UDC 351.861 https://doi.org/10.20535/2786-8729.4.2024.300614

COMPARATIVE REVIEW OF DRONE SIMULATORS

Mykola Nikolaiev
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0009-0003-0787-6726

Mykhailo Novotarskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-5653-8518

The rise of UAVs has expanded their use in complex tasks such as package delivery and coordinated swarms for military and disaster response. Due to the risks of real-world testing, simulation is crucial for safe algorithm development. This review examines current UAV simulators, identifying their strengths and weaknesses. It outlines essential factors for simulator selection, balancing diversity and standardization, to enhance UAV research efficiency and safety.

Key words: drone, UAV, swarm, simulator, sensor, environmental dynamics

1. Introduction

Unmanned Aerial Vehicles (UAVs), particularly drones, have seen a surge in applications across various sectors, including agriculture, inspection, mapping, and search and rescue efforts. There is a growing interest, particularly in aerial manipulation domains, covering operations from package delivery to inventory management and environmental sampling [1]. Directly deploying experimental algorithms on actual UAV hardware carries inherent risks, such as unpredictable behavior leading to accidents. Besides, UAV mishaps can result in significant financial losses, derail project timelines, and environmental degradation by disposing of broken parts. Moreover, the burgeoning field of machine learning necessitates extensive data collection, proving both inefficient and often infeasible when conducted through direct hardware interaction. As such, a robust UAV simulation environment is pivotal for the accelerated development and innovation within this domain. Fig. 1 depicts a drone engaged in fire suppression activities.

Fig. 1. Fire extinguishing drone in action [2]

Furthermore, the development and utilization of these vehicles extend beyond traditional applications, now venturing into aerial manipulation and enhanced human-robot collaboration. These applications encompass a variety of tasks, including but not limited to parcel delivery, warehouse operations, biological sample collection, and cooperative robotic activities [3], [4]. Directly testing new algorithms on UAVs not only presents safety risks but also poses logistical and environmental challenges due to the potential for crashes and the subsequent need for part replacements. Additionally, with machine learning models becoming integral to UAV operations, the data collection process from physical tests can become a bottleneck, making high-quality simulations invaluable for streamlining development workflows. The complexity and operational scope of UAVs is further expanded when considering drone swarms — networks of UAVs operating in coordinated clusters to accomplish tasks beyond the capacity of a single drone. Drone swarms represent a significant leap forward, enabling multiple drones to operate in a coordinated fashion, achieving complex objectives beyond the capabilities of individual drones [5]. This evolution towards collective drone operations opens new avenues in applications ranging from comprehensive military strategies to effective disaster response and meticulous agricultural surveillance. The inherent complexities and the escalated risk factors associated with the real-world deployment of drone swarms amplify the necessity for advanced simulators in the arenas of research, development, and operational planning. Given the intricate nature of simulations, especially in replicating complex flight dynamics and interactions, selecting the proper UAV simulator becomes critical. This study delves into various leading drone simulation frameworks, discussing crucial criteria and factors pivotal in the selection process. Using our investigations, we also delineate our methodology for choosing and incorporating a simulator.

2. Problem statement

The UAV simulator landscape is diverse and fragmented, presenting various features and capabilities tailored to different UAV types and specific applications. While promoting innovation, this variety complicates selecting the most suitable simulator for particular research needs. Researchers are often faced with the need to balance various trade-offs, including simulation speed, physics accuracy, sensor integration, and user interface. This fragmentation prevents the establishment of standardized benchmarks. In this regard, specific difficulties arise when comparing various simulators.

The primary problem identified in this review is the lack of a universal simulator that meets the diverse and specific needs of UAV research and development. Current simulators exhibit varying strengths and weaknesses, with no tool encompassing all necessary features, such as high-fidelity physics, diverse environmental variability, comprehensive sensor simulation, and scalability for swarm operations. Efficiently simulating the complex behaviors and interactions of multiple drones operating in coordination is a technical challenge that existing tools only partially address.

Another critical problem is integrating diverse sensor simulations. Modern UAV applications rely heavily on a variety of sensors, such as cameras, LiDAR, GPS, and IMUs, for autonomous navigation and task execution. A simulator's ability to model these sensors accurately and produce realistic data outputs is essential for developing robust UAV algorithms. However, not all simulators offer comprehensive support for the range of sensor types, limiting their applicability in specific research contexts.

User accessibility and support infrastructure are also vital considerations. The ease of use, availability of documentation, community support, and licensing costs significantly impact a simulator's adoption and long-term sustainability. Researchers require tools that are not only technically capable but also user-friendly and well-supported to ensure efficient and productive research workflows.

This scientific review aims to conduct a comparative analysis and evaluation of selection criteria to identify directions for enhancing drone simulator effectiveness. By dissecting and evaluating the current landscape of UAV simulators, this review also aims to identify critical areas for improvement and provide a detailed analysis of the selection criteria necessary for effective UAV research and development. Addressing these issues is crucial for guiding future enhancements in drone simulation technology and contributing to advancing aerial robotics.

This review hypothesizes that moving towards a more standardized approach in UAV simulation could yield significant benefits, including enhanced comparability between studies, streamlined collaborative efforts, and a unified framework for benchmarking simulator performance. Furthermore, advancements in aerodynamics modeling, particularly for multirotor and fixed-wing UAVs, are hypothesized to improve simulation accuracy and realism, thereby better supporting the development of advanced UAV technologies.

In summary, the problem statement addresses the need to comprehensively understand current UAV simulators, their capabilities, and limitations. It seeks to identify critical factors influencing simulator selection and highlight areas where improvements are needed to meet the evolving demands of UAV research and development. Through this comparative analysis, the review aims to enhance the effectiveness of drone simulators, fostering innovation and advancing the field of aerial robotics.

3. Literature review of drone simulators

Numerous physics-based simulation tools have emerged for robotics and aerial vehicles, each offering distinct pros and cons. Moreover, the landscape of these simulators is continuously evolving, introducing new advancements. This vast array of choices and ongoing innovations significantly complicates selecting the most appropriate simulator for an individual researcher's needs.

Several survey papers have been published examining simulators and their robotics application [6]. For instance, a recent review [7] covers a broad spectrum of application domains, including those for aerial vehicles, and contrasts the features of simulators across different fields. Unfortunately, not all simulators provide out-of-the-box support for drones. The dynamic models used for manipulators or terrestrial vehicles can be markedly different from those needed for aerial vehicles, particularly when considering the aerodynamic effects crucial for research. In reference [8], the authors delve into various aspects relevant to aerial delivery vehicles, including the choice of simulators. Additionally, in [9], there is an examination of simulators tailored to aerial vehicles, encompassing some that are not widely utilized, and the discussion extends to the criteria for selecting a simulator.

In [7], the authors compare four widely utilized simulators: AirSim [10], Flightmare [11], Gazebo [12], and Webots [13], focusing on their performance in terms of stability, speed, and resource efficiency in reinforcement learning (RL) environments. They explore the balance between the real-time factor and accuracy, noting that faster simulation speeds, achieved by increasing the time step, inversely affect precision. They also consider how user-friendly each simulator is, highlighting potential difficulties new users may face when learning to use the software.

However, integration of these simulators with aerial vehicles is not always straightforward. Differences in dynamics between manipulators or terrestrial vehicles and aerial vehicles are significant, especially in studies aiming to incorporate aerodynamic effects. In [8], a comprehensive evaluation of factors relevant to aerial delivery vehicles is conducted, including simulator choice. This comparison includes RotorS [14] (based on Gazebo), AirSim [10], Flightmare [11], FlightGoggles [15], and gym-pybullet-drones [16] (based on PyBullet). Likewise, [9] provides an analysis of simulators specific to aerial vehicles, assessing some less common options and outlining criteria for selecting a simulator. Additionally, extensions to popular simulators, like PRL4AirSim [17] — an AirSim-based package designed for efficient parallel RL training — are also noted.

The choice of a simulator should be driven by the intended application domain, ensuring that the chosen simulator encompasses the appropriate features and sensors for that domain. Choosing the right UAV simulator involves evaluating numerous factors that significantly impact the effectiveness of simulation outcomes. Additionally, seamless integration with standard autopilots such as PX4 and ArduPilot is commonly evaluated to ensure effective transition from simulation to real-world operation. Drawing upon our experiences and analyses of relevant literature, we have established a set of evaluative criteria and decision-making factors frequently considered in the appraisal of UAV simulators, as illustrated in Table 1. It outlines the key criteria and decision factors that are vital when assessing UAV simulation tools. Each criterion is derived from the identified needs within the UAV research and development community, focusing on aspects such as environmental variability, user interface, simulation speed, and sensor accuracy. This systematic approach aims to simplify the simulator selection process, ensuring that users can effectively match their specific requirements with the capabilities of a simulator, thereby optimizing their research and development efforts.

It is evident that precisely such targeted reviews allow developers to significantly reduce their time identifying critical problems in fields such as modeling the characteristics of drones.

Therefore, we will use these criteria to review the most successful drone simulators.

Table 1

Guidelines for Selecting UAV Simulation

Criteria	**Decision Factors**
1	2
Environmental Variability	Ability to simulate diverse environmental conditions such as weather, time of day, and geographic landscapes
User Interface and Experience	Ease of use, accessibility of controls, and availability of tutorials or guides for new users
Training and Evaluation Features	Availability of training modules, progress tracking, and performance evaluation tools
Simulation Speed	The ability to simulate in real-time or faster, which is essential for learning and development applications
Physics Accuracy	The precision required in the physics and dynamics simulations for the simulator's intended application
Visual Quality	The need for high-quality visual simulations, important for AI training in computer vision and other ML tasks
Flight Control System Compatibility	Compatibility with widely-used flight control systems such as PX4 and ArduPilot, crucial for conducting software and hardware simulation tests
Multiple UAVs	The ability to simulate multiple UAVs concurrently
Sensor Simulation	Support for integrating simulations of common sensors like cameras, inertial measurement units (IMUs), GPS, LiDAR, and optical flow sensors
UAV Models	The capability to support popular UAV models and incorporate new ones with ease
Programming Interfaces	Compatibility with various programming languages and tools like ROS, as well as environments such as Gymnasium and PettingZoo for AI training
Ease of Integration	How straightforward it is to start using and developing with the simulator, and if the project receives regular updates
Scalability and Extensibility	Ability to add more features, vehicles, or scenarios as user needs grow
Cost and Licensing	Initial and ongoing costs, including licensing fees, and any limitations on use
Community and Support	Availability of a user community, forums, and customer support for troubleshooting and development assistance

Continued of the table 1

1	2
Physical Interface Compatibility	Support for real-world UAV controllers, joysticks, and other physical interfaces to enhance the realism of the simulation experience
Graphics and Environmental Realism	Quality of graphics and environmental details, including terrain, obstacles, and realistic lighting effects for immersive simulation experiences
Data Recording and Analysis	Capabilities for recording flight data and analyzing it post-flight for training, research, and development purposes
Data Recording and Analysis	Capabilities for recording flight data and analyzing it post-flight for training, research, and development purposes
VR and AR support	Support for VR and AR technologies to enhance training realism and user immersion

Gazebo Classic [12] stands out as a dynamic and open-source research tool for simulation, celebrated for its modular configuration that facilitates the integration of varied physics engines, sensor systems, and three-dimensional environment crafting. Its utility shines particularly in tasks involving aerial manipulators, thanks to the platform's capability for generating adjustable contact surfaces, a feature critical for nuanced simulation scenarios [18, 19]. However, the reliance on older physics engines in Gazebo Classic may hinder its effectiveness in simulating the latest UAV technologies, potentially leading to less accurate representations of modern drone behaviors.

Rotor [14], developed as an enhancement over Gazebo Classic, serves as a structured environment tailored for UAV design and the crafting of control algorithms, with a special emphasis on replicating accurate vehicle dynamics. RotorS is tailored for UAV design and control algorithm development, but its specialized focus may not suit all types of UAV applications, particularly those requiring diverse environmental interactions. An extension to Rotor, *CrazyS* [20], zeroes in on the simulation of the Crazyflie 2.0 quadrotor, extending the foundational aspects of RotorS. Nonetheless, it is notable that both RotorS and CrazyS present limitations in simulating perception-driven tasks. *PX4 SITL Gazebo* [21], evolving from the RotorS framework, brings forth cutting-edge support for PX4 SITL configurations, functioning independently from ROS, thereby broadening the scope for simulating multiple vehicles simultaneously. Significantly, this simulator introduces an airspeed sensor module, a critical element for the accurate simulation of fixed-wing and VTOL aircraft. The evolution continues with the new Gazebo [22], previously referred to as Ignition. This platform, as the modern successor to Gazebo Classic, advances the field with its inclusion of quadrotor dynamics and control systems inspired by the foundational RotorS initiative, marking a new era in UAV simulation capabilities. While RotorS and CrazyS offer detailed simulation environments for UAV design, their specialized focus might limit their applicability across diverse UAV applications, potentially curtailing their utility in broader research scenarios.

NVIDIA's Isaac Sim [23] stands out as a photorealistic, high-fidelity simulator catering to diverse platforms. Isaac Sim's advanced photorealistic environments provide excellent visual feedback; however, its high-performance requirements may not be justifiable for simpler simulation tasks that do not need detailed graphics. Building upon Isaac Sim, the *Pegasus Simulator* [24] introduces an open-source add-on featuring enhanced multirotor dynamics, the capacity for parallel multi-vehicle simulations, and integration with both PX4 and ROS 2 (Fig. 2), along with added sensor functionalities including a magnetometer, GPS, and barometer. Further, *Isaac Gym* [25] offers a specialized environment for GPU-accelerated reinforcement learning, albeit with simpler visual rendering compared to Isaac Sim. Complementing this suite, *Aerial Gym* [26], an extension of the Isaac Gym Preview Release 4, excels in parallelizing simulations for numerous multirotors and facilitates the customization of obstacle randomization.

Webots [13] is recognized in the robotics community as a versatile and open-source simulator, hosting a vast array of robotic platforms. While predominantly tailored for ground-based robotics applications, Webots also includes two quadrotor models, which, despite employing simplified aerodynamic physics, offer valuable insights into aerial dynamics for beginners and intermediate users. The simulation engine is powered by the Open Dynamics Engine (ODE), providing robust physical simulations; for an in-depth examination, see reference [27]. An inventive use of Webots facilitated the creation of a triphibious robot, blending aquatic, terrestrial, and aerial functionalities, as highlighted in study [28]. In addition to these applications, Webots has been advancing in UAV simulations, contributing to the understanding and development of complex flight dynamics and control strategies. Webots is known for its robustness in ground-based robotics simulations but may not offer the same level of detail and accuracy in aerodynamics needed for more complex UAV applications.

CoppeliaSim [29], known previously as V-REP, excels as a multi-faceted robotics simulator that caters to a wide spectrum of robotics disciplines, including UAVs. It is celebrated for its extensive support for different programming languages and multiple physics engines, enabling precise simulations tailored to specific research needs. Despite this, yet the broad range of features can overwhelm new users without robust documentation or support. The choice of the physics engine, as elaborated in [27], is paramount to circumvent simulation artifacts that can affect the fidelity of results, such as unrealistic collision dynamics and sensor feedback anomalies. Among its varied uses, CoppeliaSim has been instrumental in the field of UAV navigation, particularly in developing and testing obstacle avoidance algorithms [30]. Its flexibility and precision make it an excellent tool for simulating complex UAV behaviors, including swarm intelligence, cooperative control, and interaction with dynamic environments.

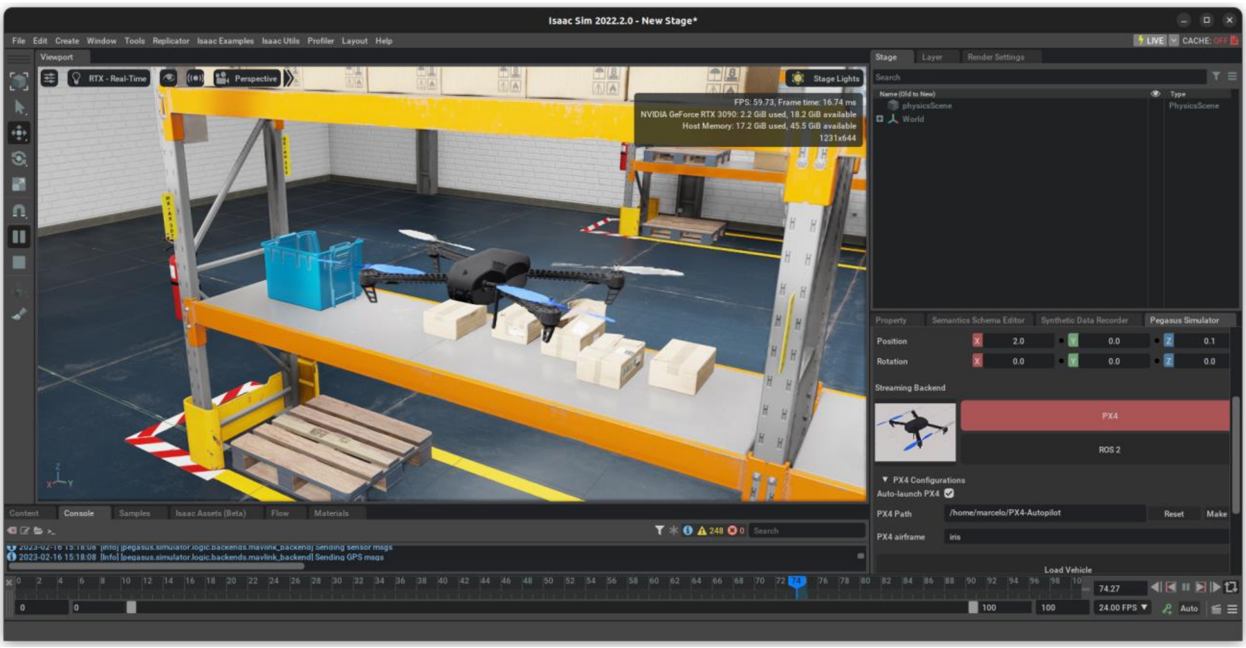

Fig. 2. Pegasus Simulator GUI [31]

MuJoCo, denoted in literature as [32], stands as a widely utilized physics engine in the domain of Machine Learning (ML). It provides interactive visualization capabilities through OpenGL rendering and supports a range of platforms, including UAV models such as the Skydio X2 and the Crazyflie 2 quadrotor.

Initiated by Microsoft, *AirSim* [10] is constructed atop the Unreal Engine, providing a suite of sensors, a weather API, and compatibility with open-source flight controllers like PX4. It's primarily engineered for AI research endeavors, offering platform-independent APIs that facilitate data retrieval and vehicular management. A notable aspect of AirSim is its higher computational requirements in comparison to alternative simulation platforms and can be a significant drawback for researchers with limited processing power, potentially slowing down iterative design and testing cycles. An extension named PRL4AirSim [17] enhances AirSim by enabling efficient parallel training for reinforcement learning (RL) tasks. While the original version of AirSim is open-source, Microsoft has shifted its focus towards Project AirSim, transitioning to a commercial licensing model. Subsequently, the Project AirSim service was discontinued.

Flightmare [33] emerges as a dynamic simulator composed of two core elements: a rendering engine based on Unity and a physics model. Both components are designed for maximum adaptability and can operate independently. The rendering engine is capable of producing lifelike visual outputs and simulating sensor inaccuracies, environmental changes, and optical distortions with a low computational footprint. The physics model provides a broad spectrum of control over robot dynamics, from straightforward, noise-free UAV models to complex simulations involving rigid body dynamics with friction, rotor drag, and even the dynamics of actual platforms. Flightmare's versatility and efficiency have made it a popular tool for ML projects, particularly in the realm of autonomous drone racing [34]. Although Flightmare provides flexibility and independence in its components, the separate management of its physics and rendering engines could complicate integration and increase the learning curve for new users.

FlightGoggles [15], in a manner akin to Flightmare, serves as an open-source simulator dedicated to achieving photorealistic simulations. It integrates two principal components: the use of photogrammetry to render camera sensors with high realism, and the application of virtual reality to incorporate authentic vehicle movement and human interaction within the simulations. Constructed using the Unity engine, FlightGoggles encompasses multirotor physics that include motor dynamics, foundational vehicle aerodynamics, and IMU bias dynamics. A distinctive attribute of FlightGoggles is its 'vehicle-in-the-loop simulation', a setup in which a vehicle navigates within a motion capture system. Here, camera imagery and exteroceptive sensor data are reproduced in Unity, while collision detection operates based on the actual pose of the real-world vehicle. Moreover, the modular framework of FlightGoggles is discussed extensively in [35] which elaborates on its capabilities to simulate realistic camera and sensor dynamics through a customizable platform, enhancing its utility for research in autonomous vehicle systems and robotic perception. FlightGoggles aims for photorealistic simulation, but its intense graphical requirements can exclude users with less advanced hardware, potentially limiting its accessibility.

Fig. 3. AirSim simulator images [36]

The *gym-pybullet-drones* [16] presents an open-source platform for the simulation of quadrotor swarms using PyBullet's [37] physics engine, specifically crafted for investigations at the intersection of control theory and machine learning. This framework offers interfaces tailored for both multi-agent systems and vision-based reinforcement learning (RL) endeavors, integrating seamlessly with the Gymnasium and PettingZoo APIs [38]. It facilitates the crafting of diverse learning environments on a Crazyflie drone platform, incorporating realistic physical interactions like collisions and aerodynamic influences such as drag, ground effect, and downwash. Additionally, gym-pybullet-drones provides sample RL workflows suitable for both individual and collective agent studies, utilizing the capabilities of Stable-baselines3 [39]. While gym-pybullet-drones excels in multi-agent control learning, its physics engine may not accurately replicate the finer aerodynamic effects, which are crucial for high-fidelity UAV simulations.

RotorTM [40] represents an open-source simulation tool specifically designed for the manipulation of aerial objects. Distinguished by its focus on cable-suspended payloads and passive attachment systems among multiple drones, RotorTM offers functionalities not typically found in prevalent simulators. It models the cables as weightless, attaching directly to the drones' center of mass (CoM). These cables can shift between taut and slack states throughout the execution of tasks, enabling the customization of the drone fleet and payload characteristics (for example, selecting between a rigid body or a point mass for the payload). RotorTM also supports configurations where drones are firmly connected to their loads. It simplifies certain physical considerations by assuming minimal drag on both the payload and the drones and omitting the impact of aerodynamic forces, under the premise that rotor dynamics occur at a much quicker pace than these elements. RotorTM focuses on specific aerial manipulation tasks, which might not generalize well to other types of UAV operations, potentially limiting its applicability for diverse research projects.

The *MATLAB UAV Toolbox* [41] serves as a comprehensive suite within MATLAB designed for the creation, simulation, evaluation, and deployment of unmanned aerial vehicles (UAVs). This toolbox provides a range of tools for the development of algorithms, analysis of flight logs, and conducting simulations. Its simulation functionalities include a cuboid simulation framework for rapid scenario development and a sophisticated 3D photorealistic simulation environment that offers synthesized camera and LiDAR data. Moreover, the toolbox features a direct interface for hardware deployment via PX4-based autopilots and supports the MAVLink communication protocol. The MATLAB UAV Toolbox offers a comprehensive simulation suite; however, its proprietary nature and the associated costs could be prohibitive for academic researchers with limited funding. Research endeavors utilizing the MATLAB UAV Toolbox in conjunction with flight simulation software [42], such as X-Plane, FlightGear, and RealFlight.

The open-source environment and benchmark suite known as *safe-control-gym* [43] focuses on safety within the realm of reinforcement learning (RL). Developed on the PyBullet physics engine [37], it aims to facilitate the comparison of control strategies and RL methodologies. It encompasses three dynamic models alongside two distinct control challenges: stabilization and trajectory tracking. Designed to accommodate both model-based and data-driven approaches, safe-control-gym incorporates safety constraints and accurately reflects real-world conditions, including uncertainties in physical attributes and state estimations. While safe-control-gym provides robust safety benchmarks, its focus on specific control challenges may not encompass the broader range of dynamics seen in more versatile UAV operations.

MARSIM [44] emerges as an open-source library coded in C/C++, dedicated to the high-fidelity simulation of LiDAR sensor outputs for UAV applications. It processes point cloud maps to generate depth images, which are then refined to simulate LiDAR points accurately. Aimed at efficient computation, MARSIM provides access to ten detailed environments, covering a variety of settings such as forests, historic buildings, offices, parking garages, and indoor areas, facilitating diverse simulation needs. MARSIM’s detailed focus on LiDAR simulation is highly specialized, which could be a limitation for projects requiring integrated simulation of multiple sensor types.

QuadSwarm [45], available as an open-source Python library, caters to the simulation of multiple quadrotors within reinforcement learning (RL) frameworks, prioritizing speed in simulation and the practical application of simulated policies to real-world conditions. It offers a range of training environments and employs domain randomization techniques to enhance RL models, achieving zero-shot policy transfer in both solo and swarm quadrotor configurations. The simulation bases its physics on the Crazyflie drone model and utilizes OpenGL for graphical rendering.

PyFly [46], a Python-based open-source simulator, is specifically developed for simulating fixed-wing aircraft operations. It boasts a comprehensive 6 degrees of freedom (6-DoF) aerodynamics model, incorporating wind conditions and stochastic turbulence for realism. Complementing PyFly, *fixed-wing-gym* [46] acts as an OpenAI Gym interface tailored to enhance reinforcement learning (RL) endeavors with PyFly, streamlining the integration of fixed-wing aircraft simulations into RL applications. PyFly's specialization in fixed-wing aircraft limits its use for researchers focused on multirotor UAV models, thereby constraining its utility for a broad spectrum of drone research.

ARCAD [47], standing for AirLab Rapid Controller and Aircraft Design, is a MATLAB-based open-source simulator targeting fully-actuated multirotors. It is designed to accelerate the processes of aircraft and controller modeling, design, and analysis. Furthermore, ARCAD specializes in the visualization of complex tasks involving physical interactions, such as performing controlled force-based maneuvers to execute tasks like writing on surfaces. Its main aim is to foster innovation in the design and control of new aircraft models through efficient simulation and analysis.

HIL-airmanip [48] introduces a unique simulation framework designed for exploring the dynamics of physical interactions between humans and drones, facilitating real-time engagement from human participants. Within this simulation environment, the forces exchanged via a haptic interface by a human operator are meticulously measured and relayed to an aerial manipulator. This manipulator is digitally represented within the RotorS simulation suite and comprises a quadrotor equipped with a six-degree-of-freedom (6-DoF) robotic arm mounted on its underside. This setup allows for intricate interaction studies and development of advanced control strategies for human-drone collaboration.

RotorPy [49] emerges as an open-source Python-based simulator, distinguished by its lightweight design and focus on delivering a detailed quadrotor model. Crafted with an emphasis on ease of use, clarity, and educational utility, it was originally developed as a pedagogical instrument for a robotics curriculum at the University of Pennsylvania. The simulator encompasses a comprehensive quadrotor model [49], including 6-DoF dynamics, the effects of aerodynamic forces, the dynamics of actuators, sensor models, and wind influence. The authenticity and accuracy of the model have been corroborated through empirical testing with a Crazyflie drone executing aerial maneuvers, underscoring its potential as a valuable resource for both educational and research applications.

Potato [50] introduces a simulation platform grounded in data-oriented programming principles, tailored for expansive swarm simulations. Mirroring the computational approach of Isaac Gym, Potato leverages GPU resources over traditional CPU-based calculations. The simulator encompasses elementary dynamics for a range of vehicles including fixed-wing drones, quadrotors, and automobiles. While Potato remains proprietary as of now, the creators mentioned [50] a future plan to release the quadrotor segment of the simulator to the open-source community.

Agilicious [51] presents a comprehensive hardware and software framework for the development of autonomous and agile flight capabilities in quadrotors, equipped with a Jetson TX2. Its simulation environment is uniquely tailored, featuring a custom modular design that accurately models aerodynamics, employing blade-element momentum theory among other methodologies, including compatibility with RotorS, Hardware-in-the-Loop (HITL) setups, and integration with visual simulation platforms like Flightmare. Agilicious' reliance on custom hardware configurations and a specific simulation platform may reduce its flexibility and increase the entry barrier for general UAV research use. Though governed by a custom license, Agilicious remains accessible to the academic community upon registration.

The *MRS UAV System* [52] offers a flight stack meticulously crafted for conducting reproducible research, enabled by realistic simulations and tangible experiments. Its simulation framework is anchored in Gazebo Classic, CoppeliaSim, or the bespoke MRSmultirotor-simulator, all of which provide lifelike quadrotor dynamics, sensors, and models. Noteworthy is its seamless compatibility across various releases of the Robot Operating System (ROS), ensuring its relevance and utility through continuous updates and support. The system is particularly favored for coordinating multirotor drone teams.

CrazyChoir [53] introduces a modular framework based on ROS 2, designed to facilitate simulations and experiments with collaborative Crazyflie drones. Leveraging Webots for simulation, it incorporates a Software-in-the-Loop (SITL) emulation of the Crazyflie firmware, offering a foundation for realistic operational scenarios.

Crazyswarm2 [54], akin to CrazyChoir, is a framework dedicated to the management of large-scale indoor quadrotor swarms, specifically with Crazyflie drones. It employs SITL for firmware simulation, wrapped in a flexible simulation architecture that supports either visual-only simulation or a customized Python-based physical simulation, enhancing its utility for swarm control research.

Aerostack2 [55] emerges as a multifaceted, open-source flight stack, ensuring compatibility with a wide array of UAV systems including PX4, ArduPilot, DJI, and Crazyflie. It utilizes Gazebo for simulation, complete with bespoke sensor models, although it presently lacks support for Software/Hardware-in-the-Loop (S/H)ITL simulation frameworks, focusing instead on versatile UAV operational capabilities. The current focus of Aerostack2 on compatibility and operational capabilities may detract from its ability to provide detailed physical simulations, potentially affecting the fidelity of research outputs.

X-Plane [56] stands out as a cross-platform commercial flight simulator primarily targeted at training pilots. It is renowned for its focus on replicating authentic flight dynamics, complemented by sophisticated simulations of weather conditions, wind, and lighting effects, catering to a realistic flight experience. *X-PlaneROS* [57] acts as a ROS-compatible interface for X-Plane, facilitating the control of large fixed-wing vehicles. It allows for the extraction of aircraft data from X-Plane, enhancing capabilities for human-robot interaction within the context of the simulator. *QPlane* [58] is introduced as a reinforcement learning toolkit for fixed-wing aircraft simulation. It is designed to integrate with external flight simulators, including X-Plane and FlightGear, providing a versatile platform for RL-based flight research. Although X-Plane offers detailed flight dynamics for fixed-wing aircraft, its applicability to UAV-specific simulations, particularly for drones and multirotors, may be limited.

FlightGear [59] is recognized as an open-source, community-supported, cross-platform flight simulator. It has garnered attention from the research community, especially for its application in UAV simulation projects [60], demonstrating its versatility and utility in academic settings.

RealFlight [61] presents itself as a commercial RC flight simulator for Windows, featuring a variety of small multirotor and fixed-wing vehicles. Its realistic simulation capabilities have made it a chosen platform for UAV research [62], highlighting its application in advanced simulation scenarios.

Table 2 provides a comparison of critical attributes across various simulation platforms.

The following tables cover simulators widely used in the aircraft field. To maintain brevity, we have excluded simulators that offer limited versatility and result in minimal use.

Rendering with OpenGL and OGRE typically yields lower visual fidelity when compared to the superior visual quality provided by Vulkan, Unity and Unreal engines. The presence of "RL" in the interface column signifies a simulator's specialization for reinforcement learning (RL) applications. A column detailing the "Latest Update" of each simulator at the time of writing this paper has also been included for reference. It is crucial to acknowledge that the status of maintenance is subject to change; therefore, we recommend that the readers view these details as a temporal representation and verify the current status when selecting a simulation tool for their research needs.

Table 2

Comparative overview of features in major drone simulators

Simulator	Physics	Rendering	Interfaces	Licensing	Last Updated	Ref
Isaac (Pegasus, Aerial Gym)	NVIDIA PhysX, Flex	Vulkan	ROS 1/2, Python, RL	Proprietary (BSD 3)	27.02.2023	[27-30]
CoppeliaSim	Bullet, ODE, Vortex, Newton, MuJoCo	OpenGL	ROS 1/2, C/C++, Python, MATLAB, Java, Lua, Octave	GNU GPL, Comm.	14.06.2024	[29]
Gazebo	Bullet, DART, TPE	OGRE	ROS 1/2, C++, RL, Python	Apache-2.0	Sep, 2023	[22]
Gazebo Classic	ODE, Bullet, DART, Simbody	OGRE	ROS 1/2, C++, RL	Apache-2.0	06.10.2023	[12]
Webots	ODE	OpenGL	ROS 1/2, C/C++, Python, MATLAB, Java	Apache-2.0	28.06.2023	[13]
AirSim	NVIDIA PhysX	Unreal, Unity	ROS 1, C++, Python, C#, Java, RL	MIT	18.07.2022	[10], [36]
Flightmare	Ad hoc, Gazebo Classic	Unity	ROS 1, C++, RL	MIT	01.12.2020	[33]
FlightGoggles	Ad hoc	Unity	ROS 1, C++	MIT	31.05.2021	[15]
gym-pybulletdrones	PyBullet	OpenGL	Python, RL	MIT	02.03.2024	[16]
RotorTM	Ad hoc	OpenGL	ROS 1, Python, MATLAB	GNU GPL	19.07.2023	[40]
MATLAB UAV Toolbox	MATLAB	Unreal	ROS 2, MATLAB	Proprietary, Comm.	2024	[41]

Table 3 offers a comparative overview of the sensors accommodated by each simulator. Abbreviations used include “Seg” for segmentation, “Mag” for magnetometer, and “Baro” for barometer. This table lists the features and sensors that are supported in the standard setups of the simulators, while recognizing that numerous simulators possess the capability to be enhanced for broader support.

Table 3
Overview of sensors included in popular aerial vehicle simulators

Simulator	RGB	Depth	Seg.	IMU	Mag.	GPS	Baro.	LiDAR	Optical Flow
Isaac (Pegasus, Aerial Gym)	Yes	Yes	Yes	Yes, No	No	No	No	Yes, No	Yes
CoppeliaSim	Yes	Yes	No	Yes	No	Yes	No	Yes	No
Gazebo	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gazebo Classic	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Webots	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No
AirSim	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Flightmare	Yes	Yes	Yes	No	No	No	No	No	Yes
FlightGoggles	Yes	Yes	Yes	Yes	No	No	No	No	Yes
gym-pybullet-drones	Yes	Yes	Yes	No	No	No	No	No	No
RotorTM	No	No	No	No	No	No	No	No	No
MATLAB UAV Toolbox	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No

Despite the advances in UAV simulators, several unsolved problems remain. First, there is no universal simulator that meets all the diverse needs of UAV research, including high-fidelity physics, diverse environmental variability, comprehensive sensor simulation, and scalability for swarm operations. This fragmentation forces researchers to use multiple simulators, increasing complexity and inefficiency.

Second, not all simulators support the full range of sensors essential for modern UAV applications, such as cameras, LiDAR, GPS, and IMUs. This limitation restricts their applicability in specific research contexts, hindering the development of robust UAV algorithms.

Third, efficiently simulating the complex behaviors and interactions of multiple drones operating in coordination remains a significant technical challenge. Many simulators struggle with scalability and realistic modeling of swarm dynamics, impeding research in coordinated surveillance, search and rescue operations, and other collaborative tasks.

Fourth, user accessibility and support infrastructure are critical issues. Many simulators lack comprehensive user support, have steep learning curves, and vary widely in terms of documentation, community support, and licensing costs. This can deter new users and reduce research efficiency.

Fifth, achieving a balance between real-time simulation speed and high-fidelity physics is challenging. Faster simulation speeds often come at the expense of precision, leading to less accurate modeling of UAV behaviors, which can negatively impact the development and validation of UAV algorithms and control systems.

Sixth, the fragmentation of UAV simulators complicates the establishment of standardized benchmarks, making it difficult to compare and evaluate different simulators comprehensively. This lack of standardization hinders collaborative efforts, reproducibility of research findings, and the overall advancement of UAV simulation technology.

Lastly, detailed aerodynamic modeling is essential for accurately simulating UAV interactions with dynamic environmental elements, particularly for fixed-wing aircraft and scenarios involving constrained environments. Many simulators lack advanced aerodynamics capabilities, resulting in less realistic simulations and affecting the development of UAV technologies that rely on precise aerodynamic interactions.

To enhance the effectiveness of drone simulators, it is essential to focus on developing more comprehensive, high-fidelity, and scalable simulation environments. Future research should aim to integrate advanced sensor simulation capabilities, improve the modeling of swarm dynamics, and incorporate detailed aerodynamic principles. Additionally, efforts should be made to standardize benchmarking practices and improve user accessibility and support infrastructure to foster collaboration and innovation in UAV research and development. Addressing these issues is crucial for advancing UAV research and development.

4. Discussion

Navigating the landscape of aerial vehicle simulators presents a nuanced challenge, poised at the intersection of technological specificity and research objectives. The undertaking, while complex, promises substantial rewards, manifesting as enhanced safety protocols, reduced experimental timelines, and diminished costs. This paper has endeavored to illuminate the breadth of robotic simulators tailored for aerial applications, providing a critical examination of their distinguishing features, supported vehicle dynamics, and sensor integration capabilities. Through a detailed comparison of widely adopted simulation platforms, we have outlined pivotal considerations that researchers should weigh when in pursuit of the optimal simulation environment.

Our comparative review is intended as a resource for the aerial vehicle research community, offering insights that facilitate informed decisions in the selection of a simulation environment. Ultimately, it is the alignment of the simulator's capabilities with the specific demands of the intended application that will dictate the success of research endeavors in aerial robotics. Through meticulous analysis, we have delineated the capabilities, strengths, and limitations of current simulators, emphasizing the essential criteria such as fidelity, environmental variability, sensor accuracy, and the simulation of drone swarms. Our investigation reveals a pressing need for simulators that can more accurately mimic real-world conditions, support a broader array of sensors, and efficiently manage the complexities of drone swarm operations.

The gaps identified in aerodynamics simulation, sensor diversity, and scalability for large-scale drone simulators underscore the necessity for targeted enhancements. These shortcomings not only limit the potential for research and development but also impede the application of UAV technology in sectors where precision, reliability, and efficiency are paramount. Addressing these issues is not merely an academic exercise but a critical step towards unlocking the full potential of UAVs in practical, real-world applications. Reflecting on the diverse array of simulators reviewed, it is evident that no single tool currently meets all the requirements of the UAV research community. This diversity, while showcasing the ingenuity and innovation in the field, also highlights the challenges in selecting the appropriate simulator for specific needs. The quest for a universally applicable simulator remains elusive, advocating for a balanced approach that combines the strengths of various tools while fostering standardization to facilitate collaboration and comparability.

Based on the performed analytical review of literary sources, several previously unsolved parts of the general problem have been identified. These include the lack of a universal UAV simulator that meets all diverse research needs, limited support for a full range of sensors, challenges in simulating swarm dynamics, and the need for improved user accessibility and support infrastructure. Additionally, there is a significant need to balance real-time simulation speed with high-fidelity physics, establish standardized benchmarks, and incorporate advanced aerodynamic modeling.

The evolution of aerial robotics has been significantly influenced by the development and refinement of simulators, marking a pivotal stride in the field. This progression, however, unveils a dichotomy rooted in the platform-dependent and application-specific needs of diverse research groups. The singular adoption of a universal simulator that caters to all envisaged scenarios remains an elusive goal, underscoring the inherent diversity and complexity of aerial robotics applications. The discourse on whether to pursue a variety of solutions or to lean towards standardization presents a compelling narrative that merits a deeper examination within the scientific context.

Balancing Diversity and Standardization. The aerial robotics community stands at a crossroads, where the merits of diversity in simulation solutions are weighed against the benefits of standardization. On one hand, the diversity of simulators fosters innovation, allowing researchers to tailor simulations to the nuanced requirements of their specific projects. On the other, standardization could streamline benchmarking efforts and enhance collaborative ventures across the spectrum of aerial robotics research. The prudent path forward appears to entail a strategic investment in a select cadre of simulators, thus amalgamating the strengths of both approaches. This strategy would not only bolster the development of high-fidelity simulations but also pave the way for a more cohesive framework for benchmarking and collaboration.

Aerodynamics and Simulation Fidelity. A focal point of the discussion is the critical role of aerodynamics in the fidelity of UAV simulations. While the consensus suggests that a majority of UAV applications, particularly those involving multirotors, may not necessitate detailed aerodynamic modeling, certain scenarios present an undeniable need for such considerations. These scenarios, including navigation in constrained environments or interaction with dynamic environmental elements, underscore the importance of integrating aerodynamic principles into simulation models, especially for fixed-wing aircraft. Addressing this need is paramount for advancing simulation technology and, by extension, the capabilities of UAVs in complex operational contexts.

Benchmarking, Standardization, and the Academic-Industry Divide. The proliferation of simulators has accentuated the need for standardized benchmarking practices within the field. The absence of a unified framework for benchmarking complicates the comparability and reproducibility of research findings, presenting a significant hurdle to the advancement of the discipline. Furthermore, the distinction between simulators developed in academic settings and those backed by industry highlights a dichotomy in sustainability and support. Bridging this gap, through collaborative efforts and resource sharing, could significantly enhance the robustness and longevity of simulation tools.

Data Sharing, Collaboration, and Resource Identification. The dialogue further extends to the necessity of data sharing and collaborative endeavors within the aerial robotics community. The sharing of simulator data and models emerges as a crucial step toward collective progress, particularly in domains such as perception and autonomous navigation. Moreover, the plethora of available simulators poses a daunting challenge for newcomers. Thus, resource identification and the provision of comprehensive surveys become invaluable tools for guiding researchers towards informed simulator selection.

Scalability for Swarm Simulations. The potential of drone swarms to revolutionize fields such as agriculture, disaster response, and defense is immense. However, our review reveals a gap in the scalability of current simulators to efficiently model the complex behaviors of large-scale swarms. Addressing this issue through distributed computing or cloud-based solutions could significantly advance our ability to simulate, analyze, and optimize swarm operations, paving the way for innovative applications and enhanced operational coordination. It is essential to develop comprehensive simulators that integrate advanced sensor simulation capabilities for more accurate and comprehensive simulations. Enhancing scalability for swarm operations is necessary to efficiently model the complex behaviors and interactions of multiple drones. Simplifying user interfaces, providing comprehensive documentation, and enhancing community support are crucial for increasing the adoption and long-term sustainability of UAV simulators. Additionally, balancing real-time speed with high-fidelity physics through innovations in computational techniques and establishing standardized benchmarks will facilitate the comparison and evaluation of different simulators, promoting collaborative efforts and reproducibility of research findings. Incorporating detailed aerodynamic principles will lead to more realistic simulations, especially for fixed-wing aircraft and scenarios involving dynamic environmental interactions.

Accessibility, Maintenance, and Sustainability. Lastly, the discussion veers towards the practical aspects of simulator selection, emphasizing the importance of accessibility, maintenance, and sustainability. Licensing considerations, the open-source availability, and the long-term support of simulators are pivotal factors that influence their utility and adaptability for research purposes. Encouraging researchers to delve into these aspects can facilitate the selection of simulators that not only meet current needs but also remain viable tools in the evolving landscape of aerial robotics research.

This allows us to state that it is appropriate to conduct a study devoted to developing more comprehensive, high-fidelity, and scalable UAV simulation environments. Addressing these gaps by integrating advanced sensor simulation capabilities, improving the modeling of swarm dynamics, and enhancing user accessibility and support infrastructure is crucial.

While the development of UAV simulators has reached remarkable heights, the path forward requires a nuanced understanding of the field's complexities. The tasks identified for scientific research based on this review do not merely aim to rectify the identified shortcomings but strive to elevate the field to new heights of innovation and application. Striking a balance between diversification and standardization, embracing aerodynamic modeling in pertinent scenarios, fostering benchmarking and standardization, and promoting data sharing and collaboration are essential strides towards realizing the full potential of aerial robotics simulations. This multidimensional approach not only aligns with the scientific context but also ensures consistency in advancing the frontiers of aerial robotics research. By addressing these issues through targeted research, we can significantly enhance the effectiveness of drone simulators. This involves integrating advanced sensor simulation capabilities, developing scalable solutions for swarm operations, improving user accessibility, balancing real-time speed with high-fidelity physics, standardizing benchmarking practices, and incorporating detailed aerodynamic principles. These efforts will contribute to the advancement of UAV research and development, fostering innovation and improving the reliability and efficiency of UAV operations. By judiciously selecting and utilizing simulation tools, researchers can significantly contribute to the evolution of aerial robotics, fostering innovation and enhancing the reliability and efficiency of UAV operations. We aspire that this paper will serve as a cornerstone for future research, inspiring a new wave of discoveries and developments in the dynamic realm of aerial vehicle simulation. In sum, the advancement of aerial robotics is intricately linked to the continuous development and refinement of simulation environments. The detailed exploration of simulators presented in this paper underscores their indispensable role in fostering innovation, enhancing operational reliability, and pushing the boundaries of what is possible in the realm of autonomous flight.

5. Conclusion

The ongoing advancements in UAV technology and their expanding applications make it imperative to develop sophisticated simulation environments that can keep pace with these developments. The relevance and expediency of this research are underscored by the critical need for reliable and versatile simulation tools that can support innovative UAV applications in various sectors.

This review has identified a primary unsolved **problem**: the absence of a universal UAV simulator that meets the diverse and specific needs of UAV research. Current simulators exhibit varying strengths and weaknesses, with no single tool encompassing all necessary features such as high-fidelity physics, diverse environmental variability, comprehensive sensor simulation, and scalability for swarm operations. This gap not only limits the potential for research and development but also impedes the application of UAV technology in sectors where precision, reliability, and efficiency are paramount.

The **goal** of future research should be to develop a comprehensive, high-fidelity, and scalable UAV simulation environment that addresses these identified gaps. The research will focus on integrating advanced sensor simulation capabilities, improving the modeling of swarm dynamics, and enhancing user accessibility and support infrastructure. The **object** of the research will be UAV simulators, and the **subject** will be their development and optimization to meet diverse research and operational needs.

From this vantage point, the task for scientific research becomes clear: to significantly advance the state of drone simulation technology, enabling it to more accurately and effectively model the behavior of a heterogeneous swarm of drones navigating and operating within dynamically changing environments. To achieve this task, it is necessary to improve scalability for swarm simulations by utilizing distributed computing solutions to manage and simulate large-scale drone swarms efficiently, and to incorporate machine learning and AI by integrating models that can learn and adapt to dynamic environments within the simulation.

Future research should focus on developing simulators with advanced aerodynamics capabilities that can accurately replicate the complex interactions between drones and their operating environments. This includes the effects of wind, turbulence, and varying weather conditions on drone flight dynamics. Also, there's a pressing need to enrich simulators with more diverse sensor models, especially those that support emerging sensor technologies in UAV applications, such as multispectral and hyperspectral imagers, advanced radar systems, and thermal cameras. Investigating distributed computing or cloud-based solutions could provide the necessary infrastructure to simulate complex, coordinated drone behaviors across vast geographical areas, marking a substantial leap forward in the realism and applicability of UAV simulations.

References

[1] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2022, doi: 10.1109/TRO.2021.3084395.

[2] "Autonomous Firefighting Drone For High-Rise Fires in China," *Intelligent Living*. Available: https://www.intelligentliving.co/autonomous-fire-fighting-drone/. [Accessed: 21-Mar-2024].

[3] S. A. H. Mohsan, *et al.*, "Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends," *Intelligent Service Robotics*, vol. 16, pp. 109–137, 2023, doi: https://doi.org/10.1007/s11370-022-00452-4.

[4] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human Interaction With Robot Swarms: A Survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2016, doi: 10.1109/THMS.2015.2480801.

[5] Y. Wang, T.-J. Zhang, Z. Cai, J. Zhao, and K. Wu, "Multi-UAV coordination control by chaotic grey wolf optimization based distributed MPC with event-triggered strategy," *Chinese Journal of Aeronautics*, vol. 33, pp. 2877-2897, 2020, doi: 10.1016/j.cja.2020.04.028.

[6] C. K. Liu and D. Negrut, "The Role of Physics-Based Simulators in Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 35–58, 2021, doi: 10.1146/annurev-control-072220-093055.

[7] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A Review of Physics Simulators for Robotic Applications," *IEEE Access*, vol. 9, pp. 51416–51431, 2021, doi: 10.1109/ACCESS.2021.3068769.

[8] J. Saunders, S. Saeedi, and W. Li, "Autonomous aerial robotics for package delivery: A technical review," *Journal of Field Robotics*, vol. 40, no. 1, pp. 3–49, 2023, doi: 10.1002/rob.22231.

[9] A. Mairaj, A. I. Baba, and A. Y. Javaid, "Application specific drone simulators: Recent advances and challenges," *Simulation Modelling Practice and Theory*, vol. 94, pp. 100–117, 2019, doi: 10.1016/j.simpat.2019.01.004.

[10] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, Springer, 2018, pp. 621–635, doi: 10.1007/978-3-319-67361-5_40.

[11] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Proc. Conference on Robot Learning (CoRL)*, PMLR, 2021, pp. 1147–1157, doi: 10.48550/arXiv.2009.00563.

[12] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2149–2154, Sep. 2004, doi: 10.1109/IROS.2004.1389727.

[13] O. Michel, "Cyberbotics Ltd. Webots: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004, doi: 10.5772/5618.

[14] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS — a modular gazebo MAV simulator framework," in *Robot Operating System (ROS): The Complete Reference (Volume 1)*, Springer, 2016, pp. 595–625, doi: 10.1007/978-3-319-26054-9_23.

[15] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *Proc. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 6941–6948, doi: 10.1109/IROS40897.2019.8968116.

[16] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, "Learning to fly—a gym environment with PyBullet physics for reinforcement learning of multi-agent quadcopter control," in *Proc. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 7512–7519, doi: 10.1109/IROS51168.2021.9635857.

[17] J. Saunders, S. Saeedi, and W. Li, "Parallel Reinforcement Learning Simulation for Visual Quadrotor Navigation," in *Proc. 2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 1357–1363, doi: 10.1109/ICRA48891.2023.10160675.

[18] A. Suarez, G. Heredia, and A. Ollero, "Compliant Bimanual Aerial Manipulation: Standard and Long Reach Configurations," *IEEE Access*, vol. 8, pp. 88844–88865, 2020, doi: 10.1109/ACCESS.2020.2993101.

[19] C. A. Dimmig, *et al*., "A Small Form Factor Aerial Research Vehicle for Pick-and-Place Tasks with Onboard Real-Time Object Detection and Visual Odometry," in *Proc. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 6289–6296, doi: 10.1109/IROS55552.2023.10342295.

[20] G. Silano, E. Aucone, and L. Iannelli, "CrazyS: A Software-In-The-Loop Platform for the Crazyflie 2.0 Nano-Quadcopter," in *Proc. 2018 26th Mediterranean Conference on Control and Automation (MED)*, Zadar, Croatia, 2018, pp. 1-6, doi: 10.1109/MED.2018.8442759.

[21] "Gazebo Classic Simulation | PX4 User Guide (main). Redirecting to latest version of document (master)," *PX4 User Guide*. Available: https://docs.px4.io/main/en/sim_gazebo_classic/. [Accessed: 24-Mar-2024].

[22] "Gazebo," *Gazebo*. Available: https://gazebosim.org/. [Accessed: 24-Mar-2024].

[23] "Isaac Sim," *NVIDIA Developer*. Available: https://developer.nvidia.com/isaac-sim. [Accessed: 24-Mar-2024].

[24] M. Jacinto, J. Patrikar, J. Pinto, J. Keller, R. Cunha, and S. Scherer, "Pegasus Simulator: An Isaac Sim Framework for Multiple Aerial Vehicles Simulation," *arXiv preprint*, arXiv:2307.05263, 2023.

[25] V. Makoviychuk, et al., "Isaac Gym: High performance GPU-based physics simulation for robot learning," *arXiv preprint*, arXiv:2108.10470, 2021.

[26] M. Kulkarni, T. J. L. Forgaard, and K. Alexis, "Aerial Gym–Isaac Gym Simulator for Aerial Robots," *arXiv preprint*, arXiv:2305.16510, 2023.

[27] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4397–4404, doi: 10.1109/ICRA.2015.7139807.

[28] X. Gu, N. Shi, C. Liu, and Y. Lin, "Design and Dynamics Simulation of a Triphibious Robot in Webots Environment," in *Proc. 2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2021, pp. 1268–1273, doi: 10.1109/ICMA52036.2021.9512574.

[29] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 1321–1326, doi: 10.1109/IROS.2013.6696520.

[30] P. Udvardy, B. Beszedes, B. Toth, A. Foldi, and A. Botos, "Simulation of obstacle avoidance of an UAV," in *Proc. 2020 New Trends in Aviation Development (NTAD)*, Sep. 2020, pp. 245–249, doi: 10.1109/NTAD51447.2020.9379113.

[31] GitHub - PegasusSimulator/PegasusSimulator: A framework built on top of NVIDIA Isaac Sim for simulating drones with PX4 support and much more, GitHub. Available: https://github.com/PegasusSimulator/PegasusSimulator. [Accessed: 21-Mar-2024].

[32] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 5026–5033, doi: 10.1109/IROS.2012.6386109.

[33] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Flightmare: A Flexible Quadrotor Simulator," in *Proc. 2021 Conference on Robot Learning (CoRL)*, PMLR, 2021, pp. 1147–1157, doi: 10.5167/uzh-193792.

[34] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, vol. 8, no. 82, 2023, doi: 10.1126/scirobotics.adg1462.

[35] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: A Modular Framework for Photorealistic Camera, Exteroceptive Sensor, and Dynamics Simulation," in *Proc. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, doi: 10.1109/IROS40897.2019.8968116.

[36] GitHub - microsoft/AirSim: Open source simulator for autonomous vehicles built on Unreal Engine / Unity, from Microsoft AI & Research, GitHub. Available: https://github.com/Microsoft/AirSim. [Accessed: 21-Mar-2024].

[37] E. Coumans, *et al*.,, "PyBullet, a python module for physics simulation for games, robotics and machine learning," 2016. Available: https://pybullet.org/.

[38] T. Mark, *et al*., "Gymnasium," Zenodo. Available: https://zenodo.org/record/8127025. [Accessed: 24-Mar-2024].

[39] A. Raffin, A. Hill, M. G. Kanervisto, and N. Ernestus, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 1, 2021. Available: https://www.jmlr.org/papers/v22/20-1364.html.

[40] G. Li, X. Liu and G. Loianno, "RotorTM: A Flexible Simulator for Aerial Transportation and Manipulation," in *IEEE Transactions on Robotics*, vol. 40, pp. 831-850, 2024, doi: 10.1109/TRO.2023.3336320.

[41] "UAV Toolbox," MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink. Available: https://www.mathworks.com/products/uav.html. [Accessed: 24-Mar-2024].

[42] N. Horri and M. Pietraszko, "A Tutorial and Review on Flight Control Co-Simulation Using Matlab/Simulink and Flight Simulators," *Automation*, vol. 3, no. 3, pp. 486-510, 2022. DOI: 10.3390/automation3030025.

[43] Z. Yuan *et al*., "Safe-Control-Gym: A Unified Benchmark Suite for Safe Learning-Based Control and Reinforcement Learning in Robotics," in *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11142-11149, Oct. 2022, doi: 10.1109/LRA.2022.3196132.

[44] F. Kong, X. Liu, B. Tang, J. Lin, Y. Ren, Y. Cai, F. Zhu, N. Chen, and F. Zhang, "MARSIM: A Light-Weight Point-Realistic Simulator for LiDAR-Based UAVs," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2954-2961, 2023. DOI: 10.1109/LRA.2023.3264163.

[45] Z. Huang, *et al.*, "QuadSwarm: A Modular Multi-Quadrotor Simulator for Deep Reinforcement Learning with Direct Thrust Control," *arXiv*, 2023. DOI: 10.48550/arXiv.2306.09537.

[46] E. Bøhn, E. M. Coates, S. Moe and T. A. Johansen, "Deep Reinforcement Learning Attitude Control of Fixed-Wing UAVs Using Proximal Policy optimization," *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 2019, pp. 523-533, doi: 10.1109/ICUAS.2019.8798254.

[47] A. Keipour, et al., "UAS Simulator for Modeling, Analysis and Control in Free Flight and Physical Interaction," in *AIAA Science and Technology Forum*, 2023, p. 1279. DOI: 10.2514/6.2023-1279.

[48] E. Cuniato, J. Cacace, M. Selvaggio, F. Ruggiero and V. Lippiello, "A hardware-in-the-loop simulator for physical human-aerial manipulator cooperation," *2021 20th International Conference on Advanced Robotics (ICAR)*, Ljubljana, Slovenia, 2021, pp. 830-835, doi: 10.1109/ICAR53236.2021.9659398.

[49] S. Folk, et al., "RotorPy: A Python-based Multirotor Simulator with Aerodynamics for Education and Research," *arXiv*, 2023. DOI: 10.48550/arXiv.2306.04485.

[50] J. Li, et al., "Potato: A Data-Oriented Programming 3D Simulator for Large-Scale Heterogeneous Swarm Robotics," *arXiv preprint arXiv:2308.12698*, 2023. DOI: 10.48550/arXiv.2308.12698.

[51] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, 2022. DOI: 10.1126/scirobotics.abl6259.

[52] T. Baca, et al., "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*, vol. 102, no. 1, p. 26, 2021. DOI: 10.1007/s10846-021-01383-5.

[53] L. Pichierri, A. Testa and G. Notarstefano, "CrazyChoir: Flying Swarms of Crazyflie Quadrotors in ROS 2," in *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4713-4720, Aug. 2023, doi: 10.1109/LRA.2023.3286814.

[54] J. A. Preiss, W. Honig, G. S. Sukhatme and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 3299-3304, doi: 10.1109/ICRA.2017.7989376.

[55] M. Fernandez-Cortizas, et al., "Aerostack2: A Software Framework for Developing Multi-robot Aerial Systems," *arXiv*, 2023. DOI: 10.48550/arXiv.2303.18237.

[56] "Flight Simulator | X-Plane 12: Flight Simulation Done Right," X-Plane. [Online]. Available: https://www.x-plane.com/. [Accessed: 24-Mar-2024].

[57] I. Navarro, J. Patrikar, J. P. A. Dantas, R. Baijal, I. Higgins, S. Scherer, and J. Oh, "SoRTS: Learned Tree Search for Long Horizon Social Robot Navigation," *arXiv*, 2023. DOI: 10.48550/arXiv.2309.13144.

[58] D. J. Richter and R. A. Calix, "QPlane: An Open-Source Reinforcement Learning Toolkit for Fixed Wing Aircraft Simulation," in *Proc. 12th ACM Multimedia Systems Conf. (MMSys '21)*, Istanbul, Turkey, Sept. 28-Oct. 1, 2021, pp. 261-266. DOI: 10.1145/3458305.3478446.

[59] A. R. Perry, "The FlightGear Flight Simulator," in *Proc. Annu. Tech. Conf. (ATEC '04)*, 2004, p. 31.

[60] Y. A. Prabowo, B. R. Trilaksono and F. R. Triputra, "Hardware in-the-loop simulation for visual servoing of fixed wing UAV," *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, Denpasar, Indonesia, 2015, pp. 247-252, doi: 10.1109/ICEEI.2015.7352505.

[61] "RealFlight RC Flight Simulator Software and Accessories," RealFlight. [Online]. Available: https://www.realflight.com/. [Accessed: 24-Mar-2024].

[62] S. J. Carlson and C. Papachristos, "The MiniHawk-VTOL: Design, Modeling, and Experiments of a Rapidly-prototyped Tiltrotor UAV," *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, 2021, pp. 777-786, doi: 10.1109/ICUAS51884.2021.9476731.

УДК 351.861 https://doi.org/10.20535/2786-8729.4.2024.300614

ПОРІВНЯЛЬНИЙ ОГЛЯД СИМУЛЯТОРІВ ДРОНІВ

Микола Ніколаєв
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0009-0003-0787-6726

Михайло Новотарський
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-5653-8518

Швидкий розвиток безпілотних літальних апаратів (БПЛА), особливо дронів, революціонізував різні сфери науки та промисловості, включаючи сільське господарство, картографію, пошуково-рятувальні операції та інші. Для відпрацювання алгоритмів реалізації складних еволюцій траєкторій при доставці посилок або при екологічному моніторингу виникає нагальна необхідність розробки середовищ моделювання для уникнення значних ризиків, пов'язаних з реальним тестуванням. Однак різноманітність і фрагментарність доступних інструментів ускладнює вибір симуляторів для вирішення конкретних завдань. Розробники змушені балансувати між такими компромісами, як швидкість моделювання, точність імітації фізичних законів, інтеграція сенсорів і якість інтерфейсу користувача. Відсутність універсального симулятора, який би включав високоточну фізику, комплексне моделювання сенсорів та масштабованість для моделювання рою дронів, є сьогодні певною проблемою. Відомі симулятори БПЛА мають певні переваги та недоліки, але жоден з них не забезпечує комплексного вирішення всіх вимог, які необхідні для сучасних досліджень і розробок. Інтеграція різних сенсорів, таких як веб-камери, LiDAR, GPS та IMU, у системи моделювання залишається технічною проблемою, що обмежує застосовність існуючих симуляторів. Крім того, доступність ефективних симуляторів і підтримка може значно відрізнятися, що також впливає на вибір та стійкість цих інструментів. Стандартизований підхід до моделювання БПЛА може підвищити ефективність порівняння результатів досліджень, спростити зусилля при виборі та створити єдину основу для оцінки продуктивності симуляторів. Прогрес в моделюванні аеродинаміки, особливо для квадрокоптерів та БПЛА, може покращити точність і реалістичність моделювання, що краще підтримуватиме розвиток передових технологій. Майбутні дослідження мають на меті розробку більш комплексних, високоточних та масштабованих середовищ для моделювання. Це включає інтеграцію інноваційних підходів моделювання сенсорів, покращення моделювання динаміки роїв і підвищення доступності та підтримки користувачів. Основні напрямки для покращення включають: інтеграцію сенсорів для моделювання їх широкого спектру, підвищення ефективності моделювання динаміки роїв у випадках складної поведінки та взаємодії між декількома дронами, спрощення інтерфейсів користувача, надання всебічної документації, забезпечення надійної підтримки спільноти, розробку стандартизованих критеріїв для порівняння та оцінки різних симуляторів, а також урахування детальних аеродинамічних принципів для покращення точності моделювання. Вирішення цих проблем при розробці симуляторів БПЛА є важливим для розвитку аероробототехніки. Таким чином, розробка середовищ для моделювання з інтегрованими можливостями сенсорів, покращеним моделюванням динаміки роїв та зручними інтерфейсами може підвищити ефективність та результативність розвитку БПЛА. Стандартизовані критерії оцінки та детальне моделювання аеродинаміки підтримуватимуть еволюцію технологій БПЛА, забезпечуючи більш безпечні, надійні та інноваційні застосування в різних сферах. Вони сприятимуть інноваціям, технологічному прогресу та операційній ефективності у реальних умовах.

Ключові слова: дрон, БПЛА, рій, симулятор, сенсор, динаміка середовища