

Article

UAV Trajectory Tracking Using Proportional-Integral-Derivative-Type-2 Fuzzy Logic Controller with Genetic Algorithm Parameter Tuning

Oumaima Moali ^{1,*}, Dhafer Mezghani ¹, Abdelkader Mami ¹, Abdelatif Oussar ² and Abdelkrim Nemra ²

¹ UR-LAPER, Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis 2092, Tunisia; dhafer.mezghani@fst.utm.tn (D.M.); abdelkader.mami@fst.utm.tn (A.M.)

² School of Control and Automation, Ecole Militaire Polytechnique, EMP, Bordj El Bahri, Algiers 16111, Algeria; oussarabdelatif@gmail.com (A.O.); karim_nemra@yahoo.fr (A.N.)

* Correspondence: moali.oumaima@gmail.com

Abstract: Unmanned Aerial Vehicle (UAV)-type Quadrotors are highly nonlinear systems that are difficult to control and stabilize outdoors, especially in a windy environment. Many algorithms have been proposed to solve the problem of trajectory tracking using UAVs. However, current control systems face significant hurdles, such as parameter uncertainties, modeling errors, and challenges in windy environments. Sensitivity to parameter variations may lead to performance degradation or instability. Modeling errors arise from simplifications, causing disparities between assumed and actual behavior. Classical controls may lack adaptability to dynamic changes, necessitating adaptive strategies. Limited robustness in handling uncertainties can result in suboptimal performance. Windy environments introduce disturbances, impacting system dynamics and precision. The complexity of wind modeling demands advanced estimation and compensation strategies. Tuning challenges may necessitate frequent adjustments, posing practical limitations. Researchers have explored advanced control paradigms, including robust, adaptive, and predictive control, aiming to enhance system performance amidst uncertainties in a scientifically rigorous manner. Our approach does not require knowledge of UAVs and noise models. Furthermore, the use of the Type-2 controller makes our approach robust in the face of uncertainties. The effectiveness of the proposed approach is clear from the obtained results. In this paper, robust and optimal controllers are proposed, validated, and compared on a quadrotor navigating an outdoor environment. First, a Type-2 Fuzzy Logic Controller (FLC) combined with a PID is compared to a Type-1 FLC and Backstepping controller. Second, a Genetic Algorithm (GA) is proposed to provide the optimal PID-Type-2 FLC tuning. The Backstepping, PID-Type-1 FLC, and PID-Type-2 FLC with GA optimization are validated and evaluated with real scenarios in a windy environment. Deep robustness analysis, including error modeling, parameter uncertainties, and actuator faults, is considered. The obtained results clearly show the robustness of the optimal PID-Type-2 FLC compared to the Backstepping and PID-Type-1 FLC controllers. These results are confirmed by the numerical index of each controller compared to the PID-type-2 FLC, with 12% for the Backstepping controller and 51% for the PID-Type-1 FLC.

Keywords: quadrotor; Type-1 FLC; Type-2 FLC; backstepping controller; robustness analyses; parameter uncertainties; genetic algorithm; wind gust



Citation: Moali, O.; Mezghani, D.; Mami, A.; Oussar, A.; Nemra, A. UAV Trajectory Tracking Using Proportional-Integral-Derivative-Type-2 Fuzzy Logic Controller with Genetic Algorithm Parameter Tuning. *Sensors* **2024**, *24*, 6678. <https://doi.org/10.3390/s24206678>

Academic Editors: Bo Zhang, Yue Wei, Shiyu Chen, Yu Hu and Yaohua Liu

Received: 19 November 2023

Revised: 1 January 2024

Accepted: 5 January 2024

Published: 17 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Using robots in search and rescue (SAR) operations has become a significant issue for a variety of applications. Search and rescue missions, as well as simulations, have revealed a number of areas where robot contributions could be improved. The quadrotor system is one of the most efficient and complicated robots, and it is widely employed in military and commercial applications. Autonomous navigation of a quadrotor (stability, trajectory tracking, obstacle avoidance, etc.) remains a difficult problem to solve.

In recent year, UAVs have known greater popularity for their applications due to inexpensive operating costs and high performance. They have been used in several fields, including surveillance and reconnaissance, battle assessment, and aerial photography. As depicted in Figure 1, a quadrotor is a multi-rotor micro aerial vehicle that is lifted and powered by four rotors. Quadrotor guidance, navigation, and control have become an active area of research among the versatile flying robotic platforms due to their exceptional rotational agility, mechanical simplicity, relatively small size, Vertical Take-Off and Landing (VTOL) ability, and affordability [1]. Furthermore, because it is unmanned, UAV operation eliminates all expenses and hazards connected with onboard human pilots in both civilian and military domains, such as landscape mapping, agricultural surveying, monitoring, and aerial photography [2].

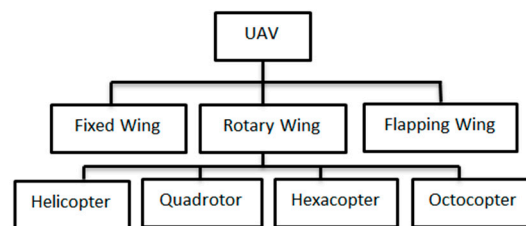


Figure 1. UAV classification.

The originality of this work consists of optimizing FLC Type-1 and Type-2 as well as backstepping controllers using a genetic algorithm, to improve their control robustness for quadrotor trajectory tracking in an uncertain environment. Three controllers are implemented and compared: the PID-Type-1 FLC, the PID-Type-2 FLC, and the backstepping controller with genetic algorithm optimization. First, a Type-2 FLC combined with a PID is compared to a PID-Type-1 FLC and Backstepping controller. Second, a GA is proposed to provide the optimal PID-Type-2 FLC tuning. The Backstepping controller, PID-Type-1 FLC, and PID-Type-2 FLC with GA optimization are validated and evaluated with real scenarios in a windy environment. Deep robustness analysis, including error modeling, parameter uncertainties, and actuator faults, is considered.

The related works are studied in the next section, followed by the system modeling with robustness analysis using the Wind Gust Model in Section 3. The different control designs are presented in Section 4, while Section 5 presents the results and discussion. The conclusions are given in Section 6.

2. Related Works

Several research works for quadrotor guidance and navigation have been proposed in the literature in the last few years. We recognize that crafting a high-quality flight design poses a significant challenge for quadrotor UAVs, given their nonlinear characteristics. In enhancing UAV flight performance, numerous researchers have introduced sophisticated methods to stabilize attitude and improve UAV positions and tracking. Several methods for automatic flight control systems have been considered for quadrotor localization, planification, and navigation. The Backstepping controller, which was used recently in [3], is widely used for UAV navigation [4]. An extension of adaptive Backstepping using neural networks is proposed in [5], where a Stable Adaptive Neural Control is proposed. Lee and Tomizuka designed a similar method using an FLC [6]. On the other hand, Pan et al. extended the use of an adaptive Backstepping approach for systems in strict-feedback form based on a multi-layer neural network [7]. Raziye et al. proposed a robust controller based on Backstepping and an Extended Kalman Bucy filter in [8]. Many researchers [9,10] have studied UAV navigation using FLCs. A novel PID FLC for quadrotor navigation using iterative learning control (ILC) was designed for a quadrotor unmanned aerial vehicle (UAV) in [11]. In the same case in [12], a hybrid PID-Fuzzy controller was studied for autonomous UAV stabilization. The Type-2 Fuzzy-PID approach is presented in [13] for the transmission line follower problem through UAVs. Nevertheless, the controllers previously

proposed remain conventional in the face of the inherent complexity of the system and the constraints arising from uncertain parameters. This is where advanced research comes into play, aiming to enhance basic adaptive controllers through novel approaches, particularly by combining and optimizing these control parameters. Various new approaches have been proposed for system stabilization.

In the study referenced in [14], the authors employed a Fuzzy-Gain scheduling mechanism to fine-tune the PID controller for stabilizing both position and altitude. It is imperative for this control strategy to be efficacious, straightforward, and resilient against uncertainties and external disturbances. Reference [15] introduced the application of an Autonomous Quadcopter Trajectory Tracking and Stabilization system using a control system based on Sliding Mode Control and the Kalman Filter. Additionally, the authors of [16] proposed an algorithm for synthesizing an optimal controller to address the mixed H_2/H_∞ control problem for stabilizing aircraft during the glidepath landing mode under uncertainty. For example, in [17], the authors implemented a data-driven neuroendocrine-PID controller for underactuated systems. The notable advantage of this proposed approach lies in its capability to swiftly tune neuroendocrine-PID parameters by measuring the system's input and output data without relying on a mathematical model.

In summary, several works have focused on the trajectory control of quadrotors. Thus, the comparison between numerous control methods makes it possible to demonstrate the robustness of the latter in an uncertain environment. This paper introduces various approaches, which can be classified into the following categories: model-based methods (using a Backstepping controller), model-free methods (employing Fuzzy Logic Controller Type-1 combined with PID and Fuzzy Logic Controller Type-2 Combined with PID), and parameter optimizations using genetic algorithms as a novelty.

3. System Modelling

3.1. Basic Concepts of Quadrotor UAVs

The model of quadrotor illustrated in Figure 2 is taken from [18] and defined as a VTOL with four propellers. Its four propellers are placed at each extremity of a cross structure, where two pairs of propellers turn in opposite directions. The displacement or the rotation motions are performed by adjusting the angular velocity speeds of each rotor. By varying the rotor speeds, it is possible to make vertical lateral and longitudinal translations as well as a rotation about vertical axis. However, the quadrotor UAV has six Degrees of freedom (DOFs), three motions of rotations and translations in Cartesian space, and is controlled by only four inputs, which are thrust force, roll, pitch, and yaw moments.

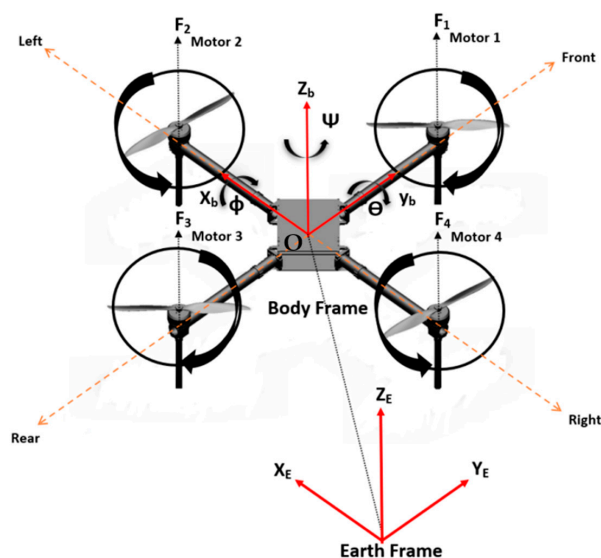


Figure 2. Quadrotor configuration [18].

3.2. Quadrotor Flight Kinematic and Dynamic Model

The quadrotor's kinematics establish a connection between the vehicle's position in the inertial frame and its velocity in the body frame. Meanwhile, the dynamics of the quadrotor describe the relationship between the applied body forces and the consequent accelerations.

3.2.1. Kinematics

For Kinematic modeling, two frames are considered, as shown in Figure 2: the Earth Frame and the Body Frame. The relationship between the Earth Frame and the Body Frame is defined by the use of X, Y, and Z axes in the body-fixed frame, where O represents the center of mass. The X axis points toward rotor 1, the Y axis toward rotor 4, and the Z axis points upwards. The linear position and orientation of the Body Frame relative to the Earth Frame are expressed as three translations (X, Y, Z) and three Euler angles (roll, pitch, yaw- φ , θ , ψ). The linear velocities of the UAV in the body frame are denoted by (U, V, W), while (p , q , r) represent the angular rates with respect to the Body Frame. The rotation matrix between the Earth Frame and Body Frame is provided below (c: cos, s: sin).

$$\mathbf{R} = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\varphi & s\psi s\theta + c\psi s\theta c\varphi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\varphi & -c\psi s\theta + s\psi s\theta c\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix} \quad (1)$$

By using the rotation matrix above, any point on the Body Frame can be expressed in the Earth Frame. An additional transformation matrix is required for the angular rates to transform the angular rates (p, q, r) in the body frame to the angular velocities ($\dot{\varphi}, \dot{\theta}, \dot{\psi}$) in the inertial frame:

$$\mathbf{R}_r = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \varphi & \sin \varphi \cos \theta \\ 0 & -\sin \varphi & \cos \varphi \cos \theta \end{bmatrix} \quad (2)$$

3.2.2. Dynamics Model

In this research, the quadrotor model is regarded as a rigid body with a concentrated point mass, featuring four rotors symmetrically positioned around its center of mass [19]. The control architecture of quadrotor is given by Figure 3. By employing Newton's equations of translational and rotational motion, we can derive a set of six equations, as described below [1,20]:

$$\begin{aligned} \ddot{x} &= (\sin\psi \sin\varphi + \cos\psi \sin\theta \cos\varphi) \frac{F_z}{m} + \frac{D_x}{m} \\ \ddot{y} &= (-\cos\psi \sin\varphi + \sin\psi \sin\theta \cos\varphi) \frac{F_z}{m} + \frac{D_y}{m} \\ \ddot{z} &= -g + (\cos\theta \cos\varphi) \frac{F_z}{m} + \frac{D_z}{m} \\ \ddot{\varphi} &= \frac{(I_{yy} - I_{zz})\dot{\theta}\dot{\psi} - J_r\dot{\theta}\omega_r + \tau_x}{I_{xx}} \\ \ddot{\theta} &= \frac{(I_{zz} - I_{xx})\dot{\varphi}\dot{\psi} + J_r\dot{\varphi}\omega_r + \tau_y}{I_{yy}} \\ \ddot{\psi} &= \frac{(I_{xx} - I_{yy})\dot{\varphi}\dot{\theta} + \tau_z}{I_{zz}} \end{aligned} \quad (3)$$

where ' m ' signifies the mass of the quadrotor, and I_{xx} , I_{yy} , and I_{zz} denote the inertial components along the x , y , and z directions in frame B. The term F_z represents the vertical thrust along the Z-axis. τ_x , τ_y , and τ_z represent the torques associated with the thrust difference of each rotor pair. Additionally, D_x , D_y , and D_z indicate the drag forces corresponding to velocities in the X, Y, and Z directions. J_r shows the rotor inertia, and ω_r is the overall speed of the rotor, as defined below:

$$\omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4 \quad (4)$$

The quadrotor input controls are given by

$$\begin{aligned} u_1 &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 &= b(\omega_4^2 - \omega_2^2) \\ u_3 &= b(\omega_3^2 - \omega_1^2) \\ u_4 &= b(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{aligned} \quad (5)$$

The relations between F_z , τ_x , τ_y , τ_z and each rotor's angular speeds can be obtained by analyzing the free body diagram of the model in Figure 2. The terms ω_1 , ω_2 , ω_3 , and ω_4 represent the angular speeds of the four rotors, b represents thrust factor, d represents drag factor, and l represents the length of each quadrotor's arm. The following matrix shows the relationships between F_z , τ_x , τ_y , τ_z and the angular velocities of the four propellers:

$$\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (6)$$

The dynamic model has the potential to be simplified and expressed as a set of nonlinear dynamic equations, which can be characterized as follows:

$$\ddot{X} = (X) + g(X)u \quad (7)$$

where u and $X \in R^4$ are, respectively, the input and state vector, given as follows:

$$u = [u_1 u_2 u_3 u_4]^T \quad (8)$$

$$X = [X_1 X_2 X_3 X_4]^T = [z \varphi \theta \psi]^T \quad (9)$$

The matrices representing the nonlinear dynamic function, denoted as $f(X)$, and the nonlinear control function, denoted as $g(X)$, can be written as follows:

$$\begin{aligned} f(X) &= \begin{pmatrix} -g \\ \dot{\theta}\dot{\psi}a_1 - \theta a_2 \Omega_d \\ \dot{\theta}\dot{\psi}a_3 + \dot{\varphi}a_4 \Omega_d \\ \dot{\theta}\dot{\varphi}a_5 \end{pmatrix} \\ g(X) &= \begin{pmatrix} u_z \frac{1}{m} & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 \\ 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & b_3 \end{pmatrix} \end{aligned} \quad (10)$$

with the following abbreviations:

$$\begin{aligned} a_1 &= (I_{yy} - I_{zz} / I_{xx}), \quad a_2 = J_r / I_{xx}, \quad a_3 = (I_{zz} - I_{xx}) / I_{yy}, \quad a_4 = J_r / I_{yy}, \\ a_5 &= (I_{xx} - I_{yy} / I_{zz}), \\ b_1 &= l / I_{xx}, \quad b_2 = l / I_{yy}, \quad b_3 = l / I_{zz}, \quad u_z = (\cos \varphi \cos \theta). \end{aligned}$$

Given the well-known values of all system parameters, the nominal representation of nonlinear systems can be articulated as follows:

$$\ddot{X} = f_0(X) + g_0(X)u \quad (11)$$

where f_0 and g_0 are the nominal value of f and g , respectively.

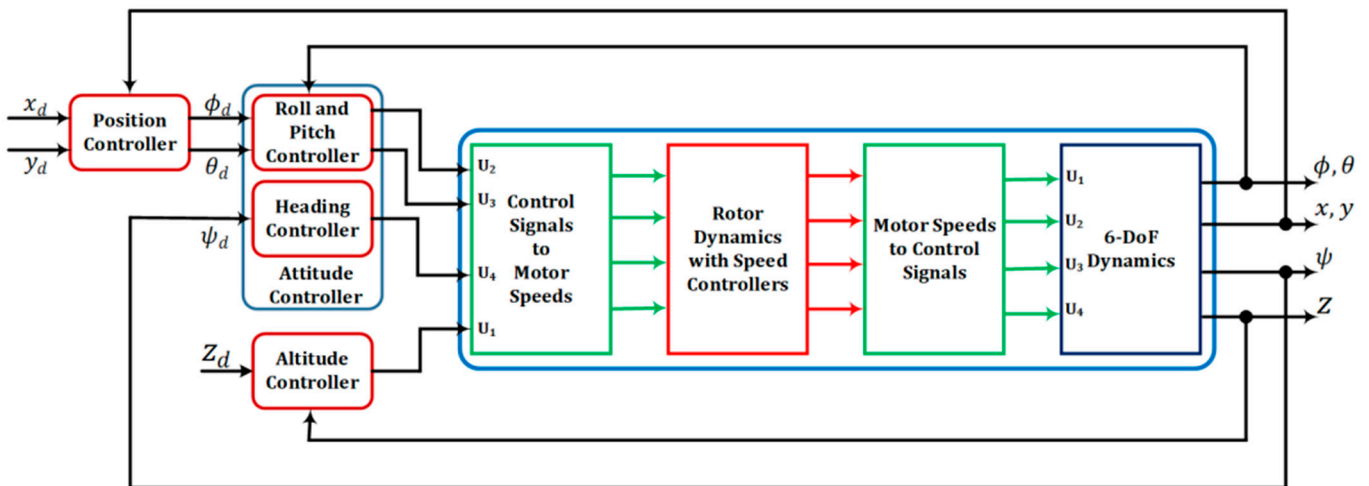


Figure 3. Control architecture of the quadrotor [21].

In our case, we are interested in the uncertainties in thrust and lift coefficients, as well as inertial moments, which pose challenges in accurately characterizing the behavior of quadrotors. These uncertainties can arise due to various factors. For thrust coefficients, changes in environmental conditions, such as temperature and air density, can affect thrust production.

In addition, environmental factors like wind gusts and turbulence can introduce variations in lift coefficients during flight, and surface contamination or damage to the airframe can impact lift generation.

The inertial moments depend on the distribution of mass within the quadrotor. Irregularities or asymmetry in mass distribution can lead to variations in the moments of inertia. In windy conditions, changes in the distribution of lift and drag forces on the vehicle may affect its rotational behavior. Inertial moments manifest in different axes, corresponding to yaw, pitch, and roll motions. Wind-induced moments can lead to changes in the quadrotor's orientation and rotation around these axes, affecting its overall stability.

3.3. Robustness Analysis Using Wind Gust Model

Wind gusts are intricate physical phenomena commonly represented using either deterministic models [22] or stochastic models [23]. The latter models assume that turbulence behaves as a stationary Gaussian random process, which is a prevalent assumption. The stationary nature implies that turbulence extends infinitely in duration, while the Gaussian process pertains to the probability of encountering a specific gust velocity at a given time. A stochastic approach employed in modeling wind gusts is the Power Spectral Density (PSD). In the PSD atmospheric turbulence model, it is presumed that the turbulence's intensity is significantly influenced by various factors and can change due to weather conditions, flight altitude, and temperature gradients [24]. Wind gust signals are generated by passing white noise through a shaping filter. In the literature, two primary shaping filters can be identified: the Dryden filter and the Von Karman filter. Given its simpler form, we opt to use the Dryden filter in this study. The filters employed to generate the Dryden spectral model are provided in [25,26], as follows:

$$\begin{aligned}
 H_u(s) &= \frac{\Delta u}{N_w} = \sigma_u \sqrt{\frac{2L_u}{\pi v}} \frac{1}{1 + \frac{L_u}{v} s} \\
 H_v(s) &= \frac{\Delta v}{N_w} = \sigma_v \sqrt{\frac{L_v}{\pi v}} \frac{1 + \frac{\sqrt{3}L_v}{v} s}{\left(1 + \frac{L_v}{v} s\right)^2} \\
 H_w(s) &= \frac{\Delta w}{N_w} = \sigma_w \sqrt{\frac{L_w}{\pi v}} \frac{1 + \frac{\sqrt{3}L_w}{v} s}{\left(1 + \frac{L_w}{v} s\right)^2}
 \end{aligned} \tag{12}$$

where N_w represents white noise, v signifies the relative velocity of the UAV quadrotor concerning the airflow, and $[\Delta u, \Delta v, \Delta w]^T$ represents alterations in body linear velocities due to wind gusts. The turbulence intensities ($\sigma_u, \sigma_v, \sigma_w$) and the turbulence scale lengths (L_u, L_v, L_w) provide descriptions of the wind gust characteristics. In regions with low altitude (where the altitude is less than 1000 feet), the turbulence scale lengths and intensities are defined as follows:

$$L_w = h, L_u = L_v = \frac{h}{(0.177 + 0.000823h)^{1.2}} \quad (13)$$

$$\sigma_w = 0.1w_{20}, \frac{\sigma_u}{w} = \frac{\sigma_v}{w} = \frac{1}{(0.177 + 0.000823h)^{0.4}} \quad (14)$$

In this paper, we consider a standard wind speed of 2.4 m/s, which corresponds to Beaufort Scale 3, and an altitude of 2 m. Here, h represents the height above the ground, and W_{20} denotes the wind speed as per the Beaufort Scale.

4. Control Design

In this paper, two categories of controllers are investigated: model-based controllers and FLCs. For the former, a Backstepping controller is implemented and validated. For the latter, two FLCs are considered: Type 1 and Type 2. The control approaches with and without the model are implemented and validated on a trajectory tracking problem using a quadrotor. Figure 4 shows the global model proposed controllers' system for a quadrotor.

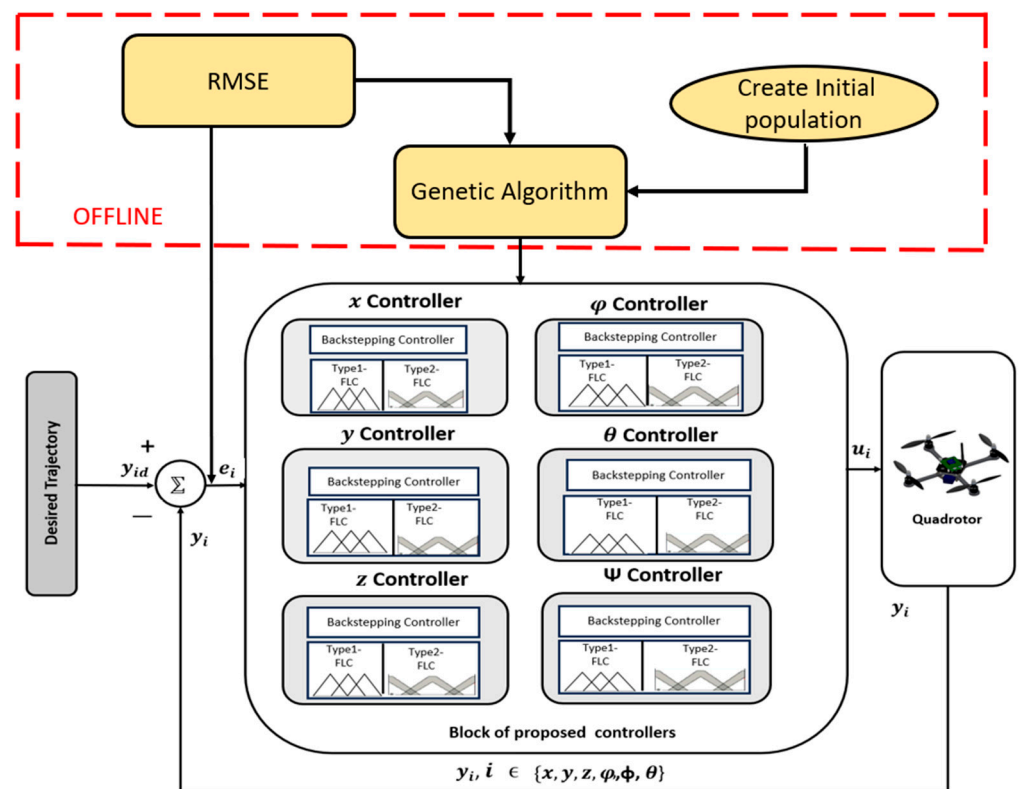


Figure 4. Global model proposed for controller system for quadrotor.

4.1. Backstepping Controller

The control objective is to develop a suitable control law for the system [8], allowing the state vector X of the quadrotor system to adhere to a predetermined reference trajectory vector X_d . The following section provides an overview of the Backstepping control approach designed for the quadrotor system:

Step 1: First, the tracking error is defined.

$$e_1 = X_d - X \quad (15)$$

where X_d is a desired trajectory specified by a reference model. Then, the derivative of the tracking error can be represented as

$$\dot{e}_1 = \dot{X}_d - \dot{X} \quad (16)$$

The first Lyapunov function is chosen as

$$V_1(e_1) = \frac{1}{2}e_1^T e_1 \quad (17)$$

The derivative of V_1 is

$$\dot{V}_1(e_1) = e_1^T \dot{e}_1 = e_1^T (\dot{X}_d - \dot{X}) \quad (18)$$

where \dot{X} can be viewed as a virtual control. The desired value of virtual control α , known as a stabilizing function, can be defined as follows:

$$\alpha = \dot{X}_d + k_1 e_1 \quad (19)$$

where k_1 is a positive constant. By substituting the virtual control by its desired value from Equation (17), we obtain

$$\dot{V}_1(e_1) = -k_1 e_1^T e_1 \leq 0 \quad (20)$$

Step 2: The deviation of the virtual control from its desired value can be defined as

$$e_2 = \dot{X} - \alpha = \dot{X} - \dot{X}_d - k_1 e_1 \quad (21)$$

The derivative of e_2 is expressed as

$$\dot{e}_2 = \ddot{X} - \dot{\alpha} = f_0(X) + g_0(X)u + L - \ddot{X}_d - k_1 \dot{e}_1 \quad (22)$$

The second Lyapunov function is

$$V_2(e_1, e_2) = \frac{1}{2}e_1^T e_1 + \frac{1}{2}e_2^T e_2 \quad (23)$$

The derivative equations of (23) are defined as

$$\begin{aligned} \dot{V}_2(e_1, e_2) &= e_1^T \dot{e}_1 + e_2^T \dot{e}_2 = e_1^T (\dot{X}_d - \dot{X}) + e_2^T (\ddot{X} - \dot{\alpha}) = e_1^T (-e_2 - k_1 e_1) + e_2^T (f_0(X) + g_0(X)u + L - \ddot{X}_d - k_1 \dot{e}_1) = \\ &= -k_1 e_1^T e_1 + e_2^T (-e_1 + f_0(X) + g_0(X)u + L - \ddot{X}_d - k_1 \dot{e}_1) \end{aligned} \quad (24)$$

Step 3: Since the system dynamics and the external disturbance are well known, and $g_0(X) \neq 0$, an ideal Backstepping can be obtained as

$$u = g_0(X)^{-1} (e_1 + k_1 \dot{e}_1 + \ddot{X}_d - f_0(X) - L - k_2 e_2) \quad (25)$$

where k_2 is a positive constant, and the term $k_2 e_2$ is added to stabilize the tracking error. Substituting (25) into (24), the following equation can be obtained:

$$\dot{V}_2(e_1, e_2) = -k_1 e_1^T e_1 - k_2 e_2^T e_2 \quad (26)$$

where $\dot{V}_2(e_1, e_2) \leq 0$, $V_2(e_1, e_2)$ is a negative semi-definite. So, the Backstepping controller in (25) will stabilize the system.

4.2. Type-1 Fuzzy Logic Controller

Fuzzy logic offers straightforward computational capabilities and is widely regarded as one of the most adaptable controllers. In this paper, first, a Takagi Sugeno Type-1 FLC controller is proposed for quadrotor trajectory tracking, as shown in Figure 5.

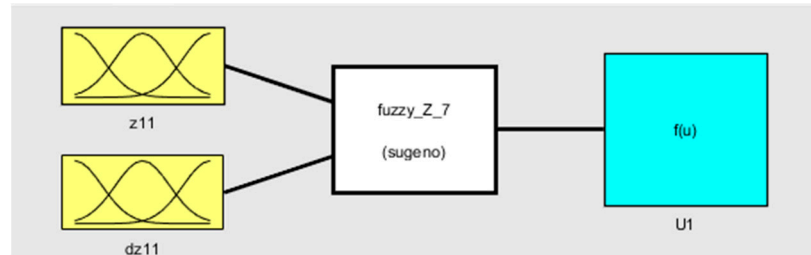


Figure 5. Type-1 fuzzy logic controller.

This paper introduces and compares four different variations for controlling roll, pitch, yaw, and height through simulation and careful observation.

The membership functions in Figure 6 are tuned based on the range sensors and the authors’ experience. Increasing the number of membership functions can provide good accuracy; however, the computational time will increase. Then, the shape and number of membership functions should be selected to maintain a suitable trade-off between precision, robustness, and computational time. Table 1, shows the quadrotor rule base.

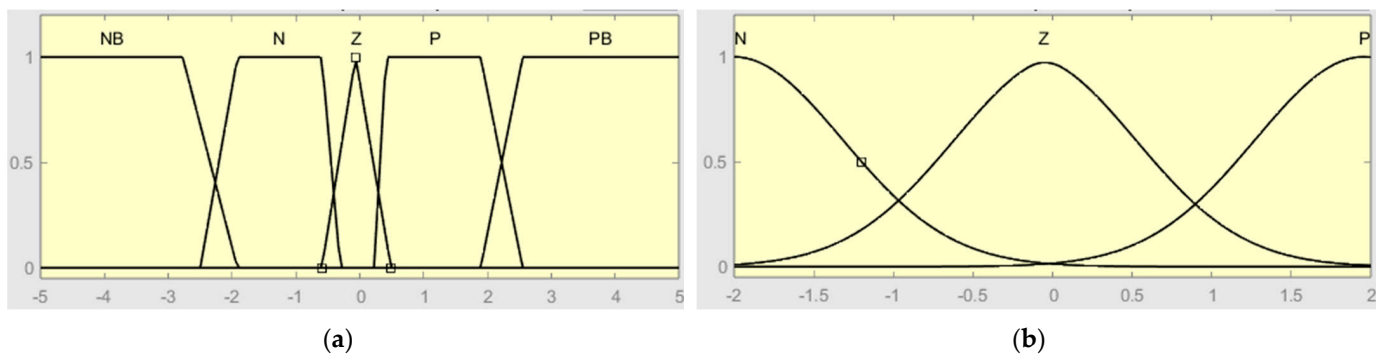


Figure 6. Membership function of Type-1-FLC: (a) first input (e); (b) second input (de).

Table 1. Quadrotor rule base.

		Height				
		NB	N	Z	P	PB
de	e					
	N	GUM	GD	GD	S	GU
	Z	GDM	GD	S	GU	GUM
P	GD	S	GU	GUM	GUM	

Where: N: Negative, Z: Zero, P: Positive, GUM: Go Up Much, GU: Go Up, S: Stand, GDM: Go Down Much, GD: Go Down, NB: Negative Big, PB: Positive Big.

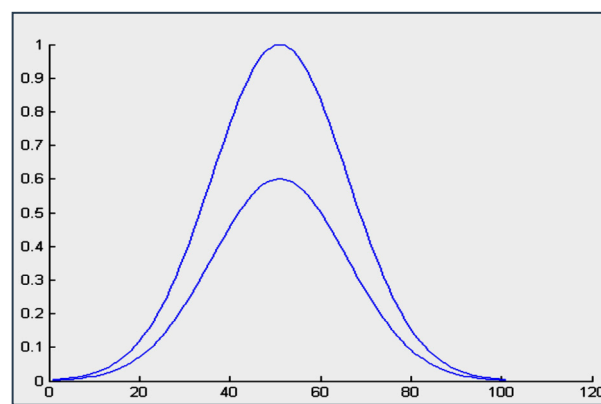
For quadrotor control, the triangular, trapezoid, and Gaussian membership functions are used. The input range is $[-2, 2]$, whereas the output variable lies in the range of $[-15, 15]$. The membership for each controller is as shown in Figure 6a,b. Table 2 presents the parameters of output.

Table 2. Parameters for output fuzzy controller.

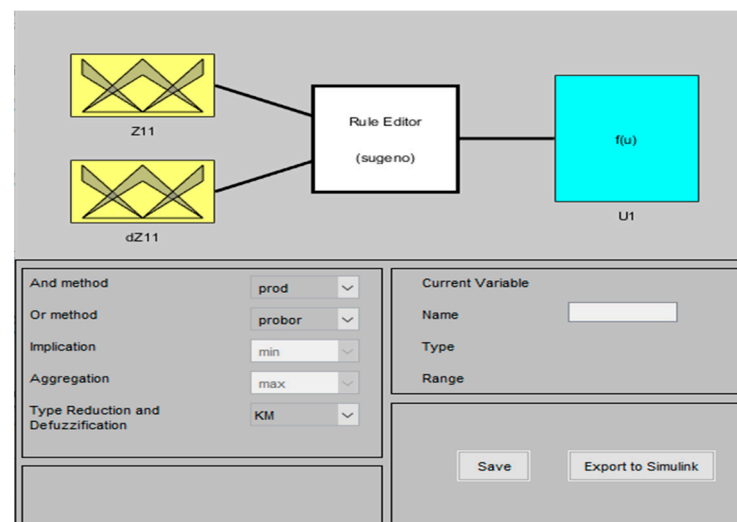
Variable	GDM	GD	S	GU	GUM
Output	−10	−8	0	8	10

4.3. Type-2 Fuzzy Logic Controller

A Type-2 fuzzy logic system or controller shares many fundamental concepts with a Type-1 FLC. In fact, a Type-2 fuzzy logic system closely resembles a Type-1 system in terms of membership functions, fuzzy rules, fuzzification, inference, and defuzzification [27]. There are only two key distinctions: firstly, in the Type-2 FLC, the membership functions are three-dimensional, as depicted in Figure 7. This third dimension represents the value of the membership function at each point within its two-dimensional domain, known as the footprint of uncertainty (FOU) [28]. Secondly, the Type-2 FLC necessitates an additional step involving type reduction.

**Figure 7.** Model of membership function for Type-2 FLC.

To create a Type-2 FLC, we expanded upon the initially suggested Type-1 controller by leveraging the IT2-FLS v1.1 Matlab/Simulink Toolbox (Figure 8). The inputs, rule base, and outputs closely mirror those of the Type-1 FLC discussed in Section 4.2.

**Figure 8.** Type-2 fuzzy logic controller.

The membership functions of the Type-2 fuzzy logic controller inputs are represented below (Figure 9a,b):

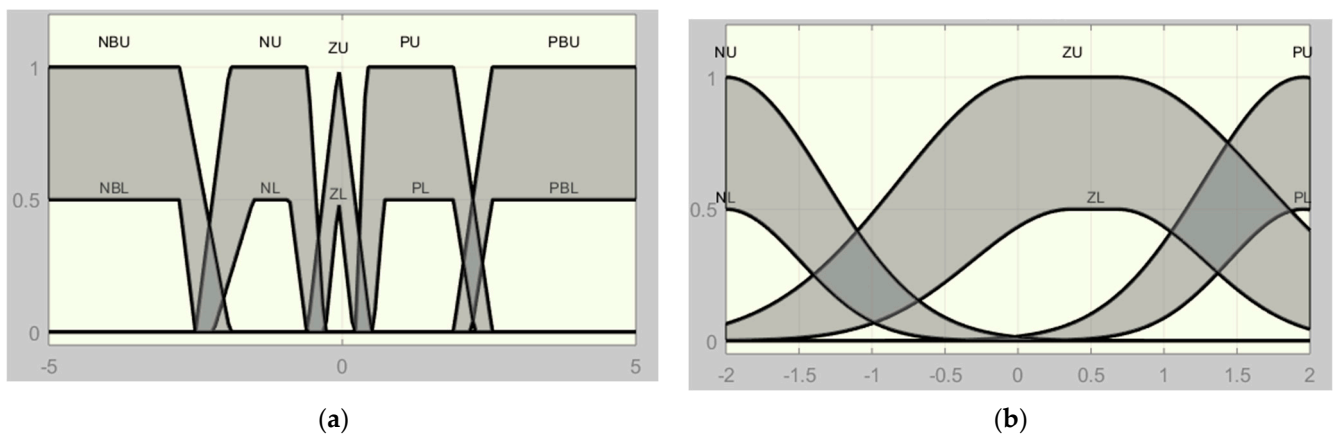


Figure 9. Membership function of Type-2-FLC: (a) first input (e); (b) second input (de).

Similar to the approach taken for the Type-1 FLC, we propose a set of 15 IF-THEN fuzzy rules for the Type-2 FLC. The output processing block utilizes the Takagi–Sugeno “SOM/PROD” inference method and adopts the “NT” type for the reduction and defuzzification process.

4.4. PID-Fuzzy Logic Controller

The main drawback of the FLC approach is the tuning of membership parameters of input and output when a significant number of rules is considered. Using human expertise is efficient for a few number of rules. However, when the number of rules increases, it becomes very difficult to tune the FLC; thus, for the solution, we propose to use a PID-FLC optimized by a genetic algorithm.

4.5. PID-FLC with GA Tuning

In the figure below (Figure 10), an overall summary diagram explains the principal of the optimal PID-Type-1 FLC controller and PID-Type-2 FLC controller based on GA parameter optimization. Red arrow indicates that the PID parameters are optimized by GA.

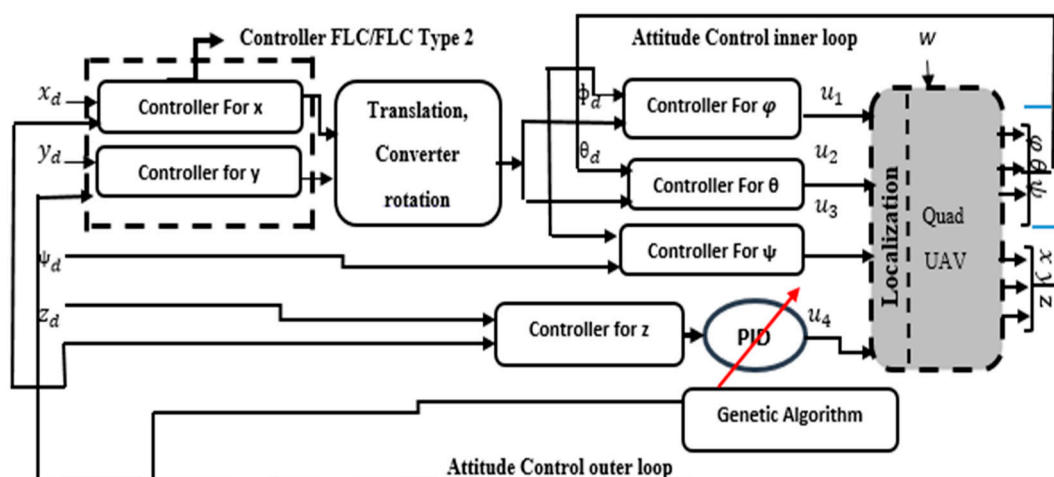


Figure 10. Quadrotor control scheme with GA optimization.

4.6. Optimal Type-1 and Type-2 FLC with PID Controller

The primary benefit of the FLC is its compatibility with a conventional PID controller. As the name implies, this control technique combines aspects of two control methods: fuzzy logic and PID [2].

The PID controller finds extensive application across various domains. Its merits include the elimination of steady-state errors, overshooting, reduction in settling time, and enhancement of system stability. The mathematical equation for the PID controller is presented below:

$$u = K_p e + K_i \int_0^t e dt + K_d \frac{d}{dt} e \quad (27)$$

To achieve the desired output in the control of the quadrotor, a combination of three PD controllers and one PID controller is employed. Below, we provide a description of each controller:

$$\begin{aligned} u_1 &= K_p(\varphi_d - \varphi) + K_d(\dot{\varphi}_d - \dot{\varphi}) \\ u_2 &= K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta}) \\ u_3 &= K_p(\psi_d - \psi) + K_d(\dot{\psi}_d - \dot{\psi}) \\ u_4 &= K_p(z_d - z) + K_i \int_0^t z + K_d(\dot{z}_d - \dot{z}) \end{aligned} \quad (28)$$

where $u_1, u_2, u_3,$ and u_4 are the control inputs and $K_p, K_d,$ and K_i are, respectively, the proportional gain, the derivative gain, and the integral gain.

4.7. Genetic Algorithm

4.7.1. Definition and Schemes

GAs are search processes that operate according to the principles of natural selection and genetics. A basic GA encompasses three key operations: Selection, Genetic Operations, and Replacement, as illustrated in the figure below, representing a typical GA cycle [25].

Genetic algorithms involve the evolutionary cycle of a series of genes (Figure 11), referred to as a chromosome; this symbolizes a potential solution to the problem at hand. Each gene within the chromosome corresponds to a component of the solution pattern. The prevalent method for representing a solution as a chromosome involves employing a string of binary digits, where each bit in the string acts as a gene. The transformation of the solution into the binary bit string is termed coding. The choice of a particular coding scheme is application-dependent. Subsequently, the solution bit strings undergo decoding to facilitate their evaluation through a fitness measure [29].

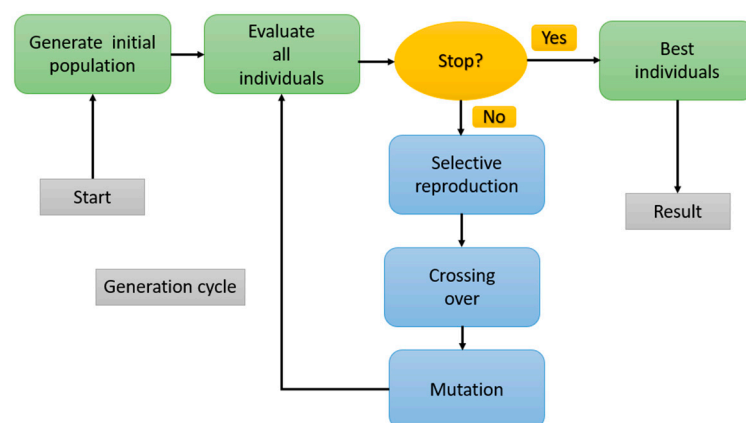


Figure 11. Generic algorithm cycle.

The motivation for employing a genetic algorithm in the control system of a quadrotor, in particular for the optimization of PID and FLC parameters, came from the fact that the latter exhibit complex, non-linear dynamics, making it difficult to determine optimal control parameters analytically [30]. In addition, GAs are highly effective in handling complex non-linear optimization problems, without requiring a detailed understanding of the system's mathematical model; they are particularly robust in addressing uncertainties and adapt well to variations, providing a means to optimize control parameters for different operating conditions. On the other hand, PID and FLC controllers often involve setting

several parameters to balance different control objectives, such as stability, responsiveness, and robustness. Thus, GAs support multi-objective optimization, enabling the exploration of a solution space that optimally balances these conflicting objectives. In addition, GAs are suitable for offline optimization, where the algorithm can iteratively explore the parameter space without the need for real-time computation [31].

There are many other optimization algorithms that are successfully applied to tune FLC parameters, such as the hybrid spiral bacteria search algorithm [32], the hybrid gray whale optimization approach [33], and many others. These approaches are good in terms of computation time, which is not really important in our case.

4.7.2. Advantages and Disadvantages of Using GAs

The following table (Table 3) summarizes the advantages and disadvantages of using genetic algorithms in the control system optimization of a quadrotor as compared with stated algorithms such as the hybrid spiral-bacterial foraging algorithm and hybrid Grey Wolf-Whale optimization approach.

Table 3. Comparison of GA and other optimization approaches.

Criteria	Genetic Algorithm	Hybrid Spiral-Bacterial Foraging	Hybrid Grey Wolf-Whale Optimization
Advantages			
Global Search Capability	✓	✓	✓
Adaptability to Complex Systems	✓	✓	✓
Multi-Objective Optimization	✓	✓	✓
Empirical Optimization	✓	✓	✓
Offline Optimization	✓	✓	✓
Solution Diversity	✓	✓	✓
Disadvantages			
Computational Intensity	✓	✓ (Varies)	✓ (Varies)
Parameter Tuning	✓	✓ (Varies)	✓ (Varies)
No Guarantee for Global Optimality	✓	✓	✓
Real-Time Applicability for FLC	✓	✓	✓

Note that the checkmarks (✓) indicate whether the respective algorithm has an advantage or disadvantage for a specific criterion. The "Varies" notation indicates that the performance may depend on specific implementations or problem instances.

4.8. Optimization Strategy of Using GA

Efficiency in terms of the noisy or stochastic objective function is, in this paper, achieved using the proposed PID-Type-1 FLC as well as the PID-Type-2 FLC, automatically tuned using the genetic algorithm. The PID-Type-1 FLC and PID-Type-2 FLC with GA optimization are shown in Figure 12.

As can be seen from previous PID FLC controllers, including the Z-Controller, φ -Controller, θ -Controller, and ψ -Controller, sixteen (16) parameters should be tuned. For this purpose, a Genetic Algorithm (GA) was used to optimize these parameters in order to minimize the Root Mean Square Error (RMSE) of the trajectory tracking using the quadrotor. In this case, the fitness function can be given by

$$f(K) = \sqrt{(X_d - X)^2 + (Y_d - Y)^2 + (Z_d - Z)^2} \quad (29)$$

where $K = [K1, K2, K3, \dots, K16]$.

The optimization problem is solved with a genetic algorithm:

$$\underset{K = 1 : 12}{\text{Argmin}} \sqrt{(X_d - X)^2 + (Y_d - Y)^2 + (Z_d - Z)^2} \quad (30)$$

GA optimization steps are given in the diagram below (Figure 13):

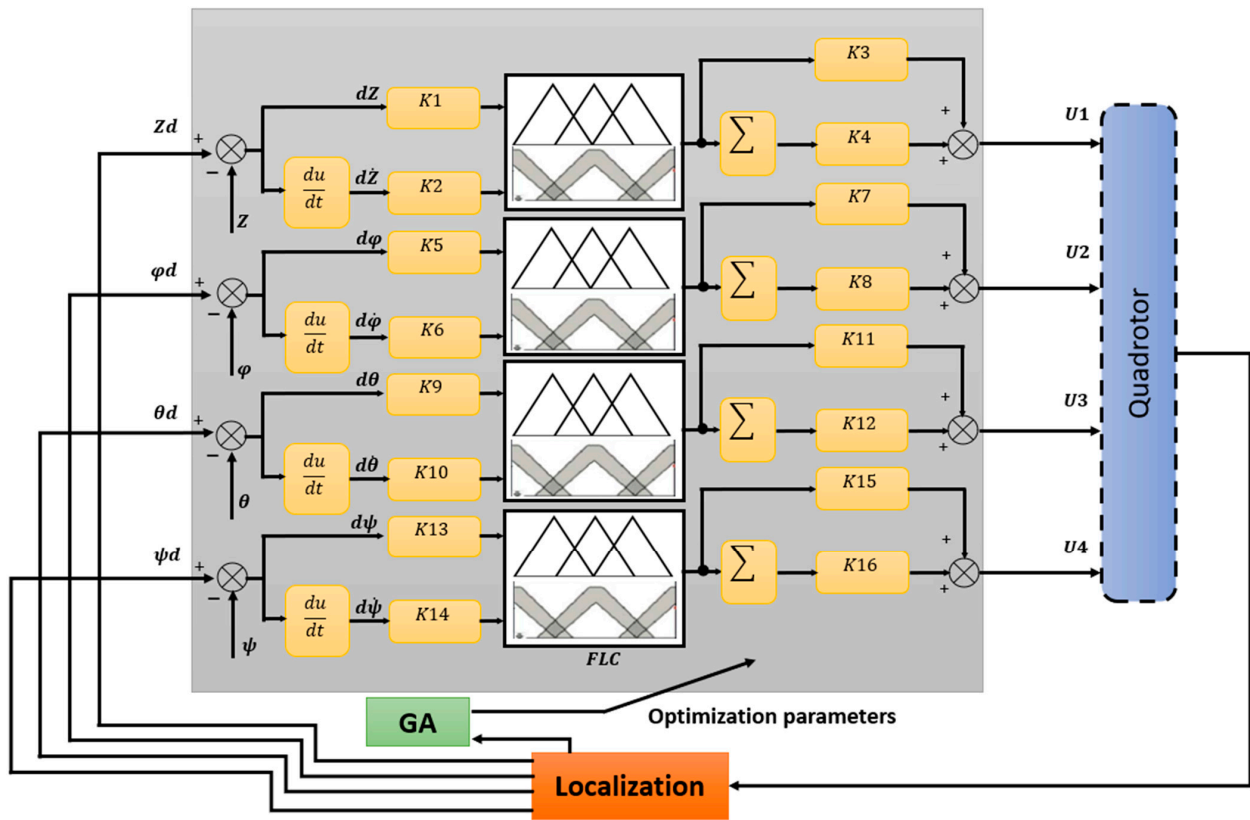


Figure 12. Architecture of optimization strategy for Type-1 FLC and Type-2 FLC using GA.

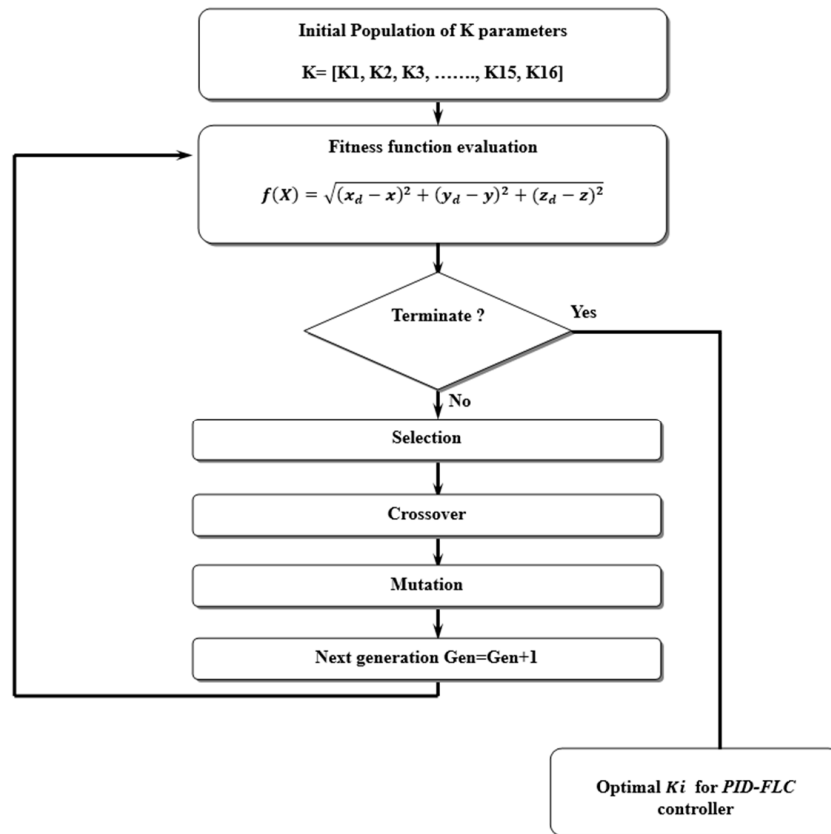


Figure 13. GA optimization step diagram.

5. Results and Discussion

To evaluate the performances of the proposed controllers, many simulation scenarios are considered. The quadrotor parameters used for simulation are taken from [1] and are listed in Table 4.

Table 4. Description of the quadrotor parameters [1].

Symbol	Description	Value
I_x	Moment of inertia X axis	0.007 kgm ²
I_y	Moment of inertia Y axis	0.007 kgm ²
I_z	Moment of inertia Z axis	0.012 kgm ²
J_r	Rotor moment of inertia	6.5×10^{-5} kgm ²
b	Thrust factor	4.13×10^{-5} Ns ²
d	Drag factor	8.5×10^{-7} Nms ²
l	Distance to the center of the quadrotor	0.17 m
m	Masse of quadrotor	0.68 kg
g	Gravitation constant	9.81 m/s ²

The proposed algorithms are validated using a trajectory tracking scenario (Figure 14). The quadrotor (shown in red) should accurately follow the predefined trajectory. Three controllers are implemented, validated, and compared using realistic scenarios.

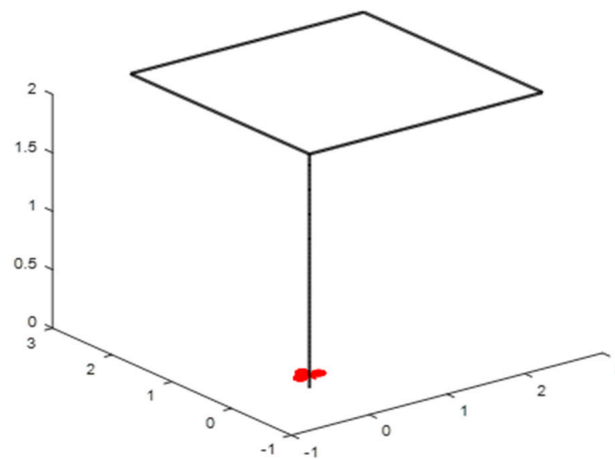


Figure 14. Trajectory of simulation.

5.1. Path Tracking for Quadrotor Using Backstepping Controller

The first step covers quadrotor trajectory tracking; the vehicle is initialized and then takes off from the initial position. The Backstepping controller is generally capable of piloting the quadrotor to the trajectory reference and stabilizing the altitude of the vehicle in a few seconds after takeoff, as shown in the figures below (Figures 15 and 16).

The obtained results of the quadrotor trajectory tracking using the Backstepping controller are illustrated in Figures 15–18. According to the obtained results, the UAV accurately follows the desired trajectory (see Section 5.4). This result is confirmed by Figures 17 and 18, where very small errors following x , y , z and θ , φ , ψ are obtained. Figures 15 and 16 illustrate the evolution of the velocity and input command of the quadrotor, respectively.

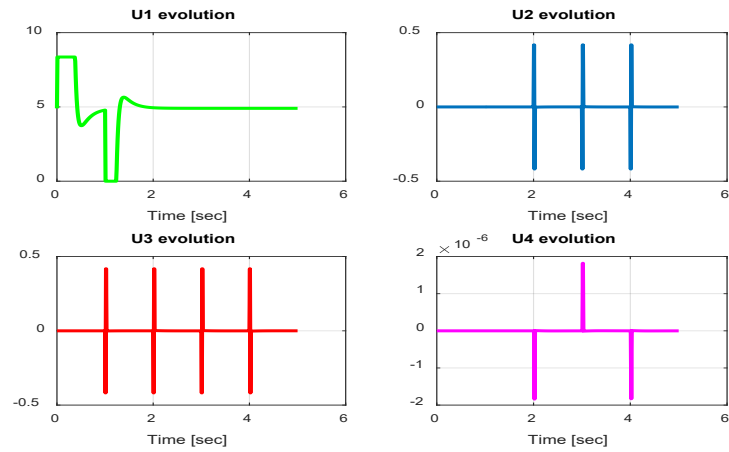


Figure 15. Quadrotor commands of Backstepping control.

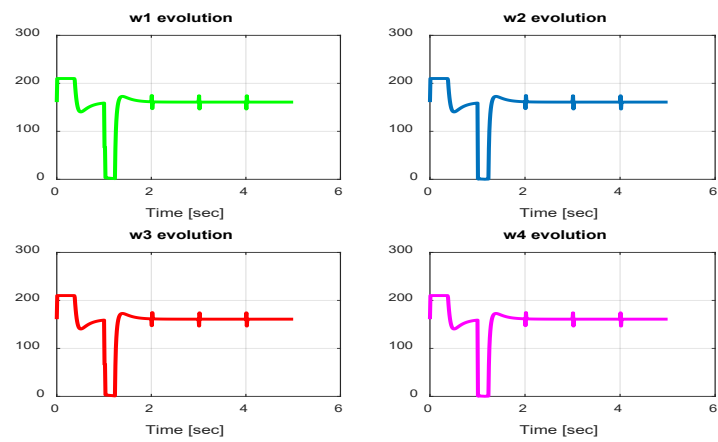


Figure 16. Motor velocities of Backstepping control.

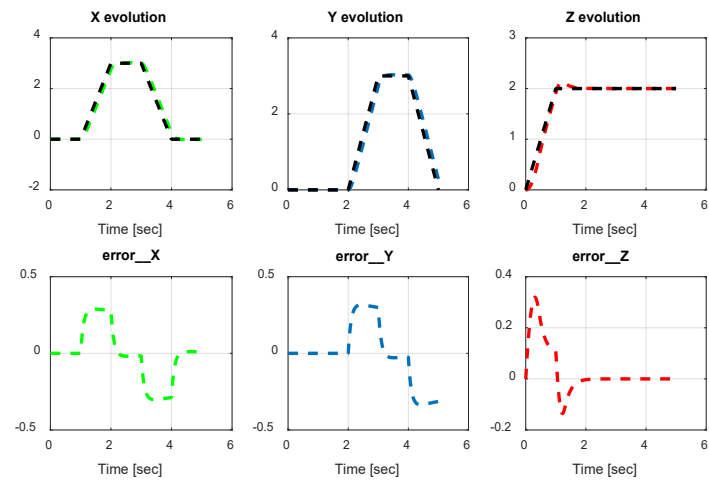


Figure 17. x, y, z errors evolution of Backstepping control.

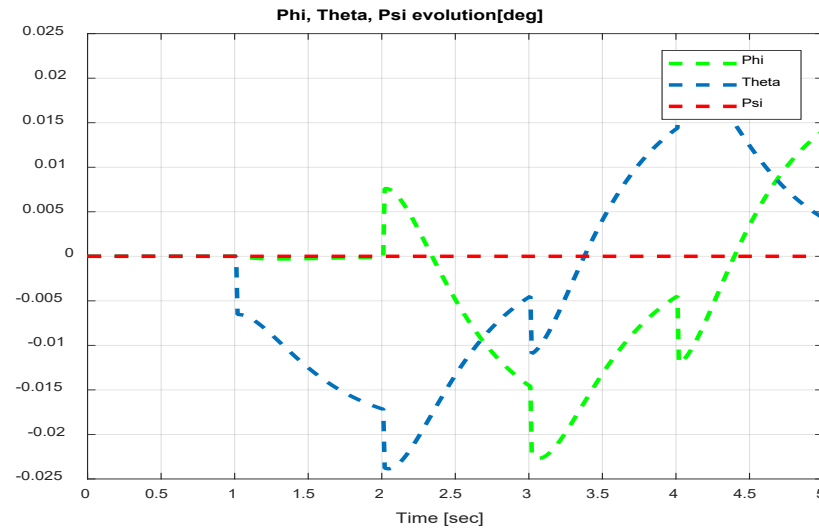


Figure 18. Quadrotor angles of Backstepping control.

5.2. Path Tracking for Quadrotor Using Type-1 Fuzzy Logic Controller

The results of trajectory tracking using the Type-1 FLC are presented in this section. The first test illustrates the accurate tracking of the trajectory. The UAV is tested with different initial positions before being piloted autonomously by simulation. In addition, the quadrotor is tested using a trajectory that has a turn and a climb, with various initial positions, to observe the ability to follow a given trajectory. The figures below show the obtained results.

The input response and velocity evolution provided by the Type-1 FLC are illustrated in Figures 19 and 20, respectively. According to the results illustrated above for quadrotor navigation using the Type-1 FLC, the quadrotor accurately follows the desired trajectory (see Section 5.4). These results are confirmed by the small value of error given in Figures 21 and 22.

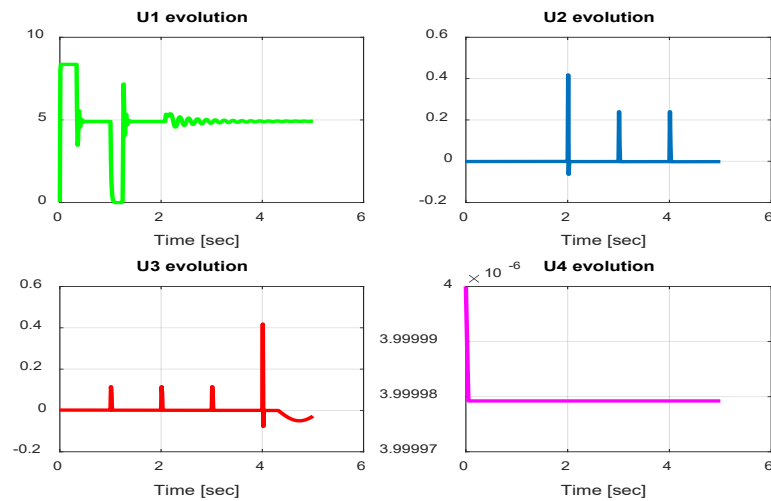


Figure 19. Quadrotor commands of Type-1 FLC.

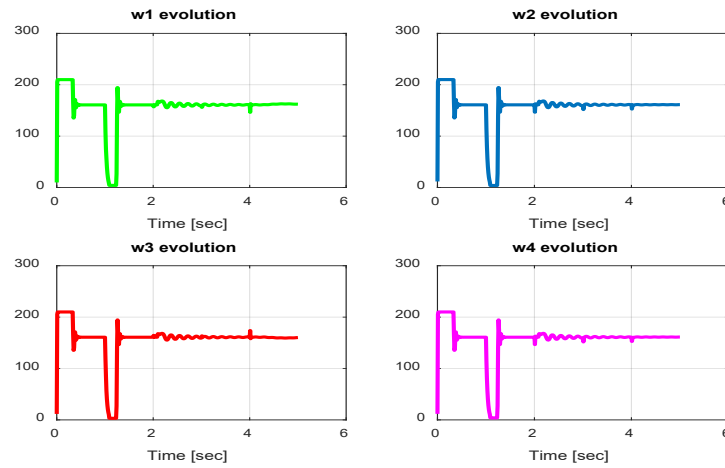


Figure 20. Motor velocities of Type-1 FLC.

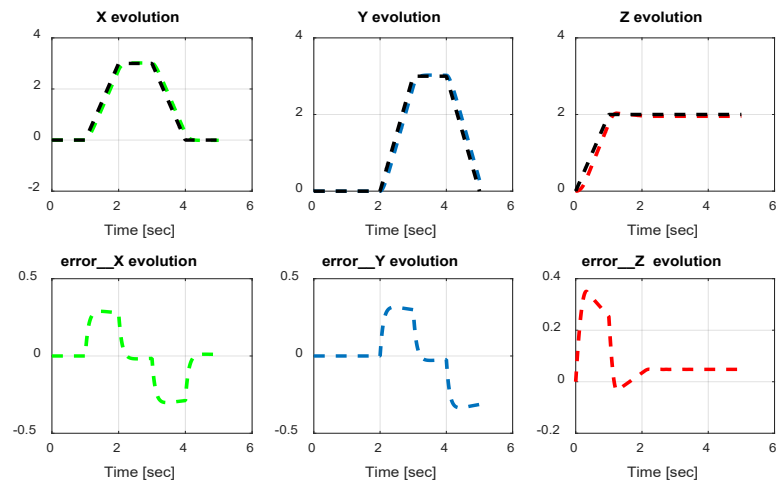


Figure 21. x , y , z error evolution of Type-1 FLC.

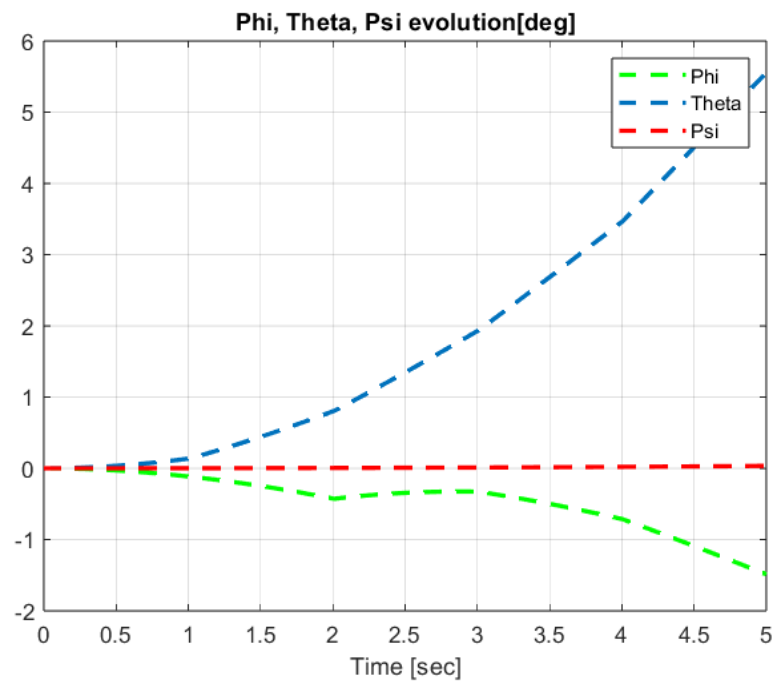


Figure 22. Quadrotor angles of Type-1 FLC.

5.3. Path Tracking for Quadrotor Using Type-2 Fuzzy Logic Controller

In this section, we present simulation results to evaluate the Type-2 FLC in a trajectory tracking scenario. The proposed controller is validated according to the desired path navigation problem; the Type-2 FLC shows good performance even with the complexity of the quadrotor parameters, as can be seen in Figures 23–27.

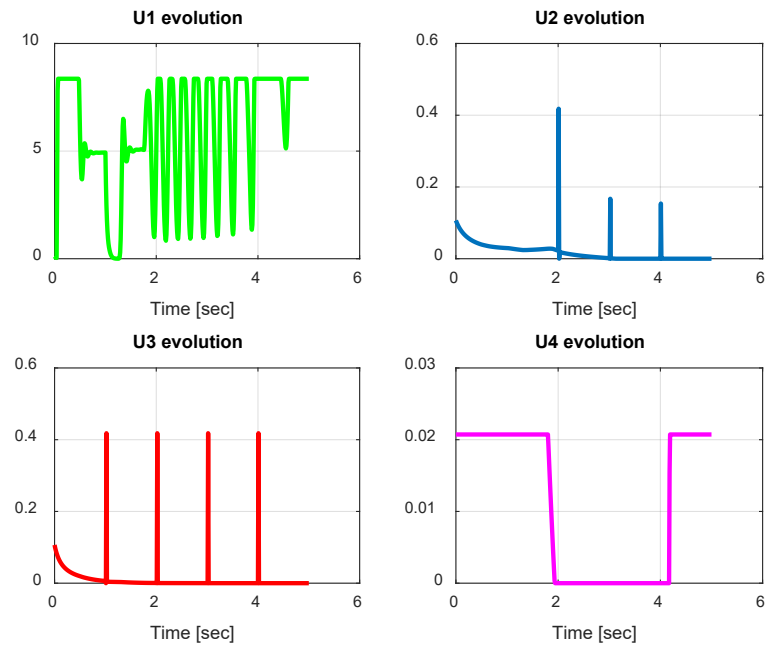


Figure 23. Quadrotor commands of Type-2 FLC.

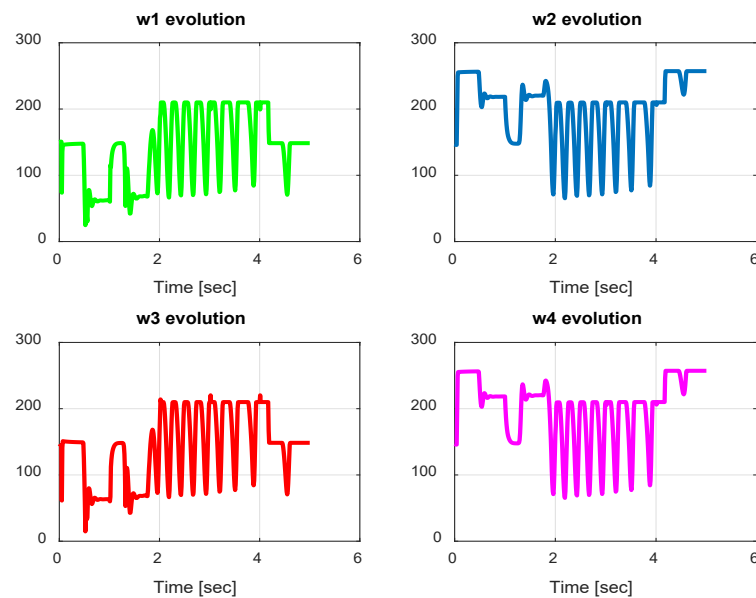


Figure 24. Motor velocities of Type-2 FLC.

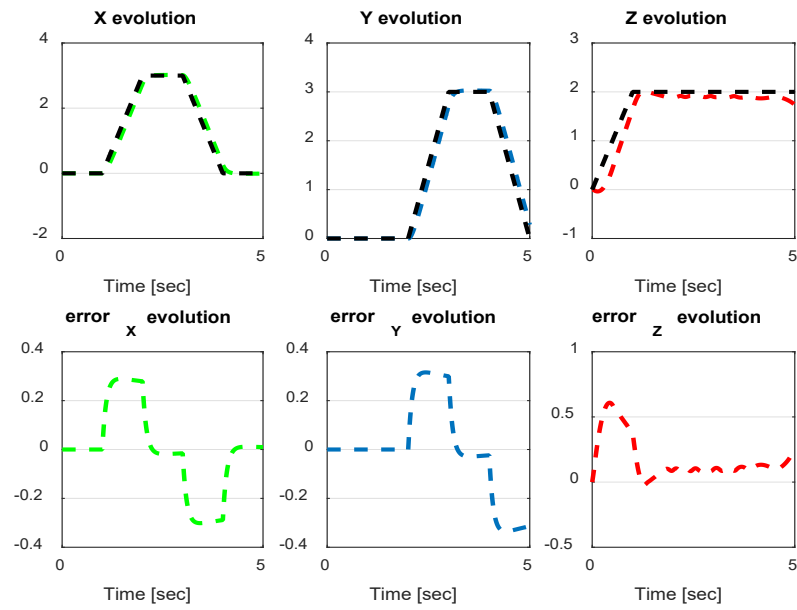


Figure 25. x, y, z error evolution of Type-2 FLC.

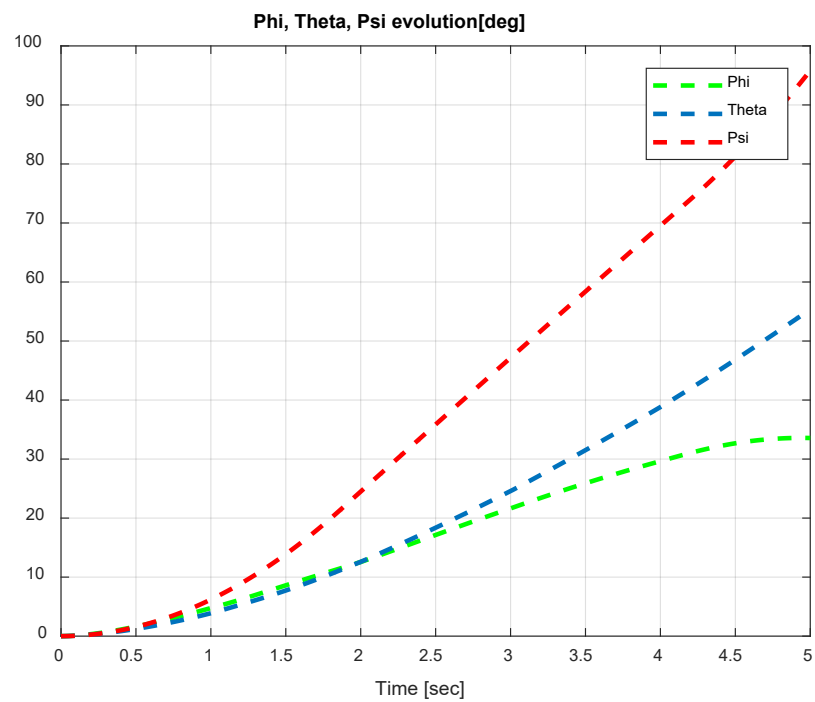


Figure 26. Quadrotor angles of Type-2 FLC.

5.4. Comparison between Proposed Controllers (Backstepping, Type-1 FLC, and Type-2 FLC)

In this section, we compare the three controllers for trajectory tracking (Figure 27) to confirm the results given in the figures above for each model. The three controllers give quite similar results for trajectory tracking, with good accuracy.

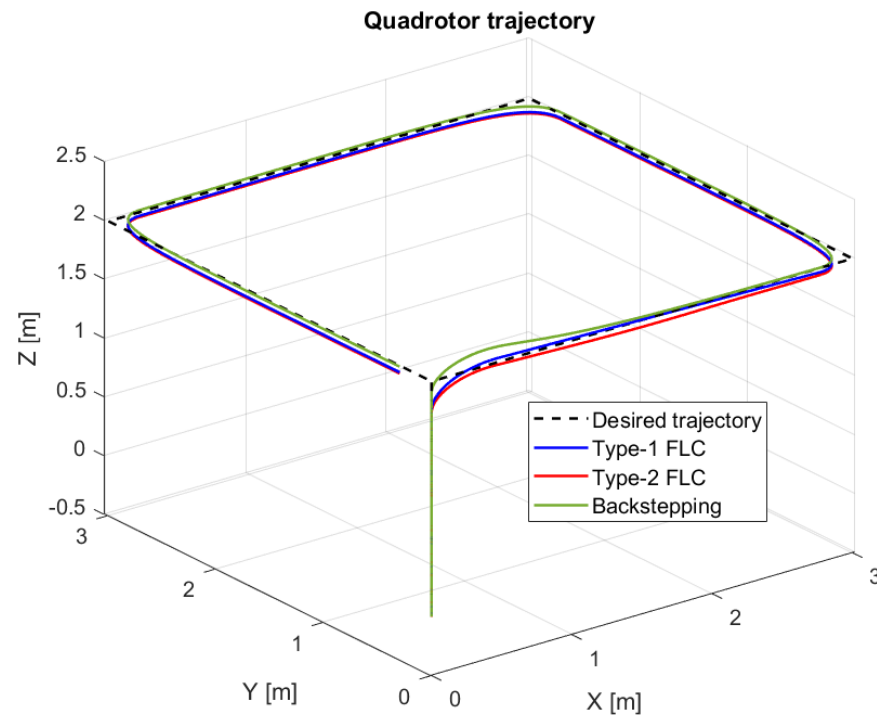


Figure 27. Quadrotor trajectory for proposed controllers.

5.5. Path Tracking for Quadrotor Using PID-Type-1 FLC and PID-Type-2 FLC Controllers with GA Tuning

The performance of the FLC controller is well known; it does not require any system modeling and is able to treat uncertainties. Furthermore, the FLC controller is robust and easy to implement. However, it can be difficult to accurately tune the fuzzy rules (in our case, 15 fuzzy rules). As a solution for this problem, and to ensure the quadrotor is able to track the predefined trajectory with high precision, PID-Type-1 FLC and PID-Type-2 FLC controllers are implemented, validated, and compared in this section. Moreover, to obtain optimal performance, the PID parameters are optimized using GA (Figure 12); the results of optimization are given in Table 5, with nine parameters tuned with GA.

Table 5. Parameters Results with GA optimization.

Controller	Parameters	PID-Type-1 FLC	PID-Type-2 FLC
Z	K1	18.20	13.8
	K2	11.98	3.98
	K4	0.47	0.6
φ	K5	2	2.24
	K6	1.5	2.86
θ	K9	2	0.19
	K10	1.4	66.42
ψ	K13	2	-1.79
	K14	15.1	5.87

5.5.1. PID-Type-1 FLC with GA Tuning Results

From the simulation results, the *PID-Type-1 FLC* controller using GA optimization provides good performances for the trajectory tracking, as confirmed by Figures 28 and 29. The position and orientation errors are illustrated, respectively, with small errors obtained. Figures 30 and 31 illustrate the input controller (quadrotor command) and the velocity evolution. We notice some chattering in the input controller when using GA optimization.

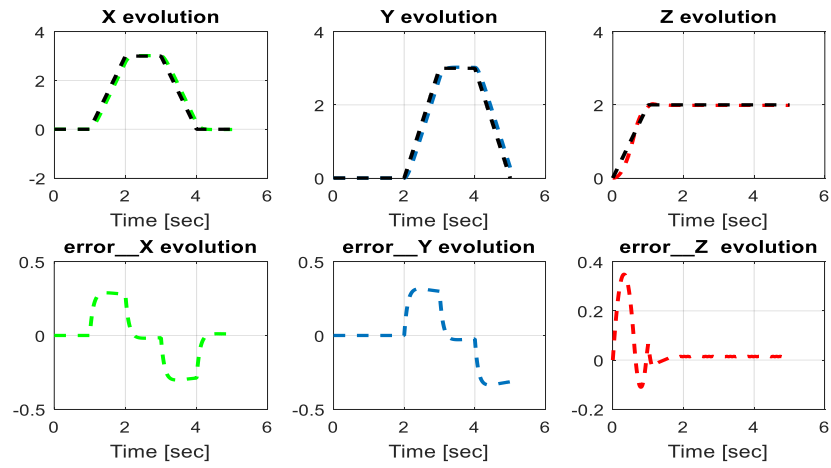


Figure 28. The errors of X, Y, and Z position using the PID-Type-1 FLC controller with GA.

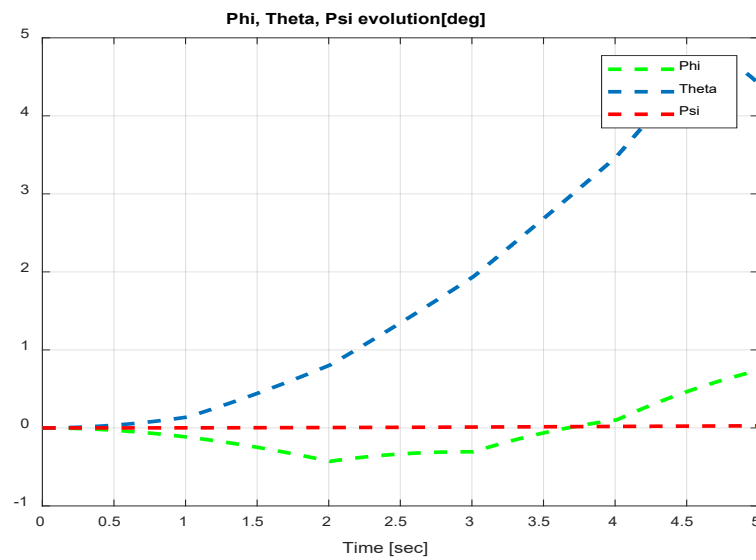


Figure 29. Quadrotor angles of PID-Type-1 FLC controller with GA.

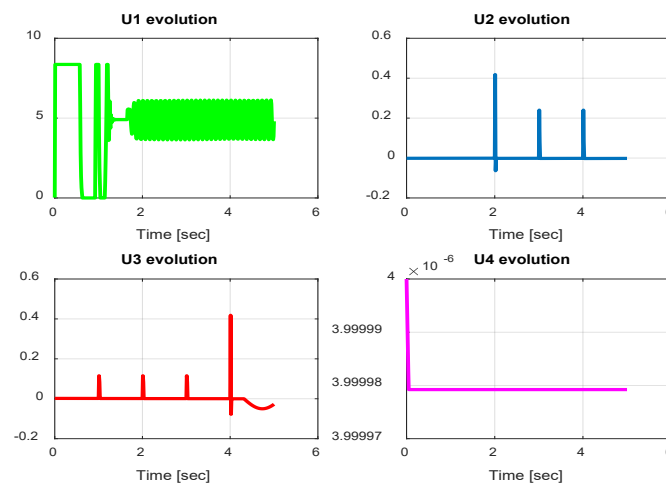


Figure 30. Quadrotor commands of PID-Type-1 FLC controller with GA.

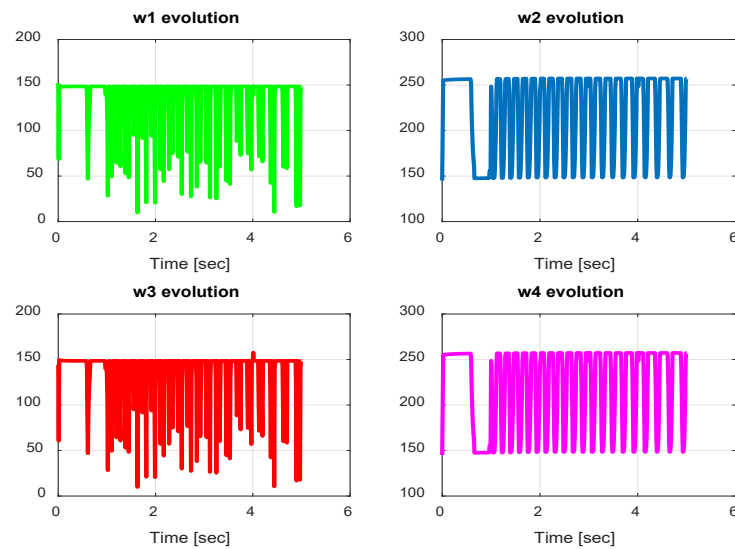


Figure 31. Motor velocities of PID-Type-1 FLC controller with GA.

5.5.2. PID-Type-2 FLC with GA Tuning Results

In this section, we present simulation results to evaluate the proposed controller (PID-Type-2 FLC) with GA parameter tuning. Figures 32 and 33 illustrate the pose errors (position and orientation, respectively). UAV commands and motor velocities are given, respectively, in Figures 34 and 35; as can be seen, the chattering effect is reduced significantly with the Type-2 FLC controller. As shown in Figure 36, the Type-2 FLC maintains good performance.

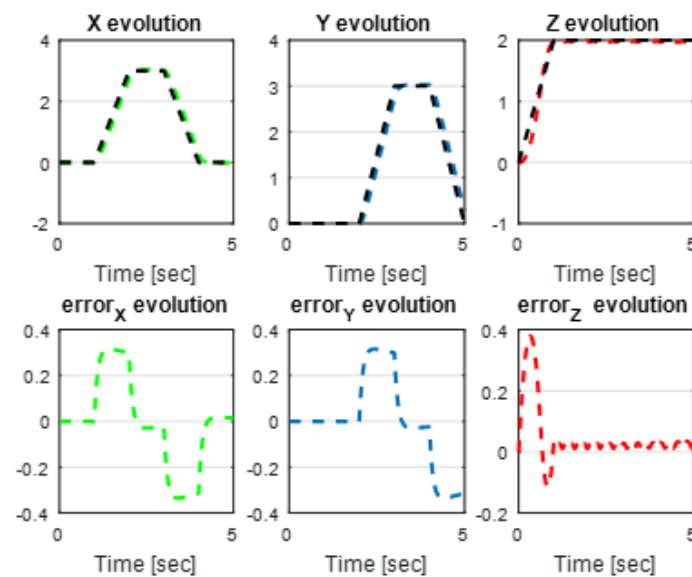


Figure 32. The errors of X, Y, and Z for the PID-Type-2 FLC controller with GA.

Figure 36 illustrates the comparison between Type-1 FLC and Type-2 FLC controllers using GA optimization. The Type-2 FLC with GA optimization performs better than the Type-1 FLC controller, due to its robustness in optimizing these parameters.

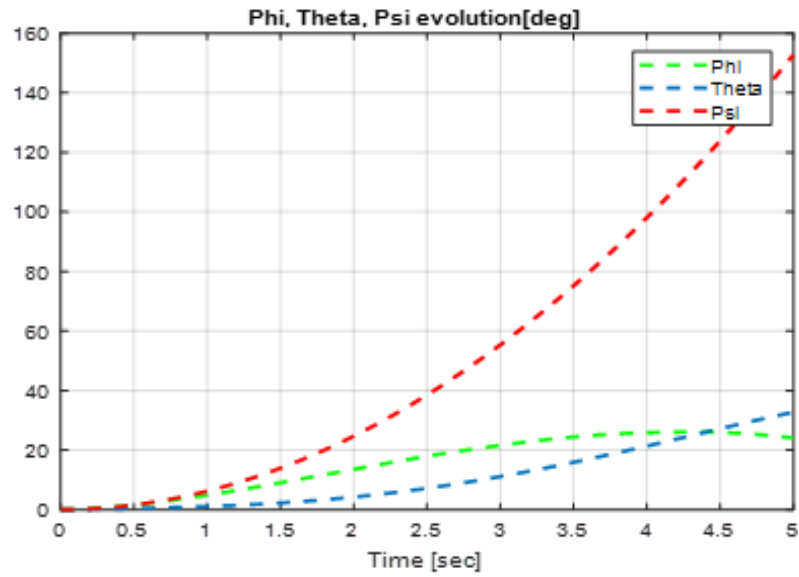


Figure 33. Quadrotor angles of PID-Type-2 FLC controller with GA.

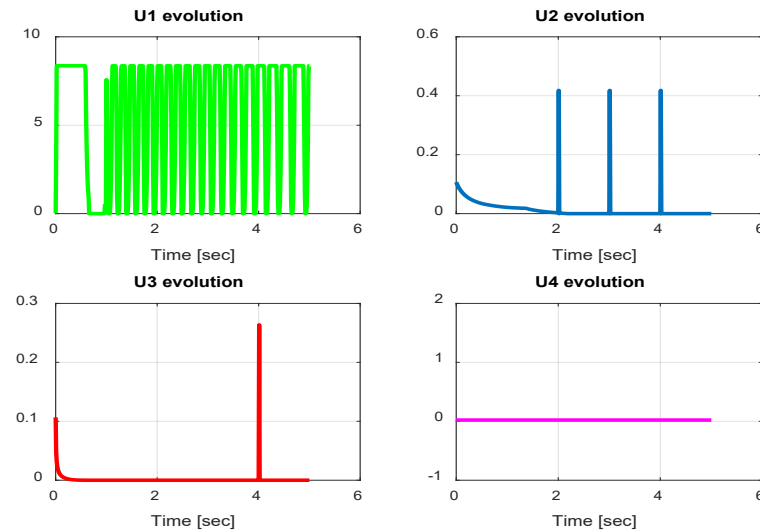


Figure 34. Quadrotor commands of PID-Type-2 FLC controller with GA.

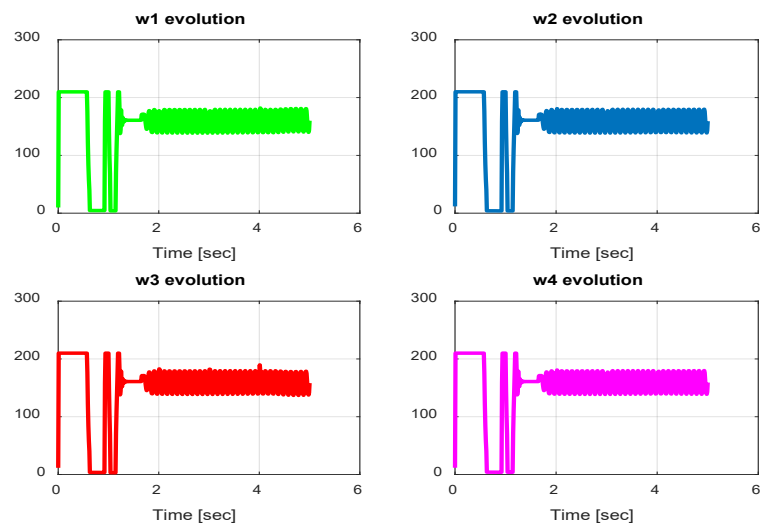


Figure 35. Motor velocities of PID-Type-2 FLC controller with GA.

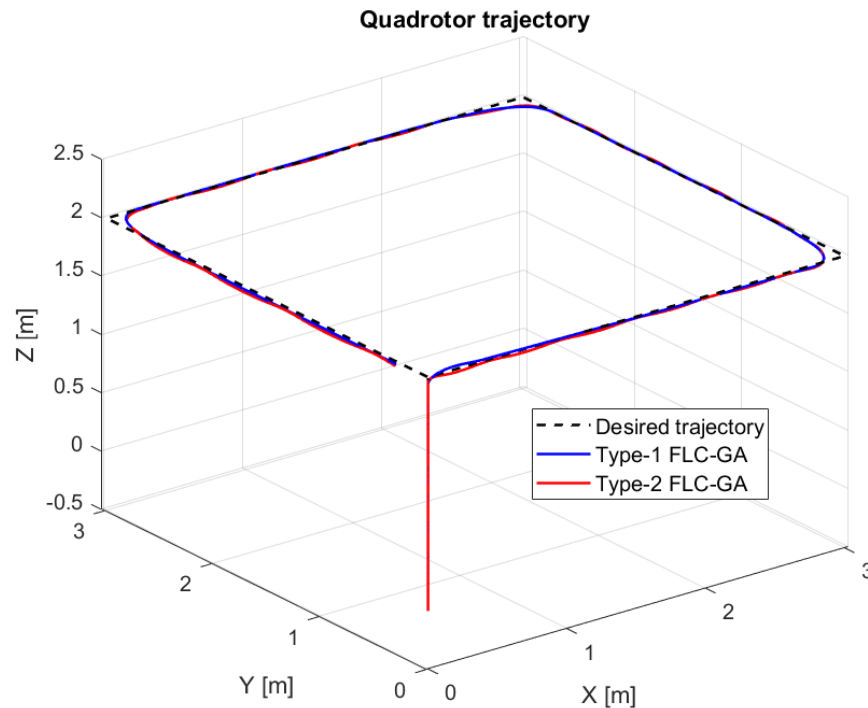


Figure 36. Quadrotor trajectory for PID-Type-1 FLC and PID-Type-2 FLC controllers with GA optimization.

5.6. Robustness Analysis

In this section, we consider a significant disturbance using a wind gust model. Simulations have been conducted to assess the robustness and efficacy of the proposed controller in tracking a specified trajectory. Two distinct simulation scenarios involving varying wind gusts are taken into account. The wind gust model outlined in Section 3 is employed to generate wind velocity along the three axes—lateral, longitudinal, and vertical. Subsequently, these velocities are treated as external disturbances affecting the translational velocity of the UAV [25].

In the first scenario, the wind gust model parameters are $L_u = L_v = 23.568$, $L_w = 3$, $\sigma_u = \sigma_v = 0.48$, and $\sigma_w = 0.14$, and a typical wind speed of 2.4 m/s is considered.

In the second scenario, the parameters of the wind gust model are $L_u = L_v = 23.568$, $L_w = 3$, $\sigma_u = \sigma_v = 0.68$, and $\sigma_w = 0.34$, and a typical wind speed of 2.4 m/s is considered.

The figures (Figure 37) presented below relate to scenario 2.

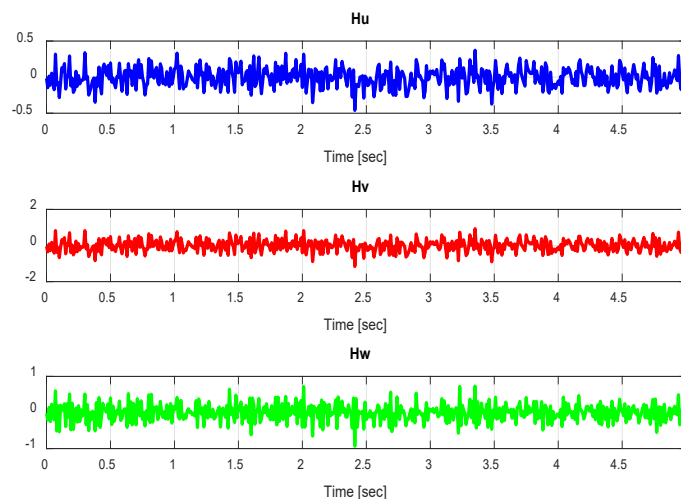


Figure 37. Wind velocity for scenario 2.

The comparison between the three controllers, Backstepping, PID-Type-1 FLC and PID-Type-2 FLC, with and without GA tuning, is illustrated in Table 6. The Root Mean Square Errors (RMSEs) are compared for the same trajectory (Figures 27, 36 and 38) using several scenarios. As illustrated in Table 6, good precision is obtained by the optimal controller (PID-Type-2 FLC), even with the presence of significant wind disturbance. This result is justified by the robustness of the Type-2 FLC and optimality of the Genetic Algorithm. According to the results illustrated in Figure 38, the Type-2 FLC provides accurate tracking of the desired trajectory compared to the Backstepping controller and Type-1 FLC. This result is confirmed by Figures 39–41, where the proposed approach provides the best accuracy.

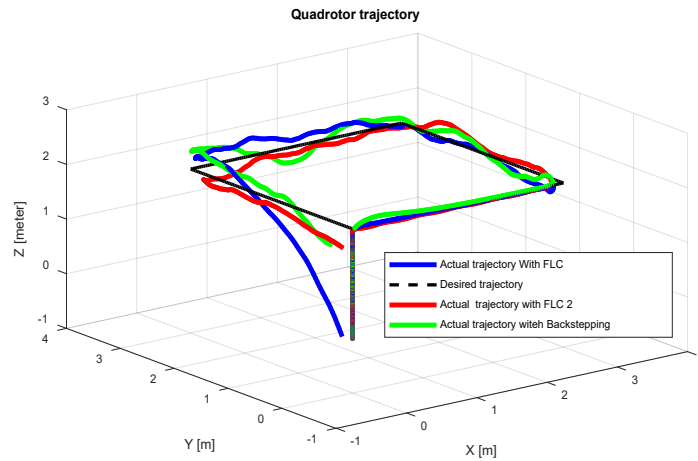


Figure 38. Quadrotor trajectory for scenario 2.

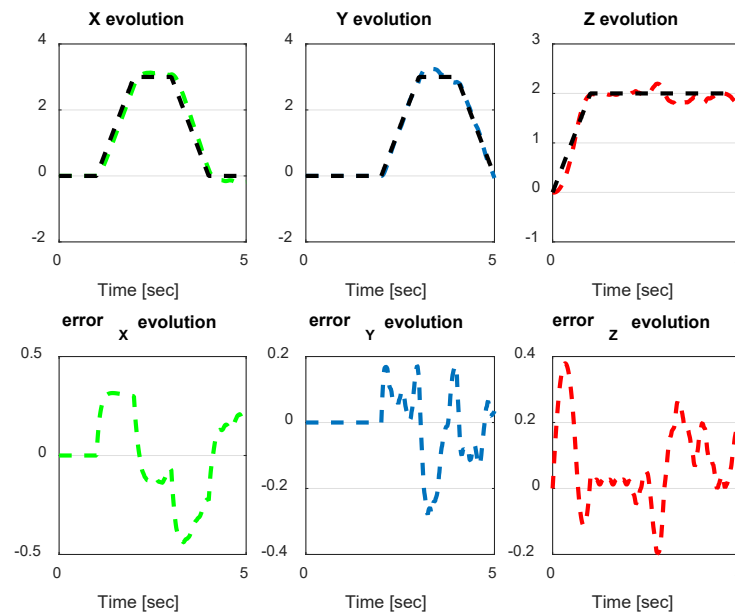


Figure 39. The errors of X, Y, and Z for PID-Type-2 FLC scenario 2.

Table 6. Parameters for outputs of the proposed controllers (RMSE).

Controllers	Approaches	Z	Y
Backstepping	Without wind	2.17	4.29
	Scenario 1	2.32	5.84
	Scenario 2	3.69	9.50

Table 6. Cont.

Controllers	Approaches	Z	Y
PID-Type-1 FLC	Without wind	3.05	4.39
	GA Optimization	2.10	4.29
	Scenario 1	2.19	6.31
	Scenario 2	6.87	7.65
PID-Type-2 FLC	Without wind	3.00	4.29
	GA Optimization	2.30	3.90
	Scenario 1	2.43	3.90
	Scenario 2	3.25	4.16

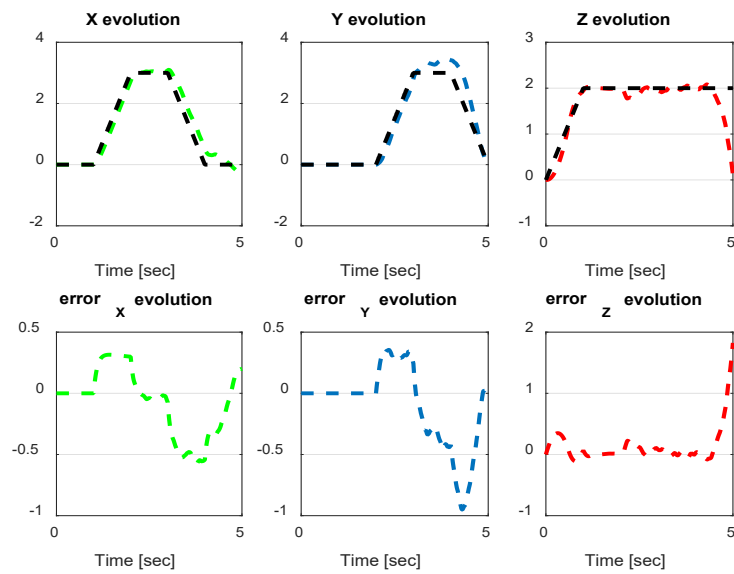


Figure 40. The errors of X, Y, and Z for PID FLC scenario 2.

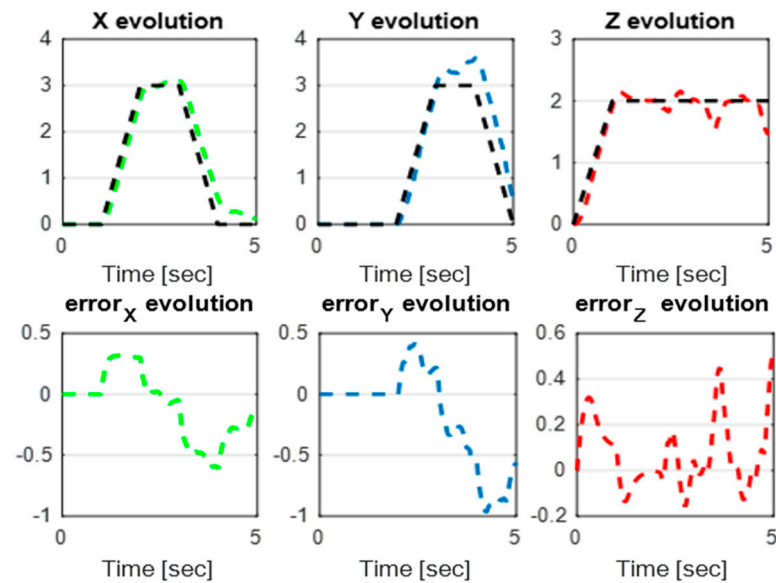


Figure 41. The errors of X, Y, and Z for Backstepping controller scenario 2.

6. Conclusions

The present research investigated the nonlinear system quadrotor aerial vehicle's trajectory tracking issue. Three types of controllers were implemented, validated, and compared. First, a model-based approach using a Backstepping controller; second, a model-free control based on a PID FLC controller; and third, a PID-Type-2 FLC. The latter is validated, firstly, using heuristic PID parameters and, secondly, using optimal PID parameters based on GA. The proposed controllers PID FLC and GA-PID-Type-2 FLC are compared on the same trajectory. Good precision was obtained by the proposed PID-Type-2 FLC. Furthermore, the proposed controller showed more robustness in the face of wind disturbances and parameter uncertainties compared to the other approaches.

Author Contributions: Conceptualization, O.M. and A.N.; methodology, D.M. and A.N.; software, O.M. and A.O.; validation, O.M., A.N. and D.M.; formal analysis, O.M.; investigation, O.M., D.M. and A.M.; resources, O.M., A.N. and A.O.; data curation, O.M.; writing—original draft preparation, O.M.; writing—review and editing, O.M., D.M. and A.N.; visualization, O.M.; supervision, D.M. and A.N.; project administration, A.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kayacan, E.; Maslim, R. Type-2 Fuzzy Logic Trajectory Tracking Control of Quadrotor VTOL Aircraft with Elliptic Membership Functions. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 339–348. [[CrossRef](#)]
2. Sattar, M.; Ismail, A. Modeling and Fuzzy Logic Control of a Quadrotor UAV. *Int. Res. J. Eng. Technol. (IRJET)* **2017**, *4*, 1494–1498.
3. Lungus, M.; Lungus, R. Adaptive Backstepping Flight Control for a mini—UAV. *Int. J. Adapt. Control Signal Process.* **2013**, *27*, 635–650. [[CrossRef](#)]
4. Krstic, M.; Kanellakopoulos, I.; Kokotovic, P. *Nonlinear and Adaptive Control Design*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1995.
5. Polycarpou, M. Stable adaptive neural control scheme for nonlinear systems. *IEEE Trans. Autom. Control* **1996**, *41*, 447–451. [[CrossRef](#)]
6. Lee, H.; Tomizuka, M. Robust adaptive control using a universal approximator for SISO nonlinear systems. *IEEE Trans. Fuzzy Syst.* **2000**, *8*, 95–106.
7. Pan, Y.; Sun, T.; Liu, Y.; Yu, H. Composite learning from adaptive backstepping neural network control. *Neural Netw.* **2017**, *95*, 134–142. [[CrossRef](#)]
8. Babaei, R.; Farhad, E.A. Robust Backstepping Control of a Quadrotor UAV Using Extended Kalman Bucy Filter. *IJMEC* **2015**, *5*, 2276–2291.
9. Baek, S.J.; Lee, D.J.; Park, J.H. Design of lateral fuzzy-PI controller for unmanned quadrotor robot. *J. Inst. Control Syst.* **2013**, *19*, 164–170. [[CrossRef](#)]
10. Talha, M.; Asghar, F.; Rohan, A.; Rabah, M.; Ho, K.S. Fuzzy Logic-Based Robust and Autonomous Safe Landing for UAV Quadcopter. *Arab. J. Sci. Eng.* **2018**, *44*, 2627–2639. [[CrossRef](#)]
11. Dong, J.; He, B. Novel Fuzzy PID-Type Iterative Learning Control for Quadrotor UAV. *Sensors* **2019**, *19*, 24. [[CrossRef](#)]
12. Carvalho, G.; Guedes, I.; Pinto, M.; Zachi, A.; Almeida, L.; Andrade, F.; Melo, A.G. Hybrid PID-Fuzzy controller for autonomous UAV stabilization. In Proceedings of the 14th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil, 15–18 August 2021; pp. 1296–1302.
13. Pussente, G.A.N.; de Aguiar, E.P.; Marcato, A.L.M.; Pinto, M.F. UAV Power Line Tracking Control Based on a Type-2 Fuzzy-PID Approach. *Robotics* **2023**, *12*, 60. [[CrossRef](#)]
14. Melo, A.G.; Andrade, F.A.A.; Guedes, I.P.; Carvalho, G.F.; Zachi, A.R.L.; Pinto, M.F. Fuzzy Gain-Scheduling PID for UAV Position and Altitude Controllers. *Sensors* **2022**, *22*, 2173. [[CrossRef](#)] [[PubMed](#)]
15. Agustina, N.P.; Darwito, P.A. Autonomous Quadcopter Trajectory Tracking and Stabilization Using Control System Based on Sliding Mode Control and Kalman Filter. In Proceedings of the 2023 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 26–27 July 2023; pp. 489–493. [[CrossRef](#)]

16. Satybaldina, D.; Dabayeva, A.; Kissikova, N.; Uskenbayeva, G.; Shukirova, A. Mixed H₂/H_∞ robust controllers in aircraft control problem. *Int. J. Electr. Comput. Eng. (IJECE)* **2023**, *13*, 6249–6258. [[CrossRef](#)]
17. Ghazali, M.R.; Ahmad, M.A.; Jusof, M.F.M.; Ismail, R.M.T.R. A data-driven neuroendocrine-PID controller for underactuated systems based on safe experimentation dynamics. In Proceedings of the 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA), Penang, Malaysia, 9–10 March 2018; pp. 61–66. [[CrossRef](#)]
18. Olguin-Roque, J.; Salazar, S.; González-Hernandez, I.; Lozano, R. A Robust Fixed-Time Sliding Mode Control for Quadrotor UAV. *Algorithms* **2023**, *16*, 229. [[CrossRef](#)]
19. Mehmood, Y.; Aslam, J.; Ullah, N.; Chowdhury, M.S.; Techato, K.; Alzaed, A.N. Adaptive Robust Trajectory Tracking Control of Multiple Quad-Rotor UAVs with Parametric Uncertainties and Disturbances. *Sensors* **2021**, *21*, 2401. [[CrossRef](#)]
20. Ban, W.; Youmin, Z.; Wei, Z. Integrated path planning and trajectory tracking control for quadrotor UAVs with obstacle avoidance in the presence of environmental and systematic uncertainties: Theory and experiment. *Aerosp. Sci. Technol.* **2022**, *120*, 107277.
21. Abdelghany, M.B.; Moustafa, A.M.; Moness, M. Benchmarking Tracking Autopilots for Quadrotor Aerial Robotic System Using Heuristic Nonlinear Controllers. *Drones* **2022**, *6*, 379. [[CrossRef](#)]
22. Mollov, L.; Kralev, J.; Slavov, T.; Petkov, P. μ -Synthesis and Hardware-in-the-loop Simulation of Miniature Helicopter Control System. *J. Intell. Robot. Syst.* **2014**, *76*, 315–351. [[CrossRef](#)]
23. Etele, J. *Overview of Wind Gust Modelling with Application to Autonomous Low-Level UAV Control*; Defence Research and Development Canada: North York, ON, Canada, 2006.
24. Alexis, K.; Nikolakopoulos, G.; Tzes, A. Constrained-control of a quadrotor helicopter for trajectory tracking under wind-gust disturbances. In Proceedings of the 15th IEEE Mediterranean Electrotechnical Conference, Valletta, Malta, 26–28 April 2010; pp. 1411–1416.
25. Ahmed, B.; Kendoul, F. Flight control of a small helicopter in unknown wind conditions. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 3536–3541.
26. Yacef, F.; Bouhali, O.; Hamerlain, M.; Rizoug, N. Observer-based Adaptive Fuzzy Backstepping Tracking Control of Quadrotor Unmanned Aerial Vehicle Powered by Li-ion Battery. *J. Intell. Robot. Syst.* **2016**, *84*, 179–197. [[CrossRef](#)]
27. Nafia, N.; El Kari, A.; Ayad, H.; Mjahed, M. Robust Full Tracking Control Design of Disturbed Quadrotor UAVs with Unknown Dynamics. *Aerospace* **2018**, *5*, 115. [[CrossRef](#)]
28. Saidi, Y.; Nemra, A.; Tadjine, M. Robust mobile robot navigation using fuzzy Type-2 with wheel slip dynamic modeling and parameters uncertainties. *Int. J. Model. Simul.* **2019**, *40*, 397–420. [[CrossRef](#)]
29. Xin, J.; Zhong, J.; Yang, F.; Cui, Y.; Sheng, J. An Improved Genetic Algorithm for Path-Planning of Unmanned Surface Vehicle. *Sensors* **2019**, *19*, 2640. [[CrossRef](#)] [[PubMed](#)]
30. Levitin, G.; Rubinovitz, J.; Shnits, B. A genetic algorithm for robotic assembly line balancing. *Eur. J. Oper. Res.* **2006**, *168*, 811–825. [[CrossRef](#)]
31. Rodríguez-Abreo, O.; Garcia-Guendulain, J.M.; Hernández-Alvarado, R.; Flores Rangel, A.; Fuentes-Silva, C. Genetic Algorithm-Based Tuning of Backstepping Controller for a Quadrotor-Type Unmanned Aerial Vehicle. *Electronics* **2020**, *9*, 1735. [[CrossRef](#)]
32. Nasir, A.N.K.; Ahmad, M.A.; Tokhi, M.O. Hybrid spiral-bacterial foraging algorithm for a fuzzy control design of a flexible manipulator. *J. Low Freq. Noise Vib. Act. Control* **2021**, *41*, 340–358. [[CrossRef](#)]
33. Dhakhnamoorthy, C.; Mani, S.K.; Mathivanan, S.K.; Mohan, S.; Jayagopal, P.; Mallik, S.; Qin, H. Hybrid Whale and Gray Wolf Deep Learning Optimization Algorithm for Prediction of Alzheimer’s Disease. *Mathematics* **2023**, *11*, 1136. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.