

Research Article

Quadrotor Identification through the Cooperative Particle Swarm Optimization-Cuckoo Search Approach

Nada El gmili ¹, Mostafa Mjahed ², Abdeljalil El kari,¹ and Hassan Ayad¹

¹Applied Physics Department, Cadi Ayyad University, Marrakesh 40000, Morocco

²Mathematics Department, Royal School of Aeronautics, Marrakesh 40000, Morocco

Correspondence should be addressed to Nada El gmili; elgmilina@gmail.com

Received 19 May 2019; Accepted 2 July 2019; Published 24 July 2019

Academic Editor: José Alfredo Hernández-Pérez

Copyright © 2019 Nada El gmili et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper explores the model parameters estimation of a quadrotor UAV by exploiting the cooperative particle swarm optimization-cuckoo search (PSO-CS). The PSO-CS regulates the convergence velocity benefiting from the capabilities of social thinking and local search in PSO and CS. To evaluate the efficiency of the proposed methods, it is regarded as important to apply these approaches for identifying the autonomous complex and nonlinear dynamics of the quadrotor. After defining the quadrotor dynamic modelling using Newton–Euler formalism, the quadrotor model's parameters are extracted by using intelligent PSO, CS, PSO-CS, and the statistical least squares (LS) methods. Finally, simulation results prove that PSO and PSO-CS are more efficient in optimal tuning of parameters values for the quadrotor identification.

1. Introduction

Over the past few years, the unmanned aerial vehicles (UAVs) demand has increased dramatically because of the wide range of civilian and military applications. Some of those applications include low cost filming, panoramic picturing, area mapping, surveillance, air pollution monitoring, hostile zones intervention, transmission lines and power distribution inspections, earth science research assistance, etc [1].

In this paper, we are particularly interested in the behavior of one kind of UAVs that has two pairs of rotating rotors attached to the end of a cross named quadrotor or quadcopter. These aircraft are 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 two motors (the left one and the right one) rotate clockwise and the two others rotate counter clockwise, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight.

The quadrotor modelling is regarded as a delicate task either using Newton–Euler [2] or Euler–Lagrange [3] formalisms.

The model obtained using these approaches is strongly nonlinear, fully coupled, underactuated, and dynamically unstable with complex behavior. So, it is of our interest to explore an efficient model parameter estimation technique to realize precise modelling results without using complicated model structures.

By using systems identification, the quadrotor can be represented by an estimated mathematical model, based only on input and output data considering the aircraft as a black-box process. Therefore, the model's parameter values need to be estimated optimally. In the literature, some classical methods allow the process identification based on the step response such as the Strejc and Broida methods that require aperiodic systems. The integrating process allows identifying systems whose output response corresponds in steady state to a variation in a ramp. However, it is hard to model the unstable divergent behavior of the highly nonlinear quadrotor system using these classical techniques. Other methods raise the problem of identification in statistical terms of parameters estimation. The least squares (LS) method minimizes the squared error between the values predicted by the model and the observed

values. This method demonstrated its superiority in parametric identification. In [4], it is shown that the problem of parametric identification of a Wiener system could be reduced to a linear parametric estimation problem by a simple input-output data recorded using recursive least squares method (RLS).

The optimization methods of evolutionary algorithms (EAs) and swarm intelligence (SI) techniques can effectively resolve complex optimization problems compared to the described classical and statistical methods. EAs use the survival principle on a set of potential solutions to produce gradual approximations to the optimum, where SI is based on the study of group behavior in decentralized and self-organized societies. Particle swarm optimization (PSO) algorithm [5, 6] and genetic algorithm (GA) [7–9] seem to be the most successful types of EAs and SI, respectively. PSO deals with problems in which a best solution can be represented as a point or surface in a D-dimensional space. This intelligent method has shown superior performances. First, it can escape from local optimization problems. Second, it has no evolution operators such as mutation. Other advantages of PSO method are its less computational complexity compared to GA and its ease of implementation.

Cuckoo search (CS) algorithm is a novel SI algorithm motivated by the aggressive breeding of a bird called “cuckoo.” An advantage of CS compared to PSO and GA is that it uses less number of parameters to be tuned, which makes it more adaptable [10]. Also, the immigration and environmental specifications of cuckoos’ groups help to converge and reach best places for breeding and egg laying [10]. Many applications of PSO and CS are proposed in recent works of real discrete optimization problems [11–14], in identification problems [15–18], and for quadrotor’s control [19–22]. However, the CS algorithm suffers from its low convergence speed, since it uses a fixed step size over generations. Our proposed cooperative PSO-CS algorithm in [22] combines the social and local search capabilities of PSO and CS. The PSO-CS offers great guidance for cuckoos to the global best positions and ensures a balance between exploitation and exploration of the search space [22].

In this paper, we propose the intelligent PSO and CS methods, the cooperative PSO-CS, and the statistical LS to optimally identify the quadrotor’s dynamics under determined operating conditions. The model used to estimate the variations of roll (ϕ), pitch (θ), yaw (ψ) angles, and altitude z during the flying process is composed of four subsystems of second order with the same structure and whose coefficients are adjusted with PSO, CS, PSO-CS, and LS to represent as best as possible the quadrotor divergent

unstable behaviors. A comparative study is done to highlight the efficiency of the proposed intelligent methods in quadrotor identification.

This paper is organized as follows. In Section 2, Newton–Euler formalism is used to derive the motion equations of the quadrotor. In Section 3, the identification strategy is explained. In Sections 4 and 5, the quadrotor identification is detailed using the proposed intelligent PSO and CS techniques, the cooperative PSO-CS, and the statistical LS. In section 6, the simulation results are given. In the last section, our conclusions are provided.

2. Quadrotor Dynamic Modelling

The quadrotor is a complex flying machine, which is strongly nonlinear, fully coupled, and underactuated (6-DOF and only four control inputs). Therefore, its aerodynamic is affected by many physical effects, including gravity, gyroscopic effects, friction, and moment of inertia. As shown in Figure 1, the quadrotor consists of 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 two pairs of propellers must spin in opposite directions to prevent the vehicle’s overturning. In Figure 1, 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 (ϕ, θ, ψ) .

For simplifying the delicate dynamic modelling of the quadrotor, various working hypotheses have been assumed [1]: the quadrotor structure is assumed rigid and symmetrical (diagonal matrix of inertia); the propellers are assumed rigid (negligible effect of deformation during rotation); a very close approximation of the aerodynamic behavior is assumed (the lift and the drag forces are proportional to the square of the rotational speed of the rotors); and the origin of reference related to the structure is fixed on the center of mass of the quadrotor.

By using Newton–Euler formalism, the equations can be written as follows:

$$\begin{cases} \zeta = v, \\ m\ddot{\zeta} = F_f + F_t + F_g, \\ J\dot{\Omega} = M_f - M_a - M_{gh} - M_{gm}, \\ \dot{R} = RS(\Omega), \end{cases} \quad (1)$$

where v is the linear velocity and m is the total mass of the quadrotor.

R is the rotation matrix:

$$R = \begin{bmatrix} \cos \psi \cos \theta & \sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi & \cos \phi \sin \theta \cos \psi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \phi \sin \theta \sin \psi + \cos \theta \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (2)$$

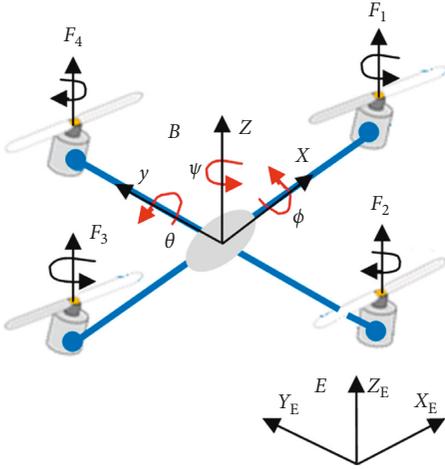


FIGURE 1: Quadrotor configuration.

Ω is the angular velocity expressed in the fixed reference:

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W \dot{\eta}. \quad (3)$$

$S(\Omega)$ denotes the oblique symmetric matrix:

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}. \quad (4)$$

J is the inertia of the system:

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}. \quad (5)$$

F_f is the total thrust force generated by the four rotors:

$$F_f = R \times \begin{bmatrix} 0 & 0 & \sum_{i=1}^4 F_i \end{bmatrix}^E = \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ \cos \phi \cos \theta \end{bmatrix} \sum_{i=1}^4 F_i, \quad (6)$$

$$F_i \approx b\omega_i^2,$$

where F_i is the force generated by the rotor i and \times means the vector product.

F_t is the drag force along the axes x , y , and z :

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fty} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} v = \begin{bmatrix} -K_{ftx} \dot{x} \\ -K_{fty} \dot{y} \\ -K_{ftz} \dot{z} \end{bmatrix}, \quad (7)$$

where K_{ftx} , K_{fty} , and K_{ftz} are the translational drag coefficients.

F_g is the gravity vector:

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}. \quad (8)$$

M_f is the moment caused by the thrust and the drag forces following the three rotations:

$$M_f = \begin{bmatrix} M_{fx} \\ M_{fy} \\ M_{fz} \end{bmatrix} = \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix}. \quad (9)$$

M_a is the moment resulting from aerodynamic friction:

$$M_a = \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix}, \quad (10)$$

where K_{fax} , K_{fay} , and K_{faz} are coefficients of aerodynamic friction.

M_{gh} is the gyroscopic moment of the propellers:

$$M_{gh} = \sum_{i=1}^4 \Omega \times J_r [0 \ 0 \ (-1)^{i+1} \omega_i] = \begin{bmatrix} J_r \Omega_r \dot{\theta} \\ -J_r \Omega_r \dot{\phi} \\ 0 \end{bmatrix}, \quad (11)$$

where J_r is the inertia of the rotors and $\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$;

M_{gm} is the gyroscopic moment due to the movements of the quadrotor:

$$M_{gm} = \Omega \times J\Omega. \quad (12)$$

Then, the dynamic model of the quadrotor can be expressed by the following equations, as given in [1]:

$$\begin{aligned} \ddot{\phi} = & \frac{lU_2}{I_x} + \frac{(I_y - I_z)}{I_x} (\dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi) \\ & \cdot (\dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta) - \frac{J_r(\omega_1 - \omega_2 + \omega_3 - \omega_4)}{I_x} \\ & \cdot (\dot{\psi} \sin \phi \cos \theta + \dot{\theta} \cos \phi) - \frac{K_{fax}}{I_x} (\dot{\phi}^2 - 2\dot{\phi}\dot{\psi} \sin \theta^2), \end{aligned} \quad (13)$$

$$\begin{aligned} \ddot{\theta} = & \frac{lU_3}{I_y} + \frac{(I_z - I_x)}{I_y} (\dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi) \\ & \cdot (\dot{\phi} - \dot{\psi} \sin \theta) - \frac{J_r(\omega_1 - \omega_2 + \omega_3 - \omega_4)}{I_y} (\dot{\psi} \sin \theta - \dot{\phi}) \\ & - \frac{K_{fay}}{I_y} (\dot{\theta}^2 \cos^2 \phi + 2\dot{\phi}\dot{\psi} \sin \phi \cos \phi \cos \theta \\ & + \dot{\psi}^2 \sin^2 \phi \cos^2 \theta), \end{aligned}$$

(14)

$$\begin{aligned} \ddot{\psi} = & \frac{U_4}{I_z} + \frac{(I_x - I_y)}{I_z} (\dot{\psi} \sin \phi \cos \theta + \dot{\theta} \cos \phi) \\ & \cdot (\dot{\phi} - \dot{\psi} \sin \theta) - \frac{J_r (\omega_1 - \omega_2 + \omega_3 - \omega_4)}{I_y} (\dot{\psi} \sin \theta - \dot{\phi}) \\ & - \left(\dot{\theta}^2 \sin^2 \phi - 2\dot{\phi}\dot{\psi} \sin \phi \cos \phi \cos \theta \right) \\ & + \frac{K_{faz}}{I_z} \dot{\psi}^2 \cos^2 \phi \cos^2 \theta, \end{aligned} \quad (15)$$

$$\ddot{x} = \frac{\cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi}{m} U_1 - \frac{K_{ftx}}{m} \dot{x}, \quad (16)$$

$$\ddot{y} = \frac{\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi}{m} U_1 - \frac{K_{fty}}{m} \dot{y}, \quad (17)$$

$$\ddot{z} = -g + \frac{\cos \phi \cos \theta}{m} U_1 - \frac{K_{ftz}}{m} \dot{z}, \quad (18)$$

with

$$\begin{cases} U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2), \\ U_2 = b(\omega_4^2 - \omega_2^2), \\ U_3 = b(\omega_3^2 - \omega_1^2), \\ U_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2). \end{cases} \quad (19)$$

Table 1 shows the definitions of the quadrotor's parameters.

3. Identification Strategy

Simulations of the quadrotor dynamical model, expressed by equations (13)–(18), with a unit step reference signal for U_1 , U_2 , U_3 , and U_4 , give unstable responses. The instability observed for the four flight variables is of divergent type, which can be simply modelled by second-order systems. In addition, the flight parameters of the quadrotor can be separated, thus giving four subsystems of order 2 having the same form $G_{mi}(p)$, with $i = \{\phi, \theta, \psi, z\}$.

$$G_{mi}(p) = \frac{K_{mi}}{(p + a_{mi})(p + b_{mi})}. \quad (20)$$

When the model of system is fixed, the identification task can be treated as an optimization problem. The basic idea of parameter estimation is to compare the time-dependent responses of the system and the model based only on inputs and outputs data [5]. Considering Figure 2, U_k are the excitation inputs and E_k are the errors of identification that characterize the difference in behavior between the quadrotor system and model, with $k = \{1, 2, 3, 4\}$. Then, PSO, CS, PSO-CS, and LS are used as adequate methods to optimally estimate the quadrotor's model parameters (K_{mi} , a_{mi} , and b_{mi}).

4. Quadrotor Identification Using Intelligent Methods

4.1. Intelligent PSO. For hard optimization problems, particle swarm optimization (PSO) was developed by Eberhart

TABLE 1: The parameters of the quadrotor.

Parameter	Definition	Value
I_x (kg·m ²)	The inertia moments with respect to the axes	$7.5 \cdot 10^{-3}$
I_y (kg·m ²)		$7.5 \cdot 10^{-3}$
I_z (kg·m ²)		$1.3 \cdot 10^{-3}$
K_{fax} (N/rad/s)	The aerodynamic coefficients	$5.567 \cdot 10^{-4}$
K_{fay} (N/rad/s)		$5.567 \cdot 10^{-4}$
K_{faz} (N/rad/s)		$6.354 \cdot 10^{-4}$
K_{ftx} (N/rad/s)	The drag coefficients	$5.567 \cdot 10^{-4}$
K_{fty} (N/rad/s)		$5.567 \cdot 10^{-4}$
K_{ftz} (N/rad/s)		$6.354 \cdot 10^{-4}$
g (m/s)	The gravity acceleration	9.806
J_r (kg·m ²)	The inertia of the rotors	$2.8385 \cdot 10^{-5}$
b (kg·m·rad ⁻²)	The thrust coefficient	$2.9842 \cdot 10^{-5}$
d (kg·m·rad ⁻²)	The drag coefficient	$3.232 \cdot 10^{-6}$
m (kg)	The total mass of the quadrotor	0.65
l (m)	The half span of the quadrotor	0.23

and Kennedy in 1995. The basic principle of PSO was inspired by the social behavior of animals moving in a swarm as bird flocking. To search for food, each bird flies in the space of solutions and determines its speed according to its personal experience and the information obtained through interaction with other swarm members [4].

The initialization matrix contains N particles dispersed in a search space along dimension j for $j = \{1, 2, \dots, D\}$. Each particle P_i ($i = 1, 2, \dots, N$) 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 (22). The mechanism of displacement of each particle is managed by three rules. Firstly, the particle tends to follow the direction of its current velocity. Secondly, it tends to move towards its best position. Finally, it tends to move to the best position reached by its neighbors [5, 23]. In fact, the new velocity matrix V_{ij} and position matrix X_{ij} of particles are calculated at iteration $(t+1)$, according to equations (23) and (24):

$$Pb_i(t+1) = \begin{cases} X_i(t+1), & \text{if } f(X_i(t+1)) < f(Pb_i(t)), \\ Pb_i(t), & \text{else,} \end{cases} \quad (21)$$

$$Pg = \min_{i=1,2,\dots,N} f(Pb_i(t)), \quad (22)$$

$$\begin{aligned} V_{ij}(t+1) = & w \cdot V_{ij}(t) + R_1 C_1 \otimes (Pb_{ij}(t) - X_{ij}(t)) \\ & + R_2 C_2 \otimes (Pg_{ij}(t) - X_{ij}(t)), \end{aligned} \quad (23)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1), \quad (24)$$

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; and R_1 and R_2 are random variables generated from a uniform distribution in $[0, 1]$.

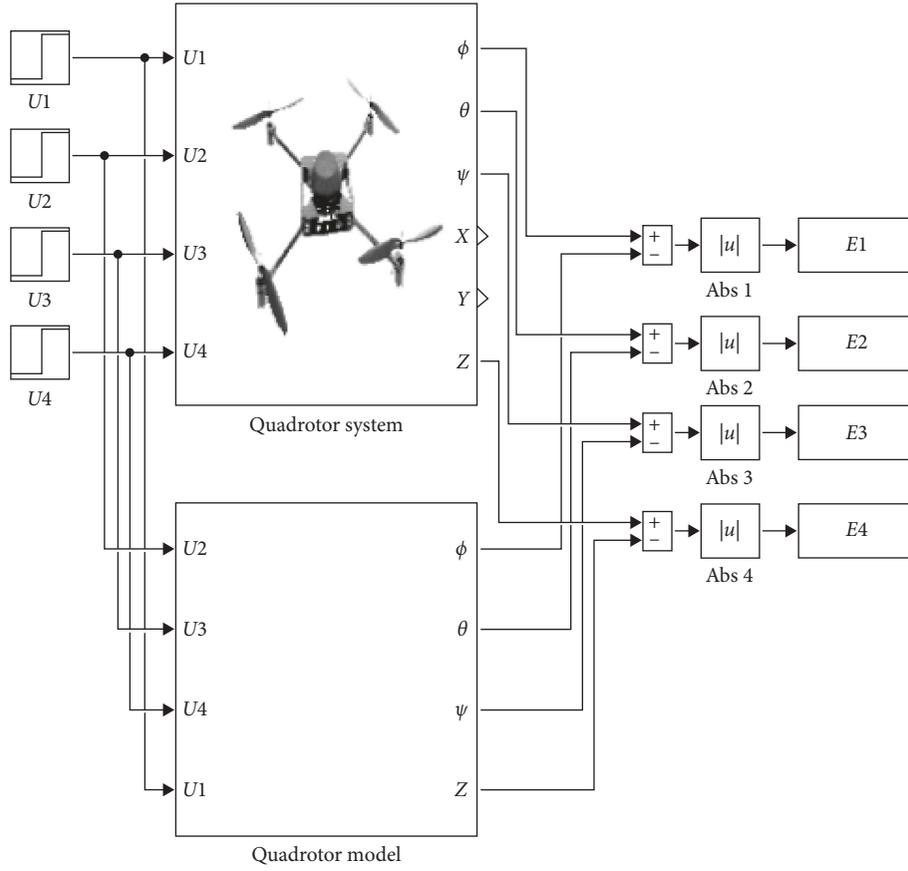


FIGURE 2: Quadrotor identification scheme.

4.2. *Intelligent CS.* Cuckoo search (CS) algorithm, proposed by Yang and Deb in 2009, is based on the life of a “cuckoo” bird. The basic principle of CS 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 are not discovered and removed by host birds. Reproduction and breeding are favoured by cuckoo groups’ immigration, converging and reaching the best places [10].

The primary population of CS contains N nests, and each nest is composed of D eggs. The best nests with a high quality of eggs (solutions) carry over to the next generations, where the quality evaluation is based on the fitness function F of the habitat (array of $1 \times D$). The host can discover an alien egg with probability P_a from $[0, 1]$, which is approximated by a fraction P_a of the N nests being replaced by new nests, having new random solutions [24].

To explore the search space when replacing solutions in the nests with new solutions, *Lévy* flight mechanism is used. The step length S from Mantegna algorithm (based on Gaussian normal distribution denoted by Norm) can be written as represented by equation (25), and σ_u^2 is the variance of the distributions given by equation (26). Therefore, a new solution $X_i(t+1)$ for cuckoo i is given by equation (27), and the fraction P_a of worse solutions is generated as given by equation (28):

$$\text{Lévy}(\lambda) \approx S = (\text{Norm}(0, \sigma_u^2)) |\text{Norm}(0, 1)|^{-1/\lambda}, \quad (1 < \lambda \leq 3), \quad (25)$$

$$\sigma_u^2 = \left\{ \frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma[(1+\lambda)/2]\lambda 2^{(\lambda-1)/2}} \right\}^{1/\lambda}, \quad (26)$$

$$X_i(t+1) = X_i(t) + \alpha \otimes \text{Lévy}(\lambda), \quad (27)$$

$$X_i(t+1) = X_i(t) + (P_a - r)(X_j(t) - X_k(t)), \quad (28)$$

where Γ is the gamma function; $\alpha > 0$ is the step size; R and r are random variable generated from a uniform distribution in the interval $[0, 1]$; $X_j(t)$ and $X_k(t)$ are two random solutions chosen by random permutation; and H is a Heaviside function.

4.3. *Cooperative PSO-CS.* The initialization matrix of the cooperative PSO-CS is of dimension $D \times N$, and the solutions’ quality is evaluated as in PSO and CS. The global best particle (or the best nest) is the particle (or the nest) that has the smallest fitness value among all potential solutions [22].

To overcome the fast convergence speed of PSO and the low convergence speed of CS, the PSO-CS combines the capacities of social thinking in PSO and local search in CS. Thus, the displacement equation is modified, by combining the *Lévy* flight random walks of cuckoos and the velocity of particles toward the global best solution Pg . New solutions $X_i(t+1)$ are given by [22]

$$X_i(t+1) = X_i(t) + \alpha \otimes \text{Lévy}(\lambda) + R_2 C_2 \otimes (Pg(t) - X_i(t)). \quad (29)$$

Both algorithms' capabilities are combined to increase the particles' diversification. PSO-CS guides the cuckoos toward the global best positions (global intelligence of the swarm). In fact, the search ability increases during iterations, and the exploration of the local and global places is achieved by the Lévy flight displacement of cuckoos and the velocity of particles toward the global best solution (Pg), as given by equation (29). The flow chart of PSO-CS is considered the same as that of CS. It is given in Figure 3. The only modification is in the expression of the velocity of displacement, which helps to search at local and global scales in order to move all cuckoos toward best environment and to quickly converge at later stage.

4.4. Parameter Setting for PSO, CS, and PSO-CS. Intelligent PSO, CS, and PSO-CS are applied to optimally select the model parameters from $N=200$ solutions. So, the search space interval is chosen sufficient to contain all possible solutions $[0, 200]$, and its dimension D is set to 12. The compromise between local and global exploration in PSO is achieved for $w=0.8$. C_1 takes random values in $[0, 0.8]$ to avoid the problem of fast convergence, while C_2 takes random values in $[0, 1.2]$ to give more importance to the global best solution Pg [22]. While in CS, the parameters used in experiments are as follows: abandon probability, $P_a=0.25$, and the Lévy flight settings, $\alpha=0.1$ and $\lambda=1.5$ [22]. In PSO-CS, the same settings of PSO and CS are kept [22]: $P_a=0.25$ and the Lévy flight settings $\alpha=0.1$, $\lambda=1.5$, and C_2 with random values in $[0, 1.2]$.

These algorithms are evaluated using a profit defined in a similar way in order to minimize the differences between the output responses of the estimated model and the quadrotor system. This fitness function is defined in equation (30) as the sum of the quadratic errors E_k previously mentioned, with $l=4$ and $k=\{1, 2, 3, 4\}$:

$$F = \frac{1}{2} \sum_{k=1}^l E_k^2. \quad (30)$$

An appropriate set of PSO, CS, and PSO-CS parameters can yield model responses close to those of the quadrotor. The maximum number of generations for the three programs (PSO, CS, and PSO-CS) is fixed as the stop criterion and set to 20.

5. Quadrotor Identification Using the Statistical LS Method

The method of least squares (LS) provides the parameters of a model so that the sum of squared errors (between the predicted and observed values) is minimal [25].

For the quadrotor's identification, we consider the same identification scheme represented in Figure 2. The vectors of measurements are extracted from the temporal responses of the quadrotor's attitude (ϕ, θ, ψ) and altitude z .

The recurring equation in equation (31) is obtained by discretizing at T_e the temporal response of each transfer

function $G_{mi}(p)$ (in equation (20)), where u_k and y_k are, respectively, discrete inputs and outputs of $G_{mi}(p)$:

$$y(k+2) = -a_0 y(k) - a_1 y(k+1) + b_0 u(k) + b_1 u(k+1). \quad (31)$$

The a_j and b_k coefficients or θ (equation (32)) to be estimated, in the sense of the LS criterion given in equation (33), is computed following equation (34). The matrix F and the vector Y are constituted from the values taken by u and y at the different sampling times:

$$\hat{\theta} = \begin{pmatrix} a_0 \\ a_1 \\ b_0 \\ b_1 \end{pmatrix}, \quad (32)$$

$$J(\hat{\theta}) = \frac{1}{2} \sum (\varepsilon(i))^2, \quad (33)$$

$$\hat{\theta} = (F^T F)^{-1} (F^T Y), \quad (34)$$

$$F = \begin{pmatrix} -y_0 & -y_1 & u_0 & u_1 \\ -y_1 & -y_2 & u_1 & u_2 \\ & & \vdots & \\ -y_{N-2} & -y_{N-1} & u_{N-2} & u_{N-1} \end{pmatrix}, \quad (35)$$

$$Y = \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix}.$$

The transition between continuous (k_{mp}, a_{mp}, b_m) and discrete parameters (a_0, a_1, b_0, b_1) is grouped in the following equations:

$$\begin{cases} a_0 = -e^{-(a_m+b_m)T_e}, \\ a_1 = -e^{-b_m T_e} - e^{-a_m T_e}, \\ b_0 = \frac{K_m}{a_m b_m} \left[e^{-(a_m+b_m)T_e} + \frac{b_m}{a_m - b_m} e^{-b_m T_e} + \frac{a_m}{a_m - b_m} e^{-a_m T_e} + 1 \right], \\ b_1 = \frac{K_m}{a_m b_m} \left[-e^{-b_m T_e} - e^{-a_m T_e} - \frac{b_m}{a_m - b_m} e^{-b_m T_e} - \frac{a_m}{a_m - b_m} e^{-a_m T_e} \right], \end{cases} \quad (36)$$

$$\begin{cases} K_{mi} = \frac{a_{mi} b_{mi} (b_0 + b_1)}{1 + a_0 + a_1}, \\ a_{mi} = -\frac{1}{T_e} \log \left(\frac{-a_0 + \sqrt{a_0^2 - a_1}}{2} \right), \\ b_{mi} = -\frac{1}{T_e} \log(a_1) - a_{mi}. \end{cases} \quad (37)$$

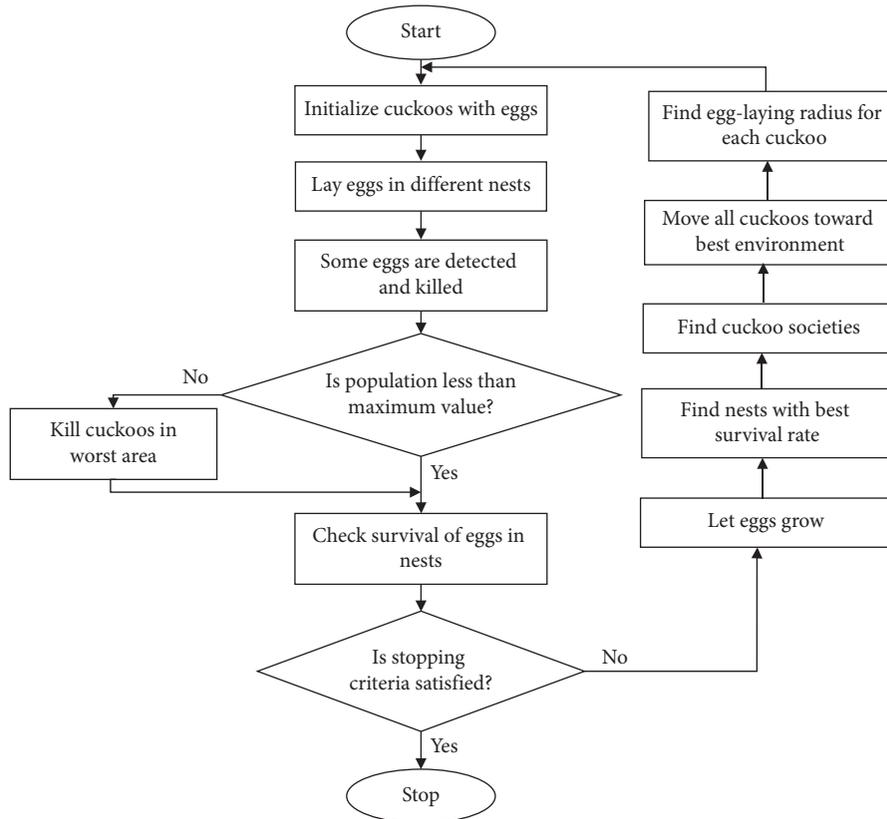


FIGURE 3: Flow chart of CS and PSO-CS.

The application of this approach to the identification of the quadrotor's movements is discussed in the following section.

6. Simulation Results

The quadrotor system is identified using a model composed of four subsystems with the same structure, expressed by $G_{mi}(p)$ in equation (20). The quadrotor model is generated in a simulation time of 500 seconds. Table 2 summarizes the parameters of these subsystems (K_{mi} , a_{mi} , b_{mi}) estimated by using PSO, CS, PSO-CS, and LS methods, with $i = \{\phi, \theta, \psi, z\}$. Figures 4–7 show the responses of the quadrotor's system and models with parameters obtained using PSO, CS, PSO-CS, and LS, in an observation time of 500 seconds. Table 3 shows the results of statistical analysis based on the integral of a positive term involving the error: the integral absolute error (IAE), the integral square error (ISE), the integral time absolute error (ITAE), and the integral time square error (ITSE) [6].

PSO and PSO-CS perform the search with high precision to avoid all solutions that are far from the desired responses of the quadrotor system (angles of roll, pitch, yaw, and altitude z). From Figures 4–7, the model responses with parameters adjusted using PSO and PSO-CS are very close to those of the quadrotor's responses, while CS and LS give the estimated model parameters that allow fitting a divergent behavior near to that of the quadrotor system.

These observations spread to 500 seconds prove the validity of the chosen model to identify the divergent responses of the quadrotor, as well as the efficiency of PSO and

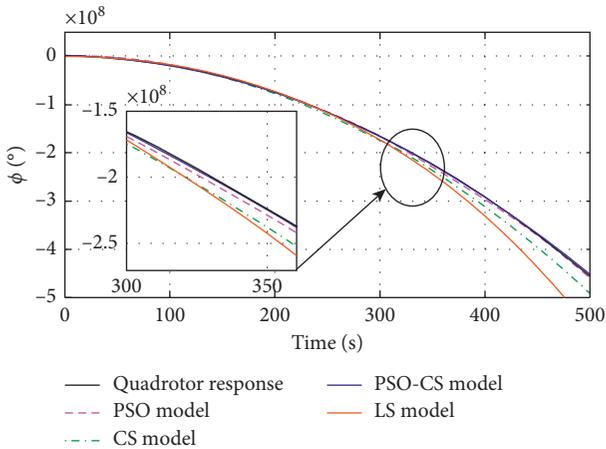
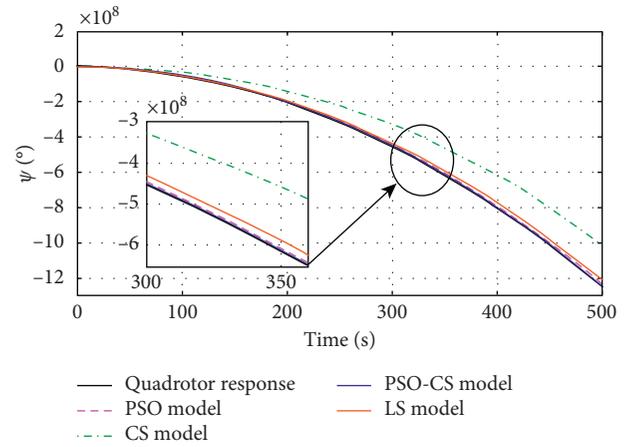
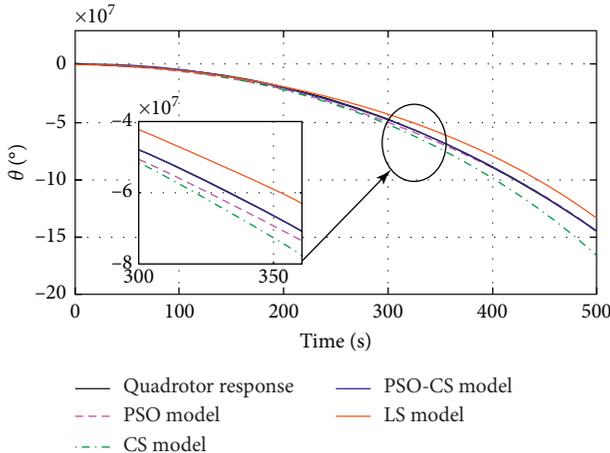
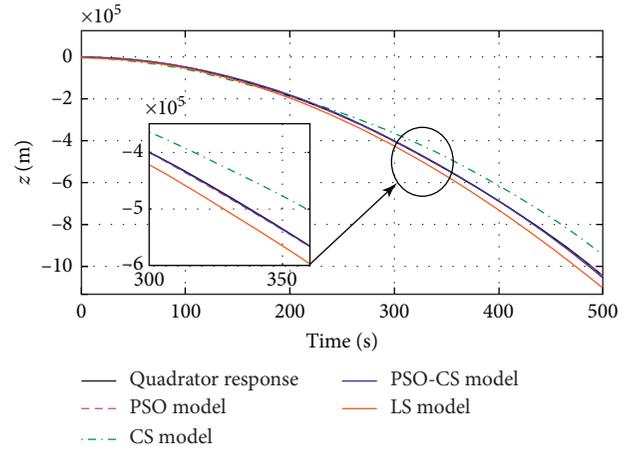
PSO-CS for seeking for the best solution, even if identifying unstable responses is a very difficult task. During the simulation time, PSO and PSO-CS effectively avoid the divergence that can appear between the model (20) with the calculated parameters and the quadrotor system defined by equations (13)–(18). PSO and PSO-CS reduce the fitness function and result in an optimal search for the parameters of the models. To confirm the effectiveness of PSO and PSO-CS in reducing the errors between the two responses for the four outputs, we present in the following table a statistical analysis based on the integral of the errors IAE, ISE, ITAE, and ITSE.

Statistical results in Table 3 show that the errors between the model outputs and the system outputs, such as IAE, ISE, ITAE, and ITSE are all smaller (higher performance) when the model is established by PSO-CS than PSO, CS, and LS.

In [26], the authors proposed a modified CS algorithm named oriented cuckoo search algorithm (OCS), where they have tested its performance for different probability distributions (DDICS1, DDICS2, MCS1, MCS2, DACS1, DACS2, and OCS-LG). The proposed oriented cuckoo search algorithm with Lévy distribution and standard Gaussian distribution (OCS-LG) has shown better performance in means of mean error. The lower value means the better performance; therefore, OCS-LG is superior to other six algorithms. From the results of Wilcoxon test, there are significant differences (p value below 0.05) among OCS-LG, DACS1, MCS1, and MCS2. In other words, OCS-LG is significantly better than DACS1, MCS1, and MCS2.

TABLE 2: Model's parameters given by PSO, CS, PSO-CS, and LS methods.

Optimization method	Parameter	$G_{m\phi}(p)$	$G_{m\theta}(p)$	$G_{m\psi}(p)$	$G_{mz}(p)$
PSO	K_{mi}	-3990.4	-1093.62	-9890.68	-9.95
	a_{mi}	-0.0009	0.0002	0.0001	-0.0001
	b_{mi}	0.0016	-0.0005	-0.0001	0.0012
CS	K_{mi}	-3850.82	-940.99	-6240	-15.589
	a_{mi}	-0.001	-0.002	-0.0016	-0.003
	b_{mi}	0.001	0.0001	$1.260 \cdot 10^{-4}$	0.015
PSO-CS	K_{mi}	-3880	-935.561	-9973.1	-9.6921
	a_{mi}	0.0009	-0.0011	$-2.812 \cdot 10^{-4}$	$-8.745 \cdot 10^{-4}$
	b_{mi}	-0.0004	$1.767 \cdot 10^{-4}$	$2.845 \cdot 10^{-4}$	0.0019
LS	K_{mi}	-3336	-908.74	-9447.8	-10.31
	a_{mi}	-0.0033	-0.0032	$-8.218 \cdot 10^{-4}$	$-2.639 \cdot 10^{-5}$
	b_{mi}	0.0026	0.0037	$7.657 \cdot 10^{-4}$	0.0010

FIGURE 4: Roll (ϕ) responses of the quadrotor and the model $G_{m\phi}(p)$, obtained using PSO, CS, PSO-CS, and LS.FIGURE 6: Yaw (ψ) responses of the quadrotor and the model $G_{m\psi}(p)$, obtained using PSO, CS, PSO-CS, and LS.FIGURE 5: Pitch (θ) responses of the quadrotor and the model $G_{m\theta}(p)$, obtained using PSO, CS, PSO-CS, and LS.FIGURE 7: Altitude (z) responses of the quadrotor and the model $G_{mz}(p)$, obtained using PSO, CS, PSO-CS, and LS.

The proposed hybrid many-objective cuckoo search (HMaOCS) for many-objective optimization problems (MaOPs) in [27] show that HMaOCS is promising in dealing with most many-objective optimization problems. It was noted that in CS, there is no guidance information (the

global best individual) and each individual in the population is not affected by any other individuals. Then, the proposed changes were focused in the population updating method, where the new population was generated from the combined parent and offspring populations to ensure better

TABLE 3: Statistical analysis of predicted errors using PSO, CS, and PSO-CS models.

Predictive output	IAE	ISE	ITAE	ITSE	
ϕ	PSO model	2.3482	0.8960	8.3712	34.6600
	CS model	9.4593	2.0600	38.0080	899.4400
	PSO-CS model	0.6122	0.1318	2.4165	6.1527
	LS model	18.775	11.127	79.059	5090.6
θ	PSO model	1.8351	$7.75 \cdot 10^{-2}$	7.2906	33.9510
	CS model	5.0279	0.5808	19.9930	256.4700
	PSO-CS model	1.6507	0.0389	4.6952	11.1720
	LS model	4.6466	0.4001	17.590	165.32
ψ	PSO model	3.6325	0.2550	13.9370	107.6700
	CS model	99.0120	155.2600	355.9000	61671
	PSO-CS model	0.1661	$3.36 \cdot 10^{-2}$	0.4550	$9.87 \cdot 10^{-2}$
	LS model	16.216	4.3374	59.174	1748
z	PSO model	$2.6 \cdot 10^{-3}$	$2 \cdot 10^{-7}$	$1.04 \cdot 10^{-2}$	$9.2040 \cdot 10^{-5}$
	CS model	$3.56 \cdot 10^{-2}$	$2.51 \cdot 10^{-5}$	0.1363	$1.0552 \cdot 10^{-2}$
	PSO-CS model	$1 \cdot 10^{-3}$	$1 \cdot 10^{-8}$	$2.8 \cdot 10^{-3}$	$4.15 \cdot 10^{-6}$
	LS model	$1.93 \cdot 10^{-2}$	$6.58 \cdot 10^{-6}$	$7.195 \cdot 10^{-2}$	$2.7307 \cdot 10^{-3}$

convergence and diversity. However, the computation time necessary to generate new solutions in this way was not considered.

Comparisons to OCS-LG and HMaOPs show that our proposed PSO-CS algorithm used Lévy flight mechanism with Gaussian normal distribution when replacing solutions in the nests with new solutions, which is the best solution attested in [26]. In addition, PSO-CS benefits from the global intelligence already calculated in each iteration in order to manage the displacements of solutions.

Our proposed programs are performed on a computer with a processor Intel® Core™ i7-3770 CPU of 3.40 GHz and 8.0 GB of RAM using Matlab R2016a. The implemented PSO-CS program consumes an additional CPU time (6 min 15 s) than PSO and CS programs that spend (5 min 07 s) and (5 min 50 s), respectively. This additional simulation time of PSO-CS is produced due to the sum of both matrices generating Lévy flight random walks of cuckoos ($X_i(t) + \alpha \otimes \text{Lévy}(\lambda)$) and the movement of particles toward the global best solution $R_2 C_2 \otimes (Pg(t) - X_i(t))$. This updating method is faster than generating solutions from the combined parent using selection and crossover operators.

7. Conclusions

In this paper, recent heuristics optimization of particle swarm optimization (PSO) and cuckoo search (CS), the proposed cooperative particle swarm optimization-cuckoo search (PSO-CS), and the statistical least squares (LS) has been applied to identify the variations of the rotational movements (ϕ , θ , and ψ angles) and the translational movement along z -axis of a quadrotor UAV. Simulation results proved the efficiency of the PSO approach and the cooperative PSO-CS in identifying optimally the quadrotor's outputs (ϕ , θ , ψ , z) compared to CS and LS methods. The proposed PSO-CS seeks for the best solution by exploiting the local search capacity of CS and benefiting from the global intelligence offered by PSO.

These achievements are the results of the good choice of the model structure to identify the quadrotor system

responses with unstable behaviors and the good adjustment of PSO, CS, and PSO-CS parameters (the fitness function, the weighting coefficients w , C_1 , and C_2 , the probability P_a , and the Lévy flight settings for α and λ). However, PSO-CS program consumes an additional CPU time to calculate the velocity update equation, and a wrong setting of the acceleration constant C_2 can cause premature convergence.

Data Availability

No data were used to support this study.

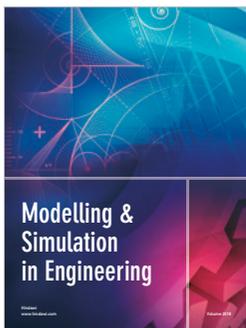
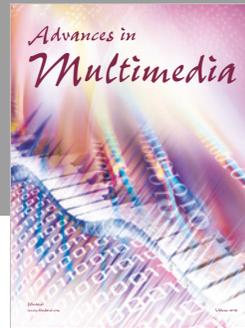
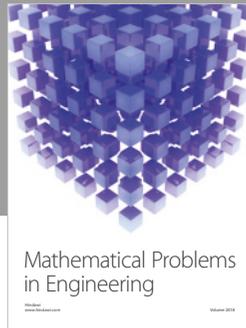
Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] R. Lozano, *Unmanned Aerial Vehicles: Embedded Control*, John Wiley & Sons, Hoboken, NJ, USA, 2013.
- [2] B. Samir, "Design and control of quadrotors with application to autonomous flying," Ecole Polytechnique federale de Lausanne, Lausanne, Switzerland, Thèse, 2007.
- [3] Y. Naidoo, R. Stopforth, and G. Bright, "Quad-rotor unmanned aerial vehicle helicopter modelling & control," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, p. 45, 2011.
- [4] R. Pupeikis, "On recursive parametric identification of Wiener systems," *Information Technology and Control*, vol. 40, no. 1, pp. 21–28, 2011.
- [5] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November–December 1995.
- [6] N. El Gmili, M. Mjahed, A. el Kari, and H. Ayad, "An improved particle swarm optimization (IPSO) approach for identification and control of stable and unstable systems," *International Review of Automatic Control (IREACO)*, vol. 10, no. 3, pp. 229–239, 2017.
- [7] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2-3, pp. 95–99, 1988.

- [8] M. Mjahed, "Optimization of classification tasks by using genetic algorithms," in *Proceedings of the 2010 the 7th International Conference on Informatics and Systems (INFOS)*, pp. 1–4, IEEE, Cairo, Egypt, March 2010.
- [9] M. Mjahed, "PID controller design using genetic algorithm technique," in *Proceedings of the ICICR*, pp. 1–4, Istanbul, Turkey, 2013.
- [10] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo search optimization- a review," *Materials Today: Proceedings*, vol. 4, no. 8, pp. 7262–7269, 2017.
- [11] W. Guo, G. Liu, G. Chen, and S. Peng, "A hybrid multi-objective PSO algorithm with local search strategy for VLSI partitioning," *Frontiers of Computer Science*, vol. 8, no. 2, pp. 203–216, 2014.
- [12] W. Guo, J. Li, G. Chen et al., "A PSO-optimized real-time fault-tolerant task allocation algorithm in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3236–3249, 2014.
- [13] G. Liu, X. Huang, W. Guo et al., "Multilayer obstacle-avoiding X-architecture Steiner minimal tree construction based on particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1003–1016, 2014.
- [14] G. Liu, W. Guo, R. Li, Y. Niu, and G. Chen, "XGRouter: high-quality global router in X-architecture with particle swarm optimization," *Frontiers of Computer Science*, vol. 9, no. 4, pp. 576–594, 2015.
- [15] A. D. Jeraldin, "Adaptive Particle Swarm Optimization based system identification and internal model sliding mode controller for mass flow system," *Journal of Control Engineering and Applied Informatics*, vol. 17, no. 4, pp. 3–13, 2015.
- [16] J. Gośliński, S. Gardecki, and W. Giernacki, "An efficient PSO-based method for an identification of a Quadrotor model parameters," in *Progress in Automation, Robotics and Measuring Techniques*, pp. 95–104, Springer, Cham, Switzerland, 2015.
- [17] A. P. Patwardhan, R. Patidar, and N. V. George, "On a cuckoo search optimization approach towards feedback system identification," *Digital Signal Processing*, vol. 32, pp. 156–163, 2014.
- [18] A. Gotmare, R. Patidar, and N. V. George, "Nonlinear system identification using a cuckoo search optimized adaptive Hammerstein model," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2538–2546, 2015.
- [19] A. Noordin, M. A. Mohd Basri, Z. Mohamed, and A. F. Zainal Abidin, "Modelling and PSO fine-tuned PID control of quadrotor UAV," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 4, pp. 1367–1373, 2017.
- [20] N. H. Abbas and A. R. Sami, "Tuning of PID controllers for quadcopter system using cultural exchange imperialist competitive algorithm," *Journal of Engineering*, vol. 24, no. 2, pp. 80–99, 2018.
- [21] N. E. Gmili, M. Mjahed, A. E. Kari, and H. Ayad, "Intelligent PSO-based PDs/PIDs controllers for an unmanned quadrotor," *International Journal of Intelligent Engineering Informatics*, vol. 6, no. 6, pp. 548–568, 2018.
- [22] N. El Gmili, M. Mjahed, A. El Kari, and H. Ayad, "Particle swarm optimization and cuckoo search-based approaches for quadrotor control and trajectory tracking," *Applied Sciences*, vol. 9, no. 8, p. 1719, 2019.
- [23] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.
- [24] A. M. Kamoona, J. C. Patra, and A. Stojcevski, "An enhanced cuckoo search algorithm for solving optimization problems," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–6, IEEE, Rio de Janeiro, Brazil, July 2018.
- [25] V. Verdult and M. Verhaegen, *Filtering and System Identification: A Least Squares Approach*, Cambridge University Press, Cambridge, UK, 2007.
- [26] Z. Cui, B. Sun, G. Wang, Y. Xue, and J. Chen, "A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems," *Journal of Parallel and Distributed Computing*, vol. 103, pp. 42–52, 2017.
- [27] Z. Cui, M. Zhang, H. Wang et al., "A hybrid many-objective cuckoo search algorithm," *Soft Computing*, pp. 1–17, 2019.



Hindawi

Submit your manuscripts at
www.hindawi.com

