Designing a Single-Degree-of-Freedom Aerodynamic Pendulum for Testing Algorithms and Mechanisms for Control of Multicopter Motors

Martin Voykov, Milad Kamzan

Abstract— **This paper presents a real-time roll stabilization control system that employs actuation system consisting of a brushless motor with a mounted propeller and a dedicated electronic speed controller (ESC) and an inertial measurement unit (IMU) integrated with an ESP32 microcontroller platform. The system utilizes a low- pass filtering for obtaining accurate gyroscope data, a complementary filter for sensor data fusion, a proportional-integral-derivative (PID) control algorithm with parameter tuning in real-time as well as telemetry readings for monitoring performance and crucial data such as revolutions per minute (RPM). One test was conducted throughout the process which exploits the ESP32 microcontroller as the main platform for collecting and monitoring data as well as controlling the system. The results of the experiment are satisfactory due to demonstrated good roll angle stabilization, done with a relatively quick response time, which allowed us to continue to the next phase of the experiment, where MATLAB would be used for computation and control while the ESP32 would be utilized as a pure data bridge. When all of the work is done and backed with acceptable test results this stand can be used for applications including but not limited to teaching and integrating artificial intelligence (AI) agents for autopilots.**

Keywords—**Propeller-Driven Pendulum System (PDBS), UAV, Proportional-Integral-Derivative (PID) controller**

I. Introduction

Single-degree-of-freedom (SDOF) stands are commonly used both to educate students in various subjects including, but not limited to dynamics and control systems, and to test and implement new methods for controls being developed, especially in unmanned aerial vehicles (UAVs) designs. This is valid due to that fact that they offer not only simple and relatively affordable solutions and parts, but also because they can be transferred into a real- world system, such as a quadcopter, in a timely manner, without much effort or reconfigurations.

One of the most widely used single-degree-of-freedom platform is the so-called single-motor aerodynamic pendulum. It employs one brushless direct current (BLDC) motor, located at the end of a rod, which moves it around a pivot point by utilizing the aerodynamic lift force, generated by the rotating propellers. In order to achieve balance around the rotational point the stand makes use of a rotary encoder, located there. In terms of software used, MATLAB/Simulink and Arduino Integrated Development Environment IDE are utilized to process the gathered by a microcontroller (ESP32, Arduino) data as well as to control the whole system i.e., by regulating the revolutions per minute (RPM) of the BLDC thus increasing or decreasing the lift force based on the angle feedback received. Those kinds of stands have been proven to be effective in utilizing classic control theory algorithms such as Proportional-Derivative-Integrated (PID) controllers [1, 2, 3, 4]. However, using encoder limits the potential of transferring the results to the UAVs used due to the fact that their size is hardly suitable for implementation.

The slightly more advanced version, namely dual-motor aerodynamic pendulum introduces two counter-rotating BLDCs equidistantly situated from the pivot point on the arm. This configuration allows the users to test bicopter arrangement as well as to evaluate more advanced control algorithms such as Linear-quadratic regulator (LQR) and Linear-quadratic-integral (LQI) controller, but also to emulate multicopter behavior and disturbances around one of its principal axes [5]. These types of stands implement inertial measurement unit (IMU) consisting of a gyroscope, accelerometer and sometimes magnetometer due to the fact that this is widely used in the UAVs nowadays. Apart from that the configuration and elements are largely similar in comparison with the single-motor stand.

Having said that, the main goal of this paper is to develop and test a functioning stand for single-degree-of-freedom of UAV testing with the option to advance it uses to not only multiple DOF, but to also be used as an evaluation and teaching setup for artificial intelligence (AI) models.

II. Overview of the Test Stand

The main part of the configuration of the aerodynamic pendulum stand described in this paper consist of a BLDC motor with a 10” (254 mm) propeller, mounted to one of the ends of a suspended aluminum arm, which is pivoted at 1/3 of its length with no additional counterweight attached on the other end. The free movement of the arm is limited to a range of -35° to +35° relative to the horizontal position. A

Received: 01.07.2026
Published: 26.07.2026
https://doi.org/10.47978/TUS.2026.76.03.005
Martin Voykov is with the Technical University of Sofia, Sofia, 1000, 8 Kl. Ohridski Blvd, Bulgaria (mvojkov@tu-sofia.bg)

Milad Kamzan is with the K. N. Toosi University of Technology, Tehran, Iran (milad.kamzan@gmail.com)

dedicated electronic speed controller (ESC) is used to control the speed and rotation direction of the motor and in order not to experience a limited time of operation the driving system does not use a lithium-polymer battery, but a power unit (PU), which allows the user to change the voltage output depending on the type of motor and ESC used. In addition to that, a 6-DOF IMU, with a 3-DOF gyroscope and a 3-DOF accelerometer, is located exactly at the pivot point of the arm fixed with a 3D printed bedding especially designed for this purpose. Both the ESC and the IMU are connected to an ESP32 microcontroller which controls and acquires data from each of them respectfully. It was decided to use ESP32 instead of a cheaper Arduino board, not only because the former offers a compact, but yet more powerful design with clock speed of over 10 times better, but also due to its ability to read inverted signals, which is essential, because the ESC is able to provide telemetry data (RPM etc.), but only through communication protocols which work with such inverted signal protocol such as SBUS2. Additionally, the system utilizes an ESP supported liquid crystal display (LCD) with the ability to show 20 characters per row of information with 4 rows available. While the system operates the LCD screen is used to monitor its current status, throttle of the BLDC and other crucial data. When a PID controller is tested there are 3 analog potentiometers used in order to change each gain accordingly and their values are updated dynamically on the screen. A full list of the used elements is listed in Table I.

TABLE I
COMPONENTS OF THE STAND

Component	Specification	Connection
BLDC Motor	BLF 420/1400	Powered by the PU via ESC and controlled by ESP32 via ESC
Propeller	10” (254 mm)	Mounted to the BLDC
ESC	IBEX 65HV	Powered by the PU and connected to the ESP32 for control and telemetry
Microcontroller	ESP32-DEVKITC-32U	Connected to the PC through USB port
IMU	MPU6050 6-DOF	Connected to the ESP32 via I2C protocol
LCD	LCD 20x4 I2C	Connected to the ESP32 via I2C protocol
Potentiometers	10 kΩ analog potentiometer (3 EA)	Connected to the ESP32 via analog pins
PU	Standard stand PU with output voltage of up to 30V	Connected to the grid

The overall configuration of the test aerodynamic pendulum is shown on Fig. 1 and the electrical wiring scheme is illustrated on Fig. 2.

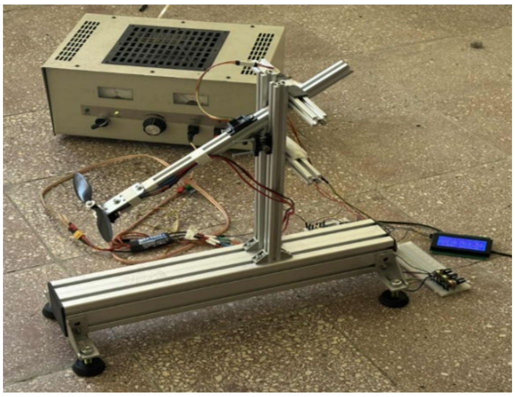

Fig. 1. Test stand configuration

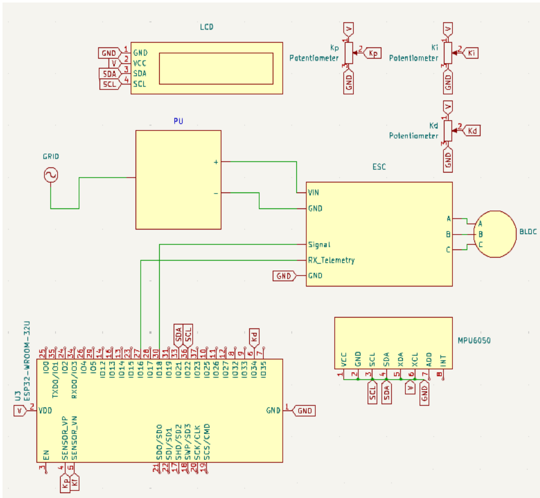

Fig. 2. Wiring scheme of the stand

III. TEST: PID CONTROLLER WITH ESP32

In this experiment, the ESP32-based system with integrated PID controller was used to estimate the roll angle and to stabilize the aerodynamic pendulum arm in horizontal position. The aim of the test was to ensure that a classical control systems algorithm can be implemented to complete the set task of the stand.

In the following part of the paper there will be a detailed overview of each step of the control process.

A. Static sensor calibration

Before the testing of the systems starts there is a calibration period of 5000 samples for the IMU. In order to take into account, the disturbances caused by the motor, during the calibration the BLDC is running at 25% throttle, which was enough to keep the arm stationary. This process determines the accelerometer bias and the gyroscope bias using equations (1) and (2):

$$AccErrorX = \frac{1}{N} \times \sum\left(\operatorname{atan2}\left(Ay, \sqrt[2]{Ax^2 + Az^2}\right)\right) \times \frac{180}{\pi} \quad (1)$$

$$GyroErrorX = \frac{1}{N} \times \sum\left(\frac{\omega x_{raw}}{131.0}\right), \quad (2)$$

where N is the number of calibration samples, Ay, Ax, Az are raw values from the accelerometer readings from each axis, ωx_{raw} is the raw values of the gyroscope readings only from the desired axis and 131.0 is the gyroscope sensitivity factor for ±250°/s sensitivity range.

B. Processing accelerometer data

After successfully calibrating the sensors, we can compute the current roll angle from the accelerometer by subtracting AccErrorX from the current angle value using equation (3):

$$AccAngleX = \text{atan2}\left(Ay, \sqrt[2]{Ax^2 + Az^2}\right) \times \frac{180}{\pi} - AccErrorX, \quad (3)$$

where Ay, Ax, Az are raw values from the accelerometer readings from each axis and AccErrorX is the bias offset.

C. Processing gyroscope data with low-pass filter

The first step of the process of getting the roll angle from the gyroscope readings is to dynamically calculate it with taking into consideration the previously computed bias. In order to do so we are subtracting the error received by equation (2) from the current readings using equation (4):

$$GyroAngleX = \frac{\omega x_{raw}}{131.0} - GyroErrorX\,, \quad (4)$$

where ωx_{raw} is the raw values of the gyroscope readings only from the desired axis, 131.0 is the gyroscope sensitivity factor for ±250°/s sensitivity range and GyroErrorX is the previously calculated bias offset.

So as to limit the appearing high-frequency noise in the measurements of the gyroscope, a low-pass filter was implemented which has a coefficient β = 0.9. The filtered values are computed adding the current raw values and the previous ones weighted by the coefficient as shown in equation (5):

$$GyroAngleX_{filtered}\,(t) = \beta \times GyroAngleX_{filtered}\,(t-1) + (1-\beta) \times GyroAngleX, \quad (5)$$

where β is the low-pass filter coefficient, $GyroAngleX_{filtered}\,(t-1)$ is the previous reading and GyroAngleX is the original gyroscope reading.

By adding the low-pass filter we significantly improve the results of the readings of the gyroscope.

D. Roll angle value calculation

After the accelerometer and gyroscope's data has been collected the system need to fuse them and to compute the roll angle. In order to do so a complementary filter has been applied to the system so that it could leverage the characteristics of both sensors. All of that is represented in equation (6):

$$\Phi(t) = \alpha\left[\Phi(t-1) + GyroAngleX_{filtered}\,(t) * \Delta t\right] + (1-\alpha) * AccAngleX(t), \quad (6)$$

where α is the complementary filter coefficient which equals to 0.96, Φ(t) is the roll angle at time t, $GyroAngleX_{filtered}\,(t)$ is the filtered values of the gyroscope, Δt is the elapsed time which is used to numerically integrate the gyroscope data in order to get roll angle changes from angular velocity and AccAngleX is current reading from the accelerometer.

The usage of the complementary filter ensures that the system gets high-frequency accuracy data from the gyroscope while simultaneously the low-frequency stability is gained from the accelerometer data.

E. PID controller implementation

As mentioned before a PID controller implemented in order to test the capabilities of the aerodynamic pendulum and that has been done as presented in equation (7):

$$PIDout(t) = Kp \times error(t) + Ki \times \int error(\tau)d\tau + Kd \times \frac{derror(t)}{dt}\,, \quad (7)$$

where PIDout is the control output, error(t) represents the error signal and is calculated by subtracting the current roll angle from the desired one and Kp, Ki and Kd are the PID controller gains respectfully.

After that the PID output is added to the throttle command using equation (8):

$$Throttle = Throttle_{base} + PIDout\,, \quad (8)$$

where $Throttle_{base}$ is a predetermined nominal operation throttle value (27%) and PIDout is the PID controller output.

In addition, the Kp, Ki and Kd gains can be tuned in real time by using potentiometers and their values are dynamically updated on the dedicated LCD screen that has been implemented to the system. This ensures that optimization of the controller parameters can be done during operation in real-time.

F. Integral windup protection

One of the biggest issues with the usage of a PID controller is the so-called integral windup which occurs when the I part of the algorithm collects large values and that results in the controller becoming unstable and hardly responsive. In order to prevent it we have implemented an adaptive limit which constrains the integral term within certain limits using equations (9) and (10) so that the system can maintain stability over time.

$$ki_{limit} = \pm\frac{25}{ki_{initial}}\,, \quad (9)$$

$$ki_{final} = ki_{initial} \times ki_{limit}\,, \quad (10)$$

where ki_{limit} is the set limit for the integral gain, $ki_{initial}$ is the initial Ki value set at the beginning and ki_{final} is the maximum value of Ki that can be reached.

G. Safety measures

Safety is a crucial part of operation of a system and it should be one of the main priorities taken into consideration while designing such a stand. Our system includes several safety features such as angular limit of 45° which stops the whole system immediately if reached. Although the aerodynamic pendulum's arm is suspended from movement as mentioned before this feature would be very useful should the suspension fails. Another safety mechanism that the stand employs is the throttle constraints limit which limits the throttle range from 1 to 45% as it was determined that the thrust produced by bigger throttle percentages is not useful, nor it is safe to load the BLDC that much. One of the most important features is the emergency stop, which when activated stops the whole system at once. Combining all of the above features ensures that the people using the aerodynamic pendulum are working under safe conditions and could do their work without worrying about it.

H. Results

This section of the paper presents the results of the practical experiments conducted with the stand. During the experimental phase many possible PID values combinations were tested and evaluated and some of them would be discussed below.

The first test was conducted utilizing PID controller with the following values Kp = 0.146, Ki = 0.020 and Kd = 0.170. As can be seen from the graph plotted from the raw data on Fig. 3, we have an initial overshoot of over 50% as the roll angle reaches about 85°. This is followed by a return to the starting position and then a 9 second rise to the setpoint of 35° after which the system is relatively stable with 5% error about it. Although the stability is present the massive overshoot and the slow speed of about 27 seconds before stabilization, we decided to pursue better results.

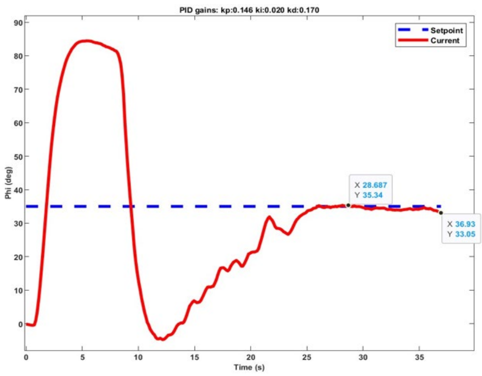

Fig. 3. Test 1 results

After decreasing the Kp value with 0.01, and increasing the Ki to 0.036 while not changing Kd value at all we managed to reduce the time to stabilization more than 2 times while maintain the error at 5% and reducing the overshoot to only 43° as seen in Fig. 4.

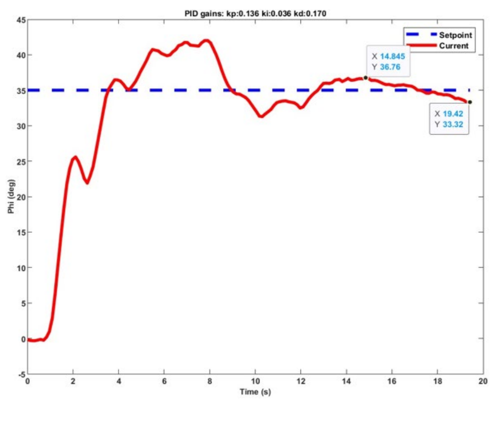

Fig. 4. Test 2 results

The results of the second test gave are satisfactory and showed that the stand has a massive room for improvement which allowed us to continue to the next phase of the project.

IV. Future Work

As discussed previously the end goal of the project is to create and implement an algorithm, which could be used in real world systems. In order to do so the following phases have been established:

1. Create a SDOF stand and test it with PID using Arduino IDE platform;
2. Migrate to MATLAB;
3. Implement PID control in MATLAB;
4. Design, test and implement neuron networks or reinforcement learning to the system;
5. Design a multi-degrees-of-freedom stand;
6. Implement to a real-world system such as an UAV.

A. Current Phase: Migration to MATLAB

At the current stage we are working on migration to MATLAB in order to control the system and monitor it in the program itself while using the ESP32 as a data bridge between the components of the stand. Using MATLAB would not only allow us to control the system in real time, but also would offer the usage of real-time graphs, which has numerous advantages. At the moment the ESP32 has been successfully repurposed as a data bridge and a graphical user interface (GUI) has been created and can be seen on Fig. 5. The GUI not only gives us the opportunity to change the PID gains as well as the target angle values in real time and stop the system via the dedicated buttons, but also provides crucial information such as the throttle and PID output percentage. In addition to that, it has a communication status area in which the user can see the data rate of the system as well as its runtime. It includes five graphs showing the current roll angle versus the target one, the PID controller output, motor throttle command, motor speed measured in RPMs and the communication rate. Having all that information in a GUI gives us the advantage to monitor many aspects of the system performance in real-time and to make timely decisions in order to get the best results.

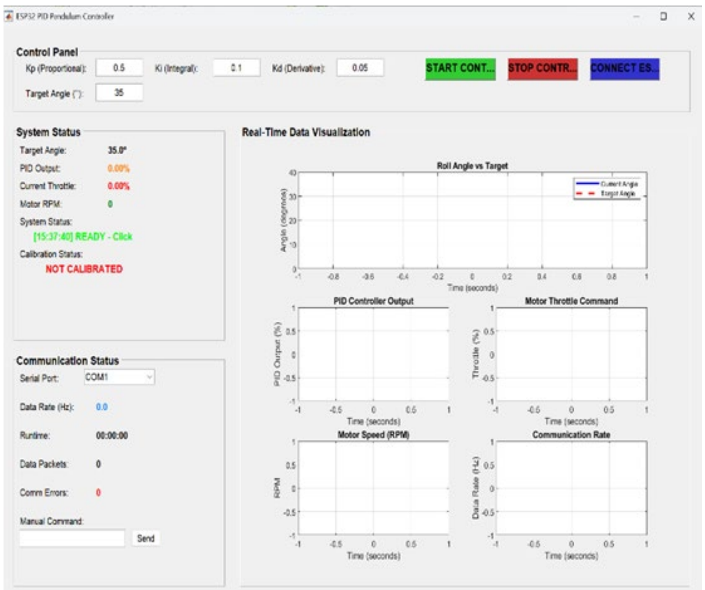

Fig. 5. MATLAB graphical user interface (GUI)

V. Conclusion

This paper presents the work that has been done in order to design and create a single-degree-of-freedom stand for implementing and testing algorithms and mechanisms for control of multicopter's motors. With the purpose of doing that several devices and sensors, such as electronic speed controller, an inertial measurement unit, an ESP32 microcontroller etc., were used. At the current stage of the project the system employs low-pass filtering and sensor data fusion as well as a proportional-integral-derivative (PID) control algorithm for using the raw data gathered in order to control the system.

During the testing phase of this work the PID controller was utilized with different gains providing us with crucial information of the areas of which the stand can be improved. The results of those were satisfactory and showed that the stand could be used with more advanced algorithms and mechanisms.

For the future, the next phases of the project will involve migrating to MATLAB, which is something that is currently being worked on, designing, testing and implementing neuron networks or reinforcement learning to the system, designing a multiple-degrees-of-freedom stand and finally implementing all of the above into a real-world system.

In addition to that, the current stand can also be used for educational purposes with undergraduate students, who are studying control theory and flight dynamics.

Acknowledgment

This work was supported by Technical University of Sofia research project № 253CH0001-04 "Building infrastructure and environment for aerospace training and research at Technical University of Sofia".

The authors would like to thank Assoc. Professor Vladimir Serbezov – Head of the Department of Aeronautics at Technical University – Sofia for supporting the work on this project.

References

[1] Rojas-Galván, R.; García-Martínez, J.R.; Cruz-Miguel, E.E.; Barra-Vázquez, O.A.; Olmedo-García, L.F.; Rodríguez-Reséndiz, J. Hunting Search Algorithm-Based Adaptive Fuzzy Tracking Controller for an Aero-Pendulum. Technologies 2024, 12, 63. https://doi.org/10.3390/technologies12050063

[2] Hamoudi, A.K., Rasheed, L.T. (2023). Design and implementation of adaptive backstepping control for position control of propeller-driven pendulum system. *Journal Européen des Systèmes Automatisés*, Vol. 56, No. 2, pp. 281-289. https://doi.org/10.18280/jesa.560213

[3] Mechatronic Aeropendulum: Demonstration of Linear and Nonlinear Feedback Control Principles With MATLAB/Simulink Real-Time Windows Target Eniko T. Enikov, Member, IEEE, and Giampiero Campa, Member, IEEE

[4] Marashian, Arash. (2021). Modeling and Control of Mechatronic Aeropendulum. 10.13140/RG.2.2.18357.40161.

[5] Rivera-Ruiz, J. R., García-Martínez, J. R., Martínez-Sánchez, T., Cruz-Miguel, E. E., Ramírez-González, L. D., Barra-Vázquez, O. A., & Odry, Á. (2025). Takagi–Sugeno–Kang Fuzzy Inference Tracking Controller for UAV Bicopter. Symmetry, 17(5), 759. https://doi.org/10.3390/sym17050759