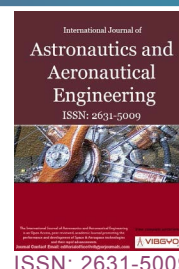




HIL Flight Simulator for VTOL-UAV Pilot Training Using X-Plane



D Al'aez^{1*}, X Olaz¹, M Prieto¹, P Porcellinis² and J Villadangos¹

¹Mathematical Engineering and Computer Science Department, Universidad Pu'blica de Navarra, Spain

²FuVeX Civil SI, Poligono Industrial Municipal (vial k), Spain

Abstract

With the increasing popularity of Vertical Take-off and Landing Unmanned Aerial Vehicles (VTOL UAVs) a new problem arises: Pilot training. Most conventional pilot training simulators are designed for full scale aircrafts, while most UAV simulators are just focused on conceptual testing and design validation. The X-Plane flight simulator has been extended to include new functionalities such as complex wind dynamics, ground effect and accurate real-time weather. A commercial HIL flight controller has been coupled with a VTOL UAV model to provide realistic flight control. The result is a complete simulation environment that has been successfully tested for pilot training of the MARVIN aircraft manufactured by FuVeX.

Keywords

HIL, Flight simulator, Pilot, Commercial, UAV, X-plane, Wind

Introduction

An Unmanned Aerial Vehicle (UAV) is often referred to a remotely piloted aircraft or an unmanned aircraft that can be controlled with or without a human operator. Rotary wing UAVs such as quadcopters and hexacopters are widely used due to their low costs, simplicity and hovering flight capabilities. The main downside of these vehicles is that their range is very limited, because all the lift required to overcome the UAV weight must be provided by the motors. Conventional fixed wing UAVs allow for longer operations but require a runway for take-off and landing. Vertical take-off and landing (VTOL) UAVs combine the advantages of both aircraft types, allowing for vertical take-off, landing and hovering capabilities with extended flight ranges. The increasing popularity of this type of UAVs has given rise to a new problem: UAV pilot training.

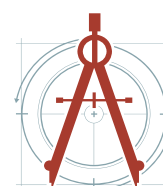
VTOL UAVs require very specific piloting skills, since pilots need to be able to control the vehicle both as an airplane and as a quadcopter. Traditional radio-controlled aircraft pilots often find it difficult to transition from airplane piloting to quadcopter piloting. The reverse is even more challenging. Conventional pilot training has traditionally been conducted in flight simulators, but transferring this technology to the training of VTOL UAV pilots in a simulator poses a series of challenges. For example, VTOL UAVs are

***Corresponding author:** Daniel Al'aez Gómez, Research Assistant, Mathematical Engineering and Computer Science Department, Universidad Publica de Navarra, Pamplona, 31006, Spain

Accepted: April 13, 2023; **Published:** April 15, 2023

Copyright: © 2023 Al'aez D, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Al'aez et al. *Int J Astronaut Aeronautical Eng* 2023, 8:064



Citation: Al'aez D, Olaz X, Prieto M, Porcellinis P, Villadangos J (2023) HIL Flight Simulator for VTOL-UAV Pilot Training Using X-Plane. *Int J Astronaut Aeronautical Eng* 8:064

extremely sensitive to wind disturbances during take-off, landing and hovering phases. Throughout this paper we will demonstrate the progress made to develop a flight simulator capable of being used to train VTOL UAV pilots in different scenarios.

Section 2 contains a brief introduction to the state of the art in UAV flight simulation. Section 3 summarizes our contributions in this field. Section 4 is devoted to modeling the simulator for our specific aircraft: The MARVIN VTOL UAV. The integration with X-Plane is explained in Section 5, and additional plugins such as live weather and ground effect are detailed in section 6. Section 7 gathers the results of the research and the operation of the fully-working simulator. Finally, conclusions are summarized in Section 8.

Related Works

The increasing popularity of electric VTOL UAVs due to their unique take-off and landing capabilities, coupled with longer ranges than rotary wing UAVs, has promoted numerous studies on the simulation of this type of aircraft over the last 20 years [1-5].

The typical simulation choice for computation is MATLAB [6,7], which offers full freedom to define the dynamics of the UAV. However, the complexity of having to derive the whole flight dynamics often incurs in over- simplifications such as only considering front-wind aerodynamics [8], or very basic PID flight controllers [9]. Some more recent works relegate the calculation of the aircraft dynamics to a flight simulator such as Real Flight [10], or X-Plane [11,12].

Using a flight simulator does not only offer the advantage of leveraging the flight dynamics computation to the internal physics engine of the simulator, but also provides a more complex world model that otherwise would require too many hours of development: Live-weather, 3D scenery and elevation maps, day-night cycles, advanced visualization features, etc.

On the other hand, most of the cited works are focused on simulating a VTOL UAV for performance analysis and conceptual design, but few of them focus on pilot training or even consider the input of a pilot [13].

Flight simulators are a very stablished tool [14] for commercial and military pilots training [15,16]. When appropriately designed, it has been demonstrated that simulation can provide competence development with pedagogical and economic advantages [17].

The increasing trend on unmanned aerial platforms, and its consequent increase in the number of pilots, has emphasized the need of pilot training to prevent accidents and reduce operating costs. A survey on commercial and open source unmanned vehicle simulators [18] concluded that it is no longer necessary to build a new simulator from scratch. Newer and/or specialized engines such as the flight simulator X-Plane or Ageia PhysX and Havok physics engines are perfectly capable of simulating complex physical interactions.

Contributions

We have developed a full-featured HIL solution of a commercial UAV- VTOL (MARVIN aircraft, manufactured by FuVeX) using X-Plane, a modified physics engine considers wind coming from any direction, accurate real- time data, historic data and weather forecasts, ground effect, and failure simulation for pilot training.

UAV-VTOL Modeling

Initially, a large amount of aircraft data has to be collected before proceeding to the model design in X-Plane. Firstly, the minimum model parts that must be present in the model should be specified. This particular UAV- VTOL is composed of a fixed wing and four motors in a quadcopter arrangement with a tilt mechanism. In quadcopter mode, all four engines are facing upwards. When transitioning to plane mode, the four arms are tilted down until all four motors point forward. The main components to be considered are:

- Wing and endplates.
- Fuselage (considering the airfoil shape).
- Engines and propellers.
- Connecting engine arms (carbon fiber rods) and tilt mechanism.
- Tail: Vertical tail plane (VTP), horizontal tail plane (HTP) and the connecting carbon fiber rod.
- Basic landing gear.

Figure 1 shows a 3D render of the UAV-VTOL both in copter and plane mode, featuring all the minimum model parts for reference, except for the landing gear.

Mass, inertia, and geometric data

The mass and inertia properties have been collected via experimental weighting of components and CAD modules. The essential values to collect for the model are: Total weight, wingspan, length, center of gravity position (x_{CG}, y_{CG}, z_{CG}), tilt center position (x_{tilt}, z_{tilt}), and inertia values ($I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$). Note that the y_{tilt} component is neglected assuming a reference system in which the Y axis denotes the distance to the midplane of the aircraft.

The relative position of each engine has to be specified as well to match the flight controller setup. In this aircraft, the rear engine arms are displaced vertically 30 mm up from the mid-plane, while front engine arms are displaced 30 mm down from the mid-plane. The mid-plane offset occurs at the joint of the tilt mechanism, and connecting carbon rods are drawn as hollow rods of a certain diameter.

Other measurements such as the tail rod length and ground clearance or landing gear height must also be provided.

Aerodynamic specifications

To properly reconstruct the wing and fuselage, many parameters have to be provided for each wing:

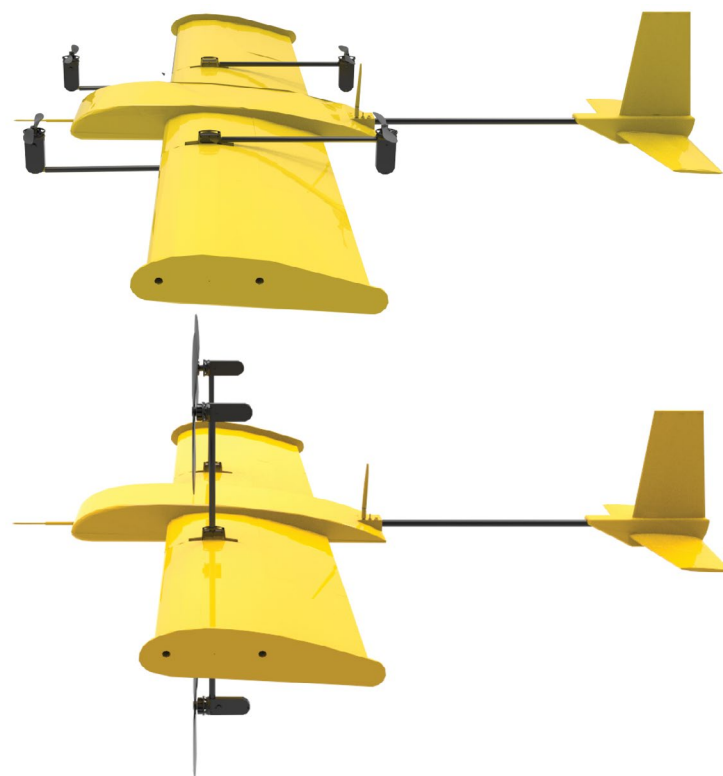


Figure 1: 3D renders of the VTOL UAV both in *copter* mode (top) and plane (bottom) mode.

Chord (root and tip chord values if the wing is tapered), sweep angle, dihedral angle, wingspan, thickness, aerodynamic center (x_{AC} , y_{AC} , z_{AC}), position relative to the fuselage, and airfoil shape.

The 2D airfoil aerodynamic coefficient plots are indispensable to match the aerodynamic performance, as it is explained in Section 5. They can be obtained from CFD simulations, potential flow solvers such as *X-Foil* [19], or airfoil databases such as Airfoil Tools.

Although it is not essential at this stage, it may also be of interest to collect aerodynamic data of the entire aircraft such as the coefficient of lift, drag or moment for different angles of attack for model validation and tweaking. Anyway, these values will be essential for implementing the wind plugin in Section 6.

Control surfaces

Only two control surfaces are available in this aircraft: elevator and rudder. Attitude changes use the help of differential thrust/mixing between the four motors. Both the elevator and the rudder are actuated by independent servos, with a maximum and minimum deflection of (+30 degrees). In this case, the entire vertical tail plane (VTP) and horizontal tail planes (HTP) act as these control surfaces.

Engines and propellers

The last modeling step is to gather motor and propeller data. The motor characteristics and dimensions are collected from manufacturer datasheets. For the propeller, generally benchmark tests need to be performed. Using a motor test stand, we have measured speed (RPM), thrust (g), current (A) and voltage (V) at different pulse widths of throttle stick percentage. When there is no propeller airfoil data available, or 3D scanning of the propeller is difficult or expensive to perform, the propeller can be recreated by matching these benchmark tests.

Operational specifications

For validation purposes and flight controller setup, some operational specifications are useful:

- Typical cruise speed (m/s).
- Stall speed (m/s).
- Transition speed from copter to plane mode (m/s).
- Maximum speed (m/s).
- Minimum turn radius (m) or standard rate of turn (degrees per second).

Once all the above-mentioned data has been collected, the aircraft model is ready to be integrated into the flight simulator.

Commercial Flight Simulator Integration

Pilot training requires both the vehicle model and the world model to perform realistically in every possible scenario that may take place during actual flight. Three of the most commercially used flight simulators are Lockheed Martin's Prepar3D, RedBird Flight Simulator and X-Plane 11.

Other simulators such as Microsoft Flight Simulator offer better graphics but with a much more degraded physics engine. X-Plane 11 has been selected over all these options due to the plugin software development kit (SDK) and a very active community. Also, it features a professional version with hardware and frame-rate checks that makes the simulator eligible for FAA- Certification.

To incorporate a new UAV aircraft model into X-Plane, the first step is to define the custom airfoil properties in Airfoil Maker (Figure 2), based on the specifications defined in Section 4. The curves described by the aerodynamic coefficients of the 2D airfoils must be matched in Airfoil Maker.

To understand the importance of matching these curves, we must first get an insight of X-Plane's internal physics engine. Aerodynamic forces and moments are calculated following a panel method similar

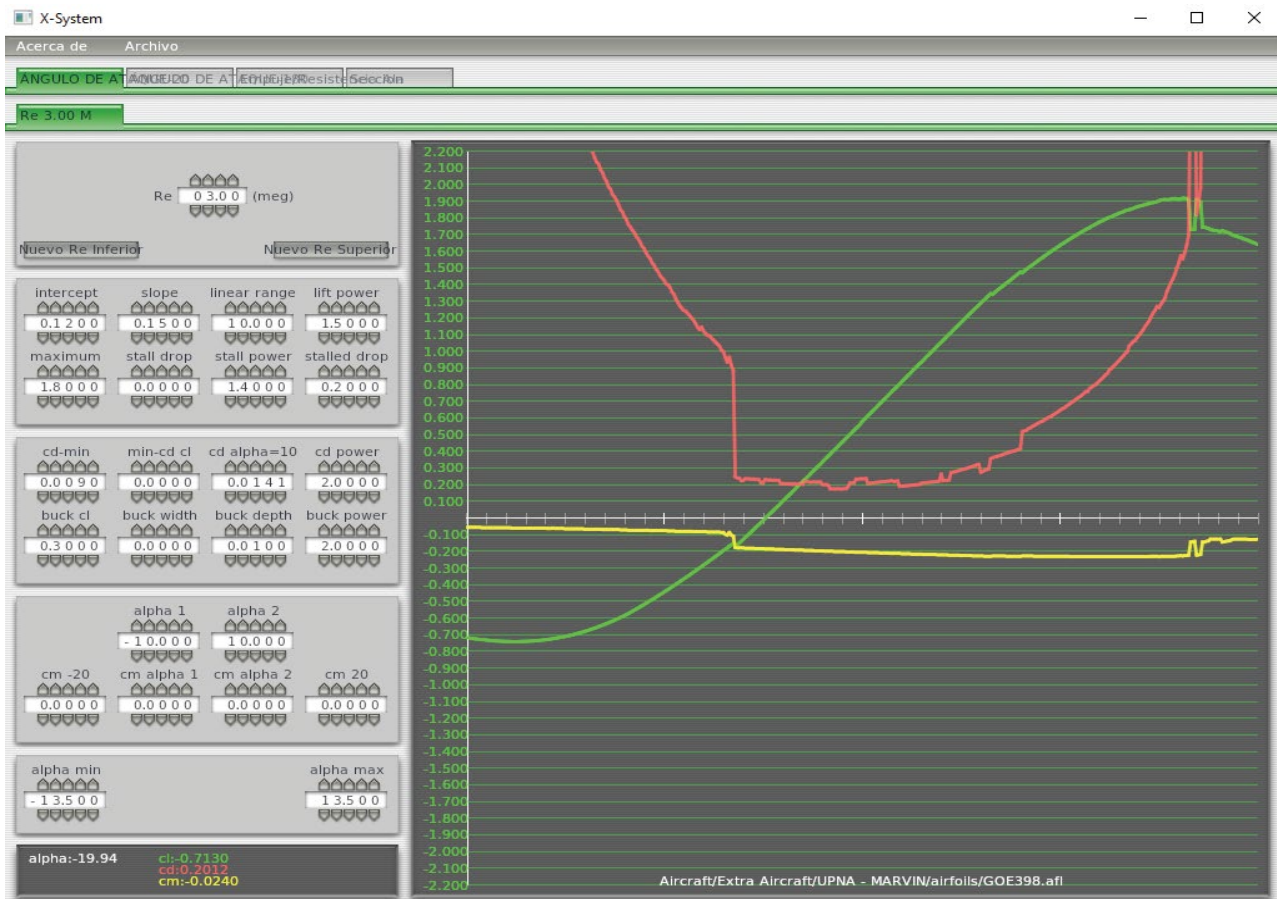


Figure 2: X-Plane's Air foil Maker screenshot featuring manually-adjusted aerodynamic coefficient curves of an air foil.

to the blade element theory, which involves breaking the aircraft into small elements and finding the force on each element at every cycle. Compressible flow effects are considered using the Prandtl-Glauert correction in subsonic flight [20]:

$$C_p = \frac{C_{p,0}}{\sqrt{1-M^2}}, \quad (1)$$

where C_p is a coefficient of pressure and M is the Mach number. Once the coefficients and areas of each element have been determined, they can be multiplied by the dynamic pressure over each panel to obtain the forces. The sum of forces over the entire aircraft is divided by the aircraft mass for linear accelerations, and moments of inertia for angular accelerations. This is a much more precise real-time approximation of how the aircraft flies compared to other simulator models such as *stability derivatives* [21].

Back to the aircraft modeling, the next step is to create a 3-dimensional model of the aircraft's main features, including: Wings, tail (HTP & VTP), control surfaces, fuselage, carbon fiber rods, motors & propellers and landing gear. All the aircraft data from Section 4 is used as an input to recreate the model, as it can be seen in Figure 3.

Additional flight testing is required to tweak the model in order to match the real flight performance and maneuverability. Once the model is ready, it can be loaded into X-Plane to start experimenting.

HIL flight controller integration

Rather than testing the flight control algorithm on a purely mathematical model of the system, real hardware can be used to run the control loop, connected to a simulation environment. The testing and the evaluation of the system are carried out in real-time, ensuring that the embedded control system can

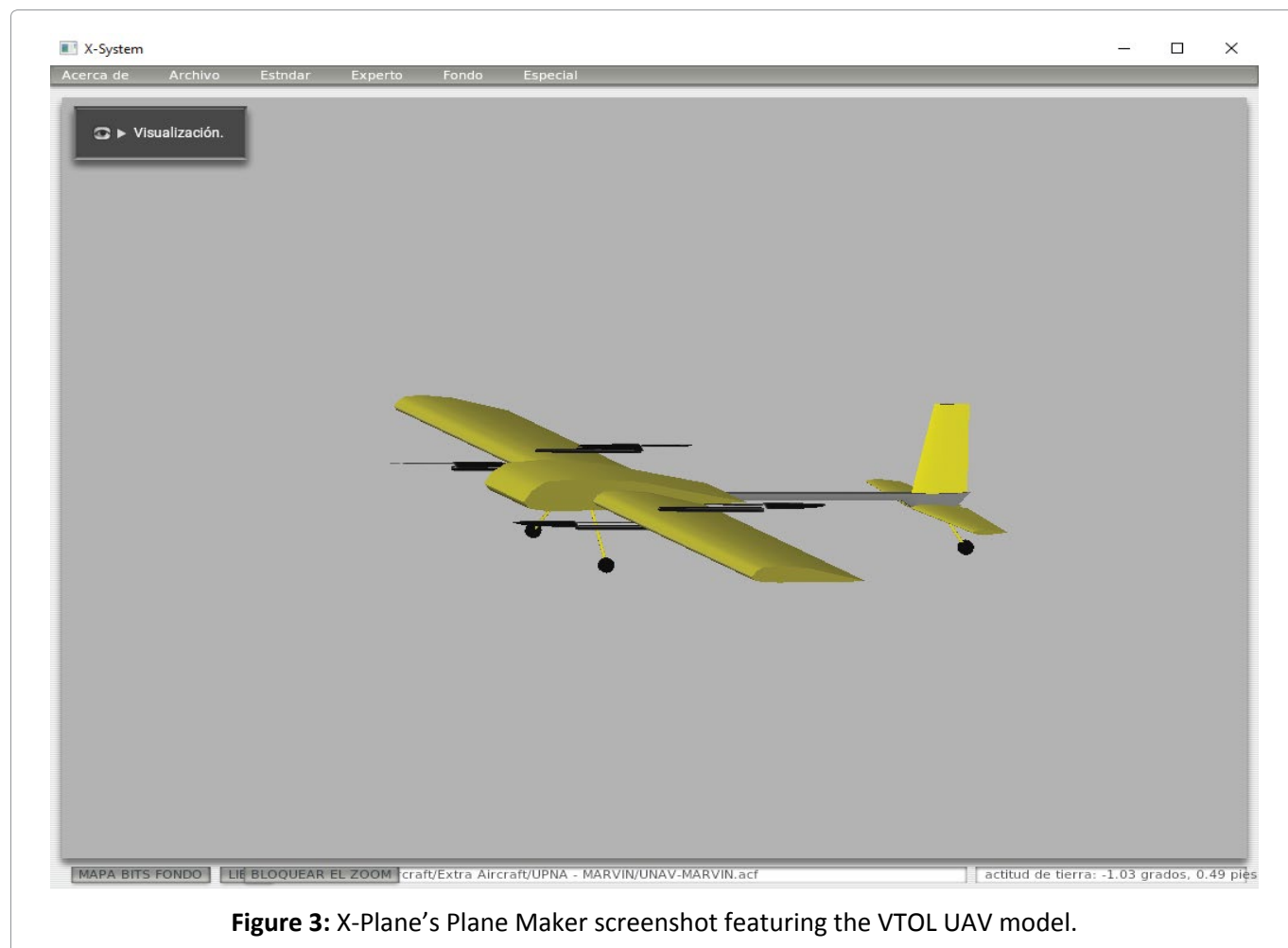


Figure 3: X-Plane's Plane Maker screenshot featuring the VTOL UAV model.

deliver the control input within the desired sampled period. One of the earliest uses of HIL simulation was for flight simulation [22], with noticeable contributions in the guided missile development and testing field [23-25].

The Veronte Autopilot [26] control system is integrated with the X-Plane simulator for highly realistic Hardware-in-the-loop simulations within a safe environment, prior to conducting real flight operations. Incorporating an actual physical flight controller into the pilot training creates an even more realistic experience. Different setups, tuning parameters and mission configurations can be tested on a real flight controller whose sensor inputs are generated by X-Plane. The Veronte Autopilot is a plug-and-play solution that enables fast and easy connection with X-Plane to start running the simulator. The same VTOL flight settings run during actual flight can be loaded for simulated flight.

The Veronte Autopilot is a fully autonomous flight controller, so it capable of fully operating even without a transmitter. The transmitter has to be connected as an external joystick. To use the joystick in the system, the pulse position modulation (PPM) output of the trainer has to be connected to a digital input of Veronte and configured as the radio input in the Veronte ground control software. After setup, the transmitter can be used as a joystick to control the aircraft both in simulation and real flight.

The ground control station software works exactly the same for both HIL and real flight scenarios, making it suitable for pilot training, mission setup, and in-flight control.

Live Weather Conditions, Wind and Ground Effect Plugins

The most critical stages in the VTOL operation are the landing and take-off phases, where the pilot input is typically required to guarantee a safe operation. Furthermore, it's in these stages when the VTOL is most affected by windshear, bursts of wind and other phenomena. Although this is a common aspect with any other type of UAVs, tilt-rotor VTOLs are especially affected by wind.

During stationary flight (take-off, landing and hover) phases, the main speed component acting on the aircraft is the windspeed. This, combined with a large wind surface area, generates large forces and moments that need to be compensated by the pilot. Although X-Plane features some options to internally compute winds and gather real-time data from weather stations (METAR), it is only able to estimate winds based on front-wind aerodynamic data, and extrapolate it to other incoming directions with a slip angle. This approximation is valid enough for aircrafts flying at relatively high forward speeds, but not for VTOLs in stages where there is no forward speed component: take-off, landing and hovering flight.

We have implemented a custom physics plugin to override the internal wind physics engine and consider wind incoming from any possible direction.

Wind forces and moments calculation

Calculating the forces and moments originated by wind coming from any direction requires aerodynamic coefficients to be pre-computed for all these possible scenarios. But mapping all the possible combinations of roll (ϕ), pitch (ϑ), yaw (ψ) angles, and simulating their respective aerodynamic coefficients of lift, drag, and lateral force, pitching, rolling and yawing moment ($C_{L'}, C_{D'}, C_{Q'}, C_{M'}, C_{LA'}, C_{NA}$) in a wind reference frame is unmanageable. Therefore, a pseudo-random set of 100 roll, pitch and yaw combinations has been selected. Aerodynamic coefficients estimated through simulation will be used to obtain an aerodynamic model of the MARVIN aircraft.

Once these coefficients have been determined using CFD RANS simulations in Ansys Fluent, a simple linear regression model is constructed for quick interpolation by the plugin. The forces and moments in wind reference frame can be calculated from:

$$L^w = \frac{1}{2} \rho \left| \vec{V}_{perceived} \right|^2 SC_L(\phi, \theta, \psi), \quad (2)$$

$$D^w = \frac{1}{2} \rho \left| \vec{V}_{perceived} \right|^2 SC_D(\phi, \theta, \psi), \quad (3)$$

$$Q^w = \frac{1}{2} \rho \left| \vec{V}_{perceived} \right|^2 SC_Q(\phi, \theta, \psi), \quad (4)$$

$$M_A^w = \frac{1}{2} \rho \left| \vec{V}_{perceived} \right|^2 ScC_M(\phi, \theta, \psi), \quad (5)$$

$$L_A^w = \frac{1}{2} \rho \left| \vec{V}_{perceived} \right|^2 ScC_{LA}(\phi, \theta, \psi), \quad (6)$$

$$N_A^w = \frac{1}{2} \rho \left| \vec{V}_{perceived} \right|^2 ScC_{NA}(\phi, \theta, \psi), \quad (7)$$

where lift (L^w), drag (D^w), lateral force (Q^w), pitching moment (M_A^w), rolling moment (L_A^w) and yawing moment (N_A^w) are determined as a function of density (ρ), surface area (S), reference chord length (c), and the modulus of the windspeed perceived by the aircraft $V_{perceived}$.

With the help of a rotation matrix, we can express the perceived wind frame forces in a body-fixed frame:

$$\begin{pmatrix} F_{Ax} \\ F_{Ay} \\ F_{Az} \end{pmatrix} = L_{bw} \begin{pmatrix} -D^w \\ -Q^w \\ -L^w \end{pmatrix}, \quad (8)$$

where L_{bw} is the rotation matrix between perceived wind axes to body-fixed axes, and F_{Ai} is the resulting aerodynamic force to be applied to body axis i . The same applies for the aerodynamic moments:

$$\begin{pmatrix} L_A \\ M_A \\ N_A \end{pmatrix} = L_{bw} \begin{pmatrix} L_A^w \\ M_A^w \\ N_A^w \end{pmatrix} \quad (9)$$

The perceived wind axes are defined as the result of combining a rotation of the aircraft in Ansys to align the wind vector with the z axis in Ansys, and a rotation to recreate the aircraft attitude (roll, pitch and yaw angles) in Ansys from the *OpenGL coordinates in X-Plane*:

- The x axis is aligned east-west with positive X pointing east.
- The y axis is aligned straight up from the ground plane.
- The z axis is aligned north-south with positive Z pointing south.

The *body-fixed frame* is defined as usual in aeronautics:

- Positive x axis points to the nose of the aircraft.
- Positive y axis points to the right wing.
- Positive z axis points to the bottom perpendicularly to the x and y axes.

The default Ansys axes are displayed in Figure 4. When the roll, pitch and yaw values are set to zero, before considering the wind, the nose of the aircraft is aligned with the z axis (blue). The x axis (red) points to the left

wing, and the y axis points up (green).

This formulation has been implemented in a Python Flask [27] server that can be run externally and connected to our X-Plane's custom developed wind plugin via CURL requests.

Live weather conditions

To extend X-Plane's METAR stations coverage, we have resorted to using an external plugin called Active Sky XP (ASXP) from HiFi Simulation Technologies [28], featuring more than 9000 weather stations sharing real-time weather data. Additionally to live-weather, this plugin offers historic data and weather forecasts, very useful when planning flights or revisiting a past scenario while training. Initially, the

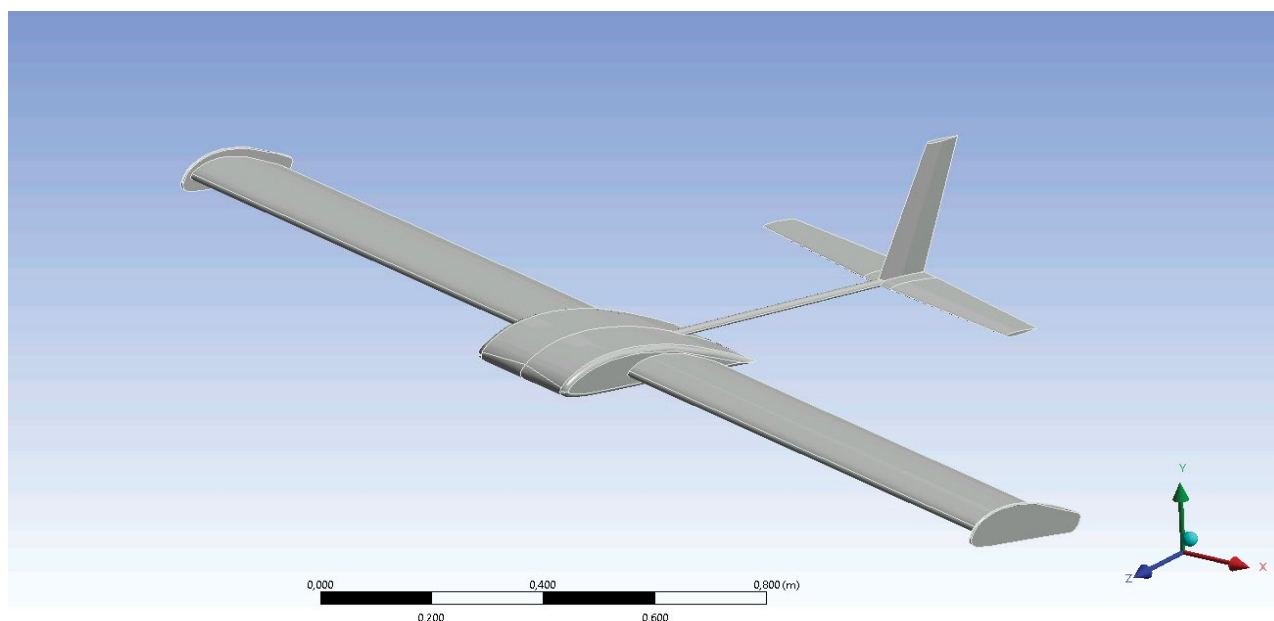
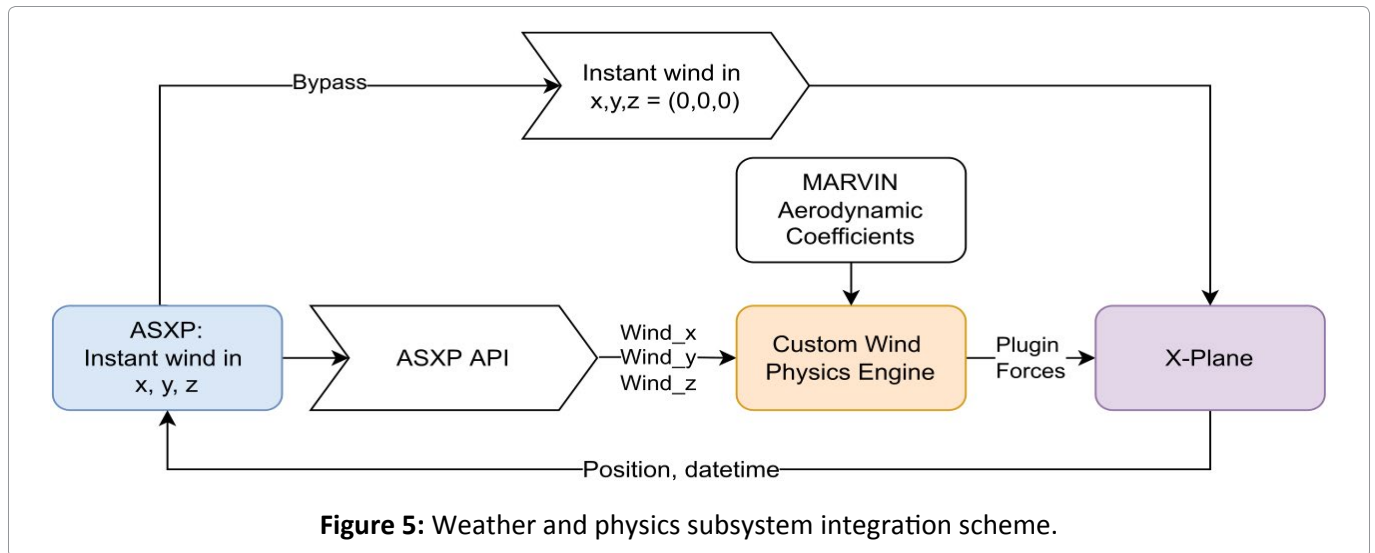


Figure 4: Meshing tool screenshot from Ansys featuring the simplified 3D model and Ansys default axes.



wind conditions were directly fed to X-Plane, so there was no way to disable the internal wind physics computation, causing both X-Plane's physics and our custom physics to overlap. To avoid this, we have partnered with HiFi to develop additional features that enable a bypass of the wind conditions between ASXP and our X-Plane plugin, as explained in Figure 5. When enabling this option via their API, the internal wind conditions in X-Plane are set to zero.

Thanks to this bypass, we are able to retrieve live wind data and compute the corresponding aerodynamic forces and moments without overlapping X-Plane's internal forces and moments due to wind.

Ground effect simulation

Another difficulty that pilots have to face during actual take-off and landing maneuvers is ground effect. Ground effect (GE) is a consequence of the distortion of the airflow below surfaces of a UAV, due to the proximity of the ground. It applies to both fixed and rotary wing aircraft [29]. As a result, the UAV experiences a continuous variation in thrust, making it difficult to control in proximity to the ground.

GE can be expressed in terms of the increase in thrust (or thrust coefficient) at constant power. Back in the 20th century, this phenomenon was extensively studied in helicopters by Betz [30], Knight and Hegner [31], Zbrozek [32], arriving to the expression formulated by Cheeseman and Bennett [33]:

$$\left[\frac{T}{T_{\infty}} \right]_{P=const} = \frac{1}{1 - (R/4z)^2}, \quad (10)$$

where T is the thrust in GE, T_{∞} is the thrust produced outside GE at the same power condition P , R is the propeller radius, and z the propeller separation from the ground.

Although several later works have been proposed [34-36], this simple equation remains as a good starting point to calculate the influence of ground effect over global thrust in hover flight. Thus, it has been implemented directly into our X-Plane's custom ground effect plugin.

Table 1 contains a summary of all the typical parameters that must be collected in order to build the simulator. Additional parameters may be required depending of the aircraft typology. These variables have been specifically defined in our simulator for the MARVIN aircraft.

Instructor station and failure simulation

Most commercial training flight simulators are equipped with an external instructor station that gives full control over the simulation software to a flight instructor. This station is often used during training to pause and resume the simulator for instructor feedback, dynamically modify the flight scenario, and emulate all sorts of aircraft failures.

Table 1: Summary of the VTOL UAV simulator parameters required for setup.

Parameter	Symbol	Notes
Mass	m	Including common payloads.
Inertia tensor	$I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$	
Center of gravity	(x_{cg}, y_{cg}, z_{cg})	
Center of tilt	(x_{tilt}, z_{tilt})	
Motor position	(x_{Mi}, y_{Mi}, z_{Mi})	Specify for each engine i.
Front/rear motor plane offset	$(Z_{off,front}, Z_{off,rear})$	
Wingspan	b	
Tail rod length	l_{tail}	
Landing gear height	h_{ldg}	
Wing chord	C_{root}, C_{tip}	Ctip only if tapered wing.
Sweep angle, dihedral angle	φ, Γ	
Thickness	t	
Aerodynamic center	(x_{AC}, y_{AC}, z_{AC})	
Wing position	(x_w, y_w, z_w)	Relative to the fuselage.
2D airfoil coefficient plots	C_L, C_D, C_m vs. α	
Global aerodynamic coefficient plots	C_L, C_D, C_M vs. α	For validation of the entire aircraft.
Control surfaces	<i>HTP, VTP, ailerons</i>	Same parameters as the wing.
Maximum and minimum deflection	β_{max}, β_{min}	For control surfaces.
Motor performance: thrust, speed, current, voltage	T, RPM, I, V vs. <i>pulse width</i>	Tabular or plot data.
Typical cruise speed	V_c	
Stall speed	V_s	
Transition speed	V_t	Minimum speed to transition from copter to plane mode.
Maximum speed	V_{max}	
Minimum turn radius or standard rate of turn	R_{turn}, RoT	
Tabular aerodynamic coefficients	$C_L, C_D, C_Q, C_M, C_{LA}, C_{NA}$	For various roll, pitch and yaw combinations.
Propeller radius	R	Used for ground effect estimation.

The commercially-available FS-Flight Control Instructor Operator Station (IOS) is the most suitable instructor station software for UAV pilot training. It is equipped in our X-Plane simulator, allowing the instructor to manage error and emergency situations to train the pilot. Multiple failures can be entered by the instructor at any time during pilot training such as a motor loss of power.

Results and Discussion

The whole VTOL UAV Simulator system layout is summarized in [Figure 6](#). The simulator is subdivided into three subsystems that can be run independently when connected to X-Plane:

- 1. Weather & Physics simulator:** including the Active Sky XP plugin, our custom wind physics engine and X-Plane plugin, and our custom ground effect plugin.
- 2. Flight Control HIL:** including the ground control station software and the Veronte Autopilot.
- 3. Additional plugins,** such as the Instructor Operation Station.

VTOL UAV Simulator System Layout

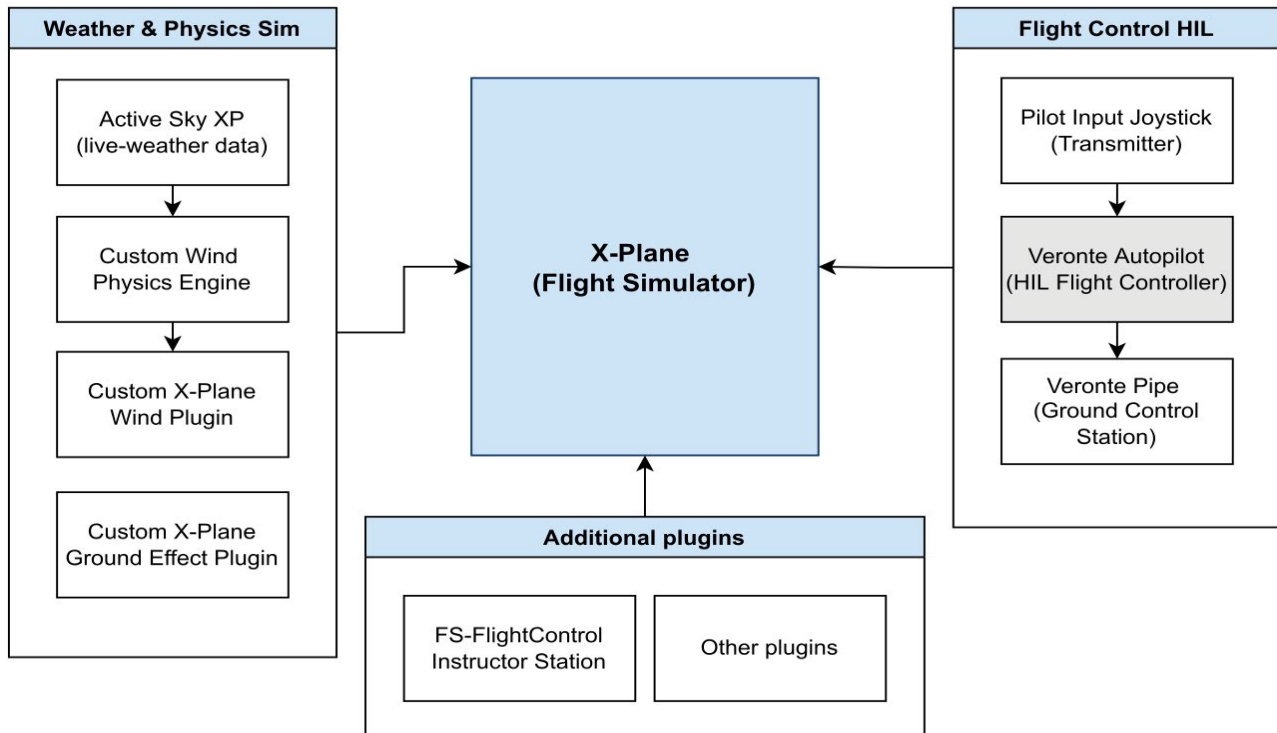


Figure 6: VTOL UAV Simulator System block diagram featuring all three subsystems connected to X-Plane.

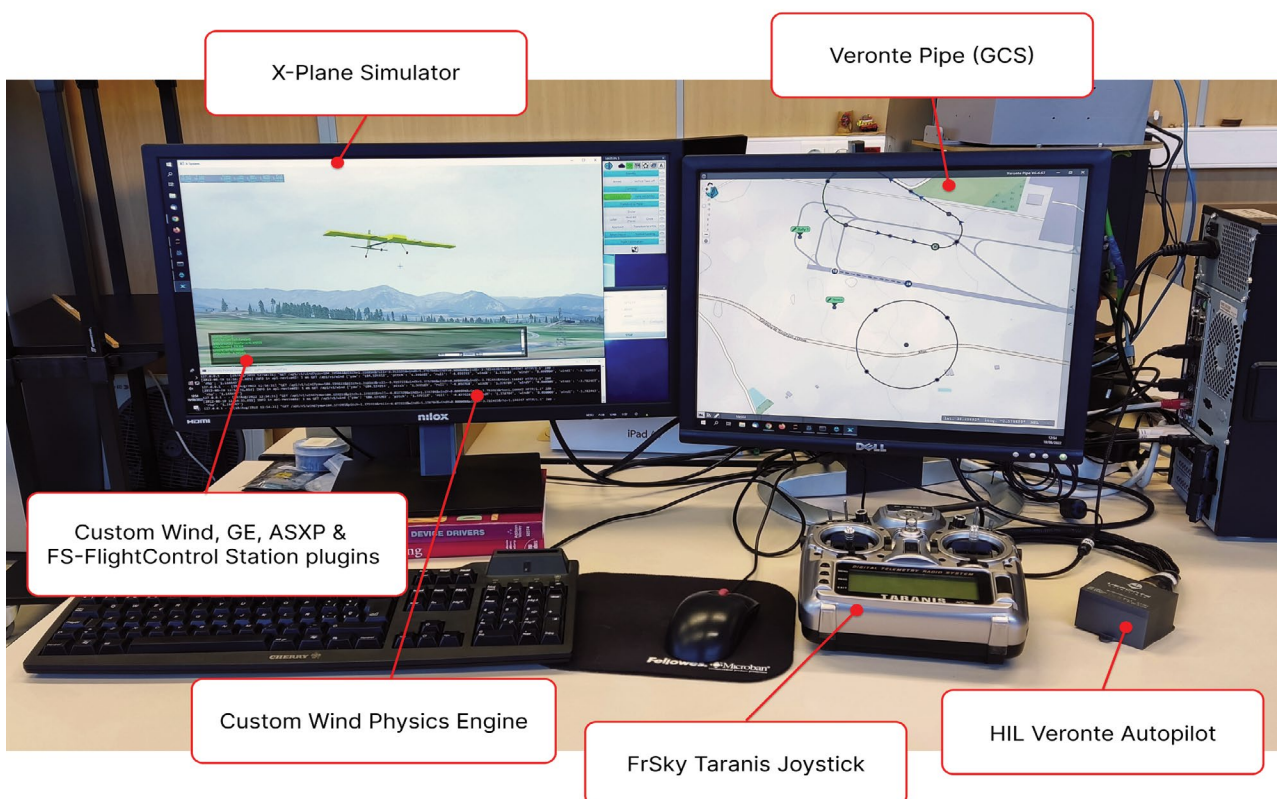


Figure 7: Annotated photograph of the full working setup as seen by a pilot during training.

The resulting flight simulation setup can be seen in [Figure 7](#). The pilot is given a computer connected to two screens, a transmitter, and peripherals to control the Veronte Pipe software.

On the left screen, the pilot is presented with an instance of X-Plane running the VTOL UAV model. The right screen contains the Veronte Pipe ground control station software, where the pilot can setup a mission, load it into the UAV and track the flight just like a real scenario.

A third screen outside the pilot view is used by an instructor to run the FS-Flight Control Instructor Station, to simulate failures, pause and resume the simulator when needed, and review the pilot's performance.

This setup is currently in use by the actual VTOL UAV operators to validate mission plans, practice before flight and basic simulator training. In the upcoming future, the authors will focus their efforts on verifying whether this simulator can provide competence development with pedagogical and economic advantages over conventional outdoors training.

Conclusion

Throughout this paper, a flight simulator has been proposed and successfully developed for VTOL UAV pilot training of the MARVIN commercial aircraft based on X-Plane. Custom plugins and a custom physics model have been implemented to tackle complex scenarios such as take-off, landing and hovering flight in the presence of wind and ground proximity.

Although these custom plugins and physics engine are designed for general use, an aerodynamic model of this particular aircraft has been constructed from multiple CFD simulations. A procedure is described to adapt the simulator to any other VTOL UAV.

A commercial flight controller has been employed to provide realistic flight control (HIL simulation) in a simulated environment. The result is a potentially high technological readiness level (TRL) [37] flight simulation environment for VTOL UAV pilot training of the MARVIN aircraft.

References

1. Ko A, Ohanian O, Gelhausen P (2007) Ducted fan uav modeling and simulation in preliminary design. AIAA Modeling and Simulation Technologies Conference and Exhibit 6375.
2. Tekinalp, Unlu T, Yavrucuk I (2009) Simulation and flight control of a tilt duct uav. AIAA Modeling and Simulation Technologies Conference 6138.
3. Cao J, Anvar AM (2012) Design, modelling and simulation of maritime uav-vtol flight dynamics. Applied Mechanics and Materials Trans Tech Publ 152: 1533-1538.
4. Muraoka K, Okada N, Kubo D, Sato M (2012) Transition flight of quad tilt wing vtol uav. 28th Congress of the International Council of the Aeronautical Sciences.
5. Ke Y, Chen BM (2017) Full envelope dynamics modeling and simulation for tail-sitter hybrid uavs. 2017 36th Chinese Control Conference 2242-2247.
6. Lyu X, Gu H, Zhou J, Li Z, Shen S, Zhang F (2018) Simulation and flight experiments of a quadrotor tail-sitter vertical take-off and landing unmanned aerial vehicle with wide flight envelope. International Journal of Micro Air Vehicles 10: 303-317.
7. Walid M, Slaheddine N, Mohamed A, Lamjed B (2014) Modeling and control of a quadrotor uav. 2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) 343-348.
8. Lyu X, Gu H, Wang Y, Li Z, Shen S, et al. (2017) Design and implementation of a quadrotor tail-sitter vtol uav. IEEE International Conference on Robotics and Automation (ICRA) 3924-3930.
9. Oosedo A, Konno T, Matsumoto K, Go K, Masuko M, et al. (2012) Design and attitude control of a quad-rotor tail-sitter vertical takeoff and landing unmanned aerial vehicle. Advanced Robotics 26: 307-326.
10. Carlson SJ, Papachristos C (2021) The minihawk-vtol: Design, modeling, and experiments of a rapidly-prototyped tiltrotor uav. International Conference on Unmanned Aircraft Systems 777-786.

- 11.ERSOY E , YALCIN MK (2018) Designing autopilot system for fixed wing flight mode of a tilt-rotor uav in a virtual environment: X-plane. International Advanced Researches and Engineering Journal 2: 33-42.
- 12.Yu L, He G, Zhao S, Wang X, Shen L (2020) Design and implementation of a hardware-in-the-loop simulation system for a tilt trirotor uav. Journal of Advanced Transportation.
- 13.Vuruskan, Yuksek B, Ozdemir U, Yukselen A, Inalhan G (2014) Dynamic modeling of a fixed-wing vtol uav. 2014 International Conference on Unmanned Aircraft Systems (ICUAS) 483-491.
- 14.Caro PW (1973) Aircraft simulators and pilot training. Human Factors 15: 502-509.
- 15.McLean GM, Lambeth S, Mavin T (2016) The use of simulation in ab initio pilot training. The International Journal of Aviation Psychology 26: 36-45.
- 16.Gu H, Wu D, Liu H (2009) Development of a novel low-cost flight simulator for pilot training. International Journal of Computer and Systems Engineering 3: 1581-1585.
- 17.Dahlstrom N, Dekker S, van Winsen R, Nyce J (2017) Fidelity and validity of simulator training. Simulation in Aviation Training, Routledge 135-144.
- 18.Craighead J, Murphy R, Burke J, Goldiez B (2007) A survey of commercial open source unmanned vehicle simulators. Proceedings 2007 IEEE International Conference on Robotics and Automation 852-857.
- 19.Drela M (1989) Xfoil: An analysis and design system for low reynolds number airfoils. Low Reynolds number aerodynamics 1-12.
- 20.Tsien HS, Lees L (1945) The glauert-prandtl approximation for subsonic flows of a compressible fluid. Journal of the Aeronautical Sciences 12: 173-187.
- 21.Selig MS (2014) Real-time flight simulation of highly maneuverable unmanned aerial vehicles. Journal of Aircraft 51: 1705-1725.
- 22.Isermann R, Schaffnit J, Sinsel S (1999) Hardware-in-the-loop simulation for the design and testing of engine-control systems. Control Engineering Practice 7: 643-653.
- 23.Cole JS Jr, Jolly AC (1996) Hardware-in-the-loop simulation at the us army missile command. Technologies for Synthetic Environments: Hardware-in-the-Loop Testing. 2741: 14-19.
- 24.Sisle ME, McCarthy ED (1982) Hardware-in-the-loop simulation for an active missile. Simulation 39: 159-167.
- 25.Eguchi H, Yamashita T (200) Benefits of hwil simulation to develop guidance and control systems for missiles. Technologies for Synthetic Environments: Hardware-in-the-Loop Testing V. 4027: 66-73.
- 26.<https://www.embention.com/es/producto/autopiloto-simple/>
- 27.Grinberg M (2018) Flask web development: Developing web applications with python. O'Reilly Media, Inc.
- 28.Active sky xp.
- 29.Aich S, Ahuja C, Gupta T, Arulmozhivarman P (2014) Analysis of ground effect on multi-rotors. International Conference on Electron- ics, Communication and Computational Engineering (ICECCE).
- 30.Betz (1937) The ground effect on lifting propellers. Tech rep.
- 31.Knight M, Hegner RA (1941) Analysis of ground effect on the lifting airscrew. Tech rep.
- 32.Zbrozek J (1950) Ground effect on the lifting rotor. ARC RM 2347 1-8.
- 33.Cheeseman I, Bennett W (1955) The effect of the ground on a helicopter rotor in forward flight.
- 34.Fradenburgh EA (1960) The helicopter and the ground effect machine. Journal of the American Helicopter Society 5: 24-33.
- 35.Marr R, Ford D, Ferguson S (1974) Analysis of the wind tunnel test of a tilt rotor power force model. Tech Rep.
- 36.Hayden JS (1976) The effect of the ground on helicopter hovering power required. In: Proc AHS 32nd Annual Forum.
- 37.Conrow EH (2011) Estimating technology readiness level coefficients. Journal of Spacecraft and Rockets 48: 146-152.