Multi-Vehicle Simulation Framework for Heterogeneous Unconventional MAVs

Luiz F. T. Fernandez* [*1,2], Murat Bronz* [†1], Thierry Lefebvre[‡2], and Nathalie Bartoli[§2]

[1]ENAC, Université de Toulouse, F-31400 Toulouse, France
[2]ONERA/DTIS, Université de Toulouse, Toulouse, France

ABSTRACT

The growing interest in using aerial vehicles for diverse missions creates the need for higher autonomy and dexterity. Unconventional configurations can potentially satisfy such requirements, at a cost of new challenges in design and control. Simulation environments are often employed to tackle such challenges while minimizing resources expenditure. In this paper, an existing simulation framework has been extended to allow for the modeling of new UAV configurations. Aerodynamic forces and moments generated by propellers and airframes are defined in separate functions to ease the understanding and allow future addition of new vehicle configurations. Incremental nonlinear dynamic inversion and vector field guidance methods are implemented as vehicle agnostic control and guidance solutions. Vehicle models are extracted and implemented into Paparazzi open-source autopilot system, which makes a seamless switch towards real flights and hardware in the loop tests, facilitating control law development. The simulator can also be used within vehicle and mission design environments. With that, the entire closed loop behavior of the system can be addressed from the design stage, allowing for the vehicle and fleet optimization while considering control and operation constraints. Finally, the whole codebase is made available for the community and can be reached at:

github.com/enac-drones/dronesim

1 INTRODUCTION

Simulators have become indispensable tools in aeronautical and robotics research. Their use is one of the factors that led to the increasing interest in deploying aerial robots for real life missions. In a fully computational environment, researchers can evaluate vehicle performance and different control strategies without expending resources to manufacture and test robots. By employing a simulator to train a learning based solution, Loquercio *et al.* [1] showed how quadcopters can safely and nimbly navigate through complex environments such as forests and disaster zones. Petracek *et al.* [2] also used simulations to assure that two unmanned aerial vehicles (UAVs) could be used to document historical buildings without endangering them. Thanks to the successful application of UAVs in different tests or even in real life scenarios, there is an increasing interest in using them in different tasks. Such missions are increasingly complex, covering demands where time plays a major role, such as health [3] related tasks, and the ones involving more interaction with humans, asking for more dexterity, such as construction [4]. To successfully perform more complex missions, novel vehicle configurations are continually being explored. Winged UAVs [5], capable of higher flight ranges, and different propulsion layouts [6], as the vehicles shown in Figure 1, capable of more dexterous flights, are often preferred. However, they create challenges with respect to accurate physics simulation.

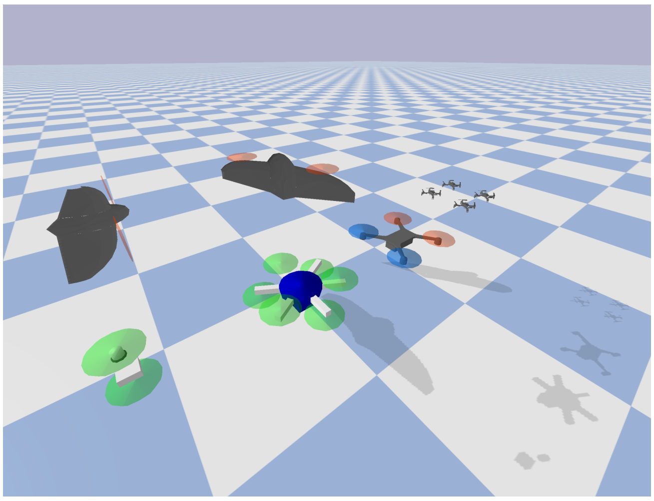

Figure 1: Different vehicle configurations flying together are shown in the simulation environment.

Precise aerodynamic modeling of airframes and propellers then becomes of paramount importance in order to correctly account for vehicle dynamics as it enables the as-

*luiz.tiberio@onera.fr
†murat.bronz@enac.fr
‡thierry.lefebvre@onera.fr
§nathalie.bartoli@onera.fr
*These authors contributed equally to this work

sessment of vehicle stability, control effectiveness, and performance in various flight regimes. Such models are usually built with high order methods in the aeronautical engineering community, using computational fluid dynamics methods as in [7] and [8]. However, the use of such computationally heavy tools remains impracticable at system simulation level for a given trajectory or full mission considering sensing and control systems, as a single point simulation can take hours to run. So, the challenge of accurately addressing aerodynamic behavior of unconventional UAVs within an acceptable computational cost for system simulation remains open.

In addition to the challenges associated with creating new vehicle configurations and defining accurate aerodynamic forces and moments, existing simulators often suffer from limited portability and adaptability. Many simulators are designed to run on specific operating systems or hardware, as NVIDIA GPUs, making them less accessible for researchers using different platforms. This can hinder collaboration and the ability to reproduce research findings across different research groups. In this paper, we present an extension to an existing simulation framework based on Bullet physics engine, which is operating system and hardware-agnostic. The simulator is intended to be used to model new vehicle designs, develop control and guidance laws and simulate mission scenarios while testing different vehicle configurations at the same time. The subsequent sections of this paper delve into the details of related work, proposed solution, and use-case scenarios, highlighting the benefits and potential applications of our approach. Contributions of the paper can be summarized as :

- Facilitate the simulation of new vehicle configurations
- Easier and more intuitive definition of aerodynamic forces and moments
- Heterogeneous multi-vehicle simulation
- Improved portability through different operating systems
- Implementation into a well known open source autopilot system for rapid transition into hardware flights.

2 RELATED WORK

The desire for a simulation tool to evaluate different vehicle designs in [9, 10], and novel control strategies for such vehicles, as in [11], led to the search of available solutions. The interest in testing different guidance strategies [12] and even learning approaches [13] increased even further the need for a reliable and flexible tool. The several available simulators for aerial vehicles are different in terms of objective, implementation, available vehicles, flexibility, and so on. So, the choice of evaluating existing solutions and selecting the most suitable one was made. Fadri *et al.* [14] presented the *RotorS*, a simulator for different multicopter vehicles. The implementation in Robot Operating System (ROS) and Gazebo facilitates inclusion of sensors for odometry and an Inertial Measurement Unit (IMU). In the same vein, Baca *et al.* [15] presented the *MRS UAV system*, a ROS-based solution to simulate different vehicles while enabling real world deployment and facilitating reproducible research. Yunlong *et al.* [16] presented the *Flighmare*, a flexible quadrotor simulator based on Unity and a physics engine that allows for the inclusion of sensing system on the simulation, evaluation of reinforcement based strategies for control, and more. From the same research group, Foehn *et al.* [17] presented *Agilicious*, an open-source and open-hardware agile quadrotor for vision-based flights. This solution comes with *agilib*, a library with minimal dependencies to integrate *Agilicious* into a given infrastructure, and *agiros*, a ROS based library that allows for easy simulation and real flight. Fernandez-Cortizas *et al.* [18] presented the *Aerostack2*, a ROS2 based simulator for multi-aerial-robots systems. Keipour *et al.* [19] presented a Matlab based simulator for fully-actuated vehicles. With the main focus of studying learning based methods, Kulkarni *et al.* [20] presented the *Aerial Gym Simulator*, based on the NVIDIA Isaac Gym. Panerati *et al.* [21] also presented a Python based simulator adapted for learning methods. The authors introduced an environment called *Gym-Pybullet-Drones*, based on the Bullet physics engine, which supports multiple quadcopters and provides realistic collisions, aerodynamic effects, and reinforcement learning interfaces. With the objective of helping in education, Folk *et al.* [22] presented the *RotorPy*, a python based multirotor simulator that considers aerodynamic effects. More simulation frameworks and research directions can be found in the extensive review by Mairaj *et al.* [23]. Considering the needs of adding different vehicle configurations and aerodynamic and propeller modeling, as well as some aspects of the available simulators, such as maturity, ease of integration with the Paparazzi autopilot system and design environment, we opted to use the *Gym-Pybullet-Drones*, extending it according to our interest.

3 EXTENSIONS

Simulating heterogeneous fleets was one of the main objective for the improvement of the existing framework. However, a unified controller, suitable with most of the new and unconventional vehicle configurations (such as coaxial birotors, fully-actuated hexacopters, and tail-sitters), is also implemented. Some of these vehicles require higher-fidelity models to correctly represent aerodynamic forces and moments. Additionally, for the purpose of mission simulations, a more generic guidance algorithm becomes necessary. All of these issues have been addressed and presented below. This tool is under active development. So new vehicles, control strategies and improvements are added in an ongoing basis.

3.1 Vehicle Agnostic Controller

The controller is based on the work of Van Wijngaarden *et al.*[24] and revolves around the control of the angular accelerations in an *incremental* way. The attitude control law calculates the new command (u_c) as a function of the current command (u_f), current angular acceleration ($\dot{\Omega}_f$) and thrust (T_f) as:

$$u_c = u_f + G^+(\nu - \begin{bmatrix}\dot{\Omega}_f \\ T_f\end{bmatrix}) \quad (1)$$

where G^+ is the Moore–Penrose pseudoinverse of the control effectiveness matrix G and ν is a virtual control law that uses a proportional feedback (k_Ω) to control the angular rates as:

$$\nu = \begin{bmatrix}k_\Omega(\Omega_{ref} - \Omega) \\ T_d\end{bmatrix} \quad (2)$$

where T_d is the desired thrust, calculated by the outer loop, and Ω_{ref} is calculated as a function of the quaternion attitude error (q_{err}) , obtained with the Kronecker product between the reference quaternion and the conjugate of the state quaternion:

$$\Omega_{ref} = k_\eta q_{err}^T \quad (3)$$

The velocity control law is:

$$v = v_f + m\left[G_T(\eta, T) + G_L(\eta, V)\right]^{-1}(a_{ref} - a_f) \quad (4)$$

where a_{ref} and a_f are the reference and current accelerations respectively, v_f is the current velocity, and $G_T(\eta, T)$ and $G_L(\eta, V)$ are velocity effectiveness matrices for thrust and lift forces, obtained with the derivatives of such forces with respect to the control vector for the outer loop, defined as roll, pitch, and thrust ($[\phi, \theta, T]$). The subscript $[.]_f$ represents the filtered signal for the current acceleration and velocity vectors. The Bullet physics engine runs at 240 Hz, while the controls run at 96 Hz.

3.2 Vector Field Guidance

A generic guidance law becomes beneficial when a wide variety of vehicle configurations needs to be simulated. So we implemented the Vector Field Guidance method from De Marina *et al.* [25]. The basic guidance law uses the velocity error to generate the reference acceleration as:

$$a_{ref} = [v_{ref} - v]\, k_v \quad (5)$$

where, v is the current velocity, k_v is decomposed as speed and heading gains for tangential and radial directions, and v_{ref} is the reference velocity coming from a predefined vector field. The acceleration terms are neglected, and only velocity information is used. The implementation requires an analytical formulation of the trajectory that is used as a reference, e.g., a circle $x^2 + y^2 + r^2 = 0$. At any point in the environment, a "level-set" can be calculated which has a non-Euclidean distance metric with respect to the reference trajectory, and by driving this distance error towards zero, the vehicle converges to the reference trajectory. Figure 2 shows an example of the outcome of the vector field guidance. The desired velocity (direction and module) for a vehicle at each point in the space (represented by the arrows) is calculated in such a way that the desired trajectory, represented by the circle with radius 100 m, is correctly tracked.

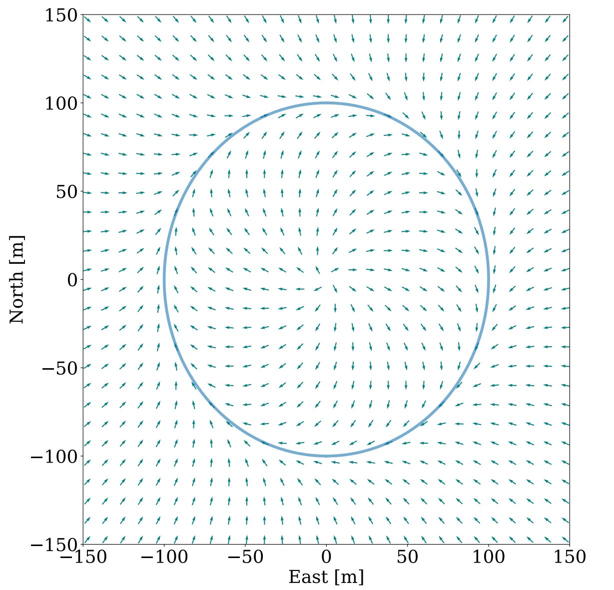

Figure 2: Illustration of vector field guidance, the arrows indicate the velocity direction at each point in the plane.

3.3 Propeller Aerodynamics

To calculate in real time the thrust and moments generated by propellers assuming different geometries and flight conditions, including forward flight, four methodologies are implemented. Both low order data-driven methodologies proposed by Gill and D'Andrea [26] are implemented and indicated when experimental data for thrust and moments for the specific propeller are known. The hybrid blade element momentum (HBEM) by Davoudi and Duraisamy [27] is implemented to account for cases where no performance data is known, only the propeller geometry. In order to use the latter, a valid Matlab license is needed. Lastly, surrogate models can also be used to account for higher fidelity models, enabling a compromise between computational cost and accuracy. Regardless of the surrogate technique or generation strategy, any *Pickle* file containing the model can be read and used for the calculation. Surrogate models created using *CCBlade* [28] were used alongside the simulator in [29] to simulate and optimize a tail-sitter in cruise flight. In [30] seven different methodologies for propeller performance calculation were compared using the proposed simulator. The goal was to observe if, by changing the calculation method in a quadcopter trajectory simulation, the impact in the control input would be noticeable. Despite the fact that desired trajectory was correctly tracked by the vehicle independently of the propulsion methodology, we obtained a significant difference in motor rpm commands, as shown in Figure 3, that compares both methods by Gill and D'Andrea [26] and HBEM [27]. The HBEM led to smaller rpm values, which could lead to a smaller battery requirement during the design phase of a vehicle.

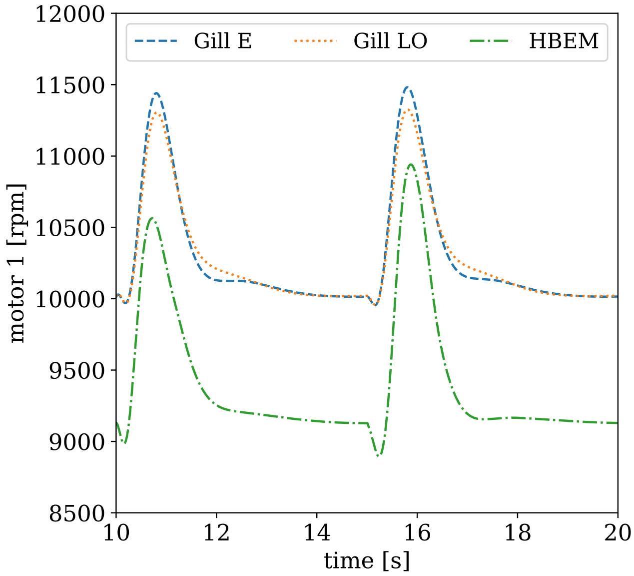

Figure 3: Discrepancies found in rpm commands for a quadcopter for different propeller methods: HBEM and the ones by Gill [30].

3.4 Vertical takeoff and landing UAVs

Modeling the aerodynamic forces and moments experienced by vertical takeoff and landing (VTOL) UAVs remains an open challenge. This kind of vehicle flies in a larger flight envelope than general aviation aircraft, including post-stall conditions. To account for that, the Φ-theory based model, as proposed by Lustosa *et al.* [31] was implemented. This model is algebraically simple and does not rely on angles of attack and sideslip, thus becoming singularity-free, while taking into account the influence of the wing, the deflection of wing control surfaces, and propeller-wing interaction. The same model has already being used by Tal *et al.* [32] to develop a control law capable of enabling agile flight of tail-sitters. In order to make this module applicable for any flying wing configuration, the contributions generated by the propeller-wing interaction in the forces and moments are not taken into account and left for future work, as the original formulation is specific for vehicles with two propellers. Below, we provide a brief explanation of how the aerodynamic forces and moments are generated. For further information, we refer the reader to [31]. Considering a flying wing UAV with 2 elevons and assuming the use of thin airfoils, three Φ-coefficients are needed to build the model using the Φ-theory: the velocity-to-force coefficient $\Phi^{(fv)}$, the velocity-to-moment coefficient $\Phi^{(mv)}$, and the angular-velocity-to-moment coefficient $\Phi^{(m\omega)}$:

$$\Phi^{(fv)} = \begin{bmatrix} C_{D_0} & 0 & 0 \\ 0 & C_{Y_0} & 0 \\ 0 & 0 & 2\pi + C_{D_0} \end{bmatrix} \tag{6}$$

where C_{D_0} and C_{Y_0} are the minimum drag and side force coefficients, respectively.

$$\Phi^{(mv)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -c^{-1}\Delta r(2\pi + C_{D_0}) \\ 0 & b^{-1}\Delta_r C_{Y_0} & 0 \end{bmatrix} \tag{7}$$

where b, c, and Δ_r are the wing span, wing mean aerodynamic chord, and flap chord percentage, respectively.

$$\Phi^{(m\omega)} = \begin{bmatrix} C_{l_p} & C_{l_q} & C_{l_r} \\ C_{m_p} & C_{m_q} & C_{m_r} \\ C_{n_p} & C_{n_q} & C_{n_r} \end{bmatrix} \tag{8}$$

where each coefficient accounts for the variation of the rolling (l), pitching (m), or yawing (n) moment coefficient with dimensionless rate of change of roll p, pitch q, or yaw r rate [33]. Additionally, it is necessary to model the so-called elevon effectiveness for both aileron and elevator configurations ζ_f and $\zeta_m \in \mathbb{R}^3$. With that, the aerodynamic force on the body frame (F_b) at the aerodynamic center is calculated as:

$$\begin{aligned} \sum F_b = -\frac{1}{2}\rho S \Phi^{(fv)} v \boldsymbol{v_b} + \\ \frac{1}{4}\rho S \Phi^{(fv)} [\boldsymbol{\zeta_f}\times](\delta_1 + \delta_2) v \boldsymbol{v_b} - \\ \frac{1}{2}\rho S \Phi^{(mv)} v \boldsymbol{B}\boldsymbol{\omega_b} + \\ \frac{1}{4}\rho S \Phi^{(mv)} [\boldsymbol{\zeta_f}\times](\delta_1 + \delta_2) \boldsymbol{B} v \boldsymbol{v_b} \end{aligned} \tag{9}$$

where $[\boldsymbol{\zeta_f}\times]$ represents the vector product operation matrix of ζ_f, ρ is the air density, S the wing area, δ_1 and δ_2 are the deflections of each elevon, v is the airspeed, $\boldsymbol{v_b}$ and $\boldsymbol{\omega_b}$ are linear and angular speed in the body frame, respectively, and $\boldsymbol{B}$ a matrix defined as a function of wing span (b) and mean aerodynamic chord (c) as:

$$\boldsymbol{B} = \begin{bmatrix} b & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & b \end{bmatrix} \tag{10}$$

The aerodynamic moment (M_b) acting on the body frame at the aerodynamic center is calculated as:

$$\begin{aligned} \sum M_b = -\frac{1}{2}\rho S \boldsymbol{B} \Phi^{(mv)} v \boldsymbol{v_b} - \\ \frac{1}{2}\rho S \boldsymbol{B} \Phi^{(mw)} v \boldsymbol{\omega_b} + \\ \frac{1}{4}\rho S \boldsymbol{B} \Phi^{(mv)} [\boldsymbol{\zeta_m}\times](\delta_1 + \delta_2) v \boldsymbol{v_b} + \\ \frac{1}{4}\rho S \boldsymbol{B} \Phi^{(m\omega)} [\boldsymbol{\zeta_m}\times](\delta_1 + \delta_2) v \boldsymbol{B}\boldsymbol{\omega_b} \end{aligned} \tag{11}$$

Figure 4 shows a visual comparison between a real and a simulated UAV.

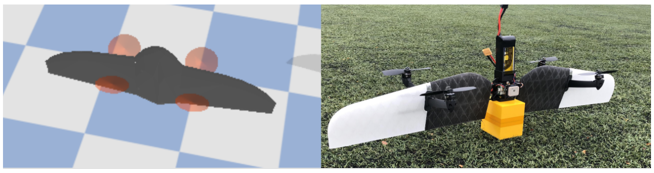

Figure 4: Simulated and real vehicle.

3.5 *Adding new extensions*

Within the simulation environment, every robot is defined in a Unified Robot Description Format (URDF) file. So, defining a new vehicle and its "physics" consists in creating a new URDF file and defining a new function in the *base aviary*. In this new function, each force and moment acting on the vehicle has to be declared, as well as where (or which link) it is applied. Every existing physics can also be updated to account for new methods, such as a new methodology for propeller performance. Control and guidance strategies are implemented in the *control aviary*, so new solutions should be added to this Python script.

4 USE-CASE SCENARIOS

In the preceding sections, we have outlined the essential building blocks of the framework necessary for simulating a comprehensive use-case. In this section, we present potential applications for the tool. The primary aim is to demonstrate the framework's potential and provide guidance and motivation for other researchers by offering ready-to-use templates to initiate their own investigations.

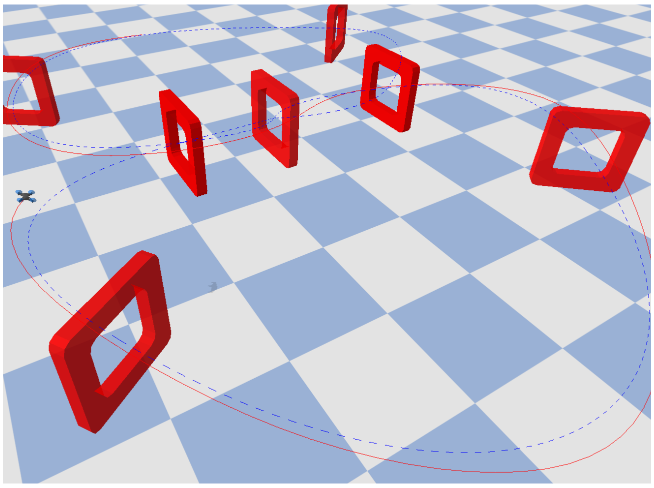

Figure 5: Simulation of a drone passing through gates, representing an example drone racing scenario, with the desired trajectory in blue and the simulated one in red.

4.1 *Drone Racing : Time Optimal Trajectory Tracking*

Drone racing has emerged as an exciting and rapidly growing sport, captivating both enthusiasts and researchers alike. With its unique blend of speed, agility, and precision, drone racing poses unique challenges that require advanced control strategies and trajectory tracking techniques. In the example shown in Figure 5, seven gates have been positioned in order to create a figure-eight maneuver. The drone starts from the middle section of the track, and calculates the fastest trajectory through the gates with a predefined sequence. The trajectory is generated by minimizing the snap of the polynomial trajectory, passing through the center of the gates. It can be seen that the desired trajectory calculated by the minimum snap (blue), and the actual flown trajectory (red) has a significant discrepancy. Other planning and control strategies can be implemented and evaluated in this context before real flights.

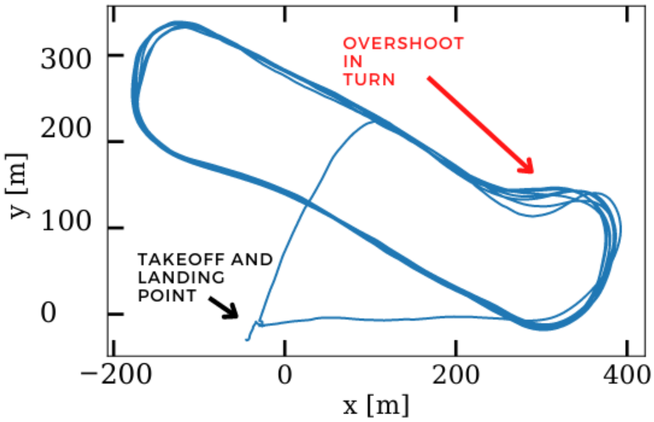

Figure 6: Flight trajectory in the IMAV2022 competition: overshoot in one of the turns caused by strong winds.

4.2 *Vehicle Design and Control Law Optimization*

The proposed simulation framework was used in a multi-disciplinary optimization process applied to a tail-sitter vehicle. A UAV designed in [10] for the IMAV2022[1] presented an undesired overshoot under high wind conditions, as shown in Figure 6. So, in [29], the entire vehicle simulation was embedded in an optimization problem, where the energy consumption for a circular lap was minimized, while reinforcing its wind robustness via constraints. The propulsion layout of the vehicle and the guidance control law gains were the designs variables for the problem. With the process, a more energy efficient and robust to wind vehicle was obtained. Figure 7 shows the differences in the flight path of the baseline and optimized vehicle.

5 IMPLEMENTATION TO PAPARAZZI AUTO PILOT SYSTEM

The integration of a simulator environment with a real drone autopilot system, such as Paparazzi[34], offers numerous advantages for the development and verification of control and guidance algorithms. By enabling the seamless trans-

[1] https://www.imavs.org/2022

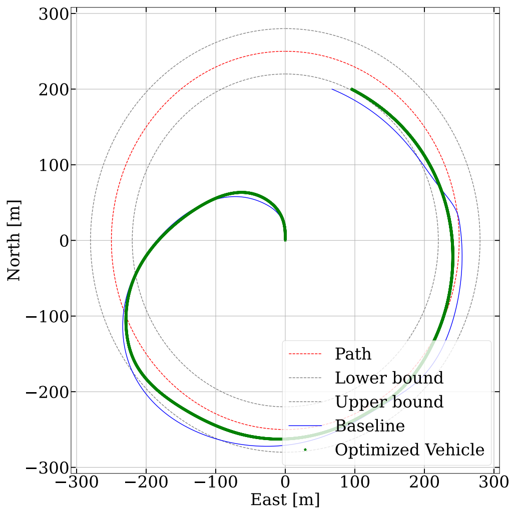

Figure 7: Flight trajectories of baseline and optimized vehicles under wind conditions, with error bounds defined as $\pm 30[m]$.

fer of code between simulation and real-world environments, this approach provides a powerful tool for iterative development, testing, and validation processes. The current develop-

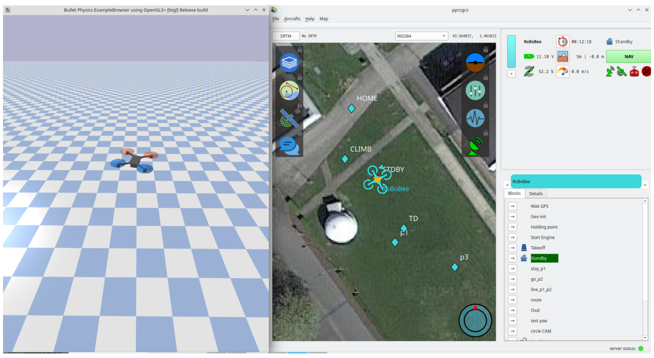

Figure 8: Simulation model running by Paparazzi Autopilot system.

ment state allows for the use of Paparazzi's ground control station to simulate an entire mission. The bridge between simulation and Paparazzi enables the full replacement of the real system with the simulation, while assuming perfect sensing, for the moment. By harnessing the power of simulation, advancements in drone technology can be accelerated, leading to improved aerial capabilities and expanded applications across various industries.

6 CONCLUSION

We presented the extension of an existing framework for simulating aerial robots adapted to account for different vehicle configurations, control and guidance strategies. This open-source solution can be used during design phase, evaluating unconventional vehicle configurations and their performance with respect to a given mission. It can also be used for control and guidance laws development, testing, and benchmark. Its simple structure and the usage of Python allows for easy implementation of new features. This tool has already been used for different applications, such as drone racing and tail-sitter design and optimization. Ultimately, it will be used as well for simultaneous mission and heterogeneous swarm of vehicles design.

ACKNOWLEDGEMENTS

This work is part of the activities of ONERA-ISAE-ENAC joint research group. The authors would like to thank the ENAC Research Engineer Fabien Bonneval for writing the Paparazzi-pyBullet wrapper that is used in this work.

REFERENCES

[1] Antonio Loquercio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Learning high-speed flight in the wild. *Science Robotics*, 6(59):eabg5810, 2021.

[2] Pavel Petracek, Vit Kratky, Tomas Baca, Matej Petrlik, and Martin Saska. New era in cultural heritage preservation: Cooperative aerial autonomy for fast digitalization of difficult-to-access interiors of historical monuments. *IEEE Robotics & Automation Magazine*, pages 2–19, 2023.

[3] Bradley Hiebert, Elysée Nouvet, Vyshnave Jeyabalan, and Lorie Donelle. The application of drones in healthcare and health-related services in north america: A scoping review. *Drones*, 4(3):30, Jul 2020.

[4] Hee-Wook Choi, Hyung-Jin Kim, Sung-Keun Kim, and Wongi S. Na. An overview of drone applications in the construction industry. *Drones*, 7(8):515, Aug 2023.

[5] Charalampos Vourtsis, Victor Casas Rochel, Nathan Samuel Müller, William Stewart, and Dario Floreano. Wind defiant morphing drones. *Advanced Intelligent Systems*, 5(3):2200297, 2023.

[6] Alessandro Mancinelli, Ewoud J.J. Smeur, Bart Remes, and Guido de Croon. Dual-axis tilting rotor quad-plane design, simulation, flight and performance comparison with a conventional quad-plane design. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 197–206, 2022.

[7] Heyecan Koyuncuoglu and Ping He. Cfd based multi-component aerodynamic optimization for wing propeller coupling. In *AIAA SCITECH 2023 Forum*, 2023.

[8] Eduardo J. Alvarez and Andrew Ning. High-fidelity modeling of multirotor aerodynamic interactions for aircraft design. *AIAA Journal*, 58(10):4385–4400, 2020.

[9] Elgiz Baskaya, Mahmoud Hamandi, Murat Bronz, and Antonio Franchi. A novel robust hexarotor capable of static hovering in presence of propeller failure. *IEEE Robotics and Automation Letters*, 6(2):4001–4008, 2021.

[10] Luiz F. T. Fernandez, Murat Bronz, Nathalie Bartoli, and Thierry Lefebvre. Development of a mission-tailored tail-sitter mav. In *13^{th} International Micro Air Vehicle Conference*, pages 159–168, Delft, the Netherlands, Sep 2022. Paper no. IMAV2022-19.

[11] Jacson MO Barth, Jean-Philippe Condomines, Murat Bronz, Jean-Marc Moschetta, Cédric Join, and Michel Fliess. Model-free control algorithms for micro air vehicles with transitioning flight capabilities. *International Journal of Micro Air Vehicles*, 12:1756829320914264, 2020.

[12] Zeynep Bilgin, Murat Bronz, and Ilkay Yavrucuk. Panel Method Based Guidance for Fixed Wing Micro Aerial Vehicles. In *International Micro Air Vehicle Conference*, Delft, Netherlands, September 2022. TU-Delft.

[13] Kirk Hovell, Steve Ulrich, and Murat Bronz. Learned multiagent real-time guidance with applications to quadrotor runway inspection. *Field Robotics*, 2:1105–1133, 03 2022.

[14] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. *Robot Operating System (ROS): The Complete Reference (Volume 1)*, chapter RotorS—A Modular Gazebo MAV Simulator Framework, pages 595–625. Springer International Publishing, Cham, 2016.

[15] Tomas Baca, Matej Petrlik, Matous Vrba, Vojtech Spurny, Robert Penicka, Daniel Hert, and Martin Saska. The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 102, Apr 2021.

[16] Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. Flightmare: A flexible quadrotor simulator. In *Proceedings of the 2020 Conference on Robot Learning*, pages 1147–1157, 2021.

[17] Philipp Foehn, Elia Kaufmann, Angel Romero, Robert Penicka, Sihao Sun, Leonard Bauersfeld, Thomas Laengle, Giovanni Cioffi, Yunlong Song, Antonio Loquercio, and Davide Scaramuzza. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *AAAS Science Robotics*, 2022.

[18] Miguel Fernandez-Cortizas, Martin Molina, Pedro Arias-Perez, Rafael Perez-Segui, David Perez-Saura, and Pascual Campoy. Aerostack2: A software framework for developing multi-robot aerial systems, 2023.

[19] Azarakhsh Keipour, Mohammadreza Mousaei, and Sebastian Scherer. A simulator for fully-actuated uavs, 2023.

[20] Mihir Kulkarni, Theodor J. L. Forgaard, and Kostas Alexis. Aerial gym – isaac gym simulator for aerial robots, 2023.

[21] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[22] Spencer Folk, James Paulos, and Vijay Kumar. Rotorpy: A python-based multirotor simulator with aerodynamics for education and research, 2023.

[23] Aakif Mairaj, Asif I. Baba, and Ahmad Y. Javaid. Application specific drone simulators: Recent advances and challenges. *Simulation Modelling Practice and Theory*, 94:100–117, 2019.

[24] D.C. van Wijngaarden, E.J.J. Smeur, and B.D.W Remes. Flight code convergence: fixedwing, rotorcraft, hybrid. In *12^{th} International Micro Air Vehicle Conference*, pages 21–27, PUEBLA, MEXICO, Nov 2021.

[25] Hector Garcia De Marina, Yuri A Kapitanyuk, Murat Bronz, Gautier Hattenberger, and Ming Cao. Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 5740–5745. IEEE, 2017.

[26] Rajan Gill and Raffaello D'Andrea. Computationally efficient force and moment models for propellers in uav forward flight applications. *Drones*, 3(4), 2019.

[27] Behdad Davoudi and Karthikeyan Duraisamy. A hybrid blade element momentum model for flight simulation of rotary wing unmanned aerial vehicles. In *AIAA Aviation 2019 Forum*, 2019.

[28] Andrew Ning. Using blade element momentum methods with gradient-based design optimization. *Structural and Multidisciplinary Optimization*, 64(2):991–1014, August 2021.

[29] Luiz F. Fernandez, Murat Bronz, Thierry Lefebvre, and Nathalie Bartoli. Assessment of closed loop dynamics in the multidisciplinary design and optimization of small uavs. In *AIAA AVIATION 2023 Forum*, 2023.

[30] Luiz F. Fernandez, Murat Bronz, Nathalie Bartoli, and Thierry Lefebvre. Assessment of methods for propeller performance calculation at high incidence angles. In *AIAA SCITECH 2023 Forum*, 2023.

[31] Leandro R. Lustosa, François Defaÿ, and Jean-Marc Moschetta. Global singularity-free aerodynamic model for algorithmic flight control of tail sitters. *Journal of Guidance, Control, and Dynamics*, 42(2):303–316, 2019.

[32] Ezra Tal and Sertac Karaman. Global incremental flight control for agile maneuvering of a tailsitter flying wing. *Journal of Guidance, Control, and Dynamics*, 45(12):2332–2349, 2022.

[33] J. Roskam. *Airplane Flight Dynamics and Automatic Flight Controls*. Number p. 1 in Airplane Flight Dynamics and Automatic Flight Controls. Design Analysis and Research Corporation, 1998.

[34] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the Paparazzi UAV System for Scientific Research. In *International Micro Air Vehicles Conference and Flight Competition*, pages 247–252, Delft, Netherlands, 2014.