Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.078545 Published Online: 20 March 2026

REVIEW

A Survey of Pixhawk/PX4 Autopilot and Its Impact on Research and Education

Nourdine Aliane*

Department of Engineering, Universidad Europea de Madrid, Villaviciosa de Odón, Madrid, Spain

*Corresponding Author: Nourdine Aliane. Email: nourdine.aliane@universidadeuropea.es

Received: 03 January 2026; Accepted: 10 March 2026

ABSTRACT: The rapid advancement of unmanned aerial vehicle (UAV) technologies has increased demand for flexible autopilot platforms suitable for both research and education. Among available options, the open-source Pixhawk/PX4 autopilot has emerged as a leading solution due to its modular architecture and robust software ecosystem. This survey examines the adoption of the Pixhawk/PX4 platform in research and education. The survey covers the analysis of the Pixhawk/PX4 autopilot software development APIs, its compatibility with ROS middleware and MATLAB/Simulink environments, and environments for software/hardware-in-the-loop simulations. Additionally, it explores the integration of Cutting-Edge technologies to enhance UAVs performance. By synthesizing these findings, this survey presents a holistic map of the Pixhawk/PX4 ecosystem, and it serves as a multifaceted resource, providing its diverse audience with critical insights to support informed tool selection for UAV development projects.

KEYWORDS: Unmanned aerial vehicle (UAV); autopilot; flight controller; Pixhawk/PX4; MAVLink; MATLAB/Simulink support package; software-in-the-loop; hardware-in-the-loop; ROS

1 Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as a transformative technology with applications spanning diverse fields [1]. They vary in size, from small consumer drones to large military-grade systems, and can be equipped with cameras, sensors, and other payloads to perform specific tasks. UAVs are typically controlled by an onboard flight controller, a compact embedded computer running software that provides necessary control and guidance.

Over the past two decades, a variety of autopilots have been developed as plug-and-play systems, catering to users who do not need extensive knowledge of internal UAV's working. At the same time, open-source initiatives have significantly contributed to democratizing access to UAV systems. These efforts have allowed developers to gain a thorough understanding of the systems' internal mechanisms and customize their designs, while also empowering researchers to experiment, prototype, and deploy UAV solutions with enhanced flexibility [2]. This context has led to the development of several open-source flight controllers, including Paparazzi [3], ArduPilot [4], Pixhawk/PX4 [5,6], LibrePilot [7], Betaflight [8], and iNAV [9]. Each platform offers distinct features and capabilities tailored to different applications and requirements.

While these mentioned open-source autopilots serve the UAV community, Pixhawk/PX4 has emerged as a preferred platform for research and education due to its distinctive balance of modularity, tool integration, and ecosystem support [2,10]. Specialized platforms such as Betaflight and iNAV, which are optimized for specific racing and navigation, respectively, lack general purpose and extensibility required for research. Compared to ArduPilot, which offers broad vehicle support but a more monolithic architecture,

Copyright © 2026 The Author. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

PX4 provides a highly modular facilitating customization and easier integration of novel algorithms [11], furthermore, PX4 offers seamless integration with MATLAB/Simulink using the dedicated Pixhawk Support Package toolbox [12,13], and a suite of Software-in-the-loop (SITL) and Hardware-in-the-loop (HITL) simulators, such as Gazebo, AirSim, jMAVSim. Furthermore, PX4 benefits from the backing of the Dronecode Foundation [14] and a large industrial academic community, ensuring ongoing development, comprehensive documentation, and reproducible hardware standards. These attributes make Pixhawk/PX4 ecosystem suited for prototyping, experimentation and education.

The rapid evolution of the Pixhawk/PX4 ecosystem has resulted in a vast and often fragmented body of knowledge, creating a significant barrier for both newcomers and experienced practitioners seeking to leverage these powerful open-source technologies. This survey analyzes the impact of the Pixhawk/PX4 autopilot on research and education, focusing on the platform's capabilities. The analysis covers the available APIs for software development and the platform's compatibility with Robotic Operating System (ROS) middleware as well as with MATLAB/Simulink environments. Additionally, it examines supported environments for SITL and HITL simulations. Finally, the survey investigates the integration of cutting-edge technologies to enhance UAV capabilities.

By synthesizing and critically analyzing these dispersed findings, this survey constructs a holistic map of the Pixhawk/PX4 ecosystem, serving as a multifaceted resource for its diverse audience. For researchers and developers, the document functions as a comprehensive directory of tools and architectures, empowering them to make informed, strategic decisions when selecting components for specific unmanned aerial vehicle (UAV) projects. Concurrently, educators will find valuable, structured insights that can be directly translated into curricula, guiding the design of effective hands-on laboratory activities in autonomous systems and related fields.

1.1 Related Work

Several recent reviews and survey studies have comprehensively explored a wide range of topics related to UAV autopilots and flight controllers, including foundational hardware and software architectures, open-source ecosystems, and the role of simulators in testing and validation.

In terms of UAV components and system architecture, the work in [11] presents a state-of-the-art survey of multirotor drones covering drone types, construction methods, and the underlying technologies. Similarly, reference [15] provides an interdisciplinary overview of UAV hardware and software architecture and their applications, while references [16,17] specifically focus on flight control systems.

A number of studies address guidance, navigation, and control in UAV systems. For instance, reference [18] discusses techniques for autopilot integration in small-unmanned aircraft systems (UAS), reference [19] emphasizes obstacle avoidance strategies, and reference [20] reviews estimation techniques across different flight phases. The reliability of UAV computing platforms is examined in [21], which analyzes fault and failure modes in mission-critical applications.

The role of open-source ecosystems and operating systems is another key consideration in UAV autopilot development. For example, reference [2] surveys open-source drone platforms, including hardware, software, and simulation environments, while reference [22] reviews open-source autopilots used in drones and compares common platforms. Further, reference [10] provides an overview of open-source UAV autopilot platforms, and reference [23] discusses these platforms from the perspective of real-time and operating systems, offering insights into their strengths and limitations.

Finally, simulation and co-simulation methodologies are recognized as critical tools for enabling safe and cost-effective UAV development. The work in [24] reviews current UAV simulators, while reference [25]

advances this area by presenting step-by-step implementations of real-time co-simulations between Simulink and flight simulators such as X-Plane and Flight Gear, demonstrating model validation and controller testing in both SITL and HITL configurations.

All these overview and survey papers provide a nuanced understanding of the UAV landscape, highlighting advancements in flight controller architecture design, the influence of open-source ecosystems, and the essential role of simulators in testing and validation.

1.2 Motivation and Contribution

Despite the extensive survey literature on UAV autopilots, existing reviews show significant gaps in their treatment of the Pixhawk/PX4 ecosystem. A primary limitation is the fragmented focus of prior work, which typically addresses specific aspects, such as software architecture, guidance and control algorithms, or open-source development, in isolation, rather than evaluating the synergistic potential of the integrated Pixhawk/PX4 platform. Furthermore, there is an evident absence of synthesis concerning the practical integration of this ecosystem with complementary development tools and emerging technologies.

To address these gaps, the primary objective of this paper is to analyze the multifaceted capabilities enabled by the Pixhawk/PX4 ecosystem for developing advanced UAV applications. Specifically, this survey examines scientific literature to aggregate evidence on: (1) the integration of companion computers for onboard high-level processing; (2) the integration with ROS middleware; (3) the combination with MATLAB/Simulink for advanced modeling and control design; (4) the deployment of SITL and HITL environments for testing and validation; (5) the incorporation of cutting-edge technologies, including machine learning (ML) and artificial intelligence (AI) models as well as cybersecurity protocols. In addition to these technical dimensions, this survey also investigates the diverse contexts in which the Pixhawk/PX4 platform has been employed in educational context.

By mapping these convergent areas, this survey aims to provide researchers and educators with a holistic reference for leveraging the Pixhawk/PX4 ecosystem in the development of autonomous systems.

1.3 Survey Organization

The remainder of this survey is organized as follows. Section 2 outlines the materials and methods guiding this review. Section 3 provides a technical overview of the Pixhawk/PX4 platform, detailing its key hardware and software features. In Section 4, we analyze the scientific literature on utilizing the Pixhawk/PX4 ecosystem for UAV applications. Section 5 shifts focus on discussing its role in educational context. Section 6 synthesizes the main findings and highlights critical issues. Finally, Section 7 summarizes the main conclusions.

2 Materials and Methods

This survey examines, through a literature review process, the adoption of the Pixhawk/PX4 platform in research and education. The study evaluates the platform's technical capabilities, focusing on tools and frameworks for UAV application development. Through an analysis of recent publications, we analyze the platform's scope, key features, and practical implementations. The review specifically addresses the following key topics:

1. Companion Computers.
2. DroneKit and MAVSDK APIs.
3. ROS Integration.
4. MATLAB SPP Toolbox.

5. SITL/HITL Simulations Environment.
6. Cutting-Edge Tech and Security.
7. Educational Use.

The search process was conducted across publicly accessible academic databases, including IEEE Xplore, ScienceDirect, MDPI, Springer, etc. The survey focuses on contributions published between 2018 and 2026. The paper selection process was guided by the PRISMA guidelines, and its workflow is summarized in Fig. 1. The search strategy employed specific inclusion criteria, utilizing the primary keywords "Pixhawk" or "PX4" in conjunction with filtering terms such as "ROS", "MATLAB", "SITL", and "HITL" using the AND Boolean operator. The initial screening encompassed peer-reviewed articles, reviews, surveys, and conference papers covering a wide range of UAV research topics. This process yielded an initial pool of 138 papers. After their screening, approximately 17% of these publications were excluded. The main reason for rejection was a superficial mention of the Pixhawk/PX4 and the publications are without any substantive utilization or discussion of its features.

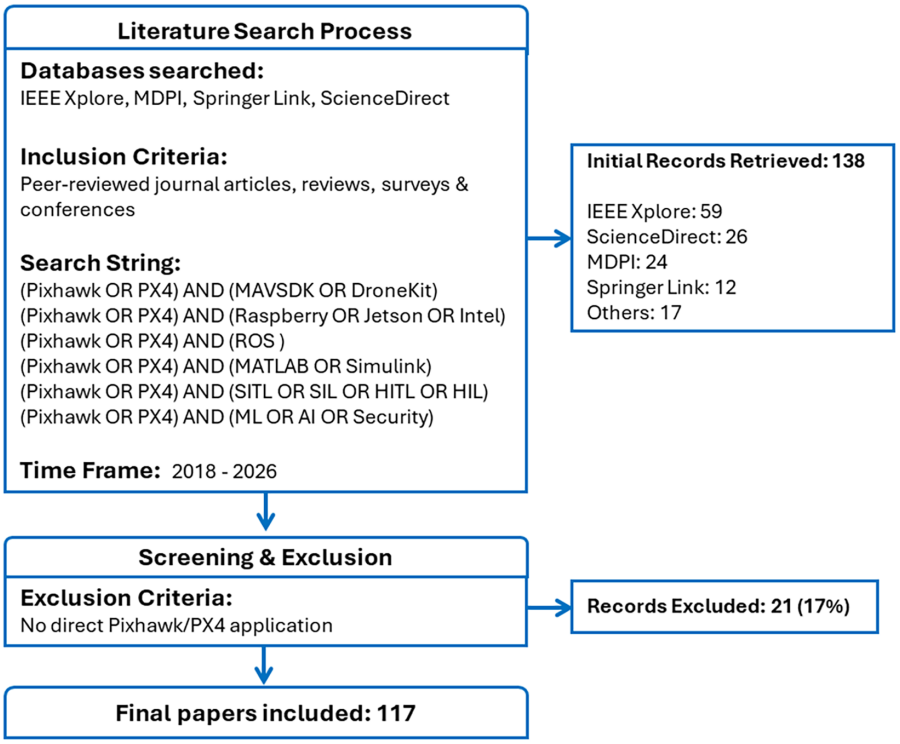

Figure 1: Review selection process flowchart.

While this survey does not claim to be a systematic review, it examines a representative selection of publications that effectively showcase the diverse frameworks associated with the Pixhawk/PX4 platform. This approach provides valuable insights into how the platform is being adopted in research and education.

The remaining 117 publications were subsequently categorized into the six predefined development topics for detailed review and discussion. This categorization was performed by first screening the abstracts and, where the focus was unclear, by reviewing the full text of the papers. In cases where an article's contributions spanned multiple categories due to overlapping themes, it was assigned to the single most relevant category to prevent duplication in the analysis. In Section 4, the scope of these development tools

will be explored, outlining their key features, and discussing their applications based on recent scientific literature addressing specific UAV challenges.

3 Technical Overview

First, it is important to signal that Pixhawk is not a specific product, nor a manufacturer. Pixhawk is a set of open standards supported by some semiconductor manufacturers, software companies and drone engineering companies. These standards cover the requirements for the autopilot hardware design. The Pixhawk/PX4 is a versatile controller designed to guide various types of vehicles such as drones, ground rovers, fixed wing aircraft, and underwater vehicles. It was initially released in November 2013 as a collaborative project between L. Meier from ETH (Zurich) and 3D Robotics [26], before becoming independent open-source hardware and software. Pixhawk/PX4 is a set of open standards supported by some semiconductor manufacturers, software companies and drone engineering companies. The Pixhawk/PX4 adheres to a set of open standards supported by semiconductor manufacturers, software developers, and drone engineering firms. The Pixhawk/PX4 autopilot consists of four main components: the Pixhawk hardware, PX4 software, MAVLink (a lightweight communication protocol), and Ground Control Station software, which are elaborated upon in the following paragraphs.

3.1 Pixhawk Hardware

The Pixhawk hardware usually comes as a single board autopilot consisting of two main units: the flight management unit (PX-FMU) and the input-output board (PX4-IO). The FMU units are based on 32-bit ARM microcontrollers running the PX4 flight controller, and feature backup failsafe 32-bit co-processors to enable recovery from a primary system failure. All platforms provide several PWM outputs for servos control and other uses and come with some internal sensors including a three-axis gyroscope, a three-axis accelerometer, a three-axis magnetometer, a barometric pressure sensor, a global navigation satellite system (GNSS), a voltage sensor for battery under voltage detection, a built-in safety switch, etc. In addition, it provides sockets for direct connection with external sensors and hardware such as GPS and compass, a buzzer, and an I2C bus for connecting to four external sensors. Finally, it offers connectivity with peripherals through UART and CAN buses. The flight controller (FC) can be operated through a radio control unit or remotely via a ground control station using the MAVLink protocol [27]. Fig. 2 illustrates the interface architecture of the Pixhawk and its associated peripherals.

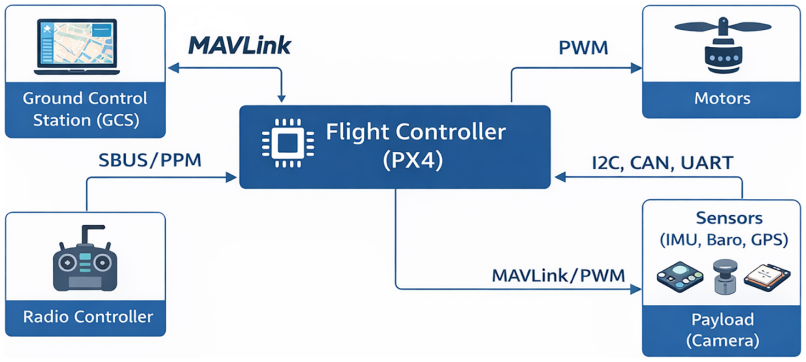

Figure 2: Pixhawk interface diagram.

Pixhawk autopilots come in various versions, differing in hardware specifications such as processor speed, PWM channel count, and IMU redundancy, providing flexibility for diverse applications. For example,

the high-end Pixhawk 6X features an STM32H753 processor, triple redundant IMUs, and dual barometers, making it ideal for demanding use cases. Meanwhile, older models like the Pixhawk-1 and Pixhawk-2 have been discontinued. In addition to official Pixhawk units, several third-party autopilots are derived from the platform, offering enhanced features or cost-effective alternatives. Examples include MindRacer, Cube Orange, and Kakute. Advanced configurations are also available, such as integrated boards combining a Pixhawk 5X/6X with a Raspberry Pi CM4 [28] or a Pixhawk 6X paired with an NVIDIA Jetson Orin in a single unit [29]. Additionally, Auterion SkyNode, launched in 2021, is an enterprise-grade Pixhawk/PX4 designed for professional and industrial drones [30]. It supports optional companion computers like the Raspberry Pi CM4 or i.MX 8M, enabling edge computing, cloud integration, and enhanced security features. Table 1 shows some versions developed according to Pixhawk standards, and Fig. 3 illustrates the appearance of select Pixhawk models.

Table 1: Main standard Pixhawk versions.

Model	Manufact	Year	MCU	Freq: Mhz	FMU	PWM	IMU
Pixhawk-1	mRo	2013	STM32F427, M4	168	v2	14	1
Pixhawk-2: Cube Black	CubePilot	2016	STM32F427, M4	168	v3	14	3
Pixhawk-2, Pixracer	mRo	2015	STM32F437, M4	180	v4	–	1
Pixhawk-3 Pro	Drotek	2017	STM32F469, M4	168	v4pr	14	3
Pixhawk-4	Holybro	2018	STM32F765, M7	216	v5	16	3
Pixhawk-4 Mini	Holybro	2018	STM32F765, M7	216	v5	12	3
Pixhawk-5X	Holybro	2019	STM32F765, M7	216	v5x	16	3
Pixhawk-6C	Holybro	2022	STM32H743, M7	480	v6c	16	3
Pixhawk-6X	Holybro	2022	STM32H753, M7	480	v6x	16	3
Pixhawk-6X, CUAV	CUAV	2022	STM32H753, M7	480	v6x	16	3

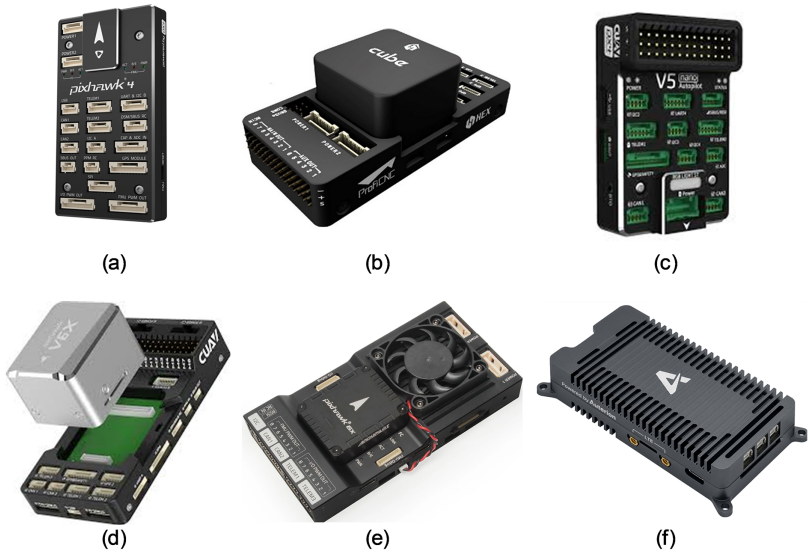

Figure 3: Some Pixhawk models: (**a**) Pixhawk 4, (**b**) Pixhawk Cube, (**c**) Pixhawk-5, 120, (**d**) Pixhawk CUAV V6X, (**e**) Pixhawk 6X + RPi on CM4, (**f**) Skynode.

The progression of Pixhawk hardware, as summarized in table-1, has significantly enhanced its suitability for research and education. The shift from STM32F4 to more powerful STM32H7 processors, along with increased clock speeds, enables the execution of computationally intensive algorithms, such as advanced filtering, real-time sensor fusion, and lightweight machine learning, directly on the autopilot, reducing reliance on external companion computers. The introduction of features like triple redundant IMUs and dual barometers improve fault tolerance and safety, which is critical for experimental reliability and student projects. Expanded connectivity (e.g., more PWM outputs, CAN bus, and Ethernet) supports the integration of diverse sensors, custom payloads, and companion computers, facilitating scalable and interdisciplinary UAV platforms. This ongoing hardware evolution, coupled with strong standardization, has made Pixhawk a reproducible, accessible, and professional-grade tool that lowers the barrier to advanced experimentation while supporting innovation in both academic and applied research contexts.

3.2 PX4 Software

PX4 is the open-source flight control software that operates on Pixhawk hardware. It runs on NuttX [31], a POSIX-compliant real-time operating system (RTOS) that enables multi-threading and advanced scheduling. The software is written in C/C++ and consists of two primary components: the flight control stack and the middleware, which handles internal communications [26].

The flight control stack is the core of PX4, responsible for navigation and control algorithms used for various drones, including multirotor, fixed wing, and vertical takeoff and landing (VTOL) airframes. It supports various flight modes, such as manual, altitude hold, and position hold. At its core, the stack manages body position, velocity, and attitude (yaw, pitch, roll) using independent PID controllers. Additionally, PX4 also employs an extended Kalman filter (EKF) to fuse data from IMUs, GPS, compasses, air speed sensors, and barometer, estimating position. The EKF algorithm serves two key functions: estimating unmeasurable states like position and velocity and refining angular orientation by filtering sensor noise.

The middleware layer acts as a bridge between the flight control stack and hardware, facilitating communication with external devices like GCS and companion computers. It consists of three main components. The micro–Object Request Broker (uORB) [32], a messaging system enabling real-time data exchange between modules using a publish-subscribe pattern and relies on shared memory. The second consists of parameter subsystem, a data table for storing and retrieving system parameters such as vehicle configurations, PID values, and mission waypoints. PX4 provides users with C/C++ callback functions to listen to changes in uORB topics associated with parameter up-dates. The third component is a hardware drivers' layer, managing system initialization, configuration, and low-level interactions like data acquisition and actuator control. This middleware allows developers to focus on high-level control algorithms without delving into hardware specifics. Finally, PX4 supports custom module integration, enabling advanced users to extend functionality by embedding C/C++ modules into the firmware. Fig. 4 illustrates the PX4 software architecture.

3.3 Ground Control Stations

A Ground Control Station (GCS) is a computer or mobile device that enables remote monitoring and control of an unmanned aerial vehicle through wireless serial communication. GCS allows operators to plan and execute autonomous missions, receive real-time telemetry data, issue flight commands, and in some implementations, process collected data and imagery. Within the Pixhawk/PX4 ecosystem, QGroundControl [33] serves as the primary reference implementation for GCS software. However, the platform maintains compatibility with alternative solutions including Mission Planner [34], MAVProxy [35], among others.

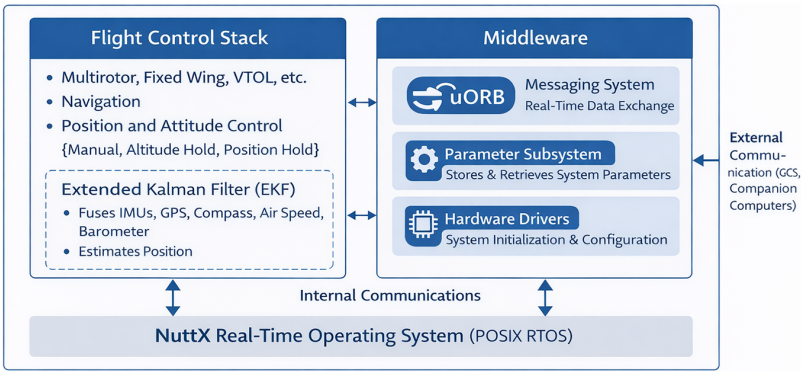

Figure 4: Internal PX4 software architecture.

3.4 MAVLINK Protocol

MAVLink (Micro Air Vehicle Link) [27] is a lightweight communication protocol for exchanging messages between unmanned vehicles and GCSs. Originally developed by Lorenz Meier at ETH Zurich, its latest version, MAVLink 2.0, was released in 2017. The protocol enables bidirectional communication, allowing UAVs to transmit telemetry data (e.g., system state, location, velocity, altitude) to the GCS while receiving commands (e.g., takeoff, landing, mission execution) in return. MAVLink employs a compact binary packet format, with a messages size ranging from 12 to 280 bytes for signed messages. Each packet includes 2 bytes for checksum data integrity and requiring acknowledgment, and it includes 13 bytes signature to ensure tamper-proof communication. Messages are serialized into binary for transmission and deserialized upon receipt, optimizing efficiency in low-bandwidth environments.

MAVLink packets can be transmitted over various communication links, including sub-GHz frequencies (433, 868, 915 MHz), Wi-Fi, serial telemetry, or TCP/IP networks. Pixhawk 5X and later models feature Ethernet support, enabling MAVLink streaming via UDP or TCP over IP networks. A key capability of MAVLink is its routing feature, which facilitates communication between networked systems (e.g., vehicles, GCSs, antenna trackers) and onboard components (e.g., autopilot, cameras, gimbals), each assigned a unique ID number. Tools like MAVLink Inspector allow users to monitor traffic, inspect messages, and visualize real-time data through dynamic charts.

The MAVLink protocol supports a diverse range of units and systems, with PX4 flight controllers serving as the central node for vehicle control and telemetry. The autopilot is often augmented by an onboard companion computer (e.g., Raspberry Pi, NVIDIA Jetson) that handles high-level tasks such as computer vision, path planning, and obstacle avoidance. On the ground, Ground Control Stations (GCS) like QGroundControl or Mission Planner provide the primary human interface for monitoring and command. The protocol also extends to onboard payloads and peripherals. For camera control specifically, MAVLink includes the dedicated Camera Protocol [36], a specialized subset that enables real-time remote adjustment of zoom, focus, shutter speed, aperture, and image or video capture. Cameras must support MAVLink either natively or through a compatible companion computer. Beyond onboard components, the protocol facilitates interoperability with external ecosystems such as ROS/ROS2 and MATlab/Simulink through dedicated bridge implementations. For testing and validation, SITL simulations and HITL rigs communicate via MAVLink to validate system behavior before deployment. Finally, custom applications built with MAVLink SDKs (e.g., DroneKit, MAVSDK) further demonstrate the protocol's versatility, enabling automated and programmable interaction with unmanned systems across a wide range of use cases. Fig. 5 illustrates the interoperability of PX4 Flight Controller through MAVLink protocol.

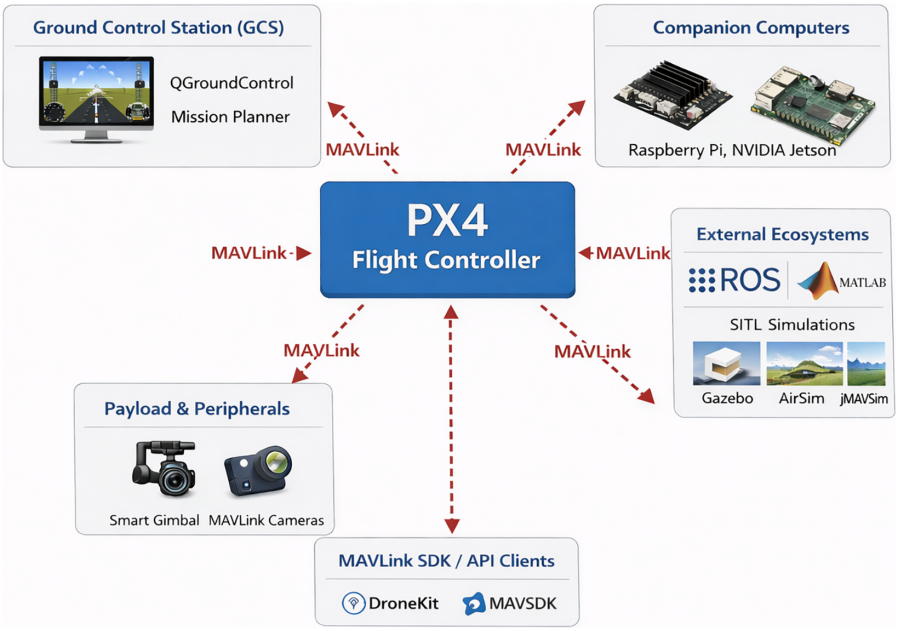

Figure 5: PX4 Flight Controller interoperability through MAVLink-based communication.

4 Tools for UAVs Applications Development

The Pixhawk/PX4 platform supports auxiliary computers to augment computational capabilities for demanding tasks. It also provides a comprehensive suite of tools for UAV application development, including the DroneKit [36] and MAVSDK [37] APIs, compatibility with ROS2 [38], and the MATLAB Support Package for Pixhawk toolbox [13]. Additionally, the platform facilitates testing and validation through SITL/HITL simulation tools [39] and enables the integration of AI-based solutions to enhance UAV performance. This section explores the scope of these tools, outlining their key features, and discussing their applications based on recent scientific literature addressing specific UAV challenges.

4.1 Support for Companion Computers

The Pixhawk/PX4 system supports integration with companion computers (CCs), such as Raspberry Pi, NVIDIA Jetson, or Intel NUC devices, to offload computationally demanding tasks, such as real-time image processing, autonomous navigation, and custom flight mode development. These CCs can be deployed in three configurations: onboard, swappable, or offboard.

In the onboard configuration, the CCs interface with the Pixhawk via serial/USB over MAVLink. Such configuration can be found in works such as in [40], which combines a Raspberry Pi for low-level tasks and an Intel NUC for high-level mission planning. The work [41] proposes a multi-UAV architecture leveraging the PX4 autopilot, where Raspberry Pi and Jetson companion computers are integrated to facilitate task distribution and optimized communication, thereby supporting swarm autonomy. Further, the works [42,43] combines NVIDIA Jetson TX1 and Nano for stereo vision and depth map processing for obstacle avoidance, [44–46] that utilize Raspberry Pi and NVIDIA Jetson for image processing and visual servoing, and [47,48] that utilize Odroid M1 and Intel i7 NUC CCs deploying dynamic flight maneuvers and predictive control.

The swappable configuration employs integrated solutions such as the Pixhawk CM4 module (combining a Pixhawk 5X/6X with a Raspberry Pi CM4) [28] or the Pixhawk 6X paired with an NVIDIA Jetson

Orin in a unified board design [29]. This approach has been used for Time-of-Flight (ToF) positioning [49], implementation of a model predictive control [50,51] demonstrating its ability to handle computationally intensive tasks within compact form factors.

The offboard configuration leverages external computing power without increasing UAV weight, utilizing Wi-Fi or 5G for high-bandwidth communication between the autopilot and CC. In this setup, the CC serves as a bridge between the UAV and the GCS, managing data transmission, telemetry, remote commands, and high-level autonomy algorithms like mission planning or swarm coordination. For instance, reference [52] demonstrates an indoor localization system using a VICON motion capture system with reflective markers, where an offboard companion computer communicates with the UAV's Pixhawk via MAVLink. Similarly, reference [53] enables aggressive UAV maneuvers through predefined waypoints, validated indoors with an OptiTrack system and offboard CC for real-time pose estimation and correction. Further, reference [54] extends this concept to multi-UAV cooperative transport of cable-suspended loads, employing Odroid-XU4 CCs (one per UAV) with centralized trajectory planning via WiFi, tested on three PX4-based UAVs.

4.2 DroneKit and MAVSDK APIs

DroneKit [36] and MAVSDK [37] are powerful APIs that enable advanced UAV control and mission management by interfacing with PX4 autopilots. DroneKit is a Python-based library that simplifies interaction with PX4 via the MAVLink protocol, allowing developers to monitor and control drone operations programmatically. It provides high-level functionalities such as real-time telemetry access, direct vehicle control, and seamless integration with other Python libraries for enhanced data processing. DroneKit can be executed on both GCS and CC. Built in C++, MAVSDK supports multiple programming languages (C++, Python, Swift, Rust, Java), though only its C++ binding allows simultaneous multi-UAV connections; other language wrappers are restricted to single-vehicle operation. A key feature is its offboard mode permitting programmatic control of UAVs by sending MAVLink commands (e.g., position, velocity, or attitude setpoints) from GCS or CC. While DroneKit continues to be utilized in legacy development projects, it is critical to note that it is no longer maintained, which poses a risk for long-term reproducibility and educational sustainability. In contrast, MAVSDK, released in 2018, is the actively maintained and most up-to-date high-level API for PX4, serving as the primary recommended interface for MAVLink-based drone control.

In this context, several articles demonstrate the use of DroneKit for UAVs control and coordination. For example, the papers [55–57] employ DroneKit to implement guided flight modes, multi-UAV cooperation, and obstacle avoidance, with commands transmitted from CC to Pixhawk flight controllers. Innovative approaches in using DroneKit to translate hand gestures [58] and voice commands [59] into UAV maneuvers, offering an alternative to traditional GCS or joystick inputs.

On the other hand, MAVSDK is also adopted for advanced UAV operations and swarm coordination. The work [60] uses MAVSDK's offboard mode to execute precision tasks like target tracking, obstacle avoidance, and surveillance through direct command and waypoint transmission, while reference [61] demonstrates the API capability for precision landing in dynamic environments by updating flight paths based on visual feedback. Swarm-specific applications are explored in [62–64], where MAVSDK enables MAVLink-based inter-vehicle communication for real-time monitoring, navigation, and collaborative mission execution.

While DroneKit offers accessible Python-based scripting for basic UAV automation, MAVSDK provides a robust and future-proof solution for advanced control and swarm applications.

4.3 ROS Integration

Pixhawk/PX4 ecosystem leverages ROS/ROS2 middleware [38], an open-source framework for robotics programming that provides a comprehensive suite of tools and utilities for UAV development, enabling rapid prototyping, simulation, and deployment. Its modular architecture and community support make it ideal for building autonomous UAV systems. Early integration relies on ROS and the MAVROS package, which acts as a bridge translating MAVLink messages into ROS topics and services. However, the current pathway for new developments is ROS2, which connects to PX4 natively via the Micro XRCE-DDS middleware. This shift is not merely nominal; ROS2 offers critical advantages for UAV applications, including a true real-time capable architecture, enhanced security features, and more reliable quality-of-service (QoS) controls for networked and multi-vehicle systems.

ROS2 is typically installed on a companion computer and interfaces with PX4 by reading or writing to internal uORB topics [32]. Beyond core communication features (topics, services, actions) and sensor integration (cameras, IMUs, GPS, LIDAR), ROS2 provides specialized packages for critical UAV functions, such as state estimation (e.g., vision inertial odometry/VIO, extended Kalman filters/EKF for sensor fusion, SLAM for navigation in unknown environments), perception functions (e.g., point cloud libraries/PCL for LIDAR and depth camera data processing), and visualization and debugging (e.g., RViz for real-time monitoring, Rosbag for data logging).

In this regard, the integration of ROS/ROS2 with the PX4 ecosystem has been widely explored in recent research. Studies such as [65–67] demonstrate the use of vision inertial odometry (VIO) and extended Kalman filters (EKF) for robust state estimation in GPS-denied or indoor environments. Simulation platforms leveraging PX4-ROS-Gazebo environments are another key focus, with enabling customizable testing of multi-robot collaboration, SLAM, and navigation algorithms [68–70], while reference [71] provides a comparative analysis of PX4-ROS-AirSim simulations versus real-world experiments. Further applications include autonomous navigation [72,73], marker-based precision landing [74,75], fault-tolerant control is explored in [76], developing a ROS-based con-troller for rotor failure, and [77,78] focus on point cloud data generation and laser-based localization, respectively, using ROS utilities like the PCL and SLAM. Collectively, these works showcase the versatility of the PX4-ROS ecosystem in advancing UAV capabilities across simulation and real-world deployment.

Furthermore, the ROS enables integration of third-party libraries through standardized packaging. By wrapping specialized algorithms like LIO-SAM (Lidar Inertial Odometry via Smoothing and Mapping), RTAB-Map (Real-Time Appearance-Based Mapping), VINS-Fusion (Visual-Inertial SLAM), and Kimera (real-time metric-semantic localization and mapping) into ROS packages, developers can leverage advanced capabilities while maintaining interoperability with existing ROS nodes.

Numerous contributions have adopted this approach, demonstrating their effectiveness across various challenging scenarios. For example, reference [79] combines ORB-SLAM3 for visual-inertial SLAM in GPS-denied environments with VINS-Fusion for robust visual-inertial odometry in dynamic conditions. Ref. [80] presents the application of LIO-SAM for precise localization using LiDAR-inertial odometry in challenging outdoor scenarios. Similarly, reference [81] leverages RTAB-Map for large-scale environmental mapping using RGB-D and LiDAR data, while reference [82] uses Kimera, a library for semantic mapping, which enables UAVs to interpret and interact with their environment at a higher level. These implementations collectively showcase how ROS-integrated perception algorithms can significantly enhance UAV autonomy across diverse operational conditions.

The ROS ecosystem's flexibility in integrating cutting-edge algorithms significantly enhances the Pixhawk/PX4 platform's capabilities, enabling seamless incorporation of state-of-the-art technologies for advanced UAV applications.

4.4 MATLAB Toolboxes

MATLAB offers UAV developers two complementary UAV toolboxes tailored for different development needs. The UAV Toolbox provides platform-agnostic tools for simulating and designing UAV algorithms, including mission planning, sensor modeling (e.g., cameras, LiDAR, IMUs), and environment simulation in both 2.5D and 3D. This toolbox is particularly suited for theoretical research and early-stage validation. The UAV Support Package for Pixhawk/PX4 [13] adds hardware-specific functions, such as direct Pixhawk interfacing, MAVLink communication, and automatic C++ code generation optimized for PX4. Together, these toolboxes streamline the entire development workflow, from algorithm design and simulation to real-world deployment, significantly accelerating prototyping and testing. This integrated solution makes MATLAB a valuable tool for UAV researchers and engineers working with PX4 autopilots.

In this regard, several contributions demonstrate the capabilities of MATLAB's UAV Toolbox for PX4-based development. Works such as [12,83] utilize MATLAB/Simulink for multicopter system development, providing step-by-step guidance that serves as valuable educational and research resources. Further contributions [84–86] focus on system identification and PID controller tuning, leveraging MATLAB's automatic C/C++ code generation to deploy optimized control algorithms directly on Pixhawk autopilots. For advanced control applications, papers [87–89] implement and validate techniques like sliding mode control and nonlinear backstepping control in MATLAB/Simulink before hardware deployment, demonstrating their effectiveness for challenging scenarios such as aerial slung load transportation. The work [90–92] demonstrate the use of model-based design and automatic code generation to deploy customized control algorithms on Pixhawk and companion computers like the Jetson Xavier. Finally, [93,94] address path planning and glider control, respectively, with simulations conducted in MATLAB/Simulink and real-world validation on Pixhawk-based systems. These works underscore the synergy between MATLAB/Simulink and PX4 for developing, testing, and deploying advanced UAV control systems.

4.5 SITL and HITL Simulation Environments

SITL and HITL simulation tools [39] are indispensable for safely testing and validating UAV software and control algorithms in a controlled environment, bridging the gap between design and real-world deployment.

In SITL simulations, the PX4 flight controller runs on a computer and controls a simulated vehicle in a virtual environment, which may operate on the same machine or a separate one. Communication between the simulated vehicle and a ground control station (GCS) is carried out via MAVLink messages, mirroring real-world interactions. In HITL simulations, the PX4 firmware runs on actual Pixhawk flight controller hardware, controlling a simulated vehicle that provides sensor feedback to the PX4 system. This approach allows developers to test most of the flight code on real hardware while maintaining a safe testing environment.

PX4 supports several open-source simulators for SITL and HITL simulations, with the most prominent being Gazebo [95], jMavSim [96], and AirSim [97]. Gazebo is the most widely used simulator in the Pixhawk/PX4 ecosystem. It enables complex 3D environment simulations, supports multi-vehicle operations, and integrates sensor models (e.g., cameras, LiDARs, sonars). Additionally, it offers ROS compatibility and cloud connectivity. AirSim, developed by Microsoft, simulates both aerial and ground vehicles in diverse environments. It features realistic visuals, dynamic weather conditions, and high-fidelity physics. jMavSim,

created by the Pixhawk team, is lightweight and easy to configure for basic UAV simulations. However, it lacks support for additional sensors beyond its default set and cannot simulate multiple vehicles simultaneously. All these simulators communicate with PX4 via MAVLink messages, exchanging sensor data from the simulation and actuator commands from the flight controller. In Gazebo, this interaction is further optimized through direct PX4 uORB topic integration.

Concerning SITL simulations, often integrated with Gazebo and ROS, have been extensively used in UAV research to validate control algorithms, path planning, and mission-specific applications. Works such as [88,98,99], leverage PX4-SITL to test advanced control methods like quasi-static state feedback (QSF), nonlinear backstepping, and disturbance observer-based control, while [100–102] focus on path planning for obstacle avoidance, ground target tracking, and automated power line inspection. Multi-UAV coordination, including swarm behaviors and collaborative payload transport, are explored in [103–105], with PX4-SITL serving as the validation platform. Further works [71,106,107] integrate MATLAB/Simulink, AirSim, and ROS with PX4-SITL to simulate and compare multi-drone scenarios against real-world experiments, while [108,109] employ PX4-SITL to evaluate vision-based algorithms and sensor fusion. Practical applications, such as optimized landing strategies and Cartesian mode implementation, are demonstrated in [110,111], highlighting PX4-SITL's versatility as a robust tool for developing, testing, and deploying advanced UAV systems.

On the HITL front, numerous contributions demonstrate the extensive use of PX4-Gazebo integrations for realistic UAV testing. For example, works [112–114] utilize Gazebo to simulate UAV dynamics and sensor models while implementing flight control algorithms on Pixhawk hardware. Advanced control techniques are investigated in [115], which validate incremental nonlinear dynamic inversion and robust control methods through Pixhawk-based HITL simulations. Vision-based applications are ad-dressed in [116,117], testing autonomous navigation and inspection systems via PX4-ROS-Gazebo integration. The versatility of HITL platforms is further shown in [118–120], which employ MATLAB/Simulink, X-Plane, and Flightmare Renderer. Finally, [121,122] leverage AirSim and custom physics engines for object detection, path planning, and agile control validation.

4.6 Integration of Cutting-Edge Technologies

The open and modular architecture of the Pixhawk/PX4 autopilot system, supported by its robust development ecosystem, as well as while it compatibility with GPU-enabled companion computers allows the implementation of AI/ML solutions to enhance UAV capabilities, including AI-driven sensor fusion for improved perception by combining LiDAR, IMU, and camera data, deep learning-based semantic perception for, and reinforcement learning optimized for navigation in complex environments. It also supports swarm intelligence for collaborative multi-agent operations. By leveraging simulation tools like Gazebo, AirSim, and jMAVSim, developers can train and validate AI models using synthetic data. This section explores the contexts and key use cases that demonstrate the integration of these emerging technologies.

In fact, numerous initiatives have successfully integrated AI-based solutions to enhance UAV capabilities. For instance, reference [123] creates a synthetic visual dataset using PX4-SITL and AirSim to train ML-algorithms for autonomous personal aerial vehicles, while references [124,125] leverages the PX4-ROS-Gazebo ecosystem to generate synthetic data for deep learning-based pose estimation and explainable path planning, respectively. Similarly, references [126–128] validate AI-driven path planning and swarm models using PX4-ROS-Gazebo simulations. Real-world implementations often rely on edge computing platforms, such as NVIDIA Jetson Nano and Jetson TX2 [129], for executing trained models in tasks like image processing and obstacle avoidance. Fault-tolerance control is also a key research area. In this front, reference [48] proposes an ANN-based fault-tolerant control system to replace faulty sensors, reference [130]

introduces a real-time propeller fault-detection method using SVM classifiers, and reference [131] addresses GPS faults through machine learning models integrated with EKF estimators.

Environmental perception has been advanced using YOLO for fire detection and geo-localization [132] and FPGA-accelerated powerline perception combining mmWave radar and RGB cameras [133]. Further applications include DNN-based real-time flood tracking [134] and Mask R-CNN-assisted litter detection [135]. Additionally, natural language processing (NLP) is also integrated into UAV control, with [128,136] employing DNNs and GPT-3.5-based architectures to translate voice commands into executable tasks within PX4-ROS-Gazebo frameworks.

Researchers are leveraging a range of advanced solutions to address UAV security challenges. Initial efforts involved several *ad-hoc* approaches, including encryption techniques such as using a Raspberry Pi for encrypted communication links [137] and implementing the ChaCha20 algorithm through the MAVSec framework [138,139]. Other proposed methods include software rejuvenation, which employs intermittent system restarts to mitigate runtime attacks [140]. Machine learning techniques, particularly LSTM-based neural networks, have also been explored for detecting denial of service (DoS) attacks [141,142]. GPS-spoofing attacks, where hijackers use devices to deceive GPS receivers, have also garnered attention. In this sense, work in [143] describes a method to detect whether it has been hijacked using only GPS measurements. Beyond these, more sophisticated solutions continue to emerge in response to evolving threats. For instance, reference [144] introduces an intelligent intrusion detection system (IDS) that utilizes principal component analysis (PCA) and one-class classifiers to detect attacks, leveraging flight logs for training the models. Similarly, reference [145] proposes a deep learning-based pilot authentication scheme, DP-Authentication, which dynamically verifies pilot legitimacy using flight data collected from the PX4 flight stack. Further, the work [146] proposes a fusion multi-tier DNN-based collaborative intrusion detection and prevention system (CIDPS) for UAV-enabled 6G networks to neutralize attacks, which are implemented on actual PX4-based UAVs. These contributions underscore the potential of AI techniques to address evolving cybersecurity challenges in UAV systems.

All these studies demonstrate the PX4 ecosystem's effectiveness in supporting AI integration across various UAV applications.

5 Pixhawk/PX4 in Educational Settings

The Pixhawk/PX4 autopilot has gained widespread adoption in educational settings across various engineering disciplines, underscoring its importance in teaching control systems, robotics, and mechatronics through hands-on and simulation-based approaches. The studies surveyed primarily report descriptive accounts of platform adoption, course integration strategies, and observed student engagement rather than formally assessed learning gains against control groups. We therefore present these as evidence of educational applicability and reported pedagogical insights.

For instance, reference [83] describes RflySim, a multi-copter platform built on Pixhawk/PX4 and MATLAB/Simulink, designed to teach low-level controller design (e.g., attitude and position control) as well as high-level applications such as autonomous flight and decision-making. Additionally, the book [12] offers a series of step-by-step experiments ranging from introductory to advanced levels, complete with detailed source codes, and introduces SITL and HITL flight testing simulations. Further, reference [133] presents a quadcopter testing platform tailored for educational purposes, emphasizing the use of MATLAB/Simulink for controller development and C++ code generation, while Refs. [147,148] describes an educational testbed for experimenting with advanced control concepts such as trajectory tracking, disturbance rejection, and fault-tolerant control. Similarly, reference [149] provides a comprehensive guide for educators on leveraging multirotor UAV platforms within the PX4 ecosystem to achieve diverse learning out-comes in undergraduate

mechatronics education, covering both hardware and software aspects to enhance engineering education. Additionally, reference [150] proposes a Pixhawk/PX4 platform for studying the longitudinal and pitch dynamics of a multirotor UAV, enabling experiments in a safe and controlled environment. Meanwhile, reference [151] explores the integration of the PX4-MATLAB/Simulink ecosystem with X-Plane and FlightGear flight simulators to create an immersive learning experience. Further, reference [25] introduces a COST UAV platform based on the PX4-ROS-Gazebo ecosystem, incorporating a RPi companion computer for teaching UAV hardware integration and theoretical concepts of guidance and navigation, and reference [152] utilizes the PX4-SITL combined with the ROS-Gazebo ecosystem as an educational UAV cybersecurity laboratory platform to implement and test various types of cybersecurity threats to UAVs.

The application of Pixhawk/PX4 in distance education and remote experimentation has also garnered significant attention. For example, reference [153] describes an educational testbed that replicates quadcopter dynamics for remote laboratory activities, while reference [154] proposes online courses and hands-on labs utilizing the PX4-ROS-Gazebo ecosystem and Raspberry Pi-based visual navigation, respectively. Similarly, reference [155] introduces OpenUAV, an open-source testbed based on PX4-ROS-Gazebo and Docker technologies for UAV education, where the platform employs Containers as a Service (CaaS) technology, enabling students to conduct virtual simulations on the cloud without requiring specialized hardware.

Finally, reference [156] presents the integration of project-based-learning (PBL) into a graduate-level optimal control course, focusing on the design, implementation, and experimental validation of optimal control strategies using a PX4-based testbed, discussing the benefits of the experiential learning approach. Similarly, reference [157] highlights the effectiveness of project-based learning (PBL) courses using Pixhawk/PX4-based platforms, providing students with practical, hands-on experience in control education.

All these contributions emphasize the versatility and broad applicability of the Pixhawk/PX4 ecosystem in educational settings, showcasing its effectiveness in promoting hands-on learning and advancing practical understanding across diverse engineering fields.

6 Discussion

The distribution of surveyed publications across key development topics is summarized in Table 2, offering valuable insights into the Pixhawk/PX4 research landscape by revealing both established priorities and emerging trends.

Table 2: Breakdown of the survey publications across key development topics.

Topics	References	Total
Companion Computer	[40–54]	14
DroneKit and MAVSDK	[54–64]	9
ROS Integration	[65–82]	17
MATLAB SPP	[12,83–94]	12
SITL and HITL	[71,88,92,98–122]	27
Cutting-Edge and Security	[48,123–146]	24
Educational Context	[12,25,83,133,147–157]	14

The first observation is the prominence of SITL and HITL simulation research, which accounts for 27 publications, the highest count among all categories, reflecting the fundamental role that simulation plays as the essential bridge between theoretical algorithms and safe real-world deployment. The second highest category, Cutting-Edge and Security with 24 publications, demonstrates the community's forward-looking

orientation and responsiveness to contemporary challenges, with substantial attention to AI integration, machine learning, and cybersecurity reflecting broader technological trends reshaping autonomous systems. ROS integration follows with 17 publications, representing a well-established and continuously evolving research domain that confirms ROS as the middleware of choice for advanced autonomy research, with researchers exploring sophisticated applications including multi-vehicle coordination and advanced perception pipelines. The use of companion computer category shows publication counts of 14 each, reflecting recognition that onboard processing is essential for real-time autonomy. The MATLAB Support Package for Pixhawk with 12 publications and DroneKit-MAVSDK APIs with 9 publications represent the smallest categories, suggesting these are more specialized or application-oriented research areas focused on control design workflows and enabling technologies rather than research subjects themselves.

With 14 publications dedicated to education representing a moderate portion of the overall literature, this figure understates the ecosystem's true pedagogical footprint. The table reveals that education is not an isolated silo, but a cross-cutting theme deeply embedded within other categories. For instance, publications on MATLAB (such as references [12,83]) and SITL/HITL simulations (such as reference [92]) often serve dual purposes, advancing technical research while simultaneously providing structured, replicable frameworks for student learning. This suggests that the educational value of the Pixhawk/PX4 ecosystem is amplified by its "learn by-doing" nature; the very tools used for cutting-edge research, such as ROS for autonomy or Gazebo for simulation, are accessible enough to serve as effective hands-on platforms in the classroom.

The distribution reveals that the top three categories account for approximately 58 percent of the total 117 publications, indicating that the research community's efforts are primarily directed toward validation methodologies, emerging technologies, and middleware integration, while the presence of multiple categories with 12–14 publications ensures a healthy diversity of research interests across hardware integration, educational applications, and specialized tools. This quantitative snapshot effectively maps the research topography of the PX4 ecosystem, revealing simulation validation as the foundational priority, cutting-edge technologies as the emerging frontier, and ROS integration as the established backbone for advanced autonomy, while providing researchers with valuable context for positioning their work within the broader literature.

To provide readers with a quick reference, Table 3 summarizes the core insights across the seven development topics.

Table 3: Takeaway summary from the Pixhawk/PX4 ecosystem analysis.

Topics	Core Takeaway
Companion Computers	Companion computers transform PX4 from a flight controller into a high-performance autonomous system by offloading heavy computation (vision, SLAM, AI, swarm coordination). The 3 configurations offer flexibility: onboard provides balanced autonomy with embedded processing; swappable integrated boards offer compact, high-performance solutions; offboard provides unlimited computing power without payload penalties. Core insight: Companion computers are the key enabler for advanced autonomy and swarm intelligence in PX4-based UAVs.

(Continued)

Table 3 (continued)

Topics	Core Takeaway
DroneKit & MAVSDK	High-level APIs simplify UAV control beyond firmware-level programming. DroneKit offers easy Python scripting but is legacy and unmaintained. MAVSDK is actively maintained, multi-language, robust, and swarm capable. Core insight: MAVSDK is the future-proof, scalable interface for advanced control and multi-UAV systems, while DroneKit remains useful mainly for legacy or educational projects.
ROS Integration	ROS2 elevates PX4 into a full robotics research platform by enabling modular perception, localization, SLAM, and advanced autonomy through standardized middleware. The shift from ROS to ROS2 brings real-time capability, improved security, and better QoS for multi-vehicle systems. Core insight: ROS2 is the backbone for state-of-the-art UAV autonomy, allowing seamless integration of cutting-edge perception and navigation algorithms.
MATLA SPP	MATLAB/Simulink provides a model-based, end-to-end workflow from simulation to automatic code generation for Pixhawk hardware. It is particularly strong for control design and validation, system identification, educational experimentation, and rapid prototyping with automatic C++ generation. Core insight: MATLAB accelerates the control design-to-deployment pipeline, making PX4 accessible for both researchers and educators focused on control theory.
SITL & HITL	SITL and HITL bridge theory and real-world deployment by enabling safe, repeatable, and scalable validation. SITL offers software-only testing with fast iteration, while HITL provides real hardware validation without flight risk. Gazebo dominates research flexibility, AirSim offers high-fidelity visuals, and jMavSim supports lightweight testing. Core insight: Simulation is fundamental to PX4 project development, enabling safe testing before real flights.
Cutting-Edge Tech & Security	The open architecture of PX4 enables seamless integration of AI/ML, edge computing, and cybersecurity solutions. Applications include AI-driven perception and sensor fusion, reinforcement learning navigation, swarm intelligence, fault detection and resilience, and cyberattack detection and mitigation. Core insight: PX4 is not just a flight stack, it is an extensible AI-ready platform capable of supporting next-generation intelligent and secure UAV systems.

(Continued)

Table 3 (continued)

Topics	Core Takeaway
Educational Context	PX4 has evolved into a comprehensive educational platform supporting hands-on learning, remote labs, cybersecurity training, and project-based learning. Strengths include integration with MATLAB, ROS, and Gazebo; SITL/HITL for safe experimentation; cloud and remote lab compatibility; and support for advanced control and autonomy courses. Core insight: PX4 effectively bridges theory and practice, making it a powerful pedagogical tool.

Finally, for those looking to get started with the PX4 autopilot, essential references begin with the work [26], which explains the PX4 architecture from real-time perspective. MAVLink is a fundamental component, and Koubaa [27] provides an overview of the MAVLink protocol, highlighting the differences between its versions and its potential to enable Internet connectivity for unmanned systems. Security aspects are further discussed in [138]. The interoperability of PX4 with other environments is extensive and crucial for many development projects. For example, references [12,83] present a series of simulations that facilitate rich interaction between PX4 and MATLAB/Simulink. Regarding interaction with ROS, the work [71] offers a detailed case study that can guide readers through the setup of the PX4-ROS ecosystem, including how to perform SITL simulations with AirSim. Similarly, the studies [108,112] outline the steps for data transfer between PX4 and Gazebo for conducting SITL and HITL simulations.

Pixhawk/PX4 Limitations and Vulnerabilities

Despite the widespread adoption of the Pixhawk/PX4 platform across research and educational applications, it is not without limitations and vulnerabilities. As UAV operations grow increasingly complex, demanding greater onboard performance, extended flight endurance, and robust communication, the platform's constraints become more apparent. These challenges manifest across three critical dimensions: the integration of companion computers, the reliance on simulation environments, and the MAVLink communication protocol.

Regarding companion computers, practical deployment is tightly constrained by power, thermal, and latency concerns. Power consumption varies significantly across platforms: a Raspberry Pi 4 can draw several amperes, potentially exceeding the 3 A capacity of typical Pixhawk power modules, while NVIDIA Jetson modules range from 5–10 W (Nano) to 10–25 W (Orin NX), representing a substantial drain on flight endurance [158–160]. Thermal limits further constrain sustained performance; Jetson SoCs throttle when junction temperatures exceed 85°C, reducing inference frequency precisely when perception tasks are most demanding. End-to-end latency introduces additional complexity: onboard processing avoids radio links but contends with shared CPU/GPU contention, whereas offboard configurations risk variable network delays. These constraints shift the discourse from functional feasibility toward deployment viability under real flight profiles.

Simulation environments present a different but equally critical challenge. Despite their extensive adoption, the realism of SITL and HITL simulations remains a central concern, as simulation-to-real (Sim2Real) discrepancies continue to undermine UAV validation. Simulated environments cannot fully replicate the coupled aerodynamic, environmental, sensing, and hardware uncertainties inherent to physical flight [161]. The tools reviewed address distinct layers of this gap: jMavSim enables rapid control-law

prototyping; Gazebo provides physics-consistent dynamics for system-level integration; and AirSim targets perceptual robustness through high-fidelity visuals. Rather than treating any single simulator as a complete solution, effective Sim2Real mitigation increasingly depends on staged validation pipelines that strategically compose these platforms—progressing from abstract control verification to sensor-rich HITL testing before deployment [162]. Simulation tools thus remain indispensable for iterative development, but they constitute one stage within a broader verification continuum rather than a substitute for physical experimentation.

Despite its versatility, the MAVLink protocol exhibits notable limitations [138,163]. Its constrained bandwidth makes it unsuitable for applications requiring high data rates or low-latency links. It is also designed primarily for short-range communication, limiting its use in beyond visual line of sight (BVLOS) operations, though relay UAVs have been employed as a workaround. The protocol is prone to latency and fluctuations in packet reception times, posing challenges for real-time remote control [164,165]. Security vulnerabilities, including packet injection and flooding attacks, further compound these issues [166].

Understanding these limitations is essential not only for mitigating risks but also for guiding future improvements that can solidify the platform's role in next-generation autonomous systems.

7 Conclusions

This survey has explored the diverse applications of the Pixhawk/PX4 platform, highlighting its widespread adoption across research and education. Its extensive use in scientific literature, coupled with its compatibility with cutting-edge technologies like AI and cybersecurity solutions, underscores the platform's foundational role in modern UAV development.

The platform's versatility lies in its seamless integration of auxiliary computers across various configurations, empowering developers to implement advanced capabilities. Its software ecosystem, including MAVSDK/DroneKit APIs, ROS integration, and MATLAB compatibility, accelerates development from simulation to deployment. Furthermore, PX4 supports SITL and HITL simulations, enabling comprehensive validation of algorithms and software before re-al-world testing. Beyond research applications, the Pixhawk/PX4 ecosystem is also adopted as a testbed for hands-on learning, facilitating both local and remote experimentation as well as project-based education in UAV systems development. These educational applications underscore the platform's versatility, bridging theoretical concepts with real world engineering challenges, and making it an unvaluable tool for engineering education.

By synthesizing these insights, this survey aims to equip researchers, developers, and educators with a deeper understanding of the Pixhawk/PX4 ecosystem, supporting informed decision-making for their UAV projects.

Looking ahead, several strategic directions can meaningfully advance the Pixhawk/PX4 platform. A primary focus must be strengthening cybersecurity, moving beyond basic safeguards to deploy robust technologies that actively counter evolving threats and prevent unauthorized access, an essential step for operations in sensitive or contested environments. At the same time, addressing the persistent simulation-to-reality gap calls for advanced validation pipelines that balance high-fidelity simulation with real-world testing, ensuring reliable algorithm transfer without compromising safety. Complementing these efforts, a concerted push for greater interoperability across diverse autopilot systems could be transformative. Establishing common communication protocols and modular interfaces would enable more flexible and collaborative UAV operations. Pursuing these interconnected avenues will not only reinforce the platform's reliability and security but also open new frontiers for innovation in autonomous systems.

Acknowledgement: During the preparation of this manuscript, the author utilized ChapGPT-4 to refine the academic language and improve some figures. The author has carefully reviewed and revised the output and accepted full responsibility for all content.

Funding Statement: The authors received no specific funding for this study.

Availability of Data and Materials: No new data were created or analyzed in this study. Data sharing is not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Kumar V, Kotler P. Transformative marketing using drones. In: Transformative marketing. Palgrave executive essentials. Cham, Switzerland: Palgrave Macmillan; 2024. doi:10.1007/978-3-031-59637-7_9.
2. Ebeid E, Skriver M, Terkildsen KH, Jensen K, Schultz UP. A survey of Open-Source UAV flight controllers and flight simulators. Microprocess Microsyst. 2018;61(1):11–20. doi:10.1016/j.micpro.2018.05.002.
3. Paparazzi. [cited 2026 Feb 15]. Available from: https://wiki.paparazziuav.org/wiki/overview.
4. ArduPilot. [cited 2026 Feb 15]. Available from: https://ArduPilot.org/dev/index.html.
5. Pixhawk. [cited 2026 Feb 15]. Available from: https://pixhawk.org/.
6. PX4. [cited 2026 Feb 15]. Available from: https://px4.io/.
7. LibrePilot. [cited 2026 Feb 15]. Available from: https://www.librepilot.org/site/index.html.
8. Betaflight. [cited 2026 Feb 15]. Available from: https://betaflight.com/.
9. iNAV. [cited 2026 Feb 15]. Available from: https://github.com/inavFlight/inav/wiki.
10. Aliane N. A survey of open-source UAV autopilots. Electronics. 2024;13(23):4785. doi:10.3390/electronics13234785.
11. Peksa J, Mamchur D. A review on the state of the art in copter drones and flight control systems. Sensors. 2024;24(11):3349. doi:10.3390/s24113349.
12. Quan Q, Dai X, Wang S. Multicopter design and control practice: a series experiments based on MATLAB and pixhawk. Singapore: Springer; 2020. doi:10.1007/978-981-15-3138-5.
13. MathWorks PSP. [cited 2026 Feb 15]. Available from: https://es.mathworks.com/help/supportpkg/px4/index.html.
14. Dronecode. [cited 2026 Feb 15]. Available from: https://dronecode.org/.
15. Telli K, Kraa O, Himeur Y, Ouamane A, Boumehraz M, Atalla S, et al. A comprehensive review of recent research trends on unmanned aerial vehicles (UAVs). Systems. 2023;11(8):400. doi:10.3390/systems11080400.
16. Durmuş A, Duymaz E, Kadıoğlu AM. A survey on flight controllers used for UAV platforms. In: Research and updates on the use of artificial intelligence in drone technology. Cham, Switzerland: Springer Nature; 2025. p. 135–42. doi:10.1007/978-3-032-07678-6_24.
17. Mohammed A, Lawal MS, Hassan A, Dandago KK, Ridwanullahi I, Denan F, et al. A survey of flight control systems and simulators of aerial robots. FUW Trends Sci Technol J. 2021;6:907–16.
18. Marshall JA, Sun W, L'Afflitto A. A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems. Annu Rev Control. 2021;52:390–427. doi:10.1016/j.arcontrol.2021.10.013.
19. Elmokadem T, Savkin AV. Towards fully autonomous UAVs: a survey. Sensors. 2021;21(18):6223. doi:10.3390/s21186223.
20. Tahir MA, Mir I, Islam TU. A review of UAV platforms for autonomous applications: comprehensive analysis and future directions. IEEE Access. 2023;11:52540–54. doi:10.1109/ACCESS.2023.3273780.
21. Ahmed F, Jenihhin M. A survey on UAV computing platforms: a hardware reliability perspective. Sensors. 2022;22(16):6286. doi:10.3390/s22166286.
22. Wang L. Review of the application of open source flight control in multi-rotor aircraft. Int Core J Eng. 2021;7(8):261–70. doi:10.6919/ICJE.202108_7(8).0036.
23. Ivashko V, Krulikovskyi O, Haliuk S, Samila A. Review of operating systems used in unmanned aerial vehicles. Inform Autom Pomiary W Gospod I Ochr Środowiska. 2025;15(1):95–100. doi:10.35784/iapgos.6786.

24. Nikolaiev M, Novotarskyi M. Comparative review of drone simulators. Inf Comput and Intell Syst J. 2024;2024(4):79–98. doi:10.20535/2786-8729.4.2024.300614.
25. Horri N, Pietraszko M. A tutorial and review on flight control co-simulation using Matlab/Simulink and flight simulators. Automation. 2022;3(3):486–510. doi:10.3390/automation3030025.
26. Meier L, Honegger D, Pollefeys M. PX4: a node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA); 2015 May 26–30; Seattle, WA, USA. Piscataway, NJ, USA: IEEE; 2015. p. 6235–40. doi:10.1109/ICRA.2015.7140074.
27. Koubâa A, Allouch A, Alajlan M, Javed Y, Belghith A, Khalgui M. Micro air vehicle link (MAVlink) in a nutshell: a survey. IEEE Access. 2019;7:87658–80. doi:10.1109/ACCESS.2019.2924410.
28. PX4-RPi CM4. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/companion_computer/holybro_pixhawk_rpi_cm4_baseboard.html.
29. PX4-Jetson. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/companion_computer/holybro_pixhawk_jetson_baseboard.html.
30. PX4-Auterion-Skynode. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/companion_computer/auterion_skynode.html.
31. PX4-NuttX OS. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/concept/architecture.html#os-specific-information.
32. PX4-uORB. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/middleware/uorb.html.
33. QGroundControl. [cited 2026 Feb 15]. Available from: http://qgroundcontrol.com/.
34. MissionPlaner. [cited 2026 Feb 15]. Available from: https://ardupilot.org/planner/.
35. MAVProxy. [cited 2026 Feb 15]. Available from: https://ardupilot.org/mavproxy/.
36. DroneKit. [cited 2026 Feb 15]. Available from: https://github.com/dronekit/dronekit-python.
37. MAVSDK. [cited 2026 Feb 15]. Available from: https://MAVSDK.MAVLink.io/main/en/index.html.
38. PX4-ROS2. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/ros/ros2_comm.html.
39. PX4-SITL-HITL. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/simulation/hitl.
40. De Cos CR, Fernandez MJ, Sanchez-Cuevas PJ, Acosta JA, Ollero A. High-level modular autopilot solution for fast prototyping of unmanned aerial systems. IEEE Access. 2020;8:223827–36. doi:10.1109/access.2020.3044098.
41. Bakirci M. A novel swarm unmanned aerial vehicle system: incorporating autonomous flight, real-time object detection, and coordinated intelligence for enhanced performance. Trait Du Signal. 2023;40(5):2063–78. doi:10.18280/ts.400524.
42. Ma D, Tran A, Keti N, Yanagi R, Knight P, Joglekar K, et al. Flight test validation of collision avoidance system for a multicopter using stereoscopic vision. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS); 2019 Jun 11–14; Atlanta, GA, USA. Piscataway, NJ, USA: IEEE; 2019. p. 989–95. doi:10.1109/ICUAS.2019.8798023.
43. Wu CH, Tu SH, Tu SW, Wang LH, Chen WH. Realization of remote monitoring and navigation system for multiple UAV swarm missions: using 4G/WiFi-mesh communications and RTK GPS positioning technology. In: 2022 International Automatic Control Conference (CACS); 2022 Nov 3–6; Kaohsiung, Taiwan. Piscataway, NJ, USA: IEEE; 2022. p. 1–6. doi:10.1109/CACS55319.2022.9969782.
44. Qu X, Wei Y, Liu Y, Su X. Design of automatic search and rescue UAV based on jetson nano combined with PX4 pixhawk flight controller and color recognition technology. In: 2024 International Conference on Electrical Drives, Power Electronics & Engineering (EDPEE); 2024 Feb 27–29; Athens, Greece. Piscataway, NJ, USA: IEEE; 2024. p. 460–6. doi:10.1109/EDPEE61724.2024.00092.
45. Kostin AS. Models and methods for implementing the automous performance of transportation tasks using a drone. In: 2021 Wave Electronics and Its Application in Information and Telecommunication Systems (WECONF); 2021 May 31–Jun 4; St. Petersburg, Russia. Piscataway, NJ, USA: IEEE; 2021. p. 1–4. doi:10.1109/WECONF51603.2021.9470584.
46. Grobler PR, Jordaan HW. Autonomous vision based landing strategy for a rotary wing UAV. In: 2020 International SAUPEC/RobMech/PRASA Conference; 2020 Jan 29–31; Cape Town, South Africa. Piscataway, NJ, USA: IEEE; 2020. p. 1–6. doi:10.1109/SAUPEC/RobMech/PRASA48453.2020.9041238.

47. Weber MP, Mishra C, Holzapfel F. Analysis and testing of data link time of flight measurements in data fusion algorithms for positioning of UAV. In: AIAA SCITECH 2024 Forum; 2024 Jan 8–12; Orlando, FL, USA. Palo Alto, CA, USA: AAAI Press; 2024. p. 2796. doi:10.2514/6.2024-2796.
48. Quidiello A, Gururajan S. Development of virtual sensors using a compact computer cluster for fault tolerant control on UAVs. In: AIAA SCITECH 2024 Forum; 2024 Jan 8–12; Orlando, FL, USA. Palo Alto, CA, USA: AAAI Press; 2024. p. 2904. doi:10.2514/6.2024-2904.
49. Kamath AK, Chan NT, Feroskhan M. Fixed-time fractional-order sliding mode control for image-based visual servoing of hexarotor. In: 2024 International Conference on Unmanned Aircraft Systems (ICUAS); 2024 Jun 4–7; Chania-Crete, Greece. Piscataway, NJ, USA: IEEE; 2024. p. 512–21. doi:10.1109/ICUAS60882.2024.10556898.
50. Jain S, Shethwala Y, Akurathi S, Das J. Payload delivery through acrobatic quadrotor flip-and-throw maneuver using model predictive control. In: 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE); 2024 Aug 28–Sep 1; Bari, Italy. Piscataway, NJ, USA: IEEE; 2024. p. 2463–70. doi:10.1109/CASE59546.2024.10711775.
51. Yan Y, Wang XF, Marshall BJ, Liu C, Yang J, Chen WH. Surviving disturbances: a predictive control framework with guaranteed safety. Automatica. 2023;158(8):111238. doi:10.1016/j.automatica.2023.111238.
52. Salunkhe DH, Sharma S, Topno SA, Darapaneni C, Kankane A, Shah SV. Design, trajectory generation and control of quadrotor research platform. In: 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA); 2016 Dec 18–20; Amritapuri, India. Piscataway, NJ, USA: IEEE; 2017. p. 1–7. doi:10.1109/RAHA.2016.7931876.
53. Pinto J, Guerreiro BJ, Cunha R. Planning aggressive drone manoeuvres: a geometric backwards integration approach. J Intell Rob Syst. 2025;111(1):16. doi:10.1007/s10846-024-02214-z.
54. Jackson BE, Howell TA, Shah K, Schwager M, Manchester Z. Scalable cooperative transport of cable-suspended loads with UAVs using distributed trajectory optimization. IEEE Robot Autom Lett. 2020;5(2):3368–74. doi:10.1109/LRA.2020.2975956.
55. Pulungan AB, Putra ZY, Sidiqi AR, Hamdani H, Parigalan KE. Drone kit-Python for autonomous quadcopter navigation. Int J Inf Vis. 2024;8(3):1287. doi:10.62527/joiv.8.3.2301.
56. Kumar GP, Sridevi B. Simulation of efficient cooperative UAVs using modified PSO algorithm. WSEAS Trans Inf Sci Appl. 2019;16(5):94–9. doi:10.1109/tmag.2010.2087316.
57. Moffatt A, Platt E, Mondragon B, Kwok A, Uryeu D, Bhandari S. Obstacle detection and avoidance system for small UAVs using a LiDAR. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS); 2020 Sep 1–4; Athens, Greece. Piscataway, NJ, USA: IEEE; 2020. p. 633–40. doi:10.1109/icuas48674.2020.9213897.
58. Kumar GH, KN VS, Patil P, Moinuddin M, Faraz M, Kumar YD. Human-computer interaction for drone control through hand gesture recognition with MediaPipe integration. In: 2024 International Conference on Vehicular Technology and Transportation Systems (ICVTTS); 2024 Sep 27–28; Bangalore, India. Piscataway, NJ, USA: IEEE; 2024. p. 1–6. doi:10.1109/ICVTTS62812.2024.10763917.
59. B L, Penmesta AV, Mohan Rao Kovvur R, Kune R. Real time voice/speech command and control system (CCS) for unmanned aerial and ground vehicles on 4G cellular/GPRS network. In: 2023 Global Conference on Information Technologies and Communications (GCITC); 2023 Dec 1–3; Bangalore, India. Piscataway, NJ, USA: IEEE; 2024. p. 1–8. doi:10.1109/GCITC60406.2023.10426146.
60. Varatharasan V, Rao ASS, Toutounji E, Hong JH, Shin HS. Target detection, tracking and avoidance system for low-cost UAVs using AI-based approaches. In: 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS); 2019 Nov 25–27; Cranfield, UK. Piscataway, NJ, USA: IEEE; 2020. p. 142–7. doi:10.1109/REDUAS47371.2019.8999683.
61. Kishore K, Dalai S, Jangir Y, Singh S, Rohan M, Shashank D, et al. 3D pure pursuit guidance of drones for autonomous precision landing. In: 2022 13th Asian Control Conference (ASCC); 2022 May 4–7; Jeju, Republic of Korea. Piscataway, NJ, USA: IEEE; 2022. p. 2218–22. doi:10.23919/ascc56756.2022.9828198.
62. Pedroche DS, Amigo D, García J, Molina JM, Zubasti P. Drone swarm for distributed video surveillance of roads and car tracking. Drones. 2024;8(11):695. doi:10.3390/drones8110695.

63. Cabuk UC, Tosun M, Dagdeviren O, Ozturk Y. An architectural design for autonomous and networked drones. In: MILCOM 2022—2022 IEEE Military Communications Conference (MILCOM); 2022 Nov 28–Dec 2; Rockville, MD, USA. Piscataway, NJ, USA: IEEE; 2022. p. 962–7. doi:10.1109/milcom55135.2022.10017877.
64. Koulianos A, Litke A. Blockchain technology for secure communication and formation control in smart drone swarms. Future Internet. 2023;15(10):344. doi:10.3390/fi15100344.
65. Sun K, Mohta K, Pfrommer B, Watterson M, Liu S, Mulgaonkar Y, et al. Robust stereo visual inertial odometry for fast autonomous flight. IEEE Robot Autom Lett. 2018;3(2):965–72. doi:10.1109/lra.2018.2793349.
66. Wang Y, Jie TY, Gimin S, Kenny LBY, Srigrarom S. Autonomous customized quadrotor with vision-aided navigation for indoor flight challenges. In: 2022 8th International Conference on Mechatronics and Robotics Engineering (ICMRE); 2022 Feb 10–12; Munich, Germany. Piscataway, NJ, USA: IEEE; 2022. p. 33–7. doi:10.1109/ICMRE54455.2022.9734105.
67. Lim E. Pose estimation of a drone using dynamic extended Kalman filter based on a fuzzy system. In: 2021 9th International Conference on Control, Mechatronics and Automation (ICCMA); 2021 Nov 11–14; Belval, Luxembourg. Piscataway, NJ, USA: IEEE; 2021. p. 141–5. doi:10.1109/iccma54375.2021.9646187.
68. Liu X, Wang Z, Qin M. Simulation experiment of CCM-SLAM based on ROS. In: 2022 34th Chinese Control and Decision Conference (CCDC); 2022 Aug 15–17; Hefei, China. Piscataway, NJ, USA: IEEE; 2023. p. 2001–5. doi:10.1109/CCDC55256.2022.10034369.
69. Xiao K, Tan S, Wang G, An X, Wang X, Wang X. XTDrone: a customizable multi-rotor UAVs simulation platform. In: 2020 4th International Conference on Robotics and Automation Sciences (ICRAS); 2020 Jun 12–14; Wuhan, China. Piscataway, NJ, USA: IEEE; 2020. p. 55–61. doi:10.1109/icras49812.2020.9134922.
70. Chen S, Zhou W, Yang AS, Chen H, Li B, Wen CY. An end-to-end UAV simulation platform for visual SLAM and navigation. Aerospace. 2022;9(2):48. doi:10.3390/aerospace9020048.
71. Ma C, Zhou Y, Li Z. A new simulation environment based on airsim, ROS, and PX4 for quadcopter aircrafts. In: 2020 6th International Conference on Control, Automation and Robotics (ICCAR); 2020 Apr 20–23; Singapore. Piscataway, NJ, USA: IEEE; 2020. p. 486–90. doi:10.1109/iccar49639.2020.9108103.
72. Boiteau S, Vanegas F, Gonzalez F. A comprehensive framework for UAV-based autonomous target finding in environments with limited global navigation satellite systems and reduced visibility. In: 2024 International Conference on Unmanned Aircraft Systems (ICUAS); 2024 Jun 4–7; Chania-Crete, Greece. Piscataway, NJ, USA: IEEE; 2024. p. 160–7. doi:10.1109/ICUAS60882.2024.10556868.
73. Antonopoulos A, Lagoudakis MG, Partsinevelos P. A ROS multi-tier UAV localization module based on GNSS, inertial and visual-depth data. Drones. 2022;6(6):135. doi:10.3390/drones6060135.
74. Daspan A, Nimsongprasert A, Srichai P, Wiengchanda P. Implementation of robot operating system in raspberry pi 4 for autonomous landing quadrotor on ArUco marker. Int J Mech Eng Robot Res. 2023;2023:210–5. doi:10.18178/ijmerr.12.4.210-215.
75. Khazetdinov A, Zakiev A, Tsoy T, Svinin M, Magid E. Embedded ArUco: a novel approach for high precision UAV landing. In: 2021 International Siberian Conference on Control and Communications (SIBCON); 2021 May 13–15; Kazan, Russia. Piscataway, NJ, USA: IEEE; 2021. p. 1–6. doi:10.1109/sibcon50419.2021.9438855.
76. Chen T, Zhuang X, Hou Z, Chen H. A pixhawk-ROS based development solution for the research of autonomous quadrotor flight with a rotor failure. In: 2022 IEEE International Conference on Unmanned Systems (ICUS); 2022 Oct 28–30; Guangzhou, China. Piscataway, NJ, USA: IEEE; 2022. p. 590–5. doi:10.1109/ICUS55513.2022.9986633.
77. Freimuth H, König M. A framework for automated acquisition and processing of As-built data with autonomous unmanned aerial vehicles. Sensors. 2019;19(20):4513. doi:10.3390/s19204513.
78. Bauersfeld L, Ducard G. RTOB SLAM: real-time onboard laser-based localization and mapping. Vehicles. 2021;3(4):778–89. doi:10.3390/vehicles3040046.
79. Wang F, Zou Y, Zhang C, Buzzatto J, Liarokapis M, del Rey Castillo E, et al. UAV navigation in large-scale GPS-denied bridge environments using fiducial marker-corrected stereo visual-inertial localisation. Autom Constr. 2023;156(3):105139. doi:10.1016/j.autcon.2023.105139.
80. Araújo AG, Pizzino CAP, Couceiro MS, Rocha RP. A multi-drone system proof of concept for forestry applications. Drones. 2025;9(2):80. doi:10.3390/drones9020080.

81. Lin HY, Chang KL, Huang HY. Development of unmanned aerial vehicle navigation and warehouse inventory system based on reinforcement learning. Drones. 2024;8(6):220. doi:10.3390/drones8060220.
82. Bauer E, Blöchlinger M, Strauch P, Raayatsanati A, Cavelti C, Katzschmann RK. An open-source soft robotic platform for autonomous aerial manipulation in the wild. arXiv:2409.07662. 2024.
83. Wang S, Dai X, Ke C, Quan Q. RflySim: a rapid multicopter development platform for education and research based on pixhawk and MATLAB. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS); 2021 Jun 15–18; Athens, Greece. Piscataway, NJ, USA: IEEE; 2021. p. 1587–94. doi:10.1109/icuas51884.2021.9476786.
84. Fareha A, Bousbaine A, Josaph AK. A hardware implementation of 6DOF quadcopter MATLAB/Simulink controller algorithm to an autopilot. In: Proceedings of the 10th International Conference on Power Electronics, Machines and Drives (PEMD 2020). Stevenage, UK: IET; 2020. p. 485–90. doi:10.1049/icp.2021.1078.
85. Saengphet W, Tantrairatn S, Thumtae C, Srisertpol J. Implementation of system identification and flight control system for UAV. In: 2017 3rd International Conference on Control, Automation and Robotics (ICCAR); 2017 Apr 24–26; Nagoya, Japan. Piscataway, NJ, USA: IEEE; 2017. p. 678–83. doi:10.1109/ICCAR.2017.7942783.
86. Bazzocchi S, Warwick S, Suleman A. Automatic autopilot tuning framework using genetic algorithms and system identification. Aerosp Sci Technol. 2025;157:109779. doi:10.1016/j.ast.2024.109779.
87. Jing Y, Wang X, Heredia-Juesas J, Fortner C, Giacomo C, Sipahi R, et al. PX4 simulation results of a quadcopter with a disturbance-observer-based and PSO-optimized sliding mode surface controller. Drones. 2022;6(9):261. doi:10.3390/drones6090261.
88. Flores G, de Oca AM, Flores A. Robust nonlinear control for the fully actuated hexa-rotor: theory and experiments. IEEE Control Syst Lett. 2023;7:277–82. doi:10.1109/LCSYS.2022.3188517.
89. Mechali O, Xu L. Distributed fixed-time sliding mode control of time-delayed quadrotors aircraft for cooperative aerial payload transportation: theory and practice. Adv Space Res. 2023;71(9):3897–916. doi:10.1016/j.asr.2022.12.037.
90. Maciel MD. Development of a low-cost avionics platform for small-scale model airplanes. In: 2021 IEEE International Conference on Mechatronics (ICM); 2021 Mar 7–9; Kashiwa, Japan. Piscataway, NJ, USA: IEEE; 2021. p. 1–6. doi:10.1109/ICM46511.2021.9385641.
91. Wang G, Cao L, Wu X, Zhou J, Song P, Cao Y. Design and implementation of model-based design for UAV intelligent decision-making and control system. In: 2023 China Automation Congress (CAC); 2023 Nov 17–19; Chongqing, China. Piscataway, NJ, USA: IEEE; 2024. p. 7530–5. doi:10.1109/CAC59555.2023.10450266.
92. Xu L, Qi G. Function observer based on model compensation control for a fixed-wing UAV. Aerosp Sci Technol. 2024;152:109325. doi:10.1016/j.ast.2024.109325.
93. Pore E, Patle BK, Thorat S. Multi-drone path planning using gradient descent and potential fields for warehouse applications. In: 2024 International Automatic Control Conference (CACS); 2024 Oct 31–Nov 3; Taoyuan, Taiwan. Piscataway, NJ, USA: IEEE; 2024. p. 1–6. doi:10.1109/CACS63404.2024.10773359.
94. El Tin F, Borowczyk A, Sharf I, Nahon M. Turn decision-making for improved autonomous thermalling of unmanned aerial gliders. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS); 2020 Sep 1–4; Athens, Greece. Piscataway, NJ, USA: IEEE; 2020. p. 1368–75. doi:10.1109/icuas48674.2020.9213839.
95. PX4-Gazebo. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/sim_gazebo_gz/.
96. PX4-JMavSim. [cited 2026 Feb 15]. Available from: https://docs.px4.io/main/en/sim_jmavsim/index.html.
97. Microsoft-AirSim. [cited 2026 Feb 15]. Available from: https://www.microsoft.com/en-us/research/project/aerial-informatics-robotics-platform/.
98. Jiang Z, Al Lawati M, Lynch A. Quasi-static state feedback output tracking for a slung load system with rotor drag compensation: PX4-SITL validation. J Intell Rob Syst. 2023;109(2):42. doi:10.1007/s10846-023-01937-9.
99. Moeini A, Rafique MA, Xue Z, Lynch AF, Zhao Q. Disturbance observer-based integral backstepping control for UAVs. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS); 2020 Sep 1–4; Athens, Greece. Piscataway, NJ, USA: IEEE; 2020. p. 382–8. doi:10.1109/icuas48674.2020.9213948.
100. Jayaweera HM, Hanoun S. A dynamic artificial potential field (D-APF) UAV path planning technique for following ground moving targets. IEEE Access. 2020;8:192760–76. doi:10.1109/ACCESS.2020.3032929.

101. Bulka E, Nahon M. Reactive obstacle-avoidance for agile, fixed-wing, unmanned aerial vehicles. Field Robot. 2022;2:1507–66. doi:10.55417/fr.2022048.
102. Wang J. Real-time obstacle avoidance and automatic power line capture control for de-icing UAVs. In: 2024 3rd International Conference on Electronics and Information Technology (EIT); 2024 Sep 20–22; Chengdu, China. Piscataway, NJ, USA: IEEE; 2024. p. 318–22. doi:10.1109/EIT63098.2024.10762259.
103. Zheng Z, Chen H, Lu Y, Xiao K, Wang X. Software-in-the-loop simulation of cooperative fixed-wing unmanned aerial vehicles. In: 2023 2nd Conference on Fully Actuated System Theory and Applications (CFASTA); 2023 July 14–16; Qingdao, China. Piscataway, NJ, USA: IEEE; 2023. p. 302–7. doi:10.1109/CFASTA57821.2023.10243344.
104. Chandran I, Vipin K. Network analysis of decentralized fault-tolerant UAV swarm coordination in critical missions. Drone Syst Appl. 2024;12:1–15. doi:10.1139/dsa-2023-0101.
105. Iriarte I, Gorostiza J, Iglesias I, Lasa J, Calvo-Soraluze H, Sierra B. An overactuated aerial robot based on cooperative quadrotors attached through passive universal joints: modeling, control and 6-DoF trajectory tracking. Robot Auton Syst. 2024;180:104761. doi:10.1016/j.robot.2024.104761.
106. Huang J, Wang F, Hu T. CoFlyers: a universal platform for collective flying of swarm drones. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2023 Oct 1–5; Detroit, MI, USA. Piscataway, NJ, USA: IEEE; 2023. p. 8808–13. doi:10.1109/IROS55552.2023.10342485.
107. Bouček Z, Flídr M. Mission planner for UAV battery replacement. In: 2024 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI); 2024 Sep 4–6; Pilsen, Czech Republic. Piscataway, NJ, USA: IEEE; 2024. p. 1–6. doi:10.1109/MFI62651.2024.10705769.
108. Dang Nguyen K, Nguyen TT. Vision-based software-in-the-loop-simulation for unmanned aerial vehicles using gazebo and PX4 open source. In: 2019 International Conference on System Science and Engineering (ICSSE); 2019 Jul 20–21; Dong Hoi, Vietnam. Piscataway, NJ, USA: IEEE; 2019. p. 429–32. doi:10.1109/icsse.2019.8823322.
109. Carvalho GF, Andrade FAA, Ramos GS, Zachi ARL, de Barros ALF, Pinto MF. Sliding mode controller applied to autonomous UAV operation in marine small cargo transport. IEEE J Miniaturization Air Space Syst. 2023;4(4):345–57. doi:10.1109/JMASS.2023.3296433.
110. Rodriguez-Ramos A, Sampedro C, Bavle H, Milosevic Z, Garcia-Vaquero A, Campoy P. Towards fully autonomous landing on moving platforms for rotary Unmanned Aerial Vehicles. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS); 2017 Jun 13–16; Miami, FL, USA. Piscataway, NJ, USA: IEEE; 2017. p. 170–8. doi:10.1109/ICUAS.2017.7991438.
111. Flores A, Flores G. Development and implementation of a new flight mode called Cartesian flight mode for easy piloting of Muli-rotor UAVs. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS); 2021 Jun 15–18; Athens, Greece. Piscataway, NJ, USA: IEEE; 2021. p. 237–42. doi:10.1109/icuas51884.2021.9476827.
112. Nguyen KD, Ha C. Development of hardware-in-the-loop simulation based on gazebo and pixhawk for unmanned aerial vehicles. Int J Aeronaut Space Sci. 2018;19(1):238–49. doi:10.1007/s42405-018-0012-8.
113. Omar HM. Hardware-In-the-loop simulation of time-delayed anti-swing controller for quadrotor with suspended load. Appl Sci. 2022;12(3):1706. doi:10.3390/app12031706.
114. Thebe KZ, Jamisola RS Jr, Ramalepa LP. A novel approach to control four multi-rotor drones in cooperative paired control using relative Jacobian. Robotica. 2023;41(10):3004–21. doi:10.1017/s0263574723000826.
115. Liu Z, Zhang Y, Chen H, Zhang Z. Incremental control system design and flight tests of a micro-coaxial rotor UAV. Aerosp Sci Technol. 2022;131:107979. doi:10.1016/j.ast.2022.107979.
116. Sandino J, Vanegas F, Gonzalez F, Maire F. Autonomous UAV navigation for active perception of targets in uncertain and cluttered environments. In: 2020 IEEE Aerospace Conference; 2020 Mar 7–14; Big Sky, MT, USA. Piscataway, NJ, USA: IEEE; 2020. p. 1–12. doi:10.1109/aero47225.2020.9172808.
117. Schofield OB, Iversen N, Ebeid E. Autonomous power line detection and tracking system using UAVs. Microprocess Microsyst. 2022;94(6):104609. doi:10.1016/j.micpro.2022.104609.
118. Hancer M, Bitirgen R, Bayezit I. Designing 3-DOF hardware-In-the-loop test platform controlling multirotor vehicles. IFAC-PapersOnLine. 2018;51(4):119–24. doi:10.1016/j.ifacol.2018.06.058.

119. Wang H, Azaizia D, Lu C, Zhang B, Zhao X, Liu Y. Hardware in the loop based 6DoF test platform for multi-rotor UAV. In: 2017 4th International Conference on Systems and Informatics (ICSAI); 2017 Nov 11–13; Hangzhou, China: IEEE; 2018. p. 1693–7. doi:10.1109/ICSAI.2017.8248556.
120. Kai JM, Hamel T, Samson C. A unified approach to fixed-wing aircraft path following guidance and control. Automatica. 2019;108(5):108491. doi:10.1016/j.automatica.2019.07.004.
121. Giovagnola J, Moro Megías JB, Fernández MM, Cuéllar MP, Morales Santos DP. AirLoop: a simulation framework for testing of UAV services. IEEE Access. 2023;11:23309–25. doi:10.1109/ACCESS.2023.3253788.
122. Bulka E, Nahon M. Autonomous fixed-wing aerobatics: from theory to flight. In: 2018 IEEE International Conference on Robotics and Automation (ICRA); 2018 May 21–25; Brisbane, QLD, Australia. Piscataway, NJ, USA: IEEE; 2018. p. 6573–80. doi:10.1109/ICRA.2018.8460610.
123. Prasad D, Adhithiyan ST, Thanmai K, Banda G. PAVeDS: a synthetic dataset for developing autonomous personal aerial vehicles. IEEE Access. 2023;11:101556–66. doi:10.1109/ACCESS.2023.3315768.
124. Pieczyński D, Ptak B, Kraft M, Piechocki M, Aszkowski P. A fast, lightweight deep learning vision pipeline for autonomous UAV landing support with added robustness. Eng Appl Artif Intell. 2024;131(2):107864. doi:10.1016/j.engappai.2024.107864.
125. He L, Aouf N, Song B. Explainable Deep Reinforcement Learning for UAV autonomous path planning. Aerosp Sci Technol. 2021;118(7):107052. doi:10.1016/j.ast.2021.107052.
126. Tullu A, Endale B, Wondosen A, Hwang HY. Machine learning approach to real-time 3D path planning for autonomous navigation of unmanned aerial vehicle. Appl Sci. 2021;11(10):4706. doi:10.3390/app11104706.
127. Luo J, Oracevic A. Adaptive machine learning for efficient anomaly detection in autonomous UAVs swarm operations. In: 2024 International Symposium on Networks, Computers and Communications (ISNCC); 2024 Oct 22–25; Washington, DC, USA. Piscataway, NJ, USA: IEEE; 2024. p. 1–8. doi:10.1109/ISNCC62547.2024.10758945.
128. Wang C, Zhong Z, Xiang X, Zhu Y, Wu L, Yin D, et al. UAV path planning in multi-task environments with risks through natural language understanding. Drones. 2023;7(3):147. doi:10.3390/drones7030147.
129. Llanes C, Netter J, Vamvoudakis KG, Coogan S. Experimental validation on aerial vehicles of real-time motion planning with continuous-time Q-learning. IFAC-PapersOnLine. 2023;56(3):31–6. doi:10.1016/j.ifacol.2023.11.006.
130. Baldini A, Felicetti R, Ferracuti F, Freddi A, Iarlori S, Monteriù A. Real-time propeller fault detection for multirotor drones based on vibration data analysis. Eng Appl Artif Intell. 2023;123(9):106343. doi:10.1016/j.engappai.2023.106343.
131. Khan A, Campos JR, Ivaki N, Madeira H. A machine learning driven fault tolerance mechanism for UAVs' flight controller. In: 2023 IEEE 28th Pacific Rim International Symposium on Dependable Computing (PRDC); 2023 Oct 24–27; Singapore. Piscataway, NJ, USA: IEEE; 2023. p. 217–27. doi:10.1109/prdc59308.2023.00034.
132. Choutri K, Lagha M, Meshoul S, Batouche M, Bouzidi F, Charef W. Fire detection and geo-localization using UAV's aerial images and yolo-based models. Appl Sci. 2023;13(20):11548. doi:10.3390/app132011548.
133. Malle NH, Nyboe FF, Ebeid ESM. Onboard powerline perception system for UAVs using mmWave radar and FPGA-accelerated vision. IEEE Access. 2022;10:113543–59. doi:10.1109/ACCESS.2022.3217537.
134. Hernández D, Cecilia JM, Cano JC, Calafate CT. Flood detection using real-time image segmentation from unmanned aerial vehicles on edge-computing platform. Remote Sens. 2022;14(1):223. doi:10.3390/rs14010223.
135. Zoric F, Franchi A, Orsag M, Kovacic Z, Gabellieri C. Towards instance segmentation-based litter collection with multi-rotor aerial vehicle. In: 2024 International Conference on Unmanned Aircraft Systems (ICUAS); 2024 Jun 4–7; Chania-Crete, Greece. Piscataway, NJ, USA: IEEE; 2024. p. 631–7.
136. Zhao W, Li L, Zhan H, Wang Y, Fu Y. Applying large language model to a control system for multi-robot task assignment. Drones. 2024;8(12):728. doi:10.3390/drones8120728.
137. Yoon K, Park D, Yim Y, Kim K, Yang SK, Robinson M. Security authentication system using encrypted channel on UAV network. In: 2017 First IEEE International Conference on Robotic Computing (IRC); 2017 Apr 10–12; Taichung, Taiwan. Piscataway, NJ, USA: IEEE; 2017. p. 393–8. doi:10.1109/IRC.2017.56.
138. Allouch A, Cheikhrouhou O, Koubaa A, Khalgui M, Abbes T. MAVSec: securing the MAVLink protocol for ardupilot/PX4 unmanned aerial systems. In: 2019 15th International Wireless Communications & Mobile Computing

Conference (IWCMC); 2019 Jun 24–28; Tangier, Morocco. Piscataway, NJ, USA: IEEE; 2019. p. 621–8. doi:10.1109/iwcmc.2019.8766667.
139. Sabuwala N, Daruwala RD. Securing unmanned aerial vehicles by encrypting MAVLink protocol. In: 2022 IEEE Bombay Section Signature Conference (IBSSC); 2022 Dec 8–10; Mumbai, India. Piscataway, NJ, USA: IEEE; 2023. p. 1–6. doi:10.1109/IBSSC56953.2022.10037546.
140. Romagnoli R, Krogh BH, de Niz D, Hristozov AD, Sinopoli B. Software rejuvenation for safe operation of cyber-physical systems in the presence of run-time cyberattacks. IEEE Trans Control Syst Technol. 2023;31(4):1565–80. doi:10.1109/TCST.2023.3236470.
141. Ali Khan N, Jhanjhi NZ, Nawaz Brohi S, Ali Almazroi A, Ali Almazroi A. A secure communication protocol for unmanned aerial vehicles. Comput Mater Contin. 2022;70(1):601–18. doi:10.32604/cmc.2022.019419.
142. Jeong S, Park E, Seo KU, Do Yoo J, Kim HK. MUVIDS: false MAVLink injection attack detection in communication for unmanned vehicles. In: Proceedings Third International Workshop on Automotive and Autonomous Vehicle Security. Reston, VA, USA: Internet Society; 2021. p. 1–6. doi:10.14722/autosec.2021.23036.
143. Feng Z, Guan N, Lv M, Liu W, Deng Q, Liu X, et al. An efficient UAV hijacking detection method using onboard inertial measurement unit. ACM Trans Embed Comput Syst. 2018;17(6):1–19. doi:10.1145/3289390.
144. Whelan J, Almehmadi A, El-Khatib K. Artificial intelligence for intrusion detection systems in Unmanned Aerial Vehicles. Comput Electr Eng. 2022;99(5):107784. doi:10.1016/j.compeleceng.2022.107784.
145. Han L, Xun Y, Liu J, Benslimane A, Zhang Y. DP-Authentication: a novel deep learning based drone pilot authentication scheme. Ad Hoc Netw. 2023;147:103180. doi:10.1016/j.adhoc.2023.103180.
146. Hadi HJ, Cao Y, Li S, Xu L, Hu Y, Li M. Real-time fusion multi-tier DNN-based collaborative IDPS with complementary features for secure UAV-enabled 6G networks. Expert Syst Appl. 2024;252:124215. doi:10.1016/j.eswa.2024.124215.
147. Veyna U, Garcia-Nieto S, Simarro R, Salcedo JV. Quadcopters testing platform for educational environments. Sensors. 2021;21(12):4134. doi:10.3390/s21124134.
148. Kotarski D, Piljek P, Pranjić M, Grlj CG, Kasać J. A modular multirotor unmanned aerial vehicle design approach for development of an engineering education platform. Sensors. 2021;21(8):2737. doi:10.3390/s21082737.
149. Kotarski D, Pranjić M, Alharbat A, Piljek P, Bjažić T. Multirotor UAV-a multidisciplinary platform for teaching mechatronics engineering. Sensors. 2025;25(4):1007. doi:10.3390/s25041007.
150. Panza S, Invernizzi D, Giurato M, Lovera M. Design and characterization of the 2DoF Drone: a multirotor platform for education and research. IFAC-PapersOnLine. 2021;54(12):32–7. doi:10.1016/j.ifacol.2021.11.006.
151. Invernizzi D, Panza S, Giurato M, Yang G, Chen K, Lovera M, et al. Integration of experimental activities into remote teaching using a quadrotor test-bed. IFAC-PapersOnLine. 2021;54(12):49–54. doi:10.1016/j.ifacol.2021.11.009.
152. Jiang Y, Yuan J, Sun L, Song HH. Development of a laboratory platform for UAV cybersecurity education. In: Proceedings of the 2021 ASEE Virtual Annual Conference Content Access. Washington, DC, USA: ASEE; 2021. doi:10.18260/1-2--36958.
153. Cañas JM, Martín-Martín D, Arias P, Vega J, Roldán-Álvarez D, García-Pérez L, et al. Open-source drone programming course for distance engineering education. Electronics. 2020;9(12):2163. doi:10.3390/electronics9122163.
154. Hong JH, Shin HS, Tsourdos A. A design of a short course with COTS UAV system for higher education students. IFAC-PapersOnLine. 2019;52(12):466–71. doi:10.1016/j.ifacol.2019.11.287.
155. Schmittle M, Lukina A, Vacek L, Das J, Buskirk CP, Rees S, et al. OpenUAV: a UAV testbed for the CPS and robotics community. In: 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS); 2018 Apr 11–13; Porto, Portugal. Piscataway, NJ, USA: IEEE; 2018. p. 130–9. doi:10.1109/iccps.2018.00021.
156. Ali MI, Tavoulareas T, Chen Z, Cescon M. Project-based learning using a quadrotor testbed. In: 2024 IEEE Conference on Control Technology and Applications (CCTA); 2024 Aug 21–23; Newcastle upon Tyne, UK. Piscataway, NJ, USA: IEEE; 2024. p. 452–7. doi:10.1109/CCTA60707.2024.10666555.
157. Castaldi P, Mimmo N. An experience of project based learning in aerospace engineering. IFAC-PapersOnLine. 2019;52(12):484–9. doi:10.1016/j.ifacol.2019.11.290.

158. Rey L, Bernardos AM, Dobrzycki AD, Carramiñana D, Bergesio L, Besada JA, et al. A performance analysis of you only look once models for deployment on constrained computational edge devices in drone applications. Electronics. 2025;14(3):638. doi:10.3390/electronics14030638.
159. Douklias A, Karagiannidis L, Misichroni F, Amditis A. Design and implementation of a UAV-based airborne computing platform for computer vision and machine learning applications. Sensors. 2022;22(5):2049. doi:10.3390/s22052049.
160. Okudera R, Li Y, Matsubara Y, Takada H. A COTS-based lightweight, low-power and versatile companion computer for nano UAVs. Sens Transd. 2025;268(1):37–44.
161. Dimmig CA, Silano G, McGuire K, Gabellieri C, Hönig W, Moore J, et al. Survey of simulators for aerial robots: an overview and in-depth systematic comparisons [survey]. IEEE Robot Autom Mag. 2025;32(2):153–66. doi:10.1109/MRA.2024.3433171.
162. Perez-Segui R, Arias-Perez P, Melero-Deza J, Fernandez-Cortizas M, Perez-Saura D, Campoy P. Bridging the gap between simulation and real autonomous UAV flights in industrial applications. Aerospace. 2023;10(9):814. doi:10.3390/aerospace10090814.
163. Kwon YM, Yu J, Cho BM, Eun Y, Park KJ. Empirical analysis of MAVLink protocol vulnerability for attacking unmanned aerial vehicles. IEEE Access. 2018;6:43203–12. doi:10.1109/ACCESS.2018.2863237.
164. Taylor M, Boubin J, Chen H, Stewart C, Qin F. A study on software bugs in unmanned aircraft systems. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS); 2021 Jun 15–18; Athens, Greece. Piscataway, NJ, USA: IEEE; 2021. p. 1439–48. doi:10.1109/icuas51884.2021.9476844.
165. Taylor M, Aurand J, Qin F, Wang X, Henry B, Zhang X. SA4U: practical static analysis for unit type error detection. In: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering; 2022 Oct 10; Rochester, MI, USA. New York, NY, USA: ACM; 2022. p. 1–11. doi:10.1145/3551349.3556937.
166. Schiller N, Chlosta M, Schloegel M, Bars N, Eisenhofer T, Scharnowski T, et al. Drone security and the mysterious case of DJI's DroneID. In: Proceedings 2023 Network and Distributed System Security Symposium; 2023 Feb 27–Mar 3; San Diego, CA, USA. Reston, VA, USA: Internet Society; 2023. p. 1–17. doi:10.14722/ndss.2023.24217.