robotics

Article

Evaluation of Hunting-Based Optimizers for a Quadrotor Sliding Mode Flight Controller

Josenalde Oliveira [1,2], Paulo Moura Oliveira [2,3] and José Boaventura-Cunha [2,3] and Tatiana Pinho [2,*]

[1] Agricultural School of Jundiaí, Federal University of Rio Grande do Norte, UFRN, Macaíba, RN 59280-000, Brazil; josenalde@eaj.ufrn.br
[2] INESC TEC—INESC Technology and Science, Campus da FEUP, 4200-465 Porto, Portugal; oliveira@utad.pt (P.O.); jboavent@utad.pt (J.B.-C.)
[3] University of Trás-os-Montes and Alto Douro, School of Sciences and Technology, 5000-801 Vila Real, Portugal
[*] Correspondence: tatiana.m.pinho@inesctec.pt

Received: 30 January 2020; Accepted: 2 April 2020; Published: 7 April 2020

Abstract: The design of Multi-Input Multi-Output nonlinear control systems for a quadrotor can be a difficult task. Nature inspired optimization techniques can greatly improve the design of non-linear control systems. Two recently proposed hunting-based swarm intelligence inspired techniques are the Grey Wolf Optimizer (GWO) and the Ant Lion Optimizer (ALO). This paper proposes the use of both GWO and ALO techniques to design a Sliding Mode Control (SMC) flight system for tracking improvement of altitude and attitude in a quadrotor dynamic model. SMC is a nonlinear technique which requires that its strictly coupled parameters related to continuous and discontinuous components be correctly adjusted for proper operation. This requires minimizing the tracking error while keeping the chattering effect and control signal magnitude within suitable limits. The performance achieved with both GWO and ALO, considering realistic disturbed flight scenarios are presented and compared to the classical Particle Swarm Optimization (PSO) algorithm. Simulated results are presented showing that GWO and ALO outperformed PSO in terms of precise tracking, for ideal and disturbed conditions. It is shown that the higher stochastic nature of these hunting-based algorithms provided more confidence in local optima avoidance, suggesting feasibility of getting a more precise tracking for practical use.

Keywords: ant lion optimizer; grey wolf optimizer; sliding mode control; particle swarm optimization; quadrotor; unmanned aerial vehicle

1. Introduction

QuadRotor (QR) is a small rotorcraft which can be remotely controlled or fly autonomously through GPS-based embedded flight plans. Besides educational and leisure applications, its professional use has been increasing for intervention in hostile environments, remote sensing, precision agriculture, and urban planning [1]. This system is nonlinear, strongly coupled, Multiple-Input Multiple-Output (MIMO), and underactuated. Besides, since its operation is subject to external disturbances, robust control strategies have been investigated. Although some approaches are based on linear approximations and Proportional-Integral-Derivative (PID) controllers [2,3], most controllers are based on nonlinear algorithms and tools, such as feedback linearization, backstepping, and Sliding Mode Control (SMC). SMC [4] is a robust control technique, based upon a switching control law and the definition of a sliding surface, function of the system state variables. In [5], backstepping and SMC have shown to be equivalent from a robustness point of view, with SMC presenting a smoother and faster response. This has motivated research on both continuous and discrete-time SMC for quadrotor control [6–8]. However, these works do not present optimization routines or tuning rules for the controller parameters. In [9], a Hurwitz-based procedure was focused on the exploration of the nonlinear coefficients of the sliding manifolds, without optimization. Usually, these QR controllers present coupled design parameters with inherent trade-off and, therefore, need to be optimally tuned to achieve simultaneous performance requirements, which is a non-trivial trial-and-error task. Although some statistic-based [10] and bio-inspired [3,11,12], optimization algorithms have been used for PID and backstepping based flight control systems, a solution for the SMC-based approach is still an open and necessary research issue [13]. This has motivated the novel work reported here on the proposal and evaluation of hunting-based search algorithms as optimization tools, compared to the known and classical bio-inspired Particle Swarm Optimization (PSO) [14]. The choice of such techniques relies on their dominant stochastically behavior, since more random operators and population subgroups (or population split) are included and, thus, one has the improved ability of escaping from local optima through the search process. Although other similar strategies are available, previous experience [15], recent results [16], application diversity, and continuous improvement on both Ant Lion (ALO) [17,18] and Grey Wolf Optimizers (GWO) [19,20] encourage and motivate their application. Regarding mobile robotics, besides an improved version for ALO [21], GWO has also been recommended for mission (and trajectory) planning [22], even with moving obstacles [23]. ALO is known for its interesting performance indicators when tested with benchmark function sets [17], where local optima avoidance, better convergence speed, and higher accuracy were achieved, when compared to PSO [14], Genetic Algorithms, and Cuckoo Search. ALO mimics the hunting strategy of the ant lion and defines two population groups: ants and ant lions (predator—prey). Likewise, GWO mimics how a population of wolves chase a prey, inspired in its social hierarchy. Both have been evaluated for different applications, such as power systems control [24–27], process planning [28], and data gathering [29], as well as their multi-objective approaches [30,31]. This fact corroborates their application potential. However, although these hunting-based approaches are still evolving with several variations available, applications to controller tuning problem are minor [32], mainly available for PID controller design [33]. Moreover, no comparison between GWO and ALO is available for SMC-based flight system and thus it defines another contribution of this research. Therefore, this work evaluated the use of ALO and GWO to adjust the SMC parameters of a quadrotor altitude and attitude controller, and the simulation results were compared with the PSO considering the main objective of setpoint tracking. The main highlights and novelty may be summarized as follows:

- First evaluation of SMC parameter optimization for quadrotor flight system with hunting-based algorithms ALO and GWO
- Parameters obtained by ALO and GWO provided more confidence and repeatability during optimization process
- Parameters obtained by ALO and GWO provided lower tracking error
- Novel extension of such optimization approaches to SMC controller tuning, usually applied to PID control

2. Quadrotor Dynamics

A quadrotor consists of a rigid body frame equipped with four rotors, as shown in Figure 1, where the following assumptions are made: (1) the center of gravity is fixed at the origin of the quadrotor B and it coincides with the inertial axis E; (2) its structure is symmetrical; and (3) thrust and drag constants are proportional to the square value of the motor’s speed.

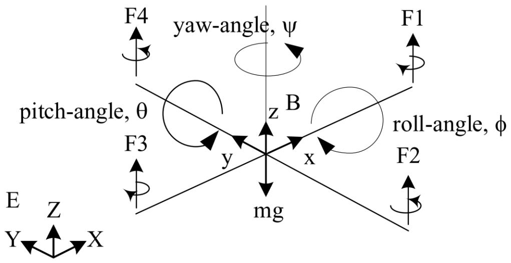

Figure 1. Quadrotor representation.

The full quadrotor Newton–Euler dynamic model in Equation (4) represents the x, y, z motions as a consequence of a roll (ϕ), pitch (θ) or yaw (ψ) rotation. A detailed quadrotor physical model derivation may be seen in [34]. Lift forces F_1, F_2, F_3, F_4 are basically the thrust generated by each propeller and they are related to the input signals U_i, $i = 1 \ldots 4$. This vehicle can be controlled and stabilized by altering rotors' speed $\omega_i, i = 1 \ldots 4$. By varying these speeds, one can change the lift forces and create motion. The collective input U_1 is the sum of each DC motor thrust. By defining the state variables as:

$$X = \left(x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}\right)^T \in \mathbb{R}^{12} \tag{1}$$

and considering that $x = x_1, \dot{x} = x_2, y = x_3, \dot{y} = x_4, z = x_5, \dot{z} = x_6, \phi = x_7, \dot{\phi} = x_8, \theta = x_9, \dot{\theta} = x_{10}, \psi = x_{11}, \dot{\psi} = x_{12}$, Equation (1) may be written as:

$$X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})^T, \tag{2}$$

with the control input vector U:

$$U = (U_1, U_2, U_3, U_4)^T \tag{3}$$

the full state-space system may be split into translational and rotational subsystems (see Table 1),with the model parameters of a micro Vertical Take-Off (VTOL) flying robot proposed in [34], and detailed in Table 2.

Table 1. Quadrotor dynamical subsystems.

Subsystems	
Translational	**Rotational**
$X_\Delta = \left(x, \dot{x}, y, \dot{y}, z, \dot{z}\right)^T \in \mathbb{R}^6$	$X_\Theta = \left(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}\right)^T \in \mathbb{R}^6$
$X_\Delta = (x_1, x_2, x_3, x_4, x_5, x_6)^T$	$X_\Theta = (x_7, x_8, x_9, x_{10}, x_{11}, x_{12})^T$
$U_\Delta = U_1$	$U_\Theta = (U_2, U_3, U_4)^T$
$\dot{X_\Delta} = f(X_\Delta, U_\Delta) = f_\Delta$	$\dot{X_\Theta} = f(X_\Theta, U_\Theta) = f_\Theta$

Table 2. Physical parameters—dynamical model.

Parameter	**Description**	**Value**
m	total quadrotor mass	0.52 kg
g	gravity acceleration	9.8 ms^{-2}
L	quadrature arm length	0.235 m
d	propeller resistance coefficient (drag factor)	7.5×10^{-7}
b	propeller lift coefficient (thrust factor)	3.13×10^{-5}
J_r	moment of propeller inertia around Z axis	8.66×10^{-7}
I_x	moment of inertia around X axis	6.228×10^{-3} kgm^2
I_y	moment of inertia around Y axis	6.228×10^{-3} kgm^2
I_z	moment of inertia around Z axis	1.121×10^{-2} kgm^2

Therefore, using the notation presented in Table 1, the quadrotor model described in [34] may be written as:

$$\dot{X} = f\left(X, U\right) = \begin{bmatrix} f_\Delta \\ f_\Theta \end{bmatrix} = \begin{bmatrix} x_2 \\ \left(\cos x_7 \sin x_9 \cos x_{11} + \sin x_7 \sin x_{11}\right) \dfrac{U_1}{m} \\ x_4 \\ \left(\cos x_7 \sin x_9 \sin x_{11} - \sin x_7 \cos x_{11}\right) \dfrac{U_1}{m} \\ x_6 \\ -g + \left(\cos x_7 \cos x_9\right) \dfrac{U_1}{m} \\ x_8 \\ x_{12} x_{10} \left(\dfrac{I_y - I_z}{I_x}\right) - \dfrac{J_r}{I_x} x_{10} \Omega_r + \dfrac{L}{I_x} U_2 \\ x_{10} \\ x_{12} x_8 \left(\dfrac{I_z - I_x}{I_y}\right) + \dfrac{J_r}{I_y} x_8 \Omega_r + \dfrac{L}{I_y} U_3 \\ x_{12} \\ x_{10} x_8 \left(\dfrac{I_x - I_y}{I_z}\right) + \dfrac{U_4}{I_z} \end{bmatrix} \tag{4}$$

where Ω_r is considered a disturbance associated to the difference between clockwise and counterclockwise propellers' rotation in Figure 1, defined as $\Omega_r = \omega_2 - \omega_1 + \omega_4 - \omega_3$. The control inputs U_i and rotors' speeds ω_i are related by the following equation:

$$\underbrace{\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}}_{U} = \underbrace{\begin{bmatrix} b & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & d \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 1 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}}_{\Omega}, \Omega = US^{-1}D^{-1} \tag{5}$$

3. Altitude and Attitude Control—Sliding Mode Flight System

This design closely follows Herrera et al. [13], and it is based on the equivalent control method [35]. A Lyapunov-based stability analysis may be found in [36]. First, a Proportional-Derivative sliding surface $S = \dot{e} + \lambda e$ is defined for each process variable (z, ϕ, θ, ψ) and respective tracking error $(e_z, e_\phi, e_\theta, e_\psi)$, to be reached by the trajectories in finite time. In this condition, named *sliding mode*, $\dot{S} = 0$ and a non-switching control signal U_{eq}, referred as *equivalent control*, is designed. The second step consists on designing the control law U_{sm}, which drives the process variable to the sliding surface, during the *reaching mode*, and it must obey a *reaching law*, considered here as $S\dot{S} < \eta |S|$, $\eta > 0$. Therefore, considering Equation (4) one has for the altitude z,

$$e_z = z_R - z = z_R - x_5 \tag{6}$$

$$S_z = \dot{e_z} + \lambda_1 e_z = \dot{z_R} - x_6 + \lambda_1 \left(z_R - x_5\right) \tag{7}$$

$$\dot{S}_z = 0, \quad \dot{S}_z = \ddot{e_z} + \lambda_1 \dot{e_z} = 0, \quad \ddot{z_R} - \dot{x_6} + \lambda_1 \left(\dot{z_R} - x_6\right) = 0 \tag{8}$$

$$U_{1_{eq}} = \left(g - \breve{\ }\mathbf{1}\left(x_6 - \dot{z}_R\right) + \ddot{z}_R\right) \frac{m}{\cos x_7 \cos x_9} \tag{9}$$

$$U_{1_{sm}} = k_1 \frac{S_z}{|S_z| + \delta_1}, \quad U_1 = U_{1_{eq}} + U_{1_{sm}} \tag{10}$$

and, following a similar procedure for roll, pitch, and yaw angles, one has the expressions presented in Table 3.

Table 3. Roll, pitch, and yaw control signals.

Roll
$e_{\phi} = \phi_R - \phi = \phi_R - x_7$
$S_{\phi} = \dot{e_{\phi}} + \lambda_2 e_{\phi} = \dot{\phi_R} - x_8 + \lambda_2 \left(\phi_R - x_7\right)$
$U_{2_{eq}} = \left[\ddot{\phi_r} - x_{10}x_{12}\left(\frac{I_y - I_z}{I_x}\right) + \frac{J_r}{I_x}x_{10}\Omega_r + \lambda_2\left(\dot{\phi_r} - x_8\right)\right]\frac{I_x}{L}$
$U_{2_{sm}} = k_2\frac{S_{\phi}}{\lvert S_{\phi}\rvert + \delta_2}, \quad U_2 = U_{2_{eq}} + U_{2_{sm}}$
Pitch
$e_{\theta} = \theta_R - \theta = \theta_R - x_9$
$S_{\theta} = \dot{e_{\theta}} + \lambda_3 e_{\theta} = \dot{\theta_R} - x_{10} + \lambda_3\left(\theta_R - x_9\right)$
$U_{3_{eq}} = \left[\ddot{\theta_r} - x_8 x_{12}\left(\frac{I_z - I_x}{I_y}\right) - \frac{J_r}{I_y}x_8\Omega_r + \lambda_3\left(\dot{\theta_r} - x_{10}\right)\right]\frac{I_y}{L}$
$U_{3_{sm}} = k_3\frac{S_{\theta}}{\lvert S_{\theta}\rvert + \delta_3}, \quad U_3 = U_{3_{eq}} + U_{3_{sm}}$
Yaw
$e_{\psi} = \psi_R - \psi = \psi_R - x_{11}$
$S_{\psi} = \dot{e_{\psi}} + \lambda_4 e_{\psi} = \dot{\psi_R} - x_{12} + \lambda_4\left(\psi_R - x_{11}\right)$
$U_{4_{eq}} = \left[\ddot{\psi_r} - x_8 x_{10}\left(\frac{I_x - I_y}{I_z}\right) + \lambda_4\left(\dot{\psi_r} - x_{12}\right)\right] I_z$
$U_{4_{sm}} = k_4\frac{S_{\psi}}{\lvert S_{\psi}\rvert + \delta_4}, \quad U_4 = U_{4_{eq}} + U_{4_{sm}}$

From Equations (9) and (10) and Table 3 expressions for the control components $U_{i_{eq}}, U_{i_{sm}}$ $(i = 2, 3, 4)$, it can be seen that twelve parameters must be adjusted, namely:

$$P = \{\lambda_1, k_1, \delta_1, \lambda_2, k_2, \delta_2, \lambda_3, k_3, \delta_3, \lambda_4, k_4, \delta_4\} \tag{11}$$

detailed in Table 4.

Table 4. Controller parameters to be optimized.

Control Component	**Controller Parameters**			
	Altitude z	**Roll** ϕ	**Pitch** θ	**Yaw** ψ
U_{eq}	λ_1	λ_2	λ_3	λ_4
U_{sm}	k_1, δ_1	k_2, δ_2	k_3, δ_3	k_4, δ_4

When the controller parameters set in Equation (11) is badly tuned, non-negligible tracking error may occur. Besides, high frequency oscillations (chattering) and control signal saturation must be kept within feasible limits. This situation is undesirable in practice [8] as it increases the risk of actuators and motors damage, as shown in Figure 2, for thrusts U_2 and U_3. Therefore, an optimal (or suboptimal) set of parameters is desired to minimize some suitable fitness function. The next section presents ALO and GWO algorithms as selected optimization metaheuristics to be evaluated within the specific problem stated in this work.

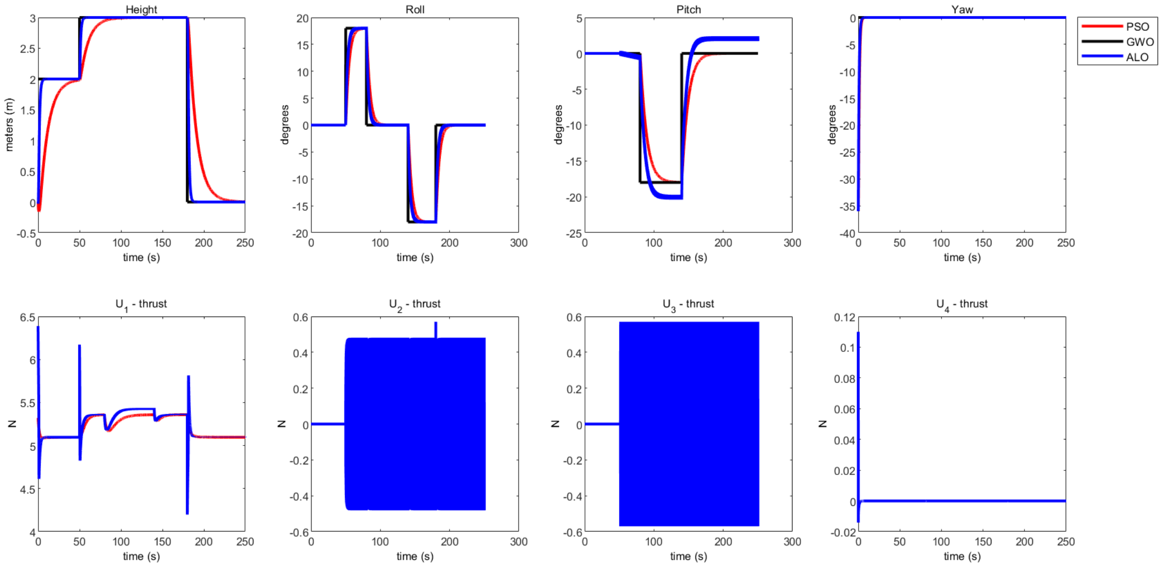

Figure 2. Example of bad tuning effect (chattering) on input signals.

4. Hunting-Based Optimizers

4.1. Ant Lion Optimizer

ALO is a nature-inspired stochastic optimization algorithm which mimics the hunting mechanism of ant lions (predator), that is, how they build pits (traps) where ants (prey) slide in and are caught. It has presented competitive results when compared to PSO [17], in terms of improved exploration, local optima avoidance, exploitation, and convergence, to solve constrained problems with diverse search spaces. The exploration (global search) is guaranteed by the random selection of ant lions to some extent and the random walks of ants around ant lions, so the probability of avoiding local optima is high [21]. The exploitation (local search) is ensured by shrinking size of ant lions traps and the promising search is saved by the elite ant lion, so the convergence accuracy is good. The main interpretation is that ant lions can build pits proportional to their fitness (the higher the fitness, the higher the pit). Its stochastic behavior also results in obtaining different solutions in each independent run, and therefore the final parameters and convergence curve represent the mean value of the population. The ALO steps are detailed next.

4.1.1. Random Walk of Ants

The basic idea of ALO is how to model ants' random walks around the ant lions and keep them inside the search space, which is achieved by a *min-max* normalization, described by:

$$X^t = [0, cs(2r(t_1)-1), cs(2r(t_2)-1), \ \ldots, cs(2r(t_T)-1)] \tag{12}$$

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i^t \tag{13}$$

where cs is the cumulative sum; t is the iteration of the random walk; T is the total number of iterations; $r(t)$ is a function defined as 1 if rnd > 0.5 or 0 if rnd ≤ 0.5; rnd is generated with uniform distribution in the interval $[0,1]$; a_i and b_i are the min and max of random walk of the variable i; and d_i^t and c_i^t are the upper bound and lower bounds of variable i at iteration t.

4.1.2. Ant Lions Building Traps

To build a trap and associate every ant with an ant lion, a roulette wheel mechanism selects an ant lion based on its fitness and the trapping in its pit is modeled as

$$c_i^t = Antlion_j^t + c^t, \qquad d_i^t = Antlion_j^t + d^t \tag{14}$$

with c^t and d^t as the minimum and maximum for the entire ant vector, considering all dimensional variables, at iteration t. $Antlion_j^t$ is the position of the selected ant lion j at iteration t.

4.1.3. The Entrapment of Ants in Traps

To move ants towards ant lions, the radius I of their random walks is decreased, reducing the level of exploitation as the iterations t increase,

$$c^t = \frac{c^t}{I}, \qquad d^t = \frac{d^t}{I}, \qquad I = 10^w \left(\frac{t}{T}\right), \tag{15}$$

with

$$w = \begin{cases} 1 & \Leftarrow & t \leq 0.1T \\ 2 & \Leftarrow & 0.1T < t \leq 0.5T \\ 3 & \Leftarrow & 0.5T < t \leq 0.75T \\ 4 & \Leftarrow & 0.75T < t \leq 0.9T \\ 5 & \Leftarrow & 0.9T < t \leq 0.95T \\ 6 & \Leftarrow & t > 0.95T \end{cases} \tag{16}$$

4.1.4. Ant Lions Catching Ants and Re-Building Traps

The action of catching a prey (ant) and rebuilding the pit is represented by replacing each ant lion with its corresponding ant, if it becomes fitter (see Equation (17)).

$$Antlion_j^t = Ant_i^t \quad \text{if } f\left(Ant_i^t\right) < f\left(Antlion_j^t\right) \tag{17}$$

Moreover, ALO has an elitism scheme, where the best of all ant lions is the elite solution represented as R_E. Every ant movement is affected by using an average between a random walk performed around an ant lion selected using the roulette wheel scheme, R_A, and a random walk around the best ant lion (elite), R_E. This is represented by Equation (18):

$$Ant_i^t = \frac{R_A^t + R_E^t}{2}. \tag{18}$$

The ALO pseudocode is presented in Algorithm 1.

Algorithm 1: ALO algorithm pseudocode.

```
begin
    Randomly initialize the populations of ants and ant lions;
    Calculate the fitness of ants and ant lions;
    Find the best of all antlions and assume it as the elite (determined optimum);
    while not (termination criterion) do
        for each search agent—ant do
            Select an ant lion using Roulette wheel;
            Update $c$ and $d$ using (15);
            Create a random walk and normalize it using Equations (12)–(13);
            Update the ant position using Equation (18);
        end
        Evaluate the fitness of all ants;
        Replace each ant lion with its corresponding ant if it becomes fitter (Equation (17));
        Update elite if the ant lion becomes fitter than current elite;
    end
end
```

4.2. Grey Wolf Optimizer

The GWO algorithm, as proposed by Mirjalili et al. [19], is inspired by the behavior of a wolf pack social hierarchy. An analogy is established between a set of potential solutions for a given problem and a population of wolves chasing a prey. Following the wolf pack social hierarchy, the GWO establishes four organization levels:

- Alpha (α) are dominant wolves and thus followed by the rest of the pack.
- Beta (β) are second in command helping alphas in the decision process and establish a bridge between alphas and the lower levels.
- Delta (δ) are third in the pack hierarchy; while submitted to alphas and betas, they submit the lowest rank, which is called omega. Deltas represent wolves such as scouts, sentinels, elders, hunters, and caretakers.
- Omega (ω) represent the rest of population solutions.

The three higher rank elements (α, β, and δ) movement mimics the prey encirclement by wolves, and it is modeled by:

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)|, \quad with \quad \vec{C} = c_1\vec{r}_2, \tag{19}$$

where t represents the current iteration, X_p is the prey position vector, X is a grey wolf position, and C is a coefficient vector, which is evaluated using a uniformly random vector r_2 generated in the interval $[0,1]$. c_1 can be an adjustable constant proposed in [19] as $c_1 = 2$. The result difference vector D is then used to move the specific element towards or away from the region where the best solution is located (the prey) using:

$$\vec{X}(t+1) = \vec{X}_p(t) - A\vec{D}, \quad with \quad a \ = 2a(t)r_1 - a(t), \tag{20}$$

where r_1 represents a uniformly random generated number in the interval $[0,1]$ and a is linearly decreased vector from $a_{max} = 2$ to $a_{min} = 0$ throughout the predefined number of iterations. If the absolute value of A is smaller than 1, this corresponds to an exploitation behavior and mimics the wolf attacking the prey. Otherwise, if the absolute value of A is larger than 1, this corresponds to an exploration behavior and mimics the wolf diverging from the prey. The proposed values by Mirjalili et al. [19] for A are in the interval of $[-2,2]$. Thus, applying the general expressions (Equations (19) and (20)) to the higher wolfs ranks (α, β, and δ) results, respectively, in:

$$\vec{D}_\alpha = |\vec{C}_1\vec{X}_\alpha(t) - \vec{X}(t)|, \quad \vec{X}_1 = \vec{X}_\alpha - A_1\vec{D}_\alpha \tag{21}$$

$$\vec{D}_\beta = |\vec{C}_2 \vec{X}_\beta(t) - \vec{X}(t)|, \quad \vec{X}_2 = \vec{X}_\beta - A_2 \vec{D}_\beta \tag{22}$$

$$\vec{D}_\delta = |\vec{C}_3 \vec{X}_\delta(t) - \vec{X}(t)|, \quad \vec{X}_3 = \vec{X}_\delta - A_3 \vec{D}_\delta \tag{23}$$

In Equations (21)–(23), the prey position corresponds, respectively, to the best position attained by each of the wolf ranks (α, β, and δ). All wolf population positions are updated using the following expression:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{24}$$

The GWO pseudocode is presented in Algorithm 2.

Algorithm 2: Pseudocode for the GWO algorithm.

```
begin
    Initialize the grey wolf population $X(t)$;
    Initialize $a$;
    Evaluate each search agent—wolf fitness;
    $X_\alpha$ = the best search agent;
    $X_\beta$ = the second best search agent;
    $X_\delta$ = the third best search agent;
    while not (termination criterion) do
        for each search agent do
            Update the position of the current search agent using Equation (24);
        end
        Update $a$, $A$ and $C$;
        Evaluate all search agents fitness;
        Update $X_\alpha$, $X_\beta$ and $X_\delta$ using Equation (21)–(23);
    end
end
```

5. Simulations and Discussion

All simulations were carried out on an Intel Core i7 3.4GHz, 8 GB DDR3 RAM, 500 GB Hard Disk Drive, Windows 7 64bits, Matlab® R-2014b and represent a time of 300 s. A sampling frequency of $f_s = 30$ Hz was selected, within the feasible sample time limit for a small quadrotor with a diameter around 50 cm [37].

5.1. Fitness Function and Optimization Methodology

ALO and GWO were used to improve iteratively a set of random potential solutions encoding in Equation (11) by minimizing a representative aggregated cost function in Equation (25), which equally balances the objectives of set point tracking and control signal variation. The results were compared to PSO subject to the same cost function. Although many PSO variants are available [38], the conventional PSO [14] is the common choice for performance comparison [17,19] and it is already available as an optimization solver in software such as Matlab®. The fitness of each potential solution (particle, ant, or wolf) were evaluated using Equation (25) considering a full quadrotor flight (see Figure 3) which explores all common VTOL movements (vertical take-off and landing and curves) during 300 s (see Table 5). Although no trajectory benchmark is available, most works consider rectangular, helical, elliptical, or mixed paths [9,13], in such a way that the main VTOL movements be explored. It is noteworthy and required that the same reference trajectory be used for all optimization algorithms. The proposed general fitness function J is defined as:

$$J = F_J^T W, \tag{25}$$

where,

$$F_J = \begin{bmatrix} J_1 & J_2 & J_3 & J_4 \end{bmatrix}^T, \quad W = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix}^T \tag{26}$$

Each component $J_i, \quad i = 1, \ldots, 4$ is given by

$$J_1 = \epsilon_{11} \sum \left| e_z^2 \right| + \epsilon_{12} \sum |U_1(k+1) - U_1(k)| \tag{27}$$

$$J_2 = \epsilon_{21} \sum \left| e_\phi^2 \right| + \epsilon_{22} \sum |U_2(k+1) - U_2(k)| \tag{28}$$

$$J_3 = \epsilon_{31} \sum \left| e_\theta^2 \right| + \epsilon_{32} \sum |U_3(k+1) - U_3(k)| \tag{29}$$

$$J_4 = \epsilon_{41} \sum \left| e_\psi^2 \right| + \epsilon_{42} \sum |U_4(k+1) - U_4(k)| \tag{30}$$

with e_z, e_ϕ, e_θ, and e_ψ defined as in Equation (6) and Table 3. To equally balance all components, $w_i = 0.25 \quad (i = 1, \ldots, 4)$ and $\epsilon_{ij} = 0.5 \quad (i = 1, \ldots, 4; j = 1, 2)$. The control action constraints are defined by the motors specifications [34] with physical parameters presented in Table 2, as follows:

$$0 \leq U_1 \leq 9.67 \tag{31}$$

$$-0.57 \leq U_2 \leq 0.57 \tag{32}$$

$$-0.57 \leq U_3 \leq 0.57 \tag{33}$$

$$-0.11 \leq U_4 \leq 0.11 \tag{34}$$

Table 5. Trajectory used in the optimization with initial condition $\psi(0) = -\frac{\pi}{10}$.

Time Interval	z_R	ϕ_R	θ_R	ψ_R
$0 < t \leq 50s$	5	0	0	0
$50s < t \leq 80s$	3	$\frac{\pi}{5}$	0	0
$80s < t \leq 140s$	1	0	$-\frac{\pi}{5}$	0
$140s < t \leq 180s$	4	$-\frac{\pi}{3}$	0	0
$t > 180s$	0	0	0	0

The same conditions were applied to the PSO algorithm [14], with cognitive and social constants, $c_1 = c_2 = 2$ and inertia vector, ω_E, linearly decayed within the range $0.9 - 0.4$, along the number of iterations ($T = 250$). A pre-defined number of independent runs $N_{runs} = 30$ was executed for each algorithm, considering a suitable and representative choice to assure variability [17,19]. Populations with size $N = 100$ were considered. It was assumed that all initial populations are different among runs and algorithm type. The bounds for the common search space are:

$$P_{min} = (0.1, 0.1, 0.1, 0.5, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)^T \tag{35}$$

$$P_{max} = (10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10)^T \tag{36}$$

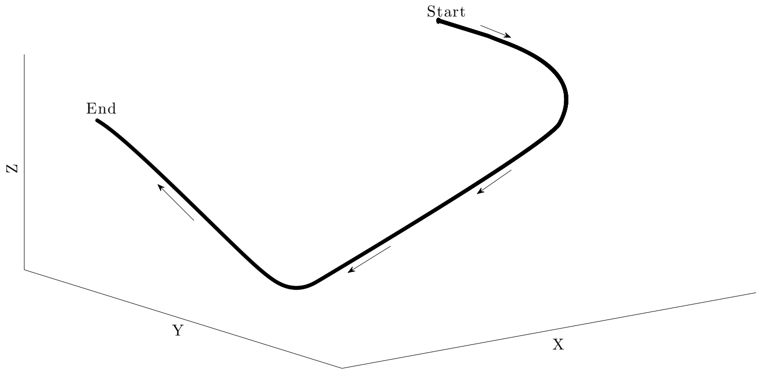

Figure 3. General trajectory used for optimization.

The optimization results achieved are presented in Table 6 for the fitness values of Equation (25) and in Table 7 for the controller parameters set. In Table 7, two sets are presented for each method: (i) the mean best values of all 30 runs; and(ii) the best set (which provided the best fitness value) throughout $N_{runs} = 30$ independent runs. The used indicators (metrics) for comparing the different algorithms are:

- statistical *Best* is the minimum fitness function value (F_i) (or best value) obtained in N_{runs}.

$$Best = min_{i=1}^{N_{runs}} F_i \tag{37}$$

- statistical *Worst* is the maximum fitness function value (F_i) (or worst value) obtained in N_{runs}.

$$Worst = max_{i=1}^{N_{runs}} F_i \tag{38}$$

- statistical *Median* is the middle fitness function value (F_i) in a sorted list (or median value) obtained in N_{runs}. If there are two middle numbers (N_{runs} is even), the median is their average.
- statistical *Mean* is the average performance of a stochastic algorithm applied N_{runs} times, where F_i^* is the optimal solution at the *i*th run.

$$Mean = \frac{1}{N_{runs}} \sum_{i=1}^{N_{runs}} F_i^* \tag{39}$$

- statistical *Standard deviation (Std)* indicates the optimizer stability and robustness, preferably as small as possible.

$$Std = \sqrt{\frac{1}{N_{runs} - 1} \sum (F_i^* - Mean)^2} \tag{40}$$

As shown in Figure 4a and by the *Best* measure in Table 6, the results achieved for PSO and GWO are almost the same. However, the exploratory behavior of GWO (see Figure 4b)) is more suitable to get a stable convergence (reflected by *Std* and *Mean* indexes), outperforming PSO. Although the *Best* value for ALO is slightly greater, its *Std* value suggests a higher probability of getting similar results at each independent run. An equivalent convergence rate for the population mean fitness value in all algorithms may be verified, around 200 iterations.

Table 6. Fitness values obtained in 30 runs.

Measure Fitness Value	**Method**		
	PSO	**ALO**	**GWO**
Best	822, 840	823, 860	822, 850
Worst	925, 520	864, 350	833, 150
Median	825, 060	836, 580	823, 350
Mean	838, 060	841, 590	823, 510
Std	0.0305	0.0153	0.0018

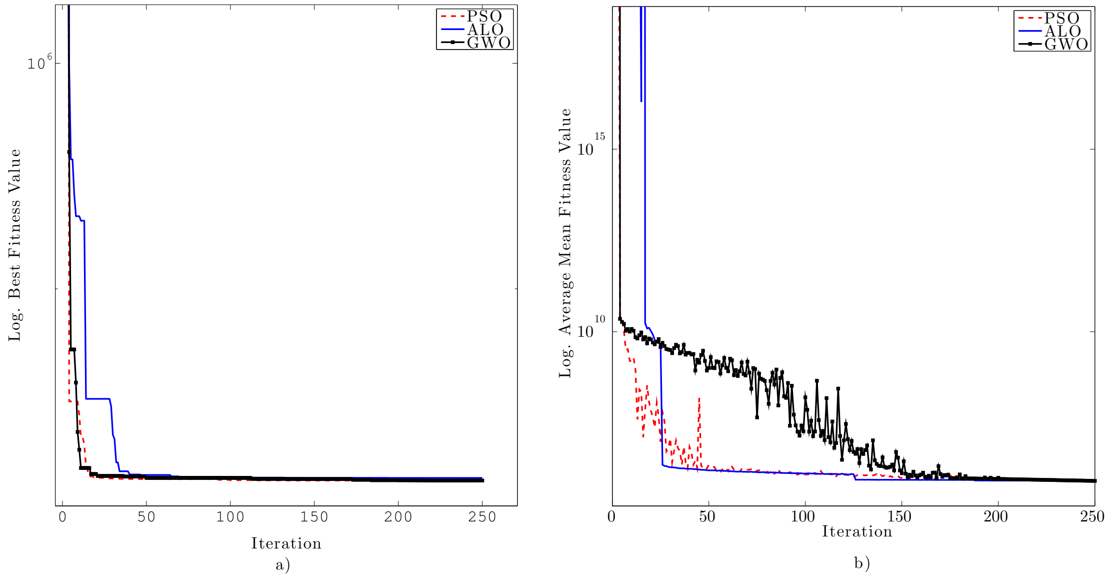

Figure 4. Convergence curves: (**a**) best fitness value; (**b**) average mean fitness value.

The required computational time is equivalent for PSO and GWO (35.49 s and 35.55 s, respectively) and sightly higher for ALO (37.26 s), for each iteration in the test hardware. Regarding the optimized parameters in Table 7, there is an interesting similarity among PSO, ALO, and GWO for the roll angle, since this combination provides the best fitness value for each algorithm. The values for the switching component U_{sm} converged to the boundary limits, with the lower bound 0.1 for the gain of this component and the upper bound for δ. The pair (k, δ) has an intrinsic trade-off and affects the disturbance rejection (robustness) and chattering. It is noteworthy that the highest λ found by ALO (using mean controller parameter set) for the roll and pitch angles presents better disturbance rejection properties in the flight simulation shown in the next section. A simple comparison between Figures 2 and 5 below gives a good understanding on how the used SMC technique reduces chattering.

Table 7. Resulting optimized controller parameters.

	Controller Parameters	Mean Values (30 runs)			Best Values (30 runs)		
		PSO	**ALO**	**GWO**	**PSO**	**ALO**	**GWO**
Height z	λ_1	0.24	0.26	0.27	0.27	0.26	0.27
	k_1	8.00	7.61	8.43	8.42	8.83	8.42
	δ_1	10.00	8.91	10.00	10.00	9.99	10.00
Roll ϕ	λ_2	2.40	3.67	0.82	0.5	0.50	0.50
	k_2	0.10	0.10	0.10	0.10	0.10	0.10
	δ_2	10.00	10.00	10.00	10.00	10.00	10.00
Pitch θ	λ_3	3.08	6.03	1.11	0.44	2.89	0.44
	k_3	1.16	1.29	0.38	0.10	0.13	0.10
	δ_3	6.12	7.55	7.62	1.54	8.80	1.68
Yaw ψ	λ_4	3.97	5.61	3.34	2.50	9.43	0.25
	k_4	0.14	1.13	0.73	0.10	0.54	1.05
	δ_4	9.08	7.67	5.64	10.00	7.18	8.68

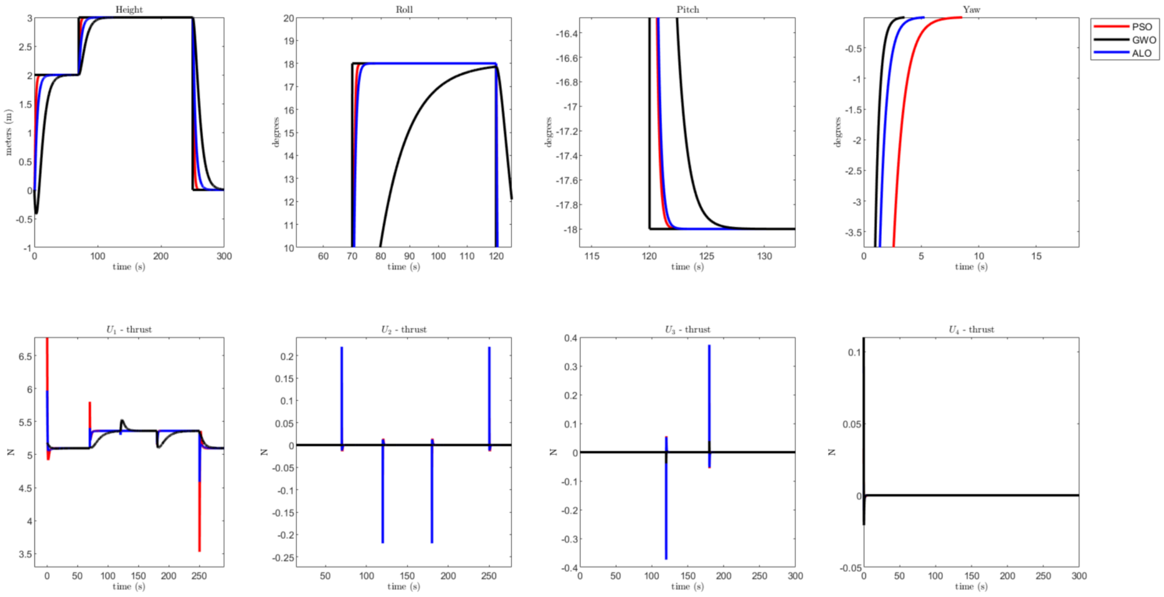

Figure 5. Example of SMC-based equivalent control method to reduce chattering.

5.2. Flight Simulation

With these parameters (Table 7), four Flight Plans (FP) were simulated (Table 8) to meet common movements such as vertical takeoff with yaw correction, hovering, maneuvering, and landing. Usually, such movements explore non-ideal conditions [6], such as constant input disturbances, parameter variations, motor failure, and noise, here simulated as a white Gaussian noise in the range of ±15 mV. A diversity of common paths (rectangular, elliptical, and helical) are also used [9,13]. These flight plans are described as follows, and, where not explicitly mentioned, all initial values are zero.

Table 8. Flight plans.

Flight Plan 1
$\psi(0) = -\frac{\pi}{5}$ $0 < t \leq 70s, \quad z_R = 3, \phi_R = \theta_R = \psi_R = 0$ $70s < t \leq 120s, \quad z_R = 3, \phi_R = \frac{\pi}{3}, \theta_R = \psi_R = 0$ $120s < t \leq 180s, \quad z_R = 3, \phi_R = 0, \theta_R = -\frac{\pi}{10}, \psi_R = 0$ $180s < t \leq 250s, \quad z_R = 3, \phi_R = -\frac{\pi}{10}, \theta_R = 0, \psi_R = 0$ $t > 250s, \quad z_R = \phi_R = \theta_R = \psi_R = 0$
Flight Plan 2—Parameter variation
$t > 0, \quad z_R = 6$ $80s \leq t \leq 130s$, Parameter variation of $+20\%$ in $m, J_r, I_x, I_y, I_z, d, b$
Flight Plan 3—Input disturbance and temporary motor failure
the same FP 1 with constant input disturbance of $+0.1$ in U_1 $80s \leq t \leq 130s, \quad U_2 = 0$
Flight Plan 4—Measurement noise
the same FP 1 with added noise in z, ϕ, θ, ψ

Below, four figures for each flight plan case (Figures 6–12) present the simulation results. For visualization purposes, the same legend is considered throughout the plots within a specific figure. The total performance indexes used in Tables 9 and 10 are defined as:

$$ISE_T = \sum e^2 = \sum {e_z}^2 + \sum {e_\phi}^2 + \sum {e_\theta}^2 + \sum {e_\psi}^2 \tag{41}$$

$$U_T = \sum |U| = \sum_{i=1}^{4} |U_i| \tag{42}$$

$$TV_U = \sum_{i=1}^{4} \sum_{k=1}^{f_s \times t_{sim}} (U_i(k+1) - U_i(k)) \tag{43}$$

$$Std_T = \sum_{i=1}^{4} Std(U_i) \tag{44}$$

where f_s is the sampling frequency and t_{sim} is the total simulation time. Std represents the usual standard deviation of Equation (40), for each control component U_i, $i = 1, \ldots, 4$.

Table 9. Performance comparison with mean controller parameters.

Flight Plan	Method	ISE_T	U_T	$\lvert TV_U \rvert$	std_T
FP1	PSO	1.58×10^3	47,240	6.17×10^3	0.13
	ALO	1.48×10^3	47,243	7.53×10^3	0.14
	GWO	1.46×10^3	47,229	7.73×10^3	0.13
FP2	PSO	3.11×10^3	47,398	9.77×10^3	0.38
	ALO	2.95×10^3	47,398	1.07×10^4	0.38
	GWO	2.87×10^3	47,398	1.11×10^4	0.38
FP3	PSO	4.26×10^3	46,690	6.68×10^3	0.20
	ALO	3.63×10^3	46,684	8.04×10^3	0.21
	GWO	3.54×10^3	46,852	8.05×10^3	0.24
FP4	PSO	1.57×10^3	47,265	6.22×10^3	0.13
	ALO	1.48×10^3	47,316	7.60×10^3	0.14
	GWO	1.47×10^3	47,367	7.62×10^3	0.15

Table 10. Performance comparison with controller parameters related to best fitness value.

Flight Plan	Method	ISE_T	U_T	$\lvert TV_U \rvert$	std_T
FP1	PSO	1.48×10^3	47,210	6.85×10^3	0.13
	ALO	1.46×10^3	47,215	8.06×10^3	0.13
	GWO	1.49×10^3	47,210	6.91×10^3	0.13
FP2	PSO	2.87×10^3	47,398	1.11×10^4	0.38
	ALO	2.85×10^3	47,398	1.16×10^4	0.38
	GWO	2.87×10^3	47,398	1.11×10^4	0.38
FP3	PSO	3.66×10^3	47,206	6.81×10^3	0.34
	ALO	3.48×10^3	47,212	8.01×10^3	0.35
	GWO	3.66×10^3	47,206	6.87×10^3	0.34
FP4	PSO	1.52×10^3	47,392	6.68×10^3	0.15
	ALO	1.49×10^3	47,433	7.87×10^3	0.15
	GWO	1.51×10^3	47,283	6.84×10^3	0.14

The feasibility of applying such hunting-based optimization algorithms can be supported by the results in Figures 6–13 and the data in Tables 9 and 10. All simulations used the mean controller parameters set presented in Table 6, which provides more diversity and suitability. At least one hunting method outperforms PSO in terms of general tracking (ISE_T): both ALO and GWO when the mean controller parameters (Table 6) are used (Table 9) and ALO when the choice is the controller parameter set related to best fitness values (Table 10). Although these benefits may require more oscillatory control signal (TV_U) when compared to PSO, the total control effort (U_T) is quite similar. Figures 6 and 7 show the ideal case with a zoom at the initial instants, to highlight that ALO provides a faster tracking for all process variables. The worst performance for GWO on pitch and roll control in Figures 6 and 12 is explained by the smaller parameters λ_2 and λ_3. However, since the quadrotor control must be considered as a whole, the ISE_T is still better for GWO when compared to PSO. For the disturbed cases in Figures 8–12, the overall best tracking for ALO and GWO is also verified. The zoom in the top plot of Figure 8 shows the best recovery achieved, when system physical parameters vary. If a motor failure is considered (Figure 10, roll plot), ALO outperforms PSO and GWO, for the same reason explained above for pitch and roll. Figure 12 presents only the altitude tracking under measurement noise, but the other process variables show similar behavior. In terms of tracking, both GWO and ALO are slightly better than PSO. This finding is in accordance to the results in [17,19,32,33]. In these two latter works, the Gravitational Search Algorithm [39] exceeds both, but here it is not considered since it is not an animal-behavior method, but inspired in physical motion laws. For an actual and digital implementation where robustness and stable convergence rate must be emphasized [7,8], GWO may be recommended in its original formulation, since it has presented a good balance between exploration and exploitation. Regarding ALO, nonsystematic adaptation methods of exploration rate along the search space have recently been proposed to outperform the original ALO [21]. However, here, this feature did not compromise the tracking results, due to the aggregated cost function type and since the control must be considered for the four motors as a whole. In Figures 7–13, for illustrative purposes, the trajectories in Table 8 are controlled using the SMC with the parameters found by the ALO (see Table 7). When the curves almost coincide with the set point, it is due to the SMC transient response, which is very fast and non-oscillatory, achieving the objectives of this work of getting a more precise tracking with equivalent and feasible control signal effort and chattering, when compared to PSO.

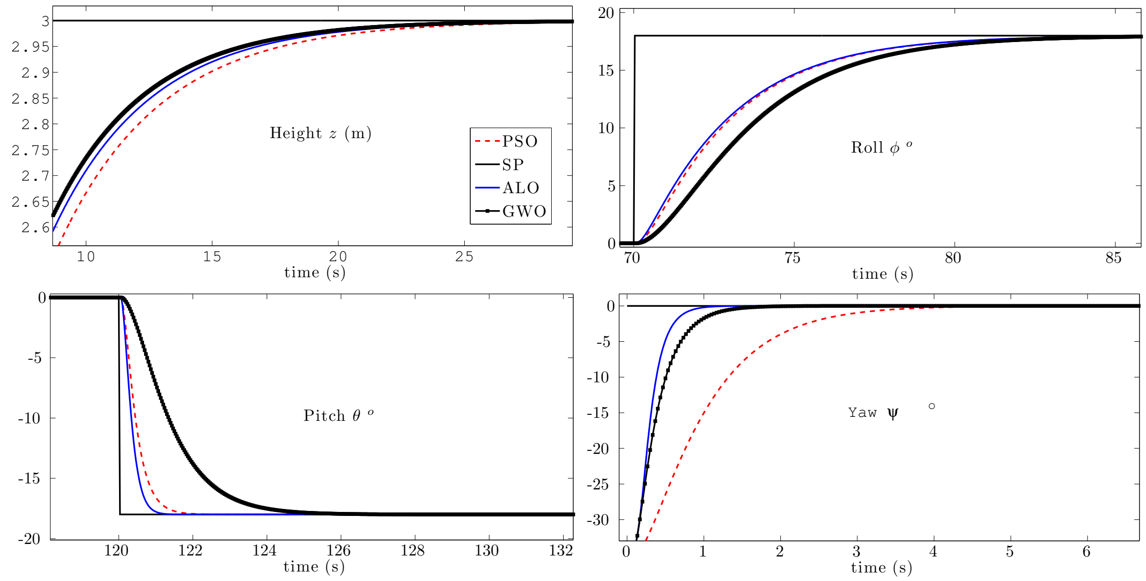

Figure 6. Flight plan 1 (zoom)—non-disturbed case.

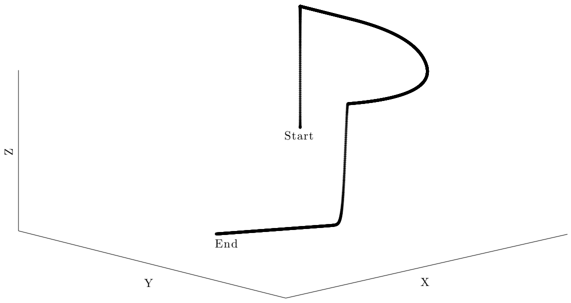

Figure 7. Flight plan 1—non-disturbed case.

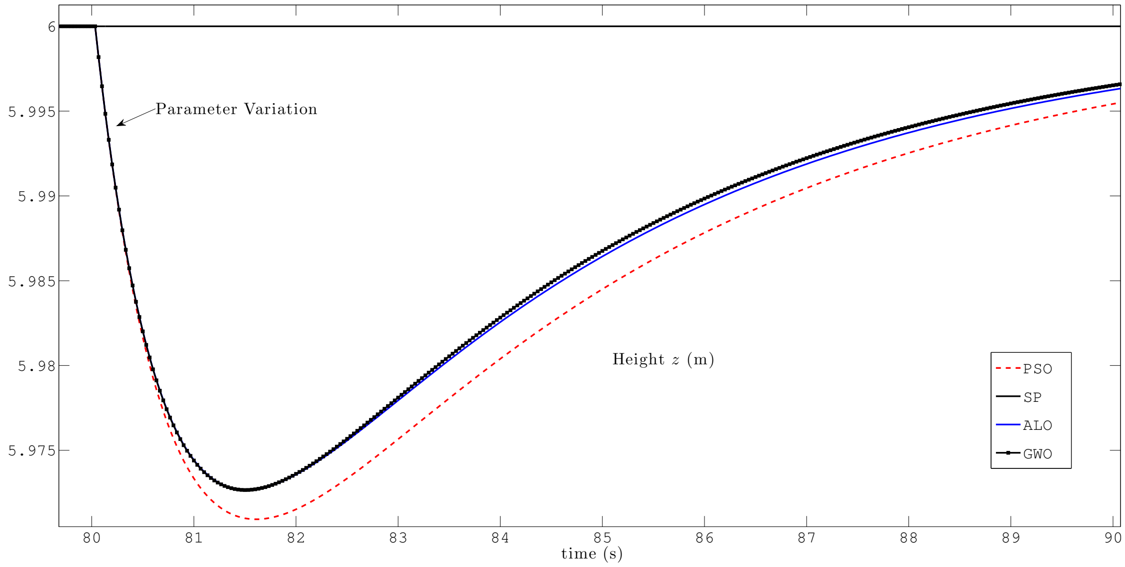

Figure 8. Flight plan 2—zoom of the effect of the parameter variation inserted at $t = 80s$.

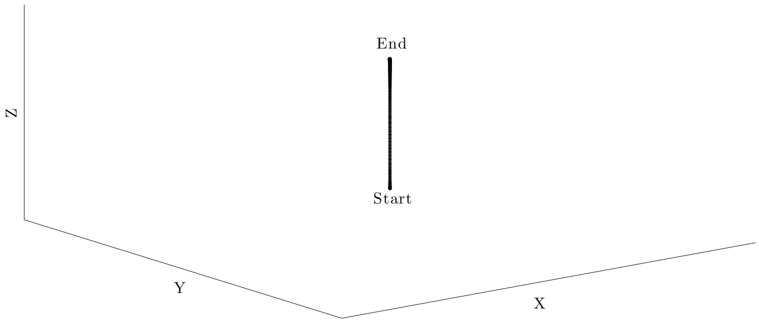

Figure 9. Flight plan 2—parametric variation.

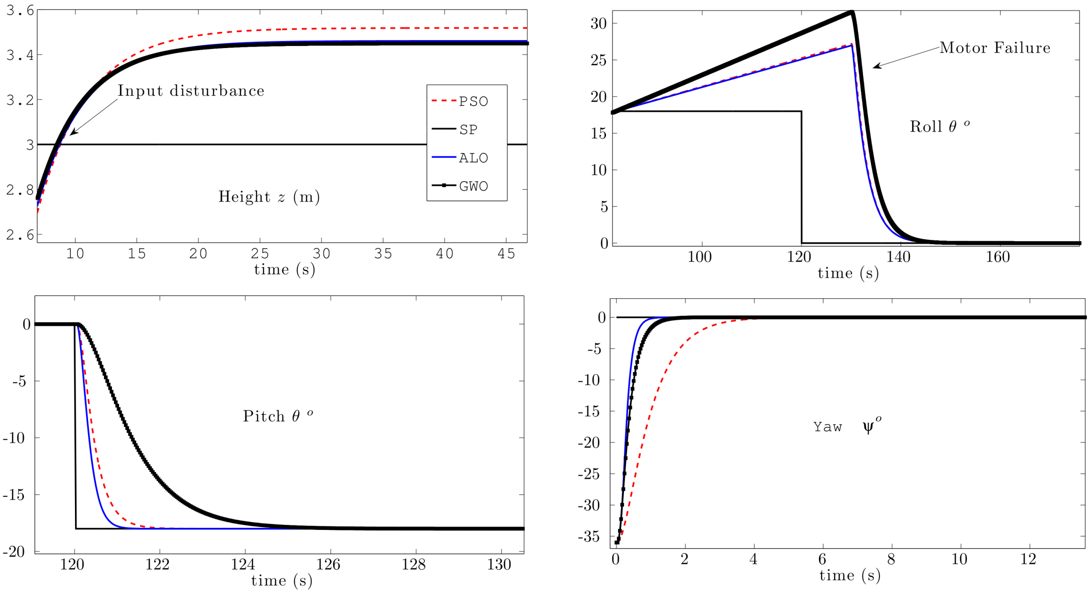

Figure 10. Flight plan 3—input disturbance in U_1 and motor failure in U_2.

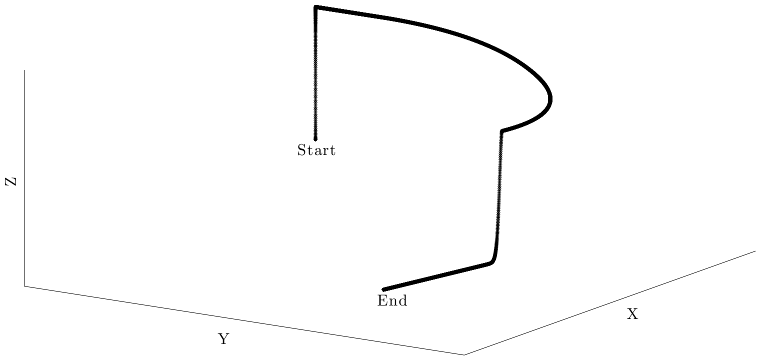

Figure 11. Flight plan 3—input disturbance and motor failure.

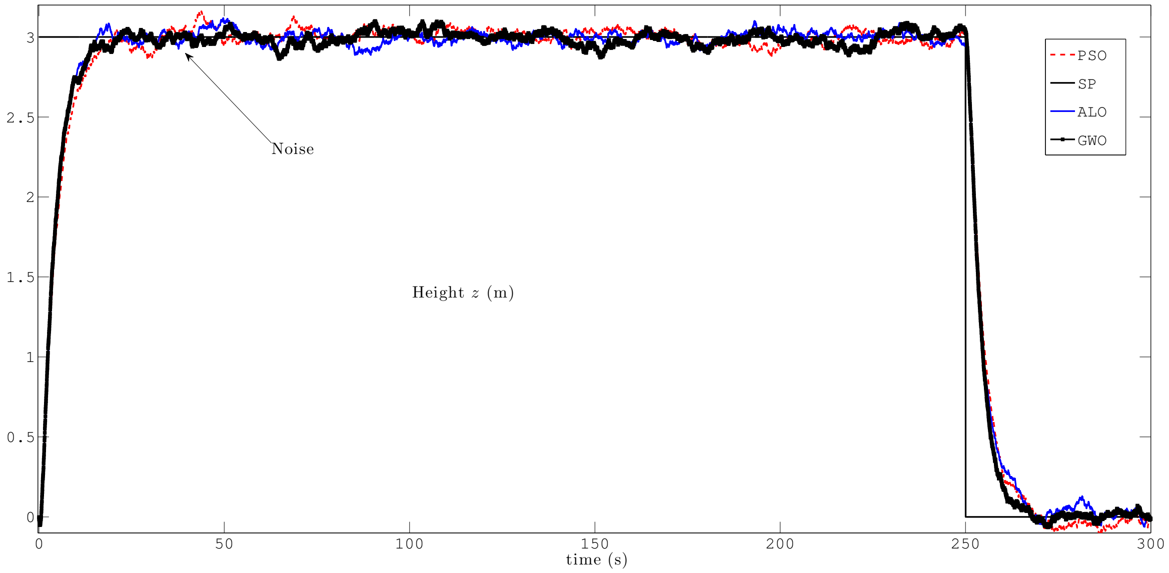

Figure 12. Flight plan 4 (zoom)—effect of measurement noise on the height control.

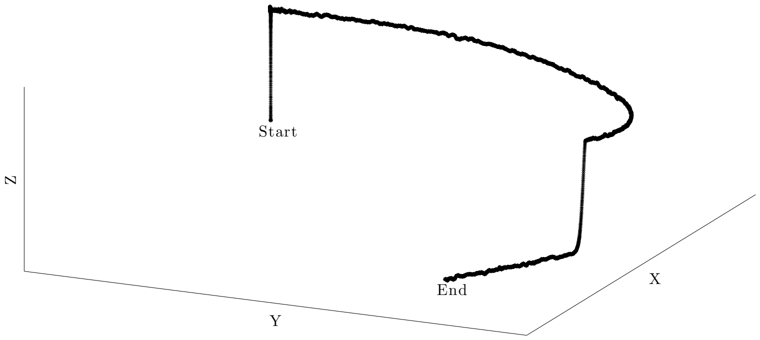

Figure 13. Flight plan 4—measurement noise.

6. Conclusions

This paper proposes the use of both ALO and GWO algorithms to solve the optimization problem of finding a suitable parameter set for the quadrotor Sliding Mode control of altitude and attitude stabilization and tracking. As trial-and-error adjustment is the common practice; this work addresses a challenging and open issue due to the strong trade-off among parameters related to set point tracking, as robustness is guaranteed and chattering is kept within feasible limits. Both techniques were simulated and compared with PSO. Both GWO and ALO outperformed PSO in terms of precise tracking, for ideal and disturbed conditions. Regarding the optimization process, since ALO does not achieve the same best fitness values found by GWO and PSO, this result stimulates further research on ALO improvements for control and robotics applications, as well as the path that PSO and other meta-heuristics optimization algorithms have undergone. This finding is in accordance to the motivation for novel ALO approaches, such as the Dynamic ALO [21] and the Chaotic ALO [15,40], to be explored in future works. Moreover, the multi-objective approach can be explored, as well as hybrid optimization techniques, since ALO and GWO presented good results for different process variables. Besides, practical implementation is intended in order to evaluate on-board specific issues (embedded) and corroborate such simulation results.

Author Contributions: The authors J.O., P.M.O., T.P. and J.B.-C. declare that they are responsible for the preparation of the manuscript entitled EVALUATION OF HUNTING-BASED OPTIMIZERS FOR A QUADROTOR SLIDING MODE FLIGHT CONTROLLER. Below are described each contribution: J.O. and P.M.O. conceived the conceptualization and original idea and developed the methodology; J.O. designed the model and the computational framework, analysed the data and carried out the simulation, writing-original draft with input from all authors; J.B.-C., besides reviewing, contributed to get funding acquisition. T.P. conducted simulation and visualization editing; All authors discussed the results and writing-review and editing process. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Simelli, I.; Tsagraris, A. The Use of Unmanned Aerial Systems (UAS) in Agriculture. In Proceedings of the 7th International Conference on Information and Communication Technologies in Agriculture, Food and Environment (HAICTA 2015), Kavala, Greece, 17–20 September 2015; pp. 730–736.
2. Dief, T.; Yoshida, S. Review: Modeling and Classical Control of Quad-rotor. *Int. J. Comput. Sci. Inf. Technol. Secur.* **2015**, *5*, 314–319. [CrossRef]
3. Bolandi, H.; Rezaei, M.; Mohsenipour, R.; Nemati, H.; Smailzadeh, S. Attitude Control of a Quadrotor with Optimized PID Controller. *Intell. Control. Autom.* **2013**, *4*, 335–342. [CrossRef]
4. Utkin, V. *Sliding Modes in Control and Optimization*; Springer: Berlim/Heidelberg, Germany, 1992.
5. Swarup, A.; Sudhir. Comparison of Quadrotor Performance using Backstepping and Sliding Mode Control. In Proceedings of the International Conference on Circuits, Systems and Control, Interlaken, Switzerland, 22–24 February 2014; pp. 79–82.
6. Runcharoon, K.; Srichatrapimuk, V. Sliding Mode Control of Quadrotor. In Proceedings of the 45th International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), Konya, Turkey, 9–11 May 2013; pp. 552–557. [CrossRef]
7. Xiong, J.J.; Zhang, G. Discrete-time Sliding Mode Control for a Quadrotor UAV. *Int. J. for Light Electron Opt.* **2016**, *127*, 3718–3722. [CrossRef]
8. González, J.; Salazar, S.; Lozano, R. Chattering-Free Sliding Mode Altitude Control for a Quad-Rotor Aircraft: Real-time Application. *J. Intell. Robot. Syst.* **2014**, *73*, 137–155. [CrossRef]
9. Zhang, E.H.; Xiong, J.J.; Luo, J.L. Second Order SMC for a quadrotor UAV. *Isa Trans.* **2014**, *53*, 1350–1356. [CrossRef]
10. Berkenkamp, F.; Schoellig, A. Safe and Robust Learning Control with Gaussian Processes. In Proceedings of the 14th European Control Conference, Linz, Austria, 15–17 July 2015; pp. 2501–2506.
11. Yacef, F.; Bouhali, O.; Hamerlain, M.; Rezoug, A. PSO Optimization of Integral Backstepping Controller for Quadrotor Attitude Stabilization. In Proceedings of the 3th IEEE International Conference on Systems and Control, Algiers, Algeria, 29–31 October 2013; pp. 462–466. [CrossRef]
12. Basri, M.; Danapalasingam, K.; Husain, A. Design and Optimization of Backstepping Controller for an Underactuated Autonomous Quadrotor Unmanned Aerial Vehicle. *Trans. Famena* **2014**, *38*, 27–44.
13. Herrera, M.; Chamorro, W.; Gómez, A.; Camacho, O. Sliding Mode Control: an Approach to Control a Quadrotor. In Proceedings of the Asia-Pacific Conference on Computer Aided System Engineering, Quito, Ecuador, 14–16 July 2015; pp. 314–319. [CrossRef]
14. Clerc, M.; Kennedy, J. The Particle Swarm—Explosion, Stability and Convergence in a Multidimensional Complex Space. *IEEE Trans. Evol. Computation* **2002**, *6*, 58–73. [CrossRef]
15. Oliveira, J.; Oliveira, P.; Boaventura-Cunha, J.; Pinho, T. Chaos-based grey wolf optimizer for higher order sliding mode position control of a robotic manipulator. *Nonlinear Dyn.* **2017**, pp. 1–10. [CrossRef]
16. Panda, M.; Das, B. Grey Wolf Optimizer and Its Applications: A Survey. In Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems, Ranchi, India, 12–13 May 2019;pp. 179–194.
17. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
18. Heidari, A.; Faris, H.; Mirjalili, S.; Aljarah, I.; Mafarja, M., *Nature-Inspired Optimizers*; Chapter Ant Lion Optimizer: Theory, Literature Review, and Application in Multi-layer Perceptron Neural Networks; Springer: Berlin, Germany, 2020; Volume 811, pp. 23–46.

19. Mirjalili, S.; Mirjalili, S.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
20. Gao, Z.M.; Zhao, J. An Improved Grey Wolf Optimization Algorithm with Variable Weights. *Comput. Intell. Neurosci.* **2019**, 1–13. [CrossRef] [PubMed]
21. Yao, P.; Wang, H. Dynamic Adaptive Ant Lion Optimizer Applied to Route Planning for Unmanned Aerial Vehicle. *Soft Comput.* **2016**, 1–14. [CrossRef]
22. Zhang, S.; Zhou, Y.; Li, Z.; Pei, W. Grey Wolf Optimizer for Unmanned Combat Aerial Vehicle Path Planning. *J. Adv. Eng. Softw.* **2016**, *99*, 121–136. [CrossRef]
23. Radmanesh, M.; Kumar, M. Grey Wolf Optimization based Sense and Avoid Algorithm for UAV Path Planning in Uncertain Environment using a Bayesian Framework. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 68–76.
24. Raju, M.; Saikia, L.; Sinha, N. Automatic Generation Control of a Multi-area System using Ant Lion Optimizer Algorithm based PID plus Second Order Derivative Controller. *Int. J. Electr. Power Energy Syst.* **2016**, *80*, 52–63. [CrossRef]
25. Nischal, M.; Mehta, S. Optimal Load Dispatch using Ant Lion Optimization. *J. Eng. Res. Appl.* **2015**, *5*, 10–19.
26. Gupta, E.; Saxena, A. Grey wolf optimizer based regulator design for automatic generation control of interconnected power system. *Cogent Eng.* **2016**, *3*, 1–20. [CrossRef]
27. Mehta, S.; Nischal, M. Ant Lion Optimization for Optimum Power Generation with Value Point Effects. *Int. J. Res. Appl. Sci. Eng. Technol.* **2015**, *3*, 1–6.
28. Milica, P.; Jelena, P.; Mitić, M.; Vuković, N.; Miljković, Z.; Babić, B. The Ant Lion Algorithm for Integrated Process Planning and Sheduling. *Appl. Mech. Mater.* **2016**, *384*, 187–192.
29. Yogarajan, G.; Revathi, T. A Discrete Ant Lion Optimization (DALO) for Solving Data Gathering Tour Problem in Wireless Sensor Networks. *Middle-East J. Sci. Res.* **2016**, *24*, 3113–3120.
30. Dubey, H.; Pandit, M.; Panigrahi, B. Hydro-thermal-wind Scheduling Employing Novel Ant Lion Optimization Technique with Composite Ranking Index. *Renew. Energy* **2016**, *99*, 18–34. [CrossRef]
31. Mirjalili, S.; Saremi, S.; Mirjalili, S.; Coelho, L. Multi-objective Grey Wolf Optimizer: A Novel Algorithm for Multi-Criterion Optimization. *Expert Syst. Appl.* **2016**, *47*, 106–119. [CrossRef]
32. Oliveira, P.; Vrančić, D. Grey Wolf, Gravitational Search and Particle Swarm Optimizers: A Comparison for PID Controller Design. In Proceedings of the 12th Portuguese Conference on Automatic Control (CONTROLO 2016), Guimarães, Portugal, 14–16 September 2016; pp. 239–249.
33. Oliveira, P.; Freire, H.; Pires, E. Grey wolf optimization for PID controller design with prescribed robustness margins. *Soft Comput.* **2016**, *20*, 4243–4255. [CrossRef]
34. Bouabdallah, S.; Murrieri, P.; Siegwart, R. Towards Autonomous Indoor Micro VTOL. *Auton. Robot.* **2005**, *18*, 171–183. [CrossRef]
35. Lee, H.; Utkin, V. Chattering suppression methods in sliding mode control systems. *Annu. Rev. Control.* **2007**, *31*, 179–188. [CrossRef]
36. Samir, Z. Sliding Mode Control of a 6 DOF Quadrotor Helicopter. *J. Electr. Eng.* **2010**, *10*, 126–132.
37. Batmaz, A.; Elbir, O.; Kasnakoglu, C. Design of a Quadrotor Roll Controller Using System Identification to Improve Empirical Results. *Int. J. Mater. Mech. Manuf.* **2013**, *1*, 347–349. [CrossRef]
38. Wahab, M.; Nefti-Meziani, S.; Atyabi, A. A Comprehensive Review of Swarm Optimization Algorithms. *PLoS ONE* **2015**, *10*. [CrossRef]
39. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
40. Zawbaa, H.; Emary, E.; Grosan, C. Feature Selection via Chaotic Antlion Optimization. *PLoS ONE* **2016**, *11*. [CrossRef]