Smart Technologies Academic Press

**Jordanian Journal of Informatics and Computing**
https://www.jjic.thestap.com/

# Development of a Stable Quadcopter System Using Open-Source Autopilot

**Ali A. Abed[1], Abdullah Hussein Talib[2], Elaf J. Majeed[1], Mahmood A. Al-Shareeda[3, 4*]**

[1]*Department of Mechatronics Engineering, University of Basrah, Basra, Iraq.*
[2]*Artificial Intelligence Engineering Department, College of Engineering, Al Maaqal University, Basra, Iraq.*
[3]*Department of Electronic Technologies, Basra Technical Institute, Southern Technical University, Basra, Iraq.*
[4]*College of Engineering, Al-Ayen University, Thi-Qar, 64001, Iraq.*

## ARTICLE INFO

**Article History**

**Received:** **27-01-2026**
**Revised:** **29-03-2026**
**Accepted:** **20-05-2026**
**Published:** **28-05-2026**

**Vol.2026, No.1**

**DOI:**

***Corresponding author. Email:*** *mahmood.alshareedah@stu.edu.iq*

**Orcid:** https://orcid.org/0000-0002-2358-3785

## ABSTRACT

In this paper, we propose the design and implementation of a tuned quad- copter derived from an open source autopilot platform. The overall operation of manual and assisted flight is accomplished using a lightweight airframe, performance optimized propulsion units, and modular hardware, software architecture. A flight controller is designed with an open source and set up on the ground control station for sensor adjustment, flight mode setting and monitoring. Some real- flight experiments are carried out for system test, including disturbance rejection experiment, payload variation experiment, position-hold flight mode validation and maximum altitude test. The experimental results show that effective external disturbance compensation is achieved, and there are stable signal autonomous-assisted flight as well as reliable altitude holding up to 20 m. This platform presents a cost-effective and repeatable solution for many UAV research laboratories, educational activities, and practical applications.

**Keywords:** Quadcopter, Unmanned Aerial Vehicle, Open-Source Autopilot, Flight Stability, Position Hold, Experimental Validation

**How to cite the article**

## 1. Introduction

With the advantage of vertical take-off and landing, simple mechanical design and high maneuverability, unmanned aerial vehicles (UAVs), especially quadcopters have been attracting an increasing amount of attention in recent years [1, 2]. This versatility allows quadcopters to be effectively used in applications of aerial monitoring, inspection, mapping, environmental surveillance and educational experimentation among others [3, 4]. The explosion of open-source hardware and software ecosystems helps quadcopters be adopted more rapidly when allowing low-cost development and flexible system change [5, 6].

Stability for flight performance is still a serious challenge in the quadcopter systems, given their natural nonlinearities and the high degree of coupling among the control axes; they are also highly vulnerable to external disturbances including wind or payload changes. Thus reliable operation is critically dependent on the efficient combination of sensing, controlling, and actuating systems [7, 8]. Commercial UAV systems are proven for reliable and professional application though such solutions are usually costly with only limited possibility of tailoring to the help scientific purposes. Open-source autopilot platforms, on the other hand, offer transparency and configurability and they are affordable which makes them appealing for research and educational applications [9, 10].

Several works [11–14] have been published in the literature to develop and control quadcopter and validate their performance by an experimental platform. Most of the existing literature focus on controller design, simulation-based validation or high- level system descriptions. However, less research is available in this field and presents systematic experimental validations that show real-flight stability under the influence of external disturbances and payload differences, assisted flight modes as well as height limits with a low-cost platform that can be reproduced accordingly[15, 16]. This gap raises the interest to have a quadcopter system implemented by means of open-source autopilot software, with well-chosen hardware components and systematic testing based on experimental results.

In this article, we design, implement and experimentally test community based open-source autopilot for a stable quadcopter system. The claimed solution integrates the lightweight airframe with an optimized propulsion unit and a modular hard- ware–software system architecture. The autopilot is programmed via a ground control station for precise sensor calibration, autonomous waypoint mission capabilities and real-time monitoring. A number of actual-flight tests are presented to evaluate disturbance rejection, payload take-up performance, position holding, and maximum-altitude operation. The main contributions of this paper can be summarized as follows:

- Design and implementation of a low-cost quadcopter platform using an open-source autopilot.
- Systematic hardware and software configuration to achieve stable manual and assisted flight.
- Experimental validation through real-flight tests, including disturbance rejection, payload variation, position-hold evaluation, and altitude testing.
- Demonstration of a reproducible UAV platform suitable for research, education, and practical applications.

The remainder of this paper is organized as follows. Section 2 presents the system architecture and hardware design. Section 3 describes the software environment and control configuration. Section 4 details the experimental setup and testing scenarios. Experimental results and discussion are provided in Section 5. Finally, conclusions and future research directions are summarized in Section 6.

## 2. Preliminaries

In this section, we present the general architecture of the built quadcopter platform, focusing on how the airframe hardware, open-source autopilot stack and ground-side configuration/monitoring tools are integrated. The system is based on a cyber–physical architecture in which perception, computation, actuation and communication work together to achieve stable flight while enabling manual or autonomous operation.

### *2.1 Architecture Overview*

Figure 1 illustrates the high-level block diagram of the platform. The quadcopter is organized into four main layers:

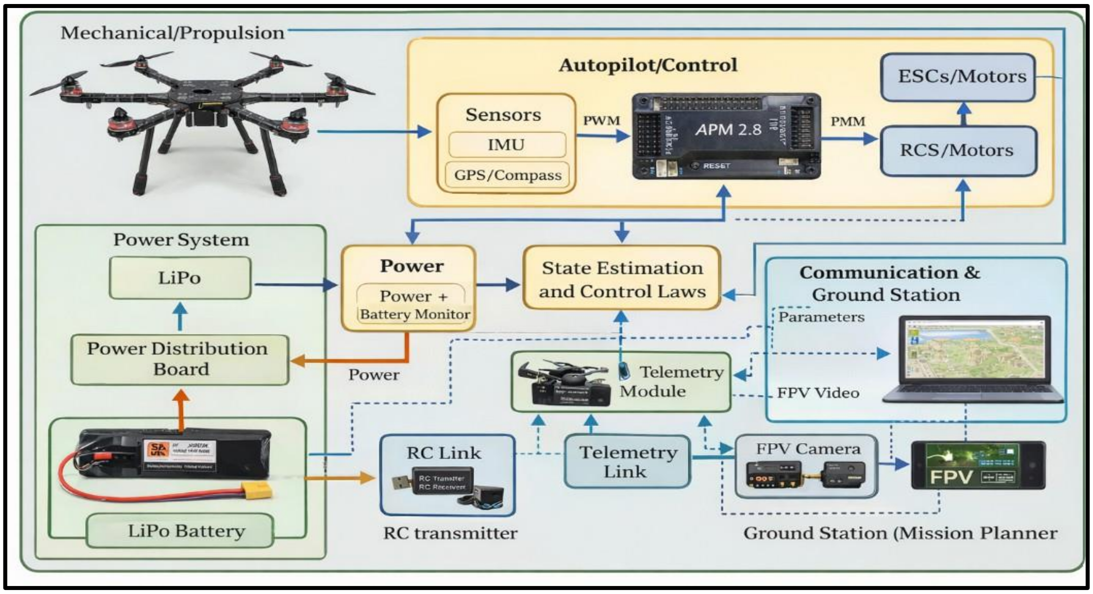


**Figure 1.** Overall architecture of the developed quadcopter system using an open-source autopilot

- Mechanical and Propulsion Layer: frame, motors, propellers, and landing gear providing lift and structural support.
- Power Distribution Layer: LiPo battery, power distribution board, and power module for regulated supply and battery monitoring.
- Autopilot and Sensing Layer: flight controller (open-source autopilot), inertial sensing, and GPS/compass modules for state estimation.
- Communication and Payload Layer: RC link for manual control, telemetry for data exchange with the ground station, and FPV subsystem for real-time video and on-screen flight information.

The architectural methodology is intended to be cheap, modular and repeatable. Leveraging an open-source autopilot and off-the-shelf hardware enables rapid development and easy maintenance, maintaining necessary features for stable flight, mission planning, and live monitoring. This renders the platform suitable as a basis for further upgrades, such as proprietary sensor improvements, advanced control tuning or specialized payload integration.

### *2.2 Hardware Requirements*

The hardware for the quadcopter system is chosen to enable stable flight, efficient power supply as well as to integrate smoothly into an open-source autopilot. The main hardware components from the developed system are shown in Figure 2. The major hardware elements, and their roles in the system operation, are addressed in subsequent subsections below.

- Frame: The frame is a mechanical part by which all other components are mounted. Yes, it has to be of little weighty and at the same time a rigid one along with absorbing a minimum of vibration. A carbon-fiber, X-type frame is chosen for its excellent strength-to-weight ratio and resilient structure. The symmetrical design makes the propellers look better and offers a good balance for drones.

- Brushless Motors: The quadcopter’s main propulsion system is provided by brushless DC motors. Such motors are known for their efficiency, speed of response, and life cycle. Four equal distant motors are positioned from the center of gravity to obtain balance lift forces. Opposing motors cancel out the reactive torque to keep you in control and yaw in balance [17, 18].

- Electronic Speed Controllers (ESCs): The rotational speed of each motor is con- trolled by electronic speed controllers [19, 20]. The ESCs are controlled via PWM signals from the flight controller, and transform DC power coming from a battery to 3-phase AC which results in the brushless motor spinning. With separate ESC control, thrust can be more finely controlled as well as giving you superior roll, pitch and yaw response.

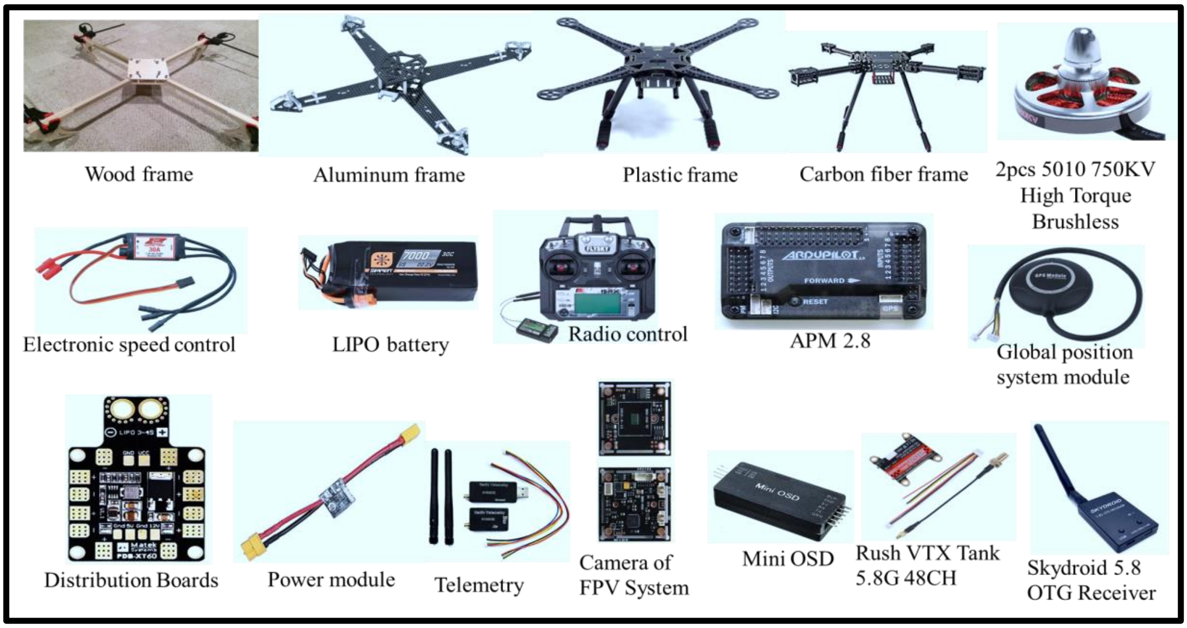


**Figure 2.** Main hardware components of the developed quadcopter system

- LiPo Battery: It is powered by a lithium polymer (LiPo) battery which has high energy capacity and supports large discharges [21, 22]. The battery capacity and discharge rating are chosen accordingly to provide adequate flight time as well as safe power source for both the motors, and on board electronics.

- Radio Control (RC) Remote: The radio-control system includes a handheld transmitter and an on-board receiver. It has manual piloting and works as the safety interface in testing or emergency situations. Attitude, throttle and flight mode selection are controlled by RC channels.

- Autopilot (APM 2.8): The quadrotor is controlled by an APM 2.8 autopilot, which acts as the CPU of the system. It processes sensor information, runs calculations for stabilization and navigation, and drives motors [21, 22].

- Global Positioning System (GPS): The GPS module supplies global position and velocity by the use of standard NMEA protocol, so the flight control system can be in navigation mode and get low-power consumption.

- Power Distribution Board: The PDB delivers power from the LiPo batteries to the ESCs and other on-board devices. It reduces wiring, provides constant volt- age output and aids in long thermally ensured power with centralized power management.

- Power Module: Real-time battery voltage and current consumption is recorded in the power module. These readings are sent to the autopilot for monitoring the voltage, flight time calculation and lights when low voltage is detected in order to avoid poor operation.

- Telemetry System: The telemetry system provides a two-way data link between the quadcopter and a ground control station. It allows monitoring of flight status, sensor data and system state in real-time, in-flight parameter adjustments and no-fly zone configuration.

- FPV System: An FPV system offers a user’s eye view from the quadcopter. What’s in the box: The package contains a camera (with the on-screen display replaceable), an OSD, a video transmitter and a video receiver. The OSD displays crucial flight data including battery voltage, receiver signal strength and navigation features such as “direction of return” and “home direction”.

## 3. Software and Control Configuration

The software environment and control setup implemented to achieve a stable and robust quadcopter flight performance is discussed in this section, using an open-source autopilot. The architecture emphasizes the relationship among ground control station, onboard autopilot firmware, sensing modules and actuation components. The ground control station (GCS) is used to flash firmware, tune parameters, plan missions and monitoring status in flight by a telemetry link. On the system, an autopilot firmware reads out sensor data from an inertial measurement units (IMU) and GPS to estimate states of a vehicle and run cascade-connection dependent control loops. The inner control loop is for stabilizing the position of the quadcopter with respect to its attitude, and the outer loop is for balancing its altitude level and position.

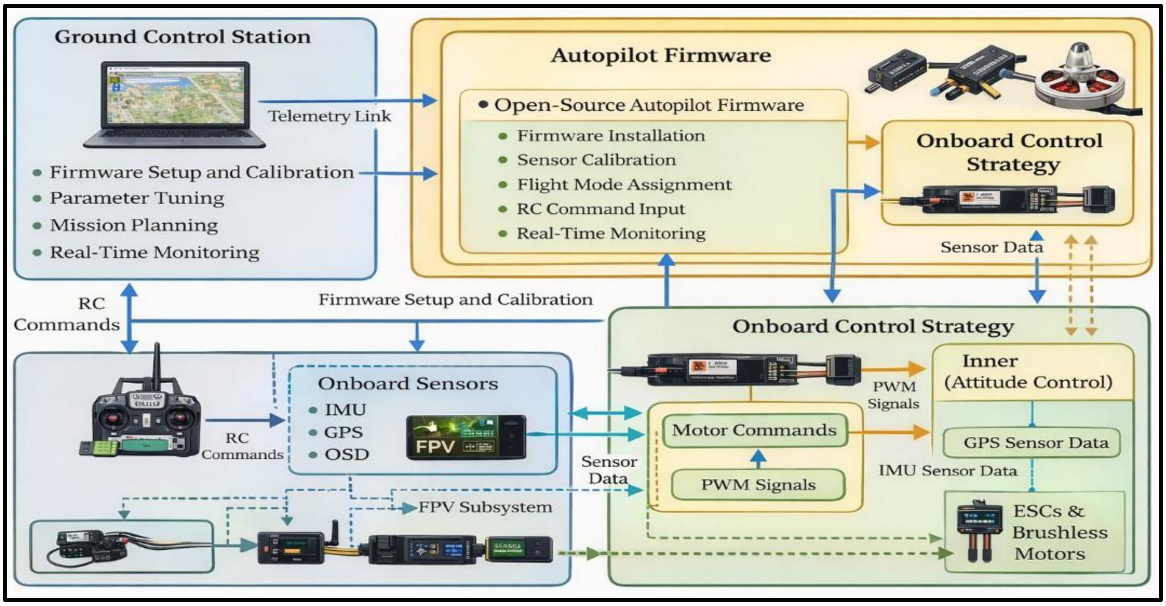


**Figure 3.** Software and control architecture of the quadcopter platform based on an open-source autopilot

Furthermore, Table1 presents the main system configuration parameters and set- points as used during initial setup and flight. They exist in the form of flight mode selection, sensor calibration settings, and control loop gains and comms interfacing specifications; they have an important role to play with each serving as a stabilizing factor for maintained quad behavior.

## 4. Experimental Setup and Testing Scenarios

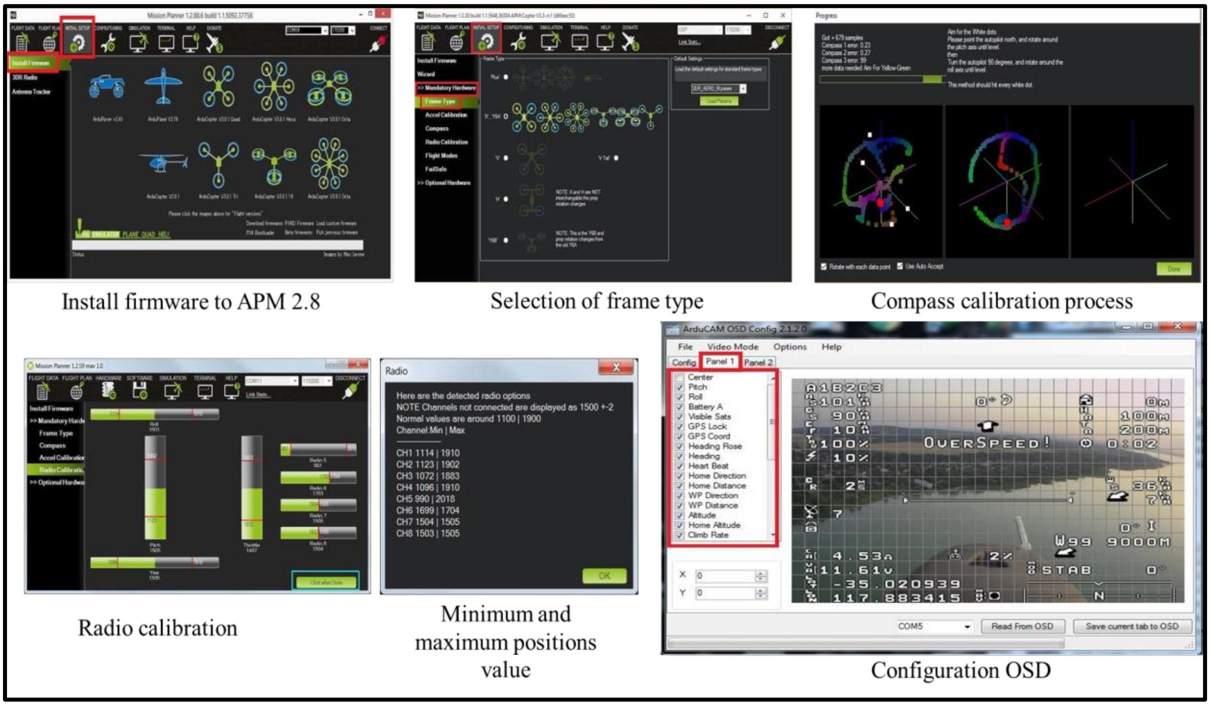
The experimental setup and testing conditions implemented to examine the stability performance and operational dependability of the developed quadcopter system are explained in this section. The experimental phase is aimed to validate the correct behavior of software configuration, sensor calibration and fast control or real flight performance under different working points.

### *4.1 Experimental Setup*

The main setup and calibration steps carried out with MP and the ArduCAM OSD configuration tools for flight testing are described in Figure 4. The experimental apparatus: firmware installation, frame type selection, sensor calibration and radio control tuning/ setup, as well the flight mode arrangement an FPV (on screen display) settings.

**Table 1.** Software and Control Configuration Parameters

| **Component / Parameter** | **Configuration** | **Description** |
|---|---|---|
| Autopilot Firmware | Open-source (APM 2.8) | Executes stabilization, navigation, and motor control algorithms |
| Ground Control Station | Mission Planner | Used for firmware upload, calibration, monitoring, and mission setup |
| Control Architecture | Cascaded control loops | Inner loop for attitude stabilization, outer loop for position control |
| Sensor Calibration | Accelerometer, Compass, RC | Ensures accurate state estimation and correct control input mapping |
| Flight Modes | Stabilize, Altitude Hold, Position Hold | Supports manual, assisted, and GPS-based autonomous flight |
| Motor Command Output | PWM signals | Generated by the autopilot and sent to ESCs for thrust control |
| Telemetry Communication | Bidirectional radio link | Enables real-time data monitoring and parameter tuning |
| FPV Configuration | Camera + OSD + VTX/VRX | Provides real-time video with flight data overlay |
| Safety Monitoring | Battery voltage/current | Prevents low-voltage operation and supports flight-time estimation |



```
Got + 679 samples
Compass 1 error: 0.23
Compass 2 error: 0.27
Compass 3 error: 99
more data needed Aim For Yellow-Green

Aim for the White dots.
Please point the autopilot north, and rotate around
the pitch axis until level.
then
Turn the autopilot 90 degrees, and rotate around the
roll axis until level.

This method should hit every white dot.
```

```
Here are the detected radio options
NOTE Channels not connected are displayed as 1500 +-2
Normal values are around 1100 | 1900
Channel:Min | Max
---------------
CH1 1114 | 1910
CH2 1123 | 1902
CH3 1072 | 1883
CH4 1096 | 1910
CH5 990 | 2018
CH6 1699 | 1704
CH7 1504 | 1505
CH8 1503 | 1505
```

Install firmware to APM 2.8

Selection of frame type

Compass calibration process

Radio calibration

Minimum and maximum positions value

Configuration OSD

**Figure 4.** Experimental setup and software configuration stages

The firmware is installed onto the autopilot via the Mission Planner interface and the type of quadcopter frame is set to ensure that motor mixing on all four motors is as it should be. Compulsory hardware calibration tasks are then carried out such as attitude capability of the accelerometer, heading estimation functionality of compass and mapping between transmitter inputs to control channels that is radio. Battery monitoring thresholds are also set in order to trigger safety checks during the flight based on voltage and current measurements.

In addition, the FPV subsystem is configured using the ArduCAM OSD configu- ration tool, where essential flight information such as battery status, altitude, speed, GPS coordinates, and flight mode indicators are selected and overlaid on the live video stream.

### *4.2 Testing Scenarios*

Several flight tests are conducted to evaluate system behavior and stability:

- Disturbance Rejection: External disturbances, such as manual pulling and payload switching are applied to the quadcopter so as to study the ability of the controller in attitude stabilization and perturbation recovery. This test verifies the performance of the cascaded control loops to reject disturbances.

- Flight Modes Testing: Various flight modes (e.g., Stabilize, Altitude Hold and Position Hold) are evaluated to measure mode switching reliability and control consistency. It is verified how the quadcopter reacts to instructions of the pilot and if it can be hold in a particular position such as altitude or position.

- Payload Carrying Test: Multiple payloads are progressively mounted to check thrust margin, power consumption and degradation of stability. The findings also indicate of the highest payload that is safely supported by the platform.

- Max Altitude Range Test: The UAV is flown to increasing altitudes and distances with telemetered data monitored to ascertain the reliability of communication, accuracy of navigation as well as robustness of the system.

- Criteria for Comparison: Flight stability, control responsiveness, sensor reliability and power consumption behavior are the characteristics that will be analyzed in the experimental evaluation. Telemetry and FPV feedback are employed to control system’s performance while ensuring the safety of experiments in controlled conditions.

The results obtained from these experiments are discussed in the following section to analyze system performance and validate the proposed design.

## 5. Results and Discussion

This section reports and analyses experimental results obtained from real-flight experiments on the developed quadcopter platform. Flight stability, disturbance rejection ability, control mode performance and flight envelope are evaluated. Experimental videos are employed as evidence to qualitatively verify that the system works under different conditions.

### *5.1 Disturbance Rejection Performance*

The disturbance rejection capability of the proposed quadcopter system is evaluated through two complementary real-flight tests designed to assess controller robustness under external and load-induced disturbances.

#### *5.1.1 Test 1: External Disturbance via Rope Pulling*

In the first experiment, an external excitation is generated by towing of the quadcopter using a rope in hover flight situation. This results in an impulse force applied on the car that deviate from its original orientation and position. The purpose of this test is to determine if the autopilot can quickly respond to a disturbance it was not expecting and recover the vehicle back to a level hover.

As illustrated in Figure 5, the quadcopter responds immediately to the applied force by adjusting motor thrust asymmetrically. The experimental behavior, demonstrated in the recorded flight video 1:https://youtu.be/GX9XtSnr II, shows that the quadcopter successfully suppresses the disturbance and returns to a stable hover without oscillatory behavior or loss of control.

**Figure 5.** External disturbance rejection test by applying a sudden force through rope pulling during hover.

### *5.1.2 Test 2: Payload-Induced Disturbance*

In the second experiment, an extra payload of around 400 g is added on top of the quadcopter in order to test stability while increasing load. This test checks thrust margin, control stability and power system adequacy for payloads near the top of the range.

The results shown in Figure 6 indicate that the quadcopter maintains stable flight despite the added mass. The recorded experiment (Video 2: https://youtu.be/ZZZQ6m9z960) confirms that the control system compensates for the increased load by generating appropriate thrust commands, enabling steady hover and smooth maneuvering. This demonstrates sufficient thrust reserve and robustness of both the control configuration and power delivery system.

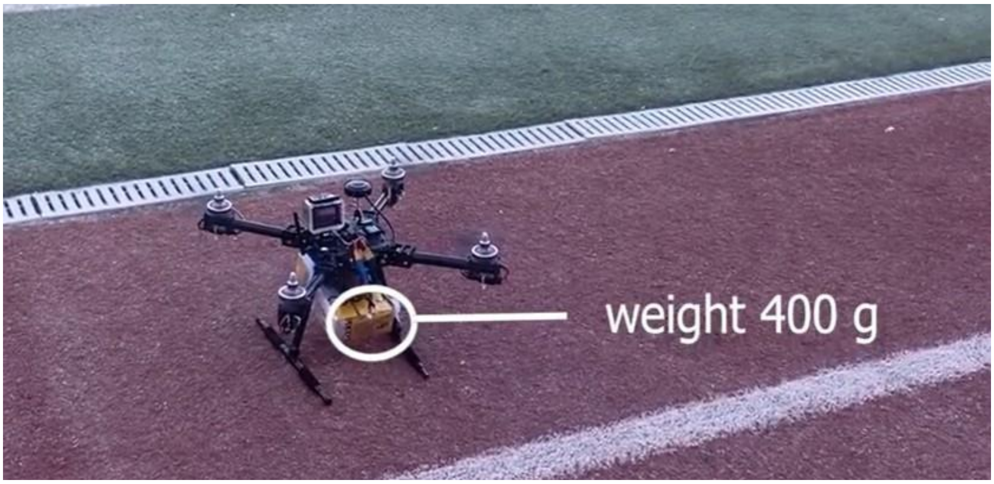


**Figure 6.** Payload-induced disturbance test with an additional 400 g mass attached to the quadcopter.

### *5.2 Test 3: Flight Mode Evaluation*

In the PosHold mode, we examine if the quadcopter is able to maintain its position and altitude with GPS and inertial sensor feedback. In this mode, the outer propulsion position controller controls the horizontal motion of vehicle and inner attitude controller stabilizes it.

**Figure 7** Flight mode evaluation using Position Hold (PosHold), demonstrating autonomous position stabilization during hover.

As shown in Figure 7, the quadcopter maintains a fixed position with minimal drift while compensating for environmental disturbances. The recorded flight experiment, available at Video 3: https://youtu.be/StJ26qu27pM, demonstrates smooth and stable behavior throughout the test duration. The results confirm correct GPS integration, reliable sensor fusion, and effective operation of the cascaded control strategy implemented by the open-source autopilot.

### *5.3 Test 4: Maximum Altitude Evaluation (20 m)*

The flying height of the quadcopter is determined by conducting flights at a ranging distance from 1 to 20 m and monitoring control performance, telemetry data and communication connectivity. This exercise is to confirm stable vertical control, reliability of sensors and robustness of total system at the desired operational conditions.

**Figure 8.** Maximum altitude test demonstrating stable ascent and hover at a height of 20 m.

As illustrated in Figure 8, the quadcopter achieves smooth ascent and maintains stable hover at the target altitude without noticeable oscillations or loss of control. Throughout the test, telemetry feedback confirms consistent sensor readings and reliable communication between the quadcopter and the ground control station. The recorded flight experiment, available at Video 4: https://youtu.be/79kY1MRMlEg, further demonstrates stable vertical performance and controlled descent, validating the effectiveness of the altitude control strategy and system configuration.

## 6. Discussion

From the experimental results of previous subsections, we can synthesize these main findings:

1. Effective disturbance rejection: The disturbance tests show that the quadcopter can stably counterbalance disturbances such as external and load-based disturbances. The motor thrust is quickly changed by the control system to counteract the disturbance and to regain a level hover, verifying that the attitude control loop is robust.

2. Performance in varying payload conditions: With a 400 g extra payload, stable flight is obtained - there is still enough thrust margin to be confident in keeping the throttle above hover. It means that the chosen propulsion and power units are sufficient for carrying moderate payloads.

3. Reliable autonomous-assisted flight modes: The Position Hold flight mode performs reliably, holding the quadcopter steady with little drift. This validates successful sensor calibration, good sensor fusion and proper design of the cascaded position and attitude controller.

4. Stable altitude control and communication reliability: The maximum altitude test demonstrates smooth ascending, stable hovering at an altitude of 20 m, and remaining controlled descending without a loss of communication. The telemetry values stay stable during the whole test period and reveal robust vertical control as well as link stability up to a distance of 17 m.
5. Open source autopilot integration validation: The open source autopilot integrated is capable of providing stable and consistent flight performance, if the configuration is done correctly - this was also found throughout all our experiments. This demonstrates its fitness for use in research, teaching and prototyping of UAV applications.

6. System limitations and future work: Although encouraging, system performance can be negatively influenced by environmental factors such as wind or GPS accuracy. Future work might seek to improve the controller by adding sensor or data-driven tuning for greater robustness.

## 7. Conclusion and Future Work

This paper demonstrated the design and experimental testing of a stable quadrotor system using an open-source autopilot platform. The design is based on a lightweight mechanical structure, an efficient propulsion system and a configurable software control framework that allows achieving reliable flight performance. Real-flight tests were performed to evaluate disturbance rejection, flight mode versatility, and operational altitude range. Experiment results indicate that the quadcopter can fly stably under external disturbances and payload changeover. Because of good sensor calibration and neat control setup there are no problems with (self-stabilizing) assisted flight, like position hold etc. In addition, stable ascent and hovering at the maximum test altitude demonstrates the reliability of the altitude control and communication systems. In conclusion, the results of this study confirm the ability of the introduced hard- ware–software integration and also demonstrate that open-source autopilot platforms are appropriate for research and educational UAV purposes.

For future work, the capabilities of our proposed quadcopter as a testbed may be further enhanced by the integration of sophisticated control algorithms, such as adaptive or learning-based strategies, toward increasing robustness to environment variations. A further experimental validation, involving quantitative performance analysis such as energy consumption for path tracking accuracy and long duration flight tests, is planned. Further, the addition of supplementary sensing modes for obstacle avoidance and autonomous navigation as well as optimization of power management to maximize flight duration will be investigated. It’s also a platform capable of being adapted to more specialized uses, such as aerial surveillance, inspection or lightweight delivery.

**Corresponding author**

**Mahmood A. Al-Shareeda**

*mahmood.alshareedah@stu.edu.iq*

**Acknowledgements**
NA.

**Funding**
No funding.

**Contributions**
A.A.A; A.H.T; Conceptualization, E.J.M; M.A.A; Investigation; M.A.A; Writing (Original Draft), A.A.A; M.A.A; Writing (Review and Editing) Supervision, A.A.A; M.A.A; Project Administration.

**Ethics declarations**
This article does not contain any studies with human participants or animals performed by any of the authors.

**Consent for publication**
Not applicable.

**Competing interests**
All authors declare no competing interests.

## References

[1] Aditya, V., Aswin, D. S., Dhaneesh, S. V., Chakravarthy, S., Kumar, B. S., & Venkadavarahan, M. (2024). A review on air traffic flow management optimization: Trends, challenges, and future directions. *Discover Sustainability*, 5(1), 519. https://doi.org/10.1007/s43621-024-00519-9

[2] Almazroi, A. A., Alkinani, M. H., Al-Shareeda, M. A., & Manickam, S. (2024). A novel DDoS mitigation strategy in 5G-based vehicular networks using Chebyshev polynomials. *Arabian Journal for Science and Engineering*, 49(9), 11991–12004. https://doi.org/10.1007/s13369-024-08938-4

[3] Abdillah, R. E., Moenaf, H., Rasyid, L. F., Achmad, S., & Sutoyo, R. (2024). Implementation of artificial intelligence on air traffic control: A systematic literature review. In *Proceedings of the 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)* (pp. 1–7). IEEE. https://doi.org/10.1109/IMCOM61040.2024.10460728

[4] Renkhoff, J., Ternus, S., & Guleria, Y. (2025). A survey on personalized conflict resolution approaches in air traffic control. *Aerospace*, 12(9), 751. https://doi.org/10.3390/aerospace12090751

[5] Alves, D., Belo-Pereira, M., Mendonça, F., & Morgado-Dias, F. (2025). Intelligent visibility forecasting at airports: A systematic review. *Environmental Research Communications*. Advance online publication. https://doi.org/10.1088/2515-7620/ad8c2b

[6] Ogunwole, O., Onukwulu, E. C., Joel, M. O., Adaga, E. M., & Ibeh, A. I. (2023). Modernizing legacy systems: A scalable approach to next-generation data architectures and seamless integration. *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(1), 901–909.

[7] Mohammed, B. A., Al-Shareeda, M. A., Al-Mekhlafi, Z. G., Alshudukhi, J. S., & Al-Dhlan, K. A. (2024). HAFC: Handover authentication scheme based on fog computing for 5G-assisted vehicular blockchain networks. *IEEE Access*, 12, 6251–6261. https://doi.org/10.1109/ACCESS.2024.3350274

[8] Castruita-López, J. F., Aviles, M., Toledo-Pérez, D. C., Macías-Socarrás, I., & Rodríguez-Reséndiz, J. (2025). Electromyography signals in embedded systems: A review of processing and classification techniques. *Biomimetics*, 10(3), 166. https://doi.org/10.3390/biomimetics10030166

[9] Baker, B., Woods, J., Reed, M. J., & Afford, M. (2024). A survey of short-range wireless communication for ultra-low-power embedded systems. *Journal of Low Power Electronics and Applications*, 14(2), 27. https://doi.org/10.3390/jlpea14020027

[10] Soto-Cruz, J., Ruiz-Ibarra, E., Vázquez-Castillo, J., Espinoza-Ruiz, A., Castillo-Atoche, A., & Mass-Sanchez, J. (2024). A survey of efficient lightweight cryptography for power-constrained microcontrollers. *Technologies*, 13(1), 3. https://doi.org/10.3390/technologies13010003

[11] Thakur, D. S., Kourav, S., Shah, S. K., & Verma, K. (2024). Area- and speed-efficient Vedic RISC processors for embedded systems. In *Proceedings of the IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 1219–1224). IEEE. https://doi.org/10.1109/CSNT60226.2024.10521946

[12] Soni, M. S., Jisan, M., Kaustav, B., & Ghosh, R. (2025). Advancements in weather monitoring systems: A comprehensive review. In *Proceedings of the 8th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)* (pp. 1–5). IEEE.

[13] Ganesan, S., Lean, C. P., Chen, L., Yuan, K. F., Kiat, N. P., & Khan, M. R. B. (2024). IoT-enabled smart weather stations: Innovations, challenges, and future directions. *Malaysian Journal of Science and Advanced Technology*, 180–190.
[14] Kumar, A., Malhotra, S., Kaur, D. P., & Gupta, L. (2022). Weather monitoring and air quality prediction using machine learning. In *Proceedings of the 1st International Conference on Computational Science and Technology (ICCST)* (pp. 364–368). IEEE. https://doi.org/10.1109/ICCST55977.2022.10044413
[15] Ujoodha, M., Pultoo, A., & Oojorah, A. (2021). Climate monitoring using an Arduino-based mobile weather station and open-source codes. *JESS: Journal of Education on Social Science*, 16(1), 105–114.
[16] Mabrouki, J., Azrour, M., Dhiba, D., Farhaoui, Y., & El Hajjaji, S. (2021). IoT-based data logger for weather monitoring using Arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Mining and Analytics*, 4(1), 25–32. https://doi.org/10.26599/BDMA.2020.9020016
[17] Michailidis, I., Mountzouris, P., Triantis, P., Pagiatakis, G., Papadakis, A., & Dritsas, L. (2025). An Arduino-based, portable weather monitoring system remotely usable through the mobile telephony network. *Electronics*, 14(12), 2330. https://doi.org/10.3390/electronics14122330
[18] Venugopalaswamy, S., Sujitha, V. H. S. D., Padmavathi, G. L., Sravya, M., Jahnavi, D. D., & Panguluri, S. K. (2024). Dual-axis solar tracking system with weather monitoring system. *International Journal of Engineering Technology and Management Sciences*, 8(2), 88–99.
[19] Stoyanov, S., Kuzmanov, Z., & Stoyanova, T. (2024). Weather monitoring system using IoT-based DIY automatic weather station. In *Proceedings of the 9th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)* (pp. 1–6). IEEE. https://doi.org/10.1109/EEAE60042.2024.10482317
[20] Talib, A. H., AL-Nakkash, A. H., Wadday, A. G., Abed, A. A., & Al-Shareeda, M. A. (2026). Real-Time Spectrum Sensing on an RTL-SDR-Based IoT Platform. *International Journal of Cybersecurity Engineering and Innovation*, *2026*(1).
[21] Alrajeh, M., Almaiah, M., & Mamodiya, U. (2026). Cyber Risk Analysis and Security Practices in Industrial Manufacturing: Empirical Evidence and Literature Insights. *International Journal of Cybersecurity Engineering and Innovation*, *2026*(1).
[22] Al-shareeda, M., Musa, H. A., Jaafar, A., Salman, A. A., Tami, Z. J., Hameed, H. M., ... & Bashkh, N. S. (2026). Design and Implementation of a Speech-to-Sign Robotic Arm for Deaf Communication. *International Journal of Cybersecurity Engineering and Innovation*, *2026*(1).