CEAS Aeronautical Journal (2021) 12:69–83
https://doi.org/10.1007/s13272-020-00475-6

ORIGINAL PAPER

Autonomous trajectory tracking of a quadrotor UAV using ANFIS controller based on Gaussian pigeon-inspired optimization

Boumediene Selma[1] · Samira Chouraqui[1] · Belkacem Selma[2] · Hassane Abouaïssa[3] · Toufik Bakir[4]

Received: 10 December 2019 / Revised: 9 September 2020 / Accepted: 21 September 2020 / Published online: 29 October 2020

Abstract

This paper develops a method to tune neuro-fuzzy controllers using metaheuristic optimization. The main purpose of this approach is that it allows neuro-fuzzy controllers to be tuned to achieve global performance requirements. This paper proposes a robust and intelligent control method based on adaptive neuro-fuzzy inference system (ANFIS) and pigeon-inspired optimization (PIO) algorithm to govern the behavior of a three-degree-of-freedom (3-DOF) quadrotor unmanned aerial vehicle (UAV). UAVs are flying platforms that have become increasingly used in a wide range of applications. However, the most recent research has aimed to improve the quality of UAVs control in order to achieve its mission accurately. The quadrotor is chosen due to its simple mechanical structure; nevertheless, these types of UAVs are highly nonlinear. Intelligent control that uses artificial intelligence approach such as fuzzy logic is a suitable choice to better control nonlinear systems. The ANFIS controller is proposed to control the movement of UAV to track a given reference trajectory in 2-D vertical plane. The PIO is used to obtain the ANFIS optimal parameters with the aim of improving the quality of the controller and, therefore, to minimize tracking error. To evaluate the performance of the ANFIS-PIO, a comparison between the proposed controller, ANFIS and proportional–integral–derivative (PID) controllers is illustrated. The results demonstrate that the proposed controller is more effective compared to the other controllers.

Keywords Unmanned aerial vehicle (UAV) · Robust control · Adaptive neuro-fuzzy inference system (ANFIS) · Pigeon-inspired optimization (PIO)

✉ Boumediene Selma
selma.boumediene@yahoo.fr

Samira Chouraqui
s_chouraqui@yahoo.fr

Belkacem Selma
selma.belkacem@yahoo.com

Hassane Abouaïssa
hassane.abouaissa@univ-artois.fr

Toufik Bakir
toufik.bakir@u-bourgogne.fr

[1] Département d'Informatique, Université des Sciences et de la Technologie d'Oran USTO'MB, 31000 Oran, Algeria

[2] Département de Génie Electrique, Faculté des Sciences et de la Technologie, Université de Mostaganem, 27000 Mostaganem, Algeria

[3] Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), Univ. Artois, EA 3926, 62400 Béthune, France

[4] ImViA Laboratory EA 7535, Univ. Bourgogne Franche-Comté, Dijon, France

1 Introduction

Unmanned aerial vehicles (UAVs) are aircrafts without human guidance. The UAVs technology has been continuously evolving with exceptional growth over the last years [1], leading to the emergence of a large number of services offered and potential applications. Drones are not meant to only serve military purposes [2], but have also become widely used in civilian and industrial domain, such as logistics and transportation [3, 4], photography and filmmaking [5], safety and security [6], mapping [7], agriculture [8, 9], monitoring [10, 11], surveillance [12], architecture [13, 14] and many other applications. Its use is increasing in most areas due to their low maintenance cost, ease of deployment, high mobility and hovering ability [15–17].

The aim of this study is to develop a robust and intelligent control method for nonlinear systems classes. This type of control method could estimate the unknown function and make the system controllable without precise knowledge of model. The introduction of expert knowledge into the control system is accomplished by a human operator able to control without knowing the mathematical model of the system. This approach is often referred to intelligent control and can be supplemented by a level of supervision that allows the adjustment of this knowledge according to the current situation of the system. It is in this context that a privileged place is devoted to fuzzy control.

Since the emergence of fuzzy logic, the control philosophy of poorly defined processes or even completely unknown ones that cannot be mathematically modeled has undergone a radical change. This is due to the fact that conventional control laws are replaced by if (condition) then (action) rules. Fuzzy logic is often used in complex systems to overcome the limitations of conventional mathematical tools. However, it has limitations, in particular on the accuracy of information expressed in natural language, providing a certain margin of instability.

In order to overcome these disadvantages, the current trend is to integrate these tools into hybrid architectures to reap the benefits of the advantages of fuzzy logic and artificial neural networks (ANNs). The use of neuro-fuzzy systems in command/control offers the possibility to model a priori knowledge and linguistic decision rules obtained by experts in the field. They take advantage of the capabilities and benefits of fuzzy inference modeled by a parallel architecture.

Among all neuro-fuzzy systems, the adaptive neuro-fuzzy inference system (ANFIS) model has the lowest root mean square error, which is why it is the most widely used model. Various research works show that the adaptive neuro-fuzzy inference system, known as ANFIS, developed by Jang [18] is able to learn quickly the behavior of a system with high precision and even better than other methods, including ANNs.

The major benefit of ANFIS is that it allows integrating the user's knowledge on inputs and outputs. However, the implementation of an ANFIS control system faces some challenges and difficulties, particularly in the choice of optimal parameters. The accuracy of ANFIS-based controller results depends on many factors, including choice of inputs parameters, type and number of membership functions (MFs) and learning algorithm. The ANFIS model establishes a relationship between the input and output of a given process through learning process to determine the optimal distribution of the MFs, i.e., the best partitioning of each variable into fuzzy subsets.

Sugeno-type fuzzy inference system model not only provides more accurate to find relationships between input and output variables but it is also computationally more efficient than the Mamdani type. Mamdani-type fuzzy inference gives an output that is a fuzzy set. Sugeno-type inference gives an output that is either constant or a linear mathematical expression. It has also an advantage that it can be combined with optimization techniques so that the system can effectively adapt to the system's characteristics.

Despite their methodical capabilities, they suffer from the problem of setting/control parameters. A key point is to find a way to automatically adjust the parameters during the execution of the algorithm.

Evolutionary algorithms (EAs) are robust and practical methods to automate the search for good solutions. They are proposed as a way to find solutions close to global optima for complex problems. A wide variety of optimization problems in different areas, in which EAs have been successfully applied, demonstrated its effectiveness. This concept must be applied to improve research and achieve effective convergence.

2 Related works

Various design methods have been widely described in the literature that allow to specify the different parameters of a fuzzy controller. They are mainly based on a learning process, which iteratively defines the best set of parameters for a given fuzzy controller structure. Currently, researchers have focused in particular on the following approaches: (a) optimization of MFs, (b) optimization of fuzzy rules, and (c) simultaneous optimization of MFs and fuzzy rules.

Several researches have been carried out on the optimization of the fuzzy controller MFs using genetic algorithms (GAs) [19–23]. Thrift is the first to introduce a method of optimization of fuzzy rules by GA; he used three bits to encode each rule [24]. In 1993, Lee and Takagi proposed a method for simultaneous optimization of MFs and fuzzy rules [25]. Several methods on the same concept were used in these works [26–29]. Other optimization techniques of tuning fuzzy systems use swarm intelligence [30–36].

Metaheuristics are mostly derived from natural-based metaphors, inspired by some biological or physical processes. For example, particle swarm optimization (PSO) is inspired by the social foraging behavior of swarms of birds or fishes [37], ant colony optimization (ACO) algorithm is inspired by the foraging behavior of real ants in the nature [38], and artificial bee colony (ABC) algorithm is inspired by the foraging behavior of honey bee swarms [39]. New algorithms have also emerged recently, including the firefly algorithm (FA) [40], the bat algorithm (BA) [41], grey wolf optimizer (GWO) [42] and pigeon-inspired optimization (PIO) [43]. PIO algorithm is a novel swarm intelligence optimizer, recently introduced by Duan and Qiao in 2014. This bio-inspired approach comprises two individual operators: map/compass operator and landmark operator. The first focuses on the embodiment of the impact of the sun and magnetic field on navigation; the latter highlights the landmark. Only a few years after its introduction, many variants of PIO algorithms were derived and widely applied to various applications in different areas [44–58]. Despite the effectiveness of the PIO, the ability of global searching is still not optimistic in complex optimization problems. In order to overcome this weakness in PIO, a Gaussian factor is introduced to balance the importance between exploration and exploitation results. In this paper, this Gaussian PIO algorithm (GPIO) is used to determine the optimal distribution of the MFs in an ANFIS controller.

This paper presents an effective method to control an UAV. This technique uses a PIO algorithm to design an ANFIS controller of type Takagi–Sugeno first order by optimizing the MFs parameters and the fuzzy rules. The rest of the paper is organized as follows: The description of the UAV model and the problem formulation are given in Sect. 3. ANFIS system with its architecture and learning algorithm is introduced in Sect. 4. The purpose of Sect. 5 is to present the PIO algorithm that will later be used for the optimization of ANFIS parameters. Section 6 presents autonomous ANFIS MFs tuning using a PIO. Section 7 details the proposed control design and strategy. In Sect. 8, simulation results and comparisons are given to demonstrate the effectiveness, quality and efficiency of the proposed hybrid controller. Section 9 gives the conclusions.

3 Model description and problem formulation

3.1 Description of the quadrotor model

It is primordial to introduce the reference coordinates to describe the full structure of the quadrotor. The quadrotor body is defined in y, z axes frame. As shown in Fig. 1, the origin is located in the center of gravity of the 3-DOF quadrotor. The Euler angle ϕ represents the orientation, defining the roll angle about the horizontal axis.

Therefore, the quadrotor is modeled to fly in two dimensions $y - z$ plane with a roll angle ϕ as shown in Fig. 1.

The state of the quadrotor is therefore $[y, z, \phi]^{\mathrm{T}}$ and there are two inputs u_1 and u_2, which represent the thrust and the moment about the x-axis, respectively.

3.2 Equation of motion

Since our drone is modeled in two dimensions, we have y and z plane and a roll angle ϕ as seen before. The equations describing the movement are written as follows:

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

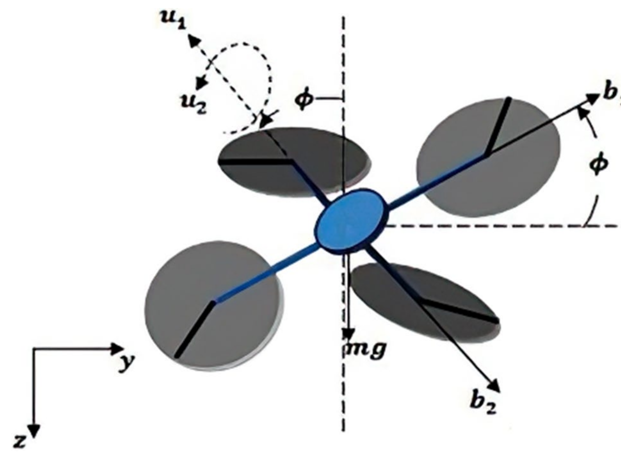


Fig. 1 Quadrotor configuration design

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

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

where m is the mass and I_{xx} is the moment of inertia.

The equations of motion can be written as follows:

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

The state space description of the quadrotor is as follows:

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

So, the first derivative of the state vector is expressed in Eq. (6). The first three parameters of the x vector represent velocities, and the last three parameters represent accelerations.

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

The vector $[u_1, u_2]$ is the input signals that can drive the dynamical system; by specifying the properties of u_1 and u_2, we can change the state of the quadrotor.

4 ANFIS system

The proposed neuro-fuzzy network is a five-layer architecture that includes the elements of a fuzzy Sugeno-type system [18]. Looking at Fig. 2, let's explain how the network works layer by layer.

Layer 1 (Fuzzification): This layer contains adaptive nodes. The outputs are the fuzzy degree of membership of the inputs, which are calculated as follows:

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

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

where μ_{A_i}, μ_{B_j} denote the membership degrees obtained from this layer.

Fuzzifying the inputs is conducted by MF such as piecewise linear, trapezoidal, triangular, singleton and Gaussian. Among the abovementioned MFs, in this paper the Gaussian function has been used for its smooth and concise notation. Therefore, $\mu_{A_i}(x)$ is expressed by;

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

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

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

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

Layer 2 (Weighting of fuzzy rules): This layer contains nodes with symbol M, which indicates fixed node. The firing strength w_k is calculated at this layer by using membership values calculated in the fuzzification layer, and the outputs are computed as follows:

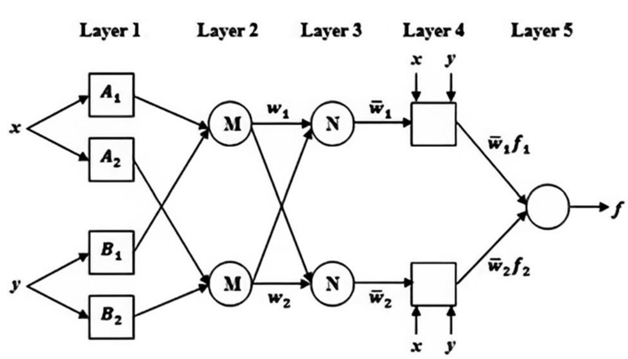


Fig. 2 The equivalent typical ANFIS architecture

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

Layer 3 (Normalization): All nodes in this layer are fixed nodes and called by N. Each node obtains the normalization by computing the ratio of the kth rule's firing strength (truth values) to the sum of all rules firing strength. The output O_k^3 at this step is given by:

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

Layer 4 (Defuzzification): Each node of this layer calculates the weighted consequent values of rules as indicated in Eq. (14).

$$O_k^4 = \overline{w_k} f_k = \overline{w_k}\left(p_k x + q_k y + r_k\right), \quad k = 1, 2. \tag{14}$$

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

Layer 5 (Summation): The output of this layer is calculated by summing the outputs of all incoming signals from the previous layer (defuzzification) to produce the overall ANFIS output.

$$O^5 = \sum_{k=1}^{2} \overline{w_k} f_k = \frac{\sum_{1=1}^{2} w_k f_k}{w_1 + w_2}. \tag{15}$$

5 Gaussian pigeon-inspired optimization algorithm

5.1 Pigeon-inspired optimization

PIO algorithm is a metaheuristic inspired by the natural behaviors of homing pigeons. This new swarm intelligence algorithm has been recently proposed by Duan and Qiao [43]. Pigeons find their way back home using three tools: earth's magnetic field, sun and landmarks. Evidence indicates that pigeons are able to use the earth's magnetic field to find their destination over long distances. According to the researchers, the sun is also involved as a form of sun navigation that uses homing pigeons, since pigeons are able to distinguish differences in altitude between the sun at the home base and at the point of release.

The PIO has two different operators: the map and compass operator and the landmark operator [43, 49]. The two operators are designed by some calculation rules, which can be adapted to many problems.

The map and compass operator and landmark operator are based on magnetic field and sun, and landmarks, respectively.

5.1.1 Map and compass operator

In mathematical PIO model, consider N pigeons in $D-$dimension search space. A pigeon i is modeled by a vector of position $X_i = \left(x_{i1}, x_{i2}, \ldots, x_{iD}\right)$ and a velocity vector noted $V_i = \left(v_{i1}, v_{i2}, \ldots, v_{iD}\right)$.

At iteration $t < T^1_{\max}$, the navigation cues for each pigeon i are provided by the map and compass operator, and the new position X_i and new velocity V_i of pigeon i are updated as follows:

$$V_i(t) = V_i(t-1) \cdot \mathrm{e}^{-St} + R \cdot \left(X_g - X_i(t-1)\right), \tag{16}$$

$$X_i(t) = X_i(t-1) + V_i(t), \tag{17}$$

where t and $t-1$ are the current iteration and the previous iteration, $T^1_{\max}$ is the maximum iteration of the current operator, S is the map and compass factor, R is a random number within [0, 1] and X_g is the current global best position (Fig. 3).

5.1.2 Landmark operator

This operator takes over the navigation system of pigeons when $T^1_{\max} < t < T^2_{\max}$, where $T^2_{\max}$ is the maximum iteration of PIO. In landmark operator, the number of pigeons in every generation is reduced by half N_{P}. The generation of position is given as follows:

$$N_{\mathrm{P}}(t) = \frac{N_{\mathrm{P}}(t-1)}{2}, \tag{18}$$

$$X_{\mathrm{c}} = \frac{\sum X_i(t) \cdot w\left(X_i(t)\right)}{N_{\mathrm{P}} \sum w\left(X_i(t)\right)}, \tag{19}$$

$$X_i(t) = X_i(t-1) + R \cdot \left(X_{\mathrm{c}} - X_i(t-1)\right), \tag{20}$$

where X_{c} is the center (average) of all positions of the pigeons at the tth iteration and the weight $w\left(X_i(t)\right)$ is calculated by Eq. (21),

$$w\left(X_i\right) = \begin{cases} f\left(X_i\right), & \text{for maximization} \\ \frac{1}{f(X_i)+\varepsilon}, & \text{for minimization} \end{cases}, \tag{21}$$

where $f\left(X_i\right)$ is the value of cost (fitness) function of the pigeon i at iteration t and ε is an arbitrary nonzero constant (Fig. 4).

5.2 Gaussian pigeon-inspired optimization

PIO has a fast convergence speed, but to avoid the possibility of premature convergence to local minima. PIO also has the common problem, which is how a poor balance between exploration and exploitation results. To formulate a more efficient model, a Gaussian distribution, or also known as normal distribution, is introduced into the PIO algorithm [59, 60]. It is added to the position iteration.

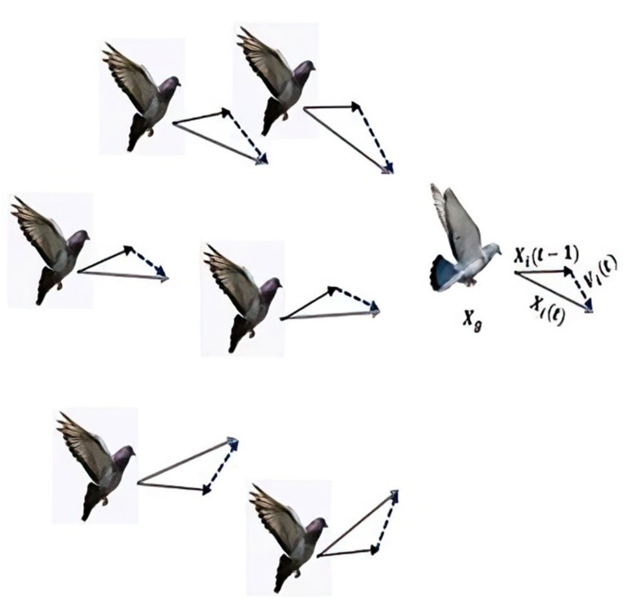


Fig. 3 Map and compass operator model of PIO

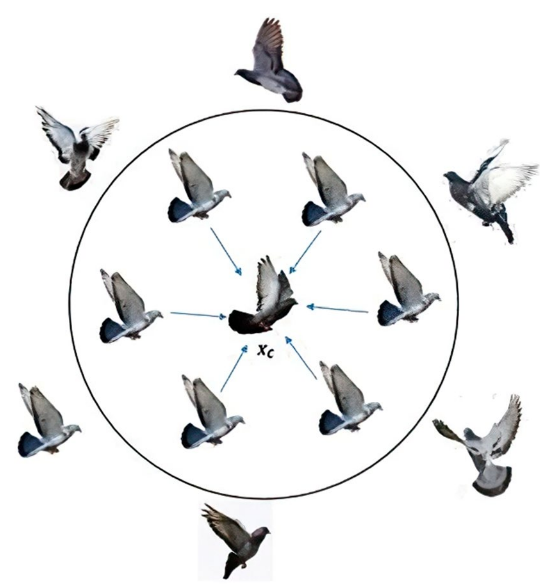


Fig. 4 Landmark operator model

5.2.1 Gaussian distribution

The random number R has an obvious feature that the random output R is a uniform distribution. It has a significant advantage which is the full search capability. In many cases, the optimization algorithm should have the ability to ensure that the destination is correct when carrying out a focused search. The method to get a random number, which satisfy the uniform distribution rule, is not good enough to meet the requirements. The searching equation in the PIO landmark operator satisfies the latent premise of Gaussian distribution, and it can be improved for achieving the global best.

The position update formula of the improved landmark operator is given as:

$$X_i(t) = \begin{cases} X_i(t-1) + 2(R_1 - 0.5)\cdot(X_c - X_i(t-1))\cdot mn & \text{if} \quad (R_2 > p) \\ X_i(t-1) + 2(R_1 - 0.5)\cdot(X_c - X_i(t-1))\cdot 2n & \text{if} \quad (R_2 \le p) \end{cases}. \tag{22}$$

where p is a flexible parameter used in order to balance the Gaussian distribution and the uniform distribution; R_1 and R_2 are two random numbers between 0 to 1.

$$\begin{cases} m = |Q| \\ n = 0.5 - 0.25\frac{t}{T^2_{\max}} \end{cases}, \tag{23}$$

where Q is a random number created according to the Gaussian distribution between 0 and 1; $T^2_{\max}$ is the maximum iteration.

6 Autonomous ANFIS MF tuning using a PIO

ANFIS controllers are designed to use linguistic information from expert knowledge in the form of fuzzy reasoning rules. The rules are expressed in the following form:

if input X is A and input Y is B then output is C.where the premises or antecedents A and B are linguistic terms that correspond to the qualitative values associated with the basic variables X and Y. These linguistic terms are described by fuzzy sets defined by MFs. Figure 5 illustrates a typical set of MFs. The consequent (conclusion) C represents the behavior associated with the conditions of validity described by the antecedent. There are two types of expressions for C, Sugeno and Mamdani. In Sugeno type, C of each rule is given as a linear combination of inputs. However, in a Mamdani type, C is expressed by a set of MFs.

In general, fuzzy rules are expressed with simple ''if–then'' relationship, which are easily obtained from expert knowledge. However, the MF design is a delicate and time-consuming process; it involves the determination of where each MF should be located in the variable space.

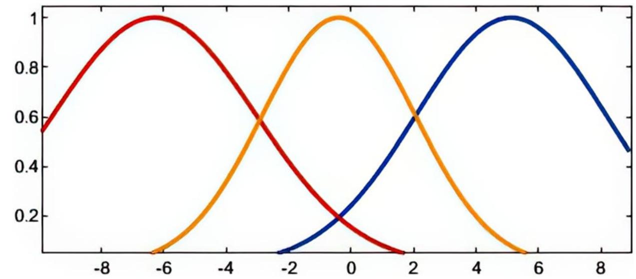


Fig. 5 A typical set of MFs in an ANFIS controller

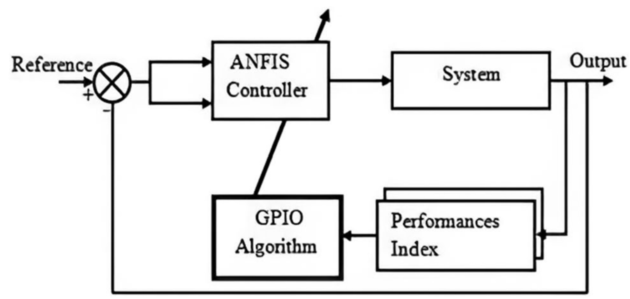


Fig. 6 The proposed method of tuning ANFIS MF by PIO

Figure 5 shows typical forms representative of the Gaussian-type MFs. In this type, each MF is determined by two parameters, which are mean and deviation. As for other types of MFs, such as trapezoidal and triangular, similar parameters also exist for the determination of the MFs shapes. Therefore, the use of PIO as a global optimization method to search a set of these parameters that are optimal improves the best control performance of the ANFIS.

Figure 6 describes the strategy of using a PIO algorithm for MFs tuning in ANFIS controller. In the proposed ANFIS-PIO model, the MF parameters of the inputs and outputs of the ANFIS are represented by each particle. As the aim of the PIO is to minimize the control error of the ANFIS controller, the adopted objective function of PIO is defined by mean-squared error (MSE) and root mean-squared error (RMSE) in order to calculate the fitness of the solution. To compute the MSE and RMSE errors values, the measured output of ANFIS and the desired output are used as given in Eqs. (22) and (23):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2, \tag{24}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}. \tag{25}$$

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

6.1 PIO representation of solutions

The NFIS controller is composed by many input and output MFs. That is why it will require a better use of the subpopulation concept.

The solution for a problem is associated with a population of homing pigeons composed a subpopulation p by vector with r positions $sp = \{p_1, p_2, p_3, \dots, p_r\}$, where each component p_i represents a homing pigeon (agent). Each subpopulation represents one possible solution.

However, each homing pigeon position is composed by the MFs parameters for the antecedents and consequents, which are real values.

The number of agents in each subpopulation (size of the subpopulation) depends on the number of MFs defined by user.

6.2 PIO for ANFIS MF tuning

The integration of PIO algorithm with ANFIS controller is made as follows:

1. The subpopulation is defined as a vector of parameter values of the input and output MFs.
2. The parameters are the antecedents (premises) and consequents (conclusions) of each fuzzy set. Each agent (homing pigeon) is composed by these parameters.
3. To check the performance of the ANFIS, it is rolled up from an initial set of parameters.
4. The information about these parameters is used as input arguments for the implementation of each sub-population adaptability (adjustment) and the achievement of the evolution of the population.
5. The cycle is repeated to complete the defined number of iterations performed by the user. The best value set for the MFs parameters is found at the end of each PIO iteration.

7 Controller design

An adaptive controller represented by an automatically tuning ANFIS by a PIO algorithm for a better and optimal choice of ANFIS MFs parameters. The adaptation law of these parameters is responsible for reducing tracking errors.

The auto-tuning (self-optimization) of ANFIS parameters by PIO specifically aims to find the optimal MFs design. This leads to automatically adjusting the fuzzy rules because the two parts (MFs and fuzzy rules) cannot be dissociated. Moreover, the optimization by the PIO, whose sole objective is the improvement of a numerical criterion, often leads to the presence of more precise fuzzy rules at the end of the optimization process, which leads to better results. It is within this framework that our motives and interests lie in the design of neuro-fuzzy controllers by PIO.

In our study, an ANFIS controller applied to a totally autonomous UAV tracking trajectory, modeled to fly in two dimensions (yz) plane with an angle ϕ which is the roll angle, both y and z equations that describe the trajectory and the roll angle ϕ are considered as an objective function. To improve controller, an optimization approach based on PIO algorithm is proposed to obtain the optimal parameters leading to the optimal or ideal trajectory and therefore to completion of the specified movement resets the control.

The trajectory to follow is defined by the vector R_T, which consists of two elements $\{y(t), z(t)\}$ as shown in Fig. 7.

Given the reference trajectory

$$R_T(t) = \begin{bmatrix} y(t) \\ z(t) \end{bmatrix}. \tag{26}$$

And the measured trajectory is defined by the vector R_C

$$R_C(t) = \begin{bmatrix} y_C(t) \\ z_C(t) \end{bmatrix}, \tag{27}$$

where $y_C(t)$ and $z_C(t)$ are the calculated parameters by the controller.

For each parameter defining the trajectory, we consider a controller with two inputs, an output $\Delta u(t)$, error $e(t)$ and its variation $\Delta e(t)$, the variation of the command, which allows to adjust to each moment the command $u(t)$, applied to the system (Fig. 8). The fuzzy rules constituting the base of the controller, in this case, have two premises.

7.1 PIO implementation procedure

According to the UAV modeling in Sect. 2, and to the proposed feedback control, initialize the plant, i.e., trajectory information including the coordinates of the two dimensions y–z plane and the roll angle ϕ.

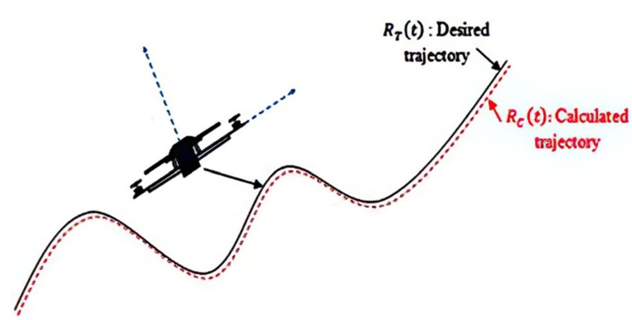


Fig. 7 Quadrotor trajectory tracking in 2D plane

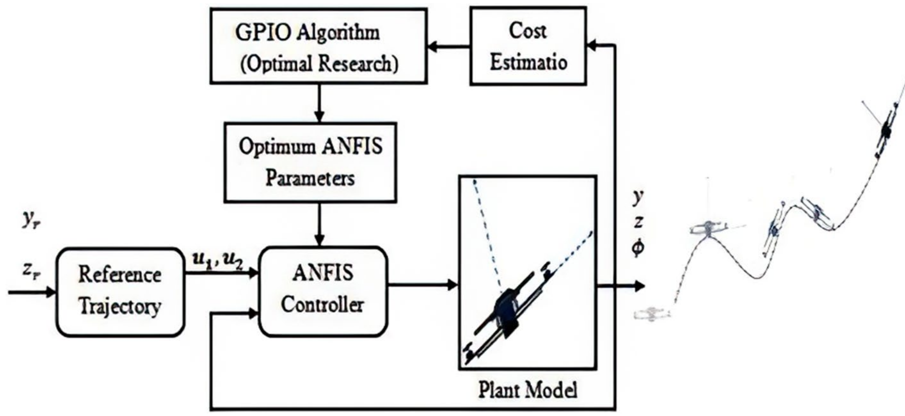


Fig. 8 Optimization ANFIS controller with PIO block diagram

Being $T^{\max}$ the PIO iteration, P is the population, N is the subpopulation number, and $[a, b]$ is the universe of discourse of each fuzzy set (or the input space). As a start, to store the best subpopulation result the algorithm shown below generates the vector global best solution with N positions.

The procedure for implementing the PIO is described as follows.

Step 1: generate the initial subpopulation SP with pigeons in the interval $[a, b]$.

Step 2: initialize the parameters of PIO algorithm: the dimension of the solution space D, population size N, the number of iterations for the two PIO operators $T^1_{\max}$ and $T^2_{\max}$ ($T^2_{\max} > T^1_{\max}$) and the map and compass factor S.

Step 3: define each subpopulation SP with a randomized position and velocity. Find the current best solution by comparing the fitness of each subpopulation SP.

Step 4: operate map and compass operator. Firstly, using Eqs. (16) and (17), update the velocity and position of every agent (pigeon). Then, find the new best solution by comparing all the subpopulation's fitness.

Step 5: if $T^{\max} > T^1_{\max}$, stop the map/compass operator and operate next operator. Otherwise, go to Step 4.

Step 6: rank all subpopulations according to their fitness values. According to Eq. (18), half of subpopulations with low fitness follow the subpopulations with high fitness. Then, calculate the center position of all subpopulations using Eq. (19), and this center represents the desirable destination.

Step 7: Update the position of each individual based on the improved landmark operator according to Eqs. (22) and (23); all subpopulations adjust their flying direction. Next, store the best solution parameters and the best fitness value, and update the global optimum.

Step 8: if $T^{\max} > T^2_{\max}$, stop the landmark operator, and output the results. Otherwise, go to Step 7.

Table 1 Parameters of quadrotor

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

8 Simulation results

In this section, we analyze the performances of our proposed method in order to control an UAV moving along a specified trajectory. The model is implemented in MATLAB/Simulink programming software. A simulation was made for illustrative purpose. The UAV is commanded to

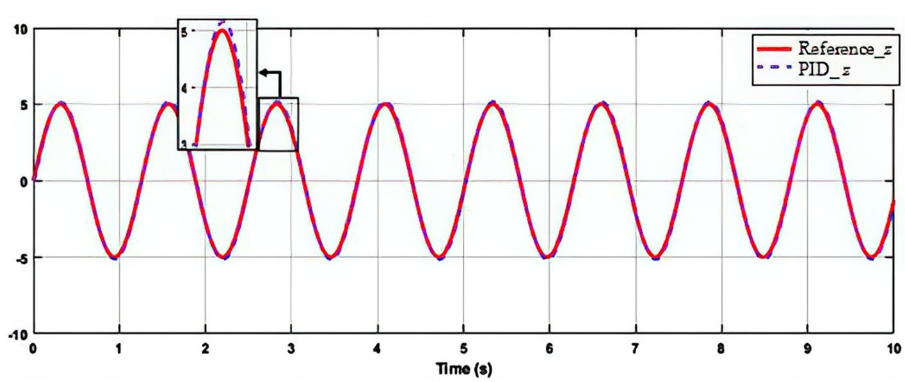


Fig. 9 PID z measured

Fig. 10 PID y measured

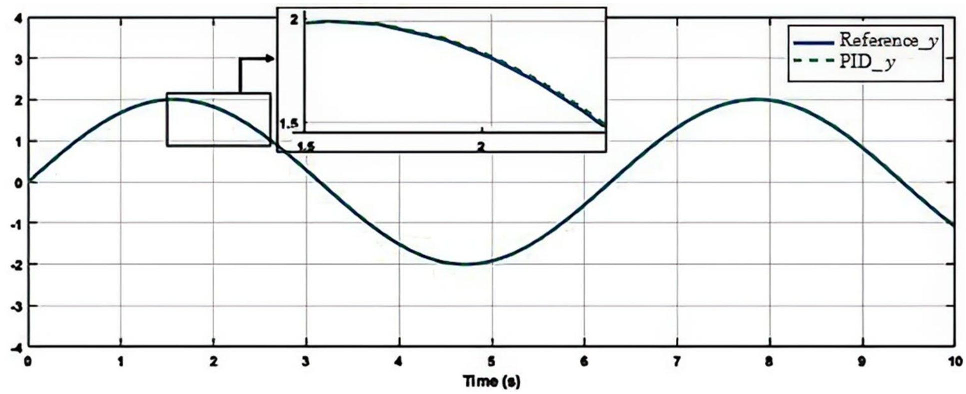
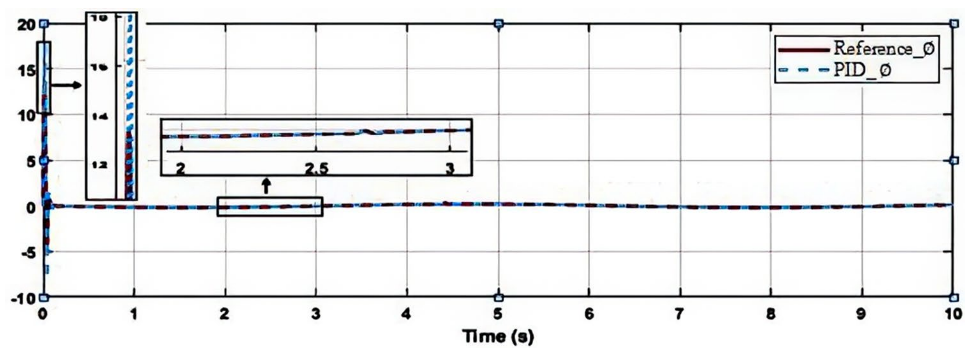


Fig. 11 PID ϕ measured



flight following a pre-defined trajectory as a function of time defined by $y(t)$ and $z(t)$.

To validate the superior performance of the proposed controller, the control system performance obtained by the ANFIS-PIO controller is compared with those obtained by PID and ANFIS controllers.

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

$$y_d(t) = 2\sin(t) \text{ and } z_d(t) = 5\sin(t).$$

Table 1 shows the parameters of the quadrotor used in the following simulations.

Finally, the three control inputs y, z, ϕ were shown below.

Trajectory tracking simulation has been presented in this section, and some simulation results are given to illustrate the control performances of the developed controller. The trajectory tracking responses of the three used controllers are shown in this section, from which it can be observed that the reference trajectory can be tracked effectively in case of the proposed ANFIS-PIO controller.

With the evaluation of the results, it's seen that ANFIS-PIO controller has successfully followed the reference but PID controller has given the poor results compared to other controllers.

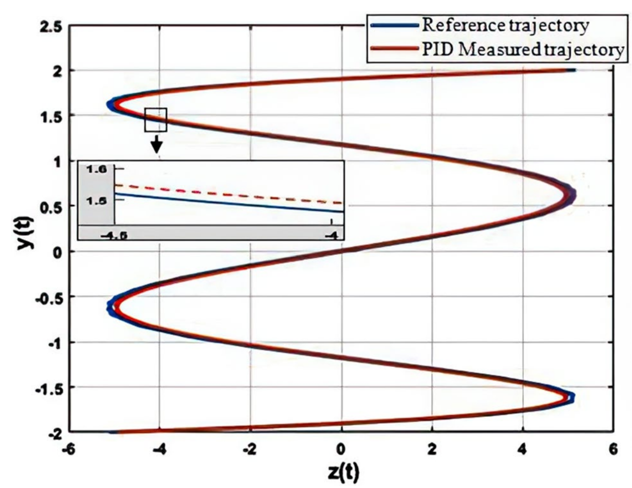


Fig. 12 PID trajectory measured

Figures 9, 10, 11 and 12 show the PID results; this controller can reproduce the tracking trajectory with error of MSE = 6.42×10^{-2} and RMSE = 0.25.

The simulation result obtained by the ANFIS is given in Figs. 13, 14, 15 and 16. It can be seen that realizes a good approximation of the system with an error MSE = 5.47×10^{-10} and RMSE = 2.34×10^{-5}.

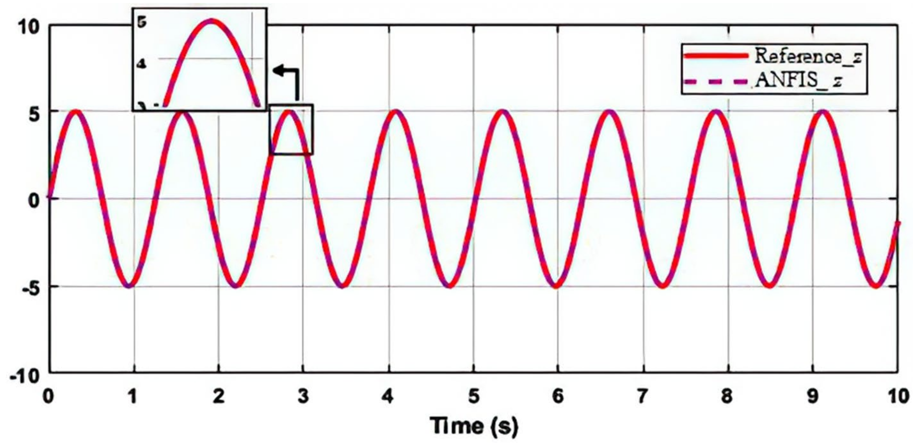


Fig. 13 ANFIS z measured

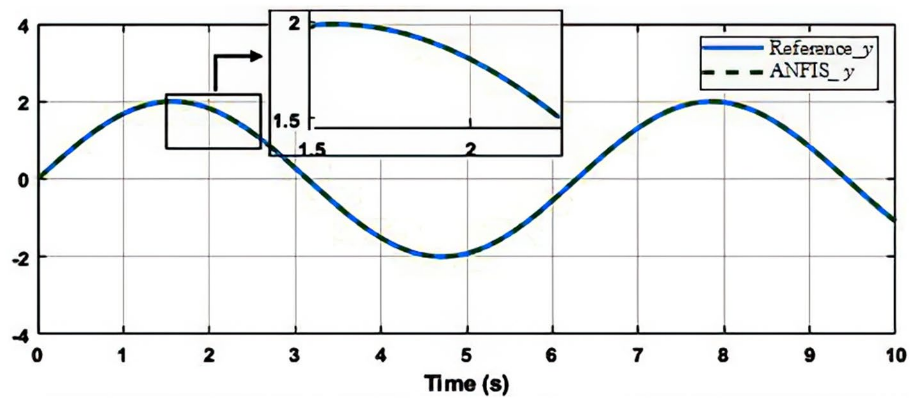


Fig. 14 ANFIS y measured

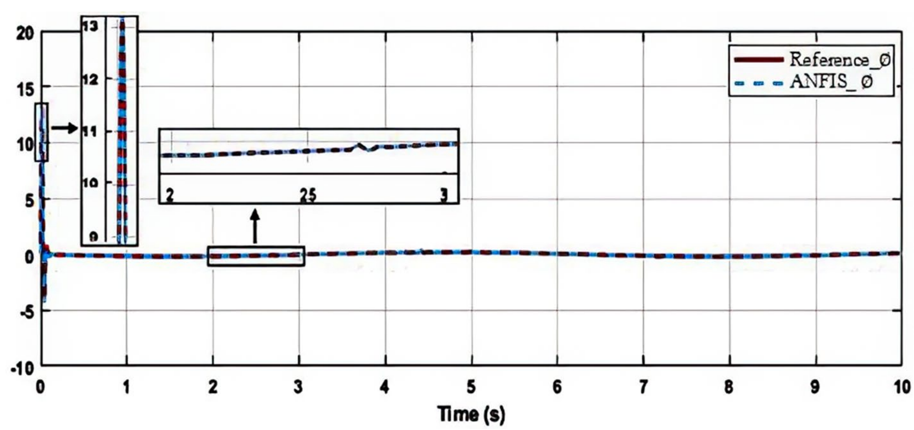


Fig. 15 ANFIS ϕ measured

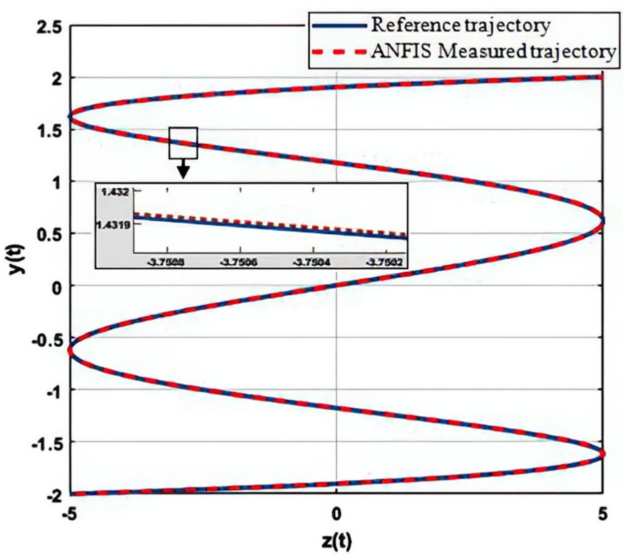


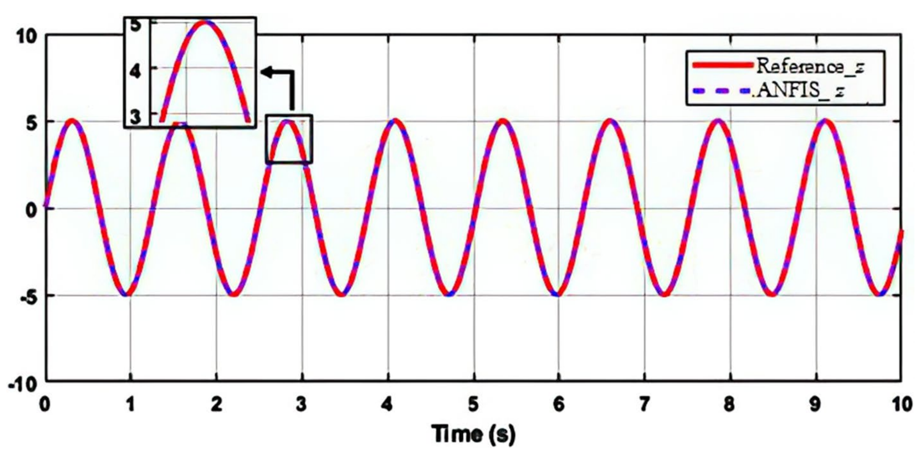
Fig. 16 ANFIS trajectory measured

Figures 17, 18, 19, 20, 21, 22, 23, 24, 25 and 26 show the ANFIS-PIO simulation results. The tracking parameter performances are shown in Figs. 17, 20 and 23. The trajectory tracking performance is illustrated in Fig. 26, where the measured trajectory of the UAV and the reference trajectory are shown together. The results obtained show that the performances of our approach are superior compared to the other controllers. The validity of the proposed controller is proved by MSE < 10^{-30} and RMSE < 10^{-15}. Figures 18, 19, 21, 22, 24 and 25 show the tracking parameters errors. It can be seen that the trajectory tracking error is zero.

The PIO algorithm provides an improvement, by comparing ANFIS performances with the same ANFIS optimized by PIO, a significant increase in accuracy. A clear improvement of the precision in the trajectory tracking is thus visible. The RMSE used to measure the accuracy of the control model decreases by 10^{10} times.

We used an evolutionary algorithm approach that is PIO. A PIO exploration of this search space is performed to identify subsets of more relevant parameters and accurate by a new MF distribution that adapts well to each linguistic variable that leads to minimizing the error between the desired trajectory and the calculated one.

Fig. 17 ANFIS-PIO z measured



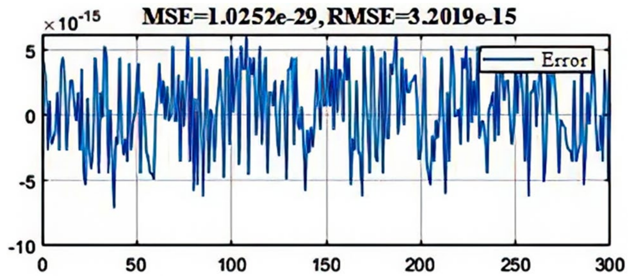


Fig. 18 Error test data (z parameter)

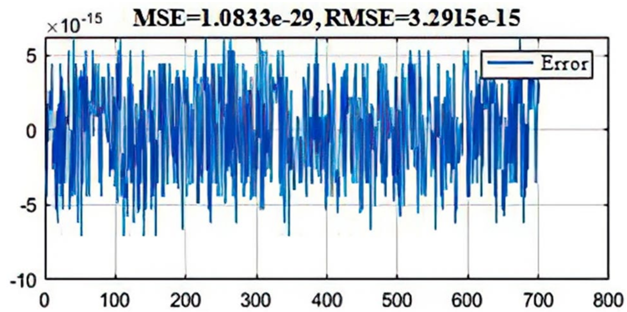


Fig. 19 Error train data (z parameter)

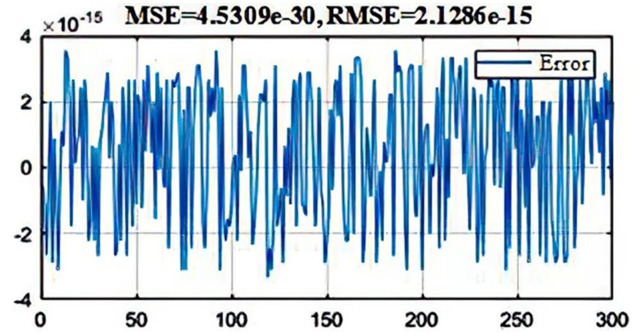


Fig. 21 Error test data (y parameter)

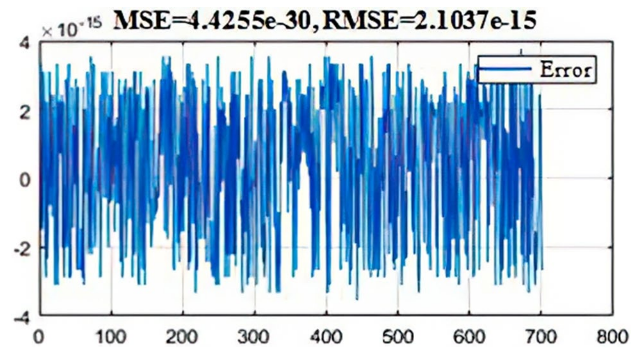
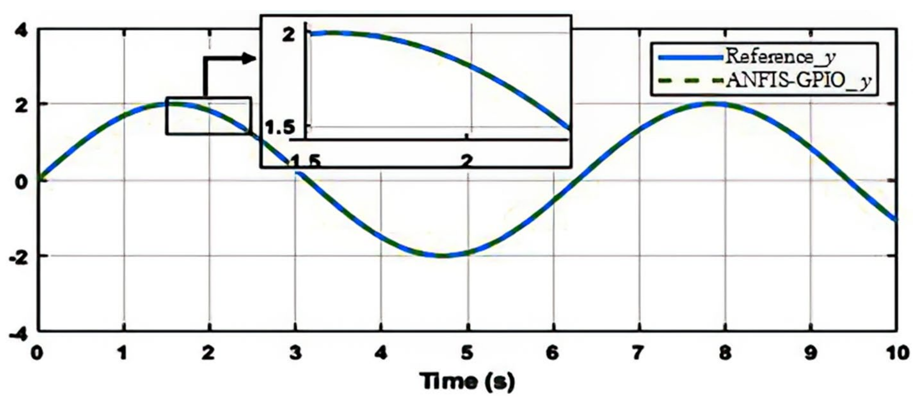


Fig. 22 Error train data (y parameter)

Fig. 20 ANFIS-PIO y measured



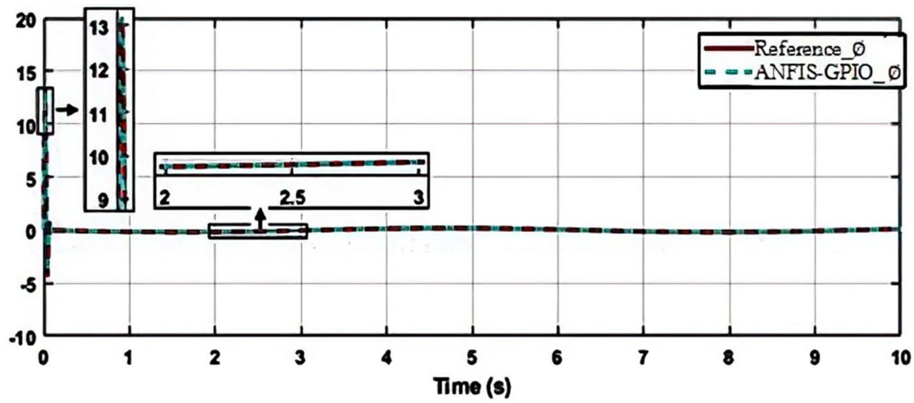


Fig. 23 ANFIS-PIO ϕ measured

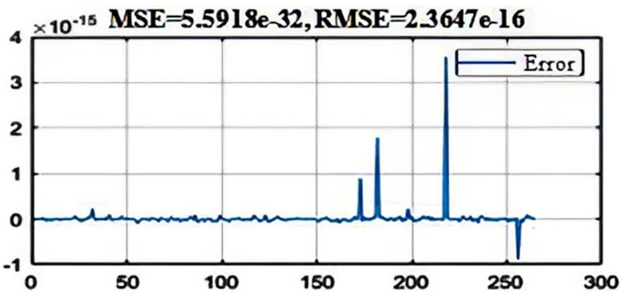


Fig. 24 Error test data (ϕ parameter)

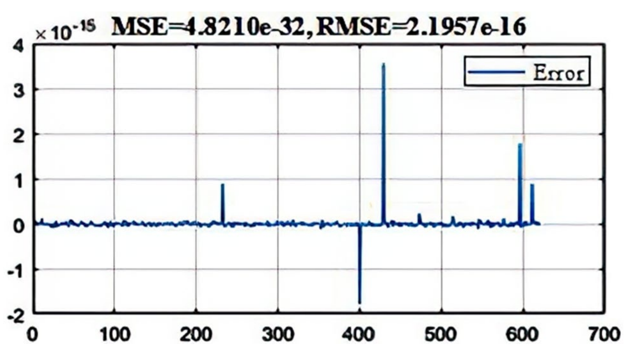


Fig. 25 Error train data (ϕ parameter)

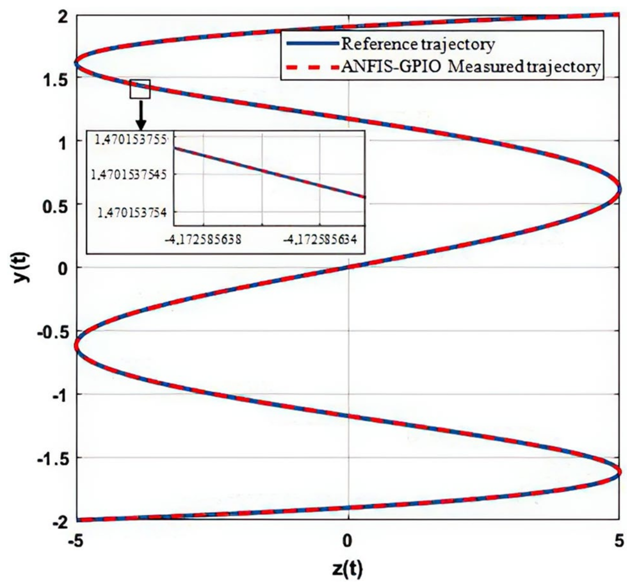


Fig. 26 ANFIS-PIO measured trajectory

9 Conclusion

In this work, we show the application of artificial intelligence to command and control systems, a method to tune neuro-fuzzy controllers using pigeon-inspired optimization (PIO) algorithm. The main advantage of this approach is that it allows adaptive neuro-fuzzy inference system (ANFIS) controllers to be tuned to achieve global performance requirements. In addition, the PIO algorithm was implemented to find the optimal distribution of parameters in the design of the ANFIS controller to improve the performances in ANFIS controllers. This proposed ANFIS-PIO controller is proposed to control the movement of three-degree-of-freedom (3-DOF) unmanned aerial vehicles (UAVs) to track a given reference trajectory. The case study was implemented using simulations. We conclude that dynamically adjusting the ANFIS parameters by PIO optimization method has improve the quality of results and minimize tracking error. The proposed controller has been proven to have better performance in comparison with already developed controllers on the same control problem.

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, S.A., Ahmad, M.S., Mustapha, A.: Adjustable autonomy: a systematic literature review. Artif. Intell. Rev. **51**(2), 149–186 (2019). https://doi.org/10.1007/s10462-017-9560-8
2. Mahadevan, P.: The military utility of drones. CSS Anal. Secur. Policy **78**, 1–3 (2010). https://doi.org/10.3929/ethz-a-006253833
3. Haidari, L.A., Brown, S.T., Ferguson, M., Bancroft, E., Spiker, M., Wilcox, A., Ambikapathi, R., Sampath, V., Connor, D.L., Lee, B.Y.: The economic and operational value of using drones to transport vaccines. Vaccine **34**(34), 4062–4067 (2016). https://doi.org/10.1016/j.vaccine.2016.06.022
4. Barmpounakis, E.N., Vlahogianni, E.I., Golias, J.C.: Unmanned aerial systems for transportation engineering: current practice and future challenges. Int. J. Transp. Sci. Technol. **5**(3), 111–122 (2016)
5. Reinartz, P., Lachaise, M., Schmeer, E., Krauss, T., Runge, H.: Traffic monitoring with serial images from airborne cameras. ISPRS J. Photogramm. Remote Sens. **61**, 149–158 (2006). https://doi.org/10.1016/j.isprsjprs.2006.09.009
6. European Aviation Safety Agency.: ''Prototype'' Commission Regulation on Unmanned Aircraft Operations. (2016). https://www.easa.europa.eu/sites/default/files/dfu/UAS%2520Prototype%2520Regulation%2520final.pdf
7. Siebert, S., Teizer, J.: Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system. Autom. Constr. **41**, 1–14 (2014). https://doi.org/10.1016/j.autcon.2014.01.004
8. Valente, J., Del Cerro, J., Barrientos, A., Sanz, D.: Aerial coverage optimization in precision agriculture management: a musical harmony inspired approach. Comput. Electron. Agric. **99**, 153–159 (2013)
9. Freeman, P.K., Freeland, R.S.: Politics & technology: U.S. polices restricting unmanned aerial systems in agriculture. Food Policy **49**, 302–311 (2014). https://doi.org/10.1016/j.foodpol.2014.09.008
10. Kanistras, K., Martins, G., Rutherford, M.J., Valavanis, K.P.: Survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In: Handbook of unmanned aerial vehicles, pp. 2643–2666. Springer, Dordrecht (2015). https://doi.org/10.1007/978-90-481-9707-1_122
11. Chow, J.Y.J.: Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy. Int. J. Transp. Sci. Technol. **5**(3), 167–185 (2016)
12. Finn, R.L., Wright, D.: Unmanned aircraft systems: surveillance, ethics and privacy in civil applications. Comput. Law Secur. Rev. **28**(2), 184–194 (2012). https://doi.org/10.1016/j.clsr.2012.01.005
13. Lei, M., Manchun, L., Yafei, W., Lihua, T., Liang, C.: Using high-resolution imagery acquired with an autonomous unmanned aerial vehicle for urban construction and planning. In: 2013 the International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE 2013), pp. 200–203 (2013). https://doi.org/10.2991/rsete.2013.49
14. Knyaz, V.A., Chibunichev, A.G.: Photogrammetric techniques for road surface analysis. In: ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Prague, Czech Republic, pp. 515–520 (2016). https://doi.org/10.5194/isprsarchives-XLI-B5-515-2016.
15. Hoffer, N.V., Coopmans, C., Jensen, A.M., Chen, Y.: A survey and categorization of small low-cost unmanned aerial vehicle system identification. J. Intell. Robot. Syst. **74**(1–2), 129–145 (2014)
16. Kanellakis, C., Nikolakopoulos, G.: Survey on computer vision for UAVs: current developments and trends. J. Intell. Robot. Syst. **87**(1), 141–168 (2017)
17. Hayat, S., Yanmaz, E., Muzaffar, R.: Survey on unmanned aerial vehicle networks for civil applications: a communications viewpoint. IEEE Commun. Surv. Tutor. **18**(4), 2624–2661 (2016)
18. Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference systems. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993)
19. Karr, C.L.: Design of an adaptive fuzzy linguistic controller using a genetic algorithm. In: Proc. of the Fourth Int. Conf. Genetic Algorithms, pp. 450–457 (1991)
20. Karr, C.L., Gentry, E.J.: Fuzzy control of pH using genetic algorithms. IEEE Trans. Fuzzy Syst. **1**(1), 46–53 (1993)
21. Herrera, F., Lozano, M., Verdegay, J.L.: Turning fuzzy logic controllers by genetic algorithms. Int. J. Approx. Reason. **12**(3–4), 299–315 (1995)
22. Shimojima, K., Fukuda, T., Hasegawa, Y.: Self-turning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm. Fuzzy Sets Syst. **71**(3), 295–309 (1995)
23. Liska, J., Melsheimer, S.S.: Complete design of fuzzy logic systems using genetic algorithms. In: Proceedings of 3rd IEEE International Conference on Fuzzy Systems, pp. 1377–1382 (1994)
24. Thrift, P.: Fuzzy logic synthesis with genetic algorithms. In: Proceedings 4th International Conference on Genetic Algorithms (ICGA'91), pp. 509–513 (1991)
25. Michael, A., Takagi, H.: Dynamic control of genetic algorithms using fuzzy logic techniques. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 76–83. Morgan Kaufmann (1993)
26. Homaifar, A., McCormick, E.: Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Trans. Fuzzy Syst. **3**(2), 129–139 (1995)
27. Chen, C.C., Wong, C.C.: Self-generating rule-mapping fuzzy controller design using a genetic algorithm. IEEE Proc. Control Theory Appl. **49**, 143–148 (2002)
28. Belarbi, K., Titel, F., Bourebia, W., Benmahammed, K.: Design of mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm. Eng. Appl. Artif. Intell. **18**(7), 875–880 (2005)
29. Refoufi, S., Benmahammed, K.: Control of a manipulator robot by neuro-fuzzy subsets form approach control optimized by the genetic algorithms. ISA Trans. **77**, 133–145 (2018)
30. Talbi, N., Belarbi, K.: Optimization of fuzzy controller using hybrid Tabu search and particle swarm optimization. In: Proc. IEEE, 11th International Conference on Hybrid Intelligent Systems (HIS), Malaysia, pp. 561–565 (2011).

31. Talbi, N., Belarbi, K.: Designing fuzzy controllers for a class of MIMO systems using hybrid particle swarm optimization and Tabu search. Int. J. Hybrid Intell. Syst. **10**(1), 1–9 (2013)
32. Juang, C.F., Chang, P.H.: Designing fuzzy-rule-based systems using continuous Ant colony optimization. IEEE Trans. Fuzzy Syst. **18**(1), 138–149 (2010)
33. Castillo, O., Neyoy, H., Soria, J., Melin, P., Valdez, F.: A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot. Appl. Soft Comput. **28**, 150–159 (2015)
34. Premkumar, K., Manikandan, B.V.: Speed control of brushless DC motor using bat algorithm optimized adaptive neuro-fuzzy inference system. Appl. Soft Comput. **32**, 403–419 (2015)
35. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. Appl. Soft Comput. **43**, 131–142 (2016)
36. Lagunes, M.L., Castillo, O., Soria, J.: Optimization of membership function parameters for fuzzy controllers of an autonomous mobile robot using the firefly algorithm. Fuzzy Log. Augment. Neural Optim. Algorithms **749**, 199–206 (2018)
37. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Networks, Perth, pp. 1942–1948 (1995)
38. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
39. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
40. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, London (2008)
41. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74. Springer, Berlin (2010)
42. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
43. Duan, H., Qiao, P.: Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int. J. Intell. Comput. Cybern. **7**, 24–37 (2014)
44. Chen, S.J., Duan, H.B.: Fast image matching via multi-scale Gaussian mutation pigeon-inspired optimization for low cost quadrotor. Aircr. Eng. Aerosp. Technol. **89**, 777–790 (2017)
45. Lin, N., Huang, S.M., Gong, C.Q.: UAV path planning based on adaptive weighted pigeon-inspired optimization algorithm. Comput. Simul. **35**, 38–42 (2018)
46. Xian, N., Chen, Z.: A quantum-behaved pigeon-inspired optimization approach to explicit nonlinear model predictive controller for quadrotor. Int. J. Intell. Comput. Cybern. **11**, 47–63 (2018)
47. Pei, J.Z., Su, Y.X., Zhang, D.H.: Fuzzy energy management strategy for parallel HEV based on pigeon-inspired optimization algorithm. Sci. China Technol. Sci. **60**, 425–433 (2017)
48. Zhang, D.F., Duan, H.B., Yang, Y.J.: Active disturbance rejection control for small unmanned helicopters via Lévy flight-based pigeon-inspired optimization. Aircr. Eng. Aerosp. Technol. **89**, 946–952 (2017)
49. Li, C., Duan, H.B.: Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function. Aerosp. Sci. Technol. **39**, 352–360 (2014)
50. Zhang, B., Duan, H.B.: Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment. IEEE/ACM Trans. Comput. Biol. Bioinf. **14**, 97–107 (2017)
51. Duan, H., Wang, X.: Echo state networks with orthogonal pigeon-inspired optimization for image restoration. IEEE Trans. Neural Netw. Learn. Syst. **27**(11), 2413–2425 (2016)
52. Jiang, P.P., Zhou, K., Zhu, Q.K., et al.: Route planning of armed helicopter based on pigeon-inspired optimization with threat heuristic. Electron Opt. Control **24**, 56–61 (2017)
53. Sushnigdha, G., Joshi, A.: Re-entry trajectory optimization using pigeon inspired optimization based control profiles. Adv. Space Res. **62**, 3170–3186 (2018)
54. Xu, X., Deng, Y.: UAV power component-DC brushless motor design with merging adjacent-disturbances and integrated-dispatching pigeon-inspired optimization. IEEE Trans. Magn. **54**, 1–7 (2018)
55. Sun, Y., Duan, H., Xian, N.: Fractional-order controllers optimized via heterogeneous comprehensive learning pigeoninspired optimization for autonomous aerial refueling hose-drogue system. Aerosp. Sci. Technol. **81**, 1–13 (2018)
56. Li, S., Deng, Y.: Quantum-entanglement pigeon-inspired optimization for unmanned aerial vehicle path planning. Aircr. Eng. Aerosp. Technol. **91**, 171–181 (2018)
57. Zhang, D., Duan, H.: Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning. Neurocomputing **313**, 229–246 (2018)
58. Qiu, H., Duan, H.: A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. Inform. Sci. **509**, 515–529 (2020). https://doi.org/10.1016/j.ins.2018.06.061
59. Dos Santos Coelho, L., Alotto, P.: Gaussian artificial bee colony algorithm approach applied to Loney's solenoid benchmark problem. IEEE Trans Magn. **47**(5), 1326–1329 (2011)
60. Zhang, S., Duan, H.: Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration. Chin. J. Aeronaut. **28**(1), 200–205 (2015)