



Airborne Computer System Based Collision-Free Flight Path Finding Strategy Design for Drone Model

Khulood Eskander Dagher^{1*} Mohammed Najm Abdullah²

¹*Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq*

²*Computer Engineering Department, University of Technology, Baghdad, Iraq*

* Corresponding author's Email: 120002@uotechnology.edu.iq

Abstract: An airborne computer system is an avionics-based model that consists of an electronic system and a specifically designed equipment. In this paper, we propose a collision-free path finding flight strategy that enhances the output performance of a drone system for free-navigation between buildings. This strategy is based on a reactive method that requires metaheuristic algorithms to process environmental information during drone flying. In addition, a robust controller is designed to achieve drone flight and reach the target point. The goal of this paper is to discuss the determination of the shortest flight path with the minimum cost function evaluation and this flight path should avoid the collision by using four heuristic algorithms including: Chaotic Particle Swarm Optimization (CPSO) algorithm, Fire-Fly (FF) algorithm, Bees (B) algorithm and a hybrid optimization algorithm that combines the CPSO, FF and B algorithms. To fly the drone and to follow the generated flightpath, six adaptive PID controllers are designed in order to control the highly nonlinear model and under-actuated drone system using an on-line CPSO algorithm to learn and tune the eighteen control gain parameters. The purpose of this control design is to precisely and quickly obtain robust thrust forces control actions to control the attitude and altitude of the drone model. The numerical results of the proposed flightpath-finding algorithms and robust control strategy confirm that the hybrid (CPSOFFB) flightpath algorithm has the minimum number of iterations and evaluation functions as well as free navigation and the shortest flight path length generated. Moreover, the proposed six adaptive PID controller results show that the four thrust forces control actions are smooth and accurately generated making the drone take off and follow the desired flight path quickly with the minimum number of cost function and around ± 10 cm minimum error tracking translation location. The maximum overshoot of the altitude did not exceed 10 cm in the transient state and the orientation error of the drone is approximately zero in the steady-state. To demonstrate the effectiveness of the proposed flight path finding and control strategy, a comparison is made with the results of other types of algorithms.

Keywords: Drone, Flight path finding algorithm, Adaptive PID controller, Chaotic particle swarm optimization algorithm, Firefly algorithm, Bees algorithm.

1. Introduction

The term “drone” could be defined as a pilotless aircraft, and so many authors use this term to refer to the category that contains many types of pilotless aircraft that are guided remotely or autonomously [1]. Several technical terms are also used when referring to “drones” such as unmanned aerial vehicles (UAVs), unmanned aircraft systems (UAS), remotely piloted vehicles (RPV), and remotely piloted aircrafts (RPA) [2]. Over the past years, many researchers have focused on drone-based airborne computing

systems that have been applied in many various fields such as air defence, precision agriculture, search and rescue operations as well as smart transport because of their outstanding advantages such as flexibility, light weight, strong mobility and good concealment [3, 4]. In addition, the drone quadcopter is useful in different fields such as monitoring systems and inspection systems in dangerous areas. These drone systems have different sizes and shapes [5].

In general, there are two problems that should be solved for the drone path tracking system, the first one is the path planning for the free flying of the

drone, which is especially important as the core technology of drones, and it can be divided into two types: global path planning and local path planning [6]. The second problem is the controller design for the drone. In this regard, many attempts have been reported in the literature for various types of controllers that depend on different types of mathematical models and tuning control methodologies to control the attitude and altitude of drones in order to solve the problem of the drone stabilization in a desired location and orientation during flying [7]. Therefore, some of the researchers have focused on flight path planning. For example, the authors in [8] developed a trajectory generation methodology that allows planning from any initial point to a target point based on feasibility under the dynamic states of the drone accelerations in each axis. The PSO algorithm was utilized in [9] for obtaining the optimal path planning of the drone during flying in an on-line manner. Moreover, a new path-planning algorithm was proposed in [10] based on the Fast Geometric Avoidance algorithm that combines the geometric avoidance of obstacles and the selection of a critical avoidance start time based on three variables including kinematics considerations, collision likelihood and navigation constraints. In addition, an on-line collision-free path generation and navigation system for drones was proposed in [11] based on geographical locations of the drone system. In [3], the authors presented a cellular communication-based connection for an autonomous drone with a LiDAR sensor and Hector SLAM for obtaining the best flight path of the drone in an indoor environment. The A-star algorithm was adopted in [6] to generate an optimal flight path in global environment with dynamic path planning of the drone. In [4], real-time autonomous obstacle avoidance for drones in a 3D space was demonstrated based on a circular arc trajectory for achieving free-navigation flight. The morphing algorithm was proposed in [12] to obtain an optimal off-line trajectory for aircrafts.

On the other hand, some researchers have focused on control design based on the mathematical model of the physical nonlinear drone system such as: The fixed PID control parameters based on the Ziegler-Nichols method, which was explained in [13] for obtaining stability in flying the quad-rotor by generating four thrust forces. In [14], an adaptive Lyapunov gain scheduled PID controller was designed for the model of the vertical flying drone quadrotor type with conventional PID and adaptive PID controllers. In another work [15], a PID controller with a nature-inspired off-line algorithm was proposed to achieve good quadrotor performance for tracking the desired flight path. In [16], the hybrid

neuro-fuzzy inference intelligent controller was proposed for the trajectory tracking of an unmanned aerial vehicle. In addition, the authors in [17] presented the decoupling fuzzy controller based on the Newton-Euler method of the mathematical kinematics and dynamics quadrotor UAV model to control the desired position of the quadrotor system. The fuzzy-PID controller was applied to a dynamic quad-rotor UAV model to achieve a robust attitude flight [18]. In another work [19], a multi-input-multi-output PID controller was proposed using linear quadratic regulator-Gaussian for attitude and altitude control of the quadrotor. Furthermore, in [20] the authors discussed the auto-tuner of a cascaded PID controller for a quadrotor UAV model based on the desired gain and phase margins to determine the response speed. In addition, an intelligent adaptive type-2 fuzzy sliding mode control with PID sliding surface was suggested in [21] for controlling the 6 DOF quadrotor aircraft. In [22], the researchers investigated the robustness and the effectiveness of the optimal model free back-stepping controller with a cuckoo search algorithm for a quadrotor helicopter to achieve the global asymptotical stability of the closed loop control quadrotor helicopter based on the Lyapunov method. Moreover, in [23] the model predictive of a nonlinear controller for UAV was proposed for tracking the desired trajectory and the constrained control of an UAV quadrotor system. In [5], a modified adaptive sliding mode control for trajectory tracking of mini-drone was presented for enhancing the performance of the mini-drone motion control. In addition, in [7] a second-order sliding mode controller with PSO tuning control algorithm was presented for achieving the desired altitude and attitude of the drone.

Therefore, the motivation of this work is taken from [4-6, 11, 24] to solve two problems: the first one is to find the shortest smooth flight path of the drone between the starting point and the target point with collision-free navigation. The second problem is to solve the decoupling between the inputs and the outputs states of the drone, which has a highly nonlinear dynamic behaviour with an under-actuated system.

The main contribution points of this paper are: the shortest and the smoothest flight path equation is generated using numerical simulation based on a proposed hybrid heuristic algorithm with high efficiency during generating the flight path of the drone. Moreover, the best value of the four thrust forces control actions are generated using numerical simulation based on the proposed six adaptive PID controllers with an on-line CPSO tuning algorithm, which is used to stabilize the drone model and to

follow and track the desired flight path equation with minimum tracking location and orientation errors by controlling the altitude and attitude of the drone.

This paper is organized as follows: Section 2 explains the mathematical drone dynamics model. Section 3 illustrates the proposed collision-free flight path finding strategy. Section 4 discusses the simulation results and shows the effectiveness and performance of the proposed strategy. Finally, the conclusions for this research are given in Section 5.

2. Drone dynamics state model

In general, a drone helicopter uses four symmetrically installed rotors that generate four thrust forces for flight. Hence, the drone consists of four rotors that have the same structure and radius and it uses two arms on a plane. The four rotors are symmetrically located on the four edges of a cross formed at the same height, as shown in Fig 1, with two coordinates. The first one is B which denotes the drone body coordinate and G which denotes the ground coordinate. Therefore, the altitude (x , y and z) of the drone motion relates the center of mass movement while the attitude (φ , θ , and ψ) represents movement around the center of drone's mass of the rotation. So, the drone model can be described in 6 DOF in the space.

These four rotors are rotated as follows: two rotors have to rotate in the anticlockwise direction (rotor 1 and rotor 3) but (rotor 2 and rotor 4) have to rotate in the clockwise direction. All external equipment of the drone such as flight controller device and battery are setup in the middle of the drone body. The drone flight is achieved by controlling the attitude and position of the drone helicopter by changing four rotors velocities to change the four thrust forces as follows:

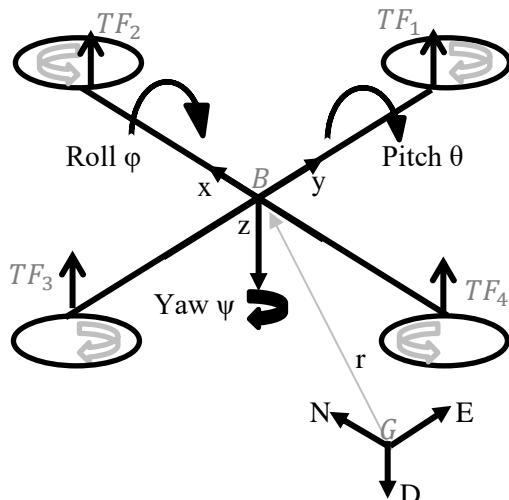


Figure. 1 The structure of the drone model

When varying four thrust forces, the lift of the drone along the z -axis is obtained.

When varying the thrust force of (TF_1) and (TF_3), the drone is moved along the x -axis and this leads to changing the pitch angle (θ) of the drone.

When varying the thrust force of (TF_2) and (TF_4), the drone is moved along the y -axis and this leads to changing the roll angle (φ) of the drone.

When the thrust force of (TF_1) and (TF_3) are increased while the thrust force of (TF_2) and (TF_4) are decreased, the aerodynamic torque effect generates a yawing moment (ψ) that makes the drone rotate around the z -axis. The state equation with the consideration of air drag of the dynamic drone model is taken from [5] as follows:

$$\ddot{x} = (C\varphi S\theta C\psi + S\varphi S\psi)U_1 \quad (1)$$

$$\ddot{y} = (C\varphi S\theta C\psi - S\varphi C\psi)U_1 \quad (2)$$

$$\ddot{z} = (C\varphi C\theta)U_1 - g \quad (3)$$

$$\ddot{\varphi} = [(I_y - I_z)\dot{\theta}\dot{\psi} - (J_r\Omega_r)\dot{\theta} + U_2] \frac{1}{I_x} \quad (4)$$

$$\ddot{\theta} = [(I_z - I_x)\dot{\varphi}\dot{\psi} + (J_r\Omega_r)\dot{\varphi} + U_3] \frac{1}{I_y} \quad (5)$$

$$\ddot{\psi} = [(I_x - I_y)\dot{\theta}\dot{\varphi} + U_4] \frac{1}{I_z} \quad (6)$$

$$\Omega_r = (w_1 - w_2 + w_3 - w_4) \quad (7)$$

$$U_1 = (TF_1 + TF_2 + TF_3 + TF_4) \frac{b}{M} \quad (8)$$

$$U_2 = l(-TF_1 - TF_2 + TF_3 + TF_4) \frac{b}{I_x} \quad (9)$$

$$U_3 = l(-TF_1 + TF_2 + TF_3 - TF_4) \frac{b}{I_y} \quad (10)$$

$$U_4 = d(TF_1 + TF_2 + TF_3 + TF_4) \frac{1}{I_z} \quad (11)$$

Where, $(\ddot{x}, \ddot{y}, \ddot{z})$ denote the acceleration of the drone in the inertial frame. $(\ddot{\theta}, \ddot{\varphi}, \ddot{\psi})$ and $(\dot{\theta}, \dot{\varphi}, \dot{\psi})$ denote the angular acceleration and angular velocity of the Euler angles (roll, pitch, and yaw), respectively. C and S represent Cos and Sin, respectively. $TF_{1,2,3,4}$ denote the thrust forces that are generated by the four rotors and can be considered as control signals to the drone system.

The definitions of physical drone parameters of Eqs. (1) to (11) are taken from [15, 24], as shown in Table 1.

Table 1. Drone physical parameters with values and units [15, 24]

Name	Parameter	Value	Unit
Quadcopter Mass	M	0.65	kg
Quadcopter Arm Length	l	0.23	m
Thrust Coefficient	b	3.13×10^{-5}	N.sec ²
Drag Coefficient	d	7.5×10^{-7}	N.m.sec ²
Inertia on x-axis	I_x	7.5×10^{-3}	kg m ²
Inertia on y-axis	I_y	7.5×10^{-3}	kg m ²
Inertia on z-axis	I_z	1.3×10^{-2}	kg m ²
Rotor Inertia	J_r	6×10^{-5}	kg m ²
Gravity Constant	g	9.81	m/sec ²
Rotational Velocities	Ω	-	rad/sec

3. Collision-free flight path finding strategy

In this work, the proposed collision-free flight path finding strategy design consists of two stages, as shown in Fig. 2. The first stage is the high level that has four steps for achieving the desired flight path equation for the drone from the starting point to the target point:

- Environmental information and modelling.
- Object detection.
- Optimal flight path finding.
- Flight path equation generation.

The second stage is the low level that has two steps to control the thrust forces of the drone during flying:

- Six adaptive PID controllers.
- An on-line learning and tuning algorithm for the control gains.

3.1 Flight-path planning algorithms

In order to fly a drone and follow the desired flight path, we need the basic requirement about the environmental information and modelling as a buildings' map. So in this work, the drone's workspace is represented as a 3D Cartesian coordinates (x, y, z). Thus, we need to determine the position of the object, which means the target point, and we need to determine the minimum distance between the starting point and the target point of the drone position by using the distance objective function as in Eq. (12) [4, 8].

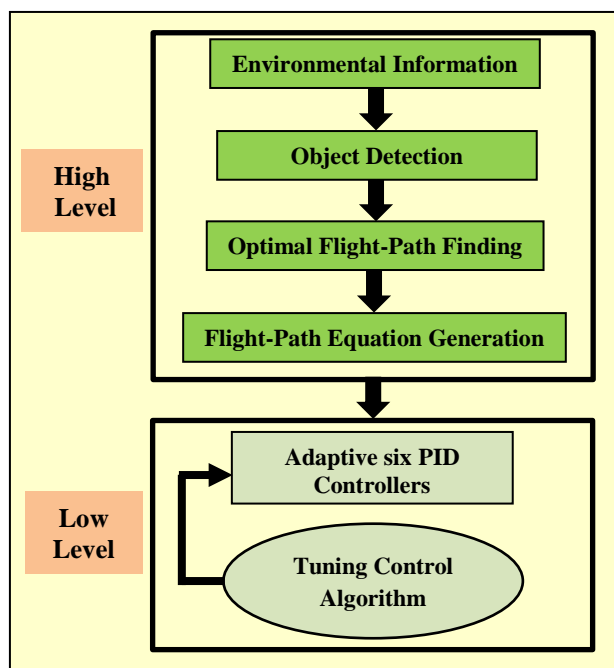


Figure. 2 Collision-free flight-path finding strategy diagram

$$Dist_{Obj} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (12)$$

Where,

$Dist_{Obj}$ is the distance between two points.

x_i and y_i are x and y coordinates of the current waypoints.

x_{i+1} and y_{i+1} are x and y coordinates of feasible waypoints in the iteration i+1.

Then, to fly the drone between these two points successfully, two problems that the drone will encounter must be solved. The first problem is that the flight path must avoid collision with building obstacles, and the second problem is saving the battery energy by reducing the length of the flight path to a minimum. In this work, these issues can be fixed using the proposed collision-free flight path finding strategy methods that use four heuristic optimization algorithms; Chaotic Particle Swarm Optimization algorithm, Fire-Fly algorithm, Bees algorithm and the proposed hybrid flight-path optimization algorithm that is based on CPSO, FF, and B algorithms. The first flight path-finding algorithm uses CPSO algorithm, in which the chaos method is incorporated into the PSO algorithm to generate more randomness in the search for the PSO algorithm and to solve the local minima problem [25]. Therefore, the particle updates its velocity and position equations as can be expressed below [25, 26]:

$$\beta^{k+1} = \mu \times \beta^k (1 - \beta^b) \quad 0 < \beta^1 < 1 \quad (13)$$

$$W = W_{max} - [(W_{max} - W_{min}) \times (\frac{iter}{iter_{max}})] \quad (14)$$

$$W_{new} = W \times \beta^{k+1} \quad (15)$$

$$v_i^{k+1} = W_{new} \times v_i^k + c_1 r (pbest_i^k - xy_i^k) + c_2 r (gbest^k - xy_i^k) \quad (16)$$

$$xy_i^{k+1} = xy_i^k + v_i^{k+1} \quad (17)$$

Where i^{th} denotes the particle numbers and k^{th} denotes the iteration number.

Table 2 illustrates the CPSO parameters' definitions that are used in the numerical simulation. The second flight path-finding algorithm uses the firefly algorithm. Therefore, the firefly algorithm formulation is based on three ideal rules as follows: the equation of the firefly movement is given in Eq. (18) [27, 28] and it represents the movement of a firefly i to another, more attractive firefly j , Eq. (19) [27, 28] describes a firefly's attractiveness, and Eq. (20) [27, 28] calculates the distance between firefly i and firefly j .

$$xy_i^{k+1} = xy_i^k + \beta_o \exp^{-\gamma r_{ij}} (xy_i^k - xy_j^k) + \alpha(rand - 0.5) \quad (18)$$

$$\beta = \beta_o \exp^{-\gamma r_{ij}} \quad (19)$$

$$r_{ij}^k = \sqrt{\sum_{t=1}^d (xy_{i,t}^k - xy_{j,t}^k)^2} \quad (20)$$

Table 2. The definitions of CPSO parameters

Parameter	Definition with value
β^0	The deterministic value is equal to 0.3
μ	The parameter value is equal to 4
W	Inertia weight is equal to 0.888
W_{min}	Minimum weight is equal to 0.3
W_{max}	Maximum weight is equal to 0.9
$iter$	Current iteration number (k)
$iter_{max}$	Maximum number of iterations
c_1, c_2	Coefficients of acceleration [1.125, 1.125]
v_i^k	The i^{th} particle's velocity in the k^{th} iteration.
$(xy)_i^k$	The i^{th} particle's position in the k^{th} iteration.
$pbest_i$	The best local fitness values for the i^{th} particle.
Gbest	The best global fitness value for all particles.

Table 3. The definitions of FF parameters

Parameter	Definition with value
β_o	The maximum attractiveness value is equal to 1.
γ	The absorption coefficient with a range of [0.1 to 10].
r_{ij}	The distance between firefly i and firefly j .
d	The number of dimensions, which is equal to 5.
α	A random value [0 to 1].
$xy_{i,t}$	The t^{th} spatial coordinate element xy_i of i^{th} firefly.
xy_i^k	The i^{th} coordinate element of a firefly in the k^{th} iteration.

Table 3 illustrates the FF parameters' definitions that are used in the numerical simulation.

The third flight path-finding algorithm uses the nature-inspired Bee algorithm that can solve the problem of getting stuck at the local optimal solution during generating the flight path. This algorithm has two search directions, the first is the neighbourhood search that consists of two types of Bees (Selected Bees and Recruited Bees). The second search is called the global search that consists of two types of Bees (Scout Bees and Fittest Bees) [29]. To carry out the Bee algorithm, the first step requires generating Scout Bees (SB) as the initialize population with random positions $xy_{Scout\ Bees}$ as solutions in the global search. Then, the distance objective function is calculated for the population. Hence, for the neighbourhood search, we will select (SLB) as the best points $xy_{Selected\ Bees}$ that have the minimum value of the distance objective function. Then, we apply the proposed patch size as in Eq. (21) for obtaining (RB) Recruited Bees $xy_{Recruit\ Bees}$.

$$xy_{Recruit\ Bees} = xy_{Selected\ Bees} + 0.1 \times xy_{Selected\ Bees} \times random(0,1) \quad (21)$$

After that, we will evaluate the distance objective function for each $xy_{Recruit\ Bees}$ position. Then, the (FB) Fittest Bees are selected from each patch having a minimum distance value. Therefore, the remaining Bees (SB-FB) are randomly generated in the global search. Table 4 illustrates the Bees algorithm

Table 4. The definitions of Bees algorithm parameters

Parameter	Definition
SB	Scout Bees in the global search
SLB	Selected Bees in the global search
RB	Recruited Bees in the local search
FB	The fittest Bees in the local search
SB-FB	The remaining Bees in the global search

parameters' definitions that are used in the numerical simulation.

In this work, we proposed a hybrid algorithm from the algorithms that were mentioned above. This hybrid algorithm consists of the Bees and CPSO-FF algorithms because the global best solution of the population does not need to be directly used in the Bee algorithm for finding new positions. Therefore, the Bee algorithm is used with the CPSO-FF algorithm in order to solve the local minima problem, in which the particles (position x,y) in the CPSO-FF might get stuck in the local minima. To solve this problem, the particles can escape from the local minima using random search as scout bees in the Bee algorithm to achieve high efficiency during the generation of the flight path. Therefore, the proposed optimization method can overcome the disadvantages and at the same time benefit from the advantages of the individual algorithms in terms of increasing the speed of learning, avoiding falling in local minima and reducing the number of fitness evaluation. In particular, we proposed a hybrid optimization algorithm that combines the Bee algorithm, the CPSO, and the Firefly searching mechanism and this algorithm is called the (B-CPSO-FF) global-local optimization approach. In this hybrid algorithm, we developed the velocity update equation of the PSO algorithm that replaced the random value (r_1, r_2) by the firefly's attractiveness equation and converted the movement of the firefly equation by the equation of the distance forms as shown in Eqs. (22) and (23).

$$r_{Pbest}^k = \sqrt{\sum_{t=1}^d (pbest_{i,t}^k - xy_{i,t}^k)^2} \quad (22)$$

$$r_{Gbest}^k = \sqrt{\sum_{t=1}^d (gbest_t^k - xy_{i,t}^k)^2} \quad (23)$$

Where r_{Pbest}^k is the distance between the best local fitness values for the i^{th} particle's position in the k^{th} iteration. r_{Gbest}^k is the distance between the best global fitness values for all particles and the i^{th} particle's position in the k^{th} iteration.

Therefore, the final particle updates its position's equation as can be expressed in Eq. (24):

$$xy_i^{k+1} = W_{new} \times xy_i^k + c_1 Exp^{-r_{Pbest}^k} (pbest_i^k - xy_i^k) + c_2 Exp^{-r_{Gbest}^k} (gbest^k - xy_i^k) \quad (24)$$

The flowchart of the proposed (B-CPSO-FF) algorithm is shown in Fig. 3.

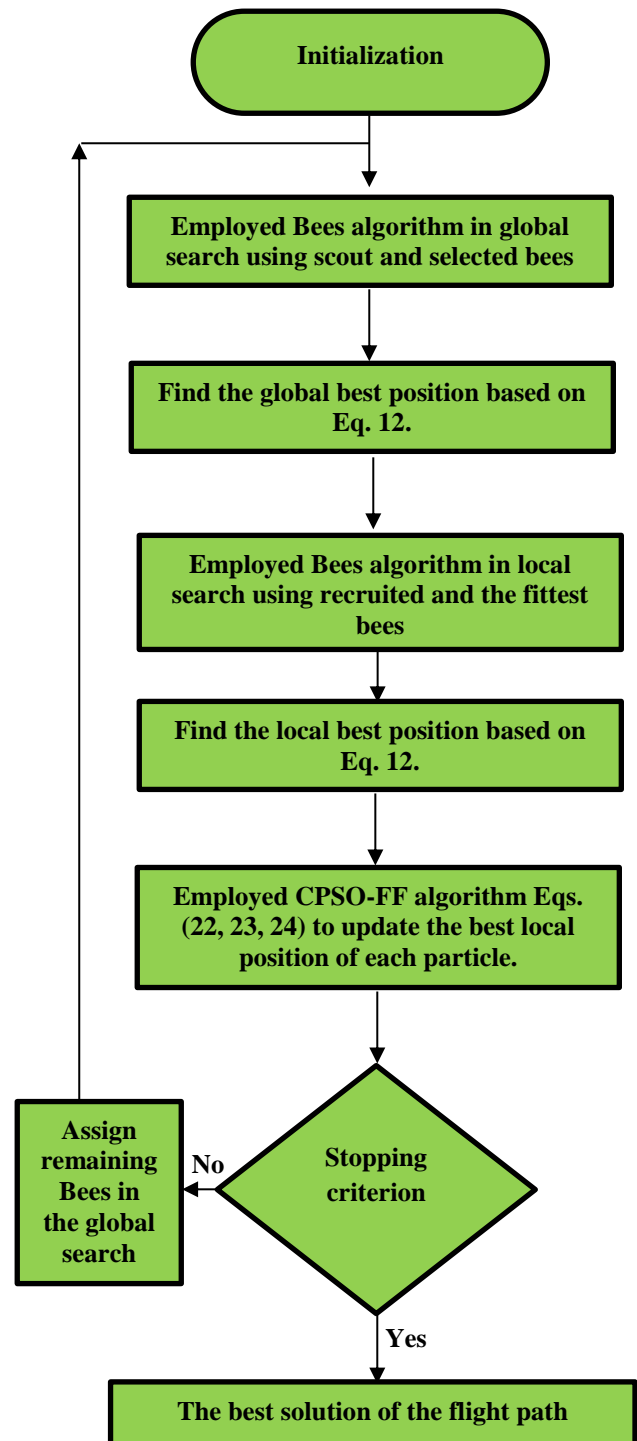


Figure. 3 A flowchart of the proposed (B-CPSO-FF) algorithm

3.2 Flight-path control algorithm

The flight-path control algorithm is based on six adaptive PID controllers because the drone model has 6-DOF, which means there are six output states ($x, y, z, \theta, \phi,$ and ψ) to be controlled. Therefore, we need six PID controllers to generate precisely and quickly the optimal four thrust forces action in order to on-

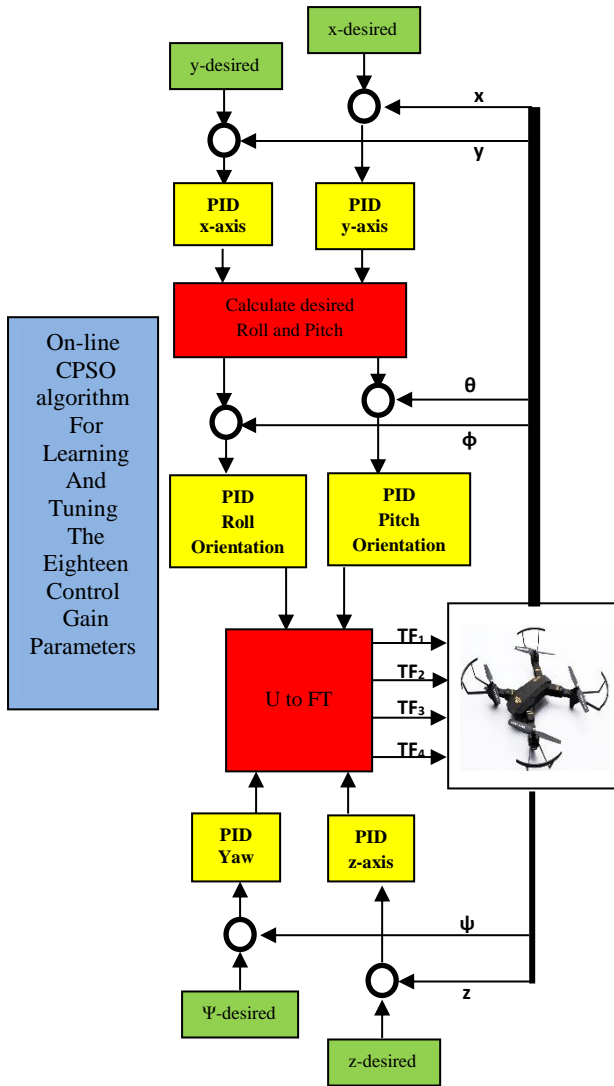


Figure. 4 An online adaptive flight-path control algorithm

line track the desired flight path for the drone as well as to stabilize the drone flying. The structure of the six adaptive PID controllers, which achieve the attitude and altitude control of the drone, is shown in Fig. 4.

The general control law of the six discrete PID controllers' equation can be expressed as in Eq. 25 [30].

$$TF - PID_{\rho}(t + 1) = TF - PID_{\rho}(t) + kp_{\rho}(E_{\rho}(t) + ki_{\rho}(E_{\rho}(t) + E_{\rho}(t - 1)) + kd_{\rho}(E_{\rho}(t) - E_{\rho}(t - 1))) \quad (25)$$

Where ρ denotes the different output states of the drone model (x, y, z) and (φ, θ, ψ) . k_p , k_i and k_d are the proportional gain, the integral gain, and the derivative gain, respectively. $TF - PID_{\rho}(t)$ denotes the thrust force control action and $e(t)$ denotes the error value between the desired states and the actual output states.

These six adaptive PID controllers have eighteen parameters (kp_{ρ}, ki_{ρ} and kd_{ρ}). Hence, we will use the CPSO algorithm for finding and tuning the control gains parameters based on Equations (13, 14, and 15) and the particles update the velocity and positioning two equations, namely Eqs. (26) and (27) [31].

$$v_i^{k+1} = W_{new} \times v_i^k + c_1 r (pbest_i^k - (kp, ki, kd)_{\rho,i}^k) + c_2 r (gbest^k - (kp, ki, kd)_{\rho,i}^k) \quad (26)$$

$$(kp, ki, kd)_{\rho,i}^{k+1} = (kp, ki, kd)_{\rho,i}^k + v_i^{k+1} \quad (27)$$

The desired Roll and Pitch orientations are proposed in Eqs. (28) and (29) that are based on the output of the PID x-axis and the output of the PID y-axis.

$$\varphi_d = \frac{1}{u_1} (u_x \sin(\psi) - u_y \cos(\psi)) \quad (28)$$

$$\theta_d = \frac{1}{u_1} (u_x \cos(\psi) + u_y \sin(\psi)) \quad (29)$$

4. Simulation results

The MATLAB package is used to execute the proposed collision-free flight path-finding strategy design as shown in Fig. 2, to investigate the efficiency of the proposed algorithm at building environment flight-path tracking, and to confirm that the drone model in both desired location and rotation are stabilized. In particular, we will use two case studies as follows:

Case I

The initial position of the drone model is $(x_o, y_o, z_o) = (10, 15, 0)$ m, respectively and the object detection location as the target position is $(x_t, y_t, z_t) = (250, 150, 5)$ m, respectively. We will use three heuristic algorithms (CPSO, FF, and Bee) as well as the proposed hybrid (B-CPSO-FF) algorithm in order to find the best flight-path between the starting point and the target point as shown in Fig.5 a and b that demonstrates the 3D and 2D blinding environment simulation of the flight path for the obstacle avoidance problem of the drone model. The proposed values of the parameters in each algorithm are presented in Table 5.

The best cost function that represents the minimum distance value between the starting point and the target point is obtained to be 310 meters when we applied the proposed hybrid algorithm based on Eq. (12). Figure 6 illustrates the different values of the cost function for the different types of algorithms

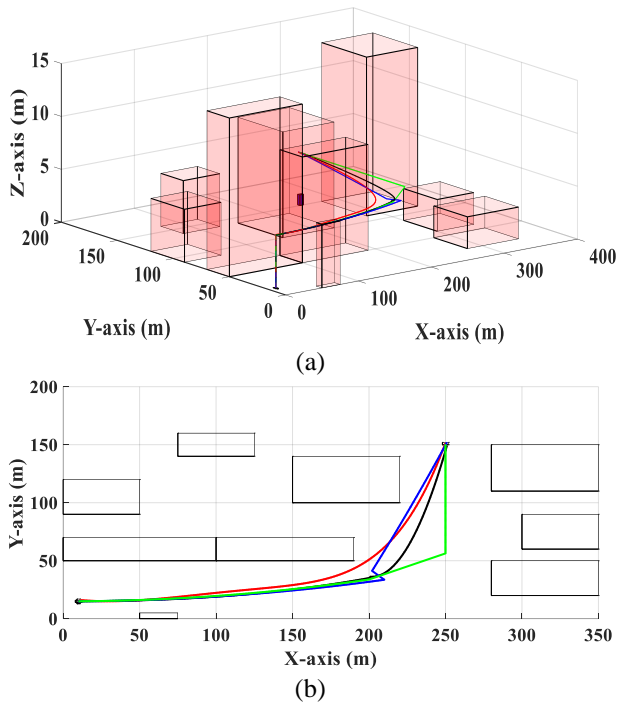


Figure 5 The drone obstacle avoidance problem in: (a) 3D blinding environment; (b) 2D blinding environment

Table 5. The best values of the parameters of CPSO, FF, and Bees algorithms

Type of Algorithm	Parameter	Value
CPSO	β^0	0.3
	μ	4
	W	0.888
	W_{min}	0.3
	W_{max}	0.9
	$iter_{max}$	50
	c_1, c_2	[1.125, 1.125]
	Number of Particles	50
FF	Particle's weights	18
	β_o	1
	γ	[1 to 10]
	d	5
	α	[0 to 1]
	Number of Iterations	50
Bees	Number of Firefly	50
	Firefly's weights	18
	SB	20
	SLB	5
	RB	50
	FB	10
Bees	SB-FB	10
	Number of Iterations	50

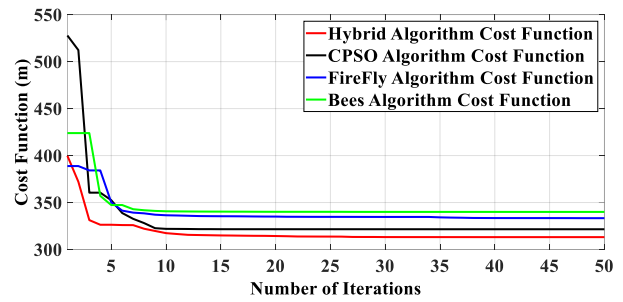


Figure. 6 The cost function response

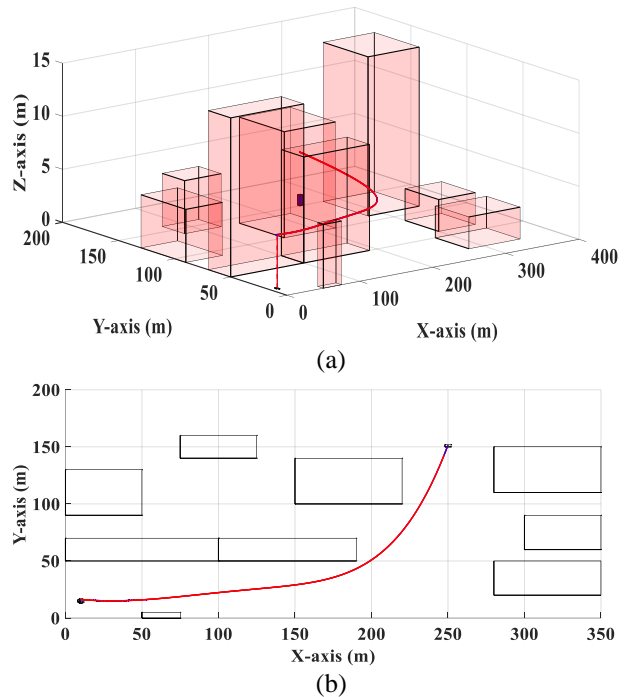


Figure. 7 The drone obstacle avoidance problem based on the hybrid B-CPSO-FF algorithm in: (a) 3D blinding environment and (b) 2D blinding environment

with a maximum number of iterations of 50. The high efficiency of the hybrid algorithm can be demonstrated during generating the flight path in terms of the short path and saving the battery energy of the drone during flying.

Hence, the best flight path of the drone model in the building map environment is obtained by the proposed hybrid algorithm due to its ability to find the minimum flight distance as well as to avoid getting stuck in local minima, which leads to obtaining the fastest and the best global solution. Fig. 7 shows the 3D and 2D blinding environment simulation of the flight path for the drone model based on the hybrid B-CPSO-FF flight path planning algorithm.

Based on the fitting function with Spline interpolation, we obtained the desired flight-path equations for three flying regions as in Eqs. (30) to (32), respectively, based on the proposed hybrid B-CPSO-FF algorithm.

$$\text{Region I} \begin{cases} X_d = 10 \\ Y_d = 15 \\ Z_d = 0.5(t) \end{cases} \quad (30)$$

$$\text{Region II} \begin{cases} X_d = t \\ Y_d = 10^{-6}X_d^3 + 5 \times 10^{-4} X_d^2 + 217 \times 10^{-6}X_d^1 + 15 \\ Z_d = 5 \end{cases} \quad (31)$$

$$\text{Region III} \begin{cases} X_d = t \\ Y_d = 116 \times 10^{-4} X_d^2 - 3.123 \times X_d^1 + 203.89 \\ Z_d = 5 \end{cases} \quad (32)$$

To achieve free-flight of the drone, which is represented by an under-actuated system with four thrust inputs and six state outputs with highly coupled nonlinear dynamic behaviour and to follow the desired flight-path equation.

We applied the on-line CPSO tuning algorithm for the six PID controllers by carrying out the 4th order Range Kutta method and solving Eqs. (1) to (11) with 0.01 second sampling time.

Table 6 presents the proposed search area of on-line control parameters values based on CPSO algorithm used in the six PID controllers in order to stabilize the drone model flying in the transient response and to follow the desired flight-path equation in the closed loop control system.

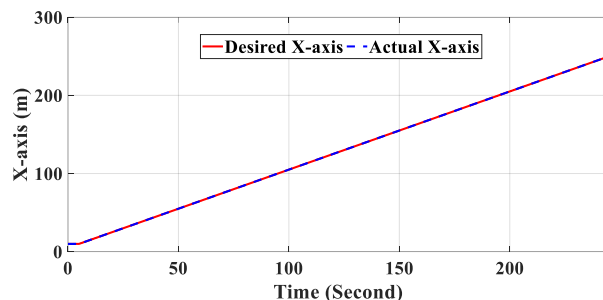
Fig. 8 (a), (b), and (c) shows the desired flight-path with translation coordinates (x, y, and z), respectively.

The 2D and 3D simulation flight-path control tracking responses for the drone model are shown in Fig. 9 (a) and (b), respectively, showing a fast flying with very small oscillation during tracking the desired flight-path equations.

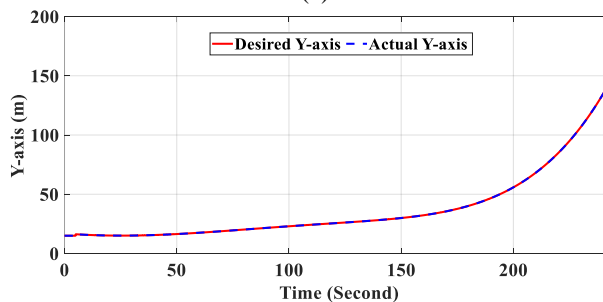
Fig. 10 demonstrates the small translation error of coordinates, and it did not exceed ±10 cm. The overshoot of the altitude is less than 11 cm only during the take-off of the drone.

Table 6. The search area of on-line CPSO control parameters values

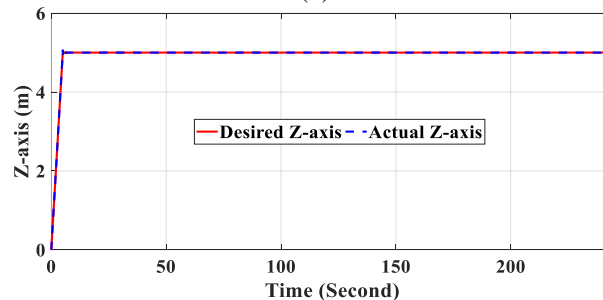
Controller Types	Range of k_p Values	Range of k_i Values	Range of k_a Values
PID-x-axis	[10 to 0.1]	[3 to 0.01]	[4 to 0.15]
PID-y-axis			
PID-z-axis	[8 to 4.5]	[2 to 1.5]	[1 to 0.33]
PID-Roll Orientation	[2 to 1.15]	[2 to 0.226]	[2 to 1.11]
PID-Pitch Orientation			
PID-Yaw Orientation	[0.145 to 0.123]	[0.22 to 0.113]	[0.23 to 0.117]



(a)



(b)



(c)

Figure. 8 The desired flight-path with translation coordinates (x, y, z)

The attitude response of the drone model during tracking the desired flight-path is shown in Fig. 11, which shows fast and smooth behaviours. Fig. 12 illustrates the response of the thrust force control action of each rotor in the drone model. These control actions have a fast response with small oscillation, with no saturation state, and with no spike actions due to the effectiveness of the on-line CPSO tuning control gains, which is used to tune the six adaptive PID controllers. This algorithm generated smooth and stable values for the parameters of the six PID controllers.

Moreover, these control actions have the ability to track smoothly the desired flight-path location and rotation of the drone model. Figure 13 shows the response of the drone speed in translation coordinate (x, y, and z) during the flight represented by the desired path equation.

Case II

To show the effectiveness of the proposed flight-path control system design, the second initial position

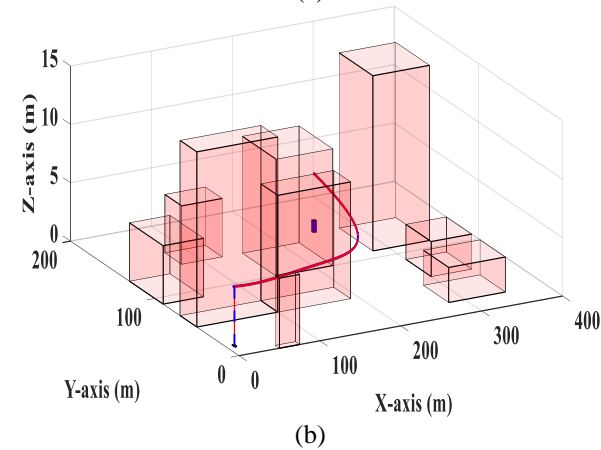
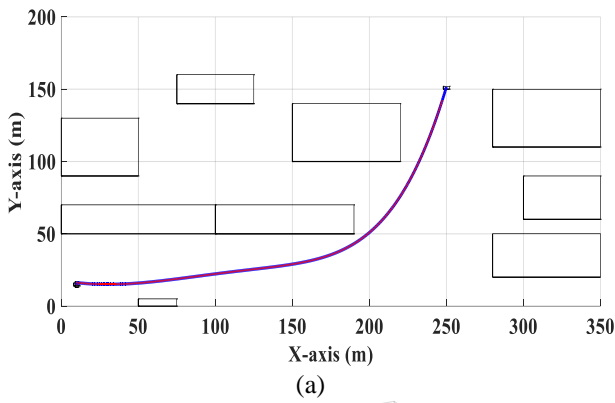


Figure. 9 The flight-path numerical simulation tracking for the drone model: (a) 2D and (b) 3D

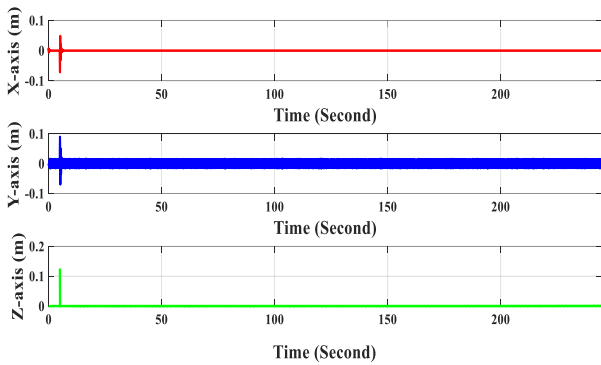


Figure. 10 The (x, y, z) axes translation error coordinates.

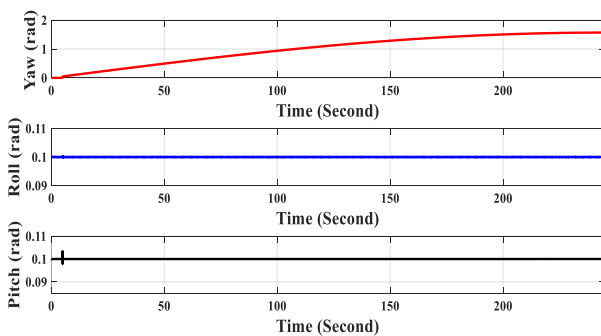


Figure. 11 The desired orientation (yaw, Roll, and Pitch) coordination

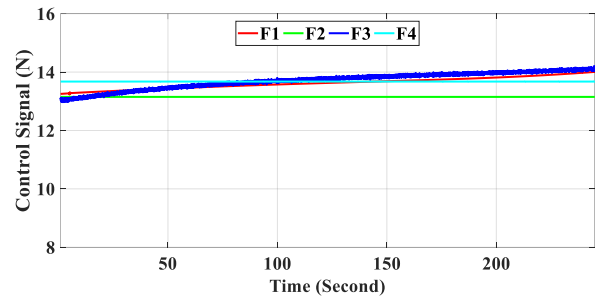


Figure. 12 The thrust force response of each rotor in the drone model

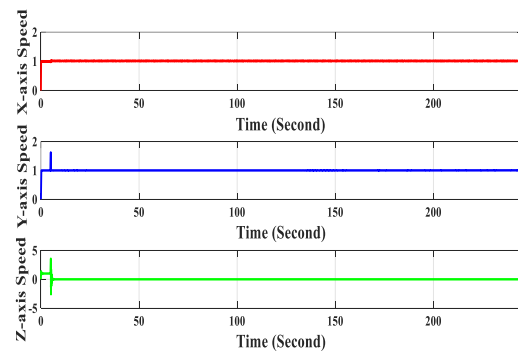


Figure. 13 The drone speed in (x, y, and z) axes frame

of the drone model is taken as $(x_o, y_o, z_o) = (25, 148, 0)$ m, respectively and the second target position is taken as $(x_t, y_t, z_t) = (340, 100, 5)$ m respectively. We used four heuristic algorithms (CPSO, FF, Bee, and the hybrid B-CPSO-FF) in order to find the best flight-path between the starting point and the target point as shown in Fig.14 a and b that demonstrates the 3D and 2D blinding environment simulation of the flight-path for the drone model.

Based on Eq. (12), the minimum value of the distance cost function between the starting point and the target point is obtained to be 337 meters when we applied the proposed high-efficiency hybrid B-CPSO-FF algorithm during generating the flight path. Fig. 15 illustrates the different values of the distance cost function for the different types of algorithms that are used during 50 iterations.

Therefore, the best flight path of the drone model in the building map environment based on the cost function equation was obtained by the proposed hybrid algorithm due to its ability to find the fastest and the best global solution and to avoid getting stuck in the local minima. Figure 16-a and b show the 3D and 2D blinding environment simulation of the flight path for the drone model, respectively.

$$\text{Region I} \left\{ \begin{array}{l} X_d = 25 \\ Y_d = 148 \\ Z_d = 0.5(t) \end{array} \right\} \quad (33)$$

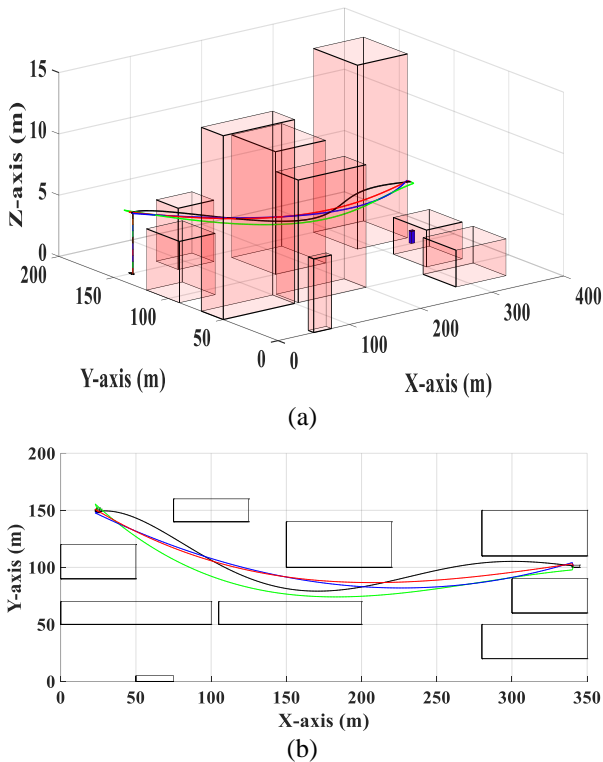


Figure. 14 The drone obstacle avoidance problem in: (a) 3D blinding environment and (b) 2D blinding environment

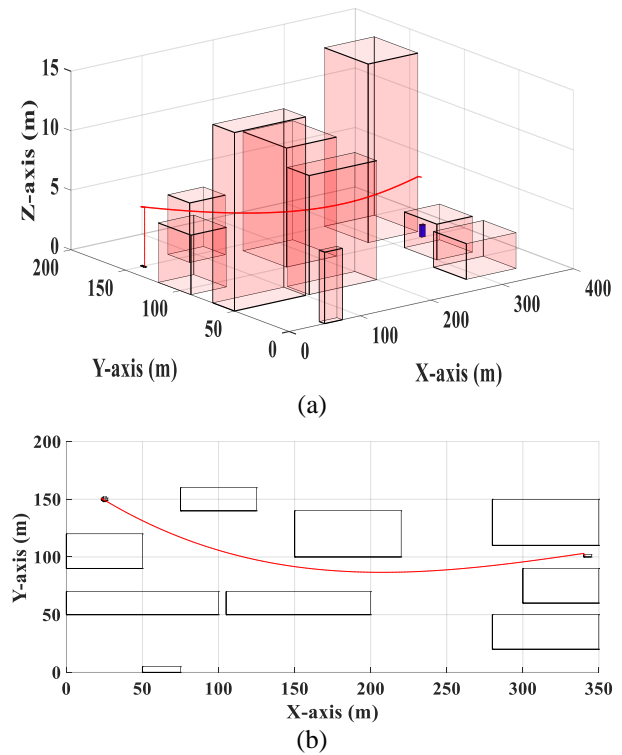


Figure. 16 The drone obstacle avoidance problem based on the hybrid B-CPSO-FF algorithm in: (a) 3D blinding environment and (b) 2D blinding environment

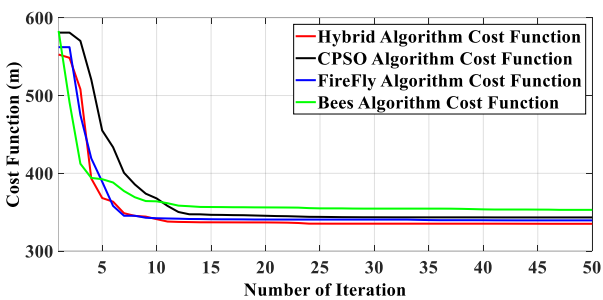


Figure. 15 The cost function response

To generate the desired flight-path equations as in Eqs. (33) and (34) for two flight regions with the shortest flight distance using the hybrid algorithm, we used a fitting function with Spline interpolation to generate these equations, as follows:

$$\text{Region II} \left\{ \begin{array}{l} X_d = t \\ Y_d = -3 \times 10^{-6} X_d^3 + 32 \times 10^{-4} X_d^2 \\ \quad - 0.9405 \times X_d + 170.84 \\ Z_d = 5 \end{array} \right. \quad (34)$$

After that, to achieve the second free-flight path of the drone system, we applied the on-line CPSO tuning control gains algorithm for the six adaptive PID controllers in order to follow the desired flight path equation. Fig. 17 (a), (b), and (c) show the

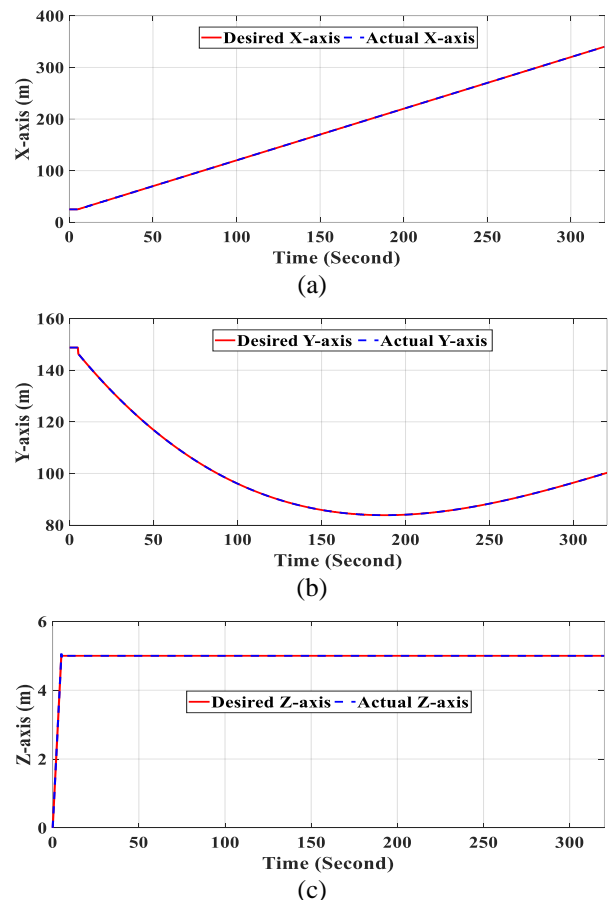


Figure. 17 The desired flight-path with translation coordinates (x, y, z)

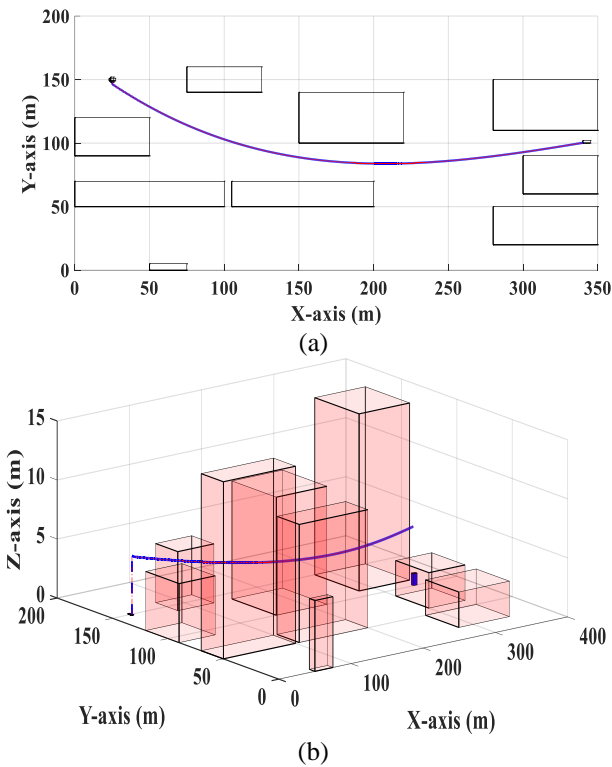


Figure. 18 The flight-path numerical simulation tracking for the drone model: (a) 2D and (b) 3D

desired flight-path with translation coordinates in (x, y, and z), respectively.

The 2D and 3D simulation of the flight-path tracking responses for the drone model are shown in Fig. 18 (a) and (b), respectively, showing fast flying with very small oscillation during tracking the desired flight path equations.

Fig. 19 demonstrates the small translation error coordinates and it did not exceed ± 10 cm. The overshoot of the altitude is less than 10 cm. The attitude of (Yaw, Roll and Pitch) responses of the drone model during flying and tracking the desired flight-path are shown in Fig. 20. In this figure, we can see fast and smooth behaviours in the yaw orientation. However, in the roll orientation, there is small oscillation that did not exceed ± 0.025 rad.

Fig. 19 demonstrates the small translation error coordinates and it did not exceed ± 10 cm. The overshoot of the altitude is less than 10 cm. The attitude of (Yaw, Roll and Pitch) responses of the drone model during flying and tracking the desired flight-path are shown in Fig. 20. In this figure, we can see fast and smooth behaviours in the yaw orientation. However, in the roll orientation, there is small oscillation that did not exceed ± 0.025 rad.

The responses of the thrust force control action of each rotor in the drone model are illustrated in Fig. 21. These control actions have a fast response with very small oscillation, no saturation state, and with no

spike actions due to the effectiveness of the on-line CPSO tuning algorithm used to tune the six adaptive PID controllers. This algorithm generated smooth and stable values for the parameters of the six adaptive PID controllers. Moreover, these control actions have the ability to smoothly follow and track the desired flight-path equation of the drone model.

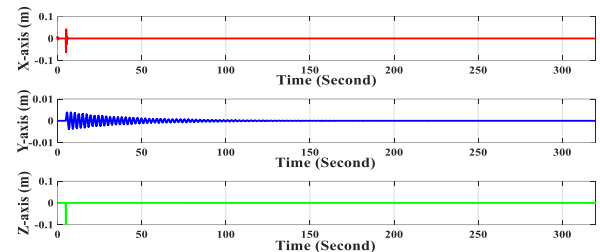


Figure. 19 The (x, y, z) axes translation error coordinates

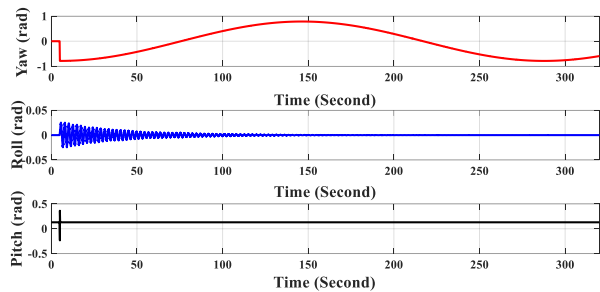


Figure. 20 The desired orientation (Yaw, Roll, and Pitch) coordination

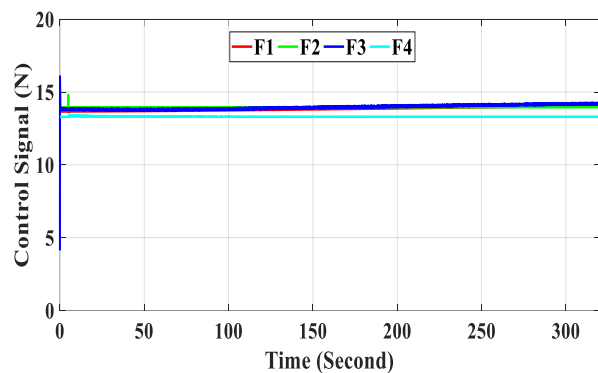


Figure. 21 The thrust force response of each rotors in the drone model

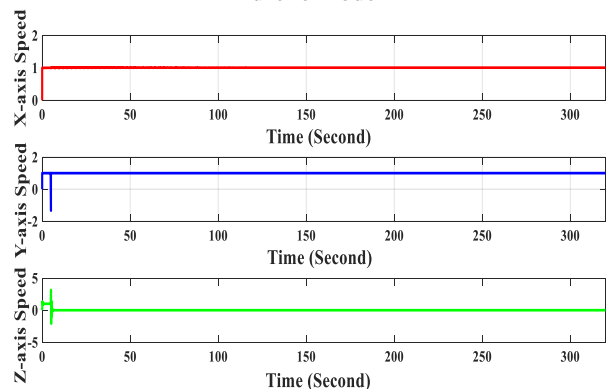


Figure. 22 The drone speed in (x, y, z) axes frame

Table 7. A comparison study between the proposed algorithm and other types of algorithms

Type of Controller	Type of Performance Index	Over-shoot value	Transient and Steady-State Error
[5] PID with try and error tuning algorithm	Off-Line ISE	4.08%	30 cm then ± 12 cm
[15] PID with genetic tuning algorithm	Off-Line ISE	7%	10 cm then ± 4 cm
[24]PID with PSO tuning algorithm	On-Line ISE	1%	10 cm then ± 5 cm
The proposed controller	On-Line MSE	2%	10 cm then zero approximately

The responses of the drone speed in translation coordinate (x, y, and z) during flying in the desired flight-path equation are shown in Fig. 22.

Table 7 demonstrates the effectiveness of this work by comparing the simulation results of the proposed algorithm with other types of algorithms taken from [5, 15, 24].

5. Conclusions

Finding the best path to the intended point is one of the challenges facing the applications of avionics in the field of unmanned aerial vehicles in this paper, the collision-free flight path- finding strategy has been proposed and simulated for the flying drone model to find and track the desired flight path for free flying navigation through the buildings' environment using the MATLAB package. The proposed hybrid B-CPSO-FF algorithm and the six on-line adaptive PID controllers have excellent ability in terms of the following:

- High efficiency of the hybrid algorithm during generating the flight path of the drone in terms of the shortest and smoothest flight-path equation.
- The drone is flied fast from the starting point and reached the target point with very small oscillation and collision-free flying.
- The maximum span for the tracking translation error of the location is less than ± 10 cm.
- The best values of thrust force control actions are generated for solving the highly coupled and nonlinear drone system behaviour.
- There was no saturation state, no spikes and the thrust force action responses were smooth, which

led to precisely tracking the desired flight-path equation. These responses were obtained using the on-line CPSO tuning algorithm, which was used to find stable positive values for the six PID controllers' parameters.

Therefore, we hope to implement the experimental work of the collision-free flight path-finding strategy with the drone model in the future work.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Khulood Eskander Dagher and Mohamed Najm Abdullah developed and contributed the hybrid flight path and six adaptive PID controllers for the drone model and verified the proposed controller. Mohamed Najm Abdullah explained the optimization algorithms. Khulood Eskander Dagher presented the dynamics of the drone model. The numerical simulation results were discussed by both authors.

References

- [1] D. R. Vilorio, E. L. S. Charris, A. M. Villamizar, and J. R. M. Torres, "Unmanned Aerial Vehicles/Drones in Vehicle Routing Problems: a Literature Review", *International Transactions in Operational Research*, Vol. 28, pp. 1626-1657, 2021.
- [2] J. Kwak and Y. Sung, "Autonomous UAV Flight Control for GPS-Based Navigation", *IEEE Access*, Vol. 6, pp. 37947-37955, 2018.
- [3] Y. J. Choi, I. N. Ramatryana, and S. Y. Shin "Cellular Communication-Based Autonomous UAV Navigation with Obstacle Avoidance for Unknown Indoor Environments", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 2, pp. 344-352, 2021.
- [4] J. Guo, C. Liang, K. Wang, B. Sang, and Y. Wu "Three-Dimensional Autonomous Obstacle Avoidance Algorithm for UAV Based on Circular Arc Trajectory", *International Journal of Aerospace Engineering*, Vol. 2021, pp. 1-13, 2021.
- [5] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, "Modified Adaptive Sliding Mode Control for Trajectory Tracking of Mini-drone Quadcopter Unmanned Aerial Vehicle", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 145-158, 2020.
- [6] X. Chen, M. Zhao, and L. Yin, "Dynamic Path Planning of the UAV Avoiding Static and

- Moving Obstacle”, *Journal of Intelligent and Robotic Systems*, 2020.
- [7] S. Mahfoudhi, M. A. Khodja, and F. O. Mahroogi, “A Second-Order Sliding Mode Controller Tuning Employing Particle Swarm Optimization”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 212-221, 2020.
- [8] M. Hehn and R. D’Andrea, “Quadcopter Trajectory Generation and Control”, In: *Proc. of International Conf. on Automatic Control*, Milano, Italy, pp. 14851491, 2011.
- [9] A. Belkadi, H. Abaunza, L. Ciarletta, P. Castillo, and D. Theilliol, “Distributed Path Planning for Controlling a Fleet of UAVs: Application to a Team of Quadrotors”, In: *Proc. of International Conf. on IFAC Paper*, Vol. 50, No. 1, pp. 15983-15989, 2017.
- [10] Z. Lin, L. Castano, E. Mortimer, and H. Xu, “Fast 3D Collision Avoidance Algorithm for Fixed Wing UAS”, *Journal of Intelligent and Robotic Systems*, 2019.
- [11] A. Ashraf, A. Majd, and E. Troubitsyna, “Online Path Generation and Navigation for Swarms of UAVs”, *Scientific Programming*, Vol. 2020, pp. 1-14, 2020.
- [12] C. Zhang, Y. Zhang, and C. Wei, “On board Trajectory Generation of Hypersonic Morphing Aircraft”, *International Journal of Aerospace Engineering*, Vol. 2021, pp. 1-11, 2021.
- [13] A. Salih, M. Moghavvemi, H. Mohammed, and K. Gaeid, “Flight PID Controller Design for a UAV Quadrotor”, *Scientific Research and Essays*, Vol. 5, No. 23, pp. 3660-3667, 2010.
- [14] A. Bouguerra, D. Saigaa, K. Kara, and S. Zeghlache, “Fault-Tolerant Lyapunov-Gain-Scheduled PID Control of a Quadrotor UAV”, *International Journal of Intelligent Engineering and Systems*, Vol. 8, No. 2, pp. 1-6, 2015.
- [15] S. E. I. Hasseni, L. Abdou, and H. E. Glida, “Parameters Tuning of a Quadrotor PID Controllers by using Nature-Inspired Algorithms”, *Evolutionary Intelligence*, 2019.
- [16] B. Selma, S. Chouraqui, and H. Abouaissa, “Fuzzy Swarm Trajectory Tracking Control of Unmanned Aerial Vehicle”, *Journal of Computational Design and Engineering*, Vol. 7, No. 4, pp. 435-447, 2020.
- [17] C. Huang, “Design of Decoupling Fuzzy Logic Controller for Quadrotor UAV”, *Journal of Physics*, Vol. 1684, pp. 1-9, 2020.
- [18] D. Li, “Fuzzy PID Controller to Control the Attitude of Quadrotor UAV”, *Gyancity Journal of Engineering and Technology*, Vol. 6, No. 1, pp. 1-11, 2020.
- [19] R. Guardeno, M. J. Lopez, and V. M. Sanchez, “MIMO PID Controller Tuning Method for Quadrotor based on LOR/LOG Theory”, *Robotics*, Vol. 8, No. 36, pp. 2-28, 2019.
- [20] X. Chen and L. Wang, “Quadrotor Cascade PID Controller Automatic Tuning”, In: *Proc. of International Conf. on Australian Control*, Newcastle, Australia, pp. 311-316, 2016.
- [21] S. Zeghlache, M. Z. Ghellab, and A. Bouguerra, “Adaptive Type-2 Fuzzy Sliding Mode Control using Supervisory Type-2 Fuzzy Control for 6 DOF Octo-rotor Aircraft”, *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 3, pp. 47-57, 2017.
- [22] H. E. Glida, L. Abdou, A. Chelihi, C. Sentouh, and S. E. I. Husseni, “Optimal Model-Free Backstepping Control for a Quadrotor Helicopter”, *Nonlinear Dynamics*, Vol. 100, pp. 3449-3468, 2020.
- [23] P. Ru and K. Subbarao, “Nonlinear Model Predictive Control for Unmanned Aerial Vehicles”, *Aerospace*, Vol. 4, No. 31, pp. 1-26, 2017.
- [24] M. N. Abdullah and K. E. Dagher, “Airborne Computer System Path-Tracking Based Multi-PID-PSO Controller Design”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 403-411, 2021.
- [25] O. A. A. Wahhab and A. S. A. Araji, “Path Planning and Control Strategy Design for Mobile Robot Based on Hybrid Swarm Optimization Algorithm”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 565-579, 2021.
- [26] H. N. Abdullah, “An Improvement in LQR Controller Design based on Modified Chaotic Particle Swarm Optimization and Model Order reduction”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 1, pp. 157-168, 2021.
- [27] H. S. Gebremedhen, D. E. Woldemichael, and F. M. Hashim, “A Firefly Algorithm Based Hybrid Method for Structural Topology Optimization”, *Advanced Modeling and Simulation in Engineering Sciences*, Vol. 7, No. 44, pp. 1-20, 2020.
- [28] A. Ezugwu, M. Agbaje, N. Aljojo, R. Els, H. Chiroma, and M. Elaziz, “A Comparative Performance Study of Hybrid Firefly Algorithms for Automatic Data Clustering”, *IEEE Access*, Vol. 8, pp. 121089-121118, 2020.
- [29] H. Deghbouch and F. Debbat, “A Hybrid Bees Algorithm with Grasshopper Optimization Algorithm for Optimal Deployment of Wireless

Sensor Networks”, *Inteligencia Artificial*, Vol. 24, No. 67, pp. 18-35, 2021.

- [30] A. S. A. Araji, “Genetic Algorithm and Elman Network Used for Tuning the Parameters of the PID Neural Controller Based Model Reference”, *Iraqi Journal of Computer, Communication, Control and Systems Engineering*, Vol. 5, No. 1, pp. 94-109, 2005.
- [31] A. A. Araji, “Design of On-Line Nonlinear Kinematic Trajectory Tracking Controller for Mobile Robot based on Optimal Back-Stepping Technique”, *Iraqi Journal of Computers, Communication and Control & Systems Engineering*, Vol. 14, No. 2, pp. 25-38, 2014.