Computers, Materials & Continua
DOI:10.32604/cmc.2021.015681
Article

Tech Science Press

Control Strategy for a Quadrotor Based on a Memetic Shuffled Frog Leaping Algorithm

Nour Ben Ammar[1], **Hegazy Rezk**[2,3,*] **and Soufiene Bouallègue**[1,4]

[1]Research Laboratory in Automatic Control (LARA), National Engineering School of Tunis (ENIT), University of Tunis EL MANAR, Le Belvédère, Tunis, 1002, Tunisia
[2]College of Engineering at Wadi Addawaser, Prince Sattam Bin Abdulaziz University, Al-Kharj, 11911, Saudi Arabia
[3]Department of Electrical Engineering, Faculty of Engineering, Minia University, Minia, 61517, Egypt
[4]High Institute of Industrial Systems of Gabès (ISSIG), University of Gabès, Gabès, 6011, Tunisia
[*]Corresponding Author: Hegazy Rezk. Email: hr.hussien@psau.edu.sa

Received: 02 December 2020; Accepted: 23 January 2021

Abstract: This work presents a memetic Shuffled Frog Leaping Algorithm (SFLA) based tuning approach of an Integral Sliding Mode Controller (ISMC) for a quadrotor type of Unmanned Aerial Vehicles (UAV). Based on the Newton–Euler formalism, a nonlinear dynamic model of the studied quadrotor is firstly established for control design purposes. Since the main parameters of the ISMC design are the gains of the sliding surfaces and signum functions of the switching control law, which are usually selected by repetitive and time-consuming trials-errors based procedures, a constrained optimization problem is formulated for the systematically tuning of these unknown variables. Under time-domain operating constraints, such an optimization-based tuning problem is effectively solved using the proposed SFLA metaheuristic with an empirical comparison to other evolutionary computation- and swarm intelligence-based algorithms such as the Crow Search Algorithm (CSA), Fractional Particle Swarm Optimization Memetic Algorithm (FPSOMA), Ant Bee Colony (ABC) and Harmony Search Algorithm (HSA). Numerical experiments are carried out for various sets of algorithms' parameters to achieve optimal gains of the sliding mode controllers for the altitude and attitude dynamics stabilization. Comparative studies revealed that the SFLA is a competitive and easily implemented algorithm with high performance in terms of robustness and non-premature convergence. Demonstrative results verified that the proposed metaheuristics-based approach is a promising alternative for the systematic tuning of the effective design parameters in the integral sliding mode control framework.

Keywords: Quadrotor; modeling; integral sliding mode control; gains tuning; advanced metaheuristics; memetic algorithms; shuffled frog leaping algorithm

1 Introduction

The quadrotor is the most popular type of Unmanned Aerial Vehicles (UAV) thanks to its multiple applications in various areas of engineering and civilian domains [1,2]. The capacities of Vertical and Take-Off Landing (VTOL) and maneuverability prove the growing use of these vehicles. The flight dynamics stabilization of a quadrotor is a significant and hard task as it is an underactuated, nonlinear, and large-scale system [3–5]. External disturbances and model uncertainties of quadrotors further increase the complexity of these aerial vehicles. All these physical and environmental features require powerful and robust controllers' design and tuning with systematic and low time-consuming procedures.

In the literature, various control strategies have been proposed to cover this kind of issue. In the work of Ma et al. [6], an optimal control method of an unmanned aerial helicopter with unknown disturbances has been designed based on the Hamilton-Jacobi-Bellman equation. A linear model predictive controller has been proposed for a tilt-rotor tricopter to handle the angular dynamics and vertical body velocity in [7]. In the work of Argentim et al. [8], a comparison between different linear controllers, i.e. ITAE-tuned PID, classical LQR, and LQR loop-shaping-tuned PID, has been established for the vertical attitude stabilization of a quadcopter platform. For these linear approaches, the stabilization cannot be guaranteed when the vehicle moves away from its flight domain. Since the nonlinear control approaches can rise above some of the restrictions and drawbacks of the linear case, various nonlinear control methods have been developed. In the work of Stebler et al. [9], a nonlinear output feedback controller has been designed leading to an asymptotic altitude and attitude tracking of a quadrotor UAV under model uncertainties and unknown external disturbances. In the work of Ben Ammar et al. [10], the stabilization of a quadrotor vehicle has been achieved thanks to a proposed fuzzy gains-scheduling integral sliding mode controller. The main advantage of the proposed fuzzy-tuned sliding mode controller is the complete compensation of the matched disturbances when the quadrotor is in the sliding phase. Since the sliding mode control approaches have been successfully applied for various complex systems, i.e. robot manipulators [11], electrical motors [12], underwater vehicles [13], and so on, the Integral Sliding Mode Control (ISMC) variant is adopted in this paper to stabilize the altitude and attitude dynamics of a quadrotor UAV. The ISMC is a robust variant of the canonical sliding mode control approach where an integral action is added to the general form of the sliding surface shape [14,15]. Such a control approach allows eliminating the reaching phase and guarantees the invariance from initial conditions.

Unfortunately, the ISMC approach claims the choice and tuning of several effective control parameters which make difficult the controller's design procedure. These effective control parameters are often selected by repetitive trials-errors based methods which become hard and time-consuming for complex and large-scale systems. To overcome this drawback, several attempts have been proposed in the literature. In the work of Dydek et al. [16], an adaptive fuzzy sliding mode controller has been designed for the attitude stabilization of an unmanned vehicle under parametric uncertainties and disturbances. Such a proposed fuzzy sliding controller has allowed the reduction of the undesirable chattering phenomenon but at the expense of slow response of the system dynamics. In the work of Chang [17], fuzzy inference systems have been used to approximate the unknown attitude dynamics of the quadrotor and eliminate the reaching phase under external disturbances and uncertainties. In the work of Zhang et al. [18], an adaptive mechanism has been designed to estimate the integral sliding mode controllers' gains to eliminate chattering behavior. All these design methods remain time-consuming, non-systematic, and difficult in terms of real-world prototyping and experimentation. So, looking for an effective and systematic tuning technique becomes a necessity instead of tedious and expensive trials-errors based methods. For this purpose, the theory of global metaheuristics and hard optimization can present a promising solution.

Several metaheuristics have been proposed in the literature and received much interest in dealing with hard and large-scale optimization problems [19]. Recently, memetic metaheuristics have shown remarkable superiority in terms of exploitation/exploration capabilities for a robust and non-premature convergence compared to the classical ones. Initially proposed by Eusuff et al. [20,21], the Shuffled Frog Leaping Algorithm (SFLA) is one of the most powerful memetic metaheuristics. In this formalism, a set of virtual frogs are partitioned into different groups called memeplexes. Within each memeplex of the swarm, the individual frogs hold ideas that can be influenced by the behavior of other frogs with better fitness and evolve through a process of memetic evolution. After a defined number of memetic evolution steps, ideas are passed among memeplexes, which are considered as different cultures of frogs. Given the high performance of this algorithm, several improvements [22–25] and applications [26–28] have been proposed in the literature.

In this paper, the selection and tuning procedure of all effective parameters of an internal sliding mode controller for a quadrotor UAV is formulated as a constrained optimization problem. The operational constraints are specified in terms of time-domain performance criteria. Such a nonlinear and large-scale optimization problem is efficiently solved based on the proposed SFLA metaheuristic compared to other global swarm intelligence algorithms. The main contribution of the proposed approach is the systematic and fast way of tuning many effective sliding mode controllers' parameters for a nonlinear and complex system. The classical trials-errors based methods, which often lead to local solutions of the tuning problem, are no longer used and the design time is further reduced. All control performances and operational constraints are implicitly specified into the formulated optimization problem which will be solved in a single run and without recourse to a repetitive test-check procedure.

The remainder of this paper is organized as follows. In Section 2, a nonlinear model of the studied quadrotor is established thanks to the Newton–Euler formalism. Section 3 describes the formulation of the integral sliding mode controllers' gains tuning as an optimization problem under operational constraints. Section 4 is devoted to the description of the proposed memetic shuffled frog leaping metaheuristic to solve at best the formulated tuning problem. In Section 5, demonstrative results are carried out to show the superiority and effectiveness of the proposed SFLA-tuned sliding mode control approach. Conclusions are drawn in Section 6.

2 Quadrotor Dynamics Modeling

A quadrotor UAV is equipped with four rotors that are independently controlled as shown in Fig. 1. The motion of such a vehicle results from changes in the rate of the rotors. The body is symmetrical, and the propellers are rigid. The thrust and drag forces are proportional to the square of the propeller's speeds denoted as ω_i, $i \in \{1, 2, 3, 4\}$.

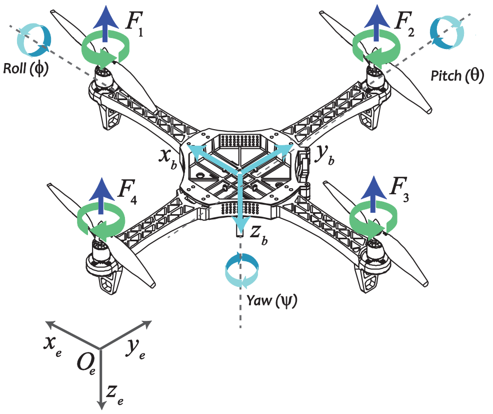


Figure 1: Mechanical structure of the quadrotor and related frames

In the flight space, the orientation of the quadrotor is accomplished by the rotation matrix $\boldsymbol{\mathcal{R}}: \boldsymbol{\mathcal{F}}_E \to \boldsymbol{\mathcal{F}}_B$ defined as follows [29,30]:

$$\boldsymbol{\mathcal{R}}(\phi,\theta,\psi) = \begin{bmatrix} \cos\psi\cos\theta & \sin\phi\sin\theta\cos\psi - \sin\psi\cos\theta & \cos\phi\sin\theta\cos\psi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\phi\sin\theta\sin\psi + \cos\psi\cos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{1}$$

where $-\pi/2 \le \phi \le \pi/2$, $-\pi/2 \le \theta \le \pi/2$, and $-\pi \le \psi \le \pi$ are the roll, pitch, and yaw Euler angles, respectively, $\boldsymbol{\mathcal{F}}_E = \{O_e, \boldsymbol{x}_e, \boldsymbol{y}_e, \boldsymbol{z}_e\}$ and $\boldsymbol{\mathcal{F}}_B = \{O_b, \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b\}$ are the earth- and body-frames, respectively.

The quadrotor's position and attitude are defined as $\boldsymbol{\xi} = [x,y,z]^T$and $\boldsymbol{\eta} = [\phi,\theta,\psi]^T$, respectively. Let a vector $\boldsymbol{\nu} = (u,v,w)^T$denotes the linear velocity of the quadrotor in the earth-frame $\boldsymbol{\mathcal{F}}_E$, while $\boldsymbol{\vartheta} = (p,q,r)^T$ represents its angular velocity in the body-frame $\boldsymbol{\mathcal{F}}_B$. The kinematic equations of the rotational and translational motions are obtained, respectively, as follows [29,30]:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2}$$

$$\boldsymbol{\nu}_e = \boldsymbol{\mathcal{R}}(\phi,\theta,\psi)\,\boldsymbol{\nu}_B \tag{3}$$

where $\boldsymbol{\nu}_e$ and $\boldsymbol{\nu}_B$ denote the linear velocities of the drone in the earth- and body-frames, respectively.

While using the Newton–Euler formalism, a complete model of the vehicle is given as follows:

$$\begin{cases} m\ddot{\boldsymbol{\xi}} = \boldsymbol{F}_{th} + \boldsymbol{F}_d + \boldsymbol{F}_g \\ \boldsymbol{J}\dot{\Omega} = \boldsymbol{M} - \boldsymbol{M}_{gp} - \boldsymbol{M}_{gb} - \boldsymbol{M}_a \end{cases} \tag{4}$$

where $\boldsymbol{F}_{th} = \boldsymbol{\mathcal{R}}(\phi,\theta,\psi)\left[0,0,\sum_{i=1}^{4} F_i\right]^T$is the thrust force of the rotors, $\boldsymbol{F}_d = diag(\kappa_1,\kappa_2,\kappa_3)\,\dot{\boldsymbol{\xi}}^T$ is the air drag force, $\boldsymbol{F}_g = [0,0,mg]^T$is the gravity force, $\boldsymbol{M} = \left[\tau_\phi,\tau_\theta,\tau_\psi\right]^T$denotes the total rolling, pitching, and yawing moments, $\boldsymbol{M}_{gp}$ and $\boldsymbol{M}_{gb}$ are the propellers' and quadrotor's body gyroscopic torques, respectively, and $\boldsymbol{M}_a = diag\,(\kappa_4, \kappa_5, \kappa_6)\left[\dot{\phi}^2, \dot{\theta}^2, \dot{\psi}^2\right]^T$is the moment resulting from the aerodynamic frictions [29,30].

Substituting the position vector and the forces expressions into Eq. (4), the following translational dynamics of the quadrotor is obtained:

$$\begin{cases} \ddot{x} = \dfrac{1}{m}\left(\cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi\right) u_1 - \dfrac{\kappa_1}{m}\dot{x} \\ \ddot{y} = \dfrac{1}{m}\left(\cos\phi \sin\psi \sin\theta - \sin\phi \cos\psi\right) u_1 - \dfrac{\kappa_2}{m}\dot{y} \\ \ddot{z} = \dfrac{1}{m}\cos\phi \cos\theta u_1 - g - \dfrac{\kappa_3}{m}\dot{z} \end{cases} \tag{5}$$

From the second part of Eq. (4), the rotational dynamics can be computed as:

$$\begin{cases} \ddot{\phi} = \dfrac{\left(I_y - I_z\right)}{I_x}\dot{\theta}\dot{\psi} - \dfrac{J_r}{I_x}\overline{\Omega}_r\dot{\theta} - \dfrac{\kappa_4}{I_x}\dot{\phi}^2 + \dfrac{1}{I_x}u_2 \\ \ddot{\theta} = \dfrac{\left(I_z - I_x\right)}{I_y}\dot{\phi}\dot{\psi} - \dfrac{J_r}{I_y}\overline{\Omega}_r\dot{\phi} - \dfrac{\kappa_5}{I_y}\dot{\theta}^2 + \dfrac{1}{I_y}u_3 \\ \ddot{\psi} = \dfrac{\left(I_x - I_y\right)}{I_z}\dot{\theta}\dot{\phi} - \dfrac{\kappa_6}{I_z}\dot{\psi}^2 + \dfrac{1}{I_z}u_4 \end{cases} \tag{6}$$

where $\overline{\Omega}_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$ is the overall residual rotor speed, I_x, I_y and I_z are the body inertia, J_r is the rotor inertia and κ_i, $i = 1, 2, \ldots, 6$ are the aerodynamic friction and translational drag coefficients.

The control inputs of the quadrotor are defined as u_1, u_2, u_3, and u_4 which represent the total thrust force in the z-axis, roll, pitch, and yaw torques, respectively:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{7}$$

where b and d are the thrust and drag coefficients, respectively.

Finally, taking $\boldsymbol{X} = \left(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}\right)^T \in \mathbb{R}^{12}$ as a state vector, a state-space representation of the studied quadrotor is obtained as follows [29,30]:

$$\dot{\boldsymbol{X}} = f\left(\boldsymbol{X}, \boldsymbol{u}\right) = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_3 \overline{\Omega}_r x_4 + a_2 x_2^2 + b_1 u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_6 \overline{\Omega}_r x_2 + a_5 x_4^2 + b_2 u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + \dfrac{1}{m}\left(\cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi\right) u_1 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + \dfrac{1}{m}\left(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi\right) u_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \dfrac{\cos\phi \cos\theta}{m} u_1 - g \end{cases} \tag{8}$$

where $a_1 = \left(I_y - I_z\right)/I_x$, $a_2 = -\kappa_4/I_x$, $a_3 = -J_r/I_x$, $a_4 = \left(I_z - I_x\right)/I_y$, $a_5 = -\kappa_5/I_y$, $a_6 = -J_r/I_y$, $a_7 = \left(I_x - I_y\right)/I_z$, $a_8 = -\kappa_6/I_z$, $a_9 = -\kappa_1/m$, $a_{10} = -\kappa_2/m$, $a_{11} = -\kappa_3/m$, $b_1 = 1/I_x$, $b_2 = 1/I_y$ and $b_3 = 1/I_z$.

3 Control Problem Statements

The control objective is to design a robust controller, with a systematic and easy tuning method of its effective design parameters, for the attitude and altitude dynamics stabilization. Desired trajectories are defined as $\boldsymbol{X}_d = [\phi_d, \theta_d, \psi_d, z_d]^T$ for the outputs $\boldsymbol{X} = [\phi, \theta, \psi, z]^T$, assumed to be accessible.

3.1 Preliminary Results

In the ISMC framework, the expression of the sliding surface is defined as follows [31]:

$$s_i\left(\boldsymbol{X}, t\right) = \left(\frac{d}{dt} + \lambda_i\right) e_i\left(\boldsymbol{X}, t\right) + \beta_i \int_0^\infty e_i\left(\boldsymbol{X}, \tau\right) d\tau \tag{9}$$

where $\boldsymbol{X}$ denotes the controlled variables, $e_i(\cdot)$ is the tracking error defined as $e_i(t) = \boldsymbol{X}_d(t) - \boldsymbol{X}(t)$, λ_i are positive constants and β_i are the integral gains, $i \in \{\phi, \theta, \psi, z\}$.

The Lyapunov candidate scalar function$V(\boldsymbol{X},t) = 1/2s^2(\boldsymbol{X},t) > 0$ and the reaching condition$s(\boldsymbol{X},t)\,\dot{s}(\boldsymbol{X},t) < 0$ lead to the following integral sliding mode control law for the altitude dynamics [10]:

$$u_1 = \frac{m}{\cos\phi\cos\theta}\left[-a_{11}x_{12} - \lambda_z\dot{z} - \beta_z(z - z_d) - g\right] - K_z\mathrm{sgn}(s_z) \tag{10}$$

where $\cos\phi\cos\theta \neq 0$, $\lambda_z > 0$, $\beta_z > 0$, and $K_z > 0$ are the effective design parameters.

Following the same steps, the control laws u_2, u_3, and u_4 responsible for generating the roll, pitch, and yaw rotations, respectively, are calculated as follows:

$$u_2 = \frac{1}{b_1}\left[-a_1x_4x_6 - a_3\overline{\Omega}_r x_4 - a_2x_2^2 - \lambda_\phi x_2 - \beta_\phi e_\phi\right] - K_\phi\mathrm{sgn}\left(s_\phi\right) \tag{11}$$

$$u_3 = \frac{1}{b_2}\left[-a_4x_2x_6 - a_6\overline{\Omega}_r x_2 - a_5x_4^2 - \lambda_\theta x_4 - \beta_\theta e_\theta\right] - K_\theta\mathrm{sgn}(s_\theta) \tag{12}$$

$$u_4 = \frac{1}{b_3}\left[a_7x_2x_4 + a_8x_6^2 - \lambda_\psi x_4 - \beta_\psi e_\psi\right] - K_\psi\mathrm{sgn}\left(s_\psi\right) \tag{13}$$

where $\lambda_j > 0$, $\beta_j > 0$, and $K_j > 0$, $j \in \{\phi, \theta, \psi\}$, are the effective design parameters of the integral sliding mode controllers for the attitude dynamics.

3.2 *Optimization-based Problem Formulation*

As shown in Eqs. (10)–(13), the design of integral sliding controllers for the drone dynamics involves the tuning of several parameters, i.e., slopes of the sliding surfaces λ_i, integral and sign functions gains β_i and K_i, $i \in \{\phi, \theta, \psi, z\}$. The selection of such a large number of effective control parameters is not a straightforward problem. Indeed, the classical trials-errors based procedures become ineffective, time-consuming, and do not guarantee an optimal solution to the problem. So, such a tuning procedure can be effectively solved thanks to proposed metaheuristics while considering the following formulation:

$$\begin{cases} \text{Minimize } f_i(\boldsymbol{x}, t) \\ \boldsymbol{x} = \left[\lambda_\phi, \lambda_\theta, \lambda_\psi, \lambda_z, \beta_\varphi, \beta_\theta, \beta_\psi, \beta_z, K_\phi, K_\theta, K_\psi, K_z\right]^T \in \boldsymbol{\mathcal{S}} \subseteq \mathbb{R}_+^{12} \\ \textit{Subject to}: \\ g_1(\boldsymbol{x}) = t_r^\phi - t_r^{\phi,\max} \leq 0; g_2(\boldsymbol{x}) = t_r^\theta - t_r^{\theta,\max} \leq 0 \\ g_3(\boldsymbol{x}) = t_r^\psi - t_r^{\psi,\max} \leq 0; g_4(\boldsymbol{x}) = t_r^z - t_r^{z,\max} \leq 0 \end{cases} \tag{14}$$

where $\boldsymbol{x} \in \boldsymbol{\mathcal{S}} \subseteq \mathbb{R}_+^{12}$ is the vector of decision variables which contains all controllers' gains, $\mathbb{R}_+^{12}$ denotes the 12-dimensional set of the positive real numbers, $f_i\colon \mathbb{R}^{12} \to \mathbb{R}$, $i \in \{z, \phi, \theta, \psi\}$ are the cost functions to be minimized under operational time-domain constraints, $\boldsymbol{\mathcal{S}} = \left\{\boldsymbol{x} \in \mathbb{R}_+^{12}, \boldsymbol{x}_{low} \leq \boldsymbol{x} \leq \boldsymbol{x}_{up}\right\}$ denotes the bounded search space, $g_m\colon \mathbb{R}^{12} \to \mathbb{R}$, $m = 1, 2, \ldots, 4$ are the operational constraints. The terms t_r^ϕ, t_r^θ, t_r^ψ, and t_r^z denote the step-response rise times of the controlled dynamics. The terms $t_r^{\phi,\max}$ $t_r^{\theta,\max}$, $t_r^{\psi,\max}$, and $t_r^{z,\max}$ denote their pre-specified maximum values.

The cost functions of the problem (14) are chosen, for each controlled dynamics as the Integral of Absolute Error (IAE) performance criteria, defined as follows:

$$f_i(\boldsymbol{x}) = \int_0^{+\infty} |e_i(\boldsymbol{x}, \tau)|\, d\tau \tag{15}$$

where $\boldsymbol{e}(\boldsymbol{x}, t) = \{e_i(\boldsymbol{x}, t)\}_{i=\phi,\theta,\psi,z}$ denotes the tracking errors between the references $\boldsymbol{X}_d(t)$ and the controlled process variables $\boldsymbol{X}(\boldsymbol{x}, t)$, i.e., $e_\phi(\boldsymbol{x}, t) = \phi_d - \phi(\boldsymbol{x}, t)$, $e_\theta(\boldsymbol{x}, t) = \theta_d - \theta(\boldsymbol{x}, t)$, and so on.

Since the optimization problem (14) is a multi-objective type, the following aggregation function is used [19]:

$$f(\boldsymbol{x}, t) = \sum_{i=1}^{4} \delta_i f_i(\boldsymbol{x}, t) \tag{16}$$

where $\delta_i > 0$ are the weighting coefficients of the aggregation function ensuring $\sum_{i=1}^{4} \delta_i = 1$.

To handle the constraints of the problem (14), the following static penalty function is used [19]:

$$\Lambda(\boldsymbol{x}, t) = f(\boldsymbol{x}, t) + \sum_{q=1}^{n_{con}} \mu_q \max\left\{0, g_q(\boldsymbol{x}, t)\right\}^2 \tag{17}$$

where μ_q are the scaling penalty parameters and n_{con} denotes the number of inequality constraints.

4 Proposed Memetic Shuffled Frog Leaping Algorithm

4.1 Overview

In the shuffled frog leaping algorithm, a population of n_{pop} frogs is randomly initialized in the feasible search space $\boldsymbol{\mathcal{S}} \subseteq \mathbb{R}^m$ and portioned into n_{mem} memeplexes. Each memeplex contains n frogs. At each iteration$k$, the position of the ith frog in the jth memeplex is denoted as $\boldsymbol{x}_i^j = \left(x_{i,1}^j, x_{i,2}^j, \ldots, x_{i,m}^j\right)$, where $1 \leq i \leq n_{pop}$ and $1 \leq j \leq n_{mem}$. Then, all frogs are sorted in descending order according to their fitness following the given ranking strategy [21]. In each memeplex, the frog with the global best fitness value, evaluated in terms of the cost function, is expressed as $\boldsymbol{x}_k^g$. Those with the best and the worst fitness values are identified as $\boldsymbol{x}_k^b$ and $\boldsymbol{x}_k^w$, respectively. Then, a process of local search for each memeplex is performed to improve only the frogs with the worst fitness through the memetic evolution [21]. At each iteration k, the new position of the frog with the worst fitness leaps towards the position of the best one according to the following motion equations:

$$\Delta_{i,k}^w = r_k\left(\boldsymbol{x}_{i,k}^b - \boldsymbol{x}_{i,k}^w\right) \tag{18}$$

$$\boldsymbol{x}_{i,k+1}^w = \boldsymbol{x}_{i,k}^w + \Delta_{i,k}^w, \quad \Delta_{\min}^w \leq \Delta_{i,k}^w \leq \Delta_{\max}^w \tag{19}$$

where r_k is a random number uniformly distributed in [0,1], $\Delta_{i,k}^w$ is the change in the ith frog position, $\Delta_{\min}^w$ and $\Delta_{\max}^w$ are the minima and maximum allowed changes.

As mentioned in [22–25], if this memetic process produces a better solution, i.e., $\boldsymbol{x}_{i,k+1}^{w}$ is better than $\boldsymbol{x}_{i,k}^{w}$, the new frog is then used to replace the old one in the memeplex. Otherwise, the motion Eqs. (18) and (19) are recalculated but concerning the global best frog $\boldsymbol{x}_{k}^{g}$ instead of $\boldsymbol{x}_{k}^{b}$. If there is no improvement, the worst frog $\boldsymbol{x}_{k}^{w}$ will be replaced by a new randomly generated solution. The memetic evolution process is then continued for a predefined number of iterations denoted as k_{ls}. The new memeplexes are reformed according to the new order ranked by the fitness value again.

4.2 Pseudo-Code

The main control parameters of the SFLA are: numbers of memeplexes n_{mem}, frogs in a memeplex n, memetic generations k_{ls}, and shuffling iterationsk_{shuff}. A pseudo-code for the SFLA is given by [22–28]:

Step 1: Initialize all control parameters: n_{pop}, n_{mem}, k_{ls}, k_{shuff}, $k_{\max}$, etc.

Step 2: Randomly generate the initial population and evaluate the n_{pop} solutions.

Step 3: Sort the population in descending order of fitness.

Step 4: Divide the population into n_{mem} memeplexes.

Step 5: Call the local search mechanism for the memetic evolution process;

- Determine the best and worst solutions $\boldsymbol{x}_{k}^{b}$ and $\boldsymbol{x}_{k}^{w}$, respectively
- Improve the worst frog position according to Eqs. (18) and (19) and repeat iterations

Step 6: Shuffle the evolved memeplexes and sort the new population

Step 7: Test the termination criterion and return the best solution.

4.3 Numerical Experimentations and Discussion

To evaluate the performance of the proposed SFLA, various standard test functions of Appendix A are used for numerical experimentation. The metaheuristics CSA, HSA, FPSOMA, and ABC are used as comparison tools. The termination criterion is set as the maximum number of iterations reaches 10000, the population size is 50 and specific control parameters are set as follows:

- SFLA [20,21]: memeplexes 5, frogs in a memeplex 10 and generations before shuffling 5.
- CSA [32]: flight length 2 and awareness probability 0.1.
- HSA [33]: harmony memory rate 0.9 and pitch adjusting rate 0.3.
- FPSOMA [34]: inertia weight 0.729, derivative 0.9, cognitive and social coefficients 0.9.
- ABC [35]: limit of abandonments 60.

The proposed metaheuristics are implemented and evaluated using a CPU Core i3, 1.70 GHz computer. The convergence histories of all reported algorithms are depicted in Fig. 2.

Tab. 1 summarizes the optimization results for independent 20 runs of test functions. Best, Worst, and Mean denote the best, worst and mean results for the minimization, respectively. STD is the standard deviation and ET denotes the elapsed time (in seconds). Results of rank-based Friedman's analysis for all proposed metaheuristics are summarized in Tab. 2. From these analyses, one can note that the proposed SFLA has worthily attained the lowest average ranks compared to the remaining methods. Furthermore, it is shown that the SFLA achieved the first best computation time for test functions f_{1-5} and the second one for the benchmark f_6.

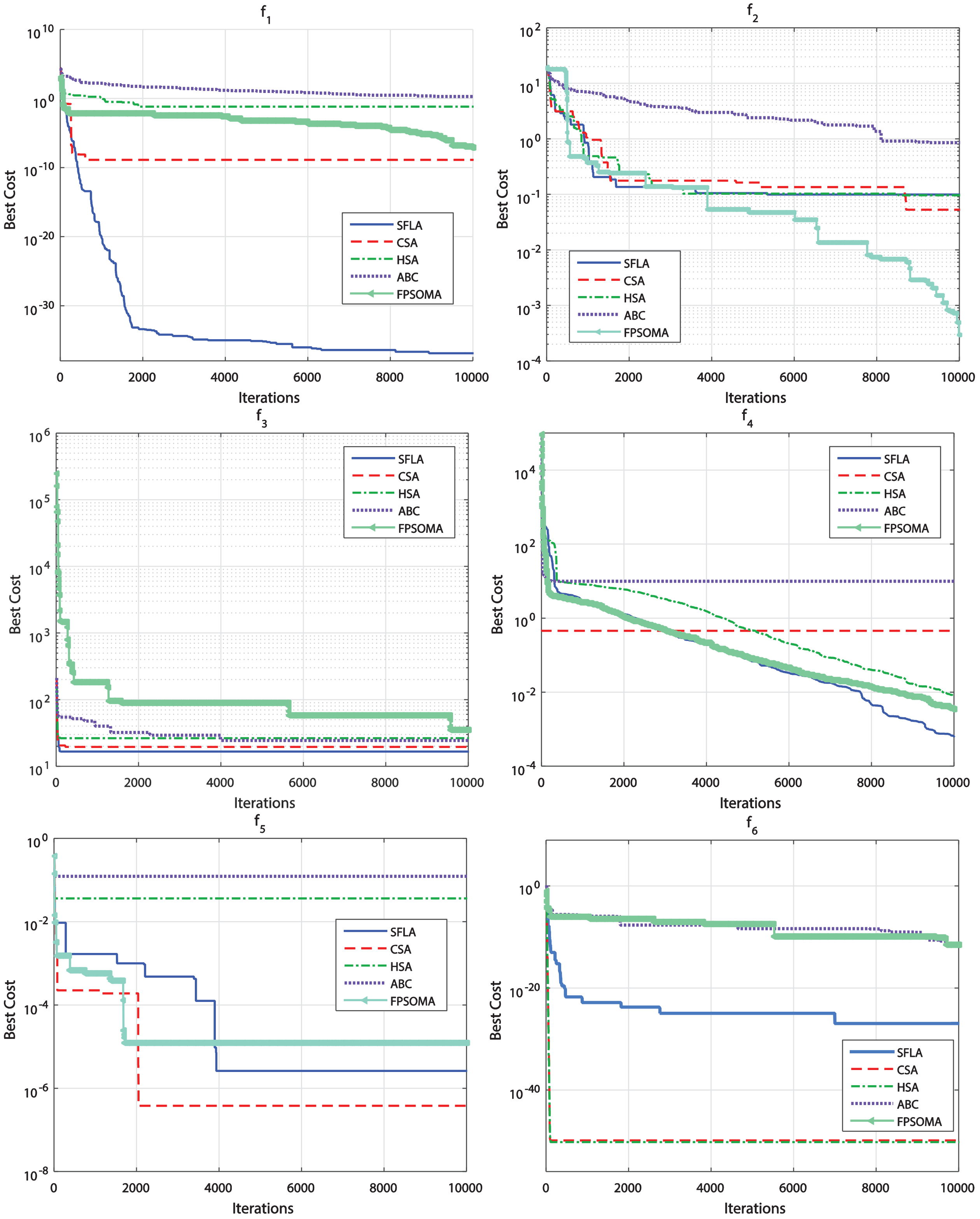


Figure 2: Convergence histories of the reported algorithms for the test functions

Table 1: Optimization results of the test functions over independent 20 runs

		SFLA	CSA	HSA	FPSOMA	ABC
f_1	Best	1.526e-37	1.530e–09	2.3110	7.042e–08	0.0020
	Mean	8.625e–37	0.2170	6.3840	9.546e–08	0.0040
	Worst	2.083e–36	2.0480	11.8240	1.261e–07	0.0060
	STD	6.229e–37	0.6430	2.8040	2.378e–08	0.0010
	ET	**44.9090**	76.3620	93.5580	50.0140	91.690
f_2	Best	0.1013	0.05411	0.0956	2.966e–04	0.8710
	Mean	0.1288	0.0960	0.1054	4.282e–04	1.4940
	Worst	0.1862	0.1304	0.1237	5.436e–04	2.1150
	STD	0.0330	0.0286	0.0123	1.182e–04	0.4900
	ET	**50.6840**	70.9230	62.2660	84.5130	99.9980
f_3	Best	16.9143	19.8991	24.8739	36.2184	26.8639
	Mean	35.8180	39.2012	54.3240	54.8999	40.9958
	Worst	60.6922	60.6923	106.4600	73.7089	56.7304
	STD	17.0888	17.6810	35.0820	13.4003	10.8007
	ET	**52.5380**	82.9330	78.3260	57.8310	92.6870
f_4	Best	0.0006	0.4147	0.0080	0.0030	9.3515
	Mean	0.8250	0.4604	1.6030	0.0090	10.0300
	Worst	4.0519	0.4888	3.9872	0.0140	11.2237
	STD	1.8037	0.0315	2.1750	0.0050	0.7640
	ET	**36.1440**	66.2670	58.1770	72.7530	54.5440
f_5	Best	2.688e–06	3.916e–07	0.0372	1.280e–05	0.1270
	Mean	4.168e–06	1.811e–06	0.3735	3.852e–05	0.3432
	Worst	5.221e–06	3.916e–06	0.4888	7.114e–05	0.4716
	STD	1.063e–06	1.595e–06	0.1905	2.577e–05	0.1395
	ET	**54.2860**	90.5220	78.4476	80.7872	102.6490
f_6	Best	1.460e–27	1.588e–50	7.696e–51	3.690e–12	3.309e–12
	Mean	2.282e–25	1.037e–44	1.808e–42	2.311e–11	1.071e–11
	Worst	7.647e–25	4.200e–44	9.041e–42	5.033e–11	2.530e–11
	STD	3.148e–25	1.810e–44	4.043e–42	2.427e–11	8.877e–12
	ET	68.947	**56.241**	70.982	120.332	98.150

Table 2: Rank-based statistical analysis of best performances

		SFLA	CSA	HSA	FPSOMA	ABC
f_1	Score	1.526e–37	1.530e–09	2.3110	7.042e–08	0.0020
	Rank	**1**	2	5	3	4
f_2	Score	0.1013	0.05411	0.0956	2.966e–04	0.8710
	Rank	**4**	2	3	1	5
f_3	Score	16.9143	19.8991	24.8739	36.2184	26.8639
	Rank	**1**	2	3	5	4
f_4	Score	0.0006	0.4147	0.0080	0.0030	9.3515
	Rank	**1**	4	3	2	5
f_5	Score	2.688e–06	3.916e–07	0.0372	1.280e–05	0.1270
	Rank	**2**	1	4	3	5
f_6	Score	1.460e–27	1.588e–50	7.696e–51	3.690e–12	3.309e–12
	Rank	**3**	2	1	4	5
Average rank		**2**	2.166	3.166	3	4.66

5 Application to the ISMC Gains' Tuning

The numeric values of the studied quadrotor's parameters are given in Appendix B. The proposed model of such a vehicle has been validated in our previous works [10,29,30].

5.1 Simulation Results and Discussion

The reported algorithms were independently executed 20 times for the problem (14) over 100 iterations and using a population size of 50. Such a set of control parameters is deduced after various runs, where the number of iterations is raised for different values of the population size as shown in Tab. 3. Increasing the number of population and iterations allows for better results but with prohibitive computation times. Tab. 4 shows the statistical results over the independent 20 runs. The convergence histories of all proposed algorithms are given in Fig. 3.

Table 3: Optimization results of (14) for different number of iterations and generations

Max. Iterations	Pop. size	Algorithms				
		SFLA	CSA	HSA	FPSOMA	ABC
50	20	3.2574	5.6237	7.2634	6,6460	12.8952
	30	3.2103	4.5084	3.2697	2.1829	12.3616
	50	1.2041	3.1752	2.2309	1.4312	10.8258
100	20	0.1219	1.3558	1.0087	0.1956	6.3534
	30	0.0877	0.0962	0.1038	0.2828	5.0768
	50	**0.0697**	**0.0755**	**0.0964**	**0.3892**	**4.1504**
200	20	0.0702	0.0754	0.1254	0.3881	4.1499
	30	0.0701	0.0751	0.0952	0.3764	4.2186
	50	0.0684	0.8001	0.0936	0.4002	4.2664

Table 4: Optimization results of the problem (14) over independent 20 runs

	Best	Mean	Worst	STD	ET
SFLA	**0.0632**	**0.0702**	**0.0813**	**0.0124**	**2294.485**
CSA	0.0752	0.1101	0.1545	0.0391	5660.380
HSA	0.0985	0.0999	0.1446	0.0452	4592.608
FPSOMA	0.2208	0.4186	0.4273	0.0128	3917.282
ABC	4.0324	4.4617	5.6352	0.9982	8148.796

From these numerical experimentations and demonstrative results, the best performance in terms of solutions quality, fastness, and exploitation/exploration capabilities is always obtained with the SFLA-based tuning approach. Such a memetic algorithm greatly outperforms all other reported metaheuristics especially against the well-known premature convergence problem as depicted by convergence curves of Fig. 3. The STD metric of the SFLA-tuned ISMC case is the best compared to those of all reported CSA, HSA, FPSOMA, and ABC algorithms. This highlights the interest of the local search operator and the memetic aspect of such a global metaheuristic. The computation time is always the least in the case of SFLA which further proves the contribution of such a control tuning method. For the closed-loop performance evaluation, the effectiveness of the SFLA-tuned ISMC approach is shown in Fig. 4 for the attitude and altitude dynamics stabilization. For the given initial conditions $[\phi_0, \theta_0, \psi_0, z_0] = [0, 0, 0, 0]$, the quadrotor dynamics are successfully stabilized around the desired references $[\phi_d, \theta_d, \psi_d, z_d] = [0.9, 0.5, 0.5, 4]$ with comparative performance. The system responses are further damped in the SFLA-tuned controllers. The rise and settling times are improved and the disturbance rejection at simulation time 10 sec is guaranteed with a fast and smooth behavior against all other algorithms. A large overshoot with remarkable oscillations is shown for the CSA- and FPSOMA-tuned controllers.

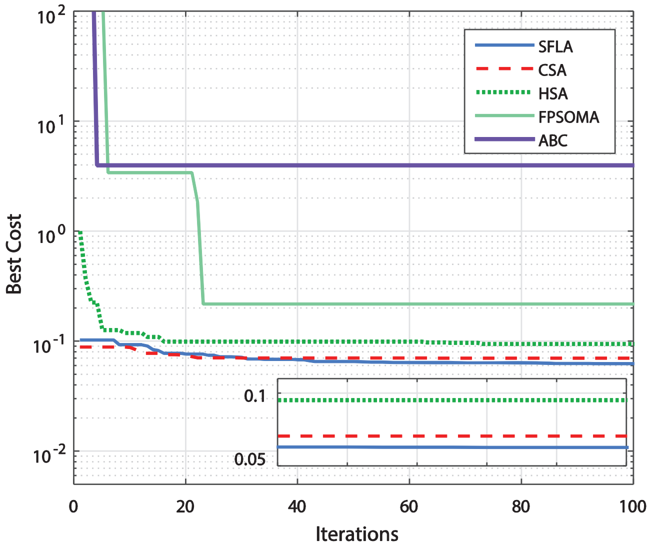


Figure 3: Convergence histories of the reported algorithms for problem (14)

5.2 Comparison with Classical ISMC Approaches

A comparative study based on the closed-loop time-domain performance is performed for the SFLA-tuned ISMC and classical ones without optimization. The cases of classical sliding mode control (SMC) [29,30], integral sliding mode control with the sign and saturation types of functions (ISMC-sign and ISMC-sat) [10] are considered.

The simulation results of Fig. 5 indicate the superiority of the SFLA-tuned controller compared to other algorithms. Indeed, all proposed methods stabilize the attitude and attitude dynamics, but the step responses are further damped and improved in the case of SFLA-based tuning algorithms. All other performance metrics are summarized for each controlled dynamics in Tabs. 5 to 8. In terms of control signal dynamics and chattering behaviors, the SFLA-tuned controllers present the best reductions of the undesirable chattering phenomenon as depicted in Fig. 6. The control signals are smoother and the oscillations as in the case of cases sign-SMC, sign-ISMC, and sat-ISMC are eliminated.

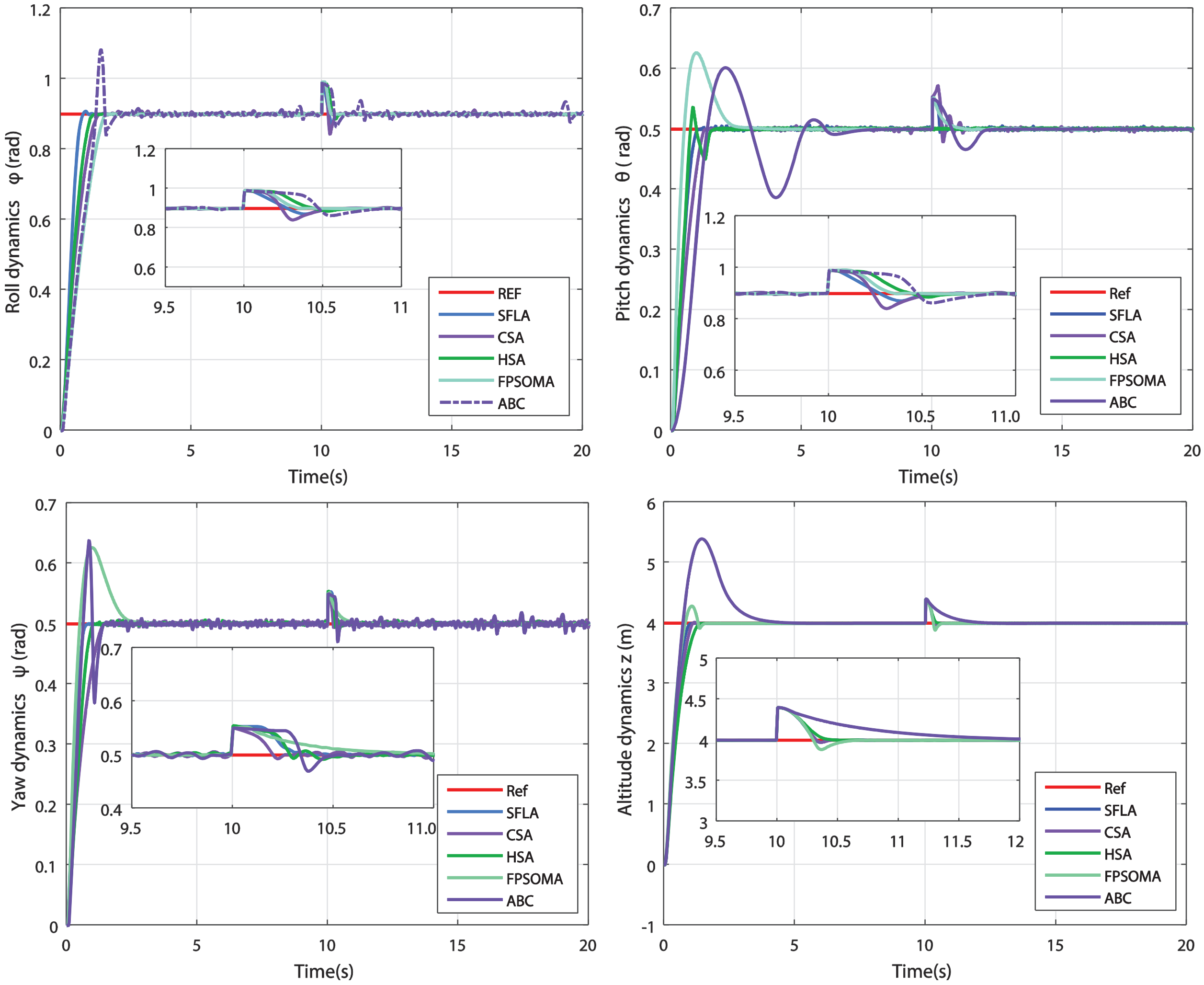


Figure 4: Time-domain performances of the controlled dynamics

Table 5: Time-domain performances comparison: altitude dynamics

Control approach	Rise time (sec)	Settling time (sec)	Overshoot (%)	Steady-state error
SMC [29,30]	1.50	2.50	0.00	0.00
ISMC-SFLA	**1.00**	**1.50**	**0.00**	**0.00**
ISMC-sat [10]	1.50	4.00	15.00	0.00
ISMC-sign [10]	1.50	5.00	25.00	0.00

Table 6: Time-domain performances comparison: pitch dynamics

Control approach	Rise time (sec)	Settling time (sec)	Overshoot (%)	Steady-state error
SMC [29,30]	0.52	0.99	0.00	0.00
ISMC-SFLA	**0.52**	**0.98**	**0.00**	**0.00**
ISMC-sat [10]	1.40	6.00	20.00	0.00
ISMC-sign [10]	1.40	10.00	35.00	0.00

Table 7: Time-domain performances comparison: roll dynamics

Control approach	Rise time (sec)	Settling time (sec)	Overshoot (%)	Steady-state error
SMC [29,30]	1.40	2.20	0.00	0.00
ISMC-SFLA	**0.60**	**1.00**	**0.00**	**0.00**
ISMC-sat [10]	1.20	7.00	22.50	0.00
ISMC-sign [10]	1.30	6.00	20.50	0.00

Table 8: Time-domain performances comparison: yaw dynamics

Control approach	Rise time (sec)	Settling time (sec)	Overshoot (%)	Steady-state error
SMC [29,30]	0.94	1.10	0.00	0.00
ISMC-SFLA	**0.91**	**1.00**	**0.00**	**0.00**
ISMC-sat [10]	1.20	3.00	20.00	0.00
ISMC-sign [10]	1.20	5.00	10.00	0.00

5.3 Sensitivity Analysis

As with all metaheuristics, the control parameters selection is critical for SFLA performance improvement. The sensitivity of such a memetic algorithm concerning the variation of its main parameters is studied for the problem (14). Fig. 7 gives the different convergence dynamics of SFLA under arbitrary variation of n and k_{ls} parameters, respectively. In all these scenarios, the convergence of the algorithm is always guaranteed but with degraded performance in terms of the exploitation/exploration capacities and the computing time. The proposed SFLA metaheuristic finds better results for problem (14) with the control parameters' values of $n = 10$ and $k_{ls} = 5$. Indeed, and as shown in [21], the number of memeplexes n_{mem} is also an important control parameter. As the population size increases, the function evaluations increase. Besides, when selectingn_{mem}, it is important to make sure that n is not too small. If there are too few frogs selected in a sub-memeplex, the information exchange is slow, resulting in longer solution times. The other main parameter is k_{ls} which can take any value greater than 1. If k_{ls} is small, the memeplexes will be shuffled frequently, reducing idea exchange on the local scale. However, if k_{ls} is large, each memeplex will be shrunk into a local optimum.

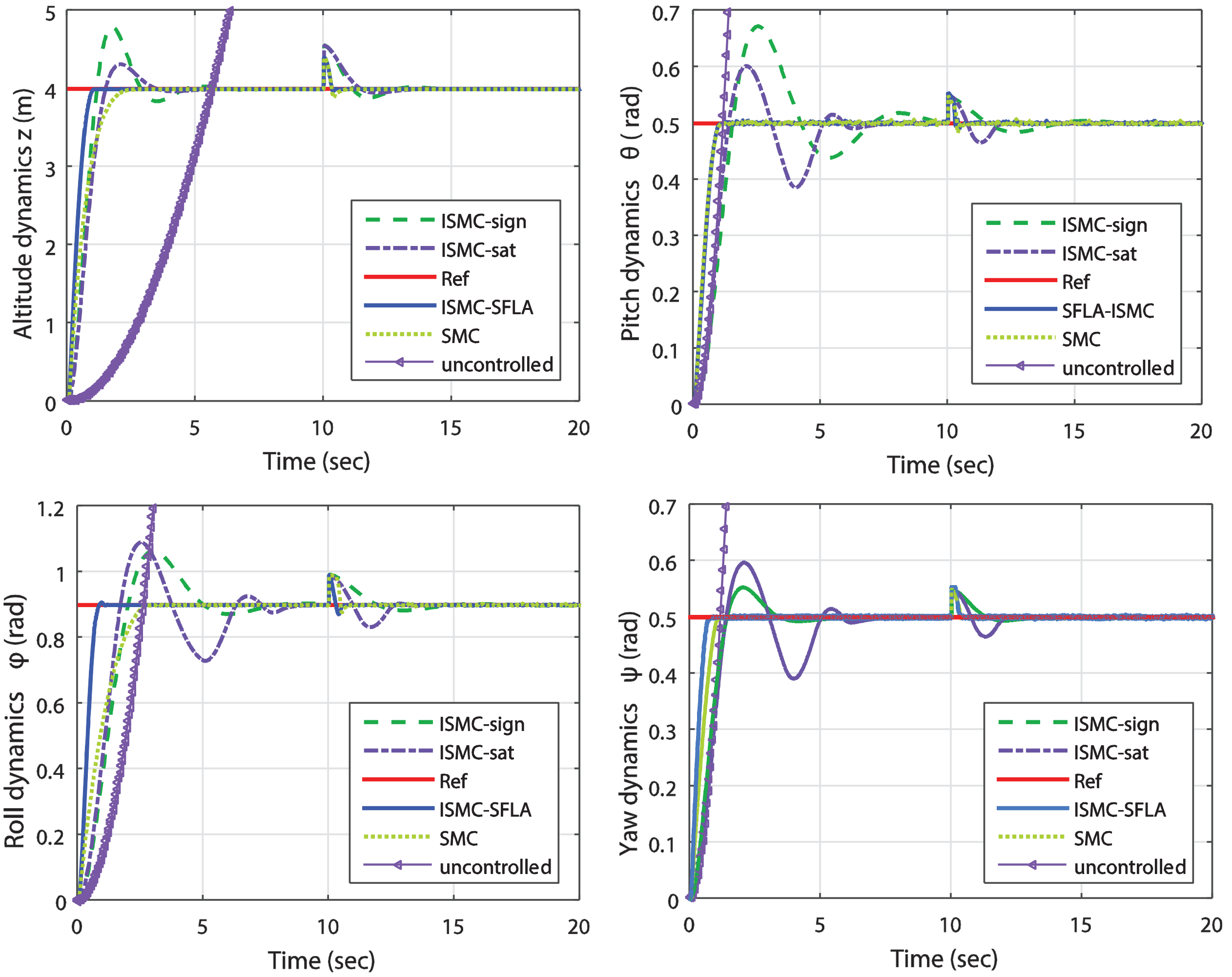
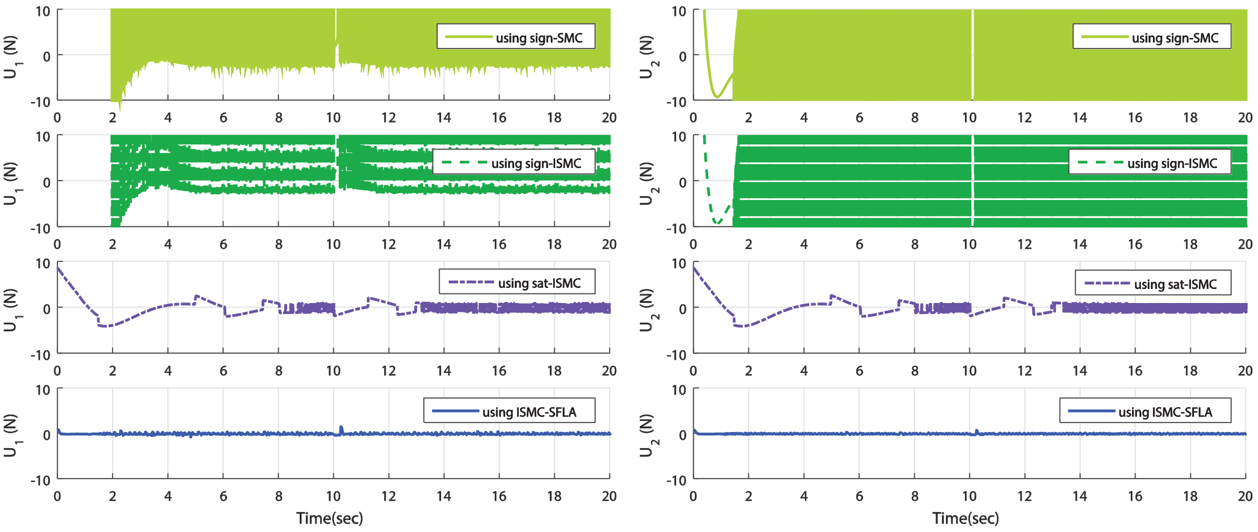


Figure 5: Performances comparison of the proposed SFLA-tuned controllers



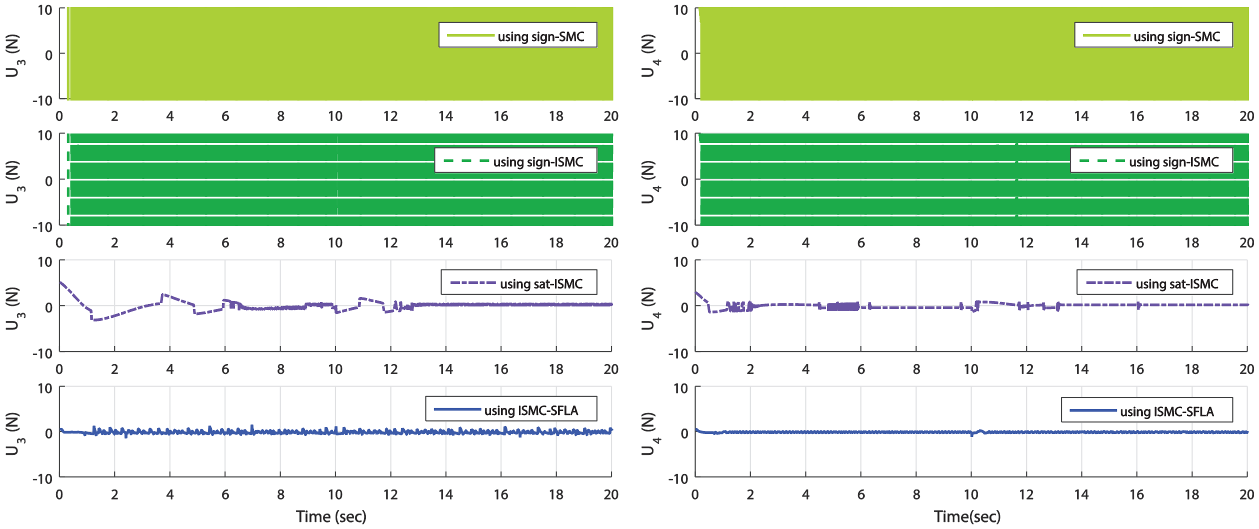


Figure 6: SFLA-based designed control laws for dynamics stabilization

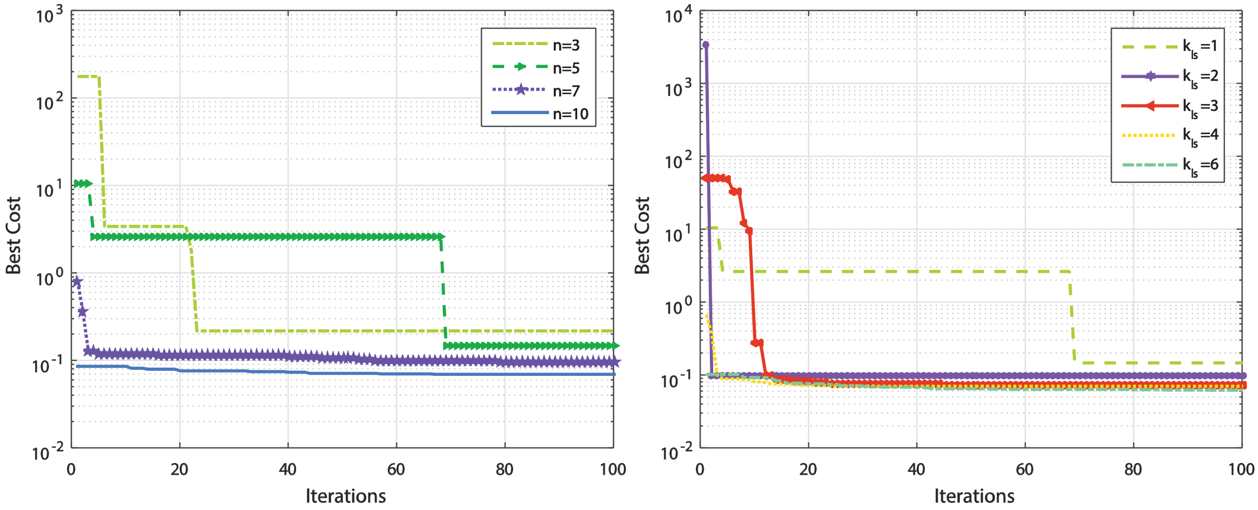


Figure 7: Convergence robustness of the SFLA under control parameters variations

6 Conclusions

In this paper, a systematic tuning strategy of all effective design parameters of the integral sliding mode control is proposed and successfully applied for the stabilization of a quadrotor UAV. The tuning parameters' procedure has been formulated as a constrained optimization problem and efficiently solved thanks to the proposed memetic SFLA metaheuristic. The classical trials-errors methods are no longer used, and the design time is further reduced. Numerical experiments were first performed to evaluate the proposed metaheuristics on standard test function benchmarks. Rank-based statistical analyses have been then presented to conclude on the performance of the different metaheuristics. While using the Newton–Euler formalism, a nonlinear model of the studied quadrotor UAV has been established and used in the altitude and attitude controllers' design and tuning. Demonstrative results, in terms of solutions optimality and time-domain performance, show the superiority and effectiveness of the proposed SFLA-tuned ISMC approach in comparison with other CSA, FPSOMA, HSA, and ABC competitive algorithms. With a local search mechanism, the robustness and insensitivity of the proposed SFLA metaheuristic are further improved. The use of such a memetic algorithm as an alternative to design more advanced control approaches is promising in the UAV framework.

Future works deal with the real-world experimentation of the SFLA-tuned ISMC approach using an embedded hardware/software board. The formulation of the proposed metaheuristics algorithms in an online tuning framework is also investigated.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. B. Sebbane, *Smart Autonomous Aircraft: Flight Control and Planning for UAV*, 1st ed., New York, USA: CRC Press, Taylor & Francis Group, 2016.

[2] K. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*, 1st ed., Dordrecht, Netherlands, Netherlands: Springer, 2015.

[3] R. Lozano, *Unmanned Aerial Vehicles: Embedded Control*, 1st ed., London, UK: Wiley-ISTE, 2010.

[4] P. Castillo, R. Lozano and A. E. Dzul, *Modelling and Control of Mini-Flying Machines*, 1st ed., London, UK: Springer, 2005.

[5] I. Fantoni and R. Lozano, *Non-linear Control for Underactuated Mechanical Systems*, 1st ed., London, UK: Springer-Verlag, 2002.

[6] H. Ma, M. Chen and Q. Wu, "Inverse optimal control for unmanned aerial helicopters with disturbances," *Optimal Control Applications and Methods*, vol. 40, no. 1, pp. 152–171, 2019.

[7] A. Prach and E. Kayacan, "An MPC-based position controller for a tilt-rotor tricopter VTOL UAV," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 343–356, 2018.

[8] L. M. Argentim, W. C. Rezende, P. E. Santos and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," in *Proc. of the 2nd ICIEV*, Dhaka, Bengladech, 2013.

[9] S. Stebler, W. Mackunis and M. Reyhanoglu, "Nonlinear output feedback tracking control of a quadrotor UAV in the presence of uncertainty," in *Proc. of the 14th ICARCV*, Phuket, Thailand, 2016.

[10] N. B. Ammar, S. Bouallègue and J. Haggège, "Fuzzy gains-scheduling of an integral sliding mode controller for a quadrotor unmanned aerial vehicle," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 132–141, 2018.

[11] V. I. Utkin and H. C. Chang, "Sliding mode control on electro-mechanical systems," *Mathematical Problems in Engineering*, vol. 8, no. 4–5, pp. 451–473, 2002.

[12] I. C. Baik, K. H. Kim and M. J. Youn, "Robust nonlinear speed control of PM synchronous motor using boundary layer integral sliding mode control technique," *IEEE Transaction on Control Systems Technology*, vol. 8, no. 1, pp. 47–54, 2000.

[13] Y. S. Song and M. R. Arshad, "Sliding mode depth control of a hovering autonomous underwater vehicle," in *Proc. of 5th ICCSCE*, George Town, Malaysia, pp. 435–440, 2015.

[14] K. Young, "Controller design for a manipulator using theory of variable structure systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 2, pp. 101–109, 1978.

[15] Y. Pan, Y. H. Joo and H. Yu, "Discussions on smooth modifications of integral sliding mode control," *International Journal of Control, Automation, and Systems*, vol. 16, no. 2, pp. 586–593, 2018.

[16] Z. T. Dydek, A. M. Annaswamy and E. Lavretsky, "Adaptive control of quadrotor UAVs in the presence of actuator uncertainties," in *Proc. of AIAA Infotech@Aerospace 2010*, Atlanta, Georgia, pp. 1–9, 2010.

[17] S. Chang and W. Shi, "Adaptive fuzzy time-varying sliding mode control for quadrotor UAV attitude system with prescribed performance," in *Proc. of 29th CCDC*, Chongqing, China, pp. 4389–4394, 2017.

[18] Y. Zhang and P. Yan, "An adaptive integral sliding mode control approach for piezoelectric nano-manipulation with optimal transient performance," *Mechatronics*, vol. 52, no. 1, pp. 119–126, 2018.

[19] M. Gendreau and J. Y. Potvin, *Handbook of Metaheuristics*, 3rd ed., Gewerbestrasse, Switzerland: Springer International Publishing AG, 2019.

[20] M. M. Eusuff and K. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning & Management*, vol. 129, no. 3, pp. 210–225, 2003.

[21] M. M. Eusuff, K. Lansey and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.

[22] W. Bi, F. Yu, N. Cao, W. Huo, G. Cao *et al.,* "Research on data extraction and analysis of software defect in IoT communication software," *Computers Materials & Continua*, vol. 65, no. 2, pp. 1837–1854, 2020.

[23] X. Duan, T. Niu and Q. Huang, "An improved shuffled frog leaping algorithm and its application in dynamic emergency vehicle dispatching," *Mathematical Problems in Engineering*, vol. 2018, no. 7896926, pp. 1–34, 2018.

[24] M. Jadidoleslam and A. Ebrahimi, "Reliability constrained generation expansion planning by a modified shuffled frog leaping algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 64, no. 4, pp. 743–751, 2015.

[25] G. Y. Zhu and W. B. Zhang, "An improved shuffled frog-leaping algorithm to optimize component pick-and-place sequencing optimization problem," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6818–6829, 2014.

[26] D. Prakash, A. Tripathi and T. K. Sharma, "Application of shuffled frog-leaping algorithm in regional air pollution control, " in *Soft Computing: Theories and Applications*, M. Pant, K. Ray, T. K. Sharma, S. Rawat, A. Bandyopadhyay, 1st ed., Singapore: Springer-Singapore, pp. 397–403, 2018

[27] J. Vijaya Kumar and D. M. Vinod Kumar, "Generation bidding strategy in a pool based electricity market using shuffled frog leaping algorithm," *Applied Soft Computing*, vol. 21, no. 5, pp. 407–414, 2014.

[28] K. K. Bhattacharjee and S. P. Sarmah, "Shuffled frog leaping algorithm and its application to 0/1 knapsack problem," *Applied Soft Computing*, vol. 19, no. 1, pp. 252–263, 2014.

[29] N. Ben Ammar, S. Bouallègue and J. Haggège, "Modeling and sliding mode control of a quadrotor unmanned aerial vehicle," in *Proc. of the 3th ACECS*, Hammamet, Tunisia, pp. 834–840, 2016.

[30] N. Ben Ammar, S. Bouallègue, J. Haggège and S. Vaidyanathan, "Chattering free sliding mode controller design for a quadrotor unmanned aerial vehicle," in *Applications of Sliding Mode Control in Science and Engineering*, S. Vaidyanathan, C. H. Lien, 1st ed., Germany: Springer-Verlag, pp. 61–79, 2017.

[31] J. E. Slotine and W. Li, *Applied Nonlinear Control*, 1st ed., New Jersey, USA: Prentice Hall, 1991.

[32] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers & Structures*, vol. 169, no. 2, pp. 1–12, 2016.

[33] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[34] Y. Mousavi and A. Alfi, "A memetic algorithm applied to trajectory control by tuning of fractional order proportional-integral-derivative controllers," *Applied Soft Computing*, vol. 36, no. 3, pp. 599–617, 2015.

[35] S. Bouallègue, F. Toumi, J. Haggège and P. Siarry, "Advanced metaheuristics-based approach for fuzzy control systems tuning," in *Complex System Modeling and Control Through Intelligent Soft Computations*, Q. Zhu and A. T. Azar, 1st ed., Germany: Springer-Verlag, pp. 627–653, 2015

Appendix A: Benchmark test functions

	Functions	Search space	Min.
Sphere	$f_1(x)=\sum_{i=1}^{d} x_i^2$	$[-100,100]^d$	0
Ackley	$f_2(x)=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)+20+\exp(1)$	$[-32,32]^d$	0
Rastrigin	$f_3(x)=10d+\sum_{i=1}^{d}\left(x_i^2-10\cos(2\pi x_i)\right)$	$[-5.12,5.12]^d$	0
Rosenbrock	$f_4(x)=\sum_{i=1}^{d-1}\left(100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right)$	$[-10,10]^d$	0
Schaffer 2	$f_5(x)=0.5+\frac{\sin^2(x_1^2-x_2^2)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	$[-100,100]^2$	0
Matyas	$f_6(x)=0.26\left(x_1^2+x_2^2\right)-0.48x_1x_2$	$[-10,10]^2$	0

Appendix B: Quadrotor's model parameters

Symbol	Description	Value/unit
b	Lift coefficient	2.984×10^{-5} $\mathrm{N}\cdot\mathrm{s}^2/\mathrm{rad}^2$
d	Drag coefficient	$3.30\times10^{-7}\mathrm{N}\cdot\mathrm{s}^2/\mathrm{rad}^2$
m	Mass	0.5 Kg
l	Arm length	0.50 m
J_r	Motor inertia	2.8385×10^{-5} $\mathrm{Kg\,m}^2$
J	Quadrotor inertia	$diag\,(0.005, 0.005, 0.010)$
$\kappa_{1,2,3}$	Aerodynamic friction coefficients	0.3729
$\kappa_{4,5,6}$	Translational drag coefficients	5.56×10^{-4}
g	Acceleration of the gravity	9.81 $\mathrm{m}\cdot\mathrm{s}^{-2}$