



Optimal trajectory tracking control of unmanned aerial vehicle using ANFIS-IPSO system

Boumediene Selma¹ · Samira Chouraqui¹ · Hassane Abouaïssa²

Received: 13 April 2019 / Accepted: 25 January 2020 / Published online: 2 March 2020
© Bharati Vidyapeeth's Institute of Computer Applications and Management 2020

Abstract Accurate and precise trajectory tracking is crucial for unmanned aerial vehicle (UAVs) to operate in disturbed environments. This paper presents a novel tracking hybrid controller for a quadrotor UAV that combines the robust adaptive neuro-fuzzy inference system (ANFIS) controller and Improved Particle Swarm Optimization algorithm (IPSO) model based on functional inertia weight. The controller is implemented in a three degrees of freedom (3 DOF) quadrotor symbolized with its non-linear dynamical mathematical model. To achieve Cartesian position trajectory tracking capability, the construction of the controller can be divided into two stages: a regular ANFIS controller to guarantee fast convergence rapidity and IPSO aims to facilitate convergence to the ANFIS's optimal parameters to accurately reproduce a desired reference trajectory. Simulation results are given to confirm the advantages of the proposed intelligent control, compared with ANFIS and PID control methods.

Keywords Unmanned aerial vehicle (UAV) · Robust control · Adaptive neuro-fuzzy inference system (ANFIS) · Improved particle swarm optimization (IPSO)

✉ Boumediene Selma
boumediene.selma@univ-usto.dz

Samira Chouraqui
s_chouraqui@yahoo.fr

Hassane Abouaïssa
hassane.abouaïssa@univ-artois.fr

¹ Département D'Informatique, Université Des Sciences Et de La Technologie D'Oran USTO'MB, 31000 Oran, Algérie

² Laboratoire de Génie Informatique Et D'Automatique de L'Artois (LGI2A), Univ. Artois, EA 3926, 62400 Béthune, France

1 Introduction

Unmanned Aerial Vehicles (UAVs) commonly called drones are aerial vehicles capable of carrying out a mission more or less autonomously [1]. These vehicles come from all sizes and shapes [2–6]. Their primary function is to extend human vision beyond the natural horizon, to perform risky or hostile environments works [7, 8]. They have been mostly used by military for surveillance use or reconnaissance missions [9], without the risk of loss of life. More recently, civilian applications have emerged such as forest fire prevention [10], structures inspection [11, 12], traffic monitoring [13, 14], photography and filmmaking [15], safety and security [16], mapping [17], and agriculture [18, 19].

In recent years, trajectory control has become one of the important problems in the study of the UAV. Advances in computer science and electronic technologies have facilitated development in automation control, and intelligent algorithms, some significant works have been implemented on the trajectory control of UAV. Authors of [20] studied the trajectory tracking control problem of a six-degree of freedom (6-DOF) quadrotor unmanned aerial vehicle (UAV) with input saturation by using a backstepping approach. In [21] authors proposed a robust nonlinear controller which combines the sliding-mode control technique and the backstepping control technique for UAV trajectory tracking. A robust decentralized and linear time-invariant controller is proposed [22] for quadrotors with multiple uncertainties to achieve trajectory tracking. The research [23] presents a detailed comparison between two state-of-the-art model-based control techniques for Micro Air Vehicles trajectory tracking; a linear versus nonlinear Model Predictive Controller is presented and compared. A continuous sliding-mode control strategy for quadrotor

robust tracking under the influence of external disturbances and uncertainties in real-time is proposed in [24]. In [25] a continuous multivariable finite-time output feedback is developed for trajectory tracking control and attitude stabilization of quadrotor helicopters. A PD-type trajectory tracking controlling with the extended state observer is proposed [26] for the quadrotor UAV to deal with the wind disturbance.

The strongest arguments for the use of intelligence controllers is that classical controllers mentioned above require the knowledge of the controlled system in term of a set of algebraic and differential equations, which analytically relate inputs and outputs. However, these models can become complex and may contain parameters which are difficult to measure or may change significantly during operation. These problems can be overcome by using intelligence based control techniques.

Intelligent control refers to approaches to control systems design, modeling, identification, and operation that use artificial intelligence techniques, such as fuzzy logic, neural networks, machine learning, evolutionary computation, and metaheuristics. Intelligent control techniques are often capable of controlling dynamical systems that, because of their complexity, are very difficult to control by other techniques.

Generally, the following two types of intelligence based systems are used for estimation and control of drives, namely:

- Artificial neural networks (ANNs)
- Fuzzy logic systems (FLSs)

Neural networks suffer from the difficulty to deal with imprecise information and the “black box” syndrome that more or less has limited their applications in practice. To overcome this drawback, NNs are integrated with fuzzy inference to form Fuzzy Neural Networks (FNNs). Such hybrid systems combining fuzzy logic, neural networks, optimization techniques, and expert systems are proving their effectiveness in a wide variety of real-world problems.

In recent years, Adaptive Neuro-Fuzzy Inference System (ANFIS), developed by JANG, 1993 [27], has gained more attraction than other types of Fuzzy Inference Systems (FIS), because the results obtained from it are more robust than of the statistical methods [28–40].

The major problem with Neuro-Fuzzy system including ANFIS is the number of rules which increase exponentially as the number of inputs increase. Thus, the more number of rules, the more is the complexity of ANFIS architecture and its computational cost. But, it is also noteworthy that over reducing the number of rules results in the loss of accuracy. To overcome this issue, this research proposes optimization technique using Improved Particle Swarm Optimization algorithm (IPSO). IPSO algorithm is a bio-inspired swarm intelligence optimizer introduced by

Kennedy and Eberhart in 1995 [41]. This algorithm has proved its effectiveness in wide range of applications. The inertia weight is an important parameter of the IPSO. It is responsible for determining the operating results. In IPSO, the model based on functional inertia weight improvement. This optimization technique is used to guarantee the convergence to the optimum ANFIS parameters that permit the controller to reproduce accurately the UAV trajectory and reduce tracking error.

This paper presents an effective method using IPSO for designing an ANFIS controller of type Takagi–Sugeno first order by optimizing the parameters of membership functions and the fuzzy rules. The remaining of this paper is organized as follows: The description of the UAV model and the problem formulation are given in Sect.2. ANFIS system with its architecture and learning algorithm are introduced in Sect. 3. The purpose of Sect. 4 is to present the IPSO algorithm that will later be used for the optimization of ANFIS parameters. Section 5 details the proposed control design and strategy. In Sect. 6, comparisons and numerical simulations results are given in order to demonstrate the optimal effectiveness of the proposed controller. Section 7 gives the conclusions.

2 Modeling of a quadrotor and problem formulation

The class of aerial vehicles considered in this work can be described as rigid bodies using a simplified dynamic model via affine parameterization for a multivariable nonlinear UAV modeled to fly in 2D space.

2.1 Dynamical model

The motion of a quadrotor as a rigid body is described by the motion of a body-fixed frame of reference $FB = (OB, \{b_1, b_2\})$ with respect to an inertial reference frame $FI = (OI, \{y, z\})$, as shown in Fig. 1.

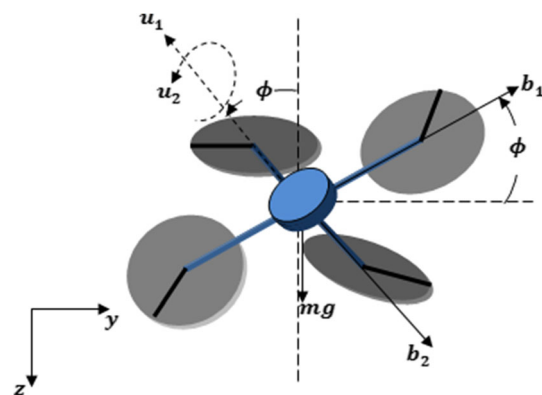


Fig. 1 Quadrotor configuration scheme

The origin located in the center of the quadrotor. It is known that the quadrotor is a 3-DOF object. The Euler angle ϕ represent the orientation of the quadrotor, where it is roll angle about the y -axis. So the quadrotor is modeled to fly in 2 dimensions yz plane with an angle ϕ witch is the roll angle as shown in Fig. 1.

Therefore, the state of the quadrotor is $[y, z, \phi]^T$ and there are two inputs u_1 and u_2 the thrust and the moment about the x -axis respectively.

The equations of motion are described by the following system;

$$\ddot{y} = \frac{u_1}{m} \sin(\phi). \tag{1}$$

$$\ddot{z} = g - \frac{u_1}{m} \cos(\phi). \tag{2}$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}}. \tag{3}$$

where m and I_{xx} are the mass and the moment of inertia of the rigid body, respectively.

It is assumed to be the UAV is near hover condition and commanded roll angle ϕ is calculated based on desired y component and is used to calculate u_2 which is net moment acting on center of gravity (CoG).

The equations of motion have been rewriting as follows;

$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \sin\phi & 0 \\ -\frac{1}{m} \cos\phi & 0 \\ 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4}$$

The state space description of the quadrotor;

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ z \\ \phi \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} \tag{5}$$

The first derivative of the state vector is presented in Eq. (12). The first three elements of the vector represent velocities and the last three elements represent accelerations.

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \dot{z} \\ \dot{\phi} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} \sin\phi & 0 \\ -\frac{1}{m} \cos\phi & 0 \\ 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{6}$$

3 Structure of the Neuro-Fuzzy controller

The structure of the Neuro-Fuzzy controller that implements a Takagi Sugeno type fuzzy inference system (FIS) is shown in Fig. 2. However, for the simplicity of the presentation, we developed an ANFIS with two inputs and only one output. The generalization for the multi-variable case is direct. The structure of the ANFIS consists of five layers:

Layer 1 (fuzzification): In this layer, the nodes are adaptive with a function. The outputs are the fuzzy membership grade of the inputs, which are given by:

$$O_i^1 = \mu_{A_i}(x), \quad i = 1, 2 \tag{7}$$

$$O_j^1 = \mu_{B_j}(y), \quad j = 1, 2 \tag{8}$$

Fuzzifying the inputs is conducted by MF such as Piecewise linear, triangular, trapezoidal, Gaussian and Singleton. Among the abovementioned MFs, this paper has used the gaussian function because of its smooth and concise notation. Therefore, as $\mu_{A_i}(x)$, given that

$$\begin{aligned} \text{Triangular: } \mu_{A_i}(x) &= \max\left(\min\left(\frac{x - a_i}{b_i - a_i}, \frac{c_i - x}{c_i - b_i}\right), 0\right), i \\ &= 1, 2 \end{aligned} \tag{9}$$

$$\begin{aligned} \text{Trapezoidal: } \mu_{A_i}(x) &= \max\left(\min\left(\frac{x - a_i}{b_i - a_i}, 1, \frac{d_i - x}{d_i - c_i}\right), 0\right), i \\ &= 1, 2 \end{aligned} \tag{10}$$

$$\text{Gaussian: } \mu_{A_i}(x) = \exp\left(-\frac{(x - c_i)^2}{\sigma_i^2}\right), i = 1, 2 \tag{11}$$

where $a_i, a_i, b_i, b_i, c_i, c_i$ and σ_i are the antecedent (premise) parameters.

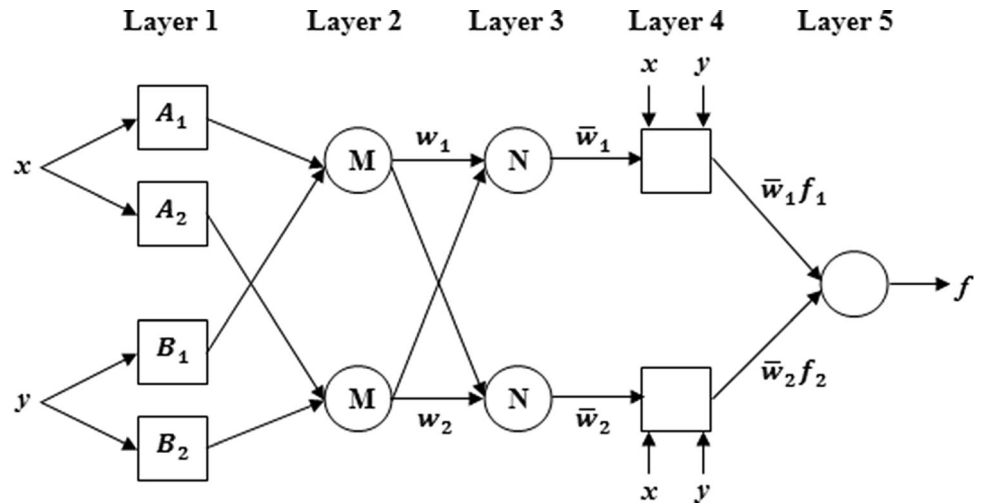
Layer 2 (Weighting of fuzzy rules): The symbol M shows every fixed node in this layer. This layer calculates the firing strength w_k by using membership values computed in fuzzification layer, and the outputs are computed as the following:

$$O_k^2 = w_k = \mu_{(A_i)}(x) * \mu_{(B_j)}(y), \quad i, j = 1, 2. \tag{12}$$

Layer 3 (normalization): Every node is fixed node and called by N. Each node obtains the normalization by calculating the ratio of the k^{th} rule's firing strength to the sum of all rules firing strength. The output O_i^3 at this step is given by:

$$O_k^3 = \bar{w}_k = \frac{\overline{w_k}}{\sum w_i} = \frac{w_k}{w_1 + w_2}, \quad k = 1, 2. \tag{13}$$

Fig. 2 The equivalent typical ANFIS architecture



Layer 4 (defuzzification): Weighted consequent values of rules are calculated in each node of this layer as given in (20).

$$O_k^4 = \bar{w}_k f_k = \bar{w}_k(p_k x + q_k y + r_k), \quad k = 1, 2. \tag{14}$$

where w_k represents the output of the third layer, and $\{p_k, q_k, r_k\}$ are consequent parameters.

Layer 5 (summation): The actual output is obtained by summing the outputs of all incoming signals that coming from the defuzzification layer to produce the overall ANFIS output.

$$O^5 = \sum_{k=1}^2 \bar{w}_k f_k = \frac{\sum_{k=1}^2 w_k * f_k}{w_1 + w_2} \tag{15}$$

where w_k is the output of the third layer, and $\{p_k, q_k, r_k\}$ is the set of parameters. These parameters refer to as the consequent parameters.

4 Particle swarm optimization

Particle swarm optimization (PSO) is evolutionary algorithm belongs to the family of stochastic iterative optimization methods. An essential advantage is that these methods can be applied to both discrete and continuous variable problems. In our case, we wanted to apply the PSO method to the problem of optimizing fuzzy logic controllers that were introduced in Sect. 3 and it's entirely described by the boundaries of fuzzy partitions introduced for each performance indicator for the database (continuous variables).

4.1 Introduction to particle swarm optimization

Swarm particle optimization is an optimization technique based on a set of fixed size agents, called particles, moving in the D-dimensional search space. The position of each particle corresponds to a setting of the fuzzy logic controller. The speed of each particle corresponds to the modifications of this parameterization between two simulations. The velocity and position of each particle are randomly initialized. The entire particles move in the search space, and the performance of each particle is regularly evaluated using cost functions. The evolution of each particle allows an intelligent exploration of the research space. All the particles form the swarm, of size S fixed at the beginning.

4.2 Principles of evolution of the particle swarm

We consider a research space of dimension D . A particle i of the swarm is modeled by a vector of position $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and a velocity vector noted $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. Each particle saves in memory the best position by which it has passed, noted $Pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$. All the best positions reached by the swarm are noted by $Gbest_i = (gbest_{i1}, gbest_{i2}, \dots, gbest_{iD})$. At the beginning of the solution search process, all the particles in the swarm are randomly initialized in the search space. Then, at each iteration, any particle of the swarm moves, linearly combining the three components of Eq. (16).

At iteration t , the velocity vector is calculated as follows [42, 43]:

$$v_{ij}(t + 1) = wv_{ij}(t) + c_1 r_1 [pbest_{ij}(t) - x_{ij}(t)] + c_2 r_2 [gbest_{ij}(t) - x_{ij}(t)] \tag{16}$$

with $j \in \{1, 2, \dots, D\}$ and w a constant, called the coefficient of inertia.

The degree of attraction towards the best position of a particle and that of these informants is respectively represented by two coefficients of cognitive confidence c_1 and social c_2 . The good exploration of particles in the search space is guaranteed by the coefficients r_1 and r_2 which are two random numbers drawn uniformly in the interval $[0, 1]$.

Equation (16) is composed of three terms which are: $wv_{ij}(t)$ a displacement physical component of which w is a variable parameter allowing to control the displacement of the particle at the next iteration, $c_1r_1[pbest_{ij}(t) - x_{ij}(t)]$ a cognitive component of displacement and $c_2r_2[gbest_{ij}(t) - x_{ij}(t)]$ a social component of displacement [44]. The position at the iteration $(t + 1)$ of the particle i is then defined by the following equation:

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t), j \in \{1, 2, \dots, D\} \tag{17}$$

As already mentioned, the PSO algorithm is population-based. In fact, all the particles are randomly positioned in

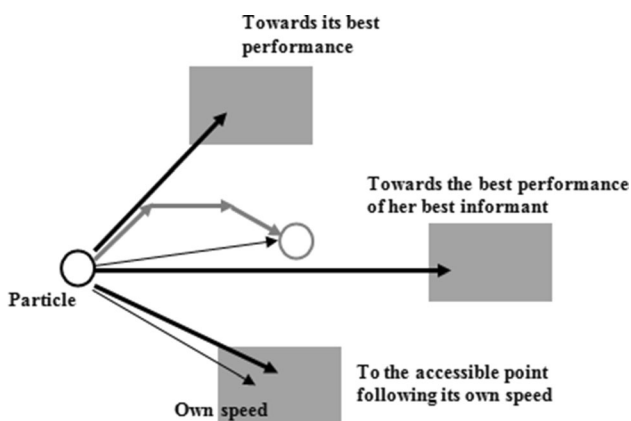
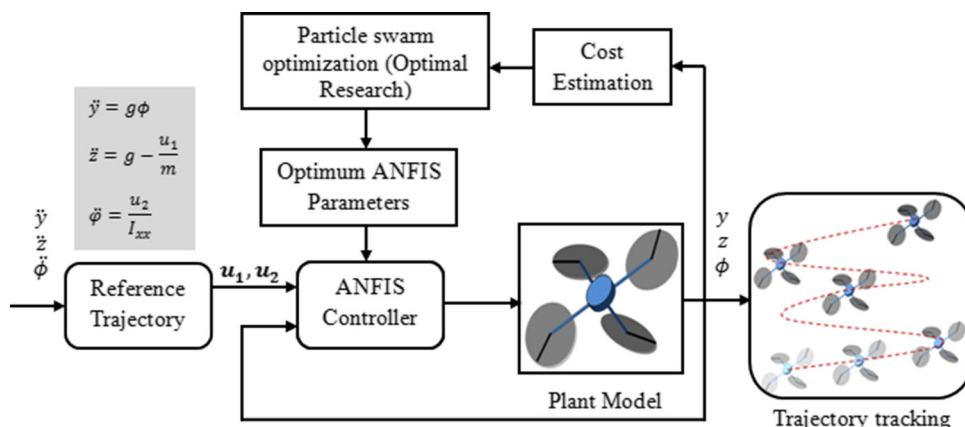


Fig. 3 Determination of the new position of a particle in a PSO process (illustration of the displacement of a particle in the research space by combining the three movement trends)

Fig. 4 The block diagram of the control scheme



the D-dimensional search space. At each iteration, the particles move according to the motion Eqs. (16) and (17), as shown in Fig. 3. Once the position change has taken place, an update affects the two vectors $Pbest_i$ and $Gbest_i$. At the iteration $t + 1$, the latter two vectors will be updated according to the two Eqs. (18) and (19).

$$Pbest_i(t + 1) = \begin{cases} Pbest_i(t), & \text{if } f(x_i(t)) \geq f(Pbest_i(t)) \\ x_i(t), & \text{otherwise} \end{cases} \tag{18}$$

$$Gbest_i(t + 1) = \text{Arg min}_i [Pbest_i(t + 1)] \tag{19}$$

In the literature, the stopping criterion of a PSO algorithm can be defined in different ways. The algorithm is executed as long as one of the three stops criteria, or all at once, of the following convergence criteria are verified [45]:

- The maximum number of iteration defined is not reached;
- The variation of the particle velocity is close to zero;
- The value of the objective is satisfactory with respect to the following relation:

$$|f(gbest_{ij}(t)) - f(gbest_{ij}(t - \alpha))| \leq \epsilon, \text{ with } \alpha \in [1, \Lambda] \tag{20}$$

The parameter ϵ represents a tolerance chosen most often of the order of 10^{-5} and Λ a number of iterations chosen of the order of 10.

4.3 The IPSO algorithm

In this section, the different steps of an IPSO algorithm in the case of single-objective optimization were presented in order:

- At the beginning, all the parameters of the algorithm are declared; the cognitive coefficients c_1 and social c_2 , the inertia factor w , the size of the population that is normally defined according to the optimization problem

- b. Each particle has initial position x_0 and initial velocity v_0 that are randomly initialized to be updated according to Eqs. (18) and (19)
- c. Evaluate all the particles of the population in the search space to be able to determine the vectors of the best initial positions P_{best_0} and G_{best_0}
- d. Calculate the velocities and positions of the particles at each iteration according to the motion Eqs. (16) and (17). Then, evaluate the positions of the particles and update the sizes P_{best} and G_{best}
- e. Repeat the fourth step as long as the stopping criterion is not met

Inertia weight is an important parameter of the standard IPSO. It determines the operating results of IPSO. Fixed inertia weight make the particles always have the same exploration competence in flight. Currently, the conventional strategy of improving inertia weight is LDIW (Liner Decreasing Inertia Weight) [46].

Table 1 Parameters of quadrotor

Symbol	Value
m	0.2 kg
I_{xx}	0.1/kg.m ²
g	9.81 m/s ²

Fig. 5 Measured $z(t)$ trajectory with PID controller

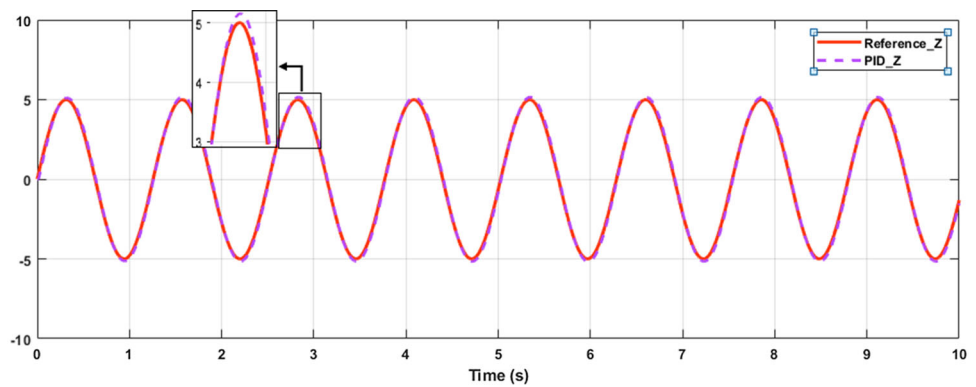


Fig. 6 Measured $y(t)$ trajectory with PID controller

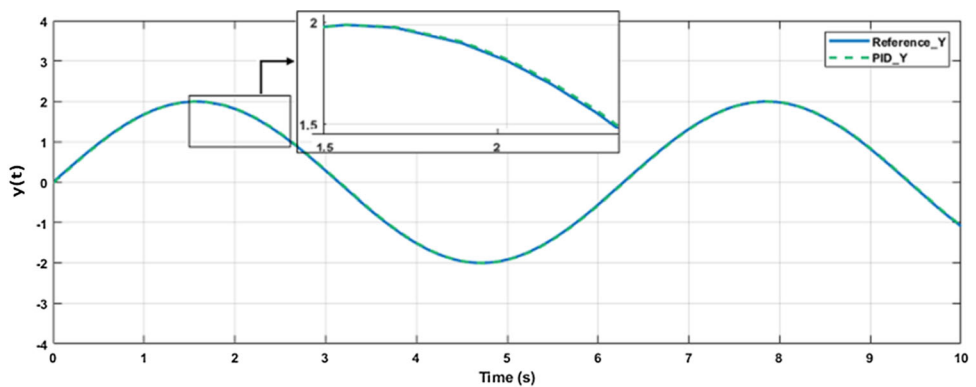
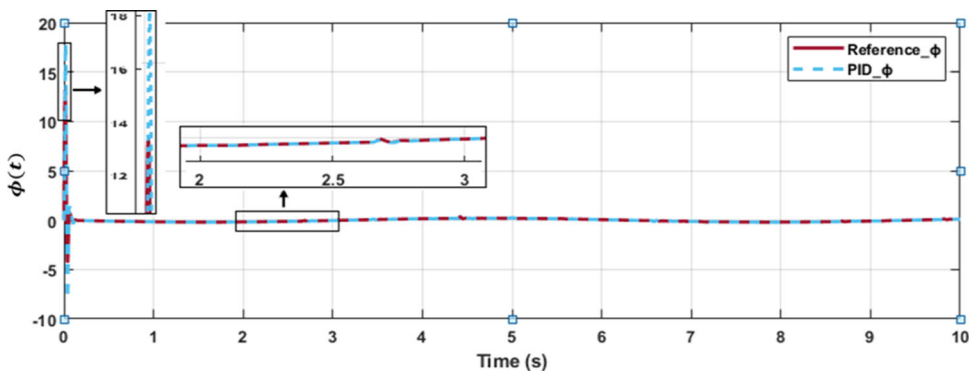


Fig.7 Measured $\phi(t)$ trajectory with PID controller



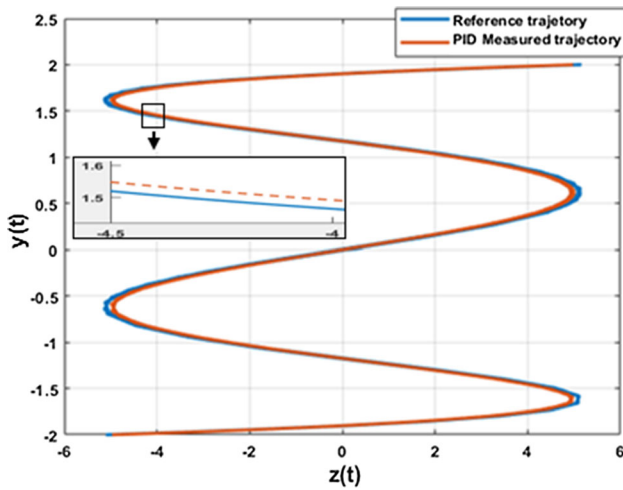


Fig. 8 Measured trajectory with PID controller

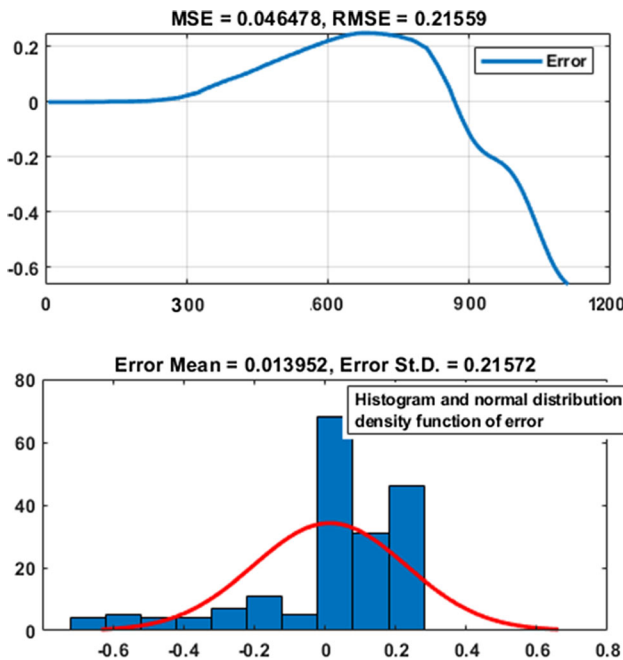


Fig. 9 Trajectory Error with PID Controller

5 Controller design

5.1 ANFIS modeling

In a neuro-fuzzy controller, fuzzy rules are applied to linguistic terms. These terms, which make it possible to qualify a linguistic variable, are defined through membership functions.

First, we define for each variable the discourse universe $[a, b]$ according to the minimum and maximum values of each variable. Second, we define Gaussian-type membership functions that constitute the fuzzy partition of the linguistic variable. It can be seen that this function depends

on the antecedent parameters. For example, for the variable $z(t)$ there are two inputs $(z_r(t), z_c(t))$ and five linguistic inputs that are partitioned on the discourse universe in each input variable. Where $z_r(t)$ is the measured trajectory and $z_c(t)$ is the reference trajectory. Therefore, a typical rule in a Sugeno fuzzy model is described as follows:

If x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$

If $x(z_r(t)) = A_1$ and $y(z_c(t)) = B_1$ then $f_1 = p_1x + q_1y + r_1$. Where $x(z_r(t))$ and $y(z_c(t))$ are linguistic variables, A_1 and B_1 are the linguistic values, f_1 denotes the first output value, and $p_1, q_1,$ and r_1 are the consequent parameters.

The ANFIS is optimized by adjusting the antecedent parameters (membership function parameters) and the consequent parameters (polynomial coefficients) to minimize a specified objective function. The most common and the most applicable fuzzy inference systems training algorithm is the back-propagation learning algorithm that is used to recursively solve the premise and consequent parameters optimization. This conventional optimization algorithm is likely to get stuck at local optima. To overcome this drawback, the use of evolutionary techniques is recommended. The evolutionary technique is used to optimize the antecedent and consequent parameters. In this work, we suggest to use IPSO technique for training the parameters of ANFIS.

The Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) functions are used as a fitness function in order to calculate the fitness of a solution. In order to compute the MSE and RMSE errors values, the output of ANFIS and its real output are used as given in Eqs. (21) and (22):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{21}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_{ii})^2} \tag{22}$$

Here, y_i is the real output, \hat{y}_i is the measured output by ANFIS and n is the number of samples.

5.2 Proposed ANFIS-IPSO

In this research work, in order to improve the training process of the ANFIS, meta-heuristic algorithm was used for optimizing the influential parameters in the ANFIS. For this reason, one of the most commonly used meta-heuristic algorithm is used which is IPSO.

In the ANFIS-IPSO approach, IPSO improves the approximation capability of ANFIS by tuning its parameters. In better words, IPSO minimizes the estimation error by tuning input and output MFs through the search space

Fig. 10 Measured $z(t)$ trajectory with ANFIS controller

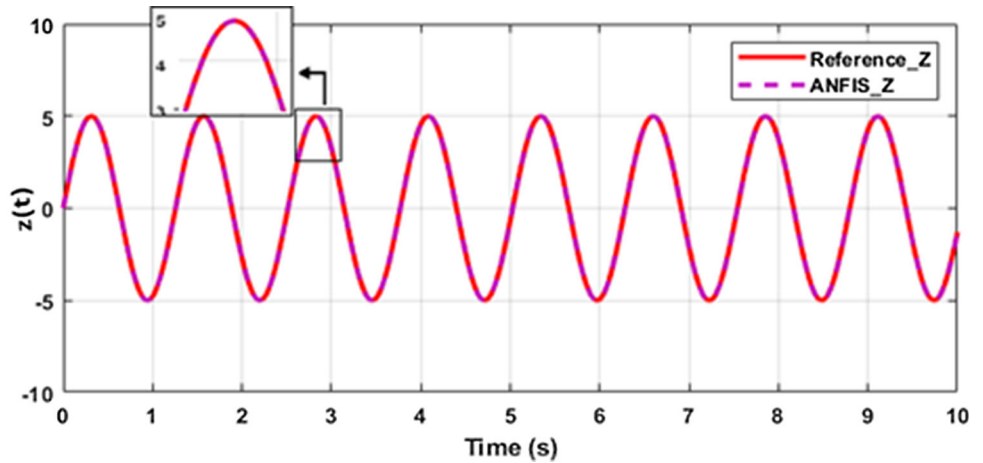


Fig. 11 Measured $y(t)$ trajectory with ANFIS controller

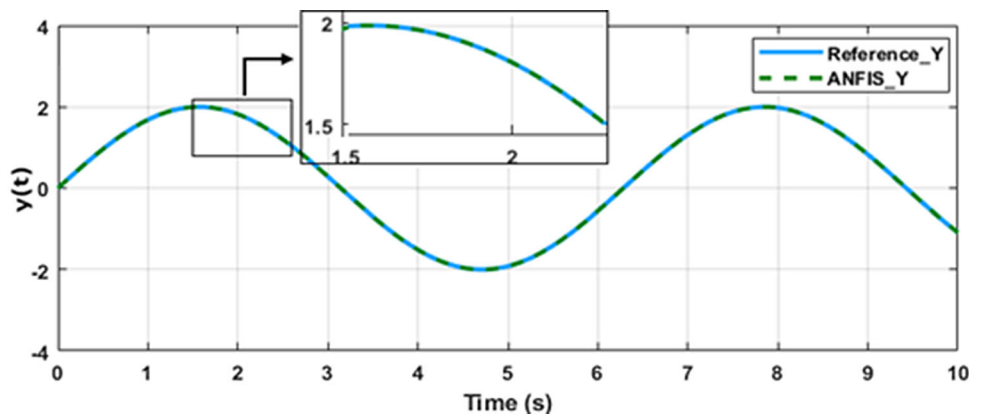
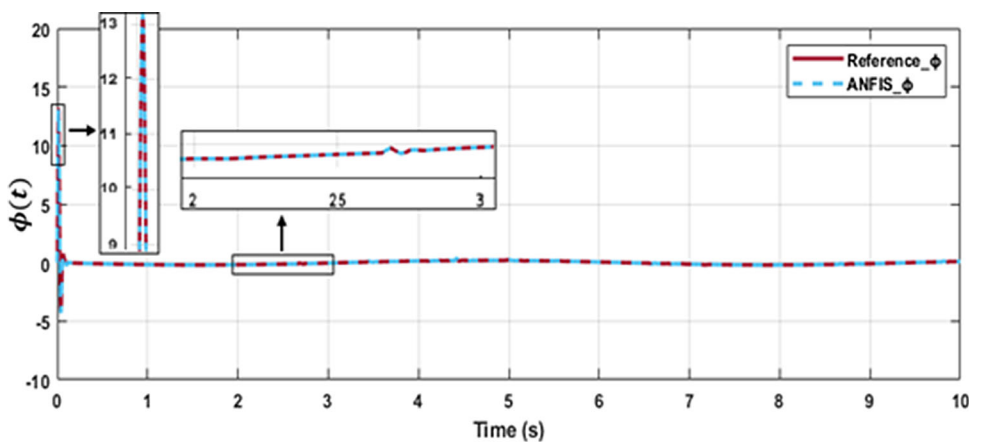


Fig. 12 Measured $\phi(t)$ trajectory with ANFIS controller



provided by ANFIS model. IPSO algorithm optimized MFs of the developed ANFIS models while minimizing both MSE and RMSE. The block diagram showing this structure is given in Fig. 4.

Every learning step of the algorithm contains the learning of antecedent and consequent parameters. The major steps of the combined algorithm are stated as follows.

- (a) Randomly initialize particle swarm in search space, including dimension of search space, particle swarm size, learning factors, maximum number of iterations, and initial speed and position of every particle
- (b) In the first iteration, the initial position of every particle is defined as the initial individual best position. Then, the value of the best position for the whole swarm is determined and the position vector of each particle is used as the antecedent and

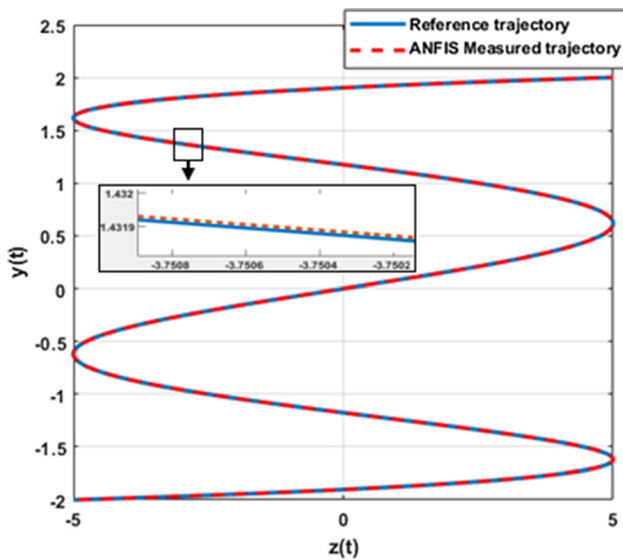


Fig. 13 Measured trajectory with ANFIS controller

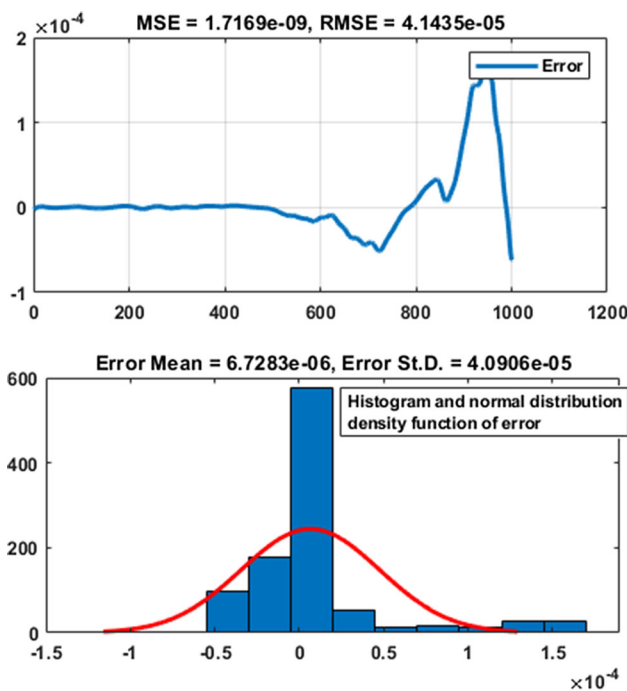


Fig. 14 Trajectory error with ANFIS controller

consequent parameters of ANFIS in sequence. It is followed by calculating the value of the output based on Eqs. (7), (8), (12), (13), (14) and (15). Next, the MSE and RMSE between the model output \hat{y}_i and the reference data for the i^{th} particle is calculated, which is also the fitness value MSE_i and $RMSE_i$ for the i^{th} particle. The fitness values of particles are compared; the particle which has the smallest fitness is selected as the best particle. Its position vector is defined as the initial global best position

- (c) The iterations were continued. For each iteration; the velocity vector v_{ij} and the position vector x_{ij} for each particle are updated based on Eqs. (16) and (17) initially
- (d) Then, the fitness values particles are calculated. Compare the current fitness values $f(x_i(t))$ with the individual best fitness value $f(Pbest_i(t))$ for each particle; if the current fitness value is smaller than $f(Pbest_i(t))$, the individual best fitness value $Pbest_i(t)$ is set as $f(x_i(t))$, and the particle’s historical optimal position along with its new position $Pbest_i(t + 1) = x_i(t)$ is updated. Comparing the particle’s current fitness value $f(x_i(t))$ with the global best fitness value $Gbest_i(t)$, if the current fitness value, $f(x_i(t))$ is smaller than $Gbest_i(t)$, the global best fitness value $Gbest_i(t)$ is set as $f(x_i(t))$, the global optimal position with its new position $Gbest_i(t + 1) = x_i(t)$ is updated, and the number of the best particle is recorded
- (e) Iterations stop when the maximum number of iterations is met. The values of $Gbest_i(t + 1)$ are the identified antecedent and consequent parameters. The parameter w will be changed after each iteration.

6 Simulation results

To validate the approach used, we carried out simulations for the adaptive control of a 3-DOF UAV robot moving along a specified trajectory, whose model parameters are estimated by three ANFIS networks. The adjustment of the network parameters is performed by an IPSO algorithm to obtain the optimal objective value. The UAV is commanded to track a preprogrammed trajectory through a function of time defined by the coordinates $y(t)$ and $z(t)$.

The control performance obtained by the proposed control system is compared with PID and ANFIS controllers to validate the superior performance of the studied controller.

The simulation results are obtained with a vertical trajectory in 2D space. The desired trajectory input is defined as:

$$y_d(t) = 2\sin(t)$$

$$z_d(t) = 5\sin(t)$$

The parameters of the quadrotor used in the following simulations are shown in Table 1.

The PID-based control was used to control the UAV robot for a control purpose. The simulation results are illustrated in Figs. 5, 6, 7 and 8. The reference trajectory

Fig. 15 Measured $z(t)$ trajectory with ANFIS-IPSO controller

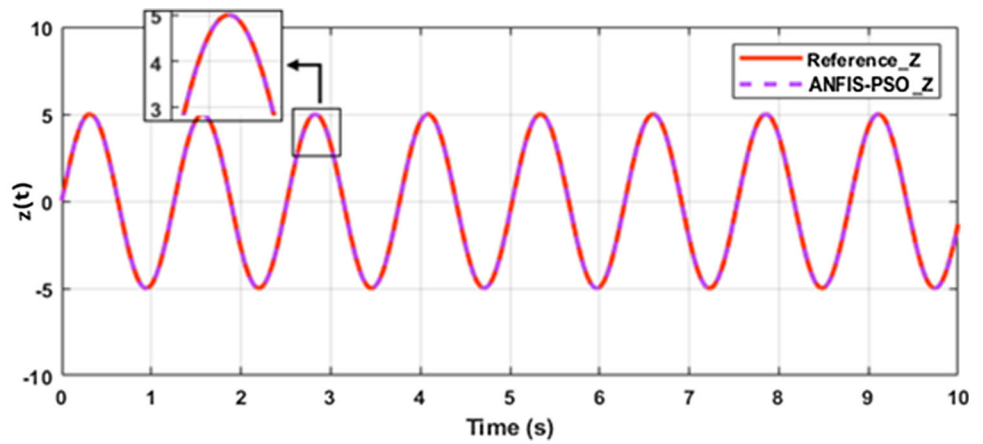


Fig. 16 Measured $y(t)$ trajectory with ANFIS-IPSO controller

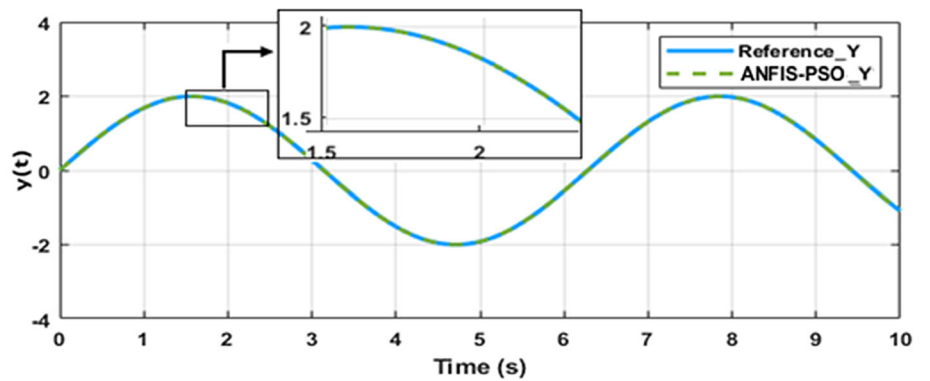
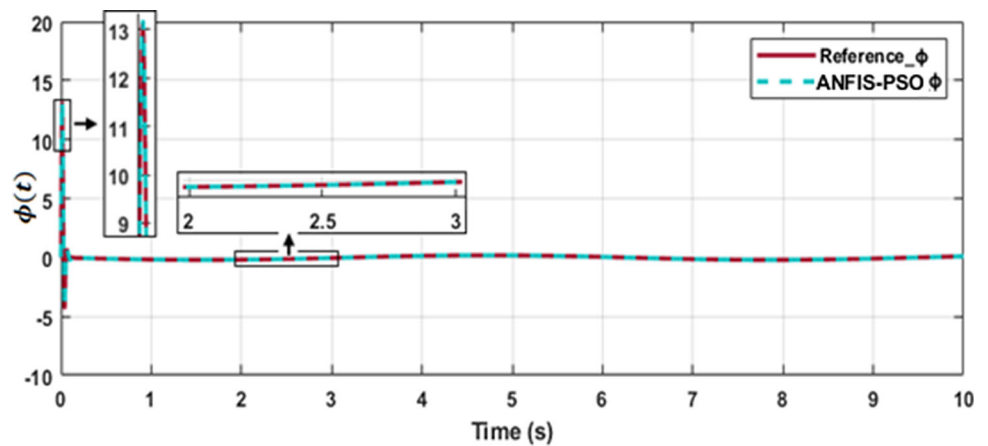


Fig. 17 Measured $\phi(t)$ trajectory with ANFIS-IPSO controller



(blue line) is compared with the measured one (red line) as shown in Fig. 8.

The difference between the calculated trajectory and the desired one can be measured by the mean square error showed in Fig. 9.

As we can see in Fig. 9, the PID controller can reproduce the tracking trajectory with error of $MSE = 4.6478 \cdot 10^{-2}$ and $RMSE = 0.21559$.

The control by ANFIS has proved its worth. The results obtained clearly justify the use of artificial intelligence techniques, of which ANFIS is a part.

Based on the simulation results obtained in Figs. 10, 11, and 12, results showed indicate that respectively the measured $z(t)$ (magenta dotted line), $y(t)$ (green dotted line) and $\phi(t)$ (cyan dotted line) trajectories of the quadrotor track exactly the desired $z(t)$ (red line), $y(t)$ (cyan line) and $\phi(t)$ (brown line) trajectories.

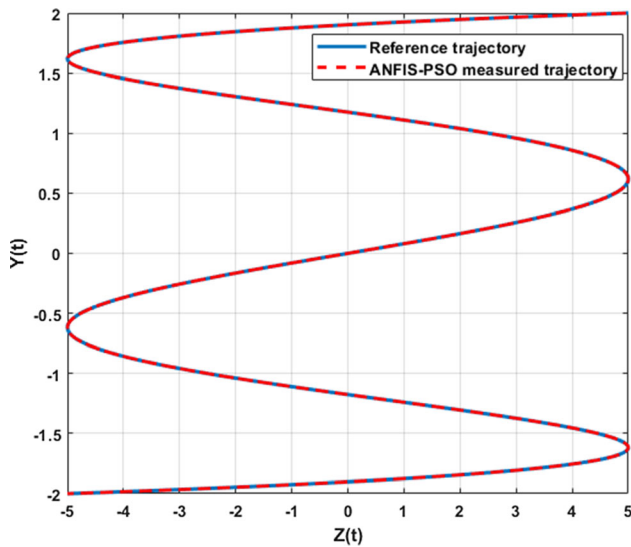


Fig. 18 Measured trajectory with ANFIS-ACO controller

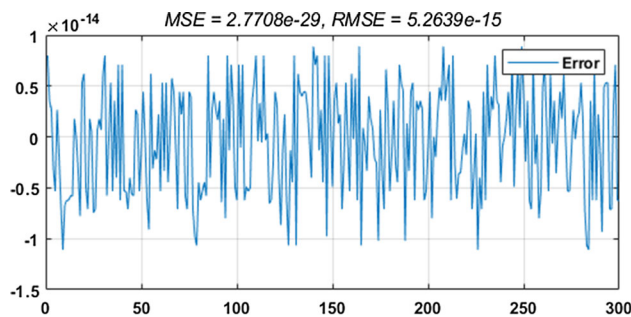


Fig. 19 Error test data (z parameter)

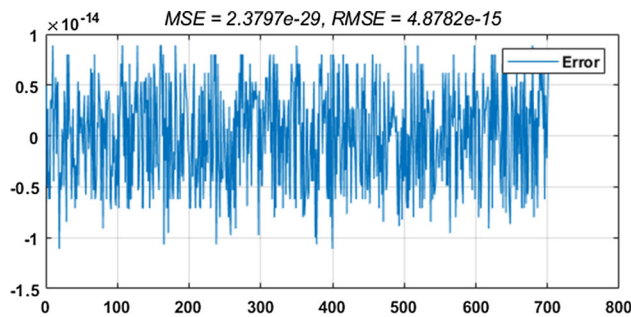


Fig. 20 Error train data (z parameter)

The ANFIS measured $(z(t), y(t))$ trajectory, in Fig. 13 (dotted red line) deviate slightly from the desired (solid blue line) trajectory. The error between the two trajectories showed in Fig. 14, indicates that the calculated mean square error with the ANFIS controller is evaluated at $MSE = 1.7169 \times 10^{-9}$ and $RMSE = 4.1435 \times 10^{-5}$, which is too small than the MSE with PID controller which approve the effectiveness of the ANFIS controller over PID controller.

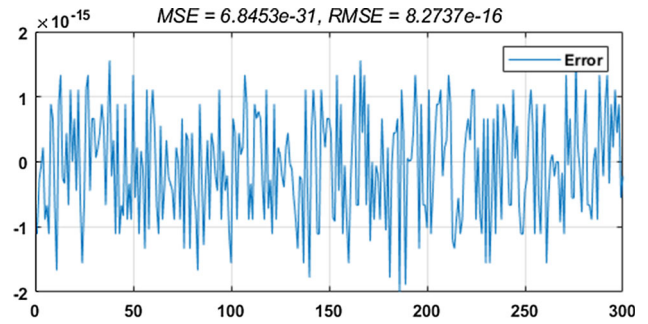


Fig. 21 Error test data (y parameter)

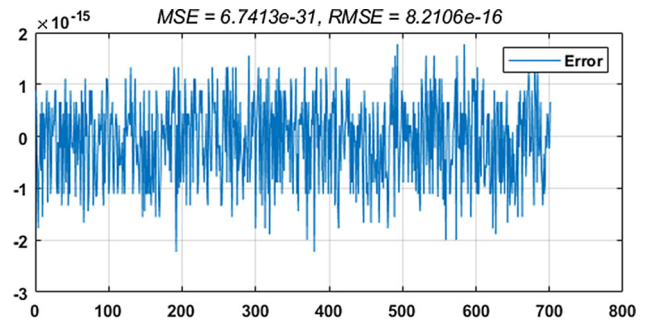


Fig. 22 Error train data (y parameter)

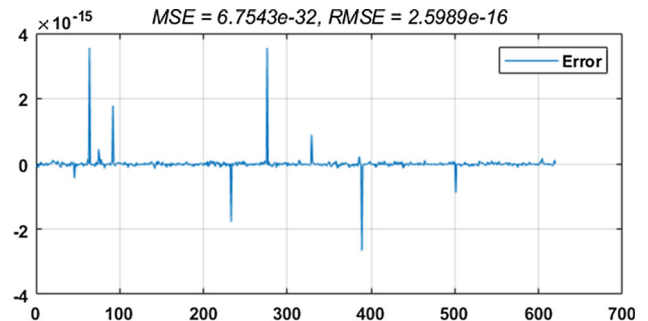


Fig. 23 Error train data (ϕ parameter)

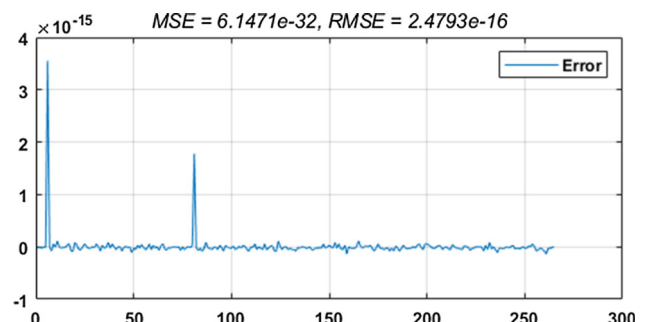


Fig. 24 Error test data (ϕ parameter)

Figures 15, 16 and 17 show the ANFIS-IPSO simulation results of the parameters $y(t)$, $z(t)$ and $\phi(t)$. From the figures, it is very clear that the measured quadrotor trajectories follow accurately the desired trajectories. The tracking

performance is illustrated in Fig. 18, where the measured trajectory (red dotted line) of the quadrotor and the reference trajectory (solid blue line) are shown together. The results obtained show that the performances of the ANFIS-IPSO hybrid approach are superior to those of the other models. It can be seen that the ANFIS-IPSO makes a very good approximation of the system with a mean error of 2.25183×10^{-16} . The Tracking errors of different trajectories ($z(t)$, $y(t)$, and $\phi(t)$) of the quadrotor are illustrated in Figs. 19, 20, 21, 22, 23, and 24. It can be clearly seen that the trajectory tracking errors converge to zero.

7 Conclusion

This research was dedicated to solving the limitations of classical control laws in the absence of model parameters. The use of control methods based on artificial intelligence techniques has become a necessity. These techniques include neural networks, fuzzy logic and evolutionary algorithms among many others.

The paper presents an Improved Particle Swarm Optimization algorithm (IPSO) applied to fitting Adaptive Neuro-Fuzzy Inference System (ANFIS) membership functions. The case study was implemented using simulations, whose main purpose is to control a quadrotor UAV trajectory tracking task.

To validate this intelligent approach, an objective function was used to optimize the ANFIS parameters via the IPSO algorithm. The results obtained were compared with a PID and ANFIS to prove the high performance of ANFIS-IPSO.

Funding The authors received no financial support for the research, authorship, and/or publication of this article.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mostafa SA, Ahmad MS, Mustapha A (2017) Adjustable autonomy: a systematic literature review. *Artif Intell Rev* 1–38
2. Lozano R (2010) *Unmanned Aerial Vehicles: Embedded Control*. Wiley, Hoboken
3. Barnhart R, Hottman S, Marshall D, Shappee E (2012) *Introduction to unmanned aircraft systems*. CRC Press, Taylor & Francis Group, Boca Raton, FL
4. Nonami K, Kendoul F, Suzuki S, Wang W, Nakazawa D (2010) *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. Springer, Berlin
5. Austin R (2010) *Unmanned Aircraft Systems: UAVS Design Development and Deployment*. Wiley, New York
6. Valavanis KP (2007) *Advances in unmanned aerial vehicles: state of the art and the road to autonomy*. Springer, Dordrecht
7. De Filippis L, Guglieri G, Quagliotti F (2011) A minimum risk approach for path planning of UAVs. *J Intell Robot Syst* 61(1):203–219
8. Primatesta S, Guglieri G, Rizzo A (2018) A risk-aware path planning method for unmanned aerial vehicles. In: *International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE pp 905–913
9. Mahadevan P (2010) The military utility of drones. *CSS Anal Secur Policy* 1–3
10. Zhang L, Mu L, Liu Z, Zhang Y, Ai J (2018) Automated maneuvering decision for UAVs in forest surveillance and fire detection missions. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp 1085–1090
11. Ma L, Li MC, Wang YF, Tong LH, Cheng L (2013) Using high-resolution imagery acquired with an autonomous unmanned aerial vehicle for urban construction and planning. In: *Proceeding International Confress on Remote Sensing, Environment and Transportation Engineering (RSETE 2013)*, Njing, China, 31, 200–203
12. Knyaz VA, Chibunichev AG (2016) Photogrammetric techniques for road surface analysis. In: *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Prague, Czech Republic, 515–520. <https://doi.org/10.5194/isprs-archives-XLI-B5-515-2016>
13. Kanistras K, Martins G, Rutherford MJ, Valavanis KP (2014) Survey of Unmanned Aerial Vehicles (UAVs) for traffic monitoring. In: *Handbook of Unmanned Aerial Vehicles*. Springer, pp 2643–2666. <https://doi.org/10.1109/ICUAS.2013.6564694>
14. Chow JYJ (2016) Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy. *Int J Transp Sci Technol* 5(3):167–185
15. Reinartz P, Lachaise M, Schmeer E, Krauss T, Runge H (2006) Traffic monitoring with serial images from airborne cameras. *ISPRS J Photogramm Remote Sens* 61:149–158. <https://doi.org/10.1016/j.isprsjprs.2006.09.009>
16. European Aviation Safety Agency (2016) *Prototype*. Commission Regulation on Unmanned Aircraft Operations, Cologne, Germany
17. Siebert S, Teizer J (2014) Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system. *Autom Constr* 41:1–14. <https://doi.org/10.1016/j.autcon.2014.01.004>
18. Valente J, Del Cerro D, Barrientos A, Sanz D (2013) Aerial coverage optimization in precision agriculture management: a musical harmony inspired approach. *Comput Electron Agric* 99:153–159
19. Freeman PK, Freeland RS (2014) Politics and technology: US polices restricting unmanned aerial systems in agriculture. *Food Policy* 49(1):302–311
20. Wang R, Liu J (2018) Trajectory tracking control of a 6-DOF quadrotor UAV with input saturation via backstepping. *J Frankl Inst Eng Appl Math* 355(7):3288–3309
21. Chen F, Jiang R, Zhang K, Jiang B, Tao G (2016) Robust Backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV. *IEEE Trans Ind Electron* 63(8):5044–5056
22. Liu H, Li D, Zuo Z, Zhong Y (2016) Robust three-loop trajectory tracking control for quadrotors with multiple uncertainties. *IEEE Trans Ind Electron* 63(4):2263–2274
23. Kamel M, Burri M, Siegwart R (2017) Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial

- Vehicles. IFAC-PapersOnLine, 50(1), 3463–3469, ISSN 2405–8963, <https://dx.doi.org/10.1016/j.ifacol.2017.08.849>
24. Ríos H, Falcón R, González OA, Dzul A (2019) Continuous sliding-mode control strategies for quadrotor robust tracking: real-time application. *IEEE Trans Ind Electron* 66(2):1264–1272
 25. Tian B, Lu H, Zuo Z, Zong Q, Zhang Y (2018) Multivariable finite-time output feedback trajectory tracking control of quadrotor helicopters. *Int J Robust Nonlinear Control* 28(1):281–295
 26. Wendong G, Jie L, Chengzhi Q, Jing Z (2018) Trajectory tracking control for a quadrotor UAV via extended state observer. *Syst Sci Control Eng* 6(3):126–135. <https://doi.org/10.1080/21642583.2018.1539931>
 27. Jang JSR (1993) ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Trans Syst Man Cybern* 23(3):665–685
 28. Abido MA, Khalid MS, Worku MY (2015) An efficient ANFIS based PI controller for maximum power point tracking of PV systems. *Arab J Sci Eng* 40(9):2641–2651
 29. Ponce P, Molina A, Cayetano I, Gallardo J, Salcedo H, Rodriguez J (2015) Experimental fuzzy logic controller type 2 for a quadrotor optimized by anfis. *IFAC PapersOnLine* 48(3):2435–2441
 30. Barman B, Kanjilal R, Mukhopadhyay A (2016) Neuro-Fuzzy Controller Design to Navigate Unmanned Vehicle with Construction of Traffic Rules to Avoid Obstacles. *Int J Uncertain Fuzziness Knowl Based Syst* 24(3):433–449
 31. Ponce P, Molina A, Cayetano I, Gallardo J, Salcedo H, Rodriguez J, Carrera I (2016) Fuzzy logic sugeno controller type-2 for quadrotors based on anfis. *Nature-inspired computing for control systems*, Springer, Cham pp 195–230.
 32. Solaiappan BS, Nagappan K (2017) AGC for multisource deregulated power system using ANFIS controller. *Int Trans Electr Energy Syst* 27(3):e2270. <https://doi.org/10.1002/etep.2270>
 33. Ounnas D, Ramdani M, Chenikher S, Bouktir T (2017) An efficient maximum power point tracking controller for photovoltaic systems using Takagi-Sugeno fuzzy models. *Arab J Sci Eng* 42:4971–4982. <https://doi.org/10.1007/s13369-017-2532-0>
 34. Kharola A (2017) Design of a hybrid adaptive neuro fuzzy inference system (ANFIS) controller for position and angle control of inverted pendulum (IP) systems. *Int J Fuzzy Syst Appl (IJFSA)* 5(1):27–42
 35. Khatoun S, Nasiruddin I, Shahid M (2017) Design and simulation of a hybrid PD-ANFIS controller for attitude tracking control of a quadrotor UAV. *Arab J Sci Eng* 42(12):5211–5229
 36. Dorzhigulov A, Bissengaliuly B, Spencer BF Jr, Kim J, James AP (2018) ANFIS based quadrotor drone altitude control implementation on Raspberry Pi platform. *Analog Integr Circuits Signal Process* 95(3):435–445
 37. Aldair AA, Obed AA, Halihal AF (2018) Design and implementation of ANFIS-reference model controller based MPPT using FPGA for photovoltaic system. *Renew. Sustain. Energy Rev.* 82(Part 3):2202–2217, <https://doi.org/10.1016/j.rser.2017.08.071>
 38. Ismail MM, Bendary AF (2020) Smart battery controller using ANFIS for three phase grid connected PV array system. *Math Comput Simulation* 167:104–118
 39. Gaur P, Singh B, Mittal AP, Arora S (2019) Implementation of ANFIS Controller-Based Algorithm Measuring Speed to Eliminate RDC Hardware in Resolver-Based PMSM. In: Singh S, Wen F, Jain M (eds) *Advances in System Optimization and Control Lecture Notes in Electrical Engineering*. Springer, Singapore, pp 147–159
 40. Dixit TV, Yadav A, Gupta S, Abdelaziz AY (2019) Power Extraction from PV Module Using Hybrid ANFIS Controller. In: Precup RE, Kamal T, Zulqadar Hassan S (eds) *Solar Photovoltaic Power Plants Power Systems*. Springer, Singapore, pp 209–232
 41. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proceedings of the IEEE international conference on neural networks*, Perth, Australia, pp 1942–1948
 42. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, New York, pp 39–43
 43. Eberhart R, Simpson P, Dobbins R (1996) *Computational Intelligence PC Tools*. Academic Press Professional, Inc., San Diego **0-12-228630-8**
 44. Dréo J, Petrowski A, Siarry P, Taillard E (2003) *Métaheuristiques pour l'optimisation Difficile*. Eyrolles, Paris
 45. Clerc M, Siarry P (2009). Une méthode inspirée de comportements coopératifs observés dans la nature : l'optimisation par essaim particulaire. *Revue d'Electricité et d'Electronique REE*, pp 25–32
 46. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: *CEC 99. Proceedings of the congress on IEEE evolutionary computation*, vol 3, pp 1945–1950.