applied sciences

MDPI

Article

Particle Swarm Optimization and Cuckoo Search-Based Approaches for Quadrotor Control and Trajectory Tracking

Nada El Gmili [1,*], Mostafa Mjahed [2], Abdeljalil El Kari [1] and Hassan Ayad [1]

[1] Faculty of Science and Technology, Cadi Ayyad University, Marrakesh 40000, Morocco; jalil.elkari@gmail.com (A.E.K.); ayad.ha@gmail.com (H.A.)

[2] Royal School of Aeronautics, Marrakesh 40000, Morocco; mjahed.mostafa@gmail.com

* Correspondence: elgmilinada@gmail.com; Tel.: +212-637-028-692

Received: 19 March 2019; Accepted: 17 April 2019; Published: 25 April 2019

Abstract: This paper explores the full control of a quadrotor Unmanned Aerial Vehicles (UAVs) by exploiting the nature-inspired algorithms of Particle Swarm Optimization (PSO), Cuckoo Search (CS), and the cooperative Particle Swarm Optimization-Cuckoo Search (PSO-CS). The proposed PSO-CS algorithm combines the ability of social thinking in PSO with the local search capability in CS, which helps to overcome the problem of low convergence speed of CS. First, the quadrotor dynamic modeling is defined using Newton-Euler formalism. Second, PID (Proportional, Integral, and Derivative) controllers are optimized by using the intelligent proposed approaches and the classical method of Reference Model (RM) for quadrotor full control. Finally, simulation results prove that PSO and PSO-CS are more efficient in tuning of optimal parameters for the quadrotor control. Indeed, the ability of PSO and PSO-CS to track the imposed trajectories is well seen from 3D path tracking simulations and even in presence of wind disturbances.

Keywords: control; quadrotor; nonlinear; PSO; particle swarm optimization; cuckoo search

1. Introduction

Over the past few years, Unmanned Aerial Vehicles (UAVs) of type quadrotors or quadcopters have seen an increasing interest since their wide range of civilian and military applications. These quadrotors are considered the most complex flying machines due to many physical effects influencing their dynamics including aerodynamic effects, gravity, gyroscopic effects, friction, and inertia. However, they have advantages over conventional helicopters. Given that the left and the right motors rotate clockwise, and the front and the rear motors rotate counterclockwise, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight.

Many researchers modeled the quadrotor using the formalisms of Newton-Euler [1,2] and Euler-Lagrange [2–4]. For both methods, the quadrotor modeling is regarded as a delicate task and the model obtained using these approaches is strongly nonlinear, fully coupled, under-actuated (Six Degrees of Freedom System (6-DOF) with only four actuators) and dynamically unstable with complex behavior. In this work, we adopt the most used formalism of Newton-Euler to define the dynamics of the quadrotor.

Regarding the quadrotor's control, there are several types of research done both in linear and nonlinear control methods [5–7]. For practicality, the linear controllers, especially PID (Proportional, Integral, and Derivative)/PD (Proportional and Derivative)/ PI (Proportional and Integral), are easy to design and simple to tune, which perform better in practical implementation [8]. In literature, PIDs are the most used for quadrotor control [2,9,10]. However, tuning the parameters of these controllers is very important to get best performances. In this scope, classical methods can be used. They include the methods of Ziegler-Nichols [11], Graham-Lathrop [12], Naslin [13], Cohen-Coon, Reference Model (RM) [14,15], and many other techniques.

Nature-inspired optimization algorithms can effectively resolve complex problems compared to classical and statistical methods. Some of these algorithms proposed in the literature are Genetic Algorithm (GA) [16,17], Particle Swarm Optimization (PSO) [18], Ant Colony Optimization (ACO) [19], Artificial Bee Colony (ABC) [20], and Cuckoo Search (CS) [21,22]. GA is inspired by the crossover, mutation, and recombination principles of genetics. PSO algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. It incorporates swarming behaviors observed in flocks of birds and schools of fish. ACO is inspired by ants' behavior in seeking food sources. The ABC algorithm is based on the natural behavior of honeybee swarm, and CS algorithm is motivated by the aggressive breeding of a bird called 'cuckoo'. The PSO method has shown superior performances. It has features such as a straightforward algorithm, rapid convergence, easy implementation, and is computationally efficient when applied to a diverse set of optimization problems [15]. In CS, immigration and environmental specifications have the advantage to help cuckoos' groups to converge and reach the best places for breeding and egg laying [21]. Another advantage of CS compared to PSO and GA is that it uses a smaller number of parameters to be tuned, which makes it more adaptable [21]. Due to these advantages, PSO and CS are still successfully applied in recent works for control problems [23–25], and PSO is especially used in tuning PID controllers for quadrotor's control [26,27]. Therefore, regarding PSO and CS programs, there are many common points. Both use the same initialization matrices of particles in n-dimensional research space for PSO and nests with D eggs (problem's dimension) for CS. The fitness function evaluates the quality of particles and nests in the same way. Then, the velocities of displacement of particles and nests are calculated in different ways for PSO and CS, which generates differences in the convergence speeds of both algorithms. For these reasons, it would be very interesting to analyze the social thinking and local search abilities of PSO and CS in quadrotor control.

The main contribution of our paper lies on using PSO and CS to improve results concerning the quadrotor full control. However, the CS algorithm suffers from a low convergence speed, since it uses a fixed step size over generations. To overcome this problem, Naik et al. [28] proposed an adaptive cuckoo search algorithm (ACS) with adaptive step size. The ACS converges to a near optimum solution faster than CS algorithm. However, the problem of ACS algorithm is that it does not achieve better solutions compared to CS in most cases. In our work, we propose the cooperative PSO-CS algorithm that combines the ability of social thinking in PSO with the local search capability of CS. Such a cooperative process is likely to offer proper guidance for cuckoos to the global best positions and to ensure a balance between exploitation and exploration of the search space. Consequently, we are interested in examining if it is possible for us to improve results of quadrotor control by effectively integrating these two algorithms with complementary strengths. Our work treats position and attitude control in order to cover all topics in quadrotor control. For this objective, the control design includes six PID controllers: Three for angles (ϕ, θ, ψ) and three for positions (x, y, z). These controllers are optimally tuned using the heuristics PSO and CS, the cooperative PSO-CS and the classical method of Reference Model (RM). A comparative study is done to highlight the efficiency of the proposed intelligent controllers of PSO and PSO-CS. Indeed, their robustness is demonstrated in presence of wind disturbance.

The rest of this paper is organized as follows. In Section 2, Newton-Euler formalism is used to establish the quadrotor dynamic model. In Section 3, the proposed algorithms of PSO, CS, the cooperative PSO-CS and RM are presented. In Section 4, the quadrotor full control is treated using PID controllers optimally tuned with the proposed methods. In the last section, our conclusions are given.

2. Quadrotor Dynamic Modeling

The quadrotor is a complex flying system, strongly nonlinear, multivariable, fully coupled, and under-actuated (6-DOF and only four control inputs). Therefore, its modeling is a delicate task. For better understanding of its dynamic modeling, different working hypotheses were assumed: A rigid and a symmetrical quadrotor structure (diagonal matrix of inertia); a rigid propellers (negligible effect of deformation during rotation); the lift and the drag forces are proportional to the square of the rotational speed of the rotors (a very close approximation of the aerodynamic behavior); and the center of mass is exactly the origin of reference related to the structure.

From the configuration shown in Figure 1, the quadrotor has two pairs of rotating rotors attached to the end of a cross and the control electronics is situated in the center of the cross. The front and the rear propellers spin counterclockwise, while the left and the right ones spin clockwise, which effectively neutralizes the unwanted reactive torque and allows the vehicle to fly without overturning.

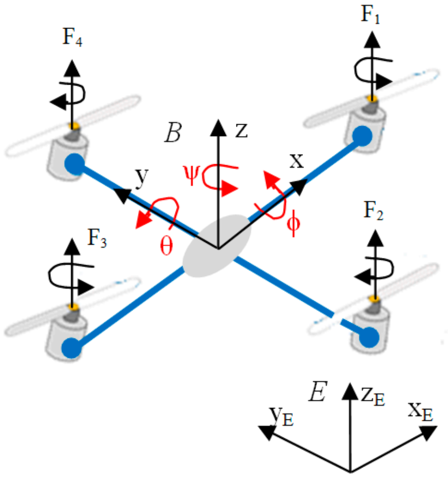

Figure 1. Quadrotor's configuration.

Notice that the absolute position of the mass center is described by the three coordinates (x, y, z) and its attitude by the three Euler's angles $(\phi,\ \theta,\ \psi)$. Using Newton-Euler formalism [2], the dynamic model of the quadrotor can be expressed as:

$$\begin{aligned}\ddot{\varphi} = \tfrac{l\,U_2}{I_x} + \tfrac{(I_y - I_z)}{I_x}\Big(\dot{\psi}\ cos\varphi\ cos\theta - \dot{\theta} sin\varphi\Big)\Big(\dot{\theta} cos\varphi + \dot{\psi}\ sin\varphi\ cos\theta\Big) \\ - \tfrac{J_r(\omega_1 - \omega_2 + \omega_3 - \omega_4)}{I_x}\Big(\dot{\psi}\ sin\varphi\ cos\theta + \dot{\theta}\ cos\varphi\Big) \\ - \tfrac{K_{fax}}{I_x}\Big(\dot{\varphi}^2 - 2\dot{\varphi}\dot{\psi}\ sin\theta^2\Big)\end{aligned} \tag{1}$$

$$\begin{aligned}\ddot{\theta} = \tfrac{l\,U_3}{I_y} + \tfrac{(I_z - I_x)}{I_y}\Big(\dot{\psi}\ cos\varphi\ cos\theta - \dot{\theta}\ sin\varphi\Big)\Big(\dot{\varphi} - \dot{\psi}\ sin\theta\Big) \\ - \tfrac{J_r(\omega_1 - \omega_2 + \omega_3 - \omega_4)}{I_y}\Big(\dot{\psi}\ sin\theta - \dot{\phi}\Big) \\ - \tfrac{K_{fay}}{I_y}\Big(\dot{\theta}^2\ cos\varphi^2 + 2\dot{\varphi}\ \dot{\psi}\ sin\varphi\ cos\varphi\ cos\theta\ + \dot{\psi}^2\ sin\varphi^2\ cos\theta^2\Big)\end{aligned} \tag{2}$$

$$\begin{aligned}\ddot{\psi} = \tfrac{U_4}{I_z} + \tfrac{(I_x - I_y)}{I_z}\Big(\dot{\psi}\ sin\varphi\ cos\theta + \dot{\theta}\ cos\varphi\Big)\Big(\dot{\varphi} - \dot{\psi}\ sin\theta\Big) \\ - \tfrac{J_r(\omega_1 - \omega_2 + \omega_3 - \omega_4)}{I_y}\Big(\dot{\psi}\ sin\theta - \dot{\varphi}\Big) - (\dot{\theta}^2 sin\varphi^2 \\ -2\dot{\varphi}\dot{\psi}\ sin\varphi\ cos\varphi\ cos\theta) + \tfrac{K_{faz}}{I_z}\dot{\psi}^2 cos\varphi^2 cos\theta^2\end{aligned} \tag{3}$$

$$\ddot{x} = \frac{\cos\varphi\ \ \cos\varphi \sin\theta + \sin\varphi \sin\psi}{m} U_1 - \frac{K_{ftx}}{m}\dot{x} \tag{4}$$

$$\ddot{y} = \frac{\sin\psi\ \sin\theta\ \cos\varphi - \cos\psi\ \sin\varphi}{m} U_1 - \frac{K_{fty}}{m}\dot{y} \tag{5}$$

$$\ddot{z} = -g + \frac{\cos\varphi\cos\theta}{m} U_1 - \frac{K_{ftz}}{m}\dot{z} \tag{6}$$

with

$$\begin{cases} U_1 = b\left(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2\right) \\ U_2 = b\left(\omega_4{}^2 - \omega_2{}^2\right) \\ U_3 = b\left(\omega_3{}^2 - \omega_1{}^2\right) \\ U_4 = d\left(\omega_1{}^2 - \omega_2{}^2 + \omega_3{}^2 - \omega_4{}^2\right) \end{cases} \tag{7}$$

where m is the total mass of the quadrotor; l the distance from the center of the body to the rotor; I_x, I_y and I_z are the inertias around x, y, and z axis; b and d thrust and drag coefficients respectively; K_{fax}, K_{fay}, and K_{faz} are frictions aerodynamics coefficients; K_{ftx}, K_{fty}, and K_{ftz} are translation drags coefficients; J_r is the rotor inertia; g represents gravity constant; ω_i is the angular speed of the motor i and finally ϕ, θ, and ψ are rotation around the roll, pitch, and yaw axis, respectively.

3. Proposed Methods

Intelligent and classical control methods of Particle Swarm Optimization (PSO), Cuckoo Search (CS), the cooperative PSO-CS, and the classical Reference Model (RM) are detailed in this section for further reason to be applied in quadrotor control.

3.1. PSO Algorithm

Particle Swarm Optimization (PSO) is a class of stochastic algorithms, which was developed by Eberhart and Kennedy in 1995 for hard optimization problems. The basic principle of PSO is inspired by the social behavior of animals moving in swarm as bird flocking. To search for food, each bird called particle, flies in the space of solutions, and determines its speed according to its personal experience and the information gained through interaction with other swarm members [18].

PSO is employed to intelligently select optimal parameters from N particles. The initialization matrix contains N particles dispersed in a D-dimensional search space. Like most optimization techniques, PSO requires a fitness function relevant to the particle's position. Each particle i stores its best position $Pb_i(t+1)$ and the best solution in its vicinity $Pg(t+1)$, which is the position of the particle that has the smallest fitness value in the swarm as expressed in Equation (9). The mechanism of displacement of each particle is managed by three rules. First, the particle tends to follow the direction of its current velocity. Second, it wants to move toward its best position. Finally, it tends to move to the best position reached by its neighbors [18,29]. In fact, the new velocity matrix V_{ij} *a*nd position matrix X_{ij} of the particle i = {1, 2, . . . , N} in the search space of dimension D, with j = {1, 2, . . . , D} are calculated at iteration *(t+1)*, according to Equations (10) and (11).

$$Pb_i(t+1) = \begin{cases} X_i(t+1) & if \quad f(X_i(t+1)) < f(Pb_i(t)) \\ Pb_i(t) & else \end{cases} \tag{8}$$

$$Pg = min_{i=1,2\ldots N} f(Pb_i(t+1)) \tag{9}$$

$$V_{ij}(t+1) = wV_{ij}(t) + R_1C_1 \otimes \left(Pb_{ij}(t) - X_{ij}(t)\right) + R_2C_2 \otimes \left(Pg_{ij}(t) - X_{ij}(t)\right) \tag{10}$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \tag{11}$$

where

- Pb_{ij} is the best position found by the particle i;
- Pg_{ij} is the best position found by the neighborhood;
- w, C_1, and C_2 are weighting coefficients;

- R_1 and R_2 are random variables generated from a uniform distribution in [0,1];
- $\otimes$ means element wise multiplications.

The weighting coefficients govern the process of finding the best solution and reflect the sociability of the particles. In order to move the particle towards the best solutions, selection of these coefficients must ensure a compromise between local and global exploration of the search space. In previous works [30,31], the constants C_1 and C_2 must verify the condition $C_1 + C_2 \leq 4$. Also, experimental results in Reference [30] show that $[1/2((C_1 + C_2) - 1)] < w \leq 1$. PSO search works in G iterations and the coordinates of the obtained global best position Pg are the parameters optimally tuned.

3.2. CS Algorithm

Cuckoo Search (CS) algorithm, proposed by Yang and Deb in 2009, is based on the life of the ″cuckoo″ bird. The basic principle of this optimization algorithm is the specific breeding and egg laying of this bird. In the habitat of other host birds, adult cuckoos lay some eggs that grow and become mature cuckoos if they are not discovered and removed by host birds. The immigration of groups of cuckoos and environmental specifications hopefully lead them to converge and reach the best places for reproduction and breeding [21].

The primary population of CS is composed of *N* nests representing a set of solutions, where each nest has multiple eggs (*D* eggs). As in nature, each cuckoo dedicates from five to 20 eggs. These values are used as the limits of the dimension *D*. Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest. The best nests with high-quality eggs (solutions) will carry over to the next generations, where the profit of a solution is obtained by evaluation of a fitness function *F* of the habitat, so the profit is an array of $1 \times D$. The number of available host nests is fixed, and a host can discover an alien egg with probability P_a from [0, 1]. In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location. For simplicity, this assumption can be approximated by a fraction P_a of the *N* nests being replaced by new nests, having new random solutions [21].

To replace solutions in the nests with new solutions, *Lévy* flights mechanism is used. A new solution $X_i(t + 1)$ for cuckoo *I* is given by Equation (12), where $\otimes$ is entry wise product, similar to that used in PSO, $\alpha > 0$ is the step size, and *R* is a random variable generated from a uniform distribution in the interval [0, 1] in order to provide a stochastic weight.

$$X_i(t+1) = X(i) + \alpha R \otimes Levy(\lambda) \tag{12}$$

The random walk via Lévy flight is more efficient in exploring the search space, as its step length is drawn from a Lévy distribution and it is much longer in the long run. The Mantegna algorithm is used as Lévy stable distribution in most problems to decide the step length. Thus, the step length *S* from Mantegna algorithm can be written as represented by Equation (13), where *u* and *v* are samples drawn from Gaussian normal distribution given by Equation (14), and ${\sigma_u}^2$ is the variance of the distributions given by Equation (15), where Γ is the gamma function [22].

$$Levy(\lambda) \approx S = u|v|^{-\frac{1}{\lambda}} \qquad (1 < \lambda \leq 3) \tag{13}$$

$$\begin{cases} u \approx Norm\left(0, \sigma_u^2\right) \\ v \approx Norm\,(0, 1) \end{cases} \tag{14}$$

$$\sigma_u^2 = \left\{ \frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma[(1+\lambda)/2]\lambda 2^{(\lambda-1)/2}} \right\}^{1/\lambda} \tag{15}$$

The fraction P_a of worse solutions are generated as given by Equation (16), where $X_j(t)$ and $X_k(t)$ are two random solutions chosen by random permutation, *H* is a Heaviside function, and *r* is a random number drawn from a normal distribution.

$$X_i(t+1) = X_i(t) + (P_a - r)\Big(X_j(t) - X_k(t)\Big) \tag{16}$$

3.3. Cooperative PSO-CS Algorithm

The initialization matrix in PSO contains N particles dispersed in a D-dimensional search space, when in CS, this matrix is composed of N nests and each nest has D eggs (problem's dimension). In both PSO and CS, this primary population can be regarded of dimension $D \times N$ and the quality of these solutions is evaluated in the same way. The global best particle in PSO (or the best nest in CS) is the particle (or the nest) that has the smallest fitness value among all potential solutions. Then, the velocities of displacement of particles and nests are calculated in different ways in PSO and CS. However, due to the fast convergence of PSO algorithm and the low convergence speed of the CS algorithm, the cooperation between PSO and CS would be very interesting to combine the ability of social thinking in PSO with the local search capability of CS. This cooperative PSO-CS aims to modify the speed equation of displacement by combining the Lévy flights random walks of cuckoos and the movement of particles toward the global best solution in their vicinity Pg. New solutions X_i $(t + 1)$ are given by:

$$X_i(t+1) = X_i(t) + \alpha R \otimes Levy(\lambda) + C_2 R_2 \otimes (Pg(t) - X_i(t)) \tag{17}$$

The use of cooperative PSO-CS helps to improve searching at global and local scales, which allows a balance between exploration and exploitation of the search space. Both algorithms are combined and involved to produce results. This combination of PSO and CS capabilities contribute to increase particle diversification in the entire search space and ensure good coverage during iterations. Thanks to this cooperation, CS algorithm can be proposed to control the quadrotor. Indeed, to evaluate the efficiency of the proposed PSO-CS method, comparisons with the intelligent approaches of PSO and CS are presented in the next sections for quadrotor's control and trajectory tracking.

3.4. RM Method

Reference Model (RM) technique is a linear method that approaches the behavior of any system of "n" order to the desired behavior of a reference system of first or second order. If the reference behavior is that of a first order, the desired settling time T_s is fixed and the value of τ is deduced, such that T_s = 3τ. The dominant pole is then to place at $-1/\tau$ and the (n−1) other poles on left of $-1/\tau$ in the complex plane [14,15]. If the reference behavior is that of a second order, with imposed values for damping ζ and natural pulsation ω_n, the two dominant poles are to place at $-\zeta\omega_n \pm j\omega_n\sqrt{}(1 - \zeta^2)$ and the (n−2) other poles on left of $-\zeta\omega_n$ in the complex plane [14,15].

The closed loop transfer function $H(p)$, of the system including the controller and its denominator $D_H(p)$ are computed. By taking into account some desired specifications (T_s, τ or ζ, ω_n), the characteristic polynomial $\mathrm{R_s}(p)$ of the reference system is evaluated.

$$\mathrm{R_s(P)} = \begin{cases} \left(p + \frac{1}{\tau}\right)\left(p + \frac{\alpha}{\tau}\right)^{n-1} & \textit{if a first order behavior desired} \\ \left(p^2 + 2\zeta\omega_n p + {\omega_n}^2\right)(p + \alpha\zeta\omega_n)^{n-2} & \textit{else} \end{cases} \tag{18}$$

The parameter "α" in (18) is made much greater than 1 to ensure the dominance of the poles $-1/\tau$ or $-\zeta\omega_n \pm j\omega_n\sqrt{}(1 - \zeta^2)$. By equating the coefficients of $D_H(p)$ and $D_s(p)$, the controller parameters can be extracted [14,15].

4. Quadrotor Control

4.1. Virtual Control

To manage the system dynamics, a simplified model is needed. By considering small rotations (*cos x* = 1 and *sin x* = *x*), the equations become linear and the PID controllers' inputs can be clearly seen. Consequently, the control inputs for ϕ, θ, and ψ are U_2, U_3, and U_4, respectively. To deal with the under-actuated part, ϕ, θ, and U_1 are used to create three virtual control inputs so that every output will be controlled separately. Physically, these virtual controls mean that the translation motion along x, y, and z are controlled indirectly by the three common inputs (ϕ, θ, and U_1). These virtual controls are given by Equation (19) and the desired trajectories of roll (ϕ_d), pitch (θ_d) and U_{1d} are obtained by inverting Equation (19).

$$\begin{cases} U_x = (cos\varphi cos\psi sin\theta + sin\varphi sin\psi)U_1 \\ U_y = (sin\psi sin\theta cos\varphi - cos\psi sin\varphi)U_1 \\ U_z = (cos\varphi cos\theta)U_1 \end{cases} \tag{19}$$

$$\begin{cases} \varphi_d = \sin^{-1}\left(\frac{U_x\ sin\psi_d - U_y\ cos\psi_d}{\sqrt{U_x^2 + U_y^2 + U_z^2}}\right) \\ \theta_d = \tan^{-1}\left(\frac{U_x\ cos\psi_d + U_y\ sin\psi_d}{cos\varphi_d}\right) \\ U_{1d} = \sqrt{U_x^2 + U_y^2 + U_z^2} \end{cases} \tag{20}$$

4.2. Control Law Design

For an appropriate control of the UAV quadrotor system described by Equations (1)–(6), PID controllers are used. These controllers are simple to design and robust. The complete control scheme for the quadrotor is represented in Figure 2. It can be clearly seen from this figure that three PID controllers are used for positions (x, y, z) and three PIDs for attitudes (ϕ, θ, ψ). The inputs are divided to two parts, the desired and the sensor signals. Desired signals for positions and yaw angle (x_d, y_d, z_d, ψ_d) are fixed by the pilot or the autopilot program. Sensor signals for desired roll and pitch angles (ϕ_d, θ_d, and U_{1d}) are computed from the three virtual inputs U_x, U_y, and U_z.

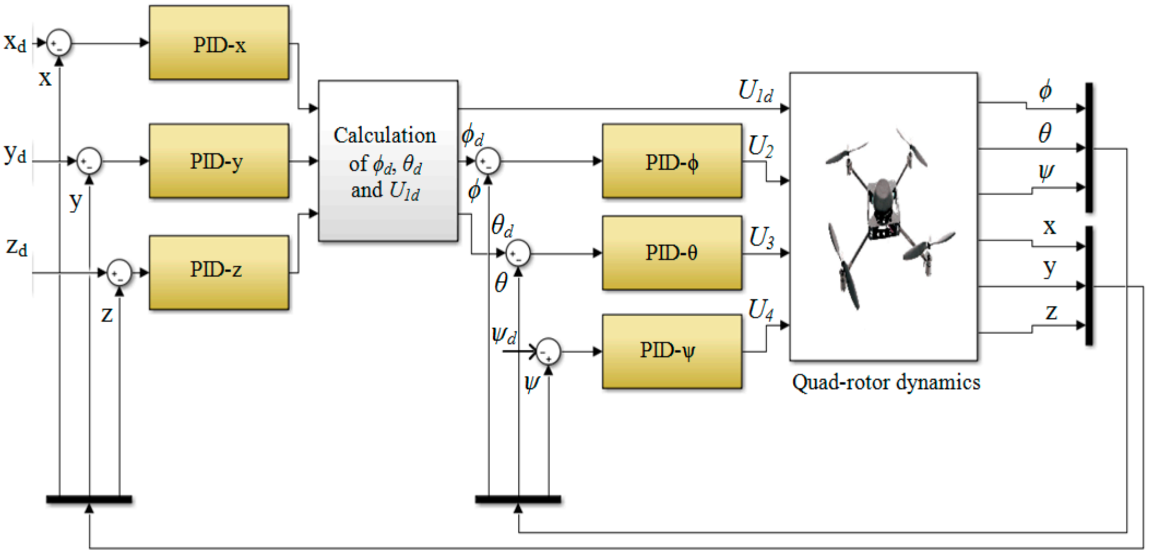

Figure 2. Complete control scheme for quadrotor.

The transfer functions $C_s(p)$ of PID controllers are given in Equation (21), where $s = \{\phi, \theta, \psi, x, y, z\}$. Their gains ($K_{ps}$, K_{is}, K_{ds}) should be optimally tuned for the quadrotor in order to ensure stable, fast, precise, and robust responses. For this objective, PSO, CS, PSO-CS, and RM methods are applied to realize an optimal setting of these parameters.

$$C_s(p) = K_{ps} + \frac{K_{is}}{p} + K_{ds}p \tag{21}$$

4.3. Quadrotor Intelligent Control Using PSO, CS, and PSO-CS

To control the nonlinear and fully coupled quadrotor defined by Equations (1)–(6), it is suitable to use a method directly applicable to nonlinear systems as PSO, CS, and the cooperative PSO-CS. Instead of linear PID controllers, the intelligent and adaptive controllers can look for optimal PIDs parameters for the six quadrotors' outputs [ϕ, θ, ψ, x, y, z].

The initial populations (N = 200) of PSO, CS, and PSO-CS are randomly dispersed in a search space that is chosen sufficient to contain all possible solutions [0, 200] and its dimension D is set to 18. To define the exploration capacity of each particle in order to improve the convergence of the process and to not exceed the velocity limits, we tested numerous values of w, and the compromise between local and global exploration in PSO was achieved for w = 0.8. The confidence coefficients C_1 and C_2 define the exploration capacity. They determine the degree of influence of the past positions of the particle itself and those of the other particles of the swarm in order to regulate the velocities relative to the best local and global positions, thus making it possible to determine the movements relative to these best positions. $C_1\mathrm{R}_1$ and C_2R_2 are represented by matrices with dimension of 200 × swarm size. We considered that C_1R_1 takes random values in [0, 0.8] to avoid the problem of fast convergence, when C_2R_2 takes random values in [0, 1.2] to give more importance to the global best solution Pg (the interval of $\mathrm{C}_2\mathrm{R}_2$ is large to that of $\mathrm{C}_1\mathrm{R}_1$). In CS, the parameters used in experiments are as follows: Abandon probability P_a = 0.25, and the Lévy flights settings, α = 0.1 and λ = 1.5. In PSO-CS, the same settings of PSO and CS are conserved: P_a = 0.25, and the Lévy flights settings, α = 0.1, λ = 1.5 and C_2 with random values in [0, 1.2].

To evaluate the optimization performance of the quadrotor's responses using the proposed approaches, the fitness function (profit) is defined in a similar way in order to minimize the differences between the desired and the controlled outputs responses of the quadrotor. This fitness function is defined in Equation (22) as the sum of the errors E_k that characterize the difference in behavior between the inputs and the outputs of the quadrotor based on the Integral Square Error (ISE), with l = 6 and k = {1, 2, 3, 4, 5, 6}.

$$F = \sum\nolimits_{k=1}^{l} \int_0^\infty E_k^2(t)dt \tag{22}$$

$$ISE = \int_0^\infty E_k^2(t)dt \tag{23}$$

The performances found with ISE are compared to those established using three other performance indices: Integral Absolute Error (IAE), Integral Time Absolute Error (ITAE), and Integral Time Square Error (ITSE).

$$IAE = \int_0^\infty \left|E_k(t)\right|dt \tag{24}$$

$$ITAE = \int_0^\infty t\left|E_k(t)\right|dt \tag{25}$$

$$ITSE = \int_0^\infty t\, E_k^2(t)dt \tag{26}$$

The maximum number of generations for the three programs (PSO, CS and PSO-CS) is fixed as the stop criterion and set to 20.

4.4. Quadrotor Classical Control Using RM

To use the linear RM technique for quadrotor control, the complete nonlinear Equations (1)–(6) should be simplified. The most used technique to simplify these nonlinear equations is to consider weak angles variations (reduced gyroscopic effects) and retain the linear portion, the Equations (1)–(6) become:

$$\begin{cases} I_x\,\ddot{\varphi} = lU_2 \\ I_y\,\ddot{\theta} = lU_3 \\ I_z\,\ddot{\psi} = U_4 \\ \ddot{x} = \frac{\cos\varphi\cos\psi\sin\theta+\sin\varphi\sin\psi}{m}U_1 - \frac{K_{ftx}}{m}\dot{x} \\ \ddot{y} = \frac{\sin\psi\sin\theta\cos\varphi-\cos\psi\sin\varphi}{m}U_1 - \frac{K_{fty}}{m}\dot{y} \\ \ddot{z} = -g + \frac{\cos\varphi\cos\theta}{m}U_1 - \frac{K_{ftz}}{m}\dot{z} \end{cases} \tag{27}$$

The inputs U_2, U_3, U_4, U_x, U_y, and U_z affect respectively angles of roll, pitch, and yaw, and positions x, y, and z, the reason for their usage as the PID controllers' inputs. By applying the Laplace transform, we obtain the following transfer functions for angles (ϕ, θ, ψ) and positions (x, y, z).

$$\begin{cases} G_\varphi(p) = \frac{\varphi\,(p)}{U_2\,(p)} = \frac{l}{I_x p^2} \\ G_\theta(p) = \frac{\theta\,(p)}{U_3\,(p)} = \frac{l}{I_y p^2} \\ G_\psi(p) = \frac{\psi\,(p)}{U_4\,(p)} = \frac{1}{I_z p^2} \\ G_x(p) = \frac{x\,(p)}{U_x\,(p)} = \frac{K_x}{p\,(\tau_x\,p+1)} \\ G_y(p) = \frac{y\,(p)}{U_y\,(p)} = \frac{K_y}{p\,\left(\tau_y\,p+1\right)} \\ G_z(p) = \frac{z\,(p)}{U_z\,(p)} = \frac{K_z}{p\,(\tau_z\,p+1)} \end{cases} \tag{28}$$

where $K_j = 1/K_{ftj}$, $\tau_j = m/K_{ftj}$ and $j = \{x, y, z\}$.

For quadrotor's attitudes and positions control, the behavior of the linear equations defined in (28) is approached to a desired reference one of a first order system. The desired behaviors for angles and positions correspond to the characteristic polynomials $R_s(p)$, where $s = \{\phi, \theta, \psi, x, y, z\}$ and α, α_1, α_2 are taken equal to 30, 2 and 30, respectively.

$$\mathrm{R}_s(p) = \left(p + \frac{1}{\tau}\right)\left(p + \frac{\alpha_1}{\tau}\right)\left(p + \frac{\alpha_2}{\tau}\right) \tag{29}$$

For all cases, we calculate the closed loop transfer functions $Hs(p)$ of the six systems including the PID controllers, for which we desire search their parameters. The computed denominators $D_{Hs}(p)$ are given in Equation (30).

$$\begin{cases} D_{H\varphi}(p) = p^3 + \frac{l\,K_{d\varphi}}{I_x}p^2 + \frac{l\,K_{p\varphi}}{I_x}p + \frac{l\,K_{i\varphi}}{I_x} \\ D_{H\theta}(p) = p^3 + \frac{l\,K_{d\theta}}{I_x}p^2 + \frac{l\,K_{p\theta}}{I_x}p + \frac{l\,K_{i\theta}}{I_x} \\ D_{H\psi}(p) = p^3 + \frac{l\,K_{d\psi}}{I_x}p^2 + \frac{l\,K_{p\psi}}{I_x}p + \frac{l\,K_{i\psi}}{I_x} \\ D_{Hx}(p) = p^3 + \frac{K_{dx}+K_{ftx}}{m}p^2 + \frac{K_{px}}{m}p + \frac{K_{ix}}{m} \\ D_{Hy}(p) = p^3 + \frac{K_{dy}+K_{fty}}{m}p^2 + \frac{K_{py}}{m}p + \frac{K_{iy}}{m} \\ D_{Hz}(p) = p^3 + \frac{K_{dz}+K_{ftz}}{m}p^2 + \frac{K_{pz}}{m}p + \frac{K_{iz}}{m} \end{cases} \tag{30}$$

5. Results and Discussions

Simulations of the proposed intelligent and classical PID controllers are carried out on the quadrotor's attitudes (ϕ, θ, ψ) and positions (x, y, z). These PIDs are tested firstly in the case of hovering to a determined position coordinates in order study the performances of the proposed intelligent PSO, CS and PSO-CS and the classical RM controllers. Then, for two different desired trajectories, the controllers' gains are optimized through the intelligent proposed approaches. The main objective is to confirm the ability of these intelligent nature inspired methods to control the quadrotor and to efficiently realize different imposed trajectories even under the presence of disturbances. Table 1 gives the parameters of the quadrotor and Table 2 presents the suggested trajectories' dynamics and their conditions.

Table 1. Quadrotor's parameters.

Parameter (unit)	Value
m (Kg)	0.65
g (m/s^2)	9.806
l (m)	0.4
b (N/rad/s)	2.9842×10^{-5}
d (N.m/rad/s)	7.5×10^{-7}
I_x, I_y (Kg.m^2)	7.5×10^{-3}
I_z (Kg.m^2)	1.3×10^{-3}
J_r (Kg.m^2)	2.8385×10^{-5}
K_{fax}, K_{fay} (N/rad/s)	5.567×10^{-4}
K_{faz} (N/rad/s)	6.354×10^{-4}
K_{ftx}, K_{fty}(N/rad/s)	5.567×10^{-4}
K_{ftz} (N/rad/s)	3.354×10^{-4}

Table 2. Trajectories' conditions and dynamics.

	Initial Conditions	Trajectory Description	Attitude's Saturation
Rectangular Trajectory	$[\phi, \theta, \psi, x, y, z] =$ $[0, 0, 0, 0, 0, 10]$	$x_d(t) = 2[u(t-4) - u(t-12)]$ $y_d(t) = 2[u(t-8) - u(t-16)]$ $z_d(t) = 10u(t-20)$ $\psi_d(t) = 0$	$-90^\circ < \phi, \theta < 90$
Helical Trajectory	$[\phi, \theta, \psi, x, y, z] =$ $[0, 0, 0, 0, 0, 2]$	$x_d(t) = 2\sin(2\pi t)$ $y_d(t) = 2\sin(2\pi t - \pi/2)$ $z_d(t) = tu(t) - (t-18)u(t-18)$ $\psi_d(t) = 0$	

5.1. Comparisons of Intelligent and Classical Control Performances

To get the quadrotor to hover at a determined position coordinates, desired signals for positions $[x_d, y_d, z_d]$ are considered step signals of *1 m* and desired yaw angle (ψ_d) is fixed to *1 degree.* Table 3 summarizes the PID controllers gains computed online using PSO, CS, and PSO-CS, and offline with the conventional RM for the quadrotor's outputs (ϕ, θ, ψ, x, y, z). Figures 3–6 show the step responses for the corresponding outputs [x, y, z, ψ] obtained with these methods. Figures 7 and 8 show the various obtained curves for the desired sensor signals as well as the controlled responses. Table 4 gives the obtained quadrotor's responses performances: Settling time (T_s), maximum overshoot (Mp), and the errors previously mentioned (ISE, IAE, ITAE, and ITSE).

For control systems, the RM method often respects the fixed set of specifications and gives acceptable results [15]. However, this method is linear, and it cannot be applied directly to quadrotor control (the nonlinear quadrotor model). It can be observed from the obtained responses in Figures 3 and 4 and the performances in Table 4 that the search for the parameters of PIDs by RM is not optimal: Responses with overshoots, settling times, and some error values exceed the amplitude of the desired signal. These unsatisfactory performances of the linear RM method in search of PID gains can be explained by the fact that the linear transfer functions defining the system (26) already have integrations.

Table 3. Optimal Proportional, Integral and Derivative (PID) parameters obtained with Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) for the quadrotor.

Quadrotor's Output	**Optimal PID Gains**					
	PSO			**CS**		
	K_p	K_i	K_d	K_p	K_i	K_d
x	42.829	0.167	12.366	27.345	0.865	14.023
y	62.881	0.226	16.065	36.714	0.505	14.056
z	34.545	14.508	9.498	14.756	8.647	8.077
ϕ	14.740	1.749	0.975	17.948	0.178	1.055
θ	13.872	0.809	0.555	19.430	0.867	1.806
ψ	15.299	0.702	0.994	7.967	0.198	0.278
	PSO-CS			**RM**		
	K_p	K_i	K_d	K_p	K_i	K_d
x	69.822	−0.011	16.974	59.150	29.250	30.550
y	45.880	0.008	15.067	39.650	19.500	20.800
z	31.859	17.724	6.715	26.066	10.394	16.865
ϕ	206.617	3.941	1.493	2.739	2.608	0.783
θ	121.842	−1.648	2.217	3.293	1.630	1.695
ψ	38.469	0.469	1.466	1.092	1.040	0.31

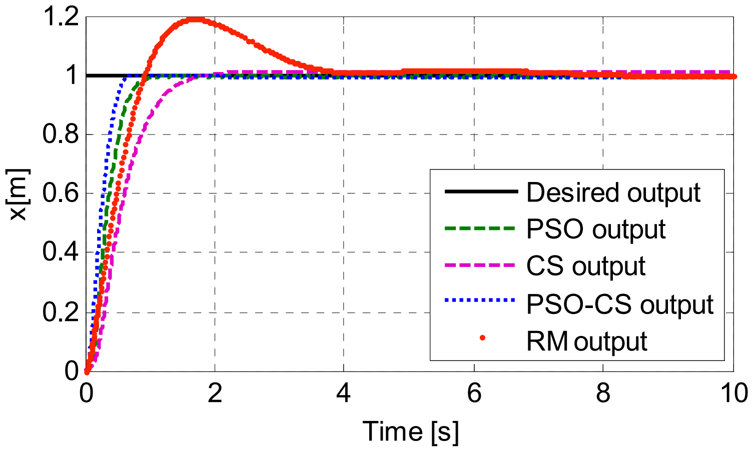

Figure 3. Quadrotor's position x using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM).

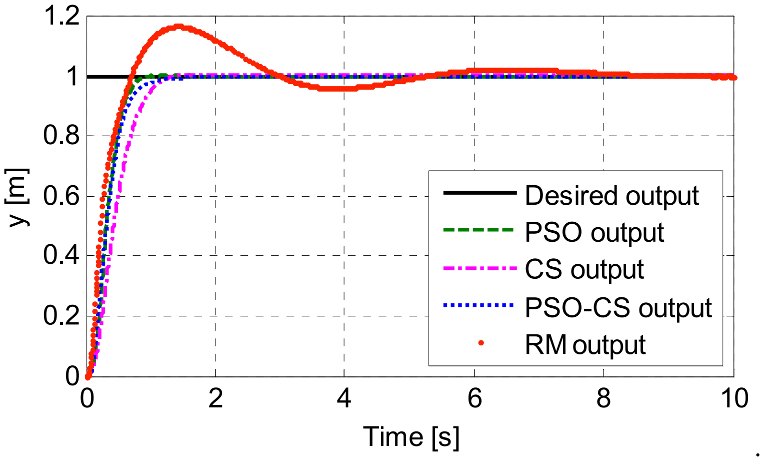

Figure 4. Quadrotor's position y using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM).

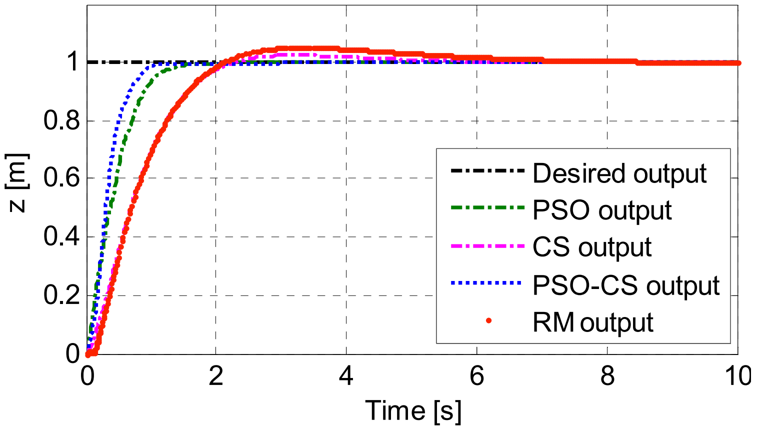

Figure 5. Quadrotor's altitude z using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM).

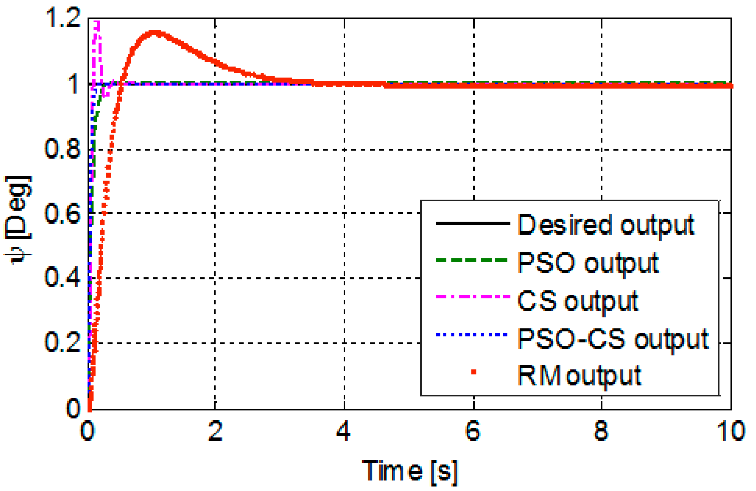

Figure 6. Quadrotor's yaw attitude (ψ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM).

CS results present a considerable overshoot in yaw angle and the errors values of ISE and IAE are around 50% of the desired altitude z, which proves that ISE value weight more significantly the quality of solutions and more adequate to evaluate the fitness function. However, the PSO approach is suitable and results in better performances. Comparisons of PSO results to those of Genetic Algorithms presented in Reference [16] show that PSO reduces the settling time T_s from 2.5 s to 0.172 s for yaw angle. For altitude z, the value of T_s is increased to 1.07 s instead of 0.4 s for GA. These results illustrate that PSO and GA techniques are highly competitive to explore the optimal solution in quadrotor control. In the same project of quadrotor control, the work presented in Reference [32] suggests three intelligent controllers, multilayer neural networks controller based on reference model (RMNN), Neural Networks with PD's controller behavior (CPDNN), and Fuzzy Logic to adjust PID controllers' gains (Fuzzy-PID). The proposed Fuzzy-PID and CPDNN achieves better performances compared to RMNN. Thus, comparing PSO results to those of Fuzzy-PID and CPDNN confirm that PSO is highly competitive to soft computing methods.

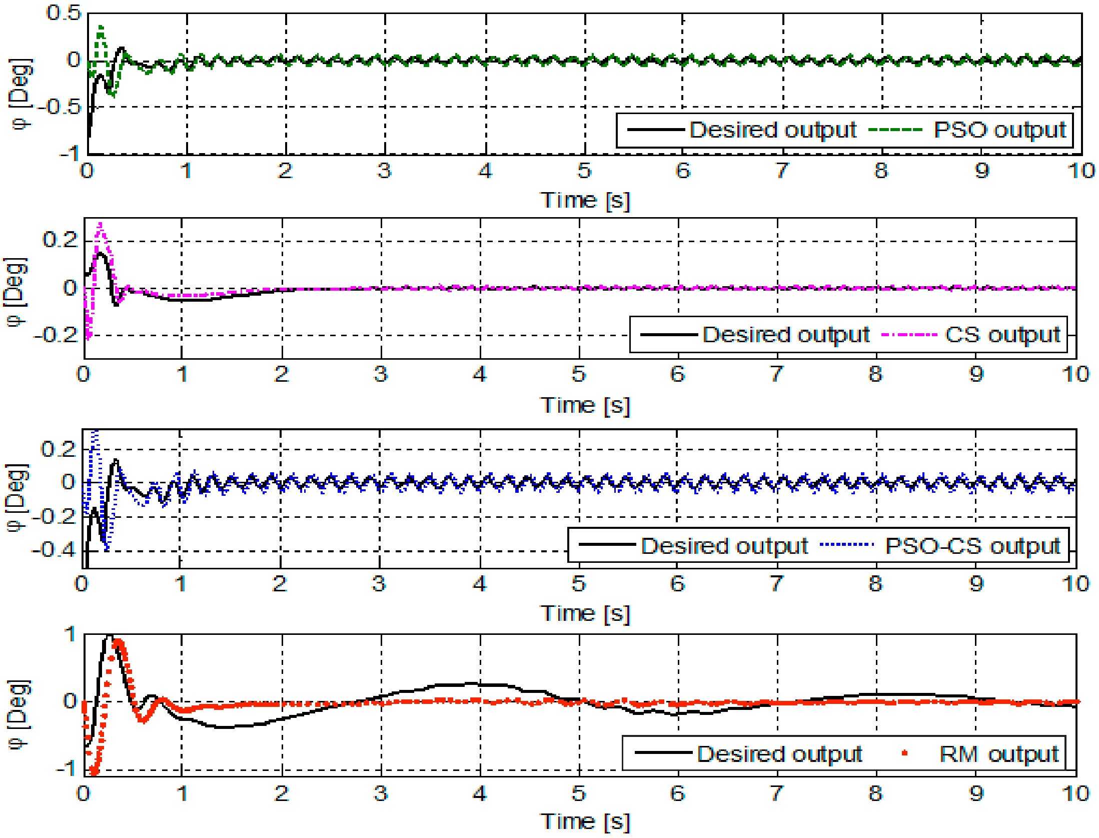

Figure 7. Quadrotor's roll attitude (ϕ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM).

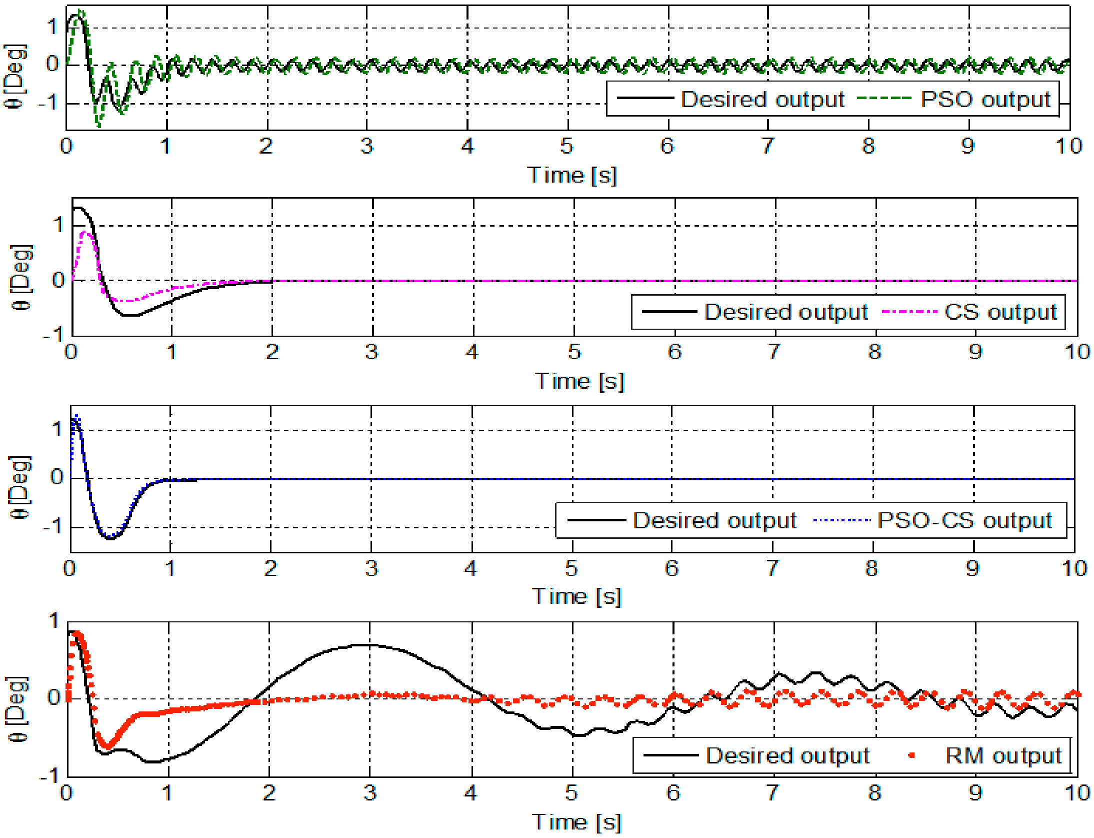

Figure 8. Quadrotor's pitch attitude (θ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM).

Considering the Figures 3–8 and Table 4, the controllers tuned using PSO-CS have good performances for positions and attitudes. Thus, PSO-CS results are seen concurrent to those obtained using PSO, where responses lead to fast settling times with no overshoot and reduced errors. The desired sensor signal curves in Figures 7 and 8 (desired roll and pitch angles (ϕ_d and θ_d)) depend on the outputs of the controllers *PID-x*, *PID-y*, and *PID-*z. It is observed from this figure that these intelligent approaches, and especially PSO and PSO-CS, allow to follow perfectly the sensor signals for roll and pitch (ϕ_d and θ_d), and give controlled responses close to the desired curves.

These programs are implemented using Matlab R2016a and performed on a computer with a processor Intel® Core™ i7-3770 CPU of 3.40 GHz and 8.0 GB of RAM. From the conducted experiments, we can get conclusions that for the same goal, the PSO algorithm spends less CPU time (10 min 45 s) than the PSO-CS algorithm (14 min 12 s) and the CS algorithm (12 min 15 s). During the calculation of the velocity update equation, PSO-CS consumes an additional CPU time as the algorithm combines the two matrices generating Lévy flights random walks of cuckoos and that generating the movement of particles toward the global best solution Pg. However, the PSO-CS and PSO algorithms get higher performances (reduced errors) than the CS algorithm and RM method. These results are achieved thanks to the good adjustment of PSO, CS and PSO-CS parameters: Position and velocity initializations, the appropriate choice of the fitness function, the good tuning of the weighting coefficients w, C_1, and C_2, the chosen probability P_a, and the Lévy flights settings for α and λ.

Table 4. Performances obtained with Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) for the quadrotor.

Output	**Performances**	**Method**			
		PSO	***CS***	***PSO-CS***	***RM***
x	T_s (s)	0.730	1.384	0.522	3.779
	M_p (%)	0.412	1.769	0.446	22.869
	ISE	0.227	0.346	0.1650	0.365
	IAE	0.324	0.492	0.234	0.714
	ITSE	0.034	0.083	0.018	0.176
	ITAE	0.088	0.262	0.038	0.847
y	T_s (s)	0.675	1.018	0.782	3.517
	M_p (%)	0.497	0.898	0.010	30.741
	ISE	0.237	0.296	0.226	0.441
	IAE	0.322	0.410	0.319	0.802
	ITSE	0.036	0.058	0.034	0.245
	ITAE	0.083	0.152	0.076	0.954
z	T_s (s)	1.070	1.821	0.813	1.855
	M_p (%)	0.107	2.463	0.002	3.467
	ISE	0.243	0.406	0.208	0.342
	IAE	0.397	0.587	0.308	0.617
	ITSE	0.050	0.129	0.031	0.122
	ITAE	0.134	0.312	0.081	0.521
ϕ	ISE	0.028	0.006	0.005	0.093
	IAE	0.197	0.053	0.032	0.431
	ITSE	0.016	0.001	0.000	0.076
	ITAE	0.388	0.036	0.013	0.741
θ	ISE	0.334	0.129	0.024	0.912
	IAE	0.886	0.271	0.068	1.509
	ITSE	0.386	0.026	0.001	1.238
	ITAE	2.006	0.134	0.023	2.822
ψ	T_s (s)	0.172	0.207	0.097	2.228
	M_p (%)	0.300	20.424	0.047	16.129
	ISE	0.036	0.037	0.022	0.138
	IAE	0.073	0.068	0.038	0.331
	ITSE	0.0012	0.012	0.000	0.033
	ITAE	0.038	0.011	0.006	0.244

5.2. Rectangular Trajectory Tracking

This experimental simulation on the quadrotor engages to test the PIDs' efficiency when the vehicle tracks a square trajectory without overturning. In this mode, inputs for desired positions (x_d, y_d, z_d) are considered pulse signals and desired yaw attitude is set to 0°. Figure 9 shows the tracking of each position coordinate (x, y, z) and the tracking trajectories in 3D plan obtained using the proposed methods. Responses for desired and controlled roll (ϕ) and pitch (θ) attitudes are illustrated in Figure 6. Table 5 resumes the errors ISE used to compute fitness function.

It can be noticeably observed from Figure 9 and Table 5 that the adaptive autotuned controllers using PSO, CS, and PSO-CS have good performances (reduced errors) for positions (x, y, z) control. PSO-CS and PSO responses always lead to fast settling times with no overshoot for x position. RM presents overshoots for positions and CS shows few overshoots for y position and altitude z. The intelligent PSO and the cooperative PSO-CS drive the system toward perfect tracking of the prescribed square trajectory. Also, all these methods allowed maintaining the yaw angle in zero degrees: no overturning of the aircraft. Figures 10 and 11 confirm that these intelligent approaches of PSO-CS nd PSO give controlled responses closed to the desired roll and pitch attitudes.

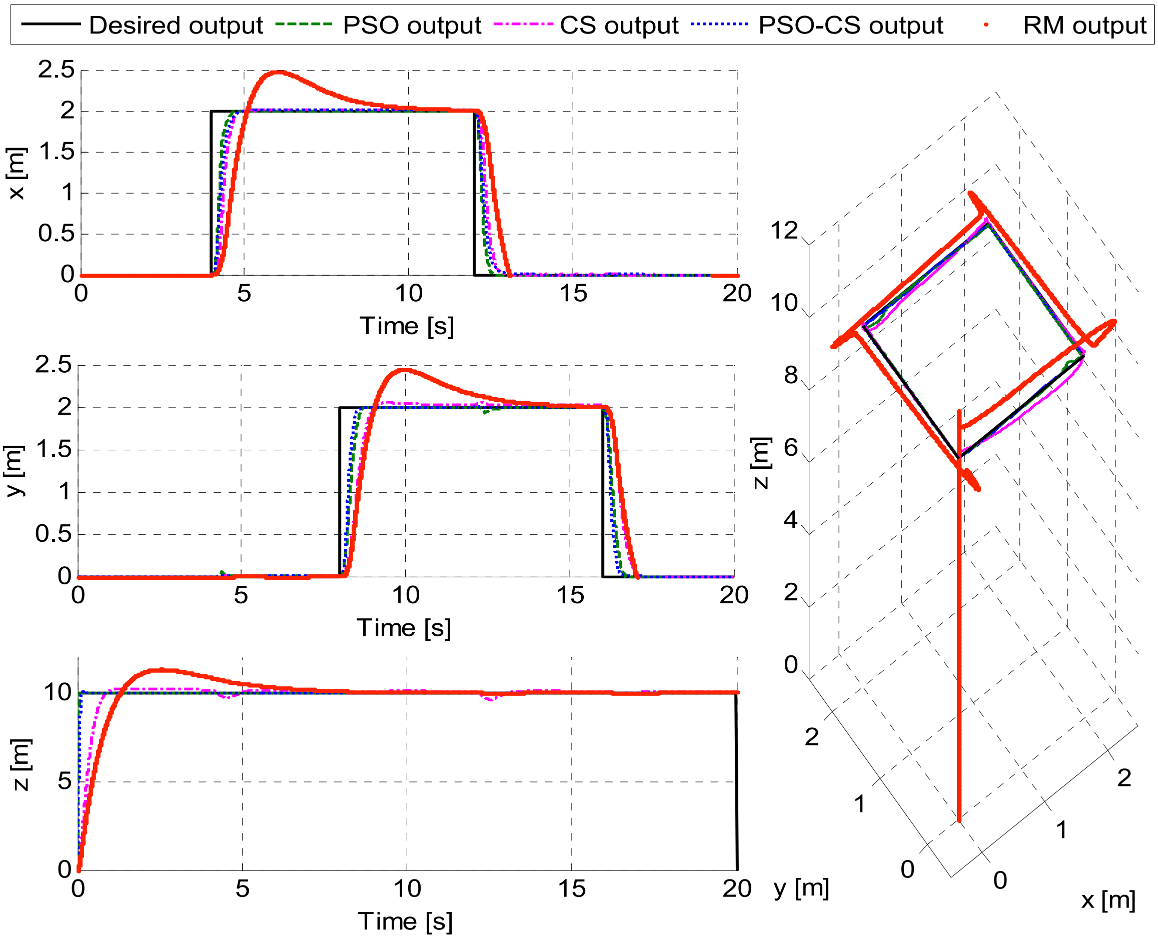

Figure 9. Quadrotor's positions (x, y, z) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) (rectangular trajectory).

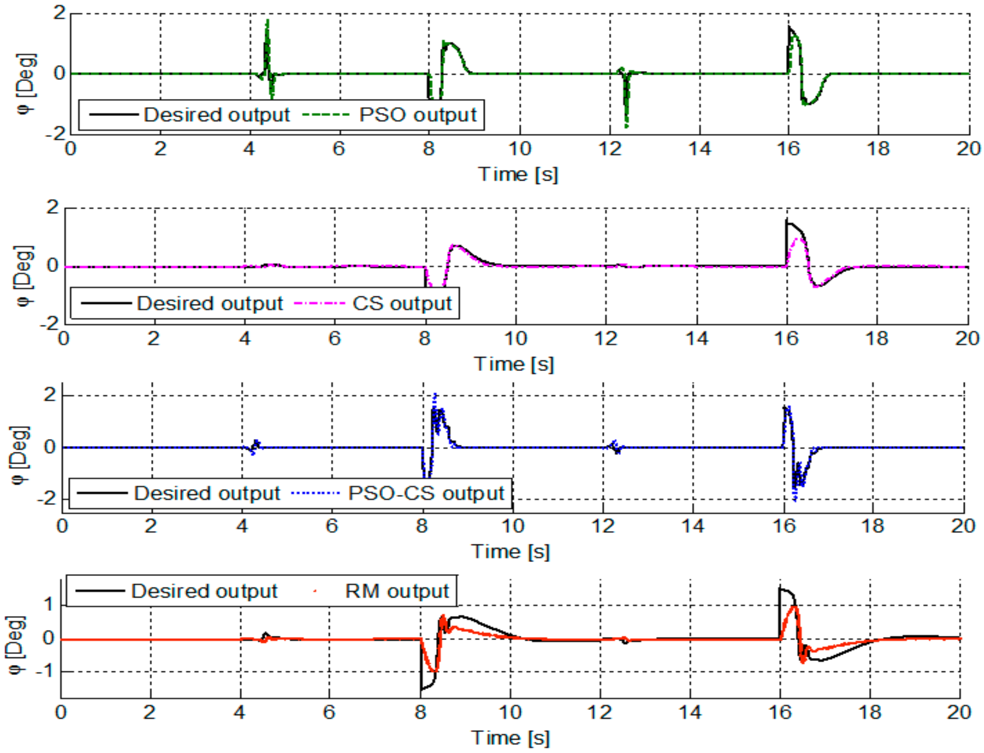

Figure 10. Quadrotor's roll attitude (*ϕ*) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) (rectangular trajectory).

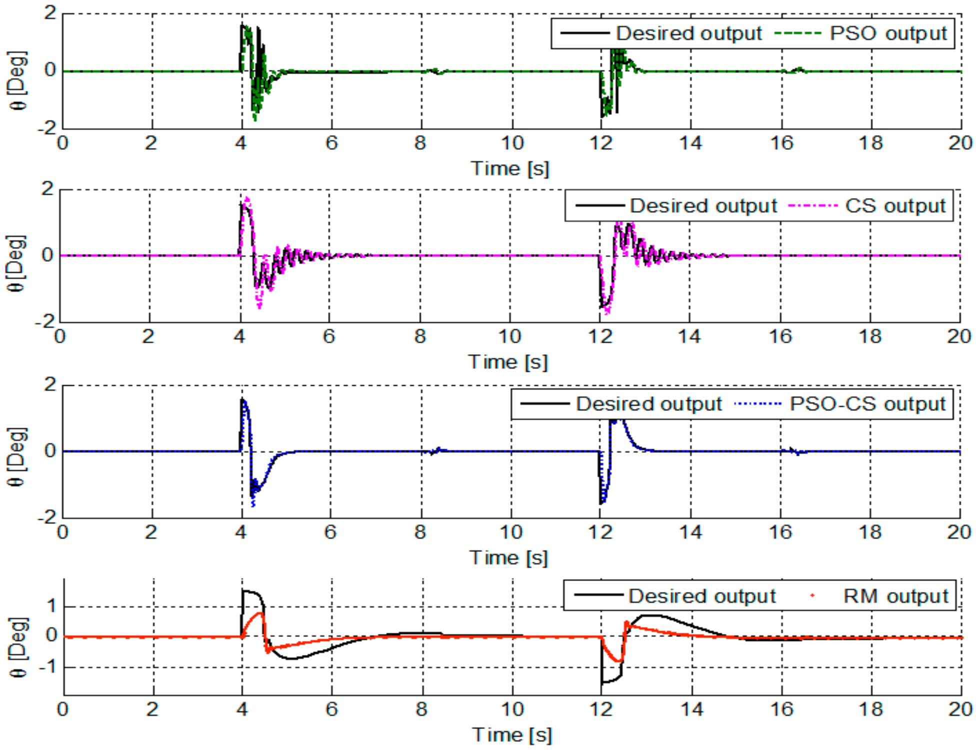

Figure 11. Quadrotor's pitch attitude (*θ*) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) (rectangular trajectory).

Table 5. Errors (ISE-Integral Square Error) obtained with Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) for the quadrotor (rectangular trajectory).

Method	**Errors ISE for Output**					
	x	y	z	ϕ	θ	ψ
PSO	1.083	0.677	3.287	0.119	0.093	7.586×10^{-36}
CS	1.447	1.327	21.46	0.152	0.318	1.001×10^{-35}
PSO-CS	0.955	0.795	1.453	0.107	0.4022	1.651×10^{-36}
RM	2.573	1.561	33.756	0.277	0.712	1.210×10^{-34}

To test the robustness of these PID controllers, external disturbance was introduced. The most likely disturbance acting on the quadrotor is the wind disturbance along z axis. The external disturbance is modeled in this simulation by a random signal of variable amplitude inserted in the closed loop scheme of altitude z. It is noticed from Figures 12–14 and the errors ISE presented in Table 6 that when random disturbance affects the quadrotor, PSO, CS, and PSO-CS controllers reject these undesirable effects and provide outputs responses close to the desired ones. Therefore, we can affirm that PSO, CS, and especially the cooperative PSO-CS, are able to tune optimally the six PID controllers and drive the quadrotor to the desired altitude z, positions x and y, and yaw angle.

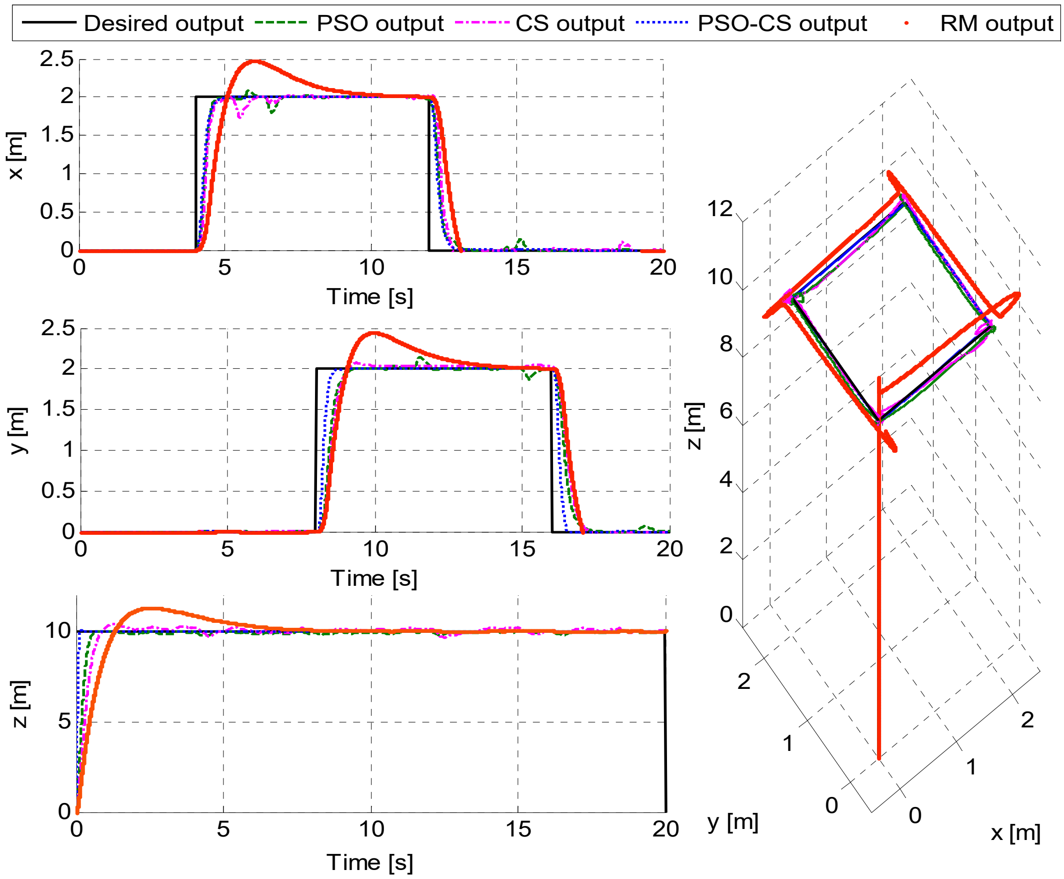

Figure 12. Quadrotor's positions (x, y, z) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) with disturbance (rectangular trajectory).

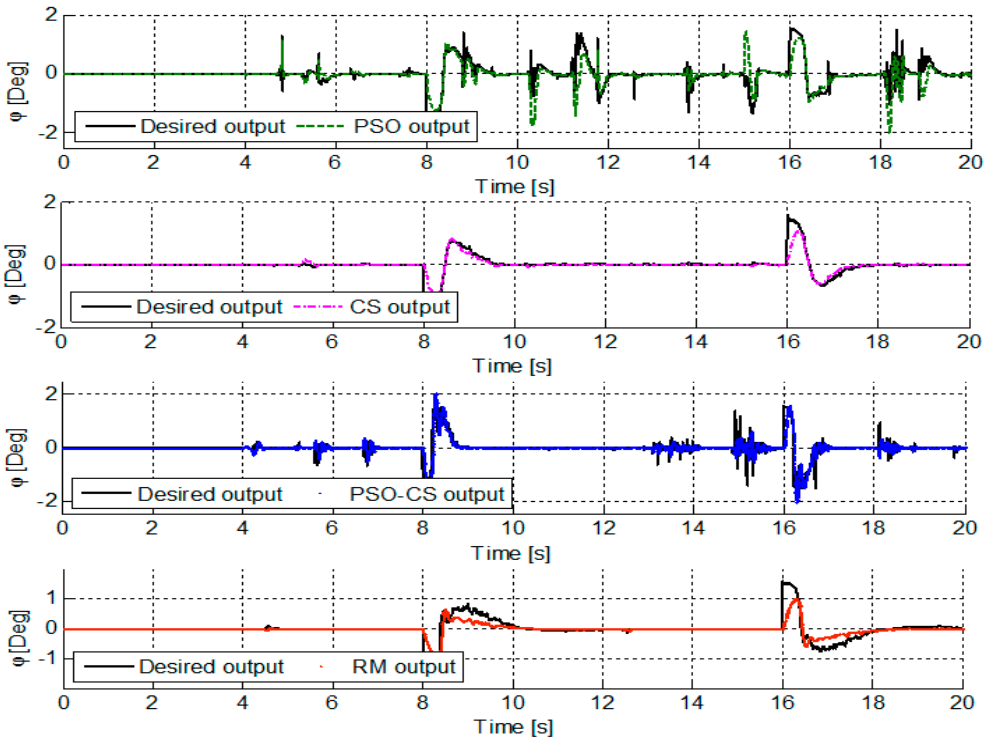

Figure 13. Quadrotor's roll attitude (ϕ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) with disturbance (rectangular trajectory).

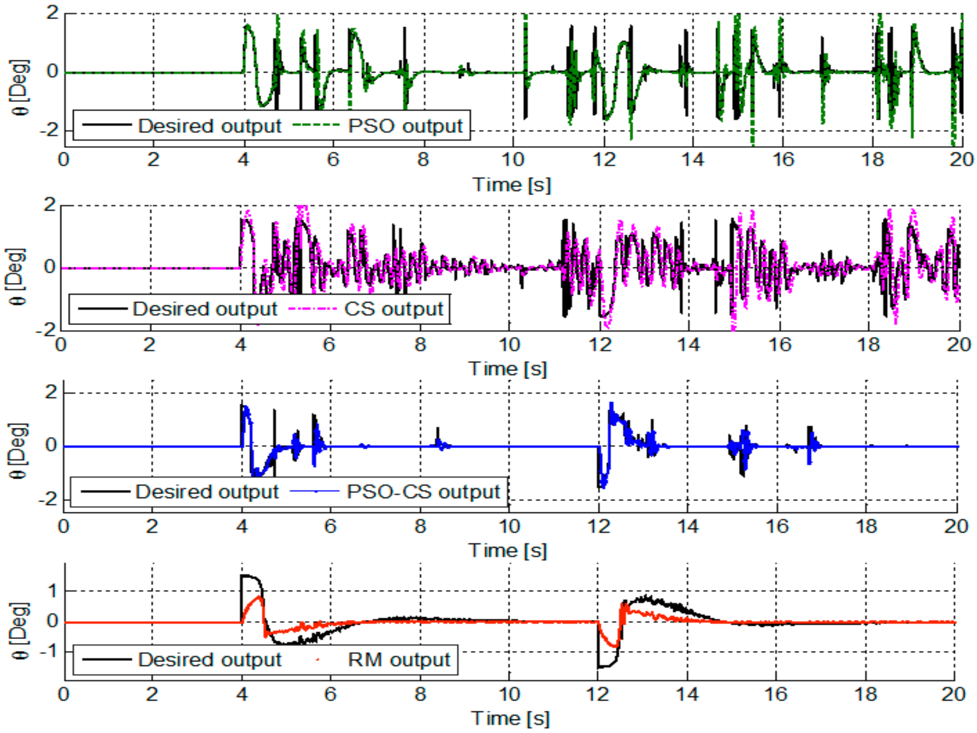

Figure 14. Quadrotor's pitch attitude (θ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) with disturbance (rectangular trajectory).

Table 6. Errors (ISE-Integral Square Error) obtained Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) for the quadrotor with disturbance (rectangular trajectory).

Method	**Errors ISE for Output**					
	x	y	z	ϕ	θ	ψ
PSO	1.254	1.161	14.212	0.524	0.982	1.056×10^{-28}
CS	1.498	1.334	21.612	0.147	2.441	7.378×10^{-35}
PSO-CS	1.079	0.685	3.287	0.183	0.195	8.157×10^{-36}
RM	2.555	1.573	33.776	0.272	0.704	5.467×10^{-35}

5.3. Helical Trajectory Tracking

Through the last simulation, we will investigate more the efficiency of the proposed intelligent control methods (PSO, CS, and PSO-CS) to search for optimal PID autotuned controllers for the quadrotor, when the desired trajectory is a helix. Simulation results of each position's tracking (x, y, z), and the tracking in 3D plan are presented in Figure 15. The tracking of roll and pitch attitudes (ϕ and θ) are presented in Figures 16 and 17. Table 7 summarizes the obtained ISE errors.

Figure 15 clearly shows the possibility of the quadrotor system to follow a linear movement in altitude z in the first mode from 0 to 18 s and to keep perfectly a stable attitude z of 20 m in the second mode from 18 to 30 s. For translation motion, the quadrotor makes a complete circle of 2 m of radius in 30 s, where we notice that the errors in x, y, and z axis are very small (from Table 7). The same figure illustrates in *3D* plan that the quadrotor tracks the helix trajectory and ends by a circle form. Also, the quadrotor avoids with high precision the overturning and maintains exactly 0° in yaw attitude.

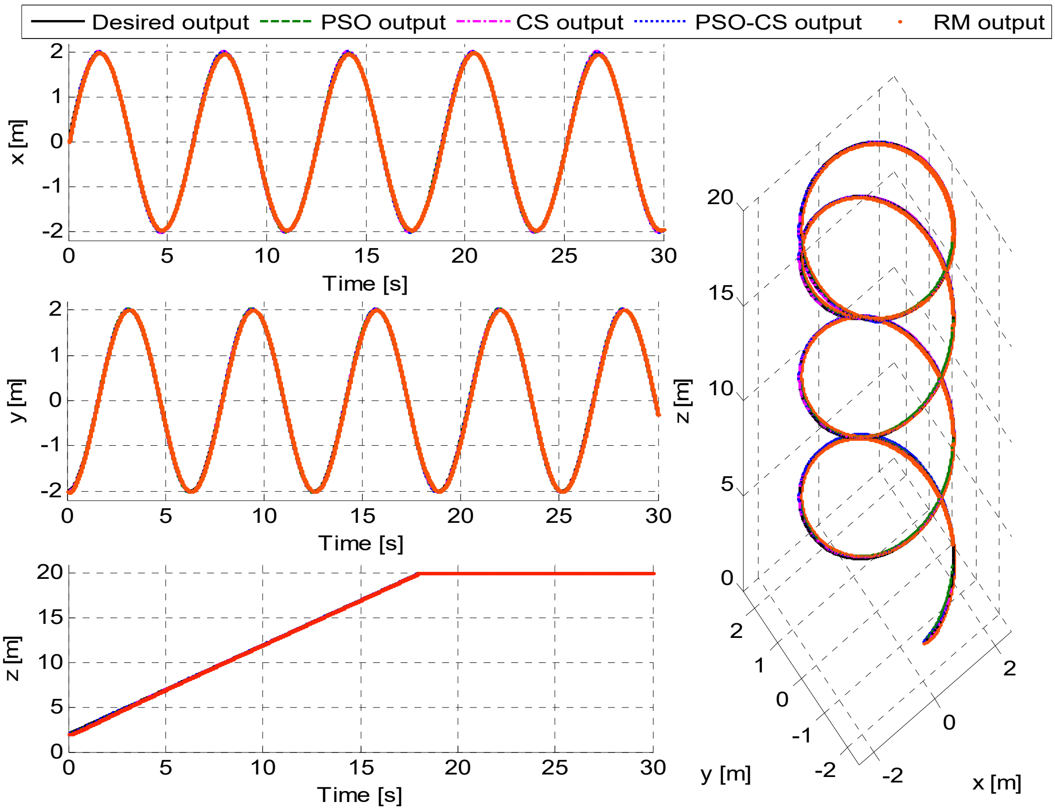

Figure 15. Quadrotor's positions (x, y, z) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) (helical trajectory).

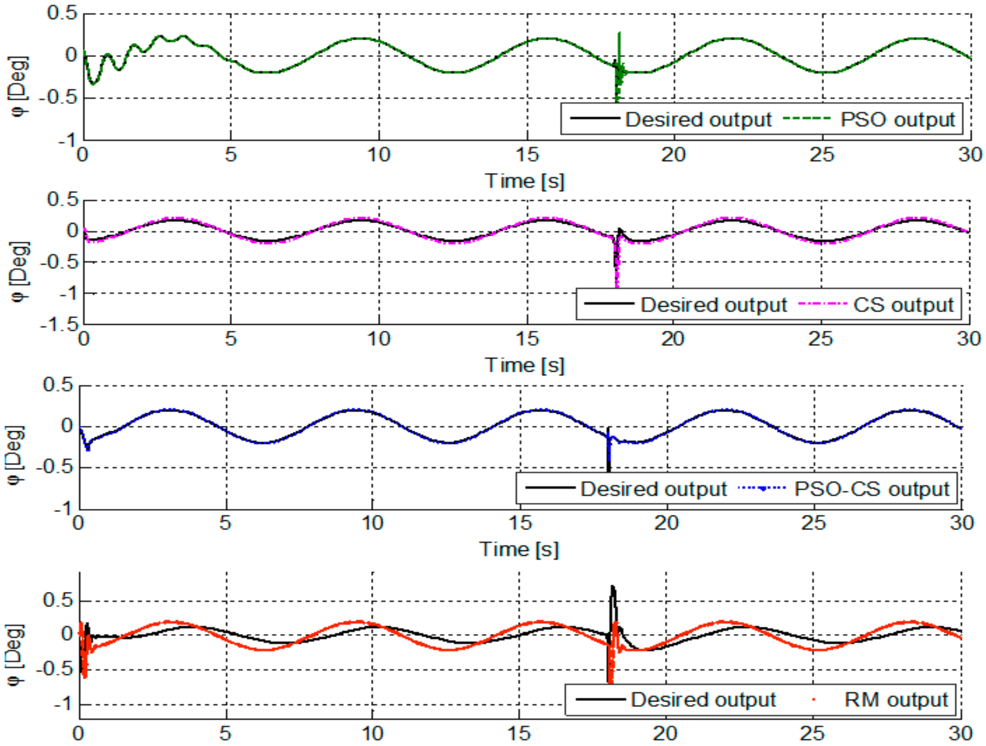

Figure 16. Quadrotor's roll attitude (ϕ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) (helical trajectory).

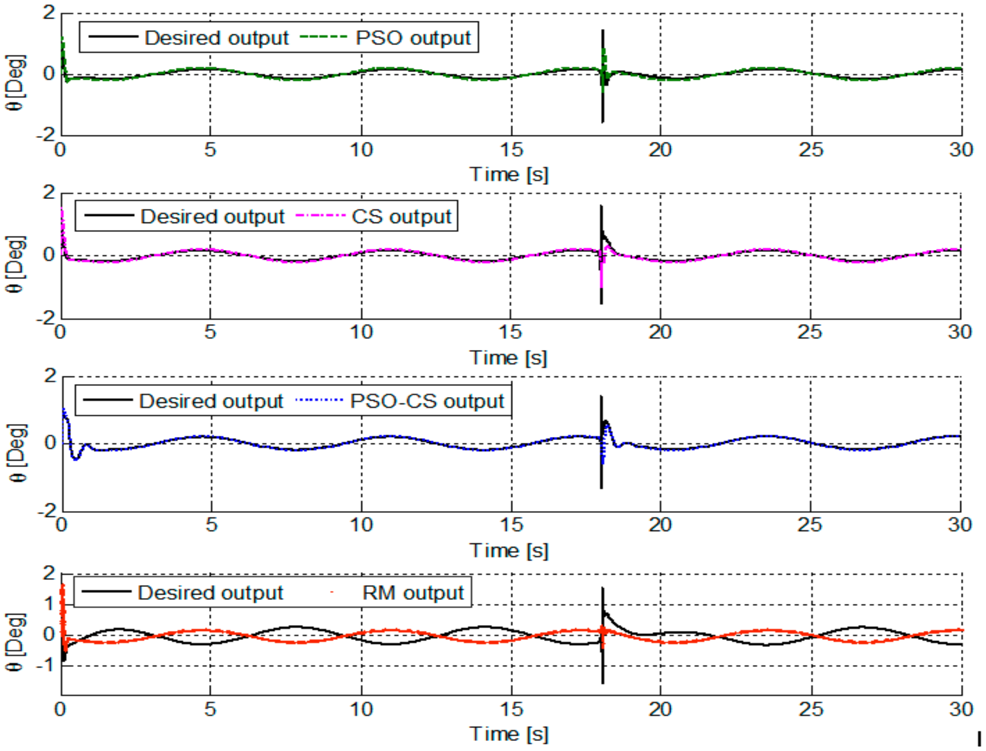

Figure 17. Quadrotor's pitch attitude (θ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) (helical trajectory).

Table 7. Errors (ISE-Integral Square Error) obtained with Particle Swarm Optimization (PSO), Cucko Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) for the quadrotor (helical trajectory).

Method	ISE Errors for Output					
	x	y	z	ϕ	θ	ψ
PSO	0.006	0.013	0.069	0.010	0.069	7.540×10^{-38}
CS	0.013	0.005	0.121	0.023	0.089	6.982×10^{-36}
PSO-CS	0.011	0.010	0.007	0.001	0.033	7.495×10^{-35}
RM	0.006	0.004	0.08	0.238	1.565	1.954×10^{-34}

Figures 16 and 17 and Table 7 illustrate that PSO, CS, and PSO-CS help to follow perfectly the sensor signals (ϕ_d and θ_d). These results prove the possibility of our system to follow the imposed combined trajectory form with precision, for all variables (positions (x and y), altitude z, and attitudes (ϕ, θ, and ψ).

These controllers are tested in order to evaluate their performances under the same disturbance (a random input of variable amplitude) inserted in the altitude z closed loop scheme of the quadrotor. It can be observed from Figures 18–20 and Table 8 that PSO and PSO-CS controllers reject the disturbance effects and drive all variables to the desired ones. Therefore, we can affirm that despite the usage of a combined trajectory form and the presence of forcible disturbance, adaptive PSO-CS and PSO are proven as remarkably robust and efficient for quadrotor control.

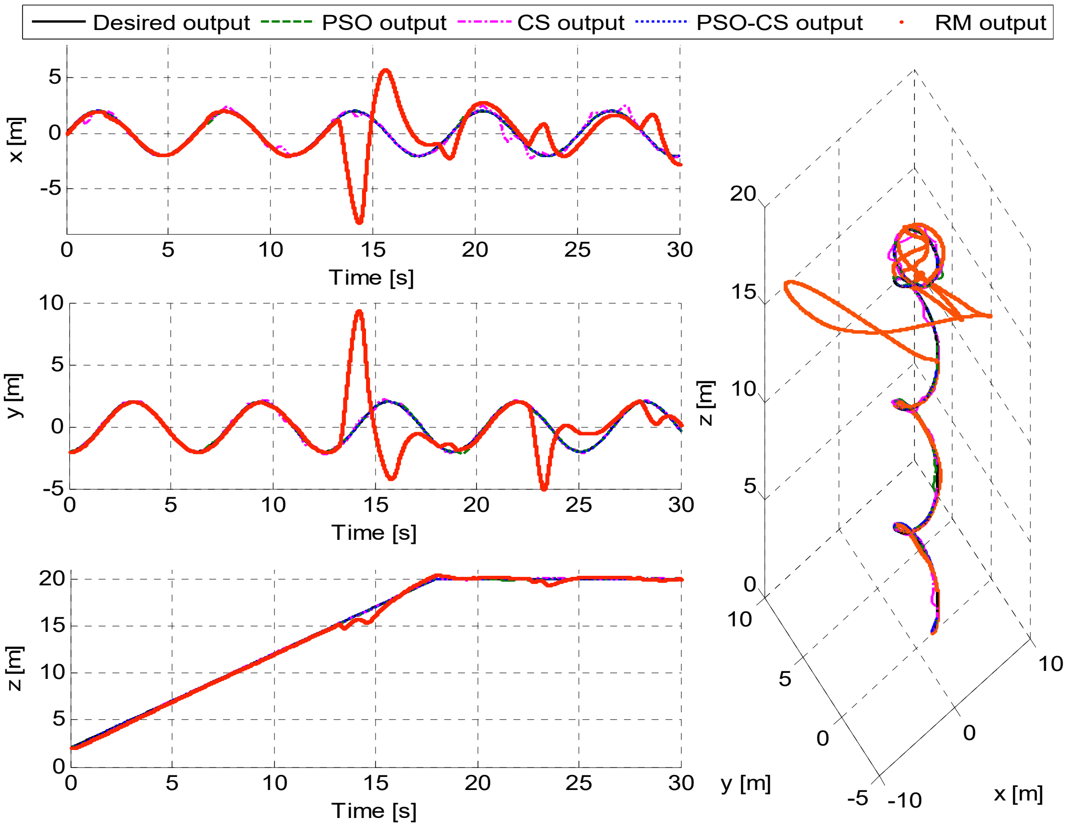

Figure 18. Quadrotor's positions (x, y, z) Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) with disturbance (helical trajectory).

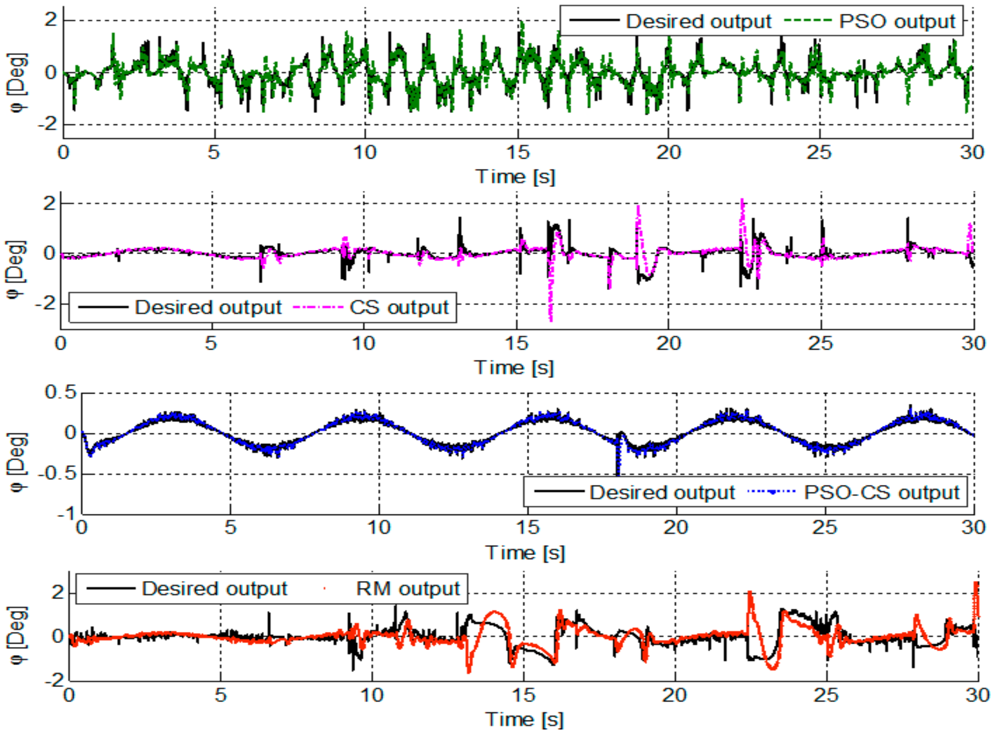

Figure 19. Quadrotor's roll attitude (ϕ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) with disturbance (helical trajectory).

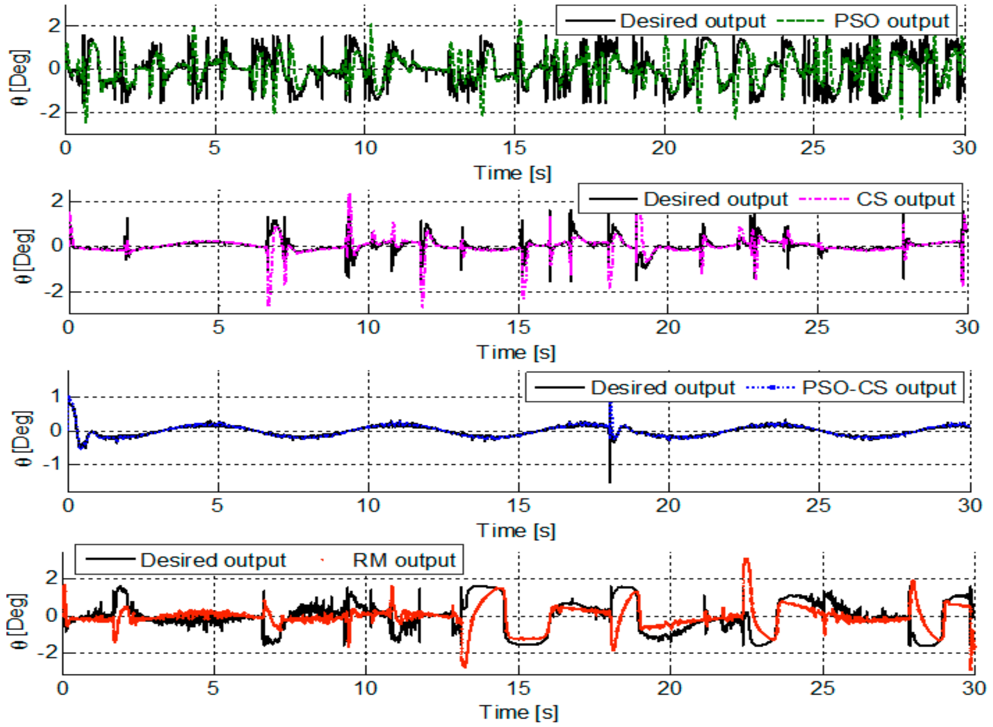

Figure 20. Quadrotor's pitch attitude (θ) using Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) with disturbance (helical trajectory).

Table 8. Errors (ISE-Integral Square Error) obtained with Particle Swarm Optimization (PSO), Cuckoo Search (CS), Particle Swarm Optimization-Cuckoo Search (PSO-CS), and Reference Model (RM) for the quadrotor with disturbance (helical trajectory).

Method	**ISE Errors for Output**					
	x	y	z	ϕ	θ	ψ
PSO	0.072	0.048	0.163	1.079	2.606	2.001×10^{-34}
CS	0.756	0.188	0.172	1.155	6.084	3.721×10^{-36}
PSO-CS	0.009	0.009	0.007	0.010	0.023	1.128×10^{-35}
RM	55.221	53.052	0.932	2.016	6.745	2.291×10^{-33}

6. Conclusions

In this paper, Particle Swarm Optimization (PSO), Cuckoo Search (CS), and the proposed cooperative PSO-CS were applied to control the quadrotor system. Results proved the efficiency of PSO approach and the cooperative PSO-CS to control optimally the quadrotor's outputs (ϕ, θ, ψ, x, y, z) compared to CS and the classical Reference Model (RM) methods. Experimental simulations also confirm the effectiveness of the aircraft using the proposed intelligent PID controllers to track the imposed rectangular and helical trajectories. Indeed, the robustness and performances of the intelligent controllers are validated in presence of disturbance. The performances of PSO-CS showed not only the stability robustness against the external disturbance, but also the fast response and the excellent tracking capacity for the different trajectories. These achievements are the results of the good settings of PSO, CS, and PSO-CS parameters. The advantages of the proposed PSO-CS in seeking for the best solutions in control problem are the results of exploiting the local search capacity of CS and benefiting from the global intelligence offered by PSO.

Concerning the limitations of our PSO-CS, one more parameter is added to CS program, which is the acceleration constant C_2. A wrong setting of this parameter can cause premature convergence. C_2 makes the cuckoos move toward the global best solution and must ensure a compromise between local and global exploration of the search space. On the other hand, the PSO-CS program consumes additional CPU time to calculate the velocity update equation over generations in order to perform the search.

Author Contributions: Data curation, N.E.; Investigation, N.E. and M.M.; Methodology, N.E. and M.M.; Ressources, N.E.; built the Simulation program and was also responsible for writing the paper. M.M.; Supervision, M.M., A.E.K., and H.A.; Validation M.M; Writing–original draft, N.E; Writing–review and editing, N.E. and M.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ACS	Adaptive Cuckoo Search
CS	Cuckoo Search
ISE	Integral Square Error
IAE	Integral Absolute Error
ITSE	Integral Time Square Error
ITAE	Integral Time Absolute Error
PI	Proportional and Integral
PID	Proportional, Integral and Derivative
PD	Proportional and Derivative
PSO	Particle Swarm Optimization
PSO-CS	Particle Swarm Optimization-Cuckoo Search
RM	Reference Model
UAV	Unmanned Aerial Vehicles

References

1. Bouabdallah, S.; Murrieri, P.; Siegwart, R. Design and control of an indoor micro quadrotor. In Proceedings of the IEEE International Conference in Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; pp. 4393–4398.
2. Bouabdallah, S. Design and Control of Quadrotors with Application to Autonomous Flying. Ph.D. Dissertation, Ecole Polytechnique Federale de Lausanne, Lausanne, Suisse, 2007.
3. Bouabdallah, S.; Noth, A.; Siegwart, R. PID vs. LQ control techniques applied to an indoor micro quadrotor. In Proceedings of the IEEE International Conference in Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; pp. 2451–2456.
4. Naidoo, Y.; Stopforth, R.; Bright, G. Quadrotor Unmanned Aerial Vehicle Helicopter Modelling & Control. *Int. J. Adv. Robot. Syst.* **2011**, *8*, 139–149.
5. Besnard, L.; Shtessel, Y.B.; Landrum, B. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *J. Frankl. Inst.* **2012**, *349*, 658–684. [CrossRef]
6. Honglei, A.; Jie, L.; Jian, W.; Jianwen, W.; Hongxu, M. Backstepping-based inverse optimal attitude control of quadrotor. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 9. [CrossRef]
7. Amin, R.; Aijun, L.; Shamshirband, S. A review of quadrotor UAV: Control methodologies and performance evaluation. *Int. J. Autom. Control* **2016**, *10*, 87–103. [CrossRef]
8. Astrom, K.J. *PID Controllers: Theory, Design and Tuning*; Instrument Society of America: Research Triangle Park, NC, USA, 1995.
9. Salih, A.L.; Moghavvemi, M.; Mohamed, H.; Gaeid, K.S. Flight PID controller design for a UAV quadrotor. *Sci. Res. Essays* **2010**, *5*, 3660–3667.
10. Junior, J.C.V.; De Paula, J.C.; Leandro, G.V.; Bonfim, M.C. Stability control of a quadrotor using a PID controller. *J. Appl. Instrum. Control* **2013**, *1*, 15–20.
11. Astrom, K.J.; Hagglund, T. Revisiting the Ziegler-Nichols step response method for PID control. *J. Process Control* **2004**, *14*, 635–650. [CrossRef]
12. Graham, D.; Lathrop, R.C. The synthesis of optimum transient response: Criteria and standard forms. *Trans. AIEE* **1953**, *72*, 273–288. [CrossRef]
13. Naslin, P. *Essentials of Optimal Control*; Iliffe & Sons Ltd.: London, UK, 1968.
14. Mjahed, M. *Control of Systems*; Royal School of Aeronautics: Marrakech, Morocco, 2016.
15. El gmili, N.; Mjahed, M.; EL Kari, A.; Ayad, H. An Improved Particle Swarm Optimization (IPSO) Approach for Identification and Control of Stable and Unstable Systems. *Int. Rev. Autom. Control* **2017**, *10*, 229–239. [CrossRef]
16. Siti, I.; Mjahed, M.; El Kari, A.; Ayad, H. New Designing Approaches for Quadcopter PID Controllers Using Reference Model and Genetic Algorithm Techniques. *Int. Rev. Autom. Control* **2017**, *10*, 240–248. [CrossRef]
17. Mjahed, M. Optimization of classification tasks by using genetic algorithms. In Proceedings of the 7th IEEE International Conference on Informatics and Systems, Cairo, Egypt, 28–30 March 2010; pp. 1–4.
18. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
19. Mohan, B.C.; Baskaran, R. A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert Syst. Appl.* **2012**, *39*, 4618–4627.
20. Karaboga, D.; Gorkemli, B.; Ozturk, C.; Karaboga, N. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **2014**, *42*, 21–57.
21. Joshi, A.S.; Kulkarni, O.; Kakandikar, G.M.; Nandedkar, V.M. Cuckoo Search Optimization—A Review. *Mater. Today Proc.* **2017**, *4*, 7262–7269. [CrossRef]
22. Kamoona, A.M.; Patra, J.C.; Stojcevski, A. An Enhanced Cuckoo Search Algorithm for Solving Optimization Problems. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6.
23. Chaine, S.; Tripathy, M. Design of an optimal SMES for automatic generation control of two-area thermal power system using cuckoo search algorithm. *J. Electr. Syst. Inf. Technol.* **2015**, *2*, 1–13. [CrossRef]
24. Zamani, A.A.; Tavakoli, S.; Etedali, S. Fractional order PID control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach. *ISA Trans.* **2017**, *67*, 222–232. [CrossRef] [PubMed]

25. Sona, D.R. PSO algorithm in quadcopter cluster for unified trajectory planning using wireless ad hoc network. In *Proceedings of the International Conference on Soft Computing Systems*; Springer: New Delhi, India, 2015; pp. 819–826.
26. Noordin, A.; Basri, M.A.M.; Mohamed, Z.; Abidin, A.F.Z. Modelling and PSO Fine-tuned PID Control of Quadrotor UAV. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2017**, *7*, 1367–1373. [CrossRef]
27. Abbas, N.H.; Sami, A.R. Tuning of PID Controllers for Quadcopter System using Hybrid Memory based Gravitational Search Algorithm–Particle Swarm Optimization. *Int. J. Comput. Appl.* **2018**, *172*, 80–99.
28. Naik, M.; Nath, M.R.; Wunnava, A.; Sahany, S.; Panda, R. A new adaptive Cuckoo search algorithm. In Proceedings of the 2nd International Conference on Recent Trends in Information Systems (ReTIS), Kolkata, India, 9–11 July 2015; pp. 1–5.
29. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [CrossRef]
30. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
31. Dréo, J.; Siarry, P. *Métaheuristiques pour L'optimisation Difficile*; Eyrolles: Paris, France, 2003.
32. El Hamidi, K.; Mjahed, M.; El Kari, A.; Ayad, H. Neural and Fuzzy Based Nonlinear Flight Control for an Unmanned Quadcopter. *Int. Rev. Autom. Control* **2018**, *11*, 98–106. [CrossRef]