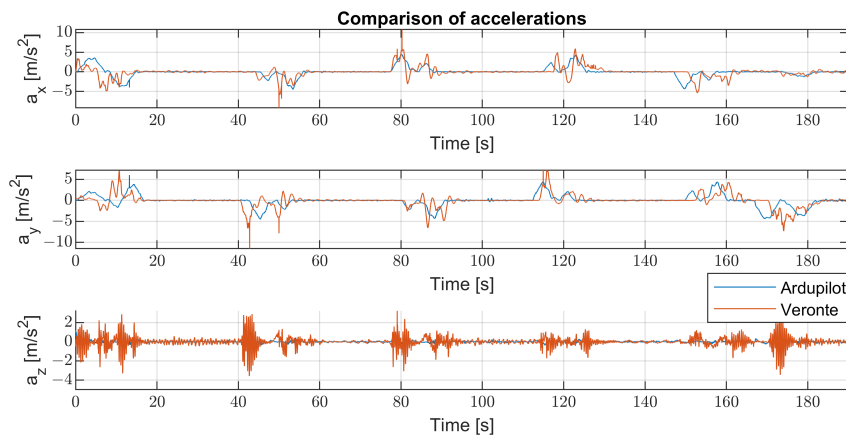


Figure 20. Comparison of orientation angles.

Similarly to Figure 18, the accelerations relative to the body frame of the drone during the missions are presented in Figure 21. The accelerations along the x-, y-, and z-axes in

the body frame are demonstrated, as previously demonstrated in two simulations and compared in the figures.

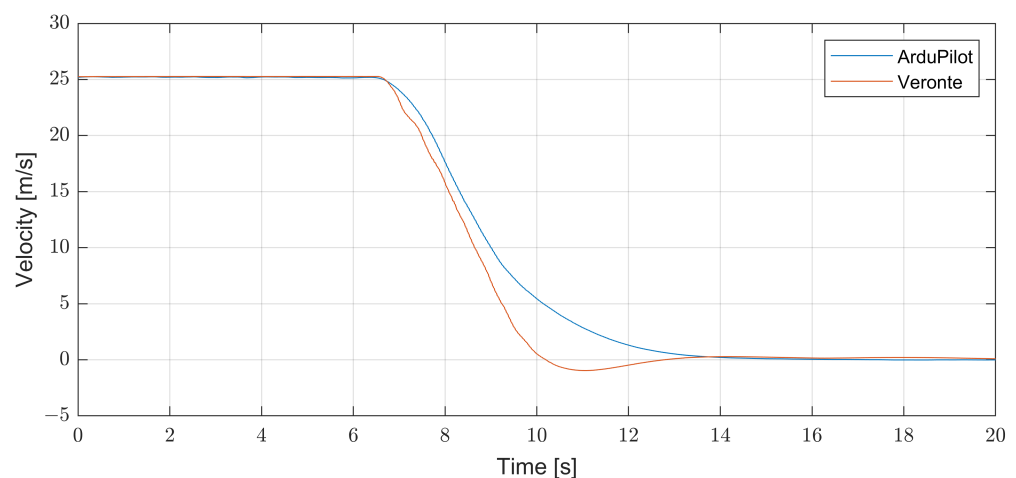


**Figure 21.** Comparison of accelerations.

Similarly to the previous scenario shown in Figure 19 with trajectories obtained during the missions, the current scenario does not show significant differences. During the Veronte autopilot mission, there were alternating overshoots, with deviations of no more than 4 m. The effect of constant winds did not significantly affect the performance of any autopilot during the mission. According to the above scenario, the Ardupilot autopilot completed the mission with a time advantage of approximately 5 s over Veronte. This consistent trend throughout the mission suggests a noticeable and repeatable difference in the performance of the two autopilots. The orientation angles and acceleration values provide evidence that the constant wind had no appreciable effect on mission performance.

### 5.3. Third Scenario Results

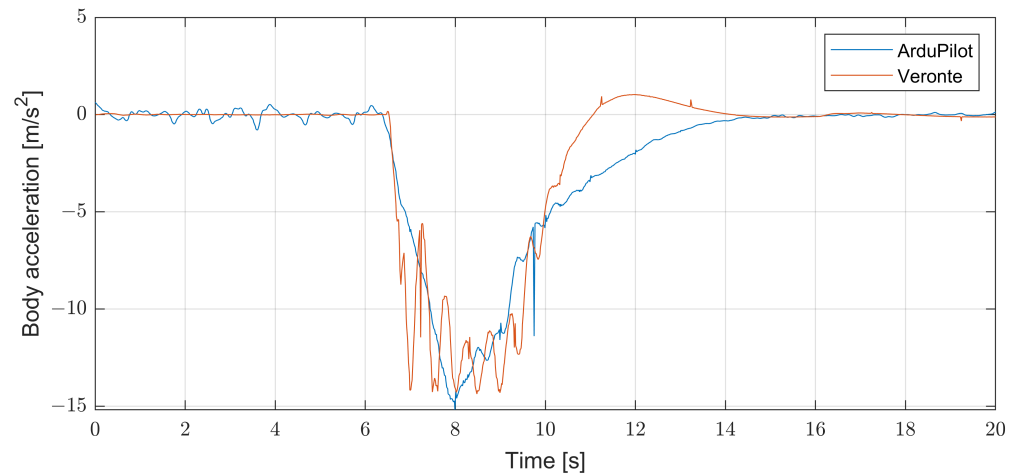
This subsection aims to present the data obtained from the final scenario, involving a rapid deceleration from 25 to 0 m/s. The flight data obtained from two simulations using two different autopilots are compared and visually represented on the graphs. Figure 22 shows the comparison of the change in velocity during the mission; in both cases, the deceleration was initiated just prior to the seventh s.



**Figure 22.** Comparison of velocities.

The data presented in blue correspond to the data obtained during the simulation, where the Ardupilot autopilot was tested, while the red colour represents data obtained during the Veronte autopilot testing. It can be observed from the graph that for both

autopilots, the response times were approximately 3 s. After that, the velocity of the object is constant. It can be seen that the object controlled by the Veronte autopilot reached 0 m/s more quickly but was experiencing overshooting. Figure 23 shows the change in acceleration during the missions of Veronte and Ardupilot.



**Figure 23.** Comparison of accelerations.

The colour scheme of the signals remains the same as before. The Veronte autopilot exhibited oscillations in its acceleration. For both cases, the maximum acceleration was about 15 [m/s<sup>2</sup>]. Overall, both autopilots performed the mission in a comparable way.

## 6. Conclusions

This section summarises the study of the comparison of the SIL simulations using commercial autopilots; the data obtained during the various simulation scenarios will be discussed, highlighting the differences between them. Additionally, an examination of the performance of the tested autopilots in distinct scenarios will be presented. The SIL simulations enabled a comprehensive assessment of each autopilot's ability to handle a variety of scenarios, from routine operations to emergency situations. Each of the autopilots show strengths and areas for improvement looking at all stages from planning the mission.

Starting with the mission-planning stage, both autopilots demonstrated user-friendly interfaces and ease of use. No problems were encountered while importing the waypoints to the ground control software. In the context of designing and tuning autopilot algorithms, the Ardupilot demonstrated much more convenient methods and tools. The main autopilot parameters have to be set by the user, such as the proportional gain of the orientation control loop and proportional, integral, and derivative gains for the angular velocity control loop, while in the Veronte autopilot, the whole structure of the autopilot has to be created, including the inner and outer loops of the control system. This complexity contributes to its increased reliability and the operator's awareness of what is happening within the control loops.

Moving on to the data obtained from the test scenarios, both of the autopilots successfully handled the three scenarios. The simulation data did not reveal significant differences in the performance of both autopilots. The trajectory flights aligned with the trajectory defined before the mission, and no significant overshootings or bias errors appeared. Any differences in the orientation angles or accelerations resulted from the autopilot algorithms, and the Ardupilot autopilot had a more complex structure, while the Veronte autopilot was based on closed-loops controllers in this research. The second scenario gave similar results, whereby both of the tested autopilots handled the presence of wind. The flight trajectory coincided with the desired one, and also the orientation angles and accelerations obtained from the simulations are similar. The third scenario showed the differences in the autopilots' performance during rapid deceleration from 25 to 0 m/s. In this comparison, the Veronte

autopilot had a better operational efficiency, the time in which the object achieved a velocity of 0 m/s was shorter than the time achieved by the Ardupilot autopilot, and the difference was equal to about 3 s.

In conclusion, choosing the Ardupilot autopilot is more beneficial for users looking for a simple and quick way to operate any object. This does not require an in-depth understanding of the design of the autopilot structure and also consumes significantly less time for preparation, tuning PID controllers, and executing flights on the respective object. There is a ready-to-use algorithm available for autotuning objects with the Ardupilot autopilot. On the other side, the autopilot Veronte is intended for users looking for a comprehensive understanding of the control algorithm for a specific object. It is also a preferable choice for users whose aim is to certify the autopilot and expect a high level of reliability from the autopilot system.

Overall, the SIL simulations provided a comprehensive platform to objectively evaluate and compare the performance of commercial autopilots. Future work may include more complex scenarios and extend the comparison to additional autopilot systems, providing a broader understanding of their capabilities and limitations in real-world applications. Furthermore, the comparison of specific autopilot systems in SIL and real flights could be conducted, and those results could show the differences between the model of the UAV and the real object and also evaluate the control algorithms on real hardware.

**Author Contributions:** Conceptualisation, M.W.; methodology, M.W.; software, M.W.; validation, M.W., N.S., and A.Z.; formal analysis, M.W.; investigation, M.W.; resources, M.W. and N.S.; data curation, M.W.; writing—original draft preparation, M.W.; writing—review and editing, N.S. and A.Z.; supervision, A.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

### Symbols and abbreviations

UAV	Unmanned Aerial Vehicle
COTS	Commercial Off-The-Shelf
SIL	Software-In-The-Loop
LLA	Latitude, longitude, altitude
NED	North East Down coordinate system
PID	Proportional–integral–derivative

### Latin symbols:

$m$	Drone mass, [kg]
$I_{xx}, I_{yy}, I_{zz}$	Full mass moments of inertia with respect to body coordinate system $O_b x_b y_b z_b$ , [kg·m <sup>2</sup> ]
$p$	Roll rate, [°/s]
$q$	Pitch rate, [°/s]
$r$	Yaw rate, [°/s]

### Greek symbols:

$\theta$	Euler pitch angle, [°]
$\phi$	Euler roll angle, [°]
$\psi$	Euler yaw angle, [°]

## References

1. Ayamga, M.; Akaba, S.; Nyaaba, A. Multifaceted applicability of drones: A review. *Technol. Forecast. Soc. Chang.* **2021**, *67*, 120677. [\[CrossRef\]](#)
2. Chao, H.; Cao, Y.; Chen, Y. Autopilots for small unmanned aerial vehicles: A survey. *Int. J. Control Autom. Syst.* **2010**, *8*, 36–44. [\[CrossRef\]](#)
3. Chao, H.; Cao, Y.; Chen, Y. Autopilots for small fixed-wing unmanned air vehicles: A survey. In Proceedings of the 2007 International Conference on Mechatronics and Automation, Harbin, China, 5–9 August 2007; pp. 3144–3149.



4. Bhatia, A.; Graziano, M.; Karaman, S.; Naldi, R.; Frazzoli, E. Dubins trajectory tracking using commercial off-the-shelf autopilots. In Proceedings of the AIAA Guidance, Navigation And Control Conference And Exhibit, Honolulu, HI, USA, 18–21 August 2008; p. 6300.
5. Kaminer, I.; Pascoal, A.; Xargay, E.; Hovakimyan, N.; Cao, C.; Dobrokhodov, V. Path following for small unmanned aerial vehicles using L1 adaptive augmentation of commercial autopilots. *J. Guid. Control Dyn.* **2010**, *33*, 550–564. [[CrossRef](#)]
6. Ambroziak, L.; Gosiewski, Z. Preliminary uav autopilot integration and in-flight testing. *Solid State Phenom.* **2013**, *198*, 232–237. [[CrossRef](#)]
7. Krawczyk, M.; Zajdel, A.; Szczepański, C. Simulation and Testing of Flight Stabilisation System Using Trimmers. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2021; pp. 185–196. [[CrossRef](#)]
8. Zajdel, A.; Krawczyk, M.; Szczepański, C. Pre-Flight Test Verification of Automatic Stabilization System Using Aircraft Trimming Surfaces. *Aerospace* **2022**, *9*, 111. [[CrossRef](#)]
9. Zielińska, T.; Pogorzelski, T. Samolokalizacja bezałogowego statku powietrznego uwzględniająca zmienną orientację kamery. *Pomiary Autom. Robot.* **2022**, *26*, 53–59. [[CrossRef](#)]
10. Kim, S.; Park, J. Path Planning with Multiple UAVs Considering the Sensing Range and Improved K-Means Clustering in WSNs. *Aerospace* **2023**, *10*, 939. [[CrossRef](#)]
11. Aydemir, H.; Zengin, U. Real-time Simulation Infrastructure for Model-based Design of Helicopter Flight Control System. In Proceedings of the 2018 AIAA Modeling And Simulation Technologies Conference, Reston, VA, USA, 26 June 2018; p. 0124.
12. Horri, N.; Pietraszko, M. A Tutorial and Review on Flight Control Co-Simulation Using Matlab/Simulink and Flight Simulators. *Automation* **2022**, *3*, 486–510. [[CrossRef](#)]
13. Coombes, M.; McAree, O.; Chen, W.; Render, P. Development of an autopilot system for rapid prototyping of high level control algorithms. In Proceedings of the 2012 UKACC International Conference On Control, Cardiff, UK, 3–5 September 2012; pp. 292–297.
14. Martin-Lammerding, D.; Astrain, J.; Córdoba, A. A multi-UAS simulator for high density air traffic scenarios. In Proceedings of the VEHICULAR 2022: The Eleventh International Conference on Advances in Vehicular Systems, Technologies and Applications, Venice, Italy, 22–26 May 2022.
15. Sun, L.; Beard, R.; Pack, D. Trajectory-tracking control law design for unmanned aerial vehicles with an autopilot in the loop. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 1390–1395.
16. Liu, C.; McAree, O.; Chen, W. Path following for small UAVs in the presence of wind disturbance. In Proceedings of the 2012 UKACC International Conference on Control, Cardiff, UK, 3–5 September 2012; pp. 613–618.
17. Hakiki, M.; Liu, H.; Lee, J. Development of autopilot control algorithm for an unmanned aerial vehicle based on simulation. *Int. J. Veh. Auton. Syst.* **2022**, *16*, 222–245. [[CrossRef](#)]
18. Çetin, E. *System Identification and Control of a Fixed Wing Aircraft by Using Flight Data Obtained From x-Plane Flight Simulator*; Middle East Technical University: Ankara, Turkey, 2018.
19. Bole, B.; Teubert, C.; Quach, C.; Hogge, E.; Vazquez, S.; Goebel, K.; Vachtsevanos, G. SIL/HIL replication of electric aircraft powertrain dynamics and inner-loop control for V&V of system health management routines. In Proceedings of the Annual Conference Of The PHM Society, New Orleans, LA, USA, 14–17 October 2013; Volume 5.
20. Sun, J.; Li, B.; Wen, C.; Chen, C. Design and implementation of a real-time hardware-in-the-loop testing platform for a dual-rotor tail-sitter unmanned aerial vehicle. *Mechatronics* **2018**, *56*, 1–15. [[CrossRef](#)]
21. Figueiredo, H.; Bittar, A.; Saotome, O. Platform for quadrotors: Analysis and applications. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 848–856.
22. Welcer, M.; Szczepański, C.; Krawczyk, M. The Impact of Sensor Errors on Flight Stability. *Aerospace* **2022**, *9*, 169. [[CrossRef](#)]
23. Gołabek, M.; Welcer, M.; Szczepanski, C.; Krawczyk, M.; Zajdel, A.; Borodacz, K. Quaternion Attitude Control System of Highly Maneuverable Aircraft. *Electronics* **2022**, *11*, 3775. [[CrossRef](#)]
24. Ardupilot Octo Quad X8. 2023. Available online: <https://ardupilot.org/copter/docs/connect-escs-and-motors.html> (accessed on 15 October 2023).
25. Wikipedia How X-Plane Works. 2023. Available online: [https://en.wikipedia.org/wiki/X-Plane\(simulator\)](https://en.wikipedia.org/wiki/X-Plane(simulator)) (accessed on 15 October 2023).
26. Ardupilot Ardupilot. 2023. Available online: <https://ardupilot.org/> (accessed on 15 October 2023).
27. Veronte Veronte Auopilot. 2023. Available online: <https://www.embention.com/product/drone-flight-simulator-autopilot-sil> (accessed on 15 October 2023).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.