Generalization to Type 2 of PSO-Optimized Type 1 PD Fuzzy Controller and its Application to a Quadrotor UAV

Meriem Benbrahim
Industrial Engineering Department, Batna 2 University, Algeria
m.benbrahim@univ-batna2.dz

Moufid Bouhentala
Electronics Department, Batna 2 University, Algeria
bouhentalam@gmail.com

Mouna Ghanai
Electronics Department, Batna 2 University, Algeria
m.ghanai@univ-batna2.dz

Kheireddine Chafaa
Electronics Department, Batna 2 University, Algeria
k.chafaa@univ-batna2.dz (corresponding author)

Najib Essounbouli
IUT de Troyes, Reims University, Champagne-Ardenne, France
najib.essounbouli@univ-reims.fr

Received: 30 November 2024 | Revised: 22 January 2025 and 9 February 2025 | Accepted: 10 February 2025

ABSTRACT

This study presents a new Type 2 fuzzy logic (interval-valued fuzzy logic) control system for a Vertical Take Off and Landing (VTOL) quadrotor. The goal of the control design is to obtain robust and stable tracking of the desired angles in the presence of disturbances and noises. The membership functions relative to the linguistic variables of the fuzzy if-then rules are chosen to control the quadrotor to track a reference trajectory. The Particle Swarm Optimization (PSO) method is used to optimize controller-free parameters. The results obtained from an extensive simulation study show that the quadrotor can operate autonomously in flight with stable orientation. The performance of the controller was evaluated for both Type 1 and Type 2 fuzzy controllers. The simulation results show the effectiveness of the proposed controller and its better performance.

Keywords-PSO; type 1 fuzzy logic; type 2 fuzzy logic; quadrotors; unmanned aerial vehicle

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), called also drones, are used in several applications: search and rescue missions, exploration, security, and surveillance [1]. In literature, UAVs with quadrotor architecture have been investigated in many works [2]. They are highly nonlinear, under-actuated, and time-varying systems. It should be noted that methods related to fully actuated systems cannot be used in under-actuated ones. Therefore, nonlinear modeling strategies and modern nonlinear control are used to realize autonomous flight with excellent performance [3].

A Type 2 Fuzzy Logic System (FLS) is a fuzzy system with antecedent or/and consequence membership having multivalued membership. In [4], the notion of a type 2 fuzzy set was defined, which is a generalization of the approach of a Type 1 fuzzy set. A Type 1 FLS (possessing membership grades that are of Type 1) is generally not able to consider rule uncertainties [5-6]. The swarm technique is inspired by biology, such as the swarming, flocking, and herding phenomena of animals [7]. The main advantage of the PSO algorithm is its fast convergence compared to Genetic Algorithms (GAs). Several algorithms have been proposed to control quadrotors. In [2], the Linear-Quadratic (LQ) and Proportional-Integral-Derivative (PID) controllers were designed and applied, demonstrating their ability to regulate such systems. This design used linearization of the vehicle model. In this work, the control systems worked very well only if the vehicle's configuration was practically the same as that of the linearized model, introducing difficulties in handling perturbations and uncertainties. In [8], an efficient feedback linearization controller was introduced, showing its efficiency and robustness to parameter uncertainties and perturbations. In [3], the backstepping technique was applied using visual feedback as the primary sensor (camera). In addition, a comparison with the feedback linearization method was given, confirming the superiority of backstepping. In [9], a new controller was proposed based on a predictive switching model to achieve accurate trajectory control in the presence of powerful wind gusts. The controller was implemented and tested successfully on position tracking, hovering, and attitude manoeuvers. In [10], an L1-optimal controller was developed and implemented for drones, and experimental data showed that the proposed controller attenuated disturbances. On the other hand, intelligent control techniques have been extensively used with this type of system, such as adaptive fuzzy control [11], neural networks [12], and deterministic learning [13].

This study proposes an intelligent control strategy based on optimized PD Type 1 and Type 2 fuzzy control systems. The proposed scheme is based on the fact that the controller parameters called input scaling factor and output scaling factor are optimized in an offline manner. The structure of the introduced method consists of: (i) Designing a Type 1 fuzzy architecture working offline and allowing finding the optimum scaling factors by using the PSO algorithm, (ii) the obtained scaling factors gains are injected into the online PD Type 1 fuzzy controller, and (iii) the PD Type 1 fuzzy controller is extended to a PD Type 2 fuzzy controller.

II. QUADROTOR DYNAMICAL MODELING

This section summarizes some developments in the mathematical modeling of quadrotors. In the literature, these systems were broadly studied and modeled in [2-8]. First of all, consider two coordinate systems of the earth and of the craft which are connected through the rotation θ around the x-axis (Roll), the rotation φ around the y-axis (Pitch), and the rotation ψ around the z-axis (Yaw). Using the Newton-Euler numerical method, the angular dynamic of the system is expressed by:

$$\tau = J\dot{\omega} + \omega \times (J\omega + J_r \Omega_r e_3) \tag{1}$$

where τ is the torque vector on the spatial axes, J is the tensor of inertia relative to the rigid body, ω is the angular velocities vector, J_r is the inertia of the rotor, the vector $e_3 = [0\ 0\ 1]^T$ denotes the orientation of the rotor and $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ is the total rotor angular speed.

Dynamics translation in a fixed reference is given by the following formula:

$$m\dot{v} = RT - mge_3 \tag{2}$$

where m is the quadrotor mass, v is the vector of Cartesian velocities, matrix R represents the desired rotation matrix, T is the thrust forces vector, g is the gravitation constant, and b is a constant (thrust factor) of proportionality between the force and the square of rotational velocity.

The considered quadrotor possesses four inputs (U_1, U_2, U_3, U_4) and six outputs $(X, Y, Z, \phi, \theta, \psi)$. The input U_1 represents the total thrust applied on the quadrotor in the z direction, U_2 and U_3designate the roll and the pitch inputs, and U_4 represents the input relative to the yaw. Note that each input U_i depends on the rotor speeds as follows:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = lb(-\Omega_2^2 + \Omega_4^2) \\ U_3 = lb(+\Omega_1^2 - \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{cases} \tag{3}$$

with l being the arm length and d being the drag factor.

Note that the system (3) is invertible. Consequently, the controlled speeds can be extracted from control signals. Using dynamical equations (1) and (2), the following cartesian and angular differential equations can be obtained:

$$\begin{cases} \ddot{X} = (cos\varphi\ sin\theta\ cos\psi - sin\varphi\ sin\psi)\ \frac{U_1}{m} \\ \ddot{Y} = (cos\varphi\ sin\theta\ sin\psi - sin\varphi\ cos\psi)\ \frac{U_1}{m} \\ \ddot{Z} = -g + (cos\varphi\ cos\theta)\frac{U_1}{m} \\ \ddot{\varphi} = \frac{J_{YY} - J_{ZZ}}{J_{XX}}\dot{\theta}\dot{\psi} - \frac{J_r}{J_{XX}}\dot{\theta}\Omega + \frac{U_2}{J_{XX}} \\ \ddot{\theta} = \frac{J_{ZZ} - J_{XX}}{J_{YY}}\dot{\varphi}\dot{\psi} + \frac{J_r}{J_{YY}}\dot{\varphi}\Omega + \frac{U_3}{J_{YY}} \\ \ddot{\psi} = \frac{J_{ZZ} - J_{YY}}{J_{ZZ}}\dot{\theta}\dot{\varphi} + \frac{U_4}{J_{ZZ}} \end{cases} \tag{4}$$

Model (4) can be simplified by defining U_x and U_y as follows:

$$\begin{cases} U_x = (cos\,\phi\, sin\,\theta\, cos\,\psi - sin\,\phi\, sin\,\psi) \\ U_y = (cos\,\phi\, sin\,\theta\, sin\,\psi - sin\,\phi\, cos\,\psi) \end{cases} \tag{5}$$

III. TYPE 2 FUZZY SYSTEMS

Type-2 fuzzy logic is characterized by the generalization of defuzzification called type reduction introduced by Karnik and Mendel [14]. Therefore, a Type 1 set can be defuzzified to a single number. In this investigation, the use of the center of sets type reduction was chosen:

$$Y_{\cos}(Y^1,...,Y^M,W^1,...,W^M) =$$

$$\int_{y^1}\cdots\int_{y^M}\cdots\int_{W^1}\cdots\int_{W^M}\cdots 1/\frac{\sum_{i=1}^{M} w^i y^y}{\sum_{i=1}^{M} w^i} = [y_l, y_r] \tag{6}$$

where Y_{cos} is the interval result given by two end points y_l and y_r , $y^i \in Y^i = [y_l^i, y_r^i]$, Y^i is the centroid, $\tilde{G}^i$ is the consequence, and $w^i \in W^i = [\underline{w}^i, \bar{w}^i]$ is the firing interval [5-6].

IV. PARTICLE SWARM OPTIMIZATION (PSO)

In the PSO algorithm, the set of individuals is named swarm, and every element is termed a particle. PSO starts with a random initialization of the population. Each particle moves in an N-dimensional space in which the solution must exist and memorizes the best-met position. Then, the elements in the swarm can tune their positions and velocities. Each element in the population of size N is defined by its relative position $x_i \in \Re^N$ and relative velocity $v_i \in \Re^N$ $(i = 1, 2, \ldots, N)$. The position of each element is a possible solution, and the best particle position is memorized as $p_{bi} \in \Re^N$ (best narrow position). The swarm remembers the best position accomplished by any of its elements as $p_g \in \Re^N$ (the best global location). However, the optimal position is obtained using a cost criteria defined as a function of the optimization problem. The position and velocity of all elements at the k^{th} iteration are updated by:

$$v_j(k+1) = w.v_i(k) + c_1.r_1(k).\left(p_{bi}(k) - x_i(k)\right) + c_2 r_2(k).\left(p_g(k) - x_i(k)\right. \quad (7)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (8)$$

where $r_1(.)$ and $r_2(.)$ are two random constants uniformly distributed in the interval $[0, 1]$ to offer random weights to the different modules contributing to the velocities of the elements, c_1 and c_2 are positive numbers describing the perceptive acceleration (govern the relative influence of the narrow knowledge on the particle movement) and the collective acceleration, respectively, and w is the inertia mass factor and controls the influence of the earlier past velocities on the present one. Equation (7) updates the velocity from the previous velocity to the present one. The actual position is then estimated by the sum of the past position and the actual velocity using (8).

V. CONTROLLER DESIGN

The proposed approach aims to consider a Type 1 fuzzy controller with adaptable parameters called scaling factors. The PSO algorithm was used for parameter adaptation. Once the fuzzy controller is PSO-optimized, it is generalized to a Type 2 fuzzy controller. During the offline optimization phase, the system consists of two main components: a fuzzy controller block and a PSO optimizer block, which is responsible for optimizing the input scaling factors (Figure 2). The control strategy employed in this investigation is a fuzzy PD controller that processes the input variables' error e_k, changes it in error Δe_k, and generates an output control signal u_k. The input scaling factors are denoted as α_e, $\alpha_{\Delta e}$, and the output control gain is denoted as G. Then, the control law can be expressed as follows:

$$u_k = G u_k(e_k, \Delta e_k, \alpha_e, \alpha_{\Delta e}) \quad (9)$$

where u_k is called a PD-Type 1 fuzzy logic control action. The angles are controlled from the plant model (4).

To control the plant, $(U_x U_y U_1)$ is considered as new input $(u_x u_y u_z)$ as in (10). Note that the position equations were already in a perfect double integrator form, which allows us to control each variable independently.

$$U_1 = \frac{m(g + u_1(\Delta e_z, e_z))}{\cos\phi \cos\theta} \quad (10)$$

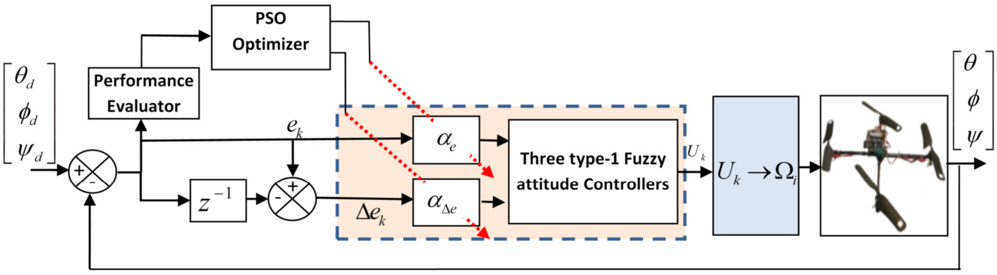

Fig. 1. The framework of the PSO-FUZZY controller for angles (Phase 1).

The proposed algorithm consists of two phases: (1) Attitude control and (2) Position control. In phase 1, a PSO-FUZZY controller system is introduced (Figure 1), where its input vector is $[\theta_d \phi_d \psi_d]$ and its corresponding output vector is $[\theta \phi \psi]$. First, this structure will work in an offline manner, in which the error $e(t) = \left[e_\phi e_\theta e_\psi\right]$ is minimized through the performance estimator and a PSO algorithm. The later calculates the fitness value represented by an Integral Absolute Error (IAE) criterion which must be determined by its minimum value $J = \sum_{k=0}^{N} |e_k|$, with N representing the number of samples. The obtained IAE is then applied to the PSO, and based on its value it will tune and optimize the unknown scaling factors $\alpha_e$and $\alpha_{\Delta e}$ by updating the solution according to (7) and (8) to provide better solutions. The novel solutions α_e and $\alpha_{\Delta e}$ are then used for the afterward iteration till a fixed number of iterations is attained. Therefore the best values of $\alpha_e$and $\alpha_{\Delta e}$ are achieved and then injected into the loop to work online. In Phase 2, a fuzzy position controller is designed based on the same approach as in phase 1. The superposition of the position controller over the attitude controller in cascade architecture (Figure 3) enables the quadrotor to perform position tracking. In phase 2 the input vector is $[x_d y_d z_d]$ and the output vector is $[xyz]$ give the new error vector e defined by $e = \left[e_x e_y e_z\right]$. The same optimization procedure is applied as in phase 1. Then, optimal values of $\beta_e = \left[\beta_e^x \beta_e^y \beta_e^z\right]$and $\beta_{\Delta e}$ are obtained. Note that in both phases, the PSO algorithm is implemented in an offline manner because it needs several iterations to converge to acceptable solutions. Each PSO particle represents a given input scaling factor of the fuzzy controller. At each iteration the PSO-FUZZY controller has to be executed once. Consequently, it must be executed several times allowing the optimization of the parameters α and β at each measurement. Finally, after the parameters' optimization, the obtained optimal scaling factor is injected into the control.

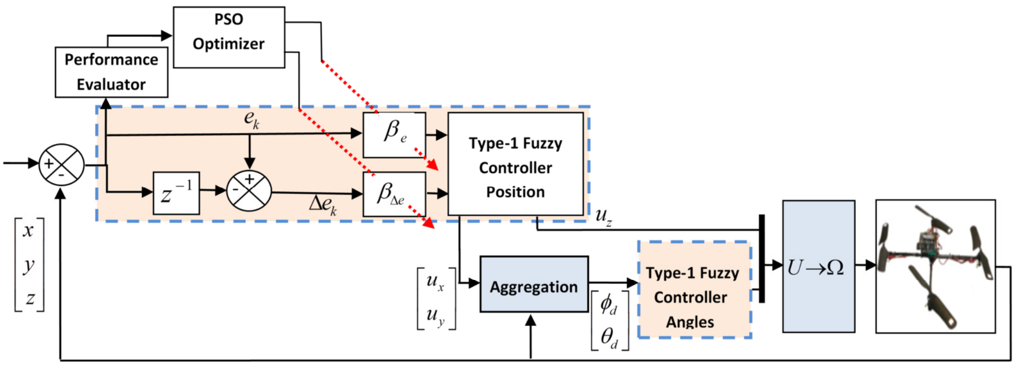

Fig. 2. Framework of the PSO-FUZZY controller for position (Phase 2).

VI. RESULTS AND DISCUSSION

Simulations were carried out in MATLAB 9.5 R2018b. The model parameters of the quadrotor were taken from [3, 15, 16] as: gravity g = 9.81 m/s^2, thrust factor, b = 3.1310^{-5}, drag factor d = 7.510^{-7}, mass m = 0.65 Kg, inertia on x I_{xx}= 0.0075 Kg.m^2, inertia on y I_{yy} = 0.0075 Kg.m^2, inertia on z I_{zz}= *0.013* Kg.m^2, length L = 0.23 m.

For Type 1 and Type 2 fuzzy controllers, 7 input membership functions and 7 output membership functions were used with 49 rules. The optimization of the controller's parameters was carried out by PSO with the following values: two dimensions for the search space, swarm population size of 5, 20 iterations, cognitive and social acceleration factors c_1 = 2 and c_2 = 2, and the inertia weight factor w set to 0.8.

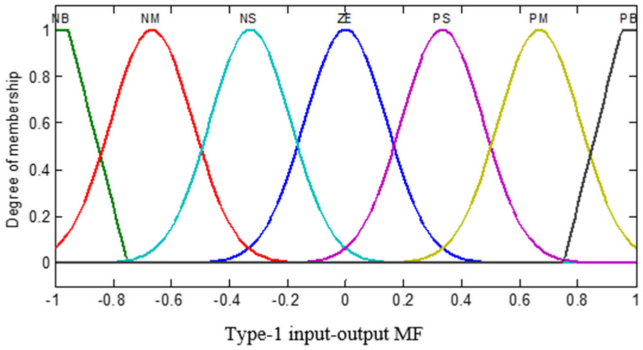

Fig. 3. Type 1 membership functions.

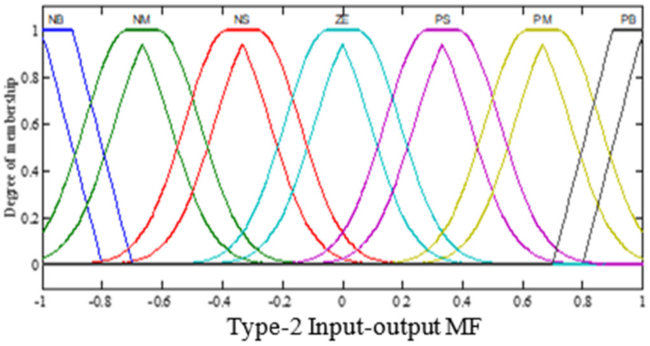

Fig. 4. Type 2 fuzzy membership functions.

The best-identified parameters are summarized in Table 1, for which the convergence of the criteria is shown in Figure 5.

TABLE I. PARAMETERS AFTER PSO

		α_e	$\alpha_{\Delta e}$
Scaling factor	**Pitch**	4.6182	0.2366
	Roll	4.6182	0.2366
	Yaw	1.0407	0.6641
	x	0.6667	3.9796
	y	0.6667	3.9796
	z	52.5060	1.0166

Figure 5 shows the evolution of the objective function, which decreases until its convergence to the minimum, leading the algorithm to stop the adaptation of the parameters.

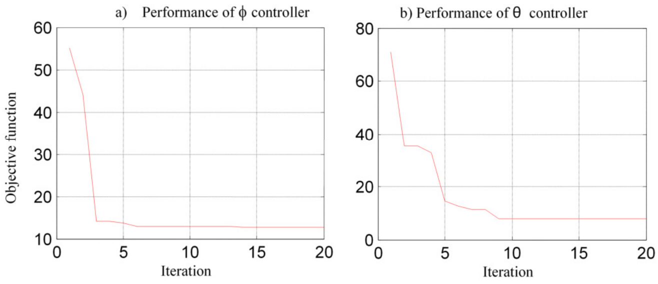

Fig. 5. Evolution of fitness function vs iterations in PSO algorithm.

Two types of experiments were used: (a) without noise and (b) with noise. The performance of tracking and stability of the desired angles and the attitude were determined. For the noisy case, simulations were carried out considering external disturbances and noise with different Signal-to-Noise-Ratio (SNR) values. In this section, two types of reference inputs were used:

i) Attitude reference trajectory: $(\phi(t), \theta(t), 0, z(t))$ defined by:

$$\begin{cases} \phi(t) = \theta(t) = \begin{cases} 00 \le t \le 2 \\ 0.22 \le t \le 10 \\ -0.2t > 10 \end{cases} \\ z(t) = \begin{cases} 00 \le t \le 2 \\ 0.1t - 0.22 \le t \le 10 \\ 2.2t > 10 \end{cases} \end{cases} \tag{11}$$

ii) Position reference trajectory:

$$(x, y, z, \psi) = (sin(t), cos(t), z(t), 0) \qquad (12)$$

Note that $z(t)$ is the same as in (11). In the experiments, initial conditions were $(x, y, z) = (0, 0, 0)$ m and $(\phi, \theta, \psi) = (0,0,0)$ rad. Figure 6 shows the response without noise.

A. Case 1: Experiments without Noise

The results for the noiseless case for the attitude reference trajectory are shown in Figures 6 and 7. Figure 6 shows the responses of Pitch, Yaw, Roll, and *z*, and Figure 7 illustrates their corresponding control actions. As can be observed, the performances of Type 1 and Type 2 controllers were very similar.

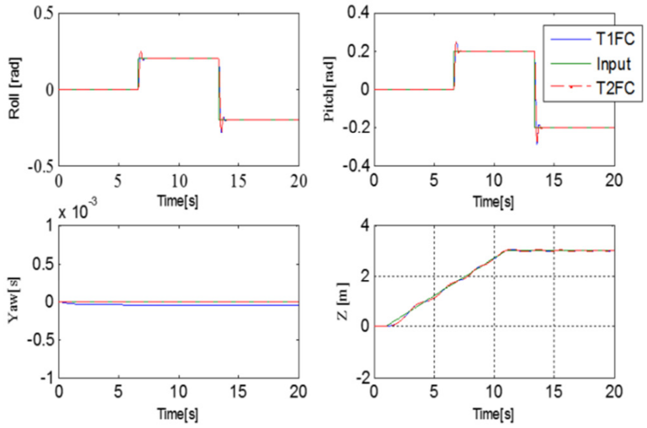

Fig. 6. Noiseless angles' tracking responses.

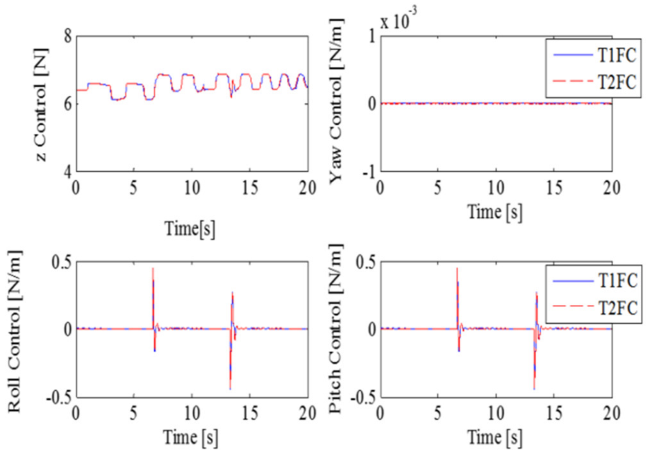

Fig. 7. Noiseless control signals for angles' tracking.

The results for the noiseless case for the position reference trajectory are shown in Figure 8, illustrating the position responses of *x*, *y*, and *z* with the corresponding angles Roll, Pitch, and Yaw. It can be observed that the Yaw performance of the Type 2 controller is better than that of Type 1, but all the other performances are very similar.

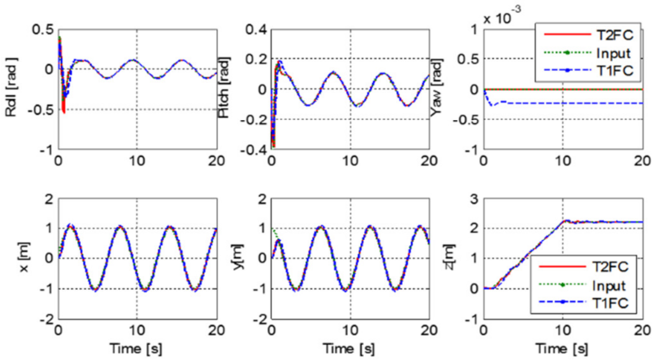

Fig. 8. Noiseless position tracking responses.

B. Noisy Experiments

In this case, noise with zero mean and different SNRs (20, 25, 30) was injected into the system to evaluate the performance of the proposed method. In the first case, the attitude reference trajectories (*SNR* = 30) are shown in Figures 9 and 10. Figure 9 illustrates the responses of Pitch, Yaw, Roll, and *z*, and Figure 10 illustrates their corresponding control actions. It can be visually observed that the performance of the Type 2 is slightly better than that of the Type 1 fuzzy controller. For the position case, the reference trajectory (*SNR* =30) is shown in Figure 11, which illustrates the responses of *x*, *y*, and *z* with the corresponding angles (Roll, Pitch, and Yaw). It can be visually observed that the performance of the Type 2 is slightly better than that of the Type 1 fuzzy controller.

Tables II and III show a quantitative comparison between Type 1 and Type 2 controllers using the Integral Absolute Error (IAE) criteria. According to the IAE simulation results, it can be observed that the Type 2 controller has considerably better results than the Type 1 one, especially with greater noise.

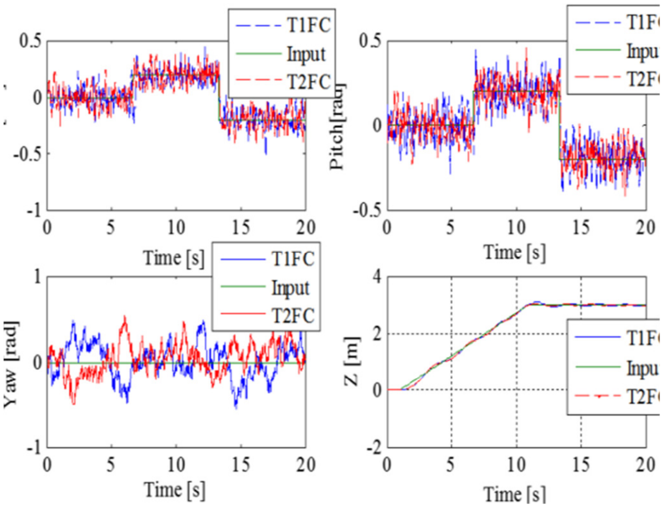

Fig. 9. Angle responses with noise (*SNR* = 30).

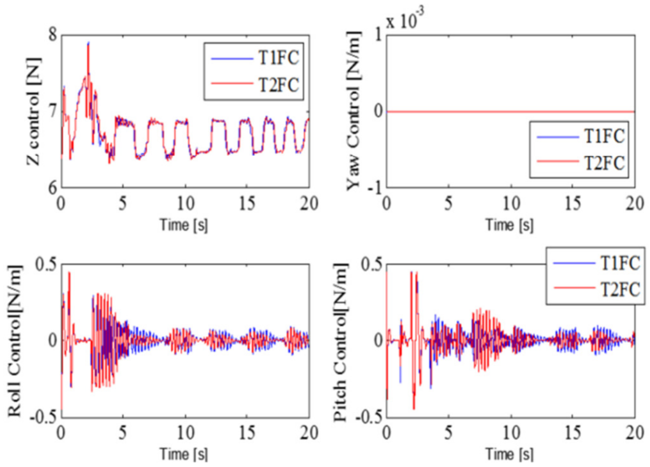

Fig. 10. Control signals for angle responses with noise (*SNR*=30).

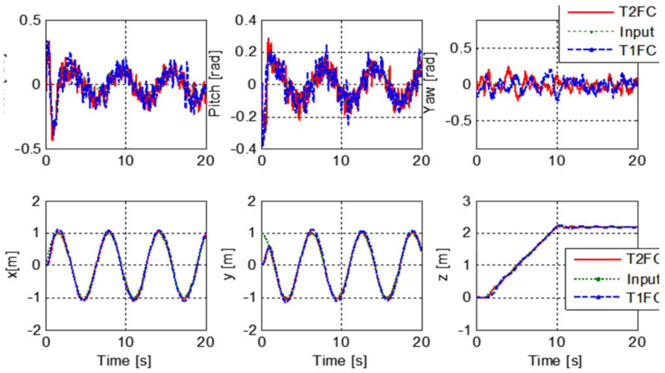

Fig. 11. Position responses with noise (SNR=30).

VII. CONCLUSION

The control of UAVs remains a critical area of research, as achieving stability, precision, and robustness poses significant challenges. Fuzzy logic and PSO offer powerful techniques for nonlinear and uncertain problems across various domains. Fuzzy logic provides a robust framework for handling imprecision. On the other hand, PSO is a powerful metaheuristic algorithm that offers an efficient approach to solving complex optimization problems.

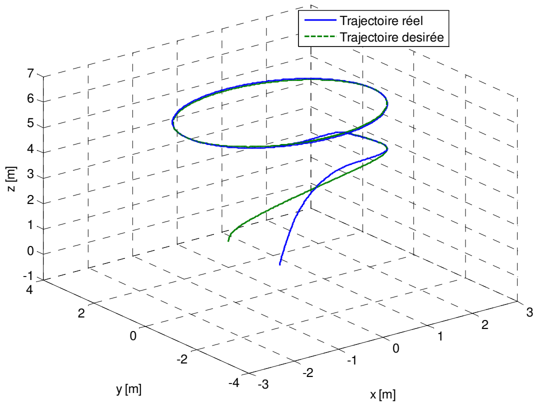

Fig. 12. Actual and desired trajectory in 3D.

Advanced control strategies, including independent Type-1 and Type-2 FLSs, have contributed to enhancing their performance in various applications. This study proposed an intelligent control strategy that starts from an initial optimized PD-Type 1 and obtains a Type 2 fuzzy controller to enhance control performance. The proposed approach relies on offline optimization of the free parameters of a Type-1 FLC, which are input and output scaling factors, to improve efficiency and generalize it to a Type-2 one. The proposed method follows a two-phase process: first, offline optimization of the free parameters of the Type 1 fuzzy controller using PSO optimization to determine optimal scaling factors. The parameters obtained were incorporated into an online PD-Type 1 fuzzy controller. Then, this optimized controller is extended to a PD-Type 2 fuzzy controller. In other words, the Type-1 membership functions are generalized to the Type-2 membership functions after the PSO step. The results underscore the effectiveness of this fuzzy dual-phase strategy for the control of quadrotors, demonstrating its potential to achieve precise and efficient control.

TABLE II. IAE COMPARISON BETWEEN T1FLCS AND T2FLCS FOR THE ATTITUDE CONTROLLER

IAE attitude	Without noise		Noise SNR=20		Noise SNR=25		Noise SNR=30	
	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
Roll	0.0852	0.0723	12.6850	10.3823	4.3744	3.2628	1.3556	1.1019
Pitch	0.0852	0.0723	11.9228	10.8336	3.8549	3.0359	1.4438	1.1740
Yaw	2.37e-05	3.85e-29	39.0753	23.7193	20.6378	7.6324	5.0893	3.0645
z	0.6837	0.3051	0.5969	0.3872	0.7592	0.3539	0.6735	0.2943

TABLE III. IAE COMPARISON BETWEEN T1FLCS AND T2FLCS FOR THE POSITION CONTROLLER

IAE position	Without noise		Noise SNR=20		Noise SNR=25		Noise SNR=30	
	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
Roll	0.1918	0.3830	15.4424	14.0746	4.8835	3.9786	1.5155	1.3953
Pitch	0.3065	0.4993	16.3995	14.3733	5.3764	4.8229	1.9291	1.8222
Yaw	2.37e-05	4.10e-29	66.4258	26.9785	11.0257	7.7021	3.9168	3.0103
x	3.5214	0.9987	10.4059	1.0228	5.6180	1.2011	3.2633	1.6821
y	15.3499	11.7644	38.4467	14.4011	16.7570	12.4340	15.5740	13.3004
z	0.6292	0.1461	1.1118	1.1510	0.5940	0.2382	0.6301	0.1462

The simulation results demonstrated the superior performance of the Type-2 fuzzy controller compared to its Type-1 counterpart, particularly in stochastic environments with noise present. This conclusion was further validated through a brief comparative study based on the IAE criterion (Tables II and III), corroborating the effectiveness of Type-2 fuzzy logic in handling uncertainties.

Future work should consider more profound optimization by considering other free parameters such as those of membership functions (centers, widths, shapes), explore real-time implementation, and consider other adaptation mechanisms.

REFERENCES

[1] Z. Jinlong *et al.*, "Control Design of the Quadrotor Aircraft based on the Integral Adaptive Improved Integral Backstepping Sliding Mode Scheme," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17106–17117, Oct. 2024, https://doi.org/10.48084/etasr.8361.

[2] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 2247–2252, https://doi.org/10.1109/ROBOT.2005.1570447.

[3] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Washington, DC, USA, 2002, vol. 1, pp. 72–77, https://doi.org/10.1109/ROBOT.2002.1013341.

[4] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[5] K. Chafaa, L. Saidi, M. Ghanai, and K. Benmahammed, "Direct adaptive type-2 fuzzy control for nonlinear systems," *International Journal of Computational Intelligence and Applications*, vol. 06, no. 03, pp. 389–411, Sep. 2006, https://doi.org/10.1142/S1469026806001897.

[6] K. Chafaa, L. Saidi, M. Ghanai, and K. Benmahammed, "Indirect adaptive interval type-2 fuzzy control for nonlinear systems," *International Journal of Modelling, Identification and Control*, vol. 2, no. 2, pp. 106–119, Jan. 2007, https://doi.org/10.1504/IJMIC.2007.014623.

[7] H. V. Nguyen, F. Deng, and T. D. Nguyen, "Optimal FLC-Sugeno Controller based on PSO for an Active Damping System," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12769–12774, Feb. 2024, https://doi.org/10.48084/etasr.6662.

[8] A. Mokhtari, A. Benallegue, and B. Daachi, "Robust feedback linearization and GH/sub /spl infin// controller for a quadrotor unmanned aerial vehicle," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1198–1203, https://doi.org/10.1109/IROS.2005.1545112.

[9] T. P. Nascimento and M. Saska, "Position and attitude control of multi-rotor aerial vehicles: A survey," *Annual Reviews in Control*, vol. 48, pp. 129–146, Jan. 2019, https://doi.org/10.1016/j.arcontrol.2019.08.004.

[10] A. C. Satici, H. Poonawala, and M. W. Spong, "Robust Optimal Control of Quadrotor UAVs," *IEEE Access*, vol. 1, pp. 79–93, 2013, https://doi.org/10.1109/ACCESS.2013.2260794.

[11] E. Nechadi, "Adaptive Fuzzy Type-2 Synergetic Control Based on Bat Optimization for Multi-Machine Power System Stabilizers," *Engineering, Technology & Applied Science Research*, vol. 9, no. 5, pp. 4673–4678, Oct. 2019, https://doi.org/10.48084/etasr.2970.

[12] Y. Zhang, Z. Fang, and H. Li, "Extreme Learning Machine Assisted Adaptive Control of a Quadrotor Helicopter," *Mathematical Problems in Engineering*, vol. 2015, no. 1, 2015, Art. no. 905184, https://doi.org/10.1155/2015/905184.

[13] Y. Jiang, C. Yang, S. Dai, and B. Ren, "Deterministic learning enhanced neutral network control of unmanned helicopter," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, Dec. 2016, Art. no. 1729881416671118, https://doi.org/10.1177/1729881416671118.

[14] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, Sep. 1999, https://doi.org/10.1109/91.811231.

[15] C. Zhang, J. Li, and Z. Gao, "Attitude control of quadrotor UAV based on fuzzy PID control under small disturbance," *Highlights in Science, Engineering and Technology*, vol. 53, pp. 199–207, Jun. 2023, https://doi.org/10.54097/hset.v53i.9725.

[16] A. M. Abitha and A. Saleem, "Quadrotor Modeling Approaches and Trajectory Tracking Control Algorithms: A Review," *International Journal of Robotics and Control Systems*, vol. 4, no. 1, pp. 401–426, Apr. 2024, https://doi.org/10.31763/ijrcs.v4i1.1324.