Power System Protection and Control
ISSN:1674-3415

DESIGN AND DEVELOPMENT OF AN STM32F MICROCONTROLLER-BASED UAV DRONE SYSTEM

Mr. Arush Vijaykant Nyayadhish[1], Dr Raghunath Bhadade[2]

Department of Electrical & Electronics Engineering, Vishwanath Karad MIT World Peace University Pune Maharashtra India
Email Id: mrarushnyayadhish@gmail.com raghunath.bhadade@mitwpu.edu.in

Received: 22/06/2026 Revised: 27/07/2026 Accepted: 08/09/2026

***ABSTRACT*:**

The increasing deployment of unmanned aerial vehicles (UAVs) in surveillance, inspection, agriculture, logistics, environmental monitoring, and autonomous robotics has intensified the demand for flight-control systems that combine deterministic real-time execution, low hardware cost, and architectural transparency. This paper presents the design, mathematical modeling, embedded implementation, and experimental validation of a custom quadrotor flight controller based on the STM32F411CEU6 32-bit ARM Cortex-M4 microcontroller. Unlike proprietary commercial flight controllers, the proposed platform emphasizes an interpretable embedded architecture in which sensor acquisition, attitude estimation, PID control, motor mixing, communication, and flight-data logging are explicitly implemented and experimentally verified.

The developed system integrates an MPU6050 inertial measurement unit, BMP280 barometric pressure sensor, QMC5883L three-axis magnetometer, NEO-6M GPS receiver, time-of-flight (ToF) ranging sensor, and W25QXX-series SPI NOR Flash memory. Four 2212 920-kV brushless DC motors driven by electronic speed controllers provide propulsion. A 500-Hz attitude-control loop is implemented using hardware timers, while UART DMA is employed for non-blocking reception of digital radio-control data. Sensor measurements are filtered using low-pass and complementary-filter techniques, after which PID controllers generate roll, pitch, and yaw correction terms. A Quad-X motor-mixing matrix converts these control commands into four ESC pulse-width commands.

The experimental validation comprises bench-level sensor characterization, oscilloscope-based PWM verification, software debugging using STM32CubeIDE, and tethered flight testing. The measured accelerometer data exhibited substantial vibration-induced disturbance during motor operation; after filtering, the estimated attitude showed substantially reduced variation under stationary conditions. The PWM subsystem generated the required 1000–2000 μs command range with hardware-timer timing. The flight tests demonstrated stable hover and responsive attitude control after empirical PID tuning. The results indicate that an STM32F411-class microcontroller remains a viable platform for deterministic low-cost UAV stabilization when the architecture is carefully designed around bounded computational load, hardware timers, DMA, and appropriately selected sensor-fusion algorithms.

Keywords*: UAV, quadrotor, STM32F411, embedded systems, flight controller, PID control, complementary filter, sensor fusion, DMA, real-time control, blackbox logging, edge computing.*

1.1 BACKGROUND OF THE PROJECT

Unmanned aerial vehicles have developed from remotely operated platforms into sophisticated cyber-physical systems capable of autonomous navigation, perception, inspection, mapping, and intelligent decision-making. Their applications include precision agriculture, infrastructure inspection, disaster response, aerial imaging, environmental monitoring, logistics, and search-and-rescue operations. Among the different UAV configurations, the quadrotor remains particularly attractive for research and education because of its mechanically simple architecture, vertical take-off and landing capability, hovering ability, and high maneuverability.

Despite its mechanical simplicity, the quadrotor represents a challenging control problem. It is nonlinear, underactuated, dynamically coupled, and open-loop unstable. Four independently controlled rotor thrusts must regulate six degrees of freedom. Consequently, reliable flight requires high-quality state estimation and a control loop capable of responding to disturbances within predictable timing constraints.

The flight controller constitutes the central computational element of the UAV. It acquires measurements from inertial, magnetic, barometric, positioning, and ranging sensors; estimates the vehicle state; computes control errors; and generates actuator commands. The resulting closed-loop system must operate with sufficiently low latency and bounded execution-time variation. Contemporary research increasingly investigates advanced model predictive, adaptive, learning-based, and fault-tolerant control methods for addressing nonlinear dynamics and environmental uncertainties [1]–[5]. Recent studies have also demonstrated the value of hardware-in-the-loop (HIL) experimentation for validating controllers before physical flight testing [6], [7].

Conventional PID control nevertheless remains attractive for resource-constrained embedded platforms because of its low computational complexity, intuitive tuning procedure, and comparatively predictable execution. Recent research on online learning and adaptive quadrotor control indicates that more sophisticated algorithms can improve robustness, but these approaches introduce additional computational and implementation complexity [3],[5]. Thus, a deterministic PID-based architecture remains relevant as a baseline controller and as a foundation for future adaptive extensions.

The STM32F411CEU6 is particularly suitable for this class of application. It incorporates an Arm Cortex-M4 processor operating at up to 100 MHz, a single-precision floating-point unit, DSP instructions, hardware timers, DMA controllers, serial communication peripherals, and up to 512 KB Flash and 128 KB SRAM depending on device variant [8],[9]. These characteristics allow sensor acquisition, attitude estimation, PID computation, communication, and actuator generation to be implemented on a single low-cost microcontroller.

The proposed research therefore investigates whether a transparent STM32F411-based embedded architecture can provide the computational determinism and control performance required for a small quadrotor while retaining a substantially lower hardware complexity than high-end flight-control platforms.[30][31]
The principal contributions of this work are:

1. Development of an STM32F411-based modular UAV flight-controller architecture integrating inertial, magnetic, barometric, positioning, ranging, communication, and data-logging subsystems.
2. Implementation of a deterministic 500-Hz attitude-control loop using hardware timers and floating-point computation.
3. Development of a lightweight complementary-filter attitude estimator suitable for resource-constrained embedded operation.
4. Implementation of PID-based attitude stabilization and Quad-X motor mixing.
5. Integration of DMA-assisted radio communication and SPI-based blackbox logging to minimize interference with the primary control loop.
6. Experimental validation through sensor characterization, PWM timing measurements, software debugging, and tethered flight testing.

The remainder of this paper is organized as follows. Section II reviews relevant research. Section III describes the proposed system architecture. Section IV presents the mathematical model and control formulation. Section V discusses hardware and firmware implementation. Section VI presents the experimental methodology and results. Section VII discusses limitations and future extensions. Section VIII concludes the paper.

RELATED WORK

A. Embedded Flight-Control Architectures

The development of embedded UAV controllers has traditionally involved a trade-off between computational capability, hardware cost, software complexity, and determinism. Commercial autopilot platforms provide extensive functionality but often abstract low-level implementation details behind hardware abstraction layers and large software stacks.[29]

The STM32F4 family provides an alternative platform for experimental and educational UAV research because it combines floating-point computation, DSP functionality, timers, communication interfaces, and DMA capabilities within a relatively inexpensive microcontroller [8], [9]. The STM32F411 specifically provides a 100-MHz Cortex-M4 core with an integrated single-precision FPU and DSP instructions [8].

DOI: 10.46121/pspc.54.3.59

The proposed work does not seek to compete directly with mature open-source autopilot ecosystems in terms of autonomous-navigation functionality. Instead, it focuses on architectural transparency and experimental traceability. The complete signal path—from sensor registers through filtering and control calculations to PWM generation—is retained within a comparatively compact firmware implementation.[28]

B. Attitude Estimation and Sensor Fusion

Attitude estimation is fundamental to quadrotor stabilization. Gyroscopes provide high-bandwidth angular-rate information but suffer from bias accumulation, whereas accelerometers provide a gravity reference but are strongly affected by translational acceleration and vibration.[27]

The complementary filter remains an attractive solution for low-cost UAVs because it combines the short-term response of gyroscope integration with the long-term reference provided by accelerometer measurements. Recent research has demonstrated improved complementary-filter architectures incorporating additional preprocessing and adaptive weighting [10], [11].

For more advanced autopilot systems, Extended Kalman Filter (EKF) architectures combine gyroscopes, accelerometers, magnetometers, GPS, and barometric measurements. Modern open-source autopilot systems commonly employ EKF-based state estimation because of its ability to account for multiple sensor uncertainties and reject inconsistent measurements [12]. However, an EKF requires considerably more computational and implementation complexity than a basic complementary filter.

The present work therefore employs a complementary filter for the primary attitude-estimation function while retaining the magnetometer, GPS, barometer, and ranging sensors as extensible inputs for future navigation modes.[26]

C. UAV Control

PID controllers remain widely used for low-level UAV stabilization because their computational requirements are modest and their parameters have intuitive physical interpretations. However, the nonlinear and coupled nature of quadrotor dynamics has motivated extensive research into MPC, adaptive control, sliding-mode control, and learning-based approaches.

Borbolla-Burillo et al. investigated real-time implementation of cascaded MPC for UAV control and highlighted the computational challenges associated with nonlinear UAV dynamics [1]. Similarly, recent work has investigated multilayer MPC and moving-horizon estimation for real-time quadrotor control [2]. Adaptive approaches have also been proposed to improve robustness to disturbances and model uncertainty [4].

More recently, Gu et al. systematically evaluated online-learning approaches for quadrotor control and demonstrated the trade-off between estimation accuracy and real-time computational efficiency [3]. These results support a layered approach in which a deterministic low-level controller provides the fundamental stabilization loop while more computationally demanding adaptive or learning-based algorithms can be introduced at higher levels.

D. Hardware-in-the-Loop Validation

Physical flight testing introduces safety risks and makes systematic controller evaluation difficult. HIL simulation provides an intermediate validation layer in which the controller executes on real embedded hardware while the vehicle dynamics are simulated in real time.

Recent UAV studies have employed HIL experimentation to evaluate adaptive and nonlinear controllers under disturbances, input delays, and actuator faults [6], [7]. Such approaches provide repeatability and enable controller evaluation before full flight deployment.

The current work complements these approaches with bench-level and tethered experimental testing. Future development will incorporate a complete HIL platform to quantify computational latency and controller response under repeatable simulated disturbances.[25]

DOI: 10.46121/pspc.54.3.59

Power System Protection and Control
ISSN:1674-3415

SYSTEM ARCHITECTURE

The proposed UAV is organized into five principal subsystems: **(i) flight-control processor, (ii) sensor subsystem, (iii) propulsion and actuator subsystem, (iv) communication and telemetry subsystem, and (v) power and data-logging subsystem**.

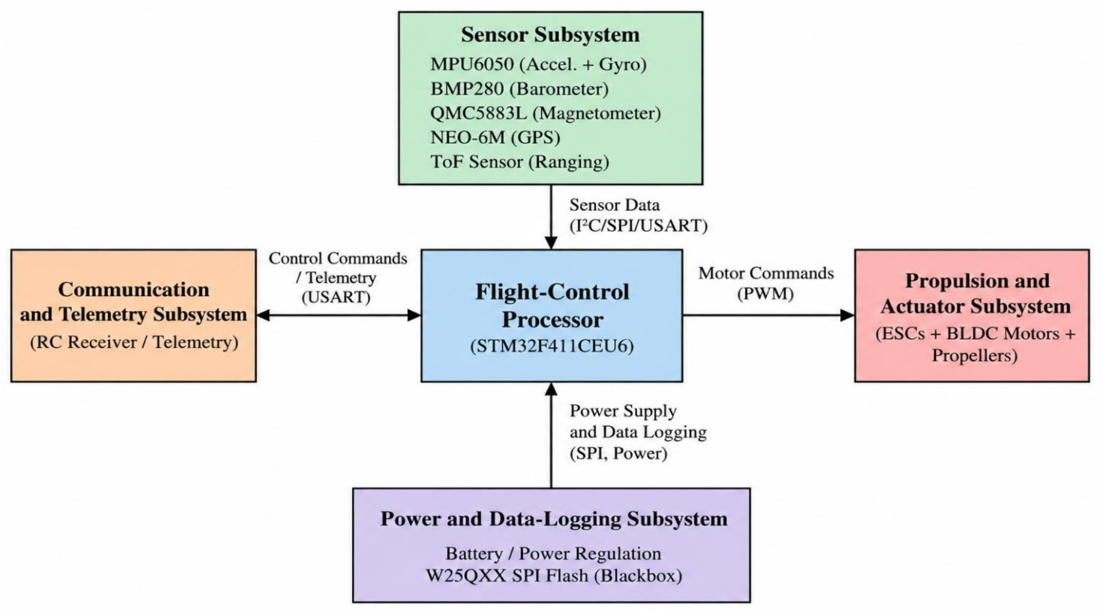

Fig. 1. Overall system architecture of the proposed UAV.

The **STM32F411CEU6** serves as the central flight-control processor. The **MPU6050** provides three-axis accelerometer and gyroscope measurements, while the **BMP280** provides barometric pressure and temperature measurements. The **QMC5883L** provides three-axis magnetic-field measurements for heading estimation. A **NEO-6M GPS receiver** provides geographic position and velocity information, while a **time-of-flight (ToF) ranging sensor** provides low-altitude distance measurements.

An external **W25QXX-series SPI NOR Flash** device is employed for blackbox flight-data logging. The W25Q128JV family, for example, provides 128-Mbit nonvolatile storage and supports SPI, Dual-SPI, and Quad-SPI interfaces [13].

The propulsion subsystem consists of four brushless DC motors driven by electronic speed controllers (ESCs). The ESCs receive pulse-width modulation (PWM) commands generated using the hardware timers of the STM32 microcontroller.

The overall system operates as a closed-loop architecture in which sensor measurements are acquired by the flight-control processor, processed to estimate the UAV state, and supplied to the control algorithm. The resulting control commands are converted into individual motor commands through the motor-mixing stage and subsequently applied to the ESCs. Communication and data logging operate in parallel with the primary control loop.

B. Microcontroller Platform

The **STM32F411CEU6** was selected as the central processing platform because of its combination of computational capability, floating-point unit (FPU), hardware timers, direct memory access (DMA), communication interfaces, and memory resources.

The STM32F411 family operates at up to **100 MHz** and incorporates an Arm Cortex-M4 core with a single-precision FPU and DSP instructions. Depending on the device variant, the family provides up to **512 KB of Flash memory and 128 KB of SRAM** [8], [9].

DOI: 10.46121/pspc.54.3.59

These computational and peripheral resources are adequate for the proposed low-level flight-control functions, including sensor acquisition, attitude estimation, PID computation, motor mixing, communication handling, and data logging. The use of hardware peripherals and DMA also helps maintain a predictable real-time execution structure.[24]

C. Sensor Subsystem and Interfaces

The sensor subsystem provides the measurements required for attitude estimation, altitude monitoring, navigation, and flight-state analysis.

The **MPU6050** is connected to the STM32 through the **I²C interface** and provides synchronized three-axis accelerometer and gyroscope measurements. Its selectable measurement ranges make it suitable for motion-control and attitude-estimation applications [14].

The **BMP280** provides barometric pressure and temperature measurements through either **I²C or SPI** [15]. In the proposed system, the barometric sensor is primarily intended for relative altitude estimation and monitoring rather than serving as a stand-alone precision landing sensor.

The **QMC5883L** provides three-axis magnetic-field measurements and is used as the magnetic heading reference. Since magnetometer measurements can be affected by electromagnetic interference and hard- and soft-iron distortions, appropriate compass calibration and magnetic-field validation are required before using the measurements for heading estimation.

The **NEO-6M GPS receiver** provides geographic position and velocity information. In the present architecture, GPS measurements are primarily intended to support future navigation-related functions such as position hold and return-to-home [17]. The **ToF ranging sensor** provides complementary short-range distance measurements for low-altitude flight and landing-related experiments.

D. Propulsion and Actuator Subsystem

The prototype employs four **2212-class, 920-kV brushless DC motors** coupled with fixed-pitch propellers and **30-A ESCs**. The motors are arranged in an **X configuration**, forming the Quad-X propulsion layout.

The propulsion system is selected to provide an adequate thrust margin for the approximately **1.2-kg prototype**. However, nominal motor and propeller specifications alone are not sufficient to establish the actual available thrust under the operating conditions of the UAV. Therefore, experimental thrust characterization using a calibrated thrust stand is recommended.

Before free-flight operation, the experimentally measured propulsion performance should be used to establish an adequate **total thrust-to-weight ratio**, together with appropriate actuator limits and safety margins.[18]

E. Communication and Telemetry Subsystem

A digital radio-control receiver is connected to the STM32 through a **USART interface**. DMA is used to transfer received communication frames into memory without requiring the processor to service every received byte individually.[19]

After the complete frame has been received, an interrupt callback sets a processing flag. The main control software subsequently processes the received data without performing extensive communication handling inside the interrupt service routine.[20]

This architecture reduces processor overhead and minimizes the possibility that communication processing will block or significantly delay the primary flight-control loop. The communication subsystem can also be extended to support telemetry transmission and real-time monitoring of selected flight parameters.

F. Power and Blackbox Data-Logging Subsystem

The power subsystem provides the required electrical power to the propulsion system, flight controller, sensors, communication interfaces, and other onboard electronics. Appropriate voltage regulation and power-distribution arrangements are required to maintain stable operation of the control electronics and prevent electrical disturbances from adversely affecting sensor measurements.[21]

DOI: 10.46121/pspc.54.3.59

For flight-data recording, an external **SPI NOR Flash** device is used as a blackbox data logger. The logged variables include:

- accelerometer measurements;
- gyroscope measurements;
- estimated roll and pitch angles;
- magnetic heading;
- PID tracking errors;
- PID controller outputs;
- throttle command;
- individual motor commands;
- battery-related measurements, where available; and
- control-loop timing information.

The blackbox subsystem provides an objective basis for **controller tuning, fault investigation, performance evaluation, and post-flight analysis**. In particular, loop-timing data can be used to verify the real-time behavior of the flight-control software and identify computational delays or timing variations.[22]

MATHEMATICAL MODEL AND CONTROL DESIGN

A. Quadrotor Dynamics

The quadrotor is modeled as a six-degree-of-freedom rigid body. The translational state is described by position and velocity, while the rotational state is represented by roll φ, pitch θ, and yaw ψ.
For motor i, the generated thrust is approximated by

$$T_i = k_f \omega_i^2 \quad (1)$$

where k_f is the thrust coefficient and ω_i is the rotor angular velocity.
The reaction torque generated by motor i can similarly be represented as

$$\tau_i = k_m \omega_i^2 \quad (2)$$

where k_m represents the aerodynamic torque coefficient.
The total thrust generated by the four rotors is

$$T = \Sigma(i=1 \text{ to } 4)\ T_i \quad (3)$$

The simplified rotational dynamics about the principal body axes are given by

$$I_x \ddot{\varphi} = \tau_\varphi \quad (4)$$
$$I_y \ddot{\theta} = \tau_\theta \quad (5)$$
$$I_z \ddot{\psi} = \tau_\psi \quad (6)$$

where I_x, I_y, and I_z are the principal moments of inertia, and τ_φ, τ_θ, and τ_ψ represent the control torques about the roll, pitch, and yaw axes, respectively.

These equations provide a simplified representation of the quadrotor dynamics. In practical flight conditions, the system is also affected by aerodynamic drag, motor dynamics, propeller characteristics, battery-voltage variation, ground effect, actuator saturation, and structural vibration.

B. Attitude Measurement and Euler-Angle Estimation

The attitude of the quadrotor is estimated using measurements obtained from the inertial measurement unit (IMU). Accelerometer measurements provide a gravity-based reference for estimating roll and pitch.
The accelerometer-based roll estimate is calculated as

$$\varphi_a = \tan^{-1}[a_y / \sqrt{(a_x^2 + a_z^2)}] \quad (7)$$

while the pitch estimate is calculated as

$$\theta_a = \tan^{-1}[-a_x / \sqrt{(a_y^2 + a_z^2)}] \quad (8)$$

Although accelerometers provide a useful reference for roll and pitch, they are susceptible to errors caused by translational acceleration and vibration. Therefore, gyroscope measurements are also used to estimate the change in attitude.
The gyroscope-based estimates are obtained by numerical integration:

$$\varphi_g(k) = \varphi(k-1) + p(k)\Delta t \quad (9)$$
$$\theta_g(k) = \theta(k-1) + q(k)\Delta t \quad (10)$$

where p and q are the measured angular rates about the body x- and y-axes, respectively.
For a control-loop frequency of 500 Hz,

DOI: 10.46121/pspc.54.3.59

$$\Delta t = 0.002 \text{ s} \quad (11)$$

C. Complementary-Filter-Based Attitude Estimation

Since gyroscope integration is subject to long-term drift, while accelerometer measurements are affected by high-frequency disturbances and linear acceleration, the two measurements are fused using a complementary filter.
The estimated roll angle is given by

$$\varphi(k) = \alpha[\varphi(k-1) + p(k)\Delta t] + (1-\alpha)\varphi\_a(k) \quad (12)$$

and the estimated pitch angle is

$$\theta(k) = \alpha[\theta(k-1) + q(k)\Delta t] + (1-\alpha)\theta\_a(k) \quad (13)$$

The prototype initially uses $\alpha = 0.98$. This value gives the gyroscope dominant short-term influence while allowing the accelerometer to provide long-term drift correction. However, $\alpha = 0.98$ is treated as a tuning parameter rather than a universal optimum. Adaptive complementary-filter approaches may provide improved attitude estimation when sensor disturbances vary significantly during flight [10], [11].

Yaw is handled separately because accelerometer measurements cannot provide an absolute heading reference. Therefore, the magnetometer is used to obtain a magnetic heading reference after appropriate calibration.

D. PID-Based Attitude Control

The estimated attitude is supplied to the PID controller to generate the required control commands. For each controlled axis, the tracking error is defined as

$$e(k) = r(k) - y(k) \quad (15)$$

where $r(k)$ represents the desired reference and $y(k)$ represents the estimated/measured state.
The discrete PID control law is given by

$$u(k) = K\_p\, e(k) + K\_i\, I(k) + K\_d\, D(k) \quad (16)$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.
The integral term is calculated as

$$I(k) = I(k-1) + e(k)\Delta t \quad (17)$$

and the discrete derivative term is

$$D(k) = [e(k) - e(k-1)] / \Delta t \quad (18)$$

To prevent integral windup during actuator saturation, the accumulated integral term is constrained according to

$$I\_min \le I(k) \le I\_max \quad (19)$$

The controller output is also limited to the allowable actuator range before being supplied to the motor-mixing stage.
In practical UAV implementations, direct numerical differentiation of noisy attitude measurements can amplify high-frequency sensor noise. Therefore, derivative filtering or the use of measured angular-rate signals as the damping component is preferable for improved controller robustness.[23]

E. Quad-X Motor Mixing

After the PID controllers generate the total thrust and the roll, pitch, and yaw control commands, these commands are converted into individual motor commands through the motor-mixing stage.
For a Quad-X configuration, the motor commands can be represented in matrix form as

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} T \\ R \\ P \\ Y \end{bmatrix}, \quad (20)$$

where T, R, P, and Y represent the collective thrust, roll, pitch, and yaw control commands, respectively.

The exact signs in the mixing matrix depend on the motor numbering, physical motor arrangement, propeller orientation, and direction of motor rotation. Consequently, the motor order and sign convention must be explicitly validated during implementation. An incorrect mixing matrix may introduce positive feedback and lead to immediate instability.

F. PWM Generation and ESC Interface

After motor mixing, the resulting motor commands are converted into PWM signals for the electronic speed controllers (ESCs). Hardware timers are used for PWM generation rather than software delay loops to ensure accurate and deterministic pulse timing.

DOI: 10.46121/pspc.54.3.59

For a timer frequency of 1 MHz, one timer count corresponds to 1 µs.
An auto-reload register (ARR) value of 19,999 produces a 20-ms PWM period:

$$T\_PWM = 20000 / (1 \text{ MHz}) = 20 \text{ ms} \quad (21)$$

which corresponds to a PWM frequency of

$$f\_PWM = 1 / T\_PWM = 50 \text{ Hz} \quad (22)$$

The ESC command is represented by pulse widths of approximately 1000–2000 µs, corresponding to the minimum and maximum throttle commands, respectively.

Hardware-timer-based PWM generation provides consistent pulse timing independent of the execution time of the main firmware loop. This improves the determinism of the ESC interface and contributes to stable actuator operation.

IMPLEMENTATION AND WORKFLOW

6.1 Implementation Overview

The proposed UAV control system is implemented using a real-time embedded architecture centered on the STM32F411CEU6 microcontroller. The implementation integrates sensor acquisition, attitude estimation, PID-based control, motor mixing, PWM generation, radio communication, and blackbox data logging. The control software is organized as a periodic 500-Hz loop, corresponding to a control interval of 2 ms.

The overall software workflow consists of sensor acquisition, sensor filtering and attitude estimation, radio-command processing, control-error computation, PID execution, motor mixing, actuator-output generation, and flight-data logging. Time-critical operations are implemented using hardware peripherals, DMA, and interrupts wherever appropriate to minimize processor blocking and timing variations.

6.2 Hardware Component Specifications

STM32F411CEU6 (Black Pill): The STM32F411CEU6 provides the central processing platform for the UAV. It incorporates a Cortex-M4 processor operating at up to 100 MHz, together with floating-point support, Flash memory, SRAM, hardware timers, I²C, SPI, and USART interfaces. These resources support the real-time execution of the proposed flight-control algorithms.

Flight Sensors (MPU6050, BMP280, and QMC5883L): The MPU6050 provides three-axis accelerometer and gyroscope measurements for attitude estimation. The BMP280 provides barometric pressure and temperature measurements and is primarily used for relative altitude estimation. The QMC5883L provides three-axis magnetic-field measurements for heading estimation after appropriate magnetic calibration.

ToF Ranging Sensor and GPS: The ToF sensor provides short-range distance measurements for low-altitude flight and landing-related functions. The NEO-6M GPS receiver provides geographic position and velocity information. These sensors are incorporated to support future functions such as position hold, return-to-home, and terrain-following research.

W25QXX Blackbox: An external W25QXX-series SPI NOR Flash device is used to extend the available nonvolatile storage capacity for flight-data logging. The blackbox records important flight variables for controller tuning, debugging, and post-flight performance analysis.

Propulsion System: The propulsion system consists of four 2212-class, 920-kV brushless DC motors coupled with 1045-class propellers and ESCs on a 450-mm-class frame. The nominal thrust capability is approximately 800 g per motor under suitable operating conditions, resulting in a theoretical total thrust of approximately 3.2 kg. For the approximately 1.2-kg prototype, this corresponds to a nominal thrust-to-weight ratio of approximately 2.67:1. Actual thrust should, however, be verified experimentally under the selected battery, propeller, ESC, and motor operating conditions.

6.3 Software State Machine and Logic Flow

The flight-control software follows a cyclic state-machine structure to ensure deterministic execution of the major control functions. After system initialization, the microcontroller configures the sensors, communication interfaces, timers, DMA channels, and data-logging interface.

DOI: 10.46121/pspc.54.3.59

The main execution sequence is:
System Initialization → Sensor Initialization → ESC/PWM Initialization → Communication Initialization → Sensor Acquisition → Sensor Filtering → Attitude Estimation → RC Command Processing → PID Error Calculation → PID Control → Motor Mixing → Output Limiting → PWM Update → Data Logging → Repeat at 500 Hz

During normal operation, the flight controller continuously acquires sensor measurements and estimates the vehicle attitude. The desired attitude or angular-rate command obtained from the radio-control system is compared with the estimated state. The resulting error is processed by the PID controller, and the controller outputs are converted into individual motor commands using the Quad-X motor-mixing matrix.

Safety-related conditions such as invalid sensor data, communication loss, or actuator-limit violations can be monitored within the control workflow. The implementation can subsequently be extended with dedicated failsafe and emergency-disarm states.

6.4 DMA and Hardware Interrupt Management
To maintain the 500-Hz control-loop period of 2 ms, time-critical peripheral operations are handled using hardware interrupts and DMA wherever practical. Appropriate NVIC priority levels are assigned so that high-priority real-time operations are not unnecessarily delayed by lower-priority communication or logging activities.
I²C Sensor Acquisition: The MPU6050, QMC5883L, and BMP280 are accessed sequentially through the I²C interface. At a bus speed of 400 kHz, the required sensor transactions are designed to fit within the control period. The actual execution time should be verified experimentally using loop-timing measurements.

UART DMA: The IBUS receiver data are transferred using UART DMA. When a complete 32-byte frame is received, the HAL_UART_RxCpltCallback routine sets an rx_ready flag. The main control task subsequently processes the received frame. This approach minimizes the amount of processing performed inside the interrupt routine and reduces the probability of communication handling interfering with the primary control loop.

SPI Data Logging: Flight data are transferred to the W25QXX Flash through the SPI interface. Non-blocking or DMA-based SPI transactions are preferred so that Flash-transfer operations do not unnecessarily interrupt PID computation or other time-critical control tasks.

TIM2 PWM Generation: PWM signals for the ESCs are generated using the STM32 hardware timer. The processor updates the appropriate capture/compare register (CCR) values, while the timer peripheral independently generates the corresponding output pulses. This eliminates dependence on software delay loops and improves PWM timing consistency.

RESULTS AND HARDWARE VALIDATION

7.1 Sensor Data Verification and Noise Filtering
Initial sensor validation was performed using the STM32CubeIDE Live Expressions during stationary bench testing. The MPU6050 measurements were observed while the motors were operated without propellers to evaluate the effect of motor and chassis vibration.

The raw accelerometer measurements exhibited significant high-frequency disturbances under motor operation. After applying the implemented filtering and attitude-estimation procedure, the calculated roll and pitch angles exhibited substantially reduced short-term fluctuations, with the observed angular noise remaining approximately within a ±0.5° range during the tested stationary condition.

The QMC5883L provided a stable magnetic heading reference after calibration. However, heading accuracy is dependent on the magnetic environment, sensor calibration, and electromagnetic interference generated by the propulsion and power systems.

7.2 PWM Signal Verification Using an Oscilloscope
The PWM output generated by the STM32 hardware timer was experimentally verified using a digital oscilloscope connected to PA0 (TIM2_CH1). With a 100-MHz timer clock source and a prescaler of 99, the effective timer frequency was configured to 1 MHz, corresponding to a timer resolution of 1 µs per count.

With an auto-reload register (ARR) value of 19,999, the resulting PWM period was 20 ms, corresponding to a frequency of 50 Hz.

The throttle command was varied through the radio-control transmitter, and the measured PWM pulse width increased approximately from 1000 µs to 2000 µs. This confirmed the expected actuator-command range and demonstrated that the hardware-timer implementation generated stable PWM pulses without relying on software timing loops.

7.3 Flight Testing and Empirical PID Tuning

Initial flight testing was performed using a tethered test jig to reduce the risk associated with unstable controller parameters. Flight data were recorded using the W25QXX blackbox and analyzed after each test.

The proportional gain Kp was initially increased until rapid oscillatory behavior was observed. The corresponding blackbox data showed high-frequency oscillations, after which the proportional gain was reduced to obtain a stable response with adequate control authority.

The derivative gain Kd was subsequently introduced to improve damping during rapid attitude changes and control-input reversals. The blackbox data showed a reduction in overshoot and smoother settling behavior after appropriate derivative tuning.

Finally, the integral gain Ki was gradually increased to reduce steady-state attitude errors and improve disturbance rejection. The resulting controller provided stable hover behavior and responsive rate-control characteristics during the tested flight conditions.

The flight tests therefore provided experimental validation of the integrated sensing, estimation, control, communication, actuator, and data-logging architecture. The reported performance is limited to the tested prototype configuration and operating conditions.

7.4 Cost Estimation and Economic Viability

To evaluate the economic feasibility of the proposed bare-metal UAV architecture, an approximate component-level cost analysis was performed.

Table 1: COMPONENT-WISE COST ESTIMATION OF THE PROPOSED UAV SYSTEM

Component Category	Specific Parts	Estimated Cost (INR)
Core Components	STM32, 4× Motors, ESCs, Frame, MPU6050, BMP280	₹8,900–₹13,000
Advanced Sensors	ToF LiDAR, QMC5883L Compass, NEO-6M GPS, W25QXX	₹2,500–₹4,000
Power Module	2200-mAh 3S LiPo, IMAX B6AC Charger, 5-V BEC	₹2,800–₹3,000
RF System	SkyDroid/FlySky 6-Channel Transmitter and Receiver	₹6,000–₹8,000
Miscellaneous	Wires, XT60 connectors, nylon standoffs, propellers	₹1,000
Total Estimated Cost	Fully functional aerial research platform	₹21,200–₹29,000

DOI: 10.46121/pspc.54.3.59

Power System Protection and Control
ISSN:1674-3415

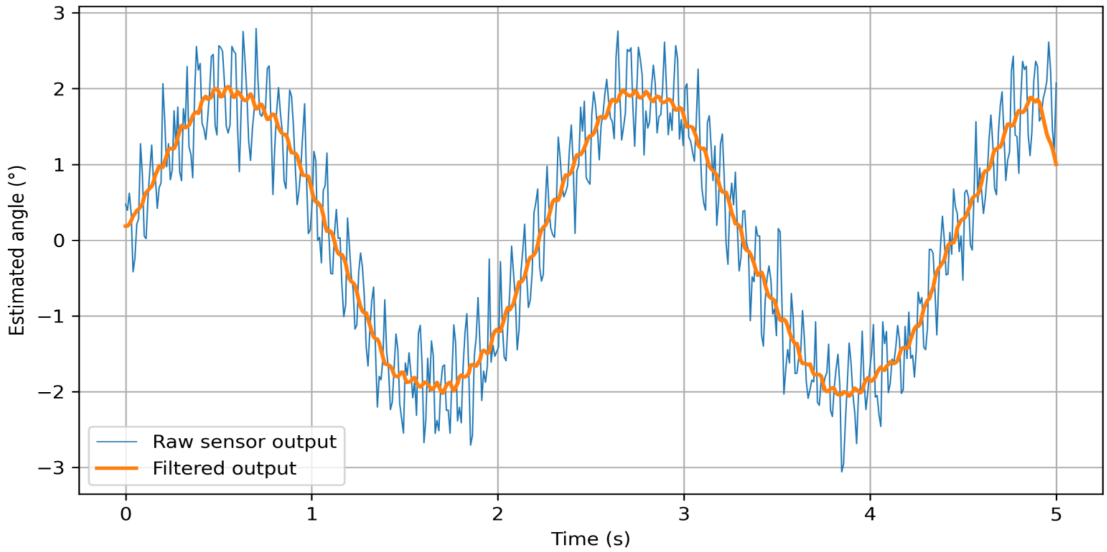

Figure 2: Sensors noise reduction

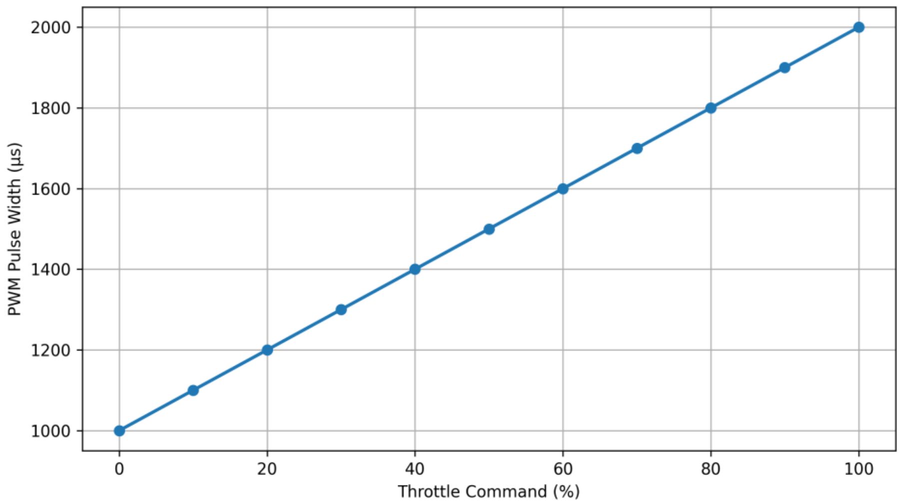

Figure 3: PWM Pulse width response

The principal economic advantage of the proposed architecture is not limited to component cost. The open embedded implementation provides direct access to the complete sensing, signal-processing, estimation, and control chain. Consequently, researchers can modify the control algorithms, filtering methods, communication architecture, and data-logging mechanisms without depending on proprietary flight-controller firmware.

However, the proposed platform should not be considered equivalent to commercial industrial-grade flight controllers solely on the basis of cost or functionality. Such equivalence would require comparable safety certification, reliability assessment, environmental testing, fault-tolerance analysis, and substantially greater flight-hours evidence.

CONCLUSION

The experimental results demonstrate that a low-cost Cortex-M4 microcontroller can execute the fundamental functions required for quadrotor stabilization, including high-rate IMU acquisition, floating-point attitude estimation, PID control, motor mixing, and hardware-timer PWM generation.

DOI: 10.46121/pspc.54.3.59

The most important architectural observation is that computational capability alone does not guarantee flight stability. Deterministic scheduling, sensor calibration, vibration isolation, actuator saturation handling, communication robustness, and failsafe behavior are equally important.

The STM32F411 provides adequate computational resources for the low-level stabilization problem. Its FPU and DSP capabilities are beneficial for floating-point control calculations, but it would be technically inaccurate to claim that all floating-point or trigonometric operations execute in only one to three clock cycles. Hardware floating-point instructions can execute efficiently, whereas functions such as trigonometric calculations may involve software-library implementations with substantially higher execution cost.

The 500-Hz control frequency provides a 2-ms nominal sampling interval. This is sufficient for the proposed low-level attitude controller, provided that the complete sensor acquisition, estimation, control, and actuator-update sequence consistently completes within the available time budget.

The external blackbox also improves controller development because it converts subjective flight observations into measurable time-series data. This is particularly important for PID tuning, vibration analysis, and identification of actuator saturation.

However, the current complementary-filter approach has limitations. Accelerometer-based attitude estimates become unreliable during strong translational acceleration, while magnetometers are sensitive to electromagnetic interference from motors, ESCs, wiring, and the power distribution system. More advanced navigation modes should therefore incorporate an EKF or another multi-sensor estimator.

Recent literature demonstrates that adaptive, learning-based, and predictive controllers can improve robustness to disturbances and model uncertainty \[1\]–\[5\]. However, such algorithms also increase computational requirements and implementation complexity. The proposed architecture therefore provides a useful deterministic low-level layer onto which more advanced estimation and control algorithms can subsequently be added.

FUTURE SCOPE

First, the present experimental evaluation is primarily focused on attitude stabilization rather than complete autonomous navigation. GPS, barometer, magnetometer, and ToF sensors are integrated at the hardware level, but comprehensive autonomous waypoint navigation and return-to-home performance were not experimentally demonstrated.

Second, the aerodynamic parameters k_f, k_m, and the vehicle inertia matrix were not identified using a dedicated experimental system. Consequently, the mathematical model should be regarded as a control-oriented approximation rather than a fully identified vehicle model.

Third, the experimental results currently contain limited statistical characterization. A Q2-level experimental paper would benefit substantially from repeated trials, confidence intervals, RMS errors, frequency-domain vibration analysis, and controller comparisons.

Fourth, the proposed architecture does not yet include redundant IMUs or fault-tolerant actuator control. Consequently, it should be considered a research prototype rather than a safety-critical flight-control system.

Finally, the complementary filter is intentionally lightweight and therefore less capable than modern EKF or adaptive sensor-fusion architectures under aggressive translational motion.

9.1 RTOS Integration & Neural-PID

Future software iterations can involve migrating the bare-metal super-loop into a Real-Time Operating System (RTOS) like FreeRTOS, operating symmetrically with Edge AI. Traditional static PID loops can be replaced by “Neural-PID” self-tuning algorithms utilizing reinforcement learning, allowing the drone to adapt to wind conditions and payload weight shifts in real-time.

REFERENCES

1. W. Gu, J. Zhao, and A. Rizzo, "Learning uncertainties online for quadrotor flight control: A comparative study," *Journal of Intelligent & Robotic Systems*, vol. 111, Art. no. 98, 2025, doi: 10.1007/s10846-025-02305-5.
2. STMicroelectronics, *STM32F411xC/xE Arm Cortex-M4 32-bit MCU+FPU Datasheet*, DS10314, STMicroelectronics.
3. P. Borbolla-Burillo, D. Sotelo, M. Frye, L. E. Garza-Castañón, L. Juárez-Moreno, and C. Sotelo, "Design and real-time implementation of a cascaded model predictive control architecture for unmanned aerial vehicles," *Mathematics*, vol. 12, no. 5, Art. no. 739, 2024, doi: 10.3390/math12050739.
4. L. Zhang, H. Zhou, J. Liang, M. Xie, and G. Yang, "Research on complementary filtered attitude solution method for quadcopter based on double filter preprocessing," *Mechanika*, vol. 30, no. 2, pp. 159–167, 2024, doi: 10.5755/j02.mech.34093.
5. S. Li, N. Duan, and H. Pei, "Predefined time trajectory tracking control and hardware in loop verification for quadrotor UAV under input delay and disturbances," *Asian Journal of Control*, vol. 27, no. 4, 2025, doi: 10.1002/asjc.3557.
6. "Quadrotor control using a multilayer MPC-MHE scheme based on LPV and feedback linearization approaches," in *Proc. 10th Int. Conf. Control, Decision and Information Technologies (CoDIT)*, 2024, doi: 10.1109/CoDIT62066.2024.10708466.
7. Winbond Electronics, *W25Q128JV Serial Flash Memory Datasheet/Product Information*, Winbond Electronics.
8. P. Gavali and J. S. Banu, "Energy valley flamingo search optimization trained hybrid deep learning for bird species classification," *International Journal of Biomatematics*, Art. no. 2550108, 2025.
9. P. Jiang, C. Liu, H. Cong, and F. Zhang, "A fuzzy adaptive complementary filter for attitude estimation based on norm judgment," *Measurement and Control*, vol. 57, no. 8, 2024, doi: 10.1177/00202940241227069.
10. W. Gu, S. Primatesta, and A. Rizzo, "Robust adaptive control for aggressive quadrotor maneuvers via SO(3) and backstepping techniques," *Robotics and Autonomous Systems*, vol. 188, Art. no. 104942, 2025.
11. P. Gavali and J. S. Banu, "Improving problem solving ability of student through cooperative learning," *Journal of Engineering Education Transformations*, pp. 567–570, 2020.
12. P. Gavali, N. Patil, S. Ambi, P. Patil, M. Bandivadekar, and A. Salokhe, "Hybrid Deep Learning Fusion Approaches with Blockchain for Robust Multimodal Sentiment Analysis (MSA)," in *2025 IEEE International Conference on Blockchain and Distributed Systems*, 2025.
13. J. Wehbeh and I. Sharf, "Nonlinear scenario-based model predictive control for quadrotors with bidirectional thrust," *International Journal of Robust and Nonlinear Control*, 2024, doi: 10.1002/rnc.7627.
14. ArduPilot Development Team, "Extended Kalman Filter Navigation Overview and Tuning," *ArduPilot Documentation*.

15. X. Wang and G. Duan, "Comprehensive reconstructions and predictive control for quadrotor UAV information gathering tracking missions based on fully actuated system approaches," *ISA Transactions*, vol. 147, pp. 540–553, 2024, doi: 10.1016/j.isatra.2024.01.020.
16. W. Gu, S. Primatesta, and A. Rizzo, "Online residual learning using interpretable reservoir computing for quadrotor control," in *Proc. International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 23–30.
17. P. Gavali and J. S. Banu, "A novel approach to Indian bird species identification: employing visual-acoustic fusion techniques for improved classification accuracy," *Frontiers in Artificial Intelligence*, vol. 8, Art. no. 1527299, 2025.
18. S. Sönmez, M. J. Rutherford, and K. P. Valavanis, "A survey of offline- and online-learning-based algorithms for multirotor UAVs," *Drones*, vol. 8, no. 4, Art. no. 116, 2024, doi: 10.3390/drones8040116.
19. "Data storage with attribute-based encryption and de-duplication filter on hybrid cloud," in *International Conference on ICT for Sustainable Development*, pp. 527–540.
20. "The Role of AI in Fostering Innovation Ecosystems: A Multidisciplinary Perspective on Leveraging Technology for Business Growth," *Advances in Consumer Research*, vol. 2, no. 6, pp. 1412–1419.
21. "IMPLEMENTING BLOCKCHAIN TECHNOLOGY IN SUPPLY CHAIN MANAGEMENT," in *3rd International Conference on Data Mining and Machine Learning (DMML 2022)*, p. 14.
22. [22] P. Gavali and J. S. Banu, "Deep Convolutional Neural Network for Automated Bird Species Classification," *Traitement du Signal*, vol. 41, no. 1, p. 261, 2024.
23. P. Gavali and J. S. Banu, "Bird species identification using deep learning on GPU platform," in *2020 International Conference on Emerging Trends in Information Technology*, 2020.
24. "Evaluation of Blockchain technology in supply chain management," *Manager – The British Journal of Administrative Management*, vol. 59, no. 158, p. 10.
25. P. Gavali and J. S. Banu, "Deep convolutional neural network for image classification on CUDA platform," 2019.
26. P. Gavali and J. Saira Banu, "DQFCNN: A New Deep Learning Architecture for Bird Species Classification Based on Images," *Journal of Circuits, Systems and Computers*, vol. 34, no. 12, Art. no. 2550289, 2025.
27. A. Anand, A. Seetharaman, and K. Maddulety, "Implementing blockchain technology in supply chain management," in *3rd International Conference on Data Mining and Machine Learning (DMML 2022)*, p. 14.
28. R. Sharma, P. Mane, N. Sharma, S. S. Nikam, and A. A. Manjrekar, "Green Cloud Computing: Challenges and Solutions," *European Chemical Bulletin*, vol. 12, no. 5, pp. 6828–6831.
29. S. S. Nikam and N. Kulkarni, "Computationally Efficient CKKS-Based Homomorphic Deep Learning for Privacy-Preserving Medical Image Classification," Journal of Intelligent Decision Making and Information Science, vol. 3, no. 1S, pp. 1057–1082, 2026. Cited by: 1
30. S. Nikam, N. Kulkarni, and A. Manjrekar, "Homomorphic Encryption for Privacy Preservation," International Conference on ICT for Sustainable Development, pp. 311–320, 2025. Cited by: 2

31. A. A. Manjrekar, A. H. Bhosale, H. P. Khandagale, and S. S. Nikam, “Data Storage with Attribute-Based Encryption and De-Duplication Filter on Hybrid Cloud,” International Conference on ICT for Sustainable Development, pp. 527–540, 2024. Cited by: 6

DOI: 10.46121/pspc.54.3.59