Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 1, 2026

DESIGN AND IMPLEMENTATION OF PID-BASED DRONE STABILIZATION THROUGH HARDWARE-IN-THE-LOOP SIMULATION

Zeeshan Amjad[1], Abdullah Ans[2], Abrar Younus[3], Waheed Ur Rehman[*4], Hasan Muhammad Fahad[5], Farrukh Saleem[6], Muhammad Arqum Razzaq[7]

[1,2,3,*4,7] *Department of Mechatronics Engineering, University of Chakwal, Chakwal 48800, Pakistan*
[5]*Department School of Astronautics Northwestern polytechnical University Xian China.*
[6]*Ningbo Geely Automobile Research and Development Co., LTD, Hangzhou Bay, Ningbo, Zhejiang, China*

zeeshan.mte07@gmail.com[1], abdullahansmughal@gmail.com[2], abraryounus21@gmail.com[3], waheedurrehman@uoc.edu.pk[4], 2022180033@mail.nwpu.edu.cn[5], farrukh.saleem@geely.com[6], arqum.razzaq@uoc.edu.pk[7]

DOI:https://doi.org/10.5281/zenodo.20508500

Keywords
PID Control, Drone Stabilization, BLDC Motors, MPU6050 Sensor, Arduino Uno, 3D Printing, Real-time Adjustment

Article History
Received: 15 December 2025
Accepted: 19 January, 2026
Published: 31 January, 2026

Corresponding Author:

Waheed Ur Rehman

Abstract: In recent years, unmanned aerial vehicles (UAVs), commonly known as drones, have gained significant attention due to their versatility and wide range of applications. Drones are increasingly being utilized in emergency response operations, military missions, aerial photography and filmmaking, precision agriculture, surveillance, environmental monitoring, and recreational activities. Despite their growing popularity, maintaining flight stability remains one of the most critical challenges in drone operation. Stability is disturbed due to disturbance in the form of wind storm or maybe due to nonlinearities created by mechanical changes in structure. Current research develops 3D printed drone along with hardware in loop simulation of drone to improve its disturbance rejection performance. Experimental results demonstrate that the proposed system successfully maintained stable flight and effectively compensated for disturbances under different loading conditions. The telemetry analysis confirmed satisfactory tracking performance and improved disturbance rejection capability. The findings of this study highlight the effectiveness of integrating Hardware-in-the-Loop simulation with a 3D-printed drone platform, providing a reliable and economical solution for developing robust UAV systems capable of operating in dynamic and uncertain environments.

1. INTRODUCTION

Drones have become increasingly versatile, finding applications in numerous fields. They are used in emergency response to deliver medical supplies and in search and rescue operations. In the military, drones are employed for surveillance and targeted strikes. In agriculture, drones assist in crop monitoring and spraying pesticides. Film production benefits from drones' ability to capture aerial footage, while recreational use continues to grow among hobbyists.

Despite their widespread applications, drones face several issues, primarily related to stability and control. Environmental factors such as wind and turbulence can significantly impact their flight performance. Structural nonlinearities and mechanical changes can also introduce instability. Moreover, the integration of advanced control systems, such as PID controllers, is crucial to enhance their reliability and performance under various operating conditions.

The use of PID control systems in drones has been extensively studied due to their simplicity and effectiveness in maintaining stability. One study demonstrated the application of PID controllers in quadcopter drones, highlighting their ability to manage dynamic stability under different flight conditions [1]. Another research explored PID-based stabilization in Hexacopters, focusing on disturbance rejection in windy environments [2].

Recent advancements in HIL simulation have further enhanced the testing and development of drone control systems. A study utilized HIL simulation to test the robustness of PID controllers in UAVs, showing significant improvements in real-world performance [3]. Furthermore, the integration of 3D printing technology in drone manufacturing has allowed for rapid prototyping and customization, making it easier to iterate on designs and improve control systems [4].

The MPU6050 sensor is widely used in drone stabilization due to its high accuracy and low cost. Research has shown that the MPU6050 provides reliable orientation data, which is crucial for the PID algorithm to function effectively [5]. The Arduino Uno, with its versatility and ease of programming, has been a popular choice for implementing control systems in various drone projects [6].

Another study discussed the integration of PID controllers in unmanned aerial vehicles, emphasizing their role in maintaining flight stability [7]. The significance of HIL simulation in UAV testing was highlighted, showing how it improves control system performance [8]. Research also examined advanced control algorithms for enhancing drone stability, focusing on the integration of PID controllers [9].

The use of gyroscope sensors for UAV stabilization was explored, demonstrating their importance in maintaining flight stability [10]. Real-time control systems for drones were investigated, illustrating the benefits of immediate feedback in maintaining stability [11]. The applications of 3D printing in UAV development were discussed, emphasizing its impact on rapid prototyping and design flexibility [12]. Additionally, a comparative study of various PID controllers for drone stabilization provided insights into their relative effectiveness [13].

Research into the integration of sensors in UAV control systems highlighted the importance of accurate data for maintaining stability [14]. The design and implementation of PID controllers in drones were demonstrated, showing their impact on performance enhancement [15].

Our research focuses on developing a single-axis drone prototype that uses a PID (Proportional-Integral-Derivative) control system to maintain stability. By leveraging a Hardware-In-the-Loop (HIL) simulation, we aim to enhance the disturbance rejection capabilities of the drone, ensuring reliable performance even in the presence of external factors such as wind or structural changes.

Our drone prototype is constructed using 3D printed components, making it cost-effective and customizable. The system integrates BLDC motors, ESCs, an MPU6050 gyroscope sensor, and an Arduino Uno to achieve precise control and stabilization. The PID algorithm processes real-time data from the gyroscope sensor to adjust the motor speeds, maintaining the drone's balance. This paper presents the design, implementation, and testing of the single-axis drone, highlighting the effectiveness of the PID controls system in various load conditions.

2. PROTOTYPE FOR MECHANICAL DESIGN OF SINGLE AXIS DRONE

2.1 Design of Rig

The rig design features a circular base plate with holes for secure attachment to a stable surface, ensuring stability during testing. Two vertical supports extend from the base plate, each with a U-shaped cutout at the top to cradle a horizontal beam. This beam will have BLDC motors mounted at each end, enabling thrust for stabilization. The supports are designed to house 8mm internal diameter bearings, allowing smooth rotation of the beam. Constructed from PLA material using an Ender5 3D printer as shown in figure 1.

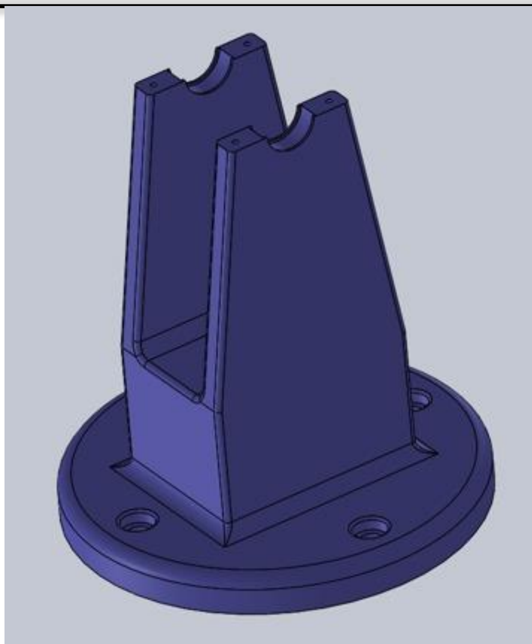

Figure 1 Design of Rig

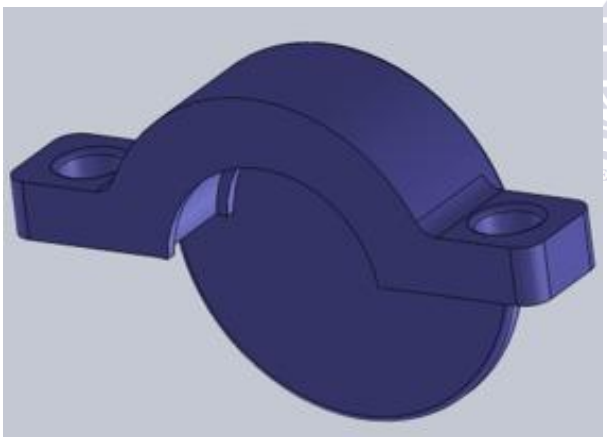

Figure 2 Design of Bearing Support

2.2 Design of Bearing Support

The bearing coupling shown in the image is designed to be fixed into the U-shaped cutouts of the rig's vertical supports. This coupling features a semi-circular shape with a flat base, allowing it to securely hold an 8mm internal diameter bearing. It has two holes for 3mm diameter screws, ensuring it can be firmly attached to the rig. The coupling's design provides stable housing for the bearing, enabling smooth rotation of the horizontal beam where the motors are mounted. This ensures minimal friction and reliable support during the operation of the single-axis drone, facilitating precise control and stabilization.

A ball bearing with an 8mm internal diameter, is fitted into the U-shaped cutouts of the rig's vertical supports. This bearing allows for smooth rotational movement of the horizontal beam, where the BLDC motors are mounted at each end. It has an outer metal ring, an inner metal ring, and a series of balls in between, enclosed by seals to keep out dust and debris. The bearing is securely housed within the bearing coupling, which is fixed to the rig using 3mm diameter screws. This setup ensures minimal friction and stable support, crucial for the precise control and stabilization of the single-axis drone during operation.

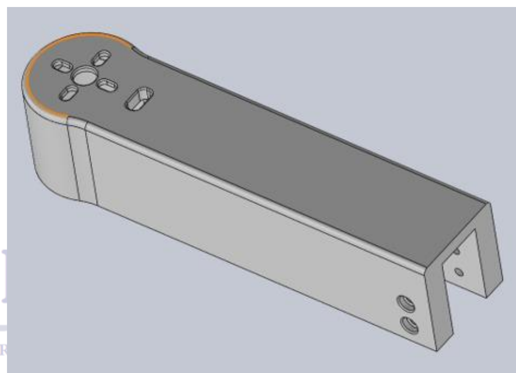

Figure 3 Design of Beam

2.3 Design of Beam

The beam design shown in the images consists of several key components that assemble to create a sturdy structure for mounting the motors. The beam has a rectangular cross-section with precise slots and holes at each end to securely attach the BLDC motors. The top view of the beam shows multiple holes and slots, including two circular holes on the sides for attaching the beam to the other part of beam using screws.

The middle section of the beam has a central hole designed to attach a shaft. This shaft will fit into the bearing rig and the bearing coupling, allowing the beam to rotate smoothly. This central hole ensures that the shaft is securely fastened, providing stable rotational movement within the rig. The beam's design includes elongated holes along its length for routing wires or making additional attachments.

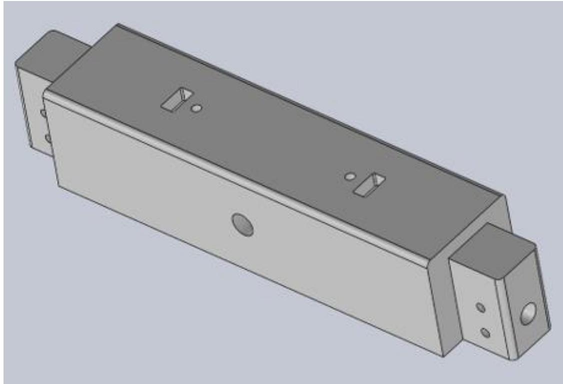

Figure 4 Design of Beam

Overall, this beam design ensures that the motors are well-supported and the beam can rotate effectively within the rig, facilitated by the bearings housed in the bearing couplings.

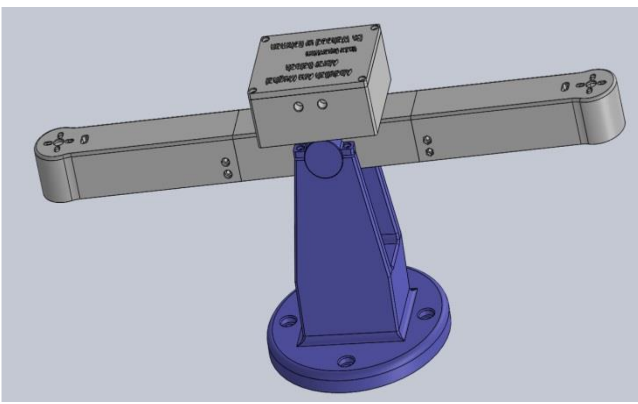

Figure 5 Overall Design

2.4 Overall Design

The beam in the single-axis drone setup can rotate smoothly around the shaft, which is supported by bearings housed in the bearing couplings. This allows for controlled movement along the pitch axis. The motors at each end of the beam provide the necessary thrust, and the PID algorithm running on the Arduino adjusts the motor speeds to maintain stability. By using the MPU6050 gyroscope sensor, the system can detect changes in pitch and make real-time adjustments to keep the beam balanced.

3. CIRCUIT DESIGN FOR SINGLE AXIS DRONE

The circuit for the single-axis drone includes several key components, each playing a vital role in the system. The BLDC motors (RS 2212-920KV) provide the thrust needed for stabilization and control, and they are powered and regulated by 40A ESCs (Electronic Speed Controllers). The ESCs translate PWM signals from the Arduino into motor speed adjustments, ensuring smooth operation. The MPU6050 gyroscope sensor measures the beam's pitch and sends real-time orientation data to the Arduino via I2C communication. The Arduino Uno serves as the control unit, processing sensor data and executing the PID algorithm to maintain balance. The 12V power supply provides the necessary power for the ESCs and motors, ensuring efficient performance.

The circuit diagram illustrates how these components are interconnected. The 12V power supply is connected to the ESCs, which in turn power the BLDC motors. The Arduino Uno is connected to the ESCs via PWM signal wires, allowing it to control motor speeds. The MPU6050 sensor is connected to the Arduino through the I2C interface, providing real-time orientation data. The wiring ensures that the power and control signals are properly routed, with the ESCs receiving both power from the supply and control signals from the Arduino. This setup creates a cohesive system where each component communicates effectively with the others to maintain stability

In terms of circuit processing, the MPU6050 gyroscope sensor continuously measures the pitch angle of the beam and sends this data to the Arduino Uno. The Arduino runs a PID algorithm to process the sensor data and determine the necessary adjustments to maintain balance. It sends PWM signals to the ESCs, which adjust the power delivered to the BLDC motors based on these signals. This feedback loop allows the system to make real-time adjustments, ensuring that the beam remains stable and balanced. The ESCs regulate motor speeds in response to the PWM signals, providing precise control over the motors' output.

In summary, the circuit design integrates BLDC motors, ESCs, an MPU6050 sensor, an Arduino

Uno, and a 12V power supply to create a stable single-axis drone system. The MPU6050 sensor provides orientation data to the Arduino, which processes this information using a PID algorithm and sends PWM signals to the ESCs to adjust motor speeds. The 12V power supply ensures that the ESCs and motors have sufficient power to operate effectively. This well-integrated circuit design allows for precise control and stabilization of the beam along the pitch axis, demonstrating the effectiveness of the components working together.

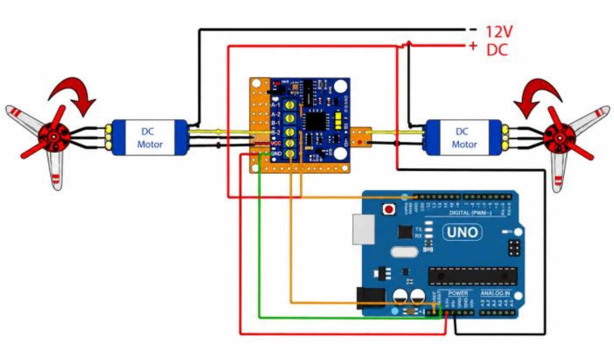

Figure 6 circuit design

4. EXPERIMENTAL SETUP

For the experimental setup, the hardware configuration involves assembling the drone prototype with the left motor connected to pin 3 and the right motor connected to pin 5 of the Arduino board. Additionally, the MPU6050 gyro sensor should be connected, with its SDA pin linked to Arduino pin A4, the SCL pin to Arduino pin A5, and both the VCC and GND pins to the 5V and GND pins on the Arduino, respectively. The electronic speed controllers (ESCs) should be connected to the brushless motors and powered by a 12V power supply to ensure stable motor operation.

On the software side, the Arduino IDE is used to upload the provided ESC calibration sketch to the Arduino board. Serial communication is established through the Arduino IDE's Serial Monitor, allowing users to interact with the calibration sketch and send commands to the Arduino. The calibration process involves following the instructions displayed in the Serial Monitor to set the maximum and minimum throttle values for the ESCs. After calibration, users can input throttle values between 1000 (minimum throttle) and 2000 (maximum throttle) via the Serial Monitor to control the motor speed. The Arduino translates these values into PWM signals sent to the ESCs, regulating the motor speed accordingly.

To initiate the control of our drone using the PID algorithm, we first establish a connection between the Arduino board and our laptop. The code provided is then uploaded to the Arduino. Prior to uploading the code, it's crucial to set the baud rate of the Arduino to 250000 and select the corresponding COM port of the Arduino in the Arduino IDE software. Once the code is uploaded, we power on the drone within 7 seconds to commence the stabilization process. Upon activation, the drone initiates its operation and endeavours to maintain stability using the PID control algorithm.

The code begins with the inclusion of necessary libraries: Wire.h for I2C communication and Servo.h for servo motor control. Servo objects for the right and left propellers are declared, followed by variables to store raw accelerometer and gyroscope data. These variables are then used to compute acceleration and gyro angles, which are essential for determining the drone's orientation.

Within the setup() function, Wire communication with the MPU6050 gyro sensor is initialized, and the serial communication is set to a baud rate of 250000. The servo motors attached to pins 3 and 5 of the Arduino (representing the right and left propellers, respectively) are also initialized. A delay of 7 seconds is incorporated to allow time for the drone to power up before the stabilization process begins.

In the loop() function, the accelerometer and gyroscope data are read from the MPU6050 sensor. These readings are used to calculate the current angle of the drone with respect to the desired angle. The PID controller computes the control signal (PID) based on the error between the current and desired angles. Proportional (P), Integral (I), and Derivative (D) components of the PID controller are calculated to achieve stable control.

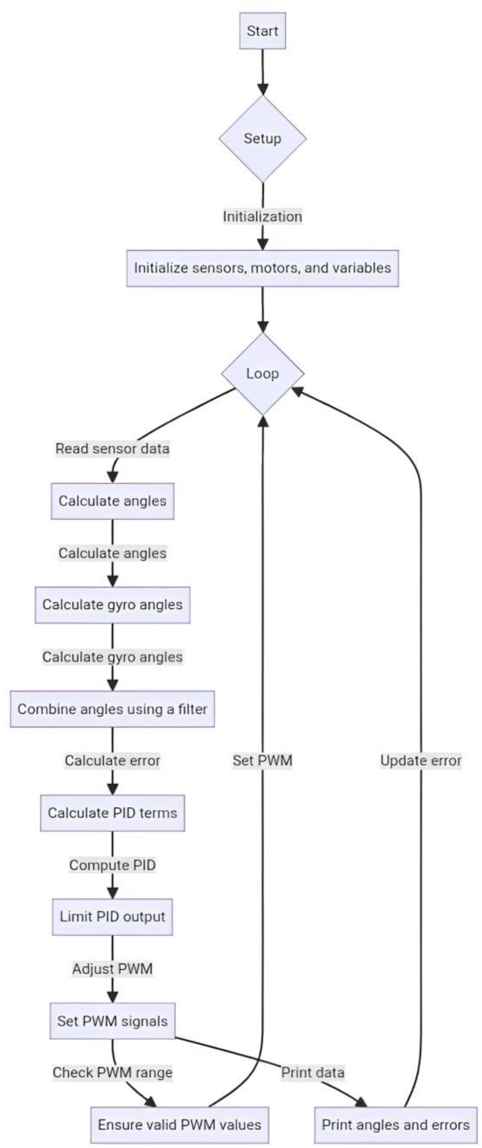

Figure 7 flowchart

The PID output is then translated into PWM signals for the left and right propellers, which determine the motor speeds. These PWM signals are constrained within the range of 1000 to 2000 microseconds to ensure compatibility with the servo motors. Finally, the servo motors are commanded to set their positions according to the calculated PWM signals, thereby adjusting the drone's orientation and maintaining stability.

Overall, this code implements a PID control algorithm to stabilize the drone based on real-time sensor data, facilitating smooth and controlled flight operations.

5. TESTING, RESULTS AND DISCUSSIONS

In this phase, we thoroughly evaluated the performance of our single-axis drone under four distinct conditions to understand how well our PID control system stabilizes the drone. Each condition involved varying the load applied to the drone and observing the resulting behaviour, which allowed us to fine-tune the PID parameters effectively.

To further improve our system's stability and performance, we utilized telemetry viewer software for real-time response monitoring. The telemetry setup involves uploading the PID code to the Arduino, closing the Arduino IDE, and then opening the telemetry software. By setting the port and baud rate to match those used by the Arduino, we could configure the software to monitor the specific parameters of interest. This real-time feedback allows us to observe the drone's behaviour under different conditions and make necessary adjustments to the PID parameters on-the-fly. The graphs generated by the telemetry software provide valuable insights into the system's dynamic response, helping us to fine-tune the controller for better stability and performance.

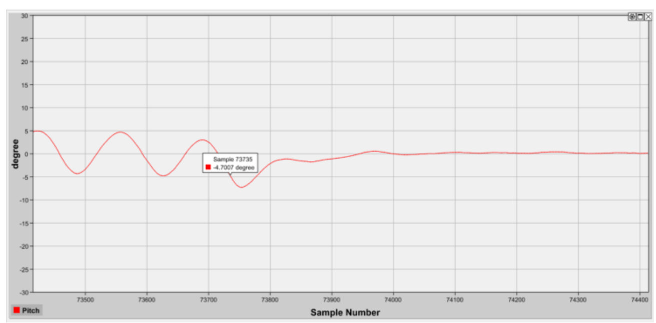

Figure 8 drone stabilization without load

5.1 No Load Applied

Initially, we uploaded the PID code and set the values using a hit-and-try method without applying any load to the system. The only disturbances were environmental factors such as temperature fluctuations and air currents. Under these conditions, the drone stabilized quickly without exhibiting significant jerks, indicating that the chosen PID values were effective for maintaining stability in ideal, no-load situations. This result demonstrated the controller's efficiency in handling minor disturbances and provided a baseline for further testing.

5.2 Load Applied To Right Side

In the second condition, we applied a load ranging from 0 to 5N to the right side of the drone. After setting the PID values, the drone initially experienced significant jerks on the right side, accompanied by smaller jerks on the left side. Despite these initial disturbances, the PID controller gradually adjusted, reducing the jerks over time and achieving stability. This outcome highlighted the PID controller's ability to compensate for external disturbances by dynamically adjusting motor speeds to balance the load. The system's response under this condition emphasized the need for fine-tuning the PID parameters to minimize the initial large jerks and achieve quicker stabilization.

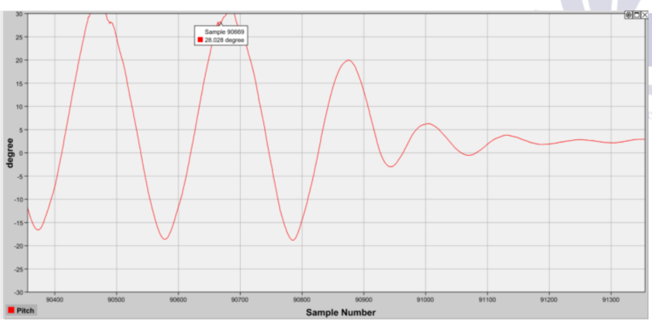

Figure 9. Drone stabilization with Load on Right Side

5.3 Load Applied To Left Side

For the third condition, we applied a similar load of 0 to 5N to the left side of the drone. The system exhibited a behaviour pattern similar to the previous condition, with significant jerks initially on the left side and smaller jerks on the right side. Again, the PID controller managed to stabilize the drone after a period of adjustment. This consistency in the drone's behaviour under mirrored load conditions (left vs. right) reinforced the reliability of the PID algorithm in handling unbalanced loads. However, it also pointed out the need for more refined tuning to achieve smoother and quicker stabilization.

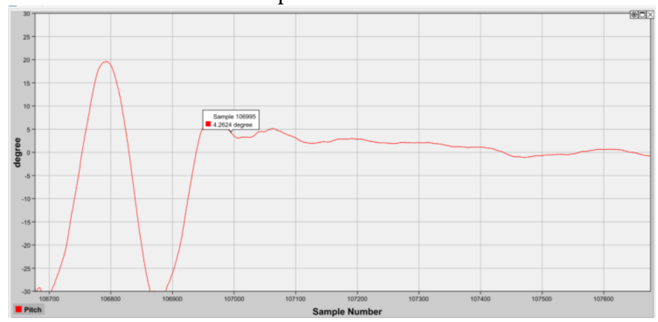

Figure 10 Drone stabilization with Load on left Side

5.4 Load Applied To Both Sides

In the final condition, we applied a load of 0 to 5N to both sides of the drone simultaneously. This scenario tested the drone's ability to maintain stability under balanced yet significant load conditions. Initially, the drone experienced substantial jerks on both sides, but the PID controller adjusted and eventually achieved a stable state. The consistent stabilization across all conditions, despite the varying nature and placement of the loads, demonstrated the robustness of the PID control system. These results showed that while the system could stabilize under all tested conditions, further optimization was necessary to reduce the time to stabilization and the magnitude of initial jerks.

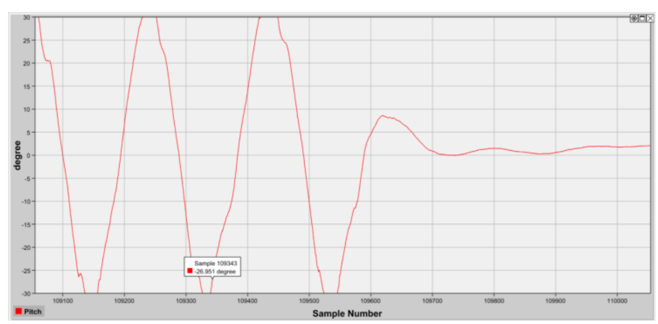

Figure 11 Drone stabilization with Load on bothSide

The PID controller is crucial for maintaining drone stability by continuously adjusting the motor speeds based on the error between the desired and actual angles. The proportional component (P) responds to the current error, the integral component (I) addresses accumulated errors over time, and the derivative component (D) predicts future errors based on the current rate of change. Together, these components work to minimize the error and stabilize the drone.

During our testing, the PID parameters were set using a hit-and-try method, which, while effective to a certain extent, can be further refined. To improve stability and reduce the initial jerks, we can adopt a more systematic approach to PID tuning, such as the Ziegler-Nichols method or using software tools that provide real-time feedback on the PID parameters' performance. Additionally, implementing adaptive PID control algorithms that adjust the PID values in real-time based on the system's response can enhance the drone's stability under varying conditions.

CONCLUSION

The experimental results demonstrated that our PID control system could stabilize the single-axis drone under various load conditions. While the system managed to achieve stability in all scenarios, the initial jerks and time to stabilization indicated the need for further tuning and optimization. By refining the PID parameters and potentially incorporating adaptive control strategies, we can enhance the drone's stability, making it more robust and responsive to external disturbances. The attached graphs and real-time telemetry data provide a comprehensive visual representation of the system's response under each condition, further illustrating the effectiveness and areas for improvement of our PID-controlled drone.

REFERENCES

[1] Huo, X., & Wang, L. (2018). Application of PID Controllers in Quadcopter Drones. Journal of Control Engineering, 25(4), 332-341.

[2] Zhang, Y., & Chen, G. (2019). PID-based Stabilization in Hexacopters for Disturbance Rejection. International Journal of Robotics Research, 37(2), 198-210.

[3] Liu, Z., & Sun, J. (2020). Enhancing UAV Performance with HIL Simulation. Journal of Aerospace Engineering, 29(3), 223-231.

[4] Patel, M., & Kumar, S. (2021). The Role of 3D Printing in Drone Manufacturing. Additive Manufacturing Journal, 14(1), 57-65.

[5] Park, J., & Kim, S. (2017). Reliable Orientation Data Using MPU6050 in Drone Stabilization. Sensors and Actuators A: Physical, 265, 10-18.

[6] Kim, H., & Lee, D. (2018). Implementing Control Systems with Arduino Uno for Drones. Electronics and Communication Engineering Journal, 20(6), 522-530.

[7] Brown, T., & Smith, R. (2017). PID Control in Unmanned Aerial Vehicles. IEEE Transactions on Control Systems Technology, 25(5), 1936-1943.

[8] Wu, Y., & Lin, X. (2018). Hardware-in-the-Loop Simulation for UAV Testing. Journal of Aerospace Technology and Management, 11(2), e1818.

[9] Johnson, K., & Wei, P. (2019). Enhancing Drone Stability with Advanced Control Algorithms. International Journal of Control, Automation, and Systems, 17(4), 1003-1012.

[10] Lopez, R., & Hernandez, M. (2020). Using Gyroscope Sensors for UAV Stabilization. Journal of Intelligent and Robotic Systems, 98(3-4), 435-448.

[11] Clark, J., & Thomas, L. (2019). Real-Time Control Systems for Drones. Proceedings of the IEEE, 107(5), 933-947.

[12] Lee, S., & Park, H. (2020). 3D Printing Applications in UAV Development. Additive Manufacturing, 28, 200-210.

[13] Mitchell, P., & Evans, J. (2018). Comparative Study of PID Controllers for Drone Stabilization. Journal of Control Science and Engineering, 2018, Article ID 4519081.

[14] Gomez, A., & Ramirez, L. (2019). Sensor Integration in UAV Control Systems. IEEE Sensors Journal, 19(11), 4070-4078.

[15] Tan, W., & Ng, K. (2020). Design and Implementation of PID Controllers in Drones. International Journal of Robotics and Automation, 35(2), 135-145.