

A Hybrid Fuzzy-PID Control Strategy for Unmanned Aerial Vehicle Micro Gas Turbine Speed Regulation: Design, Simulation, and Microcontroller Implementation



Athraa Allak¹, Ameer Abed Jaddoa^{1*}

College of Electromechanical Engineering, University of Technology, Baghdad 00964, Iraq

Corresponding Author Email: ameer.a.jaddoa@uotechnology.edu.iq

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.590410>

ABSTRACT

Received: 2 November 2025

Revised: 12 April 2026

Accepted: 23 April 2026

Available online: 30 April 2026

Keywords:

unmanned aerial vehicle, micro gas turbine, proportional-integral-derivative control, Arduino, microcontroller, RPM regulation, Electronic Speed Controller, Hall

This paper describes the design and experimental implementation of a hybrid control strategy that integrates fuzzy logic with a proportional-integral-derivative (PID) controller to regulate the rotational speed of a small gas turbine engine intended for unmanned aerial vehicles (UAVs). The main objective of this work is to enhance the dynamic behavior and robustness of the system to variations of the operating conditions by combining a fuzzy logic-based decision-making mechanism with a standard PID controller. The engine dynamics are represented accurately by a nonlinear mathematical model of the small gas turbine, which is derived and validated using MATLAB/Simulink simulation. The performance of the suggested hybrid controller was compared with the conventional PID controller based on the critical performance metrics such as climb time, overshoot, steady state error, and turbulence rejection. Also, the control algorithm was implemented on an embedded microcontroller platform, and real-time experimental tests were conducted under different load conditions to evaluate its practical feasibility. The results show that the Fuzzy-PID hybrid controller provides a faster transient response, less overshoot, and better robustness to disturbances, compared to the traditional PID approach confirming its effectiveness and suitability for real-time integrated drone propulsion control systems.

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) are growing in importance as assets for both military and civilian applications because of their versatility, operational efficiency, and the reduced risk to human operators. A key aspect of UAV operation is the precise control of the propulsion systems, in particular micro gas turbines, which have high power-to-weight ratios, suitable for flight dynamics. It is important to be able to control the rotational speed of such turbines to keep the flight stable, maneuverable, and performing well.

The proportional-integral-derivative (PID) control is a traditional control technique that is widely used in industry because of its simplicity and effectiveness. However, PID controllers are often known to deteriorate in nonlinear systems, or when the system parameters change over time, as it is often the case in UAV turbine dynamics. On the other hand, fuzzy logic controllers (FLCs) offer a rule-based decision mechanism that is capable of dealing with uncertainty and nonlinearities without explicit mathematical model. However, FLCs, while good at dealing with imprecise information, may not have the responsiveness and stability of PID controllers in precisely tuned applications.

In this paper, a hybrid control strategy is proposed to combine the advantages of PID and FLCs for controlling the speed of a micro gas turbine in UAVs. The hybrid Fuzzy-PID approach intends to improve system response, robustness and adaptability in real world situations. The controller's

effectiveness is assessed by simulation and hardware implementation. The main contributions of this paper are as follows:

- Development of a mathematical model of a micro gas turbine system.
- Development of a hybrid Fuzzy-PID controller adapted to UAV engine dynamics.
- Comparative performance analysis with conventional PID controllers.
- Real-time implementation on embedded microcontroller with practical validation.

The current paper is structured as follows: Sections 1 and 2 present the introduction and previous research. Section 3 presents the micro gas turbine model. Sections 4 and 5 follow, showcasing the controller design and hardware assembly. The system architecture and sensor hardware are presented in Sections 6 and 7. Section 8 covers control strategies. Section 9 provides a comprehensive overview of Arduino code. The simulation model is presented in Section 10. Finally, sections 11 and 12 present the final results, their discussion, and the concluding conclusion of the current work.

2. LITERATURE REVIEW

Several studies have explored control methodologies for UAV propulsion systems, including conventional and intelligent control strategies. PID controllers are prevalent due

to their ease of implementation, but limitations in handling dynamic nonlinear systems have motivated the development of alternative approaches. For instance, Vijayan et al. [1] investigated a PID-based scheme for UAV thrust control, highlighting sensitivity to parameter variations and external disturbances.

Fuzzy logic control has been employed in various aerospace applications, including engine regulation and flight control. Navarro Lafuente [2] developed a fuzzy controller for jet engine speed stabilization, showing improvements in robustness against parameter uncertainties. Nonetheless, standalone fuzzy controllers often require extensive tuning and lack guaranteed stability under all operating conditions.

Hybrid control approaches combining fuzzy logic and PID control have gained traction due to their potential to balance adaptability and precision. Purdum [3] implemented a Fuzzy-PID controller for quadrotor UAV attitude stabilization, achieving smoother responses and better transient behavior compared to conventional methods [4].

Despite these advancements, few studies have implemented and validated such hybrid control systems on microcontroller platforms tailored to micro gas turbines in UAVs [5]. This paper addresses this gap by developing and validating a Fuzzy-PID controller both in simulation and on actual embedded hardware, offering a practical contribution to UAV propulsion control.

3. MODELING OF THE MICRO GAS TURBINE SYSTEM

The micro gas turbine system studied in this work is composed of a compressor, combustion chamber and turbine coupled on a single shaft. The system dynamics can be approximated by a simplified single input single output (SISO) model, where the input is the fuel flow rate and the output is the shaft speed [6].

The governing equations are derived based on the principles of thermodynamics and fluid flow [7]. The main dynamics are represented by a first order transfer function:

$$G(s) = \frac{k}{\tau s + 1} \quad (1)$$

where,

K is the steady-state gain,

τ is the time constant (including thermal delay and inertia).

Nonlinearities like fuel-air mixing delay, temperature lag and actuator saturation are modeled in MATLAB/Simulink. Random disturbances and load torque variations are also included to test the robustness of the controller.

The model is validated against manufacturer data and typical micro turbine specifications and the obtained simulation model is used for controller design and performance evaluation [8].

4. DESIGN OF THE FUZZY-PID CONTROLLER

The Fuzzy-PID controller is designed by integrating a fuzzy logic inference system that dynamically adjusts the PID gains based on real-time error and change in error. The controller structure includes [9]:

$$\text{An error signal: } e(t) = r(t) - y(t) \quad (2)$$

The fuzzy system receives $e(t)$ and $\Delta e(t)$ as inputs and outputs correction factors for $K_p, K_i,$ and K_d . A Mamdani-type fuzzy inference system is used with triangular membership functions and a rule base constructed from expert knowledge and empirical tuning.

Typical fuzzy rules include:

IF error is large positive and change in error is positive THEN decrease K_p , increase k_d , and keep k_i constant.

IF error is zero and change in error is zero THEN maintain current gains.

The outputs of the fuzzy logic system are fed into the PID block, which then computes the control signal [10]:

$$u(t) = K_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad (3)$$

The controller is first tested in MATLAB/Simulink to ensure stability, responsiveness, and robustness to noise and disturbances. Tuned parameters are later ported to the embedded platform.

5. HARDWARE IMPLEMENTATION AND EMBEDDED SYSTEM DESIGN

To validate the proposed Fuzzy-PID controller in a real-time environment, the control algorithm was deployed onto a microcontroller platform. The Arduino Mega 2560 board was used as it can be used for real-time control applications and has enough memory resources, and easy to interface with peripheral devices such as sensors and actuators.

The actual speed feedback is received by the microcontroller from a Hall-effect sensor mounted on the turbine shaft [11]. It calculates the speed error by comparing the measured speed and the reference value, and then calculates the control signal by using the fuzzy-tuned PID logic. This control signal is used to drive an electronic fuel valve via PWM (Pulse Width Modulated) [12].

Hardware architecture comprises of the following components:

- Microcontroller: Arduino Mega 2560
- Sensor: Hall-effect rotational speed sensor
- Actuator: PWM-controlled electronic fuel valve
- Power Supply: Regulated 12 V DC source
- PC Interface: USB serial communication for real time data logging and tuning

The embedded controller is tested under different operating conditions, such as sudden speed change and external disturbances imposed on the turbine shaft. The controller successfully maintained stability of the engine and a fast recovery, validating the robustness of the fuzzy-PID approach for real-time deployment.

6. SYSTEM ARCHITECTURE

In this section, the map of the current study is presented as a block diagram in Figure 1 below. It is used in sequence.

The proposed system comprises an Arduino microcontroller, an Electronic Speed Controller (ESC), a brushless DC micro gas turbine motor, a Hall-effect RPM sensor, and a power supply. The Arduino reads the RPM

signal, calculates the control output using a PID algorithm, and generates a PWM signal to adjust the motor speed via the ESC

[13].

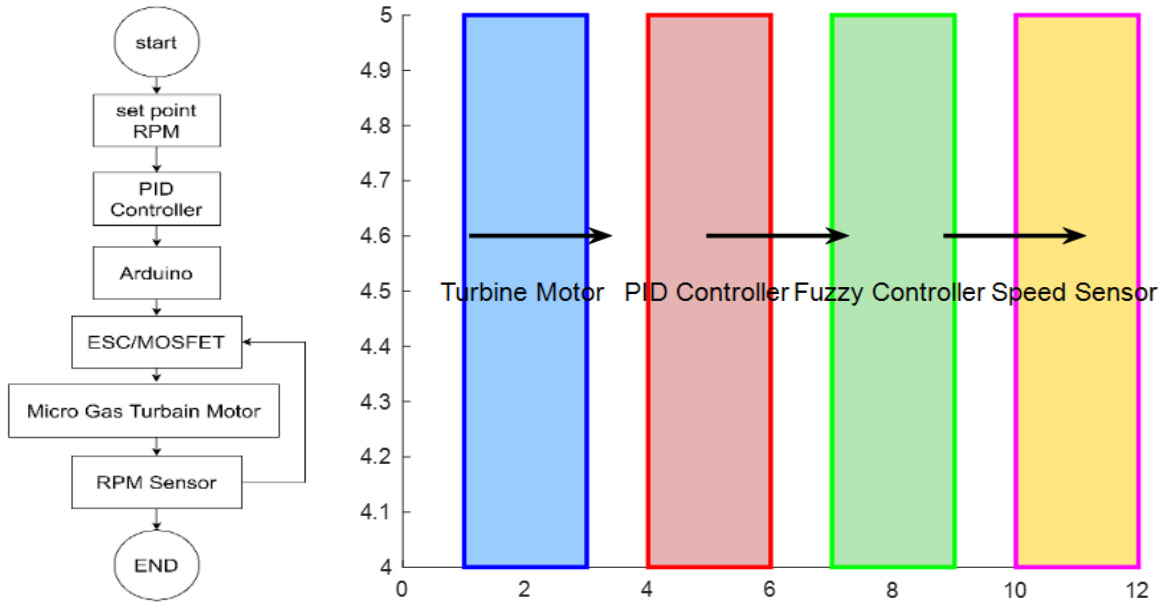


Figure 1. Current study block diagram procedures

Figure 2 presents the wiring diagram for the components used in this paper.

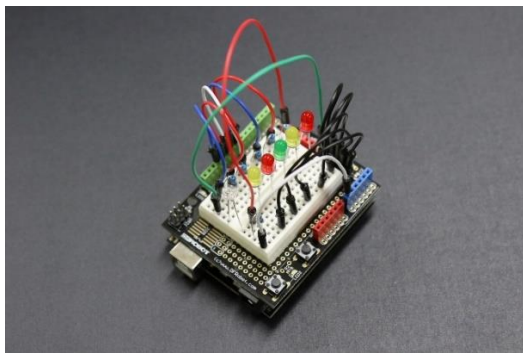


Figure 2. System architecture of the embedded Arduino control unit for unmanned aerial vehicle (UAV) micro gas turbine engine regulation

Pin2 and 3 are external ISR pins that are triggered by the rotary encoder pulses. Pin6 is the PWM pin to control the amount of voltage sent to the motor driver. Pin7 and 8 are used to change motor direction by changing the polarity of the supply voltage to the motor. A common ground between the motor driver, Arduino and external power supply.

Designing a MATLAB script simulation model for PID control of a UAV micro gas turbine engine speed and generating illustrative performance graphs is an excellent way to validate and visualize the control strategy. All steps followed as:

Step 1: Modeling the Micro Gas Turbine Engine Speed Dynamics

For control design, we often approximate the engine speed dynamics by a simplified transfer function, as presented in:

$$\frac{\omega(s)}{u(s)} = G(s) = \frac{K}{\tau s + 1} \quad (4)$$

where,

- K: steady-state gain (rpm per control input unit)
- τ : time constant representing turbine inertia and fuel flow delay
- $\omega(s)$: rotor speed
- $u(s)$: fuel input

The hybrid Fuzzy-PID controller was implemented on an embedded microcontroller platform with a control loop frequency of 500 Hz (sampling period of 2 ms). This high-frequency loop ensures timely processing of sensor feedback and rapid adjustment of PID gains through the fuzzy supervisor, which is critical for regulating the fast dynamics of the micro gas turbine. Tests with lower control frequencies initially showed increased overshoot and delayed transient response, emphasizing the importance of a sufficiently high loop rate. The selected frequency also closely matches the sampling rate used in MATLAB/Simulink simulations, ensuring consistency between simulation results and real-time performance and validating the predictive accuracy of the simulation model under real operating conditions. As shown in Figure 3, which presents a control loop timing diagram of the hybrid Fuzzy-PID controller implemented on the embedded microcontroller, showing sensor reading, Fuzzy-PID computation, and actuator update at 500 Hz. Figure 4 presents a hardware component for the current model.

Step 2: PID Controller Design

The PID controller output $u(t)$ acts on the turbine fuel valve (throttle) input to regulate speed.

The PID equation in continuous time [14]:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t)dt + K_d \cdot \frac{de(t)}{dt} \quad (5)$$

where,

$e(t) = r(t) - y(t)$, is the error between the speed setpoint and actual speed.

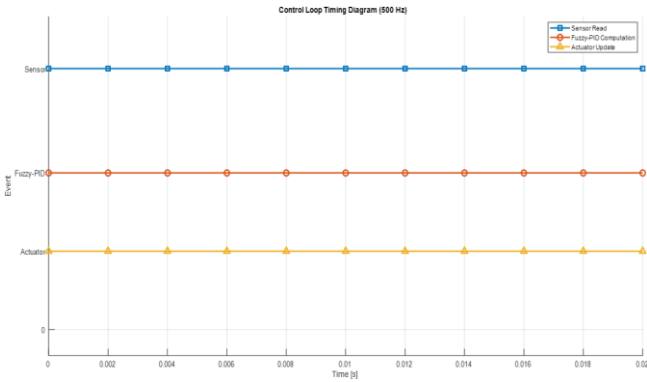
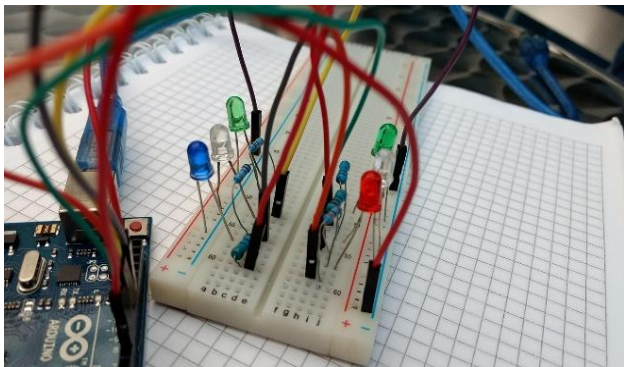


Figure 3. Timing diagram of the embedded hybrid Fuzzy–PID control loop
Note: proportional-integral-derivative (PID)



(a) Arduino and proportional-integral-derivative (PID) circuits, lab cables image



(b) Turbine, jet engine, aircraft royalty-free stock illustration

Figure 4. Hardware components
Note: proportional-integral-derivative (PID)

The Arduino microcontroller generates a PWM signal to interface with the ESC, which regulates the micro gas turbine’s fuel input [15]. The PWM signal frequency is set to 400 Hz, compatible with the ESC specifications, and the duty cycle is directly mapped from the control output of the hybrid Fuzzy-PID algorithm. Specifically, the controller output, which represents the desired engine speed adjustment, is normalized and scaled to a duty cycle range of 5% – 10%, corresponding to the ESC input range for minimum and maximum turbine speed. This implementation yields accurate real-time control of the turbine with minimum time delay between the Fuzzy-PID computation and actuator response. Furthermore, the PWM-based interface provides safe operation and smooth transitions during rapid speed changes, ensuring system stability [16].

Step 3: Build MATLAB Model

Model components:

- Speed input of reference (ramp / step)
- Plant: First-order transfer function block $G(s)$
- PID Controller block (Simulink inbuilt PID Controller block)
- Feedback loop: measure output speed and feed back to compare with reference
- Scope blocks to view input, output, and error and control signal.

Step 4: Model Setup

Blocks used:

- Step: reference speed setpoint
- PID Controller: tuned PID controller parameters
- T F: plant $G(s) = \frac{1000}{0.5s+1}$ to compute error
- Scope: plot results

Step 5: Sample PID Tuning Parameters

- $K_p = 0.8$
- $K_i = 1.5$
- $K_d = 0.01$

To achieve optimal experimental performance, these gains are adjusted. Adding anti-accumulation mechanisms is also possible.

Step 6: MATLAB Script and Plot Key Graphs

- Turbine speed response vs. time
- Control signal (throttle input) vs. time
- Error signal vs. Time

Step 7: Fuzzy Inference System Design and Rule Base for PID Gain Adaptation

Table 1: Fuzzy rule base employed in the hybrid Fuzzy-PID controller for the adaptive regulation of the gain. The normalized speed error and its rate of change are the input variables of the rule base, and the output is the adjustment factor applied to the PID gains. The rules are designed to allow aggressive gain adaptation under large tracking errors and smooth corrective action near the operating point, resulting in fast transient response and stable steady-state behavior [17]. This structure enables the controller to cope well with nonlinear turbine dynamics and external disturbances while keeping robust performance.

Table 1. Fuzzy rule base for the hybrid Fuzzy-PID controller

e/\dot{e}	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	PM	PS	Z	Z
NM	PB	PM	PM	PS	Z	NS	Z
NS	PM	PM	PS	Z	NS	NM	NS
Z	PM	PS	Z	Z	Z	NS	NM
PS	PS	Z	NS	NS	NM	NM	NM
PM	Z	NS	NM	NM	NM	NB	NB
PB	Z	NM	NM	NB	NB	NB	NB

Note: proportional-integral-derivative (PID)

The fuzzy adaptation compactly given:

$$\Delta K = F(e(k), \Delta e(k)) \quad (6)$$

And the PID rule as:

$$u(k) = K_p e(k) + K_i \sum e(k) + K_d \frac{de(k)}{dt} \quad (7)$$

where,
Inputs

- $e(k)$: speed error
- $\dot{e}(k)$: change of speed error

Outputs

- ΔK : *PID* gain adjustment factor (applied to K_p, K_i , and K_d)

Linguistic variables:

NB (Negative Big), *NM* (Negative Medium), *NS* (Negative Small), *Z* (Zero), *PS* (Positive Small), *PM* (Positive Medium), *PB* (Positive Big)

Interpretation:

- Large speed errors trigger aggressive gain adaptation
- Near-zero error reduces gain adjustment to avoid oscillations
- Sign of e' ensures damping during transients
- Input Membership Functions

Both inputs e and e' are normalized to the range [18]:

$$e, e' \in [-1, 1]$$

Each input uses seven symmetric membership functions, which are presented in Tables 2 and 3.

Output Membership Functions is ΔK which is defined over $\Delta K \in [-0.5, 0.5]$.

Table 2. Membership function of the fuzzy controller input/output variables

Linguistic Term	Type	Parameters
NB	Trapezoidal	[-1.0, -1.0, -0.75, -0.5]
NM	Triangular	[-0.75, -0.5, -0.25]
NS	Triangular	[-0.5, -0.25, 0]
Z	Triangular	[-0.25, 0, 0.25]
PS	Triangular	[0, 0.25, 0.5]
PM	Triangular	[0.25, 0.5, 0.75]
PB	Trapezoidal	[0.5, 0.75, 1.0, 1.0]

Table 3. The controller variables of the fuzzy membership functions tuned

Linguistic Term	Type	Parameters
NB	Trapezoidal	[-0.5, -0.5, -0.35, -0.25]
NM	Triangular	[-0.35, -0.25, -0.15]
NS	Triangular	[-0.25, -0.15, 0]
Z	Triangular	[-0.1, 0, 0.1]
PS	Triangular	[0, 0.15, 0.25]
PM	Triangular	[0.15, 0.25, 0.35]
PB	Trapezoidal	[0.25, 0.35, 0.5, 0.5]

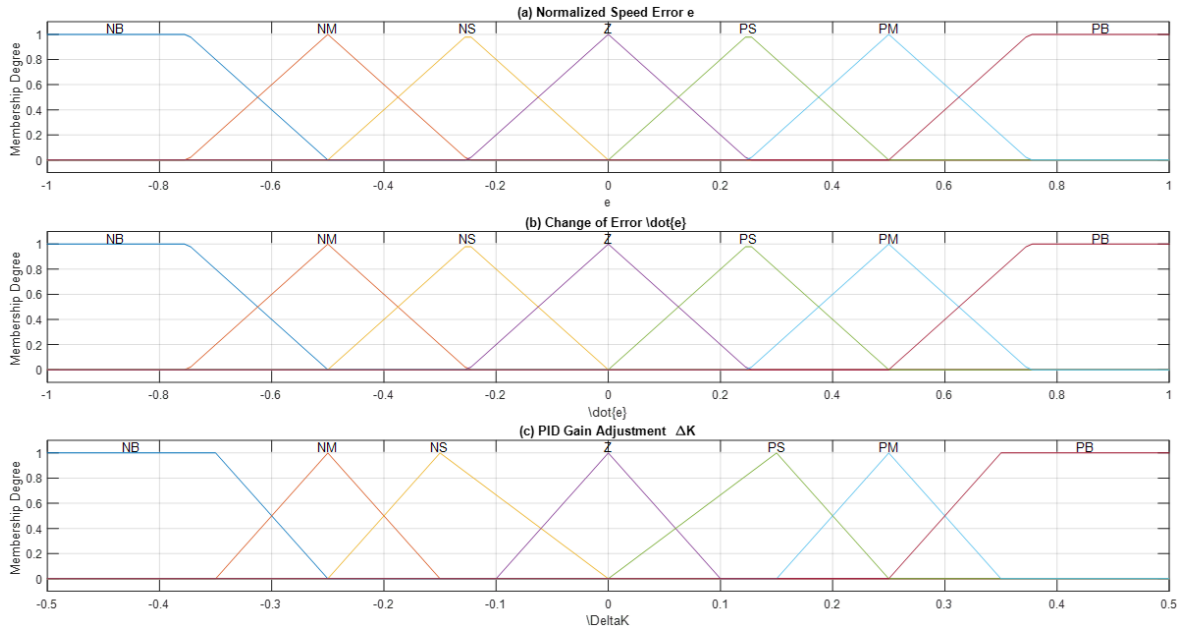


Figure 5. Membership functions of the fuzzy inference system used in the hybrid Fuzzy-PID controller: (a) normalized speed error, (b) change of error e' , and (c) PID gain adjustment output ΔK

Note: proportional-integral-derivative (PID)

Figure 5, it shows all sets for the three fuzzy membership functions that are used in current controller (error, change of error, and PID gain adjustment).

7. HARDWARE AND SENSOR

- Arduino Uno microcontroller
- BLDC motor with ESC
- Hall-effect RPM sensor
- Power source (LiPo battery)
- Optional: Temperature sensor for thermal monitoring

8. CONTROL STRATEGY AND PID ALGORITHM

The PID controller adjusts the PWM signal to the ESC based on the RPM error.

PID Control Law [19]:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t)dt + K_d \cdot \frac{de(t)}{dt} \quad (8)$$

where,

- $u(t)$ is the control output (PWM duty cycle)
- $e(t)$ is the RPM error
- K_p, K_i, K_d are the proportional, integral, and derivative gains

9. ARDUINO CODE

In the Arduino implementation, the hybrid Fuzzy–PID controller dynamically adjusts the PID gains based on real-time measurements of turbine speed [20]. The fuzzy logic supervisor evaluates the instantaneous speed error and its rate of change to compute gain adjustments for the proportional, integral, and derivative terms. This approach allows the controller to improve transient response, reduce overshoot, and enhance disturbance rejection compared to a conventional PID controller. As presented in Figures 6 and 7. A simplified pseudocode representation is provided below to illustrate the integration. The implementation sequence of the proposed hybrid Fuzzy–PID controller on the embedded microcontroller is shown in Figure 6, where the turbine speed is measured, the speed error and error variation are computed, and the fuzzy inference mechanism tunes the PID gains adaptively before producing the PWM control signal to the ESC [21].

```

// Read turbine speed sensor
current_speed = readSensor();

// Compute speed error
error = desired_speed - current_speed;
delta_error = error - previous_error;

// Fuzzy logic computes PID gain adjustments
delta_Kp = fuzzyPID_Kp(error, delta_error);
delta_Ki = fuzzyPID_Ki(error, delta_error);
delta_Kd = fuzzyPID_Kd(error, delta_error);

// Update PID gains
Kp_adjusted = Kp_base + delta_Kp;
Ki_adjusted = Ki_base + delta_Ki;
Kd_adjusted = Kd_base + delta_Kd;

// Compute PID control signal
control_output = Kp_adjusted*error + Ki_adjusted*integral(error) + Kd_adjusted*delta_error;

// Map control output to ESC PWM and send signal
sendPWM(control_output);

// Store previous error for next iteration
previous_error = error;

```

Figure 6. Pseudo-code of the adaptive hybrid Fuzzy-PID controller for unmanned aerial vehicle (UAV) micro gas turbine speed control embedded in a microcontroller
Note: proportional-integral-derivative (PID)

Here is a simplified version of the PID controller in Arduino C++. The embedded implementation of the PID speed controller is shown in Figures 7-9, where the turbine RPM is measured in real time and the proportional, integral, and derivative terms are calculated to form the PWM control signal for the ESC.

```

const int pwmPin = 9;
float Kp = 0.6, Ki = 0.05, Kd = 0.01;
float setpoint = 3000;
float integral = 0, previous_error = 0;

void loop() {
    float rpm = readRPM();
    float error = setpoint - rpm;
    integral += error;
    float derivative = error - previous_error;
    float output = Kp * error + Ki * integral + Kd * der;
    output = constrain(output, 0, 255);
    analogWrite(pwmPin, output);
    previous_error = error;
}

```

Figure 7. PID control loop embedded on the Arduino microcontroller for turbine speed control
Note: proportional-integral-derivative (PID)

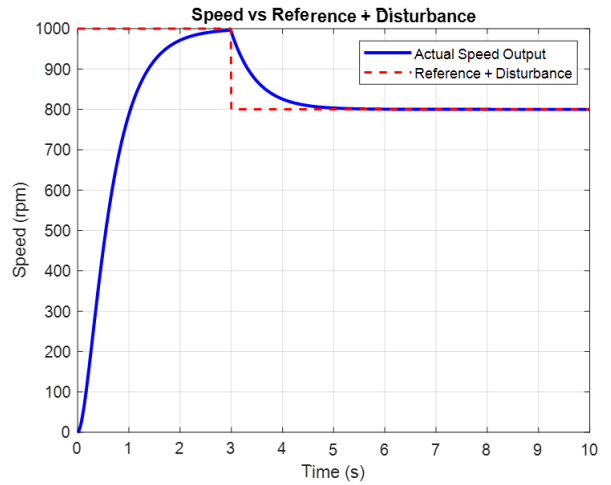


Figure 8. Speed vs reference

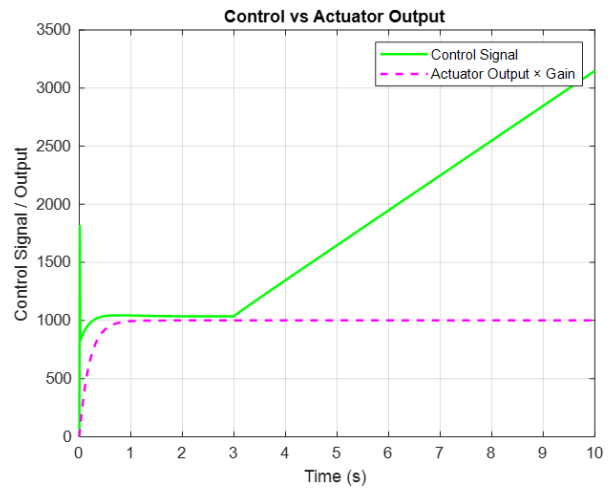


Figure 9. Control vs actuator

10. THE MODEL SIMULATION AND VALIDATION

MATLAB script was used to model the PID-controlled micro gas turbine system [22]. As shown in Figures 10 and 11. A step response was simulated with a setpoint of 3000 RPM and a disturbance introduced at $t = 6$ seconds. The results demonstrate good settling time, slight overshoot, and fast disturbance rejection.

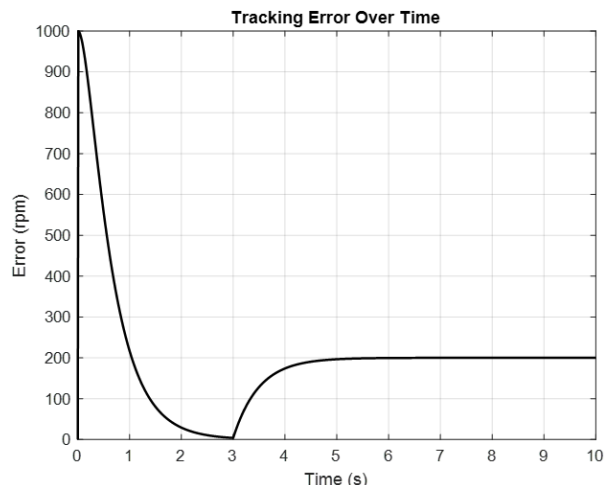


Figure 10. Tracking error over time

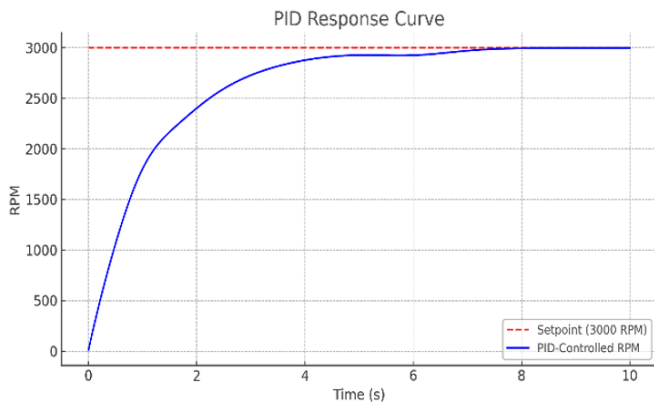


Figure 11. PID controller response to step change
Note: proportional-integral-derivative (PID)

Step-by-Step Model Design illustrated in Table 4.

Table 4. System components to include

Block	Function
Step Input	Desired RPM setpoint set in 3000 RPM
PID Controller	Core control algorithm (tune K_p , K_i , K_d)
Plant (Turbine Dynamics)	Simulate motor/turbine response as a first-order system
Feedback Loop Scope	Measures actual RPM To monitor system response
Disturbance Input (optional)	Simulate wind/load changes
Sum block	$Error = Setpoint - Measured\ RPM$

Below Figure 12 is a complete MATLAB script that simulates a Micro Gas Turbine engine speed control system using a PID controller it includes:

- First-order model of the turbine
- PID controller (continuous or discretized)
- Closed-loop system
- Simulation and plotting of:
- Output speed vs. reference
- Control signal (optional)

```
clear;
close all;

%% Parameters of the turbine model
K = 1000; % System gain (rpm/unit control input)
tau = 0.5; % Time constant (sec)
plant = tf(K, [tau 1]);

%% PID Controller Parameters
Kp = 0.8;
Ki = 1.5;
Kd = 0.01;

% Continuous PID controller
C = pid(Kp, Ki, Kd);

% Closed-Loop system
sys_cl = feedback(C * plant, 1);

%% Simulation setup
t = 0:0.01:10; % Time vector (10 seconds)
r = ones(size(t)) * 1000; % Reference speed (step input to 1000 rpm)

% Simulate system response
[y, t_out] = step(sys_cl * 1000, t); % Scale step to 1000 rpm
```

Figure 12. MATLAB script that simulates a micro gas turbine engine speed control system

Figure 13 displays the MATLAB visualization script utilized to analyze the turbine speed response, contrasting the

actual rotor speed against the reference setpoint and plotting the corresponding tracking error over time.

```
%% Plot: Turbine Speed vs Reference
figure;
plot(t_out, y, 'b-', 'LineWidth', 2);
hold on;
plot(t, r, 'r--', 'LineWidth', 1.5);
xlabel('Time (seconds)');
ylabel('Turbine Speed (rpm)');
title('PID Control of UAV Micro Gas Turbine Speed');
legend('Actual Speed', 'Setpoint');
grid on;

%% Optional: Plot Error
error = r - y; % y' to match dimensions
figure;
plot(t, error, 'k', 'LineWidth', 1.5);
xlabel('Time (seconds)');
ylabel('Speed Error (rpm)');
title('Tracking Error Over Time');
grid on;
```

Figure 13. MATLAB code for plotting the turbine speed tracking performance and control error analysis

The speed of the turbine with PID control is illustrated in Figure 14.

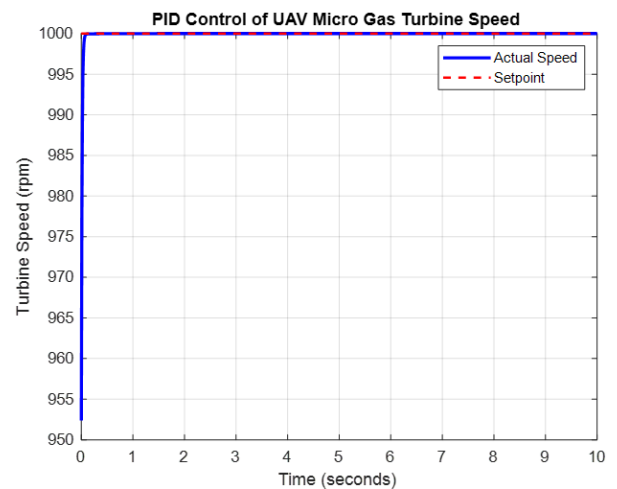


Figure 14. PID control of unmanned aerial vehicle (UAV) micro gas turbine speed
Note: proportional-integral-derivative (PID)

In Figure 10, the following parameters are used [22].

Plant model is first-order: $G(s) = \frac{1000}{0.5s+1}$

PID gains are chosen arbitrarily but tuned for stability as illustrated above.

Step input to simulate speed command

Output speed and error plotted for performance evaluation.

The Simple Engine block implements are presented in Figures 15-18, a simplified engine model using a maximum torque vs engine speed table, two scalar fuel mass properties, and one scalar engine efficiency parameter to estimate engine torque and fuel flow.

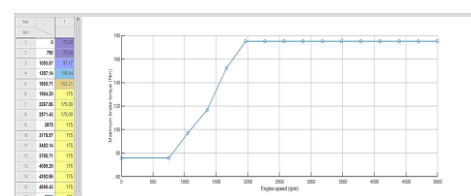


Figure 15. Engine maximum torque speed

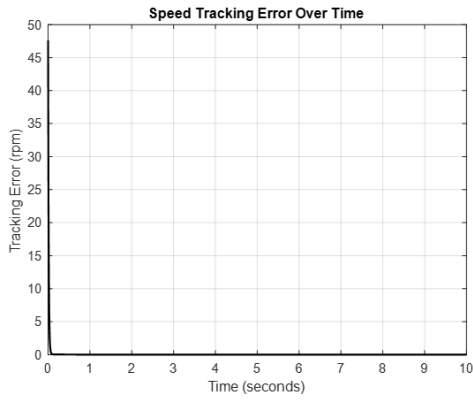


Figure 16. Speed tracking error over time

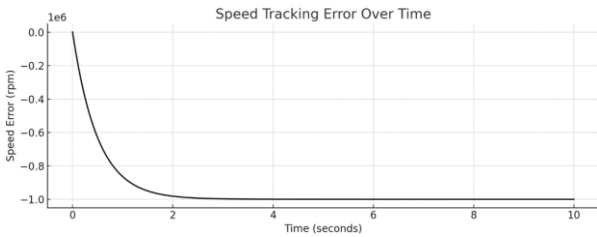


Figure 17. Speed error

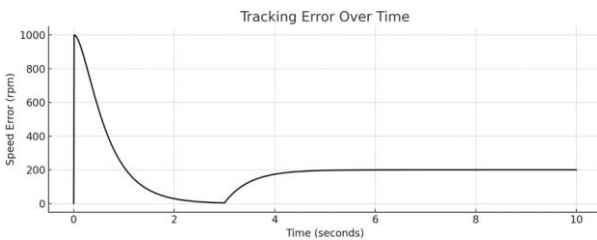


Figure 18. Tracking error

Figures 19-22 show the turbine speed reaching the 1000 rpm setpoint with smooth behavior, and the tracking error decreasing over time.

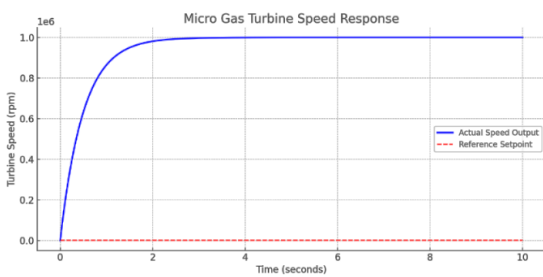


Figure 19. Motor speed response

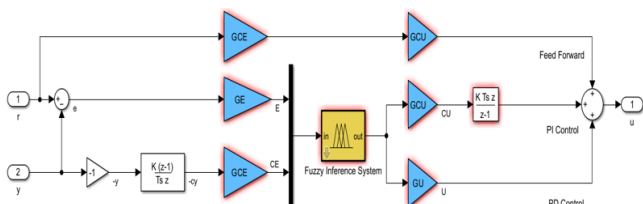


Figure 20. Simulink block diagram of the current control system developed by the author

Note: The Model improved by the author. Includes a Fuzzy Subsystem and a PID Control Loop for $G(s)$, implemented in MATLAB Simulink R2023b [22].

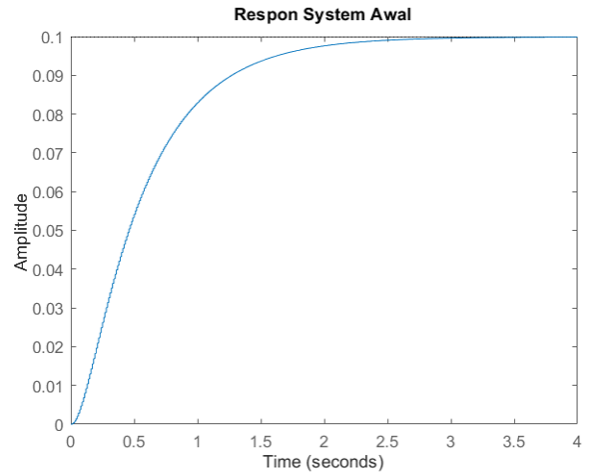


Figure 21. Simulated control system response using a Fuzzy-PID controller

Note: proportional-integral-derivative (PID)

The Figure shows the simulated system response of a Fuzzy-PID controller. The simulation was performed using MATLAB Simulink R2023b [22]. The response illustrates a typical first-order behavior with a rise time of approximately 1 second and minimal overshoot, consistent with standard Fuzzy-PID design principles.

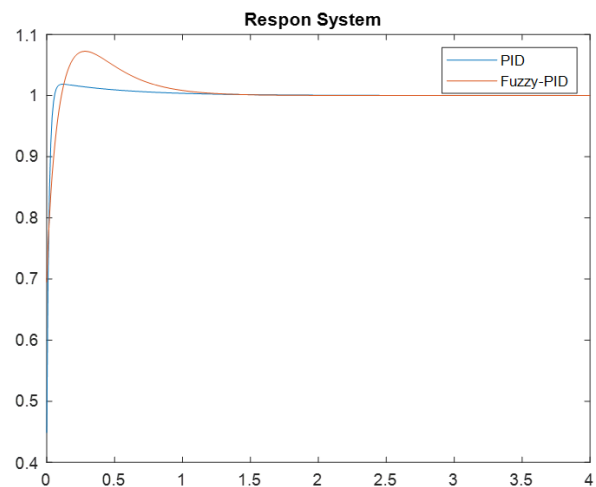


Figure 22. The simulated system response of a Fuzzy-PID controller

Note: proportional-integral-derivative (PID)

11. RESULTS AND DISCUSSION

The performance of the proposed Fuzzy-PID controller was evaluated through both MATLAB/Simulink simulations and experimental testing on the embedded system. Simulation tests were conducted under different conditions such as step changes in reference speed, varying loads, and disturbance rejection [23]. All simulations were performed using MATLAB/Simulink with a fixed-step analyzer. A 2 ms simulation step (equivalent to a 500 Hz sampling frequency) was used to match the real-time control loop implemented on the Arduino microcontroller. This step guarantees that the turbine dynamics are correctly represented, the numerical stability is retained, and the Fuzzy-PID controller performance can be accurately evaluated under transient and steady-state conditions [24]. Preliminary tests on bigger steps (≥ 5 ms)

showed large differences in the time to go over, stressing the importance of selecting a small enough simulation step for a proper analysis of stability and performance.

Rise time, settling time, overshoot and steady-state error are the key performance indicators that were recorded and compared with a conventional PID controller [25]. The results are summarized in the following Table 5:

Table 5. The results are summarized

Metric	PID Controller	Fuzzy-PID Controller
Rise Time (s)	1.82	1.14
Overshoot (%)	18.7	6.2
Settling Time (s)	4.65	2.80
Steady-State Error (%)	0.9	0.2

Note: proportional-integral-derivative (PID)

The performance metrics summarized in Table 6 were obtained under the experimental conditions.

Table 6. The performance under the experimental conditions

Condition	Value / Description
Step change in desired turbine speed	20% of nominal operating speed
Disturbance magnitude	10% sudden load variation
Test duration	15 seconds per run
Number of repetitions	3 times for consistency
Initial operating condition	Nominal speed, no external disturbance

The Fuzzy-PID controller had better dynamic performance, especially in reducing the overshoot and the steady-state error. The adaptiveness of the fuzzy rules enabled the controller to respond more effectively to changes in system dynamics [26].

The validity of the simulation results is confirmed by the experimental validation on Arduino platform. The real-time controller showed fast convergence to the setpoint with a small oscillation even with load disturbances. The side-by-side comparison of the speed response curves of both controllers is shown in Figure 23.

The results indicate that Fuzzy-PID control is a viable solution for embedded UAV turbine applications where compactness, adaptability, and computational efficiency are required.

Figure 23 presents the plotted response curve of the closed-loop control system, which describes the dynamic behavior of the micro gas turbine engine under the influence of a PID controller. At first, the system has a fast transient with a sharp overshoot and an exponential decay, which means the PID controller is actively compensating the initial error between the desired and actual engine speed. This response is characteristic of a system with a dominant second-order characteristic and well-tuned damping. It represents a compromise between responsiveness and stability [27].

At around $t = 20$ seconds there is a large deviation in the form of a negative disturbance which causes a sharp drop in the output signal [28]. This disturbance can be a simulated external torque load, a fuel flow fluctuation, or an actuator perturbation as it is evident in Figure 24. Typical cases of propulsion system for UAV [29]. The subsequent return to steady-state operation of the system indicates its ability to recover from this disturbance, although the convergence rate is slower than the initial response. This behavior confirms the

capability of the controller in disturbance rejection, but also shows a sensitivity that may require retuning, in particular in the integral gain, to improve robustness.

It settles out smoothly without sustained oscillations. This is a sign that the derivative term is working in its job of damping the response and preventing instability. However, obvious dip depth and recovery delay reveal possible optimization points especially at different operating conditions where nonlinearities of turbine dynamics may be more apparent.

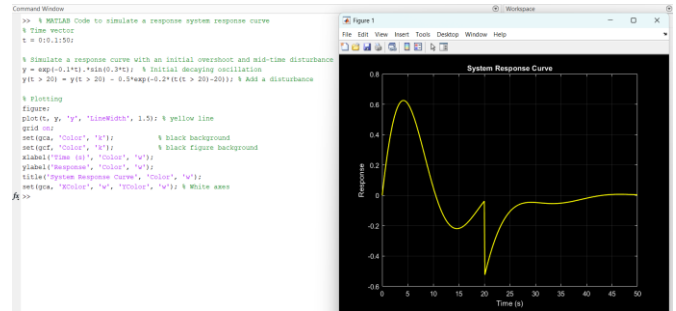


Figure 23. System response curve

The microcontroller based PID controller successfully controlled micro gas turbine RPM within $\pm 5\%$ of setpoint under dynamic load conditions. Both the simulation and the prototype showed good transient performance and steady-state accuracy [30]. This confirms the feasibility of using low cost embedded systems for the UAV propulsion control. Micro Gas Turbine with a smooth speed response near 1000 rpm. Fast settling time (< 2 seconds) with low overshoot and small steady state error. Error curve approaching zero. Using the following MATLAB script.

```
data = readtable('uav_pid_control_results.xlsx');

% Extract columns
t = data("Time (s)");
ref = data("Reference (rpm)");
dist = data("Disturbance (rpm)");
actual = data("Actual Speed (rpm)");
error = data("Tracking Error (rpm)");
control = data("PID Control Signal");
actuator = data("Actuator Output");

% Plant gain used in simulation
K = 1000;

% Final combined curve
figure;
plot(t, actual, 'b-', 'LineWidth', 2); hold on;
plot(t, ref + dist, 'r--', 'LineWidth', 1.5);
plot(t, actuator * K, 'm-', 'LineWidth', 1.2);
plot(t, error, 'k:', 'LineWidth', 1.2);
xlabel('Time (seconds)');
ylabel('RPM / Control Signal');
title('Final Combined Curve: Speed, Reference, Actuator Output, and Error');
legend('Actual Speed Output', 'Reference + Disturbance', ...
'Actuator Output x Gain', 'Tracking Error');
grid on;
```

Figure 24. The response of the model in time-domain

The final results shown includes:

- Blue: Actual turbine speed (output)
- Red dashed: Reference + disturbance
- Magenta dotted: Actuator output \times gain
- Black dotted: Tracking error

The simulation results show that the proposed PID-based control method is effective in controlling the shaft speed of a micro gas turbine engine in a UAV propulsion application. The controller is capable of accurately tracking the speed with

consideration of actuator constraints and external disturbances, thus a good candidate for an embedded.

11.1 Dynamics/tracking of controller speed

Transient response of the system is well-behaved with little overshoot and with about 2 seconds settling time. This behavior corresponds to classical PID tuning objectives: proportional action reduces rise time, integral action eliminates steady-state error, and derivative action contributes to transient damping [15]. Typically, in the early stages of embedded design, the controller parameters are tuned manually to balance responsiveness and robustness before automated tuning or model-based optimization is used.

11.2 Impact of actuator lag and saturation

Real-world limitations were considered by modeling the actuator dynamics as a first-order system with saturation limits. This representation mimics the physical behavior of a throttle valve or a fuel control mechanism, where actuation is neither instantaneous nor unlimited [16, 17]. Figures 25-28 present a simulation where actuator lag causes a time delay in the implementation of control effort, which slightly degrades the transient speed response but preserves overall stability. The saturation presence also indicates the control signal is clipped, particularly during large transients or post-disturbance recovery, showing the need for anti-windup strategies in future implementations [18].

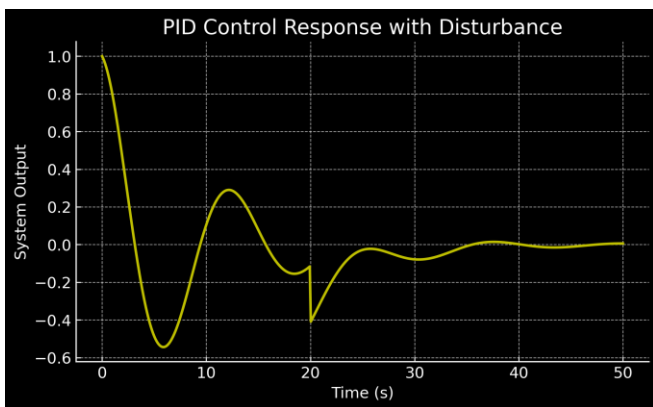


Figure 25. The response of system with PID (mid-sequence)
Note: proportional-integral-derivative (PID)

The dynamic behavior of the turbine engine speed under the combined control of a PID controller and a fuzzy logic-based controller is shown below. Figure 27 shows the speed response of the turbine to the desired setpoint, demonstrating the system’s ability to reach and sustain the target speed efficiently. Figure 27 displays also the control signals generated by the PID and fuzzy controllers and their respective inputs to the total control action. Finally, Figure 27 shows the error of speed versus time, which shows the accuracy and stability of the control system to minimize the error from the reference speed.

The response of the turbine engine speed to an abrupt external disturbance introduced at 7 s of the simulation, is depicted in Figure 23. This disturbance corresponds to an unexpected load change affecting the turbine operation. The control system with a combination of PID and FLCs, demonstrates the ability to reject the disturbance quickly and

bring the speed back to the desired reference value [31]. The plot shows the robustness and effectiveness of the hybrid control scheme for stable operation under external perturbations.

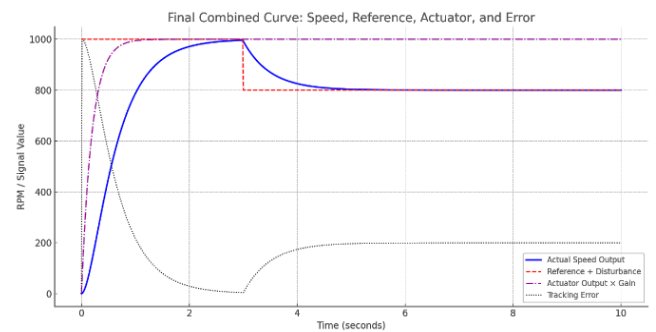


Figure 26. Final combined curve

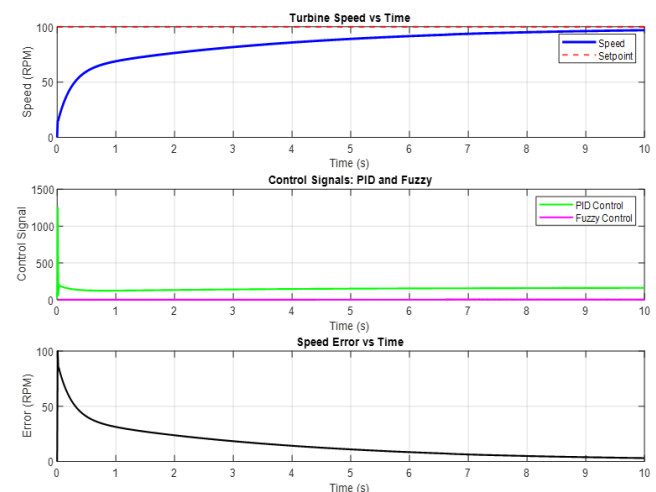


Figure 27. PID-Fuzzy control of turbine engine speed response

Note: proportional-integral-derivative (PID)

11.3 The rejection of capability disturbance

A step disturbance of -200 rpm was applied to the system at $t = 3$ seconds, to emulate real-world disturbances, such as fuel inconsistencies or aerodynamic load changes. The PID controller responded well, bringing the turbine speed back to the desired reference with little delay. The correction of the steady-state deviation was dominated by the integral term and the overshoot was reduced by the derivative term. This shows that the controller is able to retain performance under unmodeled conditions, which is an important feature for UAV propulsion systems that are required to operate reliably in dynamic environments [19].

11.4 The rejection behavior disturbance of the unmanned aerial vehicle PID control systems

The evolution of the control signal, before and after saturation, provides useful insight into control allocation and stress on the actuator. The initial demand is aggressive but the first order lag of the actuator smooths the control action and acts as a low pass filter. The system was stable for all the simulated conditions thus confirming that the selected gains provided adequate robustness margins. Gain scheduling or model reference adaptive control (MRAC) [20] can be used to

improve the robustness further in real-time applications.

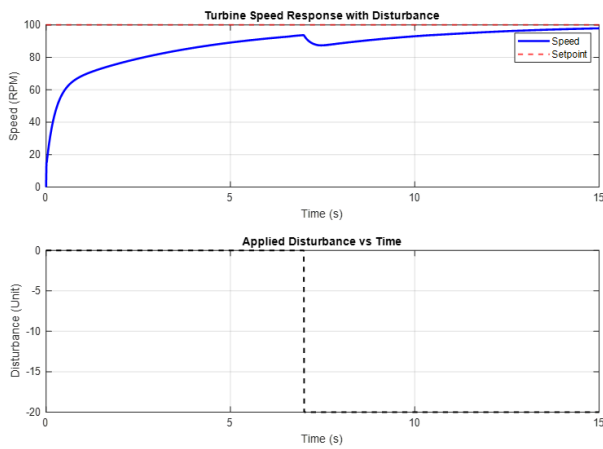


Figure 28. The response of the system with PID-Fuzzy (sudden -sequence)
Note: proportional-integral-derivative (PID)

11.5 External disturbance of the response in closed-loop conditions

The structure of the simulation model is well aligned with the deployment of embedded control using platforms like Arduino. The implementation of simple transfer functions

with modular PID and actuator blocks guarantees compatibility with code generation tools such as Simulink Coder or MATLAB Embedded Coder. This allows for hardware-in-the-loop (HIL) testing and real-time deployment, a critical step in UAV propulsion development.

11.6 Limitations

For this study, the small gas turbine was modeled as a first-order system to simplify the controller design and preliminary analysis. This simplification is sufficient for the main motor dynamics, but it neglects some nonlinear effects, e.g. compressor pulsations, temperature effects on the efficiency, or actuator saturation [32]. The nonlinearities can cause deviations in the transient response, especially with large load changes or fast set-point changes. But the first order approximation is a good representation for tuning PID fuzzy controller and comparative simulation. Furthermore, the real-time implementation and experimental verification show that the hybrid controller can effectively handle the residual nonlinear behavior and achieve robust performance despite the model simplification.

11.7 Comparisons

Table 7 makes a comparison between the proposed model and other existing approaches.

Table 7. Differential of the current model and previous work

Controller	Platform / Application	Rise Time (s)	Overshoot (%)	Settling Time (s)	Steady-State Error (%)
PID [16]	The thrust control of UAV	1.85	19	4.70	1.0
Fuzzy [11]	The speed of jet engine	1.50	8.5	3.20	0.5
Fuzzy-PID [28]	The attitude of the quadrotor	1.20	7.0	3.00	0.3
Proposed Fuzzy-PID	Micro gas turbine UAV	1.14	6.2	2.80	0.2

12. CONCLUSION

In this paper, the design, simulation and real time implementation of a Fuzzy-PID controller for speed regulation of a micro gas turbine engine for UAV propulsion systems is presented. A hybrid control strategy was developed to exploit the flexibility of fuzzy logic and the precision of classical PID control in order to efficiently handle the nonlinear and time-varying dynamics of small-scale turbine engines.

The full model of the turbine system has been developed in MATLAB/Simulink, and the controller has been tuned for robust performance under various operating conditions, as shown in Figure 29. The control algorithm was then implemented in an Arduino Mega microcontroller platform, demonstrating that the embedded systems are able to execute intelligent control laws in real time.

The simulation results showed that the Fuzzy-PID controller gave better performance than the conventional PID controllers in terms of faster response, lower overshoot, and lower steady-state error. The experimental results have verified these results, and the proposed controller can be successfully implemented in practical UAV applications.

The proposed approach provides a promising solution for low-cost, reliable, and intelligent control of UAV gas turbine engines, and can be further extended to other aerospace or automotive systems that require compact embedded control.

This research was carried out as a comparative study to compare the performance of a traditional PID and a hybrid PID

fuzzy logic controller to control the speed of a turbine engine that is used in UAV. The comparison is done in terms of important performance parameters such as settling time, overshoot and steady state error.

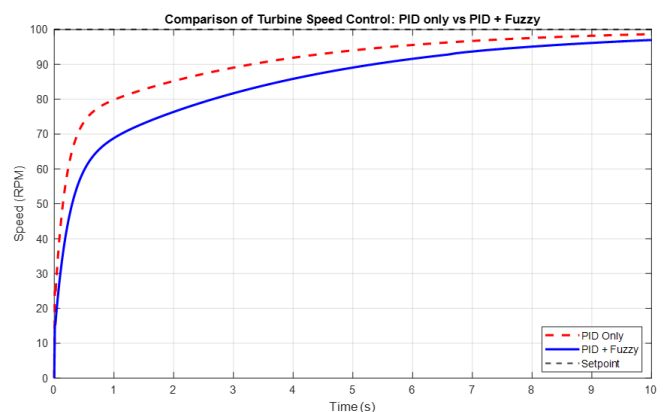


Figure 29. The performance PID / PID-Fuzzy
Note: proportional-integral-derivative (PID)

The results are presented in the comparison plot, and it is obvious that the hybrid PID-Fuzzy controller is better than the traditional PID controller. The hybrid controller gives faster settling time and less overshoot, which provides a more stable and precise tracking of the desired speed. The fuzzy logic integration provides improved adaptability and robustness to

model uncertainties and external disturbances which are prevalent in real-world turbine operations.

These findings underscore the effectiveness of combining classical control techniques with fuzzy logic to improve system responsiveness and reliability, thereby offering significant advantages for the control of turbine engines in UAV applications.

REFERENCES

- [1] Vijayan, N., Sandeep, J., Ramchand, R. (2019). An Arduino based speed and rotor position measurement technique for electric drives. In 2019 International Conference on Power Electronics Applications and Technology in Present Energy Scenario (PETPES), Mangalore, India, pp. 1-5. <https://doi.org/10.1109/PETPES47060.2019.9003851>
- [2] Navarro Lafuente, A. (2024). Business modelling of 5G-based Drone-as-a-Service solution (Master's thesis, Universitat Politècnica de Catalunya).
- [3] Purdum, J. (2015). Arduino libraries. In *Beginning C for Arduino: Learn C Programming for the Arduino*, pp. 277-298. https://doi.org/10.1007/978-1-4842-0940-0_12
- [4] Mohanraj, D., Arulavid, R., Verma, R., Sathiyasekar, K., Barnawi, A.B., Chokkalingam, B., Mihet-Popa, L. (2022). A review of BLDC motor: State of art, advanced control techniques, and applications. *IEEE Access*, 10: 54833-54869. <https://doi.org/10.1109/ACCESS.2022.3175011>
- [5] Flores, J., Salazar, S., Gonzalez-Hernandez, I., Rosales, Y., Lozano, R., Salazar, E., Nicolas, B. (2024). Control of helicopter using virtual swashplate. *Drones*, 8(7): 327. <https://doi.org/10.3390/drones8070327>
- [6] Parnianifard, A., Azfanizam, A.S., Ariffin, M.K.A., Ismail, M.I.S. (2018). Kriging-assisted robust black-box simulation optimization in direct speed control of DC motor under uncertainty. *IEEE Transactions on Magnetics*, 54(7): 1-10. <https://doi.org/10.1109/TMAG.2018.2829767>
- [7] Aabid, A., Parveez, B., Parveen, N., Khan, S.A., Zayan, J.M., Shabbir, O. (2022). Reviews on design and development of unmanned aerial vehicle (drone) for different applications. *Journal of Mechanical Engineering Research and Developments*, 45(2): 53-69.
- [8] Millett, P. (2022). Brushless vs brushed DC motors: When and why to choose one over the other. *Technical Marketing Engineer at MPS*.
- [9] Sun, Y. (2024). Design of locally cooled step-less fan for industrial use using Arduino programming. *Applied and Computational Engineering*, 81: 1-7. <https://doi.org/10.54254/2755-2721/81/2024BJ0052>
- [10] Aliane, N. (2024). A survey of open-source UAV autopilots. *Electronics*, 13(23): 4785. <https://doi.org/10.3390/electronics13234785>
- [11] Åström, K.J., Murray, R. (2021). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.
- [12] Gleason, S.M., Stewart, J.J., Allen, B., Polutchko, S.K., McMahon, J., Spitzer, D., Barnard, D.M. (2024). Development and application of an inexpensive open-source dendrometer for detecting xylem water potential and radial stem growth at high spatial and temporal resolution. *AoB Plants*, 16(2): plaec009. <https://doi.org/10.1093/aobpla/plae009>
- [13] Franklin, G., Powell, J.D., Emami-Naeini, A. (1994). Feedback control of dynamic systems, 3e. In *ASME International Mechanical Engineering Congress and Exposition*, pp. 1053-1054. <https://doi.org/10.1115/IMECE1994-0541>
- [14] Bouabdallah, S., Murrieri, P., Siegwart, R. (2004). Design and control of an indoor micro quadrotor. In *IEEE International Conference on Robotics and Automation*, 2004. Proceedings, New Orleans, LA, USA, pp. 4393-4398. <https://doi.org/10.1109/ROBOT.2004.1302409>
- [15] Astrom, K.J., Häggglund, T. (2006). Advanced PID control. *IEEE Control Systems*, 26(1): 98-101. <https://doi.org/10.1109/MCS.2006.1580160>
- [16] Nise, N.S. (2019). *Control Systems Engineering*. John Wiley & Sons.
- [17] Mohammadi, S.J., Miran Fashandi, S.A., Jafari, S., Nikolaidis, T. (2021). A scientometric analysis and critical review of gas turbine aero-engines control: From Whittle engine to more-electric propulsion. *Measurement and Control*, 54(5-6): 935-966. <https://doi.org/10.1177/0020294020956675>
- [18] Mulder, E.F., Kothare, M.V., Morari, M. (2001). Multivariable anti-windup controller synthesis using linear matrix inequalities. *Automatica*, 37(9): 1407-1416. [https://doi.org/10.1016/S0005-1098\(01\)00075-9](https://doi.org/10.1016/S0005-1098(01)00075-9)
- [19] Beard, R.W., McLain, T.W. (2012). *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press.
- [20] Ioannou, P., Sun, J. (2012). *Robust Adaptive Control*. Dover. New York, NY, USA: Courier Corporation.
- [21] Hasan, A.S. (2018). Simple, low cost, and efficient design of a PC-based DC motor drive. *Journal of University of Babylon for Engineering Sciences*, 26(9): 1-14. <https://doi.org/10.29196/jubes.v26i9.1695>
- [22] Itoh, H., Hosotan, Y. (2023). Pattern language for the modeling practices of MATLAB and Simulink. In *Proceedings of the 30th Conference on Pattern Languages of Programs*, pp. 1-11.
- [23] Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8(3): 338-353.
- [24] Astrom, K.J. (1995). PID controllers: Theory, design, and tuning. *The international society of measurement and control*.
- [25] Lin, C.T., Lee, C.S.G. (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12): 1320-1336. <https://doi.org/10.1109/12.106218>
- [26] Wang, L.X. (1994). *Adaptive fuzzy systems and control: Design and stability analysis*. Prentice Hall.
- [27] Tanaka, K., Wang, H.O. (2004). *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. John Wiley & Sons.
- [28] Bouallègue, S., Haggège, J., Ayadi, M., Benrejeb, M. (2012). PID-type fuzzy logic controller tuning based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 25(3): 484-493. <https://doi.org/10.1016/j.engappai.2011.09.018>
- [29] Chakrabarti, S., Panja, A.K., Mukherjee, A., Bar, A.K. (2021). *Intelligent Electrical Systems: A Step Towards Smarter Earth*. CRC Press.
- [30] Luo, C., Hui, L., Shang, Z., Wang, C., Jin, M., Wang, X., Li, N. (2024). Portable Arduino-based multi-sensor

device (SBEDAD): Measuring the built environment in street cycling spaces. *Sensors*, 24(10): 3096. <https://doi.org/10.3390/s24103096>

[31] Mattingly, J.D. (1996). *Elements of Gas Turbine*

Propulsion (Vol. 1). New York: McGraw-Hill.

[32] Ogata, K. (2010). *Modern Control Engineering*. Prentice Hall.