DEVELOPMENT AND EXPERIMENTAL VALIDATION OF A GENERAL-PURPOSE FLIGHT CONTROLLER FOR AUTONOMOUS AEROSPACE PLATFORMS

CRISTIAN-TUDOR GHEORMA[1,2], KHALED HACHEM[1,2]

Abstract. This paper presents the development and first experimental validation stage of a general-purpose flight controller intended for use across multiple autonomous aerial platforms, including multicopters, fixed wing aircraft, vertical take off and landing systems, and thrust vector controlled vehicles. The work is based on the development of the proposed Flight Controller, a modular embedded control platform designed to integrate the essential functions required for autonomous flight, including real time data acquisition, onboard signal processing, attitude estimation, actuator command generation, communication, data storage, and experimental validation of control algorithms. The controller was conceived as a flexible research and development platform, combining inertial sensing, pressure measurement, satellite navigation capability, external communication interfaces, actuator outputs, protected power regulation, and modular expansion options within a unified architecture. The first version of the system focuses on validating the electrical design, sensor integration, software structure, communication interfaces, and basic control capability required for stable operation in flight conditions. This version has already been successfully tested on a quadcopter, demonstrating its ability to operate as a multicopter flight control unit and confirming the functionality of the complete sensing, processing, and actuation chain under real operating conditions. Through this initial implementation and flight testing campaign, the project establishes the foundation for a deployable, configurable, and general-purpose flight controller suitable for future aerospace research, autonomous vehicle development, and thrust vector control applications.

Key words: flight controller 1, autonomous aerial vehicles 2, multicopter 3, fixed-wing aircraft 4, vertical take-off and landing 5, embedded systems 6, control algorithms 7.

[1] INCAS - National Institute for Aerospace Research "Elie Carafoli"
[2] National University of Science and Technology POLITEHNICA, Bucharest

Ro. J. Techn. Sci. – Appl. Mechanics, Vol. 71, N° 1, P. 77–91, Bucharest, 2026

DOI:10.59277/RJTS-AM.2026.1.05

1. INTRODUCTION

The development of autonomous aerial systems has created a strong need for compact and reliable flight-control platforms capable of operating under real time constraints while remaining adaptable to different vehicle configurations. Multicopters, fixed wing aircraft, vertical take off and landing systems, and thrust vector controlled vehicles differ in geometry, propulsion, and flight dynamics, yet they share the same basic control chain: onboard sensing, state estimation, control-law execution, and actuator command generation. This common functional basis makes the development of a general-purpose flight controller both technically relevant and practically useful [1–4].

In many research environments, controllers are developed for one specific platform and later become difficult to reuse in other aerospace applications. Such fragmentation leads to repeated design effort, different software bases, and nonuniform testing procedures. By contrast, modern open autopilot ecosystems have shown the value of configurable architectures that can support several vehicle classes from a common hardware and software foundation [1], [4]. For research and development work, however, there is still a need for dedicated experimental platforms that allow direct access to sensing, control, communication, and expansion layers while remaining simple enough to modify and validate.

The present work addresses this need through the development of a general-purpose flight controller intended for experimental autonomous aerospace applications. The controller was conceived as a modular embedded platform able to support inertial sensing, pressure measurement, satellite navigation, onboard logging, actuator control, external communication, and future platform-specific extensions. Rather than being restricted to a single vehicle, the design was meant to serve as a reusable control core for several aerial systems, with the main differences between platforms handled through firmware configuration and control allocation.

The first implementation has already been validated on a quadcopter platform. This choice is appropriate for an initial validation stage because a quadcopter requires continuous attitude stabilization, fast actuator updates, and reliable processing of inertial data [2], [5]. In the present paper, the quadcopter is used as the first experimental vehicle for proving the feasibility of the controller architecture. In parallel, bench-level measurements were also performed in order to characterize timing behavior, estimation performance, and control response without requiring an additional flight campaign.

2. SYSTEM REQUIREMENTS AND DESIGN OBJECTIVES

The proposed flight controller was developed as a compact and configurable embedded platform for experimental autonomous aerial vehicles. Its main role is to provide the basic functions required for flight control: sensor acquisition, onboard processing, attitude estimation, actuator command generation, communication, and data storage. Since these functions are common to several vehicle types, the controller was designed as a general-purpose platform rather than as a system dedicated only to one aerial configuration.

A major design objective was modularity. Existing flight-control architectures show that a common hardware and software base can be adapted to multicopters, fixed wing aircraft, vertical take off and landing vehicles, and other autonomous systems [1], [3], [4]. In this work, the first practical validation was performed on a quadcopter, because this platform requires continuous attitude stabilization, fast motor command updates, and reliable sensor feedback [2], [5].

The controller was also designed with future aerospace applications in mind. For this reason, the system includes sensing interfaces, navigation capability, onboard logging, actuator outputs, communication ports, programming access, protected power regulation, and expansion options. These features allow the controller to be used first as a multicopter flight-control unit, while preserving the possibility of later adaptation to fixed wing, vertical take off and landing, or thrust vector controlled vehicles [3], [6].

Table 1

Main system requirements and design objectives

Requirement	**Design objective**	**Purpose**
Sensor acquisition	Integrate motion, pressure, and navigation data	Support attitude estimation and flight control
Actuator outputs	Provide multiple command channels	Control motors, servos, or future actuation systems
Onboard processing	Run estimation and control algorithms locally	Enable real time operation
Data storage	Record flight and sensor data	Support post-flight analysis and tuning
Communication	Provide external interfaces	Allow telemetry, configuration, and integration
Power protection	Regulate and protect internal supply lines	Improve reliability during operation
Modularity	Keep expansion interfaces available	Support future vehicle configurations

Based on these requirements, the first version of the controller was designed as a validation-oriented platform. Its immediate objective was to demonstrate stable operation on a quadcopter, while its long-term objective is to provide a reusable architecture for experimental autonomous aerospace systems.

3. HARDWARE ARCHITECTURE

The hardware architecture of the controller was developed around a central high-performance microcontroller supported by dedicated sensing, communication, storage, and power subsystems. The purpose of the board was not merely to host a processor, but to integrate all functional layers required by a self-contained flight-control unit.

The processing core was connected to multiple serial, timer, and digital interfaces so that the board could simultaneously handle sensor data, communication channels, actuator outputs, and auxiliary functions. Particular attention was given to the allocation of microcontroller pins, since the usefulness of a flight controller depends strongly on clean peripheral separation and predictable routing of time-critical signals. The pin mapping used in the design is shown in Fig. 1.

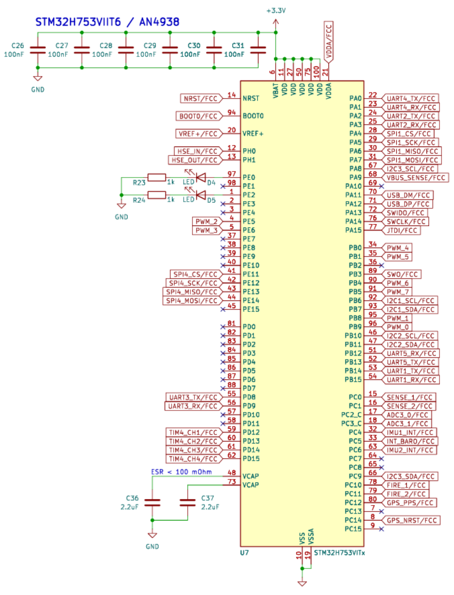

Fig. 1 – Pin mapping and peripheral allocation of the main microcontroller used.

The sensing layer was designed to include redundant attitude-related information through more than one inertial sensing source, complemented by a pressure sensor for altitude-related measurements and a satellite navigation receiver for position and timing support. This combination allows the controller to serve both as a stabilization unit and as a future navigation-capable flight computer. In practical terms, the sensing architecture was selected so that short-term dynamics, long-term attitude correction, and pressure-derived vertical information could all be processed on the same platform [7–10].

The board also integrates nonvolatile data storage, which is essential for flight testing and post-processing. Logging capability is particularly important during early controller development, when raw sensor data, filtered estimates, control outputs, and execution timing must all be examined in order to refine the firmware and tune the control loops. In the same spirit, the board includes a programming and debugging interface, allowing fast firmware updates and direct development access.

From the actuator point of view, the hardware provides multiple output channels intended primarily for propulsion control in multicopter applications, while preserving the possibility of extension to servomechanisms or other actuation layouts. Additional expansion connectors were included in the design so that future sensors, external processors, or mission-specific modules can be integrated without requiring a complete redesign of the control board.

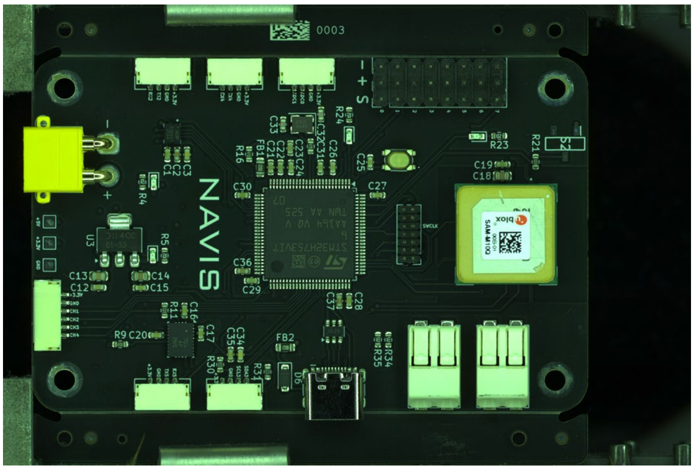

Fig. 1 – Manufactured prototype of the developed flight controller during production.

The power subsystem was treated as a major part of the architecture. A flight controller must remain electrically stable under varying supply conditions, and for this reason the board includes regulated internal power domains, protection against improper power input, local filtering for sensitive measurement lines, and separate consideration for logic and analog sections. This was particularly important because the controller was intended not only for laboratory use, but also for onboard installation in vibrating and electrically noisy environments.

The developed board is shown during production in Fig. 2, where the compact layout, interface grouping, and physical integration of the main functional blocks can be observed.

4. SOFTWARE AND FIRMWARE ARCHITECTURE

The firmware was organized as a layered architecture in which sensing, estimation, control, and communication were separated into functional modules. This approach simplifies both debugging and future reuse on other vehicle types. At the lowest level, the firmware handles peripheral initialization, sensor communication, and timed acquisition. Above this layer, the estimation and control routines are executed periodically inside the main control cycle.

The operational flow of the controller can be summarized as follows:

1. acquire raw sensor data;
2. preprocess or calibrate sensor readings;
3. estimate the vehicle attitude;
4. read pilot or mission setpoints;
5. execute the outer and inner control loops;
6. generate actuator commands;
7. transmit and log selected variables.

The main control task was implemented as a periodic loop. Deterministic execution of this loop is essential because excessive jitter or missed deadlines can degrade control performance and, in extreme cases, destabilize the vehicle. For this reason, one of the experimental measurements included in the present paper is the execution-time characterization of the control loop.

Sensor processing and attitude estimation

The controller uses inertial data as the primary source of short-term motion information. Accelerometer and gyroscope measurements are combined in order to obtain a more reliable attitude estimate than either source could provide alone. The gyroscope captures fast motion but tends to drift when integrated over time, whereas the accelerometer provides a gravity-based reference but is sensitive to vibration and translational disturbances. To combine these two sources, a Kalman filter was implemented in the estimation layer [7], [11].

In its simplified form, the estimation model may be written as:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \tag{1}$$

$$z_k = Hx_k + v_k, \tag{2}$$

where x_k is the state vector, z_k is the measurement vector, and w_k and v_k represent process and measurement noise, respectively. In the present application, the filter was used to reduce noise and drift in roll and pitch estimation while preserving sufficient responsiveness for control.

Control architecture

The control system was organized as a cascaded proportional-integral-derivative structure, which is a common and practical solution for multicopter stabilization [2], [5]. The outer loop regulates attitude by comparing the desired roll and pitch angles with the estimated angles and generating desired angular-rate commands. The inner loop then regulates the actual angular rates to produce actuator-level commands.

For one attitude axis, the outer loop may be expressed as

$$e_{\phi} = \phi_{ref} - \phi \, , \tag{3}$$

$$p_{ref} = K_{p,\phi} e_{\phi} + K_{i,\phi} \int e_{\phi} \mathrm{d}t + K_{d,\phi} \frac{de_{\phi}}{dt} \, , \tag{4}$$

where ϕ_{ref} is the desired angle, ϕ is the estimated angle, and p_{ref} is the commanded angular rate. The inner loop then uses the rate error

$$e_p = p_{ref} - p \, , \tag{5}$$

to compute a correction of the form

$$u_p = K_{p,p} e_p + K_{i,p} \int e_p \mathrm{d}t + K_{d,p} \frac{de_p}{dt} \, , \tag{6}$$

Communication and logging

The firmware also includes communication and logging routines, both of which are essential during development. Communication interfaces are used for setup, debugging, and real-time monitoring, while onboard logging preserves data for later analysis. In the experimental phase, the logged variables included raw inertial signals, filtered attitude estimates, setpoints, control outputs, and control-loop timing.

Overall, the firmware was designed to remain modular enough for future adaptation to other vehicle types. While the sensing and estimation layers can be preserved with minor changes, the control allocation and high-level flight logic can later be modified according to the dynamics and actuation method of the target platform.

5. CONTROL STRATEGY FOR QUADCOPTER VALIDATION

The first practical validation of the controller was performed on a quadcopter platform. This vehicle type was selected because it represents a demanding but manageable test case for a newly developed controller. In a quadcopter, the flight controller must continuously regulate roll, pitch, and yaw by commanding the four propulsion units according to the desired vehicle motion and the estimated current state.

The adopted control strategy was based on a cascaded proportional-integral-derivative structure. The outer controller acts on attitude, while the inner controller acts on angular rate. This arrangement is well suited to multicopters because it separates the slower kinematic objective of reaching a desired orientation from the faster dynamic objective of damping and correcting body rates [2], [5].

For an X-type quadcopter configuration, the actuator commands were obtained through a standard motor-mixing stage. If T denotes collective thrust and τ_ϕ, τ_θ, and τ_ψ denote roll, pitch, and yaw control terms, then the motor commands may be written in simplified form as

$$m_1 = T + \tau_\phi + \tau_\theta - \tau_\psi, \tag{7}$$

$$m_2 = T - \tau_\phi + \tau_\theta + \tau_\psi, \tag{8}$$

$$m_3 = T - \tau_\phi - \tau_\theta - \tau_\psi, \tag{9}$$

$$m_4 = T + \tau_\phi - \tau_\theta + \tau_\psi. \tag{10}$$

This allocation transforms the controller outputs into four individual motor commands. In practice, saturation and safety limits were applied in order to keep the commands within allowable ranges.

The use of this platform allowed the controller to be evaluated as a true flight-control unit rather than only as an electronics demonstrator. At the same time, the same control structure can later be adapted to other vehicles by modifying the actuator-allocation layer and the corresponding high-level control logic.

6. EXPERIMENTAL SETUP

The experimental work presented in this paper combines two complementary stages. The first stage consists of the successful integration of the controller on a quadcopter platform, which verified that the developed unit can function onboard a real aerial vehicle. The second stage consists of bench-level characterization tests performed with the flight controller connected to a laptop, which allowed quantitative analysis of timing, estimation, and control behavior even in the absence of a new flight campaign.

Fig. 3 – Rendering of the quadcopter configuration used as the first validation platform.

The controller was mounted on a quadcopter platform representative of a small multirotor test vehicle. The platform included four propulsion units, electronic speed controllers, a power distribution arrangement, a battery pack, and the developed flight controller mounted near the vehicle center. Care was taken to ensure correct mechanical placement and proper electrical interfacing between the board and the propulsion system.

The purpose of the quadcopter validation was to confirm the correct operation of the full sensing-processing-actuation chain under real operating conditions. This stage demonstrated that the controller could acquire onboard data, estimate attitude, generate motor commands, and act as the main control unit of the vehicle.

The tested quadcopter platform is illustrated in Fig. 3 and Fig. 4. The rendering in Fig. 3 shows the intended onboard integration of the controller, while the photograph in Fig. 4 presents the actual vehicle used for the experimental validation.

Fig. 4 – Photograph of the actual quadcopter test platform equipped with the developed controller

7. RESULTS

The results presented in this section correspond to a bench-level functional validation of the developed flight controller. The controller was not mounted on the quadcopter during this measurement sequence, and no free-flight data are analyzed in this section. Instead, the board was tested as a standalone autopilot unit, connected to the laptop for data acquisition and observation. Manual disturbances were introduced by physically moving and disturbing the printed circuit board in order to excite the inertial sensors and observe how the estimation and control algorithms reacted.

During the test, the main control loop was configured to run at 650 Hz, corresponding to an update period of approximately 1.54 ms. This frequency was used for sensor acquisition, Kalman-based attitude estimation, cascaded PID control computation, and output command generation. For the control-response observation, roll and pitch reference values of 10 degrees were imposed. Since the board was tested without the drone, the resulting plots do not represent the physical response of the aircraft. They represent the internal response of the controller to measured attitude variations and imposed setpoints.

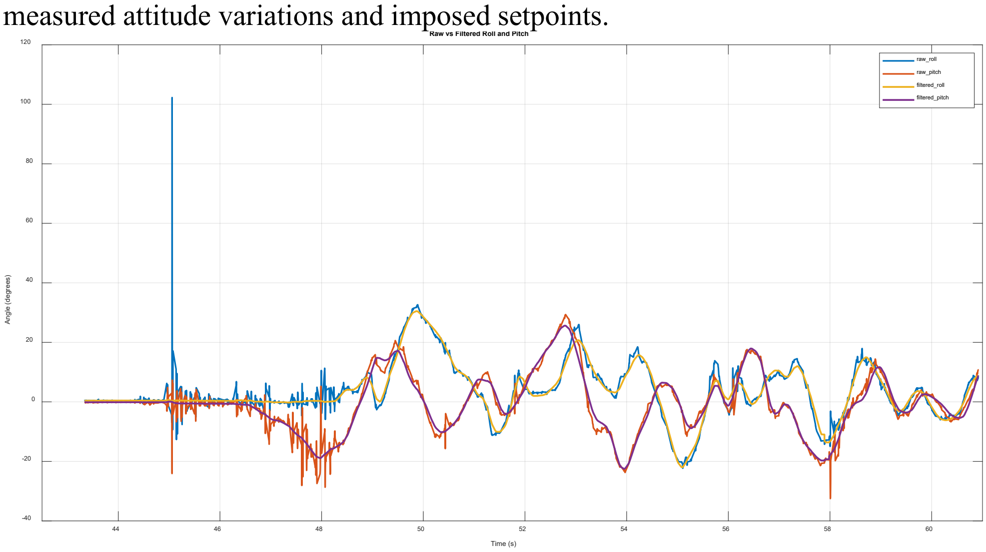

Fig. 5 – Raw vs filtered Roll and Pitch

Figure 5 presents the raw and Kalman-filtered roll and pitch angles recorded during the bench test. The raw signals show visible high-frequency variations and short-duration peaks caused by the manual disturbances applied to the board and by the noise present in the inertial measurements. The filtered signals follow the same general attitude evolution, but with reduced abrupt variations. This confirms that the Kalman filter attenuates measurement noise while preserving the useful motion information required by the control algorithm.

Figure 6 presents the roll and pitch rate commands generated by the outer attitude loop. During the test, roll and pitch setpoints of 10 degrees were applied. In the cascaded control structure, the outer loop compares the imposed attitude reference with the estimated attitude and generates the corresponding angular-rate command. The rate-command signals therefore show the intermediate response of the controller between the attitude error and the final actuator outputs.

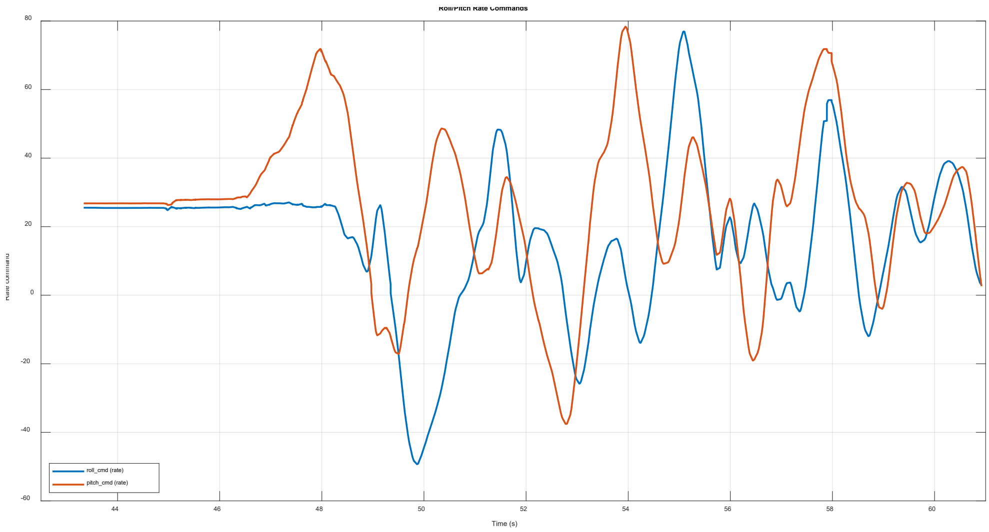

Fig. 6 – Outer attitude loop Roll and Pitch rate commands.

The rate commands vary as the board is disturbed, which indicates that the outer loop reacts to the difference between the 10 degree references and the estimated attitude. Changes in sign and amplitude appear when the estimated attitude moves relative to the imposed setpoint.

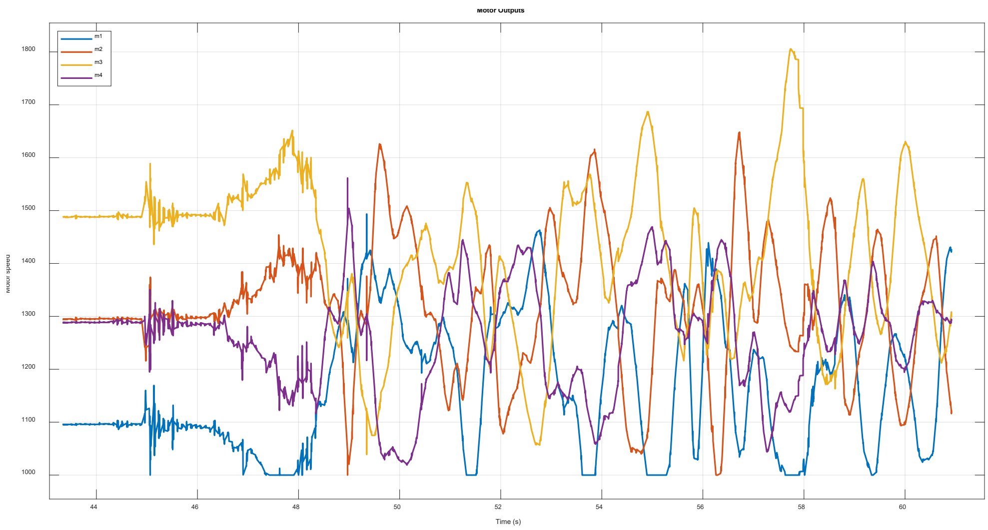

Fig. 7 – Actuator outputs PWM signal.

Figure 7 shows the motor command outputs generated by the controller. These values were computed by the firmware and represent the commands that would be sent to the four propulsion units in a quadcopter configuration. Since no drone was connected during this test, the plot must be interpreted as a validation of the control and mixing logic, not as a measurement of actual motor or vehicle behavior.

8. DISCUSSION

The results obtained in this work confirm that the developed controller is more than a conceptual hardware prototype. The successful integration on a quadcopter platform shows that the board can operate as a practical flight-control unit, while the bench characterization provides a more detailed view of its internal behavior. In particular, the timing measurements support the real time viability of the firmware, and the estimation results show that the sensing and filtering layers provide sufficiently stable information for control use.

The comparison between raw and filtered attitude signals is especially important. In aerial vehicles, control performance depends strongly on the quality of the estimated state. A noisy estimate can lead to excessive controller activity, while an overly slow estimate can delay the correction of disturbances. The bench results suggest that the implemented Kalman filter achieves a useful compromise between smoothing and responsiveness.

The cascaded proportional-integral-derivative response also supports the selected control structure. This architecture remains widely used in practical multicopter applications because it is relatively simple to tune and offers a clear separation between attitude-level and rate-level dynamics [2], [5]. The present implementation behaved consistently in bench testing, which indicates that the controller logic has been correctly implemented and is suitable for continued flight-oriented tuning.

At the same time, the present validation remains limited in scope. The quadcopter integration confirms practical operability, but the results presented in detail are bench measurements rather than a complete flight-data campaign. For this reason, the paper should be interpreted as a first-stage validation study. It demonstrates architectural correctness and functional feasibility, but not yet full performance characterization under all flight regimes or across all intended vehicle classes.

9. EXTENSION TOWARD GENERAL-PURPOSE AEROSPACE USE

Although the first experimental application of the controller was a quadcopter, the system was designed from the beginning with broader aerospace applicability in mind. This is evident in both the hardware and firmware architecture. On the hardware side, the board offers multiple sensing, communication, and actuator interfaces that are not restricted to a single vehicle layout. On the firmware side, the separation between sensing, estimation, control, and output allocation makes it possible to adapt the same platform to different vehicle dynamics with limited restructuring.

For fixed wing aircraft, the controller could be used by preserving the sensing and estimation layers while replacing the multicopter mixing stage with control-surface and throttle allocation. For vertical take off and landing vehicles, the same controller could support both hover and forward-flight modes through a higher-level mode-management strategy and different actuator mappings [3]. For thrust vector controlled vehicles, the controller could also serve as an embedded stabilization and data-acquisition unit, provided that the actuation and vehicle models are adapted to the corresponding flight regime [6].

This broader applicability is one of the main motivations behind the present design. Rather than producing a controller tightly coupled to one experimental platform, the work aims to establish a common embedded core that can be progressively extended across several aerospace projects. In this sense, the quadcopter validation represents the first proof of functionality, while future projects would further confirm the general-purpose nature of the architecture.

10. CONCLUSION

This paper presented the development and initial validation of a general-purpose flight controller intended for experimental autonomous aerial platforms. The controller was conceived as a modular embedded system integrating sensing, processing, communication, data logging, actuator control, and protected power regulation within a single compact architecture.

The first implementation was successfully integrated on a quadcopter platform, demonstrating that the controller can operate as a practical multicopter flight-control unit. In addition, a bench-level characterization campaign was performed in order to evaluate control-loop execution time, Kalman-filtered attitude estimation, static attitude-estimation stability, and the response of the cascaded proportional-integral-derivative controller structure. These results support the functional correctness of the implemented sensing-processing-actuation chain and show that the controller provides a viable basis for continued aerospace control development.

Although the current validation stage is limited mainly to multicopter use and bench characterization, the hardware and firmware architecture were developed with broader adaptability in mind. The presented work therefore represents a first but important step toward a reusable flight-control platform that can later be extended to fixed wing aircraft, vertical take off and landing systems, and thrust vector controlled aerospace vehicles.

Received on June 16, 2026

REFERENCES

1. PX4 DEVELOPMENT TEAM, *PX4 Autopilot User Guide*, Dronecode Foundation, online documentation, accessed January 2026.
2. PX4 DEVELOPMENT TEAM, *Controller Diagrams*, Dronecode Foundation, online documentation, accessed January 2026.
3. PX4 DEVELOPMENT TEAM, *Vertical Take Off and Landing Vehicles*, Dronecode Foundation, online documentation, accessed January 2026.
4. ARDUPILOT DEVELOPMENT TEAM, *ArduPilot – Versatile, Trusted, Open*, ArduPilot online documentation, accessed January 2026.
5. BRESSAN, G., INVERNIZZI, D., PANZA, S., LOVERA, M., *Attitude control of multirotor unmanned aerial vehicles: cascade P/PID vs PI-like architecture*, Politecnico di Milano, Italy, 2019.
6. DEMEMES, T., *Active Attitude Control of Rocket Launchers*, KTH Royal Institute of Technology, Degree Project in Space Technology, Paris, France, 2024.
7. STMICROELECTRONICS, *Application Note AN4938 – Recommendations for Hardware Development Based on STM32 Microcontrollers*, STMicroelectronics, online technical documentation, accessed January 2026.
8. BOSCH SENSORTEC, *BMP390 Digital Pressure Sensor Datasheet*, Bosch Sensortec, online technical documentation, accessed January 2026.
9. BOSCH SENSORTEC, *BNO055 Intelligent 9-axis Absolute Orientation Sensor Datasheet*, Bosch Sensortec, online technical documentation, accessed January 2026.
10. U-BLOX, *SAM-M10Q Standard Precision GNSS Antenna Module Data Sheet*, u-blox AG, online technical documentation, accessed January 2026.
11. LIANG, W., *Attitude Estimation of Quadcopter through Extended Kalman Filter*, Lehigh University, Graduate Thesis, 2017.