

The quadrotor dynamic modeling and study of meta-heuristic algorithms performance on optimization of PID controller index to control angles and tracking the route

Ata Jahangir Moshayedi¹, Mahyar Gheibollahi², Liefia Liao³

^{1,3}School of Information Engineering, University of Science and Technology, China

²Khomeinishahr Branch, Azad University, Iran

Article Info

Article history:

Received May 12, 2020

Revised Jun 7, 2020

Accepted Aug 4, 2020

Keywords:

Firefly algorithm
Genetic algorithm
Optimization algorithms
Particle swarm optimization algorithm
PID controller
Quadrotor
Robot
UAV
Unmanned flier

ABSTRACT

In the last decade, because of the unique specification of vertical fliers, scientists and researchers had a special focus on them. The particular abilities of these fliers can be mentioned such as: high maneuver ability, low expenses, decrease in radar identifier and low threat for the human life. They also have no limitation in dimension. Moreover, because of some applications like photography, topography, news coverage, study of power lines and aerology analysis, they can be notable for using. These fliers also are significantly important because of monitoring in urban regions, agricultural harvest and spray poison, illegal imports, exports administration and fire distinction in order to control the fire. Besides, seek and rescue missing people and also natural disasters can be pre-determined which causes stimulus investigators to act and put different topics in front of them. One of these fields is using meta-heuristic algorithms with the capability of using in control systems. The PID controller as a classic model has some limitations, but by optimization of special index through meta-heuristic algorithms, it has shown acceptable results. In this study, first, the history of vertical fliers and quadrotor are investigated. Then, after a review of overused methods, the quadrotor control has been done. Afterward, the cinematic and dynamic of quadrotor is presented. Next by designing of PID controller, PID index optimization by nature inspired algorithm, particle swarm optimization (PSO), genetic algorithms (GA), and firefly algorithms (FA) have been studied. Dynamic system, controller and mentioned optimization methods of PID controller index have also been implemented in MATLAB software. Also, with due attention to the comparison criteria the PID-PSO controller has shown the best performance. Next, by applying challenging routes, the stability of controller in the simulation is evaluated. Then, making quadrotor is done in practice along with introducing the used implementation of PID-PSO controller results on the real robot, and its stability is evaluated practically.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ata Jahangir Moshayedi,
School of Information Engineering,
University of Science and Technology,
Jiangxi, China.
Email: ajm@jxust.edu.cn

1. INTRODUCTION

In recent years, quadrotor usage to do different tasks is raising. High maneuverability, low expenses, decrease the use of radar identifier, low threat for human life and no limitation on dimension, are the factors of leading to more attention [1]. The quadrotor is an unmanned flier with the frame shape of X and H. It is

the subgroup of multicopter with the help of 4 fans to propel force, as a quad (four) called quadrotor. It is also change-able according to the frame and equipment. These supplies have four operators or rotor at the end of each arm which in their structure, the arms are symmetrical, same distance and same dimension. The rotors are Turning in clockwise and counter clockwise manner to move the body [2]. In general, controlling quadrotor movements is done by setting engine velocity. To control robot movements 3 angles of 'roll', 'pitch' and 'yaw' should be set. Moving up is done by using speed adjustments in every rotor through pulse and width medication to get favorable output [3]. The quadrotors with the help of four rotors and separate propeller and pairwise turning in the opposite direction make neutral torque and it leads to the resulting force [4, 5]. Generally, the flier is imagined in the space as a free object and with 6 degrees of freedom. So, the flier's position and its situation can be described completely. The position is containing three linear and three circular movements consisting of turning around X-axis (roll), turning around Y-axis (pitch) and turning around Z-axis (yaw) [4] to control and tune the Quad rotor controller and various methods are investigated which mentioned in our previous research paper [6]. In this study after modelling the Quadrotor and implying the real parameter to the model, the PID parameter are tuned with the help of three famous meta-heuristic algorithms and then after selecting the best performance the extracted values are loaded on real platform and the real performance and simulation are compared. In this research paper the Quadrotor modeling and its relative parameter which is used to simulate the Quadrotor are described in Section 1. Then, in Section 2 the PID controller and three Meta heuristic optimization algorithms named particle swarm optimization (PSO), genetic algorithms (GA), and firefly algorithms (FA) are elaborated. Section 4 describes the simulation result and comparison of mentioned algorithm performance. This comparison is based on quadrotor modeling in the MATLAB software to optimize the PID controller and the quadrotor stability with the four defined paths. In section 5, to compare the simulation and real situation, the extracted optimized parameter from PSO algorithm are implemented on real platform. Then, it compares the error between the planned path and real robot movement. The paper ended with the conclusion in section 6. The main key point in this study is the compression study of real performance and simulation of designed unmanned aerial vehicle (UAV) with the meta-heuristic algorithms in order to optimize the PID controller.

2. SYSTEM MODELING

The system modeling included cinematic and dynamic quadrotor, driven by Newton–Euler equations. In order to study the quadrotor movement Its needed to have two coordinate systems [7]. One related to the ground coordinate device and another related to space which requires Euler's angles. In Euler's angles the angles are resulted from the rotation of the body coordinate system and relative to the earth coordinate device (XYZ) as follows.

a. Euler angles

For finding position of quadrotor in the space, the frame coordinate system should be placed toward the ground coordinate system. The position of quadrotor in the space is described by location vector, but to define its orientation in the space, the Euler angles are required [7]. These angles are the results of spinning around axis: Φ : roll angle (rotating around X-axis), θ : pitch angle (rotating around Y-axis), ψ : yaw angle (rotating around Z-axis). See Figure 1.

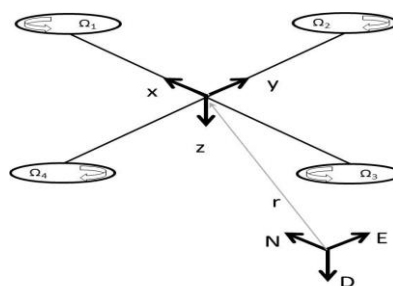


Figure 1. Cinematic model of the quadrotor body

As its name implies Earth frame considering (Figure 1) as an inertial frame it uses the N-E-D notation where the axes point to the north, east and downwards respectively. Besides, the body frame is at the center of the quadrotor body, with its x-axis aiming towards propeller 1, y-axis pointing towards propeller 2 and the z-axis is indicating to the ground. The quadrotor orientation is described by using roll, pitch, and yaw angles (φ , θ and ψ) representing rotations about the x, y, and z-axes, respectively. Considering the order of

rotation to be roll (φ), pitch (θ) then yaw ψ), the rotation matrix r which is derived based on the sequence of principle rotations shown in (1).

$$R = \begin{bmatrix} C\theta C\psi & S\phi S\theta S\psi & C\phi S\theta C\psi + S\phi S\psi \\ C\theta S\psi & S\phi S\theta C\psi + C\phi S\psi & C\phi S\theta S\psi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \quad (1)$$

As it shown in (1), c and s represent \cos and \sin as the derivation of the rotation. The rotation matrix R will be used in framing the dynamics model of the quadrotor, and its importance is due to the fact that some states are measured in the body frame (e.g. the thrust forces shaped by the propellers) although some others are measured in the inertial frame (e.g. the gravitational forces and the quadrotor's position). Therefore, to have a relation between both types of states, observation was done for a transformation from one frame. To obtain data about the angular velocity of the quadrotor, typically an on-board inertial measurement unit (IMU) is hired to have the velocity in the body coordinate frame. Then to relate the Euler rates, as in (2):

$$\dot{\eta} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T \quad (2)$$

that are measured in the inertial frame and angular body rates $\omega = [p \quad q \quad r]^T$, a transformation is needed as (3):

$$\omega = R_r \dot{\eta} \quad (3)$$

where shown in (4):

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

In (4), 'S' is the abbreviation of 'sin' and 'C' is the abbreviation of 'cos' and parameters ϕ , θ , ψ are roll, pitch, and yaw angles, respectively. Its noted that Around the hover position, small angle assumption is made where $\cos \varphi \equiv 1$, $\cos \theta \equiv 1$, and $\sin \varphi \equiv 0$, $\sin \theta \equiv 0$. Thus R_r can be simplified to an identity matrix I [7]. To present of modeling, the system is required of torque calculation of quadrotor, swirl movement equations, the non-gravitational force of quadrotor and dynamic of rotors, which is describing in the following.

b. The quadrotor torque

Among applied force to turn, there is a force, called aerodynamics or lifting force and there is a produced torque, called aerodynamic torque which has shown in (5).

$$M_B = \begin{bmatrix} IK_f(-\Omega_2^2 + \Omega_4^2) \\ IK_f(\Omega_1^2 - \Omega_2^2) \\ K_M(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (5)$$

In (5) M_B is the general equation of torque applied on a quadrotor, l is arm quadrotor, K_f is an aerodynamic force and Ω_1 is the velocity of the first rotor, Ω_2 is the velocity of the third rotor, and Ω_4 is the velocity of the fourth rotor.

c. Swirl movement equations

Swirl movement equation of quadrotor is taken from Newton's second law and ground inertia frame which is shown in (6) [7].

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \quad (6)$$

As it has shown in (6), $r = [x \ y \ z]^T$, which represent the distance of quadrotor to the inertial frame, m (quadrotor weight), g (quadrotor weight) and F_B (non-gravitational force applied on quadrotor) and R is the connection of two frames coordinate system of quadrotor toward coordinate system [7].

d. Non-gravitational force applied on the quadrotor

Whenever the quadrotor is in the horizontal orientations, the produced force by turning fans is the only non-gravitational force applied on it which is corresponding square angular velocity of fan (6). So, no non-gravitational force is effective on quadrotor. In this case F_B is defined such as (7).

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (7)$$

In (7), Ω_1 is the first Ω_2 is second, Ω_3 is the third rotor velocity, Ω_4 is the fourth rotor velocity, and K_f is the aerodynamic force [7].

e. Rotors dynamics

Usually brushless motor (BLDC) used in the quadrotor are DC motors or brushless, which have high torque and low friction. The dynamic of DC and brushless motor, in normal conditions, is like the regularly fixed engines, which has shown in (8).

$$v = \frac{R_{\text{mot}}}{K_{\text{mot}}} J_r \dot{\Omega}_i + K_{\text{mot}} \Omega_i + K_M R_{\text{mot}} \Omega_i^2 \quad (8)$$

In (8), K_{mot} is motor torque constant, K_M is aerodynamic moment constant, R_{mot} is motor circuit resistance and J_r is rotor's inertia [7].

f. The state space model

Obtaining the state space model, helps to design of the controller and it is required X state variable and their corresponding element.

g. X state variable

Considering the state vector of the quadrotor to be (9):

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \quad (9)$$

which is mapped to the degrees of freedom of the quadrotor in the following manner (10) [7].

$$X = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}]^T \quad (10)$$

In (10), x , y , and z are the positions; ϕ , θ , and ψ are the Euler angles; and their derivation which indicates the component velocity. The state space vector is notable because of the capability of showing position angle and the linear velocity of quadrotor [7].

h. Input variable of the state vector

The parameter U as a control input vector containing of four inputs; U_1 through U_4 is defined as, shown in (11).

$$U = [U_1 \ U_2 \ U_3 \ U_4] \quad (11)$$

According to (11), each input it can be shown as (12)–(16).

$$U_1 = K_f(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (12)$$

$$U_2 = K_f(\Omega_2^2 + \Omega_4^2) \quad (13)$$

$$U_3 = K_f(\Omega_1^2 - \Omega_3^2) \quad (14)$$

$$U_4 = K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (15)$$

In the above relations, k_f is aerodynamic forces, k_m is fixed torque and Ω is the angular velocity resulted from rotors. Finally, the relation of (12)–(15) can be shown as the matrix form as shown in (16) [7].

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ -K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (16)$$

According to relation (16), u_1 is the power of the four routers that gives the quadrotor height from Z to Z . u_2 is the result of the difference in torque between engine and 4, which results in a quadrotor rotation around the roll angle from Φ to Φ . u_3 is the difference of torque between the motors 7 and 3 that rotates the quadrant around the angle of the pitch from θ to θ . u_4 results from the torque difference between the two

pair of motors clock and clockwise and it rotates around the angle of your from ψ to ψ . The speed of the rotors can be calculated by the control input. Therefore, the inverse relation between the control input and the rotor speed can be obtained as (17) [8].

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & 0 & \frac{1}{2K_f} & \frac{1}{4K_M} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & -\frac{1}{4K_M} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & \frac{1}{4K_M} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & -\frac{1}{4K_M} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (17)$$

In the above relations, Ω is the speed of constant torque, k_m is aerodynamic force and K_f is the angle of rotors. In the relation (17) the control inputs can be calculated by taking root from the rotor speed [9]. At the end, by taking the square root of that, the velocity of rotors can be calculated from the control inputs as shown in (18):

$$\begin{aligned} \Omega_1 &= \sqrt{\frac{1}{4K_f} U_1 \frac{1}{2K_f} U_3 + \frac{1}{4K_M} U_4} \\ \Omega_2 &= \sqrt{\frac{1}{4K_f} U_1 \frac{1}{2K_f} U_2 + \frac{1}{4K_M} U_4} \\ \Omega_3 &= \sqrt{\frac{1}{4K_f} U_1 \frac{1}{2K_f} U_3 + \frac{1}{4K_M} U_4} \\ \Omega_4 &= \sqrt{\frac{1}{4K_f} U_1 \frac{1}{2K_f} U_2 + \frac{1}{4K_M} U_4} \end{aligned} \quad (18)$$

Taking the square root of that, the rotors' velocities can be calculated from the control inputs as follows. In the above equation, K_f is aerodynamic force, K_M is constant torque, and Ω is the angular velocity of rotors.

i. Rotational equation of motion

It leads to dynamic equations of the system are described in (19) to (21).

$$\ddot{\phi} = \frac{1}{I_{xx}} U_2 - \frac{J_r}{I_{xx}} \dot{\theta} \Omega_r + \frac{I_{yy}}{I_{xx}} \psi \dot{\theta} U_2 - \frac{I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \quad (19)$$

$$\ddot{\theta} = \frac{1}{I_{yy}} U_3 - \frac{J_r}{I_{yy}} \dot{\phi} \Omega_r + \frac{I_{zz}}{I_{yy}} \dot{\phi} \dot{\psi} - \frac{I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi} \quad (20)$$

$$\ddot{\psi} = \frac{1}{I_{zz}} U_4 - \frac{I_{xx}}{I_{zz}} \dot{\theta} \dot{\phi} + \frac{I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \quad (21)$$

By select the control input vector U , it is clear that the rotational subsystem is fully actuated, it is only reliant on the rotational state variables x_1 to x_6 that correspond to ϕ , ϕ , θ , $\dot{\theta}$, ψ , $\dot{\psi}$ respectively.

j. Translational equation of motion:

Considering (10) and (6) the total inserted torque on the UAV can be written as (22) [7]:

$$F_b = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \quad (22)$$

which after replacing the parameter in (4), (23) to (25) can be substituted:

$$\ddot{x} = \frac{-U_1}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \quad (23)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi) \quad (24)$$

$$\ddot{z} = g - \frac{-U_1}{m} (\cos \phi \cos \theta) \quad (25)$$

Based on mentioned equations, J_r is rotors torque, I_{XX} , I_{YY} , I_{ZZ} are the torque from inertia in the major axis of body frame $\dot{\phi}$ $\dot{\theta}$ $\dot{\psi}$ are velocity, $\dot{\psi}$ $\dot{\phi}$ $\dot{\theta}$ are instant momentum of angular Euler [7]. Finally, according to the required parameters by physical simulation of quadrotor in CATIA software, some quadrotor parameters are computed and the rest of them are estimated by similar physics of robot's evaluation which have shown in Table 1. As it has presented in Table 1, I_{XX} , I_{YY} , I_{ZZ} and I_{rotor} are the moments of inertia around X-axis, Y-axis, Z-axis and motor axis respectively, m is the weight of motor, b is the propeller result force index, d is the propeller counterpoise index, g is the earth gravitational force.

Table 1. Quadrature parameters tested

No	Parameters	Content	Unit
1	I_{XX}	$5.49 * 10^{-4}$	kgm^2
2	I_{yy}	$5.49 * 10^{-4}$	kgm^2
3	I_{zz}	$8.28 * 10^{-4}$	kgm^2
4	I_{rotor}	$5.51 * 10^{-5}$	kgm^2
5	m	0.620	kg
6	b	$2.9 * 10^{-5}$	Ns^2
7	l	0.14	m
8	d	$1.1 * 10^{-6}$	Nms^2
9	g	9.81	Ms^{-2}

3. PID CONTROLLER

The PID controller belongs to those group of controllers, which work based on the feedback, and more famous as a classical controlling and, it's design and implementation is very customary [8]. One of the great challenges of the flier robot pilots is obtaining suitable PID index according to the physical structure of quadrotor. Most of the quadrotor's pilots try the false and true method which leads to make so many problems and hardship to the robot. So, obtaining the PID indexes is significant [9]. To control of mentioned parameters, first, PID controller by meta-heuristic algorithms have been used which have been described in the following. Equation (26) is the general equation for PID controller which K_p is proportional, gain, K_d is derivative gain and K_i is integrator gain [10].

$$U(t) = K_p e(t) + K_i \frac{de(t)}{dt} + K_d \int e(t) dt \quad (26)$$

According to (10), the control inputs, and PID controller of the quadrotor are defined such as (11) to (26) [11, 12]. In the above equations, parameters K_p , K_{dz} , K_{iz} respectively are controlled proportional, integrator and derivate gain, in height control. The indexes of $K_{p\phi}$, $K_{d\phi}$, $K_{i\phi}$ are control gains to control 'roll'. $K_{p\theta}$, $K_{d\theta}$, $K_{i\theta}$ are controlled gains to control 'pitch' and $K_{p\psi}$, $K_{d\psi}$, $K_{i\psi}$ are control gains to control 'yaw' [13, 14]. In order to find the PID parameter, Ziegler–Nichols rules are an innovative method of controlling parameters in obtaining values of integral derivative proportional controller coefficients. This method has been developed by Ziegler and Nichols. In this state, the system reaches the boundary of satiability and the period of k_u fluctuation and its output are fined oscillating [15]. To use this method, the quadrotor linear dynamic should be modeled. Table 2 shows the Ziegler–Nichols rules for adjusting control gains [16].

Table 2. Coefficients of PID controller response with the step input function

No	Parameters	K_p	K_d	K_i
1	Pitch	97	1.5	0.8
2	Roll	63	3.7	0.2
3	Yaw	43	4.8	0.1

3.1. PID optimization

In order to optimize the PID parameter, an appropriate cost function must be considered in order to appropriately quantify the control interest in the control algorithm. The cost function can be defined on the basis of desirable characteristics. For example, the amount of its excess is specified by M_p , time to ascent specified by t_r lasting response value is defined by e_{ss} and settling time is defined by t_s . The penalty

function method is used to define the cost function for optimization algorithms. According to this method, the cost function is defined as (27):

$$J_i = e^{-\beta}(M_p + e_{ss}) + (1 - e^{-\beta})(t_s + t_r) \tag{27}$$

As mentioned in (27), β value can be changed. Considering the same condition of influencing the intended attributes on the cost function, this value is equal 0.69 [17–20]. Based on the cost function definition, the parameters of each response of optimization problems such as K_{pz}, K_{dz}, K_{iz} are respectively the proportional-derivative and integration gains in control height. The $K_{p\phi}, K_{d\phi}, K_{i\phi}$ are the control gain for roll control. $K_{p\theta}, K_{d\theta}, K_{i\theta}$ are the control gain for pitch control and $K_{p\psi}, K_{d\psi}, K_{i\psi}$ are the control gain for yaw control which is twelve parameters altogether. In other words, each response of problems are indicating twelve values for PID index and the duty of the defined algorithms is finding the best answer considering all introducing defined parameter's [15]. In PID indexes optimization, the nature-inspired algorithm has used [16] which are described shortly as following.

3.1.1. PSO algorithm

The PSO algorithm is taken from the aggregation of particles and it was inspired from bird's troop and fishes. These kinds of birds are putting in a searching space of pixel table in random and in every repetition, the closed neighbor of particle has selected and speed of the particle is replaced with the speed of its closest neighbor [17]. This action causes the fast movement of a group in the same direction with uncertain movement and converges without changing [18].

3.1.2. GA

Gene is the main saving unit in genetic. Inside cells, genes are connected to each other to produce chromosome. This new cell called 'diploid' and it caused through meiosis. In meiosis, each chromosome of the diploid cells makes a copy of themselves. Then, chromosomes group (new and main) are compounded to each other [21]. After it, chromosomes will separate one more time and 4 diploid cells are made. Mutation can take place in each stage. Every single change in chromosomes transfer to the children [22].

3.1.3. FA

FA is based on flickering which is discussed up to now. The base of flickering is just like a firefly's behavior. The flickering of these insects is based on a biological illumination process. So, determination of correct function of the signal producing system is yet to be discussed. The base of flickering is to create a connection and attract the opposite sex or bait to hunt. In general, flickering can be used as a protection alert mechanism [23]. According to the related light criteria, the light intensity I , in the distinct distance of a toward the source light generation, has the reverse ratio with the square value of this distance.

In the other words, after increasing distance from resource and light attraction by the air, it's intensity goes weak [24, 25]. In order to compare optimization methods of parameters the same conditions between mentioned algorithms have been considered. Table 2 has shown the algorithm repetition and consideration.

As it is shown in Table 3, the components which can be defined in the same way, are including the population size in the most algorithm repetition. Some of parameters also are related to the internal setting of algorithms and they are not common between algorithms. These parameters have been determined and set by the help of values in the research articles or they set experimentally which have shown Table 4.

In Table 4, the PSO algorithm parameters, σ is the particle velocity, $c_1 c_2$ are the local impact indexes. With attention to the local particle and the best major particle, w is the added weight index in the relation of speed update. In the GA algorithms, p_c is the possibility of crossing, p_m is the possibility of changing in bit, M_u is the mutation index. The parameters of FAs, I_0 is the light source intensity and γ is the light absorption index. β_0 is the equivalent of each firefly's attraction rate, α is the random parameters, and δ is the uniform distribution changing range.

Table 3. The values of fixed parameters for testing three algorithms

Name	Amount
The size of the initial population	50
Most replicated algorithms	30

Table 4. Specific parameters values for setting each of the three algorithms

Algorithm	PSO		GA		FA	
The parameters of each algorithm and their values	w	1	P_c	0.5	γ	1
	c_1	2	P_M	0.3	β_0	2
	c_2	2	δ	0.05	α	0.2
	σ	0.1	M_u	0.5	δ	0.05

4. SIMULATION

After the modeling, the Quad rotor the first step of this research is to test the model performance with three Meta-Heuristic Algorithms of PSO, GA, and FA to determine the PID controller parameters.

4.1. PSO algorithm

In order to determine PID controller index, optimized by PSO algorithm, to control pitch, roll, and yaw for step function the algorithm optimization process has shown in Figures 2(a)–(c). It is noted that with respect to the direct infernal relation of Euler angles the best cost is recognizable in each figure. As it is shown in Figure 2, the algorithm in the control of roll, pitch, and yaw angles has proper function and overshoot rate, steady state error, settling time, have shown low increasing time and the algorithm has reached to damping in 24 repetitions. With attention to Figure 2, the PID index of particles swarm optimization has shown in Table 5. With respected to the controller performance presented in Figure 2, the functional criteria obtained from the PSO algorithm have shown in Table 6.

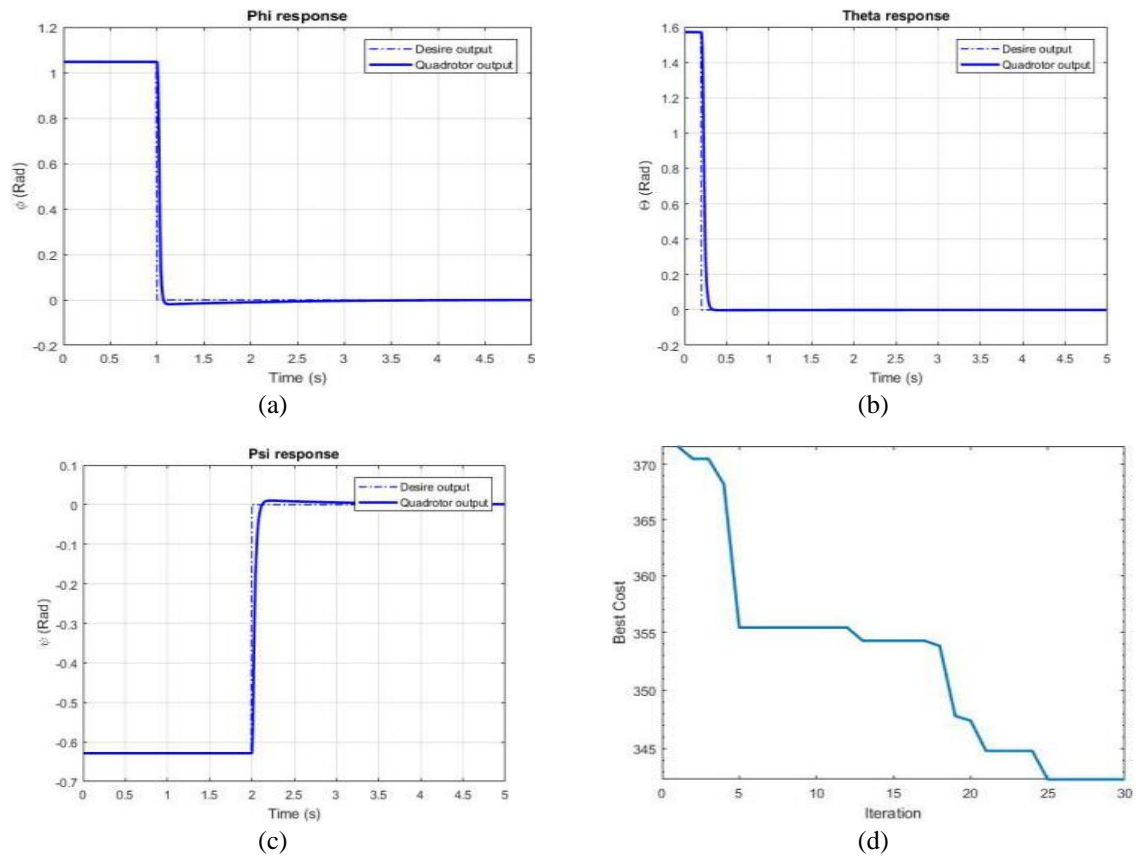


Figure 2. PID-PSO: (a) pitch, (b) roll, (c) yaw, and (d) best cost

Table 5. PID-PSO controller response coefficients to stack function input

No	Parameter	K_I	K_P	K_D
1	Pitch	0.5492697	0.7065999	53.78851
2	Roll	0.6367265	0.9605094	53.85274
3	Yaw	0.626404	2.83406	79.40869

Table 6. PID-PSO controller performance metrics

No	Parameter	Rise time	Settling time	Steady state	Overshoot
1	Pitch	0.0700	0.0679	0.0005	1.8186
2	Roll	0.1000	0.0994	0.0000	0.0721
3	Yaw	0.0900	0.0858	0.0011	1.6330

In Table 6, the maximum value of overshoot criteria is related to the pitch parameter and the minimum value is related to the roll. In the steady state error criteria, almost all of them are zero. In the settling time, the maximum value is related to the pitch parameter and the minimum value is related to the roll and increasing time, the maximum value is related to the roll and the minimum value belongs to the pitch.

4.2. GA

The results of PID controller optimizing by GA in controlling pitch, roll, and yaw angles, to the step function, have shown in Figures 3(a)–(c) and the algorithm optimization process has shown in Figure 3(d). As it has been shown in Figure 3, the algorithm has no good operation in to control of roll, pitch, and yaw angles. So, it could not track the function properly.

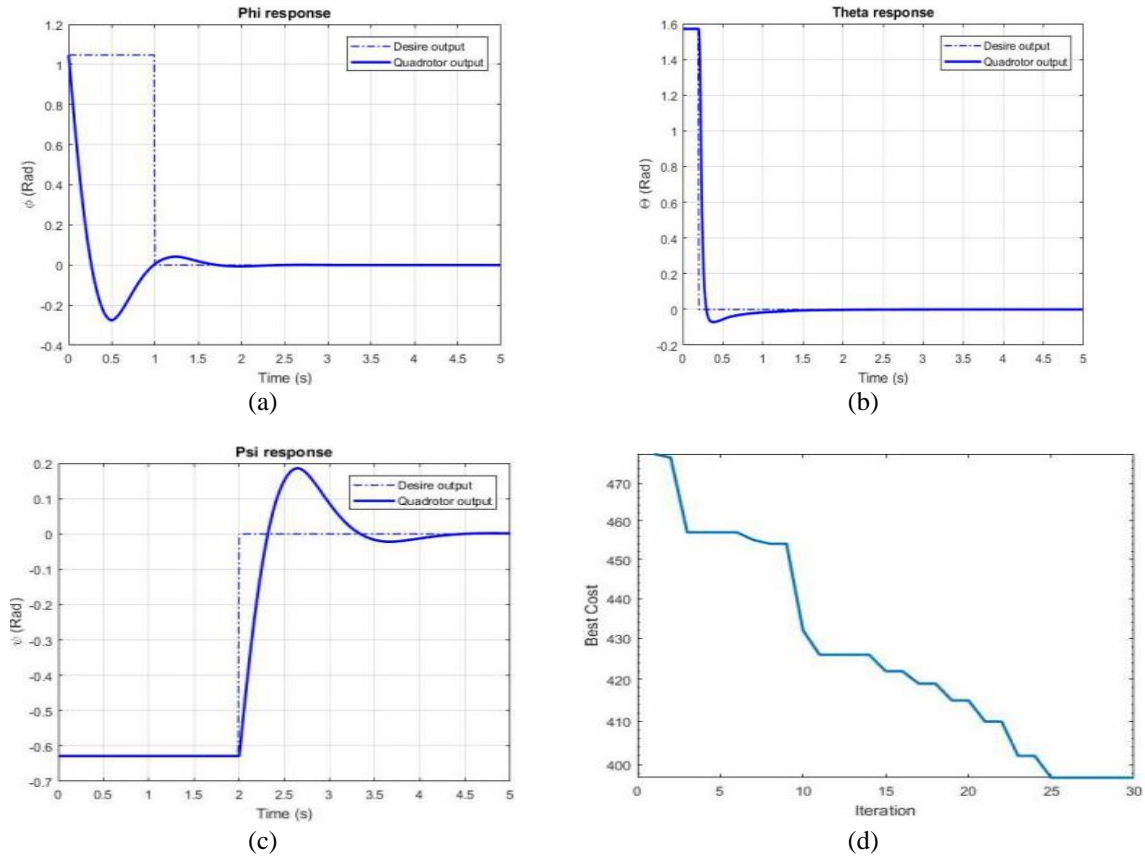


Figure 3. PID-GA: (a) pitch, (b) roll, (c) yaw, and (d) best cost

The overshoot rate, steady state error and settling time, shows the very low increasing time and the algorithm in 25 repetitions has reached to damping, with attention to figures the PID indexes of GA and controller response a have shown in Tables 7 and 8. In Table 8, the maximum value of overshoot criteria is related to yaw parameter and the minimum value is related to the roll. In steady state error criteria almost all of the values are zero. In settling time, the maximum value belongs the yaw parameter and the minimum value belongs the pitch and in rise time, the maximum value connected to the pitch parameter and the minimum value connected to the roll.

Table 7. The PID-GA controller coefficients

No	Parameter	K_I	K_P	K_D
1	Pitch	0.62179	6.95585	38.0361
2	Roll	0.298181	0.645642	42.0017
3	Yaw	0.509005	7.23946	47.6407

Table 8. The PID-GA controller response coefficients to the step input

No	Parameter	Rise time	Settling time	Steady state	Overshoot
1	Pitch	0.7400	0.1118	0.0000	26.3005
2	Roll	0.1000	0.3389	0.0000	4.4842
3	Yaw	0.2900	1.1238	0.0017	29.6089

4.3. FA

The results of PID controller, which optimized by FA in controlling yaw, roll and pitch angles to the step function input are presented in Figures 4(a)–(c) and the algorithm optimization process has shown in

Figure 4(d). According to the Figures 4(a), (b), (c) the PID indexes of FA have shown in Table 9. As it is shown in Figure 4, the algorithms had a good performance to control of roll, pitch and yaw also overshoot rate, steady state error and settling time, shows the very low increasing time. Moreover, algorithm in 27 repetitions have reached to damping. The with attention to the controller function, in Figures 4(a)–(c) the operating parameters of the FA have shown in Tables 9 and 10.

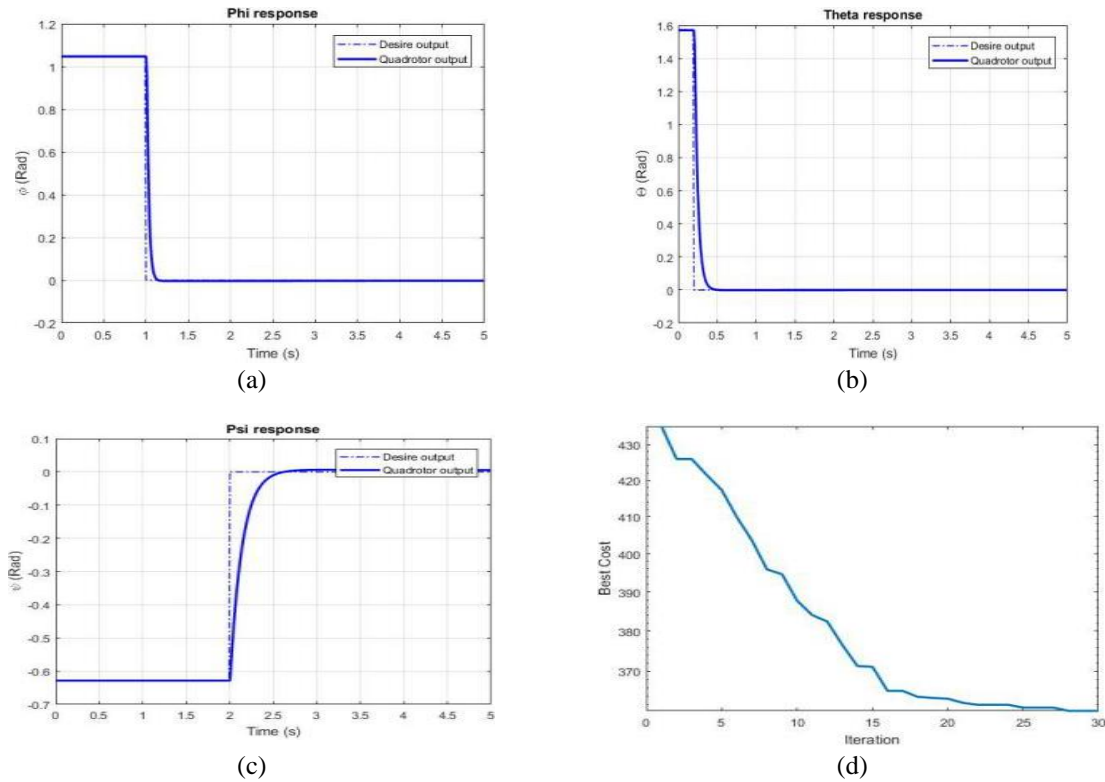


Figure 4. PID-FA: (a) pitch, (b) roll, (c) yaw, and (d) best cost

Table 9. PID-FA controller response coefficients to input of the step function

No	Parameter	K_I	K_P	K_D
1	Pitch	0.0578509	68.3606	1.71204
2	Roll	0.665255	75.3099	3.22299
3	Yaw	0.01	42.6745	6.06757

Table 10. Performance metrics for the PID-FA controller

No	Parameter	Rise time	Settling time	Steady state	Overshoot
1	itch	0.10	0.0978	0.0026	0.3494
2	Roll	0.1700	0.1693	0.0000	0.0510
3	Yaw	0.3500	0.3453	0.0055	1.0326

In Table 10, the maximum value of overshoot criteria belongs to the yaw parameter and the minimum value related to the roll parameter. In steady state error, almost all values are zero. In settling time, the maximum value is connected the yaw parameter and the minimum value is connected the pitch. In the increasing time, the maximum value is related to the pitch.

4.4. Comparisons of PID controller optimization methods

As already stated, the five criteria of overshoot, steady state error, settling time, rising time are used to compare the coefficients of PID tuning via the mentioned algorithms. this comparison are summaries in Tables 11–13 for pitch, roll and yaw parameters. As it shown in Table 11, the controller performance in pitch parameter, the PID_FA, PID_PSO algorithms show better performance in overshoot than the two Ziegler–Nichols methods and PID_GA. The value of steady state error in all algorithms are very low and the settling time parameters, for the PID_PSO and PID_FA, perform better in compare to the two other algorithms. Table 12 shows the comparison of the controller’s performance in the roll parameter. Which it shows the PID_FA and PID_PSO algorithms, there is a better performance in the overshoot and settling time criteria have better performance towards Ziegler–Nichols and PID_GA.

Table 11. Controller performance in the pitch parameter

No	Pitch parameter	Overshoot	Steady state error	Settling time	Rising time
1	Ziegler–Nichols	13.8848	0.0000	0.5569	0.1600
2	PID-PSO	1.8186	0.0005	0.0679	0.0700
3	PID-GA	26.3305	0.0000	0.1118	0.7400
4	PID-FA	0.3494	0.0026	0.0978	0.1000

Table 12. Controller performance in roll parameter

No	Pitch parameter	Overshoot	Steady state error	Settling time	Rising time
1	Ziegler–Nichols	1.3338	0.0000	0.6084	0.6100
2	PID-PSO	0.0721	0.0000	0.0994	0.1000
3	PID-GA	4.4842	0.0000	0.3389	0.1000
4	PID-FA	0.0510	0.0000	0.1693	0.1700

Table 13. The controller performance in the yaw parameter

No	Pitch parameter	Overshoot	Steady state error	Settling time	Rising time
1	Ziegler–Nichols	5.0736	0.0002	0.0758	0.0900
2	PID-PSO	1.6330	0.0011	0.0858	0.0900
3	PID-GA	29.6089	0.0000	1.1238	0.2900
4	PID-FA	1.0326	0.0055	0.3453	0.3500

The steady state error in all algorithms is zero and Settling Time Criteria in PID_PSO and PID_GA is the least amount compare with other algorithms. As it decapitated in Table 13 from the performance of the controllers, the PID-PSO and PID_FA algorithm have the better performance in overshoot, towards Ziegler–Nichols and PID_GA methods. The steady state error is too less in all algorithm and for the settling time parameters, PID_PSO algorithm and Ziegler–Nichols have the better performance in comparison with PID_GA and PID_FA methods. Then the result shows the meta heuristic algorithm are act better than Ziegler–Nichols. then to have the to have a better study of algorithms, the other comparing criteria including three parameter of average computation time, success rate and standard deviation with the following description are considered:

- **The calculation time average:** the time of calculating to turn algorithm once.
- **Success rate:** The success rate of each algorithm is to achieve optimality parameters.
- **The standard deviation of obtained responses:** the standard deviation is a statistical index which is equal to the second root of variance. It shows the deflection of members in a complex from the average value. As much as the standard deviation is closer to zero, the data values dispersion is lower, and as much as this value is more, the data values dispersion is more the result of compared parameters are summarized in Table 14.

As it is shown in Table 14, PID-PSO has the least time, Success rate and the least standard deviation. So PID-PSO shows the better performance. As the last stage of simulation and to evaluate of PID-PSO controller, some routes with different challenges have applied to them and the function of them in the face of applied route is investigated. In the following, by using a circular, spiral, and 8-shaped and spline routes to the system (Table 15), the controller performance is evaluated (Figure 5). The average of tracking error in examination routes has shown in Table 15. As it has been shown in Figure 5, the system function in tracking routes of resource, shows the proper performance. But in some routes, some errors have been seen. According to Table 15, the most error is related to 8-shaped (Figure 5 (c)) and the least error is related to spiral-shaped.

Table 14. The pitch parameter and controller performance

No	Pitch parameter	Irritation time	Success rate	Standard deviation
1	PID-PSO	141	%94	0.09
2	PID-GA	157	%72	0.31
3	PID-FA	200	%86	0.42

Table 15. The value of the main error of tracking the test path

No	Path	Path mathematical equation	Error
1	Circle	$x = 0.3 \sin(0.3t)$ $y = 0.3 \cos(0.3t)$	7.38
2	spiral	$x = 0.1t \cos(0.3t)$ $y = 0.1t \sin(0.3t)$	3.31
3	8-shaped	$x = 0.5 \sin(0.3t)$ $y = 0.5 \sin(0.3t) \cos(0.3t)$	8.41
4	spline	$x = \{0, 0.2, 0.1, -0.3, 0.3\}$ $y = \{0, 0.4, -0.2, 0, 0.3\}$	3.82

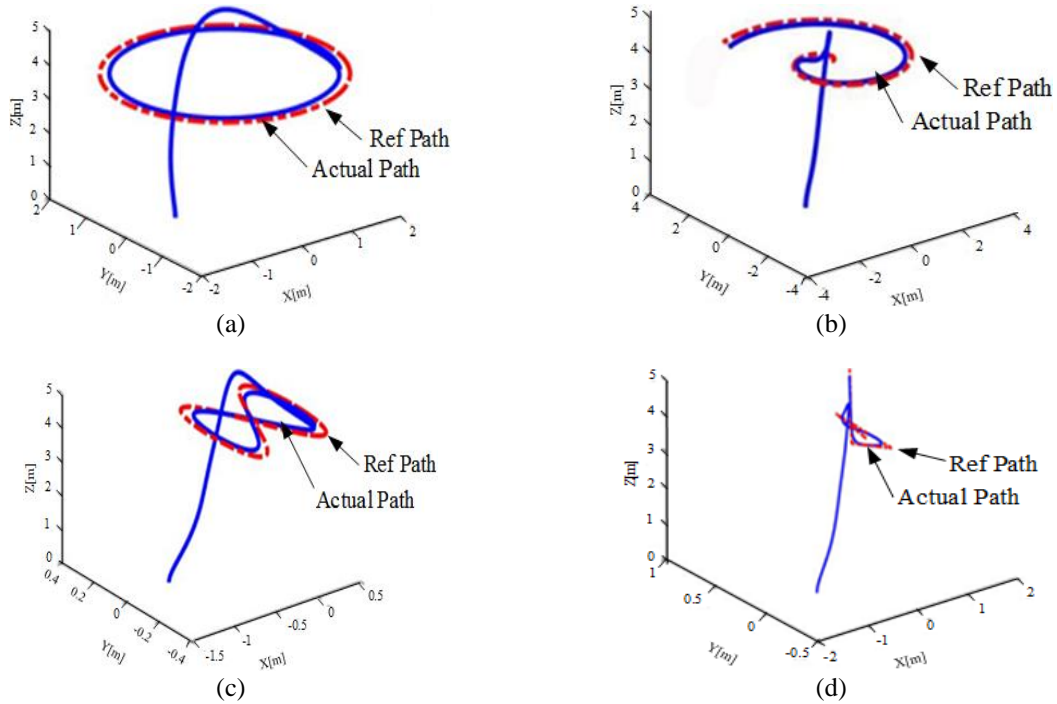


Figure 5. Quad rotor simulation result in tracking various paths: (a) circle path, (b) spiral path, (c) shaped path, and (d) spline path

5. STRUCTURE ASSEMBLY

5.1. Index practical implementation

The second stage in the designed plan is PID index implementation in practice and in the real size of quadrotor. As it is shown in Table 16 and Figure 6, gained values on the quadrotor have tested. The quadrotor consists of body, engine, controller system and energy supply. Designing a quadrotor often are related to its application. which all mentioned components have selected, for example, to carry the weight, the high-power consuming motor needing high power battery have been considered. For speed racing, the H shaped body with the low weight and small size have used. So, based on the application, the shape of designing is different. In this study, a sample of quadrotor consists of mentioned components in Table 16 have considered.

For the implementation of data, the software ‘mission planner’ has been used. Mission planner is an open source software, uses the windows operating system, installed on the ground station computer system. Mission planner setting steps:

- a. selecting kind of robot frame
- b. accelerometer calibration
- c. compass calibration
- d. board and power supply selection
- e. radio calibration
- f. reference route selection [26, 27]

Table 16. Specifications of quadrotor components

NO	Components quadrotor	Model
1	Airframe	Tarot TL280H
2	Motor	Air 2205 2000 KV
3	Propeller	10 * 4.5 mm
4	Control Board	ArduPilot APM 2.8
5	GPS	Ublox NEO-7M
6	Telemetry	433 HZ
7	Speed Control	T-Motor Air 15 A
8	Radio Transmitter	TX: NET-Q118G
9	Radio Receiver	RG831B, 8ch 2.4 GHz
10	Battery	PULSE 2250 mAh 3S LiPo Battery



Figure 6. Quadrotor image is made

5.2. Selecting the reference route for quadrotor

- **The first step, setting the start point position:** The start point position is the place which robot is ready to fly in there. It means, if RTL has been considered, the robot will return to the start point.
- **The second step, setting route point:** To do this in the drop-down menu of the flight plan, the desired command in each row (point, height,) should be selected. After initial setting in mission planner and setting the PID indexes. the used operations and routes in simulation containing Figure 5 such as circle, spiral, 8 shaped and spline have tested in real which results of real tests can be seen in Figure 7 and Table 17.

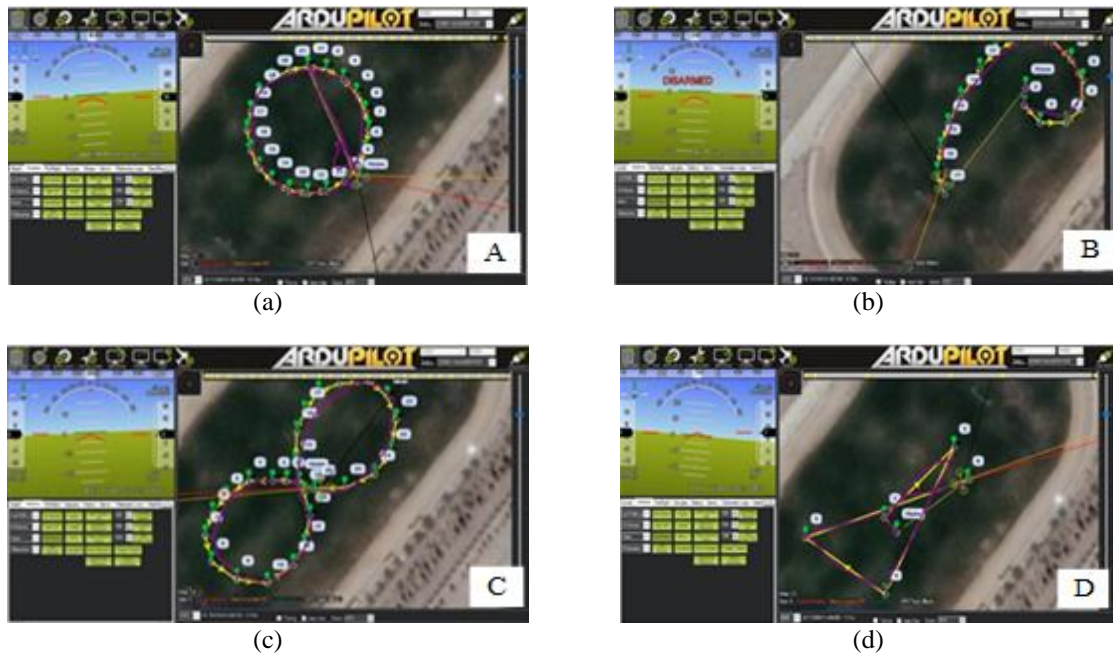


Figure 7. Real quadrotor path tracking performance in various path:
(a) circle path, (b) spiral path, (c) shaped path, and (d) spline path

The results of tests in the circle path (Figure 7(a)), spiral path (Figure 7(b)), 8-shaped paths (Figure 7(c)) and spline path (Figure 7(d)), show that the robot has a good performance in the tracking of path points and also corners. In some points, the path tracking was not done in a good way, because of some problems, such as GPS module error about 4-meter disconnection of the signal from telemetry module and system disturbance through the wind. In these tests, according to the weather station site, the wind speed was almost 7 km/h and the real robot moved up with a maximum height of 15 meters.

After implementation of control parameters on the real robot, the tracking path error in a practical test using mission planner software [27] has been shown in Table 17. According to the table, the maximum percentage of error is related to the 8-shaped path (Figure 7(c)) and the minimum error is related to the spline path (Figure 7(b)). Some of the result of the real experiment have been shown in YouTube link [28].

Table 17. Traceability tracking errors in practical tests

No	Path	Error
1	Circle	14.74%
2	Spiral	12.43%
3	8-shaped	18.84%
4	Spline	11.30%

6. CONCLUSION

In this study, by using Newton–Euler method, the quadrotor dynamic has been studied and also the general equations of the system have been described. The control of quadrotor, a PID controller indexes, the nature-inspired optimization algorithm including PSO, GA, and FA has been introduced and implemented in the simulation. The simulation of the designed model has been done in MATLAB software and quadrotor

stability parameters have been investigated. The PSO algorithm, with attention to the comparing criteria such as overshoot rate, error, setting time, increasing time, calculating time, and standard deviation has operated as the best algorithm. In order to verify evaluation for the simulated system, four different routes with different challenges have been applied to the robot as different routes and the robot function has been studied in tracking the routes. Then, the simulated parameters of real quadrotor have been implemented and the robot functions in challenging situations with the real routes have been investigated. With attention to the test, in simulation mode, the most percentage of error is related to the 8-shaped path and the least error is related to the spiral path (Figure 5). In the practical mode, the most percentage error is related to the 8-shaped and the least error is related to the spine (Figure 7) and this difference can be in the result because of data sending error (telemetry) and GPS. With attention to the all direction, it can be said with considered indexes for the system, it has a good performance in stability and tracking the resource route and its applicable in various application in the robotic field [29–31]. The result shows the MATLAB capability and its navigation on the real platform.

REFERENCES

- [1] S. Bouabdallah, "Design and Control of Quadrotors With Application To Autonomous Flying," Doctoral Thesis, École Polytech. Fédérale Lausanne, À La Fac. Des Sci. Tech. L'Ingénieur, 2007.
- [2] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," M.S. Thesis, Department of Automatic Control, Lund University, 2008.
- [3] D. R. Borah and L. Debnath, "A review on Quadcopter Surveillance and Control," *ADBU-journal Eng. Technol.*, vol. 4, no. 1, pp. 116–119, 2016.
- [4] A. Gibiansky, "Quadcopter Dynamics, Simulation, and Control Introduction Quadcopter Dynamics," pp. 1–18, 2012.
- [5] J. G. Leishman, "The breguet-richet quad-rotor helicopter of 1907," *Vertiflite*, vol. 47, no. 3, pp. 58–60, 2002.
- [6] A. J. Moshayedi and M. Gheibollahi., "Review of The Methods Used to Control Quadrotor," in *9th National Conference of Mechanical Engineering*, 2017, pp. 20-26.
- [7] D. Hanafi, et al., "Simple GUI Wireless Controller of Quadcopter," *International Journal of Communications, Network and System Sciences*, vol. 06, no. 01, pp. 52–59, 2013.
- [8] H. M. ElKholly, "Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches," Master Thesis, Am. Univ. Cairo, 2014.
- [9] G. Szafranski and R. Czyba, "Different Approaches of PID Control UAV Type Quadrotor," in *Proc. Int. Micro Air Veh. Conf. summer Ed.*, 2011, pp. 70–75.
- [10] J. C. V. Junior, et al., "Stability Control of a Quad-rotor Using a PID Controller," *Brazilian Journal of Instrumentation and Control*, vol. 1, no. 1, p. 15, 2013..
- [11] A. J. Moshayedi, A. S. Roy, and L. Liao, "PID Tuning Method on AGV (Automated Guided Vehicle) Industrial Robot," *Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering*, vol. 12, no. 4, pp. 53–66, 2020.
- [12] A. Merheb and H. Noura, "Novel bioinspired stochastic tuning of a quadrotor PD controller," in *2012 2nd Australian Control Conference*, Sydney, NSW, 2012, pp. 227-232.
- [13] S. J. Haddadi, et al., "Attitude control and trajectory tracking of an autonomous miniature aerial vehicle," in *2015 AI & Robotics (IRANOPEN)*, pp. 1–6, 2015.
- [14] A. L. Salih, et al., "Modelling and PID controller design for a quadrotor unmanned air vehicle," in *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, 2010, pp. 1-5.
- [15] N. I. R. Pacheco, D. D. C. R. Resende, and P. A. A. Magalhaes, "Stability Control of an Autonomous Quadcopter through PID Control Law," *J. Eng. Res. Appl.*, vol. 5, no. 54, pp. 7-10, 2015.
- [16] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing Genetic Algorithms*. 2012.
- [17] S. E. Selvan, S. Subramanian, and S. T. Solomon, "Novel Technique for PID Tuning by Particle Swarm Optimization," in *Proc. 7th Annu. Swarm Users/Researchers Conf.(SwarmFest 2003)*, 2003, pp. 1–10.
- [18] C. Liu, C. Yan, and J. Wang, "Hybrid particle swarm optimization algorithm and its application in nuclear engineering," *Annals of Nuclear Energy*, vol. 64, pp. 276–286, 2014..
- [19] S. R. Barros dos Santos, S. N. Givigi, and C. L. N. Júnior, "An experimental validation of reinforcement learning applied to the position control of UAVs," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Seoul, 2012, pp. 2796-2802.
- [20] M. Mihoko and S. Eguchi, "Robust Blind Source Separation by Beta Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1859–1886, 2002.
- [21] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43.
- [22] L. Scrucca, "GA: A Package for Genetic Algorithms inR," *Journal of Statistical Software*, vol. 53, no. 4, pp. 1–37, 2013.
- [23] J. Hižak and R. Logožar, "An overview of the genetic algorithm and its use for finding extrema – with implementations in matlab," *Tehnički glasnik (Technical Review)*, vol. 10, no. 3-4, pp. 55-70, 2016.
- [24] S. Arora and S. Singh, "The Firefly Optimization Algorithm: Convergence Analysis and Parameter Selection," *International Journal of Computer Applications*, vol. 69, no. 3, pp. 48–52, 2013.

- [25] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," *Stochastic Algorithms: Foundations and Applications Lecture Notes in Computer Science*, pp. 169–178, 2009.
- [26] M. Vanin, "Modeling, identification and navigation of autonomous air vehicles," Magister Thesis, Dipartimento di Ingegneria Dell'informazione, Università degli Studi di Padova, 2013.
- [27] M. Enomoto and Y. Yamamoto, "Modelling, simulation and navigation experiments of Unmanned Aerial Vehicle," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, 2015, pp. 482–487.
- [28] A. J. Moshayedi and M. Gheibollahi, "The Quad rotor Dynamic Modeling and PID parameter optimization," Youtube, <https://www.youtube.com/watch?v=YMs6tw5-ow0>, 2019.
- [29] A. J. Moshayedi, *et al.*, "Path planning and trajectory tracking of a mobile robot using bio-inspired optimization algorithms and PID control," in *2019 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Tianjin, China, 2019, pp. 1–6.
- [30] A. J. Moshayedi and D. C. Gharpure, "Evaluation of bio inspired Mokhtar: Odor localization system," *2017 18th International Carpathian Control Conference (ICCC)*, Sinaia, 2017, pp. 527–532.
- [31] A. J. Moshayedi, L. Jinsong, and L. Liao, "AGV (automated guided vehicle) robot: Mission and obstacles in design and performance," *J. Simul. Anal. Nov. Technol. Mech. Eng.*, vol. 12, no. 4, pp. 5–18, 2019.

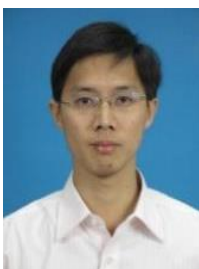
BIOGRAPHIES OF AUTHORS



Dr. Ata Jahangir Moshayedi is an associate professor at Jiangxi University of Science and Technology, China and Ph.D. in Electronic Science from Savitribai Phule Pune University, India. He has research interests in machine olfactory. He is also IEEE member, Instrument Society of India as a life member, lifetime member of Speed Society of India, member of editorial team of various conference and journals: *International Journal of Robotics and Control*, *JSME*, *Bulletin of Electrical Engineering and Informatics*, *International Journal of Physics and Robotics Applied Electronics*, etc. His 80 papers are published in national journals and conferences, 2 books published, 1 patent and 2 under review ones owned. His achievements include: Best Invention for "Design of ergonomic mattress recognizer system" in *7th Festival of Inventors Imitator Innovators Azad University*, 5–7 March, 2019 Najaf Abad branch, Iran; Best Robot Demo Presentation in *International Conference on Advanced Robotics and Intelligent Control (ICARIC 2018)* Jishou, Hunan, China, 11–14 Oct 2018; 3rd rank *IRANOPEN 2018*; *International Robotic Game UAV Outdoor*, Tehran, Iran, March 2018; Best paper presentation (2 papers) in *The Fifth International Conference of Electrical Engineering and Computer with Emphasis of Native Knowledge*, Alame Tabatabaee, Feb 2018; Selected Design, Pipe Inspection Robot, *ICROM 2017, The 5th International Conference on Robotics and Mechatronics*, Tehran, Iran November, 2017; *Avishkar 2012 OMC* (odor movie camera) selected for round III state level from Pune University, India, Avishkar 2009; Design and Development of Fuzzy Spin Coater (team working) selected for round III state level from Pune university India. His research interest includes robotics and automation/sensor modeling/bio inspired robot, mobile robot olfaction/plume tracking, embedded system/machine vision-based systems/virtual reality, machine vision/artificial intelligence.



Mahyar Gheibollahi was born in 1986 and has done his BE and ME in control engineering. He received his ME (Master of Engineering) in the control engineering form Azad University Komeinishahr Branch, Iran in 2018 and currently work as the technical supervisor in petroliam company. His research interest includes: Robotics and Automation, UAV modeling. Some of his achievements include 3rd rank *IRANOPEN 2018*; *International Robotic Game UAV Outdoor*, Tehran, Iran, March 2018.



Liao Liefu is a professor at Jiangxi University of Science and Technology, visiting scholar at Rochester University of Technology, PhD in Management Science and Engineering from Xi'an Jiaotong University, and key young and middle-aged teacher in Jiangxi Province. He won the second prize of the *14th Outstanding Social Science Achievement* in Jiangxi Province in 2011. Third prize of the *15th Provincial Outstanding Achievement in Social Science* of the Province. In 2015, it won the top ten teachers at Jiangxi University of Science and Technology. Published 50 papers, presided over 4 National Natural Science Foundations, "Research on the Relationship between Organizational Learning and Knowledge Level Based on Multi-Agent Simulation" won the *14th Second Prize of Outstanding Achievements in Social Science* in Jiangxi Province, "Knowledge management strategy and organization learning: An agent-based simulation" has won the third prize of *The 15th Outstanding Social Science Achievement* in Jiangxi Province.