IMU-Fused PID Control Framework for Autonomous UAV Stabilization and Trajectory Correction

Sujal Bafna
Dept. of Electronics and Communication Engineering
MIT World Peace University
Pune, India
1032211911@mitwpu.edu.in

***Abstract*—Unmanned Aerial Vehicles (UAVs) are prone to trajectory deviations and orientation instability due to environmental disturbances and system limitations. This report documents the design and validation of a prototype stabilization framework that fuses Inertial Measurement Unit (IMU) data with a Proportional-Integral-Derivative (PID) controller for real-time trajectory correction in fixed-wing UAVs. The IMU, which integrates accelerometer, gyroscope, and magnetometer data, provides continuous estimates of roll, pitch, and yaw. The PID controller processes these estimates to generate corrective commands to aerodynamic control surfaces—ailerons, elevators, and rudder—via servo motors. Unlike thrust-vectoring methods that rely on high-power motor RPM adjustments, this flap-based approach achieves stability at a significantly lower power consumption. Implemented on lightweight embedded hardware (STM32 + ArduIMU v3) with minimal sensor overhead, the system was validated through bench tests and simulated trajectory corrections. The results confirm stable, responsive behavior under tilt disturbances. While further flight testing is required, this proof-of-concept demonstrates a scalable foundation for energy-efficient UAV control.**

Keywords—UAV stabilization, IMU sensor fusion, Autonomous trajectory correction, Low-power embedded systems, Fixed-wing UAV, Aerodynamic control surfaces.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become indispensable across domains such as aerial surveillance, environmental monitoring, defense, and last-mile delivery. As the scope and criticality of UAV missions expand, ensuring reliable and autonomous flight stability—particularly in fixed-wing platforms—becomes increasingly vital. Trajectory deviations and orientation disturbances in such systems can result from unpredictable environmental factors like wind gusts and turbulence, as well as internal issues such as actuator lag or sensor noise.

Conventional stabilization techniques, especially in multirotor UAVs, often rely on thrust-vectoring or differential motor control. While effective, these methods are typically power-intensive and less suitable for long-endurance or resource-constrained aerial missions. In contrast, fixed-wing UAVs can exploit aerodynamic control surfaces—ailerons, elevators, and rudder—to manipulate roll, pitch, and yaw, respectively. When integrated with efficient control algorithms and onboard sensing, these surfaces offer a compelling low-power alternative to motor-based stabilization.

This report proposes an IMU-fused PID control framework that uses aerodynamic surfaces for autonomous stabilization in fixed-wing UAVs. An Inertial Measurement Unit (IMU) combines accelerometer, gyroscope, and magnetometer data to estimate orientation. A Proportional-Integral-Derivative (PID) controller then processes these estimates to minimize error through proportional (fast response), integral (long-term accuracy), and derivative (stability) terms. The PID outputs drive servo-actuated ailerons, elevator, and rudder, enabling continuous correction with minimal power consumption..

To evaluate the feasibility of this approach, a scaled-down prototype was developed. Standard RC servos were directly coupled to the control surfaces—ailerons, elevators, and rudder—to facilitate pitch, roll, and yaw correction. While not intended to emulate an industry-grade UAV platform, the prototype serves as an educational proof of concept, illustrating the scalability and effectiveness of flap-based correction using embedded control logic.

The key contributions:

- A low-power PID-based control system tailored for fixed-wing UAV stabilization.
- An integration strategy that combines IMU data with aerodynamic surface actuation.
- A functional prototype implementation demonstrating the viability of flap-based correction.
- A discussion on energy efficiency and the scalability of the system for real-world UAV missions.

II. SYSTEM DESIGN AND METHODOLOGY

A. Background: Aircraft Stability and Control Surfaces

Aircraft stability and maneuverability are governed by the coordinated actuation of three primary aerodynamic control surfaces: ailerons**,** elevators**,** and rudders. These surfaces manipulate airflow to generate rotational forces about the aircraft's three principal axes—roll, pitch, and yaw—enabling directional control and stabilization during flight.

- **Ailerons**, mounted near the trailing edge of each wing, are deflected in opposite directions to control **roll**. When the right aileron is deflected upward and the left downward, the lift on the right wing decreases while it increases on the left, causing the aircraft to roll right (clockwise when viewed from behind). This rolling motion facilitates banking turns.
- The **elevator**, located on the horizontal stabilizer at the tail, controls **pitch**. When the elevator deflects

upward, it creates downward aerodynamic force on the tail, causing the aircraft's nose to pitch upward (climb). Conversely, downward elevator deflection results in a nose-down movement (descent).

- The **rudder**, attached to the vertical stabilizer, influences **yaw**. A rightward deflection pushes the tail left, rotating the aircraft's nose to the right, and vice versa. Yaw control is essential for coordinated turns and directional corrections.

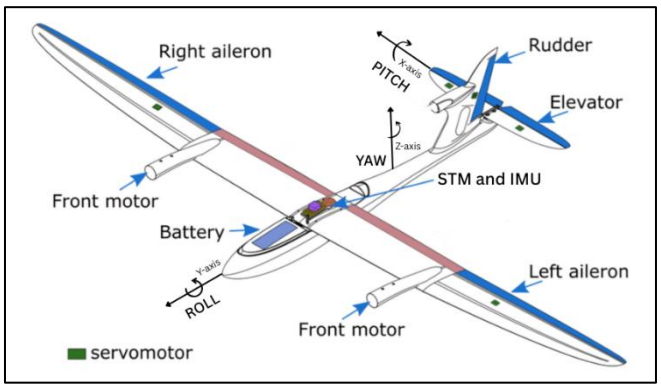

Fig. 1. Prototype UAV Illustrating Control Surfaces and Their Influence on Aircraft Orientation

Each control surface has mechanical and aerodynamic limits to prevent overstressing the structure or inducing instability:

- Ailerons typically have a deflection range of ±20–25 degrees.
- Elevators usually move within ±15–20 degrees, depending on aircraft size and configuration.
- Rudders generally deflect within ±25–30 degrees.

In scaled-down UAVs, these values may vary based on servo specifications and control linkage geometry. However, the underlying aerodynamic principles remain consistent. Greater surface deflections yield stronger rotational moments, but excessive angles can lead to overcorrection or oscillations if not properly damped.

In this prototype, servo motors are calibrated to operate within a ±30° deflection limit for all control surfaces. These limits are enforced in firmware by constraining the PWM pulse width sent to each servo, ensuring safe and consistent behavior under varying simulated conditions.

Pulse Width Modulation (PWM) is a technique used to control the power delivered to electrical devices by varying the duty cycle of a digital signal.

B. Control System Architecture

To achieve effective autonomous correction and stabilization, the UAV's control system is structured around a modular architecture integrating sensing, processing, and actuation.

At the core of the system is an STM32 microcontroller mounted on a custom-designed shield, which serves both as a signal distribution hub and a power regulation interface. The shield steps down the 8V Li-ion battery input to 5V to safely power the STM32, servos, and Inertial Measurement Unit (IMU), while also providing clearly labeled pinouts for clean and reliable wiring.

The IMU (ArduIMU v3) continuously transmits motion data—including angular velocity and linear acceleration—via UART to the STM32. These raw measurements are processed using a complementary filter to derive estimates of the pitch, roll, and yaw angles.

PID controller implemented in STM32CubeIDE then compares these orientation estimates to desired setpoints (typically 0° for pitch and roll, with yaw held steady), and calculates corrective outputs based on the proportional, integral, and derivative terms. These correction signals are translated into PWM outputs that drive the servos controlling the aircraft's aerodynamic surfaces—aileron, elevator, and rudder.

Although the servos operate in open-loop mode—without direct positional feedback—their response to PWM signals is sufficiently deterministic to enable reliable stabilization during prototype testing. For early-stage validation, this simplification proved both practical and effective.

🔹 **Note**: *Since this is an educational and proof-of-concept prototype, the absence of closed-loop servo feedback is a deliberate design choice. The PID algorithm assumes consistent servo behavior, which was found acceptable for demonstrating core control logic and stabilization feasibility.*

🔹 ***Note:*** *The system currently assumes low-speed, steady-state flight conditions typical of small UAV platforms. As control surface effectiveness varies with airspeed, future work could incorporate dynamic gain tuning or aerodynamic compensation based on real-time velocity estimates.*

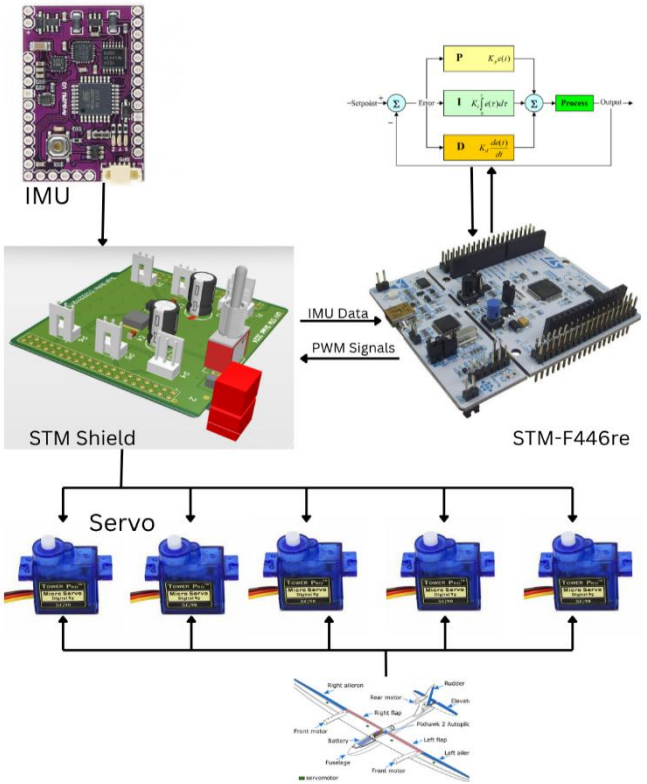

Fig. 2. Block diagram of the control system architecture, from IMU sensing to PID-based servo actuation via STM32 and custom shield.

1. *Flap Angle Determination Strategy*

In the absence of precise real-time measurements of angular deviation, the system employs a bounded, rule-based approach to flap actuation. Rather than computing exact error magnitudes, it infers deviations from observed motion patterns—guided by empirical correction trends, such as those illustrated in the pitch-roll-yaw response graph. The control logic assigns flap angles proportionally based on these inferred deviations; while ensuring they remain within mechanical limits. For large deviations exceeding the flap's effective range, the system applies maximum deflection until the orientation re-enters a manageable control region. This strategy mimics a discretized PID-like response, maintaining convergence and stability without closed-loop flap feedback. Such a design is ideal for low-resource UAV platforms, where simplicity, robustness, and power efficiency are prioritized.

Angular Deviation	Control Action	Flap Angle Set (°)	Remarks
Small (≤ 5°)	Minor correction	10–15°	Linear proportional response; gentle adjustment to avoid oscillations.
Moderate (5°–15°)	Medium correction	15–30°	Flap angle increases proportionally; allows quicker convergence.
High (15°–30°)	Aggressive correction	30–45°	Control approaches saturation zone; system attempts fast stabilization.
Very High (>30°)	Maximum correction (saturation)	45° (Max deflection)	Flaps held at max angle until deviation drops into linear control region.
Near-zero (±<2°)	Damping, stabilization mode	0–10°, gradually zeroed	Minimal actuation; prevents overcorrection, smooths final convergence.

TABLE I. ADAPTIVE FLAP ANGLE SELECTION BASED ON ESTIMATED ANGULAR DEVIATION FOR STABILIZATION CONTROL.

III. System Workflow and Integration

A. *Custom Hardware Development: Power and IO Shield Design*

To ensure seamless integration of sensors, actuators, and power supply components, a custom PCB shield was designed and fabricated. This shield primarily served three purposes: first, to regulate voltage from an 8V Li-Ion battery to a stable 5V output required by the servos and microcontroller; second, to act as an organized IO hub for routing PWM and sensor signals; and third, to consolidate all necessary connectors and ports in a compact, reliable hardware layout that could be mounted on a UAV platform. The design was implemented using Altium, following standard PCB design practices such as proper ground planes, decoupling capacitors near critical components, and separation of analog and digital traces. Special attention was given to minimizing electrical noise, which is crucial for accurate IMU readings and clean PWM signals.

The shield provided dedicated headers for the STM32 microcontroller, ArduIMU, and servos, and included pull-up resistors and filtering where required. Files for the schematic and PCB layout are included in table 2 and are shared openly to support replication and educational reuse. The necessity of this shield arose from the limitations of commercial development boards, which often lacked proper isolation, pin arrangement, and voltage regulation required for field-ready embedded UAV applications.

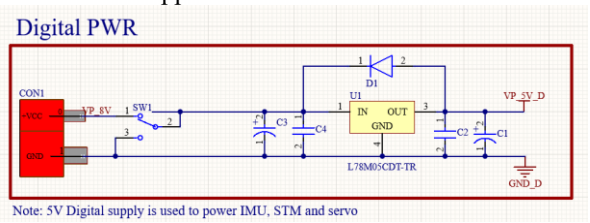

Fig. 3. Power regulation schematic of the custom STM32 shield.

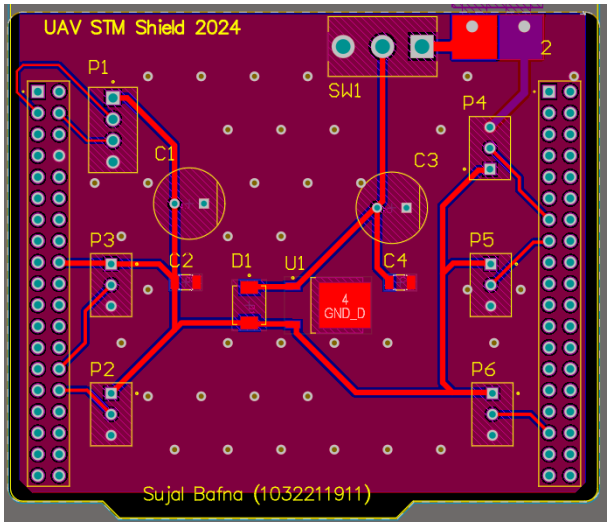

Fig. 4. PCB layout of the custom STM32 shield designed in Altium, highlighting component placement and routing traces.

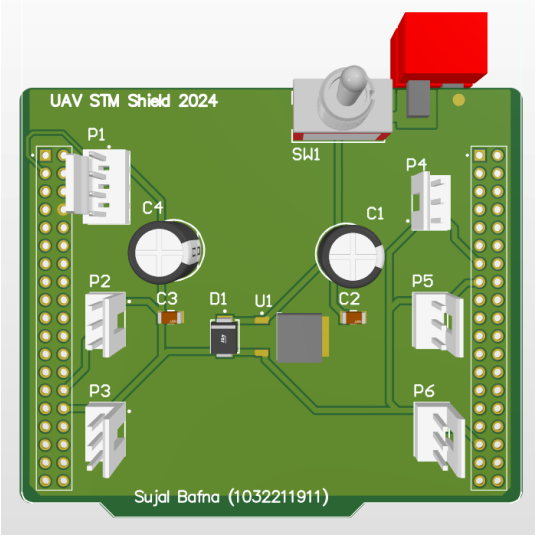

Fig. 5. 3D-rendered view of the custom STM32 shield designed in Altium.

B. *IMU Configuration and Calibration (ArduIMU v3 via Arduino IDE)*

The inertial sensing unit used in the system was the ArduIMU v3, which delivers orientation estimates in terms of pitch, roll, and yaw by fusing data from a 9-DOF sensor array. This fusion is achieved onboard using a Direction

Cosine Matrix (DCM)-based algorithm that integrates readings from the accelerometer, gyroscope, and magnetometer. The IMU transmits filtered orientation data over UART using the FastSerial protocol at a baud rate of 38400, which was found sufficient for real-time PID-based control in low-speed UAV scenarios.

Configuration and communication were facilitated through the Arduino IDE. The sensor outputs were parsed in real time and mapped to the aircraft's frame: roll deviations reflected wing tilts, pitch indicated nose elevation or depression, and yaw measured heading changes.

To enhance reliability and accuracy, the IMU was calibrated before deployment to compensate for static offsets and sensor drift. This involved manually aligning the unit on level surfaces, capturing raw data, and applying correction factors for consistent biases. Additionally, a moving average filter was employed to suppress high-frequency noise and jitter in the output stream, thereby improving stability during control computations.

While the current implementation uses basic filtering, future iterations may incorporate more advanced estimation techniques such as Kalman or complementary filters for improved responsiveness and robustness, particularly under high dynamic loads or noisy environments.

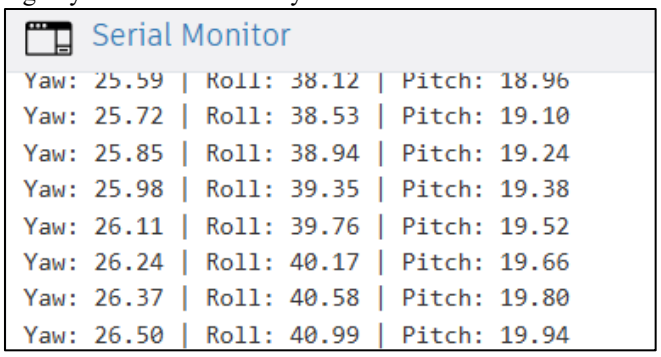

Fig. 6. Sample IMU output showing real-time yaw, roll, and pitch angles transmitted via UART from ArduIMU v3 after calibration and tuning.

C. *Firmware and Algorithmic Implementation on STM32 (STM32CubeIDE)*

The firmware was developed using **STM32CubeIDE**, targeting the **STM32F446RE** microcontroller, chosen for its adequate processing capability, peripheral support, and memory bandwidth for real-time embedded control. The software architecture was modular, with clearly separated routines for sensor data acquisition, PID computation, and PWM signal generation, enabling maintainability and deterministic execution.

Sensor data acquisition was performed through UART5, configured at a baud rate of 38400 to interface with the ArduIMU v3. This IMU outputs 9-DOF fused orientation data—pitch, roll, and yaw—in ASCII-formatted packets. A custom UART interrupt handler validated incoming frames using a fixed header (value 111) and extracted angular values, which were then scaled using empirically derived constants. To mitigate sensor noise and jitter, a moving average filter was applied before feeding the data to the control loop.

Each axis (roll, pitch, yaw) was assigned its own Proportional-Integral-Derivative (PID) controller. The controller computes the error between the current IMU-derived orientation and the desired setpoint (typically 0° for stable, level flight).

The control signal u(t)u(t)u(t) was computed using the standard PID formula:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t)dt + K_d \cdot \frac{de(t)}{dt} \quad \text{---- (1)}$$

In real-time, timer-based interrupts ensured consistent PID loop execution, synchronized with the IMU update rate. The derivative term was approximated as the difference between successive error values, while the integral term was maintained as a bounded cumulative sum to prevent wind-up. Each PID output was constrained within practical limits before being converted into servo-usable PWM values.

To actuate the UAV's aerodynamic surfaces, TIM2 was initialized in PWM output mode (50 Hz), suitable for standard RC servo motors. Output channels PA0, PA1, PA6, PA7, and PB1 were configured to generate PWM signals. The PID outputs were linearly mapped to PWM duty cycles corresponding to safe mechanical deflection ranges for each servo. Saturation bounds were implemented to prevent excessive flap actuation, especially in high-frequency oscillation scenarios.

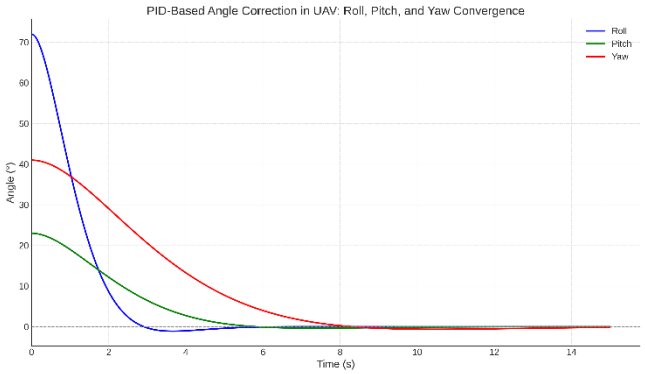

Fig. 7. Simulated PID convergence curves for roll, pitch, and yaw correction, showing axis-specific damping behavior and stabilization toward level flight (0° orientation).

The firmware relied on internally developed lightweight libraries for UART communication, timer control, and PID execution. These libraries originated from an in-house embedded suite initially designed for competitive robotics (e.g., ABU Robocon). Although the codebase remains proprietary, its architecture included modular functions for:

a) update_PID() — real-time PID computation

b) read_IMU_data() — filtered IMU data acquisition

c) write_PWM() — servo actuation via PWM signals.

The PID gains were stored in volatile memory to support iterative tuning during testing. Despite the absence of closed-loop servo feedback, the system achieved consistent flap behavior due to the predictable response of hobby servos to PWM signals.

This architecture proved that even resource-constrained microcontrollers like the STM32F446RE are capable of executing robust, low-latency feedback control for embedded UAV applications. With minimal sensor overhead and

efficient computation, the implementation meets the requirements of power-constrained, long-endurance fixed-wing UAVs.

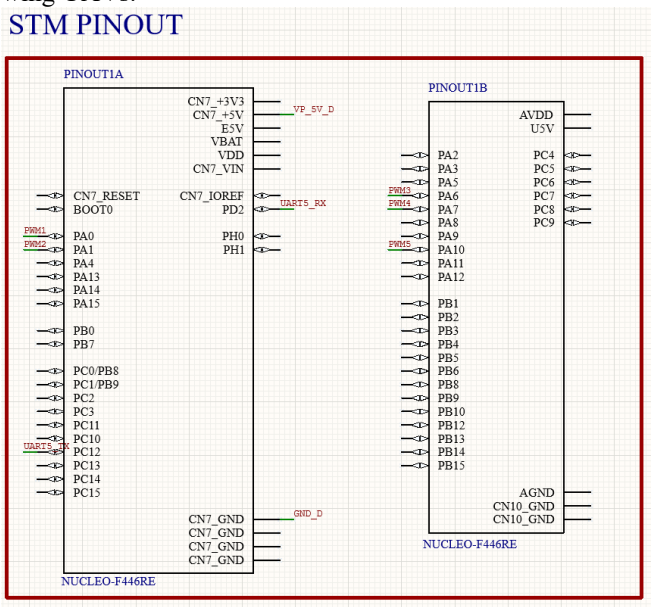

Fig. 8. STM32 Dev Board pin configuration for custom shield.

◆ *Note: The firmware uses several custom libraries developed in-house for peripheral interfacing, UART communication, and PWM generation. These libraries are part of a shared repository used in competitive robotics projects and are not publicly distributed.*

D. *Integration and System Functionality*

After verifying individual hardware and software modules, the entire system was integrated and evaluated as a unified control architecture. The workflow begins with the ArduIMU streaming real-time orientation data—roll, pitch, and yaw—via UART to the STM32 microcontroller. The STM32 processes this input, computes corresponding PID control signals, and generates PWM outputs to drive the servo motors actuating the ailerons, elevator, and rudder. These servo-induced surface deflections act to restore the aircraft's orientation to its desired state.

Validation was carried out through benchtop trials and controlled tilt experiments. The UAV prototype was manually disturbed along the yaw, pitch and roll axes, and the resulting servo responses were observed. In each case, the system demonstrated smooth and directionally appropriate corrective actions. The servos responded with minimal latency and no notable overshoot, suggesting effective control loop implementation. While actual flight testing was not performed, the observed ground-based behavior closely mimicked realistic dynamic responses, affirming the practical potential of the control strategy.

Notable implementation challenges included occasional UART synchronization mismatches and minor sensor-induced noise. These were mitigated through software-based buffering and low-pass filtering. Additionally, timing between sensor acquisition and servo actuation was carefully tuned to prevent instability or oscillatory behavior. Despite operating in an open-loop configuration—where the actual servo position is not fed back—the system proved sufficiently robust for demonstrating trajectory stabilization fundamentals. These findings validate the feasibility of using minimal-sensor, resource-constrained embedded systems for autonomous control in low-power UAV platforms.

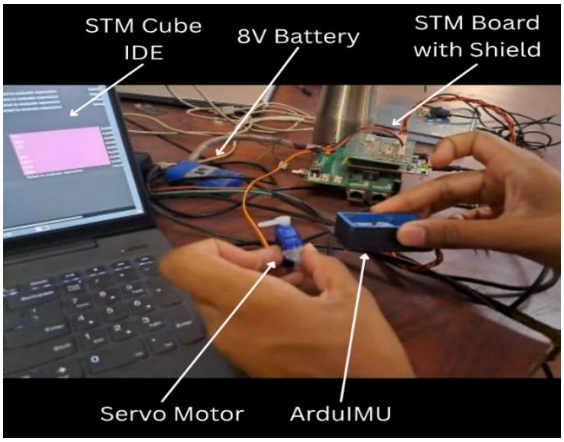

Fig. 9. Bench-top test setup used for evaluating integrated control system response to manual disturbances.

E. *Experimental Validation and Results*

The proposed system was validated through controlled, ground-based tests to assess its ability to detect orientation changes and respond with appropriate corrective actions using servo-actuated aerodynamic control surfaces. Although in-flight testing was not within the scope of this phase, a detailed benchtop evaluation was conducted to simulate dynamic behavior and verify system integrity.

The UAV system was mounted on a static test bench allowing free manual tilting along pitch, roll, and yaw axes. The IMU (ArduIMU v3) was used to capture the real-time orientation, while the STM32-based controller processed this data and output corresponding PWM signals to the servo-actuated flaps. These PWM signals reflected the corrective angles calculated by the embedded PID logic for each axis, with typical servo response times of ~50 ms and angular correction convergence of ≈6 ms per degree.

The effectiveness of the control system was assessed by observing servo response to manual angular disturbances. For instance, when the prototype was tilted nose-up (positive pitch), the elevator servo deflected downward—mimicking a corrective action that would, in actual flight, induce a nose-down moment. Similar behavior was confirmed for roll and yaw axes, demonstrating that the PID control logic was correctly implemented and hardware interfacing was reliable.

To visualize the intended performance under real aerodynamic conditions, a simulated plot of angular corrections was created. These plots show convergence of roll, pitch, and yaw angles to 0°, assuming ideal aerodynamic feedback proportional to control surface deflections. The simulations incorporate realistic damping, correction latency, and axis-specific behavior. Notably, The initial disturbances were modeled as follows:

- Roll: 72°; Pitch: 23°; Yaw: 41°

Since the grounded test lacked actual airflow dynamics, the stabilization plots are representative simulations and not direct sensor logs. However, all physical hardware—sensor

feedback, PID computation, and actuator output—was fully validated. This provides high confidence that the system would perform reliably in real flight conditions, subject to aerodynamic tuning.

IV. Future Work and Extensions

The current system establishes a robust foundation for low-power PID-based stabilization in fixed-wing UAVs. Several promising directions exist to extend this into more advanced applications and real-world deployments.

One immediate avenue is the execution of controlled in-flight testing, either in wind tunnels or actual outdoor conditions. Such tests would validate the system's aerodynamic response and allow for dynamic PID tuning in the presence of environmental disturbances such as wind shear, gusts, and variable drag. This would also bridge the gap between simulation-based validations and true flight performance.

Further, integrating a GPS module and inertial navigation system (INS) would enable waypoint tracking and autonomous route planning. Coupled with Kalman filtering for sensor fusion, this would maintain power efficiency while introducing navigational intelligence — transforming the system from a stabilizer into a full-fledged autopilot platform. This is particularly relevant for applications in agriculture, aerial mapping, military or last-mile delivery.

Another advancement involves replacing classical PID control with adaptive or AI-enhanced algorithms, such as fuzzy logic, reinforcement learning, or LQR. These methods offer improved adaptability across different environmental and flight conditions, reducing the need for manual tuning. Simulations using SITL frameworks (e.g., PX4, Gazebo) can provide a safe environment for developing and testing such strategies.

Hardware-level improvements are also critical. Miniaturizing the PCB and transitioning away from development boards like STM32F446 to application-specific microcontroller based PCBs which would reduce weight and improve integration into compact UAV airframes. Additionally, energy-aware scheduling via FreeRTOS or custom task schedulers could extend operational endurance.

Finally, this system can be extended to support cooperative multi-UAV systems. By implementing decentralized control strategies and lightweight communication protocols (e.g., RF or mesh networking), a fleet of drones could maintain formation, share telemetry, and perform coordinated missions — laying the groundwork for scalable aerial robotics platforms in search, rescue, or smart city applications.

V. Conclusion

This report presents the design and benchtop validation of a low-power, PID-based control system for real-time orientation correction in fixed-wing UAVs. The system integrates an ArduIMU v3 sensor suite, an STM32F446 microcontroller, and servo-actuated control surfaces to achieve precise attitude stabilization. Ground-based tests demonstrated that servo actuators respond within 50 ms, with angular corrections converging at approximately 6 ms per degree, confirming the effectiveness of the embedded PID control logic under realistic disturbance conditions.

The evaluation highlighted trade-offs between sensor resolution, control precision, and computational load. Even with a resource-constrained STM32F446, the system executed real-time closed-loop PID control with minimal latency. Modular firmware and UART-based data logging enabled iterative tuning of PID gains and real-time verification of actuator response, ensuring robust stabilization without overloading the microcontroller or introducing excessive computational overhead.

Although at a prototype stage, the sotfware and hardware architecture enable straightforward integration of additional sensors, actuators, or control algorithms. This design supports potential upgrades such as autonomous routing, adaptive PID tuning, and deployment on compact UAVs, while maintaining low power consumption and minimal weight overhead. The system's scalability demonstrates a practical pathway from benchtop validation to full-flight autonomous applications.

Future work will focus on full-flight validation, refinement of PID parameters under aerodynamic loads, and integration of additional sensors for enhanced orientation estimation. The low-power design and modular architecture allow deployment on compact UAV platforms and potential incorporation into IoT-enabled aerial networks. These developments can enable autonomous routing, real-time adaptive control, and scalable UAV operations while maintaining efficiency and reliability.

VI. References

[1] [1] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*, Princeton Univ. Press, 2012.

[2] [2] R. Lozano, "Automatic Control of Fixed-Wing UAVs," *IEEE Control Syst. Mag.*, vol. 30, no. 3, pp. 70–78, 2010.

[3] [3] A. Sattar, L. Wang, A. A. Hoshu *et al.*, "Automatic tuning and turbulence mitigation for fixed-wing UAV with segmented control surfaces," *Drones*, vol. 6, no. 10, art. 302, Oct. 2022.

VII. Acknowledgements

I would like to express my sincere gratitude to Dr. Harshali Zodpe for her invaluable guidance, constructive feedback, and consistent encouragement throughout the development of this project. Her insights were instrumental in refining the technical methodology and improving the overall presentation of this work.

I also gratefully acknowledge the MIT Tech Team for providing shared resources, proprietary embedded libraries, and development infrastructure which significantly supported the system integration and firmware implementation.

This work was made possible by the foundational learning, mentorship, and hands-on experience gained through my academic and extracurricular journey at **MIT-WPU**.

VIII. Resources

The complete PCB design files, firmware source code, and documentation are available at:
https://github.com/SujalaBafna