

Optimization of fuzzy membership functions via PSO and GA with application to quad rotor

B. Safaee and S. Kamaleddin Mousavi Mashhadi*

School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.

Received 06 March 2016; Accepted 22 September 2016

*Corresponding author: sk_mousavi@iust.ac.ir (S. K Mousavi).

Abstract

Quad rotor is a renowned under-actuated unmanned aerial vehicle (UAV) with widespread military and civilian applications. Despite its simple structure, the vehicle suffers from inherent instability. Therefore, control designers always face a formidable challenge in their stabilization and control goal. In this paper, the fuzzy membership functions of the quad rotor fuzzy controllers are optimized using nature-inspired algorithms such as particle swarm optimization (PSO) and genetic algorithm (GA). Finally, the results of the proposed methods are compared, and a trajectory is defined to verify the effectiveness of the designed fuzzy controllers based on the algorithm with better results.

Keywords: *Fuzzy Controller, GA, Membership Functions, PSO, Quad Rotor.*

1. Introduction

Quad rotor is a kind of unmanned aerial vehicle (UAV) with 4 rotors in a cross-configuration that has been widely employed in civil and military applications including traffic surveillance, inspections, monitoring, aerial photography, search, and rescue in hazardous environments that are inaccessible for human involvement. Compared to other UAVs, they have a simple structure, low cost, easy maintenance, vertical take-off and landing capability, and rapid maneuvering. Quad rotors are under-actuated systems. They possess some advantages over single-blade helicopters in that they can be controlled by changing the rotor speed and have a simpler design and an easier control due to their fixed pitch blades. Quad rotors are inherently unstable based on several reasons elaborated as follow. First, their dynamic model features include non-linearity, high coupling, being under-actuated and static instability. Secondly, quad rotors are affected by body gravity, air resistance, propeller driving force, gyro impacts, and so on. Therefore, basically there is uncertainty in the dynamic model, which accentuates the need for a proper controller. Several controlling methods have been proposed including intelligent PID algorithm [1], LQR [2], loop forming H_∞ method,

sliding mode variable structure control algorithm [3], adaptive controller [4], feed-back linearization algorithm, back-stepping algorithm [5], hybrid control algorithm and so on [6]. In the following literature, some recent methods have concisely been presented. In [7], a passivity-based adaptive back stepping controller has been proposed to control a type of quad rotor. Reference [8] adopts a robust control for automatic tracking, taking-off, and landing. In [9], a feedback linearization method has been introduced based on the piecewise bilinear model. In [10], the performance of the quad rotor in the presence of wind disturbances has been assessed when back-stepping and feedback linearization controllers are applied on the quad rotor. Literature [11] has introduced the design of a neural adaptive controller using an extreme learning machine (ELM) to control the Euler angles of a quad rotor. A PD fuzzy control has been proposed in [12] to control the attitude. In [13], the controlling goal is to design an intelligent fuzzy controller in an effort to optimize the controller input. An adaptive fuzzy control strategy has been proposed in [14] to solve the problem of trajectory tracking for quad rotors in the presence of model parameter uncertainties and

external disturbances. In order to reach the desired attitude and height, a linear active disturbance rejection control (LADRC) approach is utilized for the quad rotor control. LADRC has the capacity to estimate and compensate for the generalized disturbance, and reduce the system to a unit gain double integrator, which can be easily implemented and is robust to environmental disturbances [15]. In [16], a new non-linear controller has been proposed using a back-stepping-like feedback linearization method to control and stabilize the quad rotor. A novel asymptotic tracking controller has been proposed in [17] for a quad rotor using the robust integral of the signum of the error (RISE) method and the immersion and invariance (I & I)-based adaptive control methodology. In [18], the designers have endeavored to gain the stability and tracking control problem of a quad rotor in the presence of modeling error and disturbance uncertainties associated with aerodynamic and gyroscopic effects, payload mass, and other external forces/torques induced from an uncertain flying environment. A PD sliding surface in the sliding mode control has been considered in [19] for a vertical take-off and landing aircraft. A control scheme based on back-stepping control and its combination with fuzzy system has been introduced in [20] to decrease the chattering phenomenon. A control algorithm for a finite-time tracking of a quad rotor has been introduced in [21]; it is based upon the use of feed-back linearization method and finite-time output control. The proposed multi-Lyapunov function-based switching control algorithm in [22] is employed to achieve tracking of Cartesian space motion and the heading angle of a quad rotor.

In this paper, a fuzzy controller due to its capability to handle non-linear systems and model uncertainties is proposed to control both the transitional and angular movements of a quad rotor. Choosing a proper fuzzy membership functions is a time-consuming task, especially when there are a number of fuzzy controllers in the system simultaneously. The main contribution of this paper is to overcome this problem by means of parametrically defining fuzzy membership functions and tuning them based on the minimization of a specific objection function. GA and PSO are selected as the optimizing methods owing to their capability in locating global minima. This paper is organized as follows. Section 1 deals with a brief discussion of the quad rotor configuration and dynamic model. Next, the quad rotor fuzzy controller design is discussed in

section 2. Section 3 concerns a concise introduction of PSO and GA. Finally, the simulation results derived from the proposed algorithms are compared and a trajectory is defined to illustrate the superiority of the best designed fuzzy controller over the not optimized fuzzy controller in section 5.

1.1. Quad rotor configuration

According to figure 1, the front and rear rotors rotate clockwise, while the right and left rotors rotate counter-clockwise. Increasing (decreasing) the speed of all rotors with the same amount generates the vertical motion or thrust. A roll motion can be obtained by increasing (decreasing) the left rotor speed, while decreasing (increasing) the speed of the right rotor. Similarly, a pitch motion is controlled by an increase (a decrease) in the speed of the rear rotor while decreasing (increasing) the front rotor speed. The yaw angle is achieved by increasing the clockwise pair speeds and decreasing the counter-clockwise pair speeds, simultaneously.

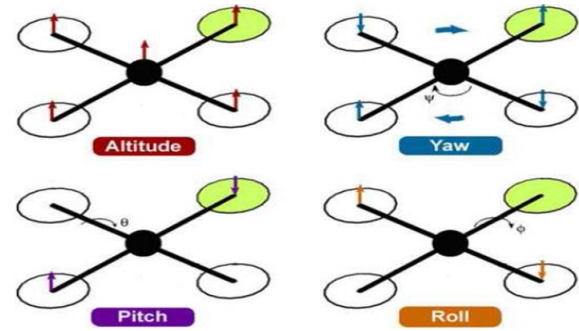


Figure 1. Quad rotor dynamics [23].

2. Model dynamics of quad rotor vehicle

2.1. Reference frames

In order to obtain the quad rotor model, we need to introduce three frames, as follow:

The inertial frame, $F_i = (\vec{x}_i, \vec{y}_i, \vec{z}_i)$, is an earth-fixed coordinate system with the origin located on the ground, for example, at the base station. By convention, the x-axis points towards the north, the y-axis points towards the east, and the z-axis points towards the center of the earth.

The body frame, $F_b = (\vec{x}_b, \vec{y}_b, \vec{z}_b)$, with its origin located at the center of gravity (COG) of the quad rotor, and its axes aligned with the quad rotor structure such that the x-axis \vec{x}_b is along the arm with front motor, the y-axis \vec{y}_b is along the arm with right motor, and the z-axis $\vec{z}_b = \vec{x}_b \times \vec{y}_b$, where '×' denotes the cross-product.

The vehicle frame, $F_v = (\vec{x}_v, \vec{y}_v, \vec{z}_v)$, is the inertial frame with the origin located at the COG of the quad rotor. The vehicle frame has two variations, F_ϕ and F_θ . F_ϕ is the vehicle frame, F_v , rotated about its z-axis \vec{z}_v by an angle ψ so that \vec{x}_v and \vec{y}_v are aligned with \vec{x}_b and \vec{y}_b , respectively. F_θ is the frame F_ϕ rotated about its y-axis, \vec{y}_ϕ , by a pitching angle, θ , such that \vec{x}_ϕ and \vec{z}_ϕ are aligned with \vec{x}_b and \vec{z}_b , respectively. Figure 2 shows these main frames [23].

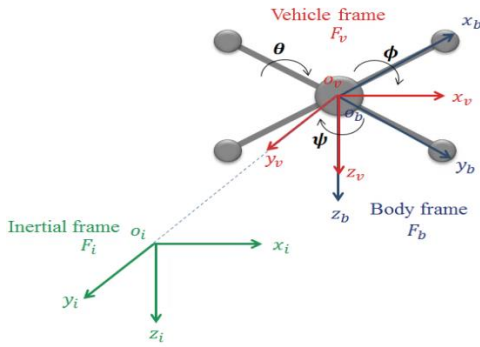


Figure 2. Inertial, body and vehicle frames [23].

2.2. Quad rotor kinematics

If we define $P_F^T = [p_x, p_y, -p_z]$ as quad rotor position and $\Omega_F^T = [\phi, \theta, \psi]$ as the orientation within a given frame, then the correlation between the quad rotor speeds in the three pre-defined frames can be stated as [23]:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ -\dot{p}_z \end{bmatrix}_{F_i} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix}_{F_v} = [R_{F_v}^T]^T \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix}_{F_b} \quad (1)$$

where, $[R_{F_v}^T]^T$ is the rotational matrix mapping the F_b frame to the F_v frame and is defined as:

$$[R_{F_v}^T]^T = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi - C\phi S\psi & C\phi S\theta C\psi + S\phi S\psi \\ C\theta S\psi & S\phi S\theta S\psi + C\phi C\psi & C\phi S\theta S\psi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \quad (2)$$

where, 'S' stands for 'Sin' and 'C' stands for 'Cos', for simplicity. The relation between the angular speeds in the body and the vehicle frames can be obtained using the rotational matrixes, as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{F_b} = R_{F_\phi}^{F_b}(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{F_\phi}^{F_b}(\phi) R_{F_\theta}^{F_\phi}(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{F_\phi}^{F_b}(\phi) R_{F_\theta}^{F_\phi}(\theta) R_{F_v}^{F_\theta}(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3)$$

where:

$$R_{F_\theta}^{F_\phi}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (4)$$

$$R_{F_v}^{F_\phi}(\psi) = \begin{bmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{bmatrix} \quad (5)$$

$$R_{F_\phi}^{F_b}(\phi) = R_{F_\theta}^{F_b}(\theta) = R_{F_v}^{F_b}(\psi) = I \quad (6)$$

Therefore,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{F_b} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{F_v} \quad (7)$$

It follows that:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{F_v} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{F_b} \quad (8)$$

The quad rotor equations of motion can be described by (1) and (7).

2.3. Quad rotor dynamics

Before getting into the quad rotor dynamic model based on the Newton- Euler formalism, we need to make some assumptions as:

- Quad rotor is a solid body.
- It has a symmetrical frame.
- Its center of gravity is the center of the rigid body.

The moment of inertia is calculated by supposing the quad rotor as a central sphere of radius r and mass M_o encircled by four point masses representing the motors. Each motor is assumed to have a mass m and joined to the central sphere

through an arm of length l . As mentioned earlier, a quad rotor is symmetric about its three axes, and, consequently the inertial matrix is symmetric:

$$J = \begin{bmatrix} j_x & 0 & 0 \\ 0 & j_y & 0 \\ 0 & 0 & j_z \end{bmatrix} \quad (9)$$

Dynamic of the quad rotor in the presence of external forces applied on its center of gravity in the body frame can be derived based on the Newton- Euler formalism, as follows:

$$\begin{bmatrix} M I_{3 \times 3} & 0 \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{P}}_{F_b} \\ \ddot{\mathbf{Q}}_{F_b} \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{Q}}_{F_b} \times M \dot{\mathbf{P}}_{F_b} \\ \dot{\mathbf{Q}}_{F_b} \times J \dot{\mathbf{Q}}_{F_b} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{F_b} \\ \boldsymbol{\tau}_{F_b} \end{bmatrix} \quad (10)$$

where, M refers to the total mass of the quad rotor and $\mathbf{F}^T = [f_x, f_y, f_z]$ and $\boldsymbol{\tau}^T = [\tau_\phi, \tau_\theta, \tau_\psi]$ are the applied external force and the torque vectors on the quad rotor center of gravity. Therefore, the transitional and rotational dynamic model would be [23]:

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{bmatrix}_{F_b} = \begin{bmatrix} \dot{\psi} \dot{p}_y - \dot{\theta} \dot{p}_z \\ \dot{\phi} \dot{p}_z - \dot{\psi} \dot{p}_x \\ \dot{\theta} \dot{p}_x - \dot{\phi} \dot{p}_y \end{bmatrix}_{F_b} + \frac{1}{M} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_{F_b} \quad (11)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = J^{-1} \left\{ \begin{bmatrix} 0 & \dot{\psi} & -\dot{\theta} \\ -\dot{\psi} & 0 & \dot{\phi} \\ \dot{\theta} & -\dot{\phi} & 0 \end{bmatrix} J \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \right\} = \begin{bmatrix} \frac{j_y - j_z}{j_x} \dot{\theta} \dot{\psi} \\ \frac{j_z - j_x}{j_y} \dot{\phi} \dot{\psi} \\ \frac{j_x - j_y}{j_z} \dot{\phi} \dot{\theta} \end{bmatrix}_{F_b} + \begin{bmatrix} \frac{1}{j_x} \tau_\phi \\ \frac{1}{j_y} \tau_\theta \\ \frac{1}{j_z} \tau_\psi \end{bmatrix}_{F_b} \quad (12)$$

2.4. Torques and aerodynamic forces

With the kinematic and dynamic model mentioned above, we will be able to define the forces and torques applied on the quad rotor. The forces include the aerodynamic lift generated by each rotor and the gravitational force acting in the opposite direction of the generated total lift. The moments are comprised of the torques generated to obtain the roll, pitch, and yaw movements [23].

Total thrust: the total thrust of the quad rotor is the sum of each propeller thrust:

$$T = T_f + T_r + T_b + T_l \quad (13)$$

Roll torque: this torque can be generated when the thrust of the left rotor increases, while the right thrust decreases:

$$T_\phi = l(T_l - T_r) \quad (14)$$

Pitch torque: this torque is produced by increasing the front rotor thrust and decreasing the rear rotor thrust at the same time:

$$T_\theta = l(T_f - T_b) \quad (15)$$

Yaw torque: this torque, which causes the quad rotor to revolve around its z axis, stems from the clockwise rotation of the front and rear rotors in conjunction with the counter-clockwise of the right and left rotors. In other words, it is the sum of the torques generated by four rotors:

$$\tau_\psi = \tau_f + \tau_b + \tau_r + \tau_l \quad (16)$$

Gravity force or weight: along with other forces, this force is applied on the quad rotor center of gravity. In the vehicle frame, it is represented as:

$$\mathbf{W}_{F_v} = \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix} \quad (17)$$

where, g is the gravitational constant. If we map this force from the vehicle frame to the body frame, it can be re-written as follows:

$$\mathbf{W}_{F_b} = \mathbf{R}_{F_v}^{F_b} \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix} = \begin{bmatrix} -Mg \sin \theta \\ Mg \cos \theta \sin \phi \\ Mg \cos \theta \cos \phi \end{bmatrix} \quad (18)$$

Finally, dynamic equations of the quad rotor considering these forces and torques can be summarized as:

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{bmatrix}_{F_b} = \begin{bmatrix} \dot{\psi} \dot{p}_y - \dot{\theta} \dot{p}_z \\ \dot{\phi} \dot{p}_z - \dot{\psi} \dot{p}_x \\ \dot{\theta} \dot{p}_x - \dot{\phi} \dot{p}_y \end{bmatrix} + \begin{bmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{-f_z}{M} \end{bmatrix} \quad (19)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}_{F_b} = \begin{bmatrix} \frac{j_y - j_z}{j_x} \dot{\theta} \dot{\psi} \\ \frac{j_z - j_x}{j_y} \dot{\phi} \dot{\psi} \\ \frac{j_x - j_y}{j_z} \dot{\phi} \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{j_x} \tau_\phi \\ \frac{1}{j_y} \tau_\theta \\ \frac{1}{j_z} \tau_\psi \end{bmatrix} \quad (20)$$

3. Quad rotor controller design

The proposed controller can control the quad rotor position and attitude, simultaneously. Our aim is to reach the control states (p_x, p_y, p_z) and Ψ to their desired values $(p_{xd} = 10m, p_{yd} = 10m, p_{zd} = 25m, \Psi_d = \frac{\pi}{6} \text{ rad})$, and keep the roll and pitch angles close to zero. The control strategy can be seen in figure 3.

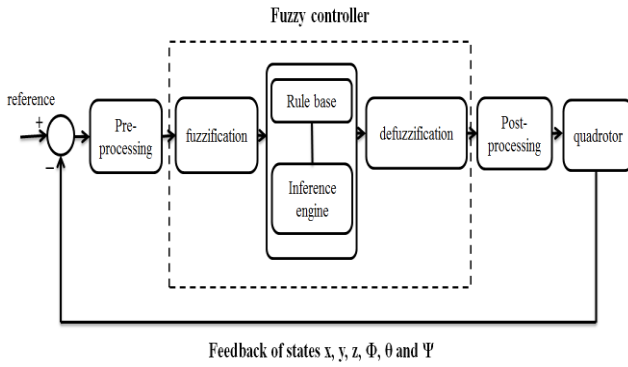


Figure 3. Control strategy [23].

The supposed PWM_{mot} is denoted as the PWM of the front motor (f), back motor (b), left motor (l), and right motor (r). Then the relation between the torques and thrust and PWM_{mot} can be expressed as:

$$T_{mot} = K_T \times PWM_{mot} \quad (21)$$

$$\tau_{mot} = K_\tau \times PWM_{mot} \quad (22)$$

where K_T and K_τ are the dependent parameters of the motors. Hence, we obtain:

$$\begin{bmatrix} PWM_f \\ PWM_r \\ PWM_b \\ PWM_l \end{bmatrix} = G \times \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (23)$$

with:

$$G = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -l \times K_T & 0 & l \times K_T \\ l \times K_T & 0 & -l \times K_T & 0 \\ -K_\tau & K_\tau & -K_\tau & K_\tau \end{bmatrix}^{-1} \quad (24)$$

In order to aim for the controlling goal, six fuzzy controllers are required. The inference engine implemented in the fuzzy controllers is the Mamdani fuzzy model using the min-max operator for the aggregation and the centroid of area method for defuzzification. All of these controllers have an error, which is the difference between the actual value of a state and its desired value, and the error rate as their inputs. The former is normalized to the interval $[-1,1]$, and the latter is normalized to the interval $[-3,3]$ based on the actuators range. The block diagram of the flight control is depicted in figure 4.

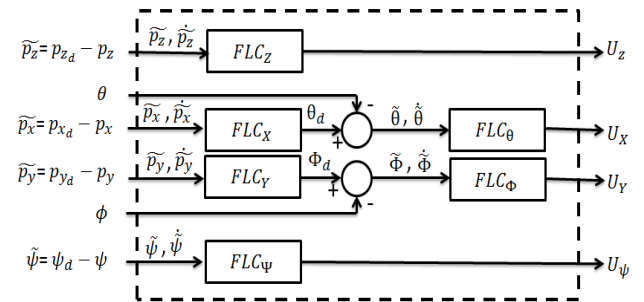


Figure 4. Flight control block diagram.

Each fuzzy controller has three triangular fuzzy membership functions for its each input and one output. Note that the fuzzy membership functions are chosen triangular due to their less sensitivity to the noise and easy implementation [24]. Therefore, nine fuzzy rules can be defined as shown in figure 5. For example, as shown in this figure, the following rule can be defined:

If error (e) is Negative (N) and error rate (ė) is Positive (P), then U is Zero (Z).

$\begin{matrix} e \\ \dot{e} \end{matrix}$	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

Figure 5. Table of fuzzy rules.

The desired angles ϕ_d and θ_d are not explicitly provided but are continuously predicted by FLC_y and FLC_x . As depicted earlier in figure 3, the pre-processing block is responsible for the calculation and normalization of the error and error rate, and

the post-processing block employs the output signals of the fuzzy controllers to achieve the PWM value of each motor [23]:

$$\begin{aligned} PWM_f &= Sat(U_z + U_x - U_y + Offset) \\ PWM_r &= Sat(U_z + U_y + U_x + Offset) \\ PWM_b &= Sat(U_z - U_x - U_y + Offset) \\ PWM_l &= Sat(U_z - U_y + U_x + Offset) \end{aligned} \quad (25)$$

where, ‘offset’ is a prior-biased that counteracts the quad rotor weight. The PWM values are confined to a maximum threshold that is highly dependent on the maximum speed of the motors.

4. Membership function optimization

4.1. Genetic algorithm (GA)

GA, as a population-based algorithm, is probably the most popular algorithm due to its widespread applications in complex problems and parallelism. It also has applications in fuzzy systems [25-27]. This algorithm is inspired by the mechanism of natural selection. In a typical GA, every potential solution of a problem is regarded as a chromosome. The chromosome degree of the “goodness” is determined by its fitness value. Each GA initiates with a population of chromosomes. The chromosomes with higher fitness value are more likely to be selected and go through the next parts of the GA cycle, which are crossover and mutation that help the algorithm to culminate in a new generation. A typical GA works as follows [28]:

1. Initiate with a randomly produced population of n chromosomes.
2. Calculate each chromosome fitness.
3. Repeat the following steps until n off-springs have been generated:
 - a. Select a pair of chromosomes from the population with less fitness values (for the minimization purpose).
 - b. Perform cross-over to the pair to form two new off-springs.
 - c. Mutate the two off-springs, and put the resulting chromosome in the new population.
4. Replace the current population with the new one.
5. Go to step 2.

4.2. Particle swarm optimization (PSO)

PSO is inspired by the social behavior of birds. A typical PSO starts with a swarm of particles in a multi-dimensional search space, where each particle represents a potential solution of the problem. Each particle has a position and speed,

which are modified according to (29) and (28), respectively. A particle position is updated according to its best personal position (Pbest) and the best global position of all particles (Gbest), according to (26) and (27):

$$x_i^p(k+1) = \begin{cases} x_i^p(k) & \text{if } f(x_i(k+1)) > f(x_i^p(k)) \\ x_i(k+1) & \text{if } f(x_i(k+1)) \leq f(x_i^p(k)) \end{cases} \quad (26)$$

$$x_g = \min\{x_i^p(k)\} \quad (27)$$

$$\begin{aligned} v_i(k+1) &= \eta \times v_i(k) + c_1 r_1(k) \times (x_g - x_i(k)) \\ &+ c_2 \times r_2(k) \times (x_i^p - x_i(k)) \end{aligned} \quad (28)$$

where, η is the inertia factor or each particle momentum, $x_i(k)$ is the current position of particle i , $r_1(k)$ and $r_2(k) \sim U(0,1)$ are random values in the range $[0,1]$, c_1 , c_2 are positive weighting constants representing the importance of Gbest and Pbest, respectively, and v_i is restricted to its maximum and minimum value.

The position of particle i is updated as [29]:

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (29)$$

The search mechanism of PSO according to what was stated above can be seen in figure 6. PSO application in a fuzzy system can be seen in the literature [30-33].

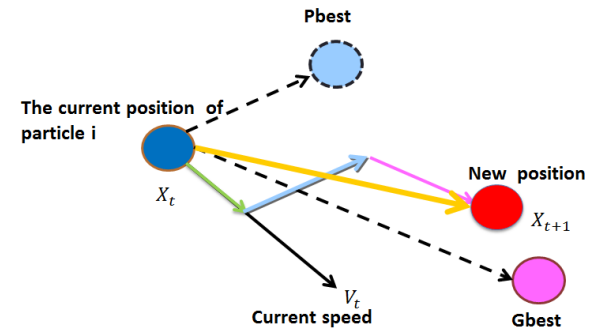


Figure 6. Search mechanism of PSO.

4.3. Problem statement

For the optimization purpose of the fuzzy membership functions, we need to define the parametric fuzzy membership functions. In order to reduce the computational efforts, all of the six fuzzy controllers are assumed identical with the same parametric fuzzy membership functions. Hence, by introducing two parameters like a and b for each membership function of a fuzzy

controller, as illustrated in figure 7, the total number of parameters to be optimized will be six.

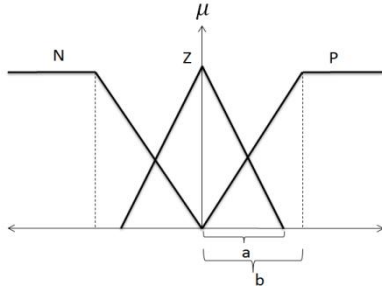


Figure 7. Parameters defined for fuzzy membership functions.

The objective function that should be minimized in order to obtain the fuzzy membership function parameters is chosen as Integral Time Square Error (ITSE):

$$cost = \int_0^t t(ee^T) dt \quad (30)$$

where, e is the vector including the errors of the states and t denotes the simulation time. This objective function penalizes larger errors more heavily than smaller errors, and can produce the fast settling time.

5. Simulation results

In this section, we present the simulation results derived using the GA and PSO algorithms applied on the fuzzy controllers in the MATLAB software.

The GA and PSO specifications used in the simulation can be seen in tables 1 and 2.

Table 1. Specifications of GA.

Mutation	Cross-over	Selection	Stopping criteria	Population size
Constraint dependent	Scattered	Stochastic uniform	After 200 generations	100

Table 2. Specifications of PSO.

Iterations	c_1, c_2	η	Population size
40	2	1	2^6

The optimized fuzzy membership functions of the fuzzy controllers using GA and PSO can be seen in figure 8.

The blue graph indicates the fuzzy membership functions of the GA algorithm and the red graph shows the fuzzy membership functions of the PSO algorithm.

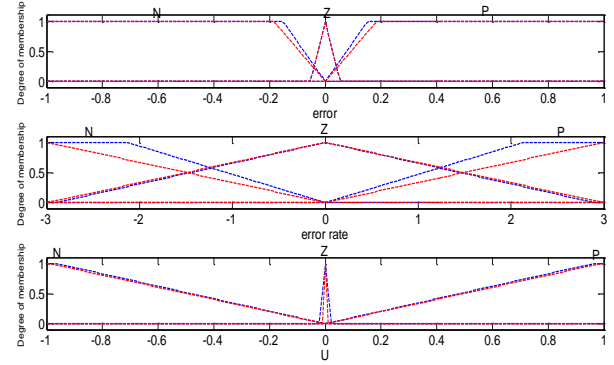


Figure 8. Optimized fuzzy membership functions using GA (blue graph) and PSO (red graph).

Figures 9 and 10 represent the positions and angles of the quad rotor derived from optimized fuzzy controllers with GA and PSO, respectively.

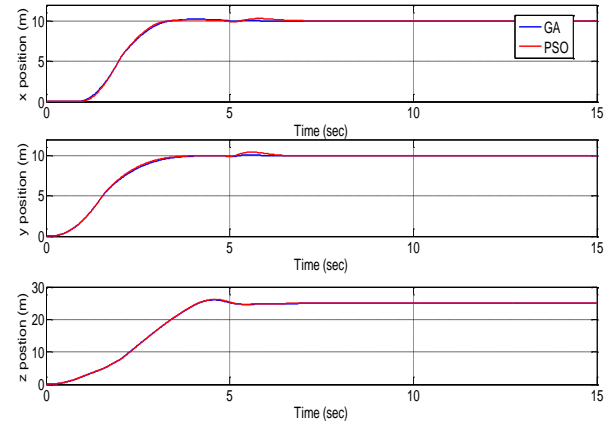


Figure 9. Positions of quad rotor.

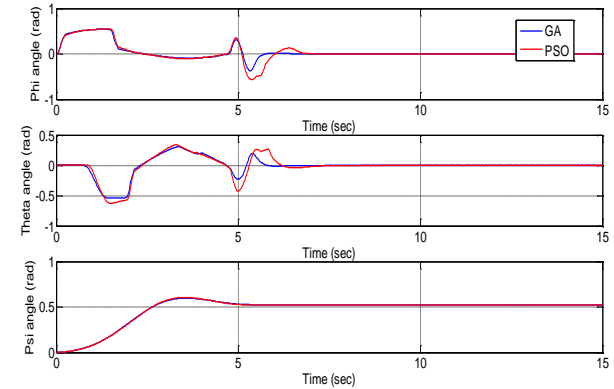


Figure 10. Angles of quad rotor.

We can observe in figures 9 and 10 that all states converged into their desired value ($x_d = 10m$,

$$y_d = 10m, z_d = 25m, \psi_d = \frac{\pi}{6} \text{ rad}, \phi_d \approx 0 \text{ rad},$$

$\theta_d \approx 0 \text{ rad}$) after a short time. According to figure 9 percent of overshoot is negligible (almost zero). Generally, the position and orientation controls

with both GA and PSO depict almost close and satisfactory results.

Table 3 shows the objective function values and total simulation times of the proposed algorithms.

Table 3. Final results of GA and PSO.

Algorithm	Fitness value	Simulation time (s)
GA	8.1102×10^4	3.0045×10^4
PSO	8.054×10^4	2.597×10^4

According to table 3, PSO has both a less fitness value and a less simulation time compared to GA.

Figure 11 illustrates the evolution of fitness vs. total individuals in GA.

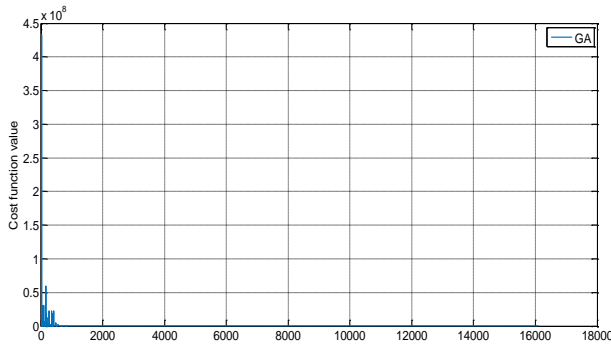


Figure 11. Evolution of fitness value vs. total individuals in GA.

Figure 12 illustrates the evolution of fitness value vs. iterations in PSO algorithm.

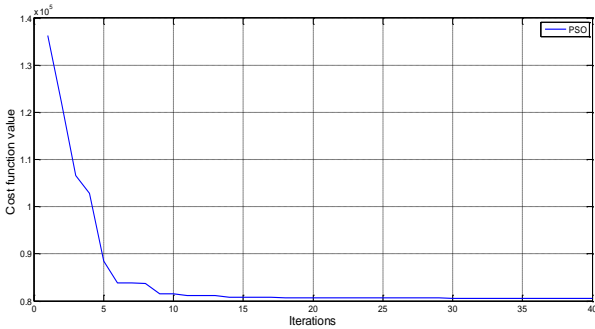


Figure 12. Evolution of fitness value vs. iterations in PSO.

Observing figures 11 and 12 shows that the cost function decreases until it converges a specific value, and this means that those algorithms results will no longer experience improvement.

In order to validate the performance of the fuzzy control approach with optimized membership functions, we define a trajectory comprises x_d and y_d as ramp functions with slope 1 and z_d as a step function. Figure 13 presents the quad rotor trajectory in a 3D space. The optimized fuzzy membership functions are tuned based on PSO.

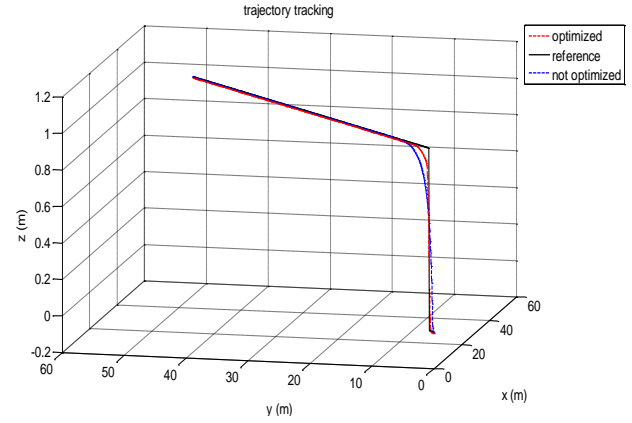


Figure 13. Flight trajectory.

As shown in figure 13, the quad rotor tracking capability has improved with the optimized fuzzy control approach in terms of a faster settling and a rise time.

6. Conclusion

In this paper, the problem of quad rotor flight control was addressed. A fuzzy logic control approach, which does not require the exact mathematical model and has the ability to compensate for the uncertainties and handle non-linearities, was proposed to control the position and orientation of the quad rotor. It comprises six individual fuzzy controllers with Mamdani inference engine. Owing to the high dependence of the fuzzy controllers' performance on their fuzzy membership functions and eliminating the need for the time-consuming trial and error approach, we defined specific parameters for each membership function, and tuned them through minimization of the proposed objective function employing GA and PSO. The investigation of these optimization algorithms applied on the fuzzy controllers was conducted in MATLAB. We compared two proposed methods and observed from the results obtained that both designed fuzzy controllers possess almost close satisfactory performance in the translational and angular quad rotor movement control. To be more specific, considerably small and negligible percent of overshoot and short rise and settling time can be seen in the quad rotor position responses. However when we compared the proposed methods under careful scrutiny, we observed that PSO had a lower fitness value and simulation time. Hence, it can be concluded that the PSO performance outclasses the GA performance in terms of fitness value and simulation time in this case study. Finally, a trajectory was defined to validate the tracking performance of the designed fuzzy controllers

whose fuzzy membership functions were optimized by PSO. The simulation result showed that the designed controllers yielded satisfactory performance in term of faster settling and rise time compared to the not optimized fuzzy controllers.

References

- [1] Su, J., Fan, P., & Cai, K. (2011). Attitude control of quadrotor aircraft via nonlinear PID. *Journal of Beijing University of Aeronautics and Astronautics*, vol. 37, no. 9, pp. 1054-1058.
- [2] Zhang, Z., & Cong, M. (2011). Controlling quadrotors based on linear quadratic regulator. *Applied Science and Technology*, vol. 5, pp. 38-42.
- [3] Besnard, L., Shtessel, Y. B., & Landrum, B. (2012). Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 658-684.
- [4] C Diao, C., Xian, B., Yin, Q., Zeng, W., Li, H., & Yang, Y. (2011). A nonlinear adaptive control approach for quadrotor UAVs. In *Control Conference (ASCC)*, 8th Asian Control, Taiwan, pp. 223-228.
- [5] Bouabdallah, S., & Siegwart, R. (2005). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 2247-2252.
- [6] Li, Y., & Song, S. (2012). A survey of control algorithms for quadrotor unmanned helicopter. In *Advanced Computational Intelligence (ICACI)*, 2012 IEEE Fifth International Conference on , pp. 365-369.
- [7] Ha, C., Zuo, Z., Choi, F. B., & Lee, D. (2014). Passivity-based adaptive backstepping control of quadrotor-type UAVs. *Robotics and Autonomous Systems*, vol. 62, no. 9, pp. 1305-1315.
- [8] Liu, H., Bai, Y., Lu, G., Shi, Z., & Zhong, Y. (2014). Robust tracking control of a quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, vol. 75, no. 3-4, pp. 595-608.
- [9] Taniguchi, T., Eciolaza, L., & Sugeno, M. (2014). Quadrotor control using dynamic feedback linearization based on piecewise bilinear models. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pp. 1-7.
- [10] Araar, O., & Aouf, N. (2014). Quadrotor control for trajectory tracking in presence of wind disturbances. In *Control (CONTROL)*, 2014 UKACC International Conference on , pp. 25-30. IEEE, 2014.
- [11] Zhang, Y., Xu, B., & Li, H. (2015). Adaptive neural control of a quadrotor helicopter with extreme learning machine. In *Proceedings of ELM-2014 Volume 2*. Springer International Publishing, pp. 125-134.
- [12] Petruševski, I., & Rakić, A. (2014). Simple fuzzy solution for quadrotor attitude control. In *Neural Network Applications in Electrical Engineering (NEUREL)*, 2014 12th Symposium on. IEEE, pp. 93-98.
- [13] Fakurian, F., Menhaj, M. B., & Mohammadi, A. (2014). Design of a fuzzy controller by minimum controlling inputs for a quadrotor. In *Robotics and Mechatronics (ICRoM)*, 2014 Second RSI/ISM International Conference on . IEEE, pp. 619-624.
- [14] Yacef, F., Bouhali, O., & Hamerlain, M. (2014). Adaptive fuzzy backstepping control for trajectory tracking of unmanned aerial quadrotor. In *Unmanned Aircraft Systems (ICUAS)*, 2014 International Conference on, Orlando, FL, USA, pp.920-927.
- [15] Gao, H., Liu, C., Guo, D., & Liu, J. (2015). Fuzzy adaptive PD control for quadrotor helicopter. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2015 IEEE International Conference on , pp. 281-286.
- [16] Choi, Y. C., & Ahn, H. S. (2015). Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. *IEEE/ASME transactions on mechatronics*, vol. 20, no.3, pp. 1179-1192.
- [17] Zhao, B., Xian, B., Zhang, Y., & Zhang, X. (2015). Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology. *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp.2891-2902.
- [18] Islam, S., Liu, P. X., & El Saddik, A. (2015). Robust control of four-rotor unmanned aerial vehicle with disturbance uncertainty. *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1563-1571.
- [19] Herrera, M., Chamorro, W., Gómez, A. P., & Camacho, O. (2015). Sliding Mode Control: An approach to Control a Quadrotor. In *Computer Aided System Engineering (APCASE)*, 2015 Asia-Pacific Conference on . IEEE, pp. 314-319.
- [20] Basri, M. A. M., Husain, A. R., & Danapalasingam, K. A. (2014). Robust chattering free backstepping sliding mode control strategy for autonomous quadrotor helicopter. *International Journal of Mechanical & Mechatronics Engineering*, vol. 14, no. 3, pp. 36-44.
- [21] Margun, A., Bazylev, D., Zimenko, K., & Kremlev, A. (2015). Trajectory-tracking control design and modeling for quadrotor aerial vehicles. In *Control and Automation (MED)*, 2015 23th Mediterranean Conference on. IEEE, pp. 273-277.
- [22] Yesildirek, A., & Imran, B. (2014). Nonlinear control of quadrotor using multi Lyapunov functions. In *2014 American Control Conference*. IEEE, pp. 3844-3849.

- [23] Raza, S. A. (2010). Design and control of a quadrotor unmanned aerial vehicle (Doctoral dissertation, University of Ottawa (Canada)).
- [24] Wang, L. X. (1999). A course in fuzzy systems. Prentice-Hall press, USA.
- [25] Surmann, H. (1996). Genetic optimization of a fuzzy system for charging batteries. IEEE Transactions on Industrial Electronics, vol. 43, no. 5, pp. 541-548.
- [26] Arslan, A., & Kaya, M. (2001). Determination of fuzzy logic membership functions using genetic algorithms. Fuzzy sets and systems, vol. 118, no. 2, pp. 297-306.
- [27] Sabzi, H. Z., Humberson, D., Abudu, S., & King, J. P. (2016). Optimization of adaptive fuzzy logic controller using novel combined evolutionary algorithms, and its application in Diez Lagos flood controlling system, Southern New Mexico. Expert Systems with Applications, vol. 43, pp. 154-164.
- [28] Martínez-Soto, R., Castillo, O., & Castro, J. R. (2014). Genetic algorithm optimization for type-2 non-singleton fuzzy logic controllers. In Recent Advances on Hybrid Approaches for Designing Intelligent Systems . Springer International Publishing, pp. 3-18.
- [29] Maldonado, Y., Castillo, O., & Melin, P. (2013). Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications. Applied Soft Computing, vol. 13, no.1, pp. 496-508.
- [30] Shi, Y., & Eberhart, R. C. (2001). Fuzzy adaptive particle swarm optimization. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on . IEEE., vol. 1, pp. 101 - 106.
- [31] Esmin, A. A. A., Aoki, A. R., & Lambert-Torres, G. (2002). Particle swarm optimization for fuzzy membership functions optimization. In Systems, Man and Cybernetics, 2002 IEEE International Conference on, vol. 3, pp. 6-pp.
- [32] Safari, S., Ardehali, M. M., & Sirizi, M. J. (2013). Particle swarm optimization based fuzzy logic controller for autonomous green power energy system with hydrogen storage. Energy conversion and management, vol. 65, pp. 41-49.
- [33] F. Soleiman Nouri, M. Haddad Zarif and M. M. Fateh (2014). Designing an adaptive fuzzy control for robot manipulators using PSO, Journal of AI and Data Mining (JAIDM), vol. 2, no. 2, pp. 125-133.

بهینه سازی توابع عضویت فازی کوادروتور توسط الگوریتم های بهینه سازی ذرات و ژنتیک

بیتا صفایی و سید کمال الدین موسوی مشهدی*

دانشکده برق، دانشگاه علم و صنعت ایران، تهران، ایران.

ارسال ۲۰۱۶/۰۳/۰۶؛ پذیرش ۲۰۱۶/۰۹/۲۲

چکیده:

کوادرورتور یک وسیله هوایی بی سرنشین شناخته شده است که کاربردهای گسترده نظامی و شهری دارد. کوادرورتور برخلاف ساختار ساده ای که دارد، ذاتاً ناپایدار است. به همین علت، طراحان همواره با چالش بزرگی در پایدارسازی و کنترل آن مواجه هستند. در این مقاله، توابع عضویت کنترل کننده های فازی کوادرورتور توسط الگوریتم های ملهم از طبیعت مانند الگوریتم بهینه سازی ذرات و الگوریتم ژنتیک بهینه سازی می گردند. در انتها، نتایج روش های پیشنهادی با یکدیگر مقایسه می شوند و یک مسیر برای بررسی توانایی ردیابی کنترل کننده های فازی طراحی شده توسط بهترین روش، تعریف می گردد.

کلمات کلیدی: کنترل کننده فازی، الگوریتم ژنتیک، توابع عضویت، الگوریتم بهینه سازی ذرات، کوادرورتور.