electronics

MDPI

Article

Optimised Sliding Mode Control of a Hexacopter: Simulation and Experiments

Chaoran Sun [1], Stephen A. Agha [2], Zaharuddin Mohamed [3] and M. Hasan Shaheed [1,*]

[1] School of Engineering and Materials Science, Queen Mary University of London, London E1 4NS, UK
[2] Department of Engineering and Mathematics, Sheffield Hallam University, Sheffield S1 1WB, UK
[3] School of Electrical Engineering, Universiti Teknologi Malaysia, Skudai 81310, Malaysia
* Correspondence: m.h.shaheed@qmul.ac.uk

Abstract: Hexacopters are a kind of unmanned aerial vehicle (UAV) with six actuators and six degrees-of-freedom motions. The control of a hexacopter drone is a critical challenge. This paper presents a nonlinear dynamical model for a hexacopter and complete control approaches based on sliding mode control theory. Furthermore, this study proposed an effective control tuning method based on an optimisation algorithm. The controller has been improved by the grey wolf optimisation (GWO) algorithm, an iteration algorithm inspired by the social hierarchy and hunting behaviour of grey wolves. The improvement of the controller has been verified both experimentally and in simulations. The performance of the sub-controller for an attitude angle was tested in a test bench, and the whole flight controller was tested in simulation hexacopters, which are highly manoeuvrable, nonlinear aerial vehicles with six independent rotors and capacity for vertical take-off and landing. This article presents a derivation of the nonlinear dynamical model for a hexacopter, which includes the aerodynamic drag, and inertia counter torques. Flight control based on sliding mode control theory, which generally shows good performance on nonlinear systems, is developed and implemented. Given the need to simultaneously tune controller parameters, two nature inspired optimisation routines (GWO and PSO) are applied and compared for effectiveness in controller tuning. Results indicate that GWO-based tuning produces superior outcomes in terms of controller performance. This is in addition to the fact that PSO parameters require tuning rather than random selection of algorithm parameters. A reduced-order physical prototype is presented for the validation of the tuning routine on the roll/pitch control. The results indicate good agreement between simulation and experimental outcomes, with about 10.4% improvement in the tracking performance of roll DOF when GWO is applied to tune the controller.

Keywords: robotics; optimisation; control; GWO; sliding mode; hexacopter

Citation: Sun, C.; Agha, S.A; Mohamed, Z.; Shaheed, M.H. Optimised Sliding Mode Control of a Hexacopter: Simulation and Experiments. *Electronics* **2022**, *11*, 2519. https://doi.org/10.3390/electronics11162519

Academic Editor: Ahmed Al Durra

Received: 20 June 2022
Accepted: 4 August 2022
Published: 11 August 2022

1. Introduction

In recent years, multirotor UAVs (commonly known as drones) have become popular and widely used. Compared to traditional helicopters, drones have a simper mechanical structure, which is easier to control [1]. As a result, drones are widely used for aerial filming. They have potential applications in search and rescue operations, environment monitoring, law enforcement, precision agriculture, traffic management, delivery, etc.

The most commonly used drones are quadcopters or quadrotor helicopters. Drones with more rotors, however, have a higher thrust-to-weight ratio and a greater load capacity but generally, higher energy consumption. However, a drone with more rotors, such as a hexacopter, can consume lower energy under certain mass distribution conditions than a quadrotor with the same material and the same motor, as demonstrated in Sun's research [2]. This study investigates the enhancement of the potential of a hexacopter through the development of an optimised sliding mode control (SMC) model.

Although more research has focussed on quadrotors, the core of the flight control structure is similar for both quadrotors and hexacopters.

In order to meet the requirements of remote control and the autonomous flight of a drone, designing a flight controller is a critical task. The study of control usually starts from the PID control theory. The traditional PID control approach can partially meet these requirements. Salih et al. applied the PID control approach for the remote control of a quadcopter. The results show that the controller responds quickly. However, there are obvious oscillations and overshoots [3]. Mustapa et al. applied PID control for the altitude controller of a quadcopter. Altitude control is a relatively simple part of a drone flight controller, which can be addressed by the classic PID approach [4].

Agha [5] and Abidali [6] each studied the PID controller as a baseline controller for solar-powered drones and used other approaches to improve the performance. Bouabdallah's early work on quadrotor flight control was mainly focused on the backstepping control approach, which was validated in experiments [7]. Bouabdallah also studied the sliding mode control (SMC) approach with Siegwart [7–9]. J.A. Ricardo Jr and D.A. Santos studied a smooth second-order SMC for a hexacopter for position and attitude control problems [10]. Moya et al. studied SMC for a drone and confirmed its advantages over PID controllers [11]. In Chiew et al.'s research about altitude and yaw control of a quadcopter, second-order SMC outperformed PD control in all considered performance indicators. [12] Abrougui et al. proposed a robust SMC scheme to stabilise the attitude of a hexacopter during waypoint tracking [13].

Tuning is a critical process of control approaches development; it refers to adjusting the coefficients of the controller to improve performance. Agha, for example, used the Ziegler–Nichols tuning method in the case of PID control, and both Agha and Abidali used fuzzy logic theory to improve the performance of the PID controllers [5,6]. The tuning of SMC has also been studied, for example, Concha et al. developed a tuning algorithm for SMC and applied it into attenuating the structural vibration. [14] Nekoukar and Dehkordi applied fuzzy logic-based theory on a sliding mode controller of a quadcopter, which achieved robust past tracking. [15]

Optimisation algorithms have been widely used in tuning, for example, particle swarm optimisation (PSO) and grey wolf optimisation (GWO). Both approaches involve biomimicry to obtain the optimal solution. They are characterised by the fact that they do not require in-depth knowledge of the system to obtain the optimal solution; just as animals arriving in a new environment can also obtain their prey quickly in some way. So, these approaches have a wide range of applications. Liu et al. applied PSO on SMC for a vector machine [16]; Kumar et al. utilised PSO- and GWO-optimised SMC on frequency regulation. [17] These approaches have also been applied to drones; for example, PSO has previously been used for tuning the PID control of a quadrotor UAV by Noordin et al. [18] and Yin et al. [19]. Qu et al. developed a hybrid GWO algorithm and applied it on the path planning of a drone. [20] However, the study of optimisation-based SMC optimisation for controlling a hexacopter is limited. This study improves the SMC flight controller by using optimisation approaches, such as PSO and GWO.

2. Dynamical Model

Two coordinate systems are needed to describe the translational and rotational movement of a hexacopter. They are the inertial coordinate ($\boldsymbol{O_iX_iY_iZ_i}$), based on earth, and the body-fixed coordinate ($\boldsymbol{O_bX_bY_bZ_b}$) attached to the centre of mass of the drone. For a hexacopter, $\boldsymbol{X_b}$-axis is on a strip, $\boldsymbol{Y_b}$-axis is on an angle bisector of two strips, and $\boldsymbol{Z_b}$-axis is defined by the right-hand rule, as demonstrated in Figure 1. The position of the drone is expressed by vector $\boldsymbol{r}$ in inertial coordinate**:**

$$\boldsymbol{r} = [x \quad y \quad z]^{\boldsymbol{T}} \tag{1}$$

Moreover, the velocity of the drone is expressed in body-fixed coordinate:

$$\boldsymbol{V} = [v_x \quad v_y \quad v_z\ ]^{\boldsymbol{T}} \tag{2}$$

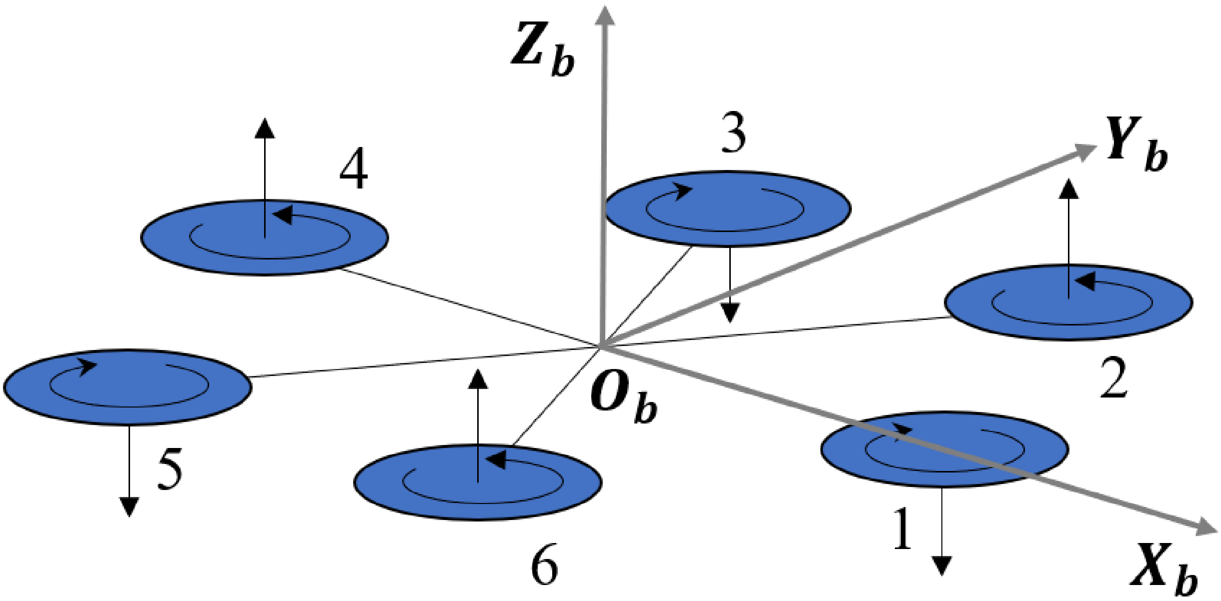

Figure 1. Kinematic scheme of a hexacopter.

The attitude of the drone is described by Euler angles $\varphi\ \theta\ \psi$, which expresses roll, pitch, and yaw angles, respectively. In addition, the transformation between inertial and body-fixed coordinates depends on the attitude angles. The two coordinates are coincident initially, and the drone rotates around the *z*, *y*, and *x* axes, as shown in Figure 2. $O_1 \times {}_1Y_1Z_1$ and $O_2X_2Y_2Z_2$ are two intermediate coordinates after the first and second rotation; accordingly, $\dot{\boldsymbol{r}}_1 = [\dot{x}' \quad \dot{y}' \quad \dot{z}']^{\boldsymbol{T}}$ and $\dot{\boldsymbol{r}}_2 = [\dot{x}'' \quad \dot{y}'' \quad \dot{z}'']^{\boldsymbol{T}}$ are the intermediate velocity vectors. The mutual orientation of the coordinates is described by the rotation matrix $\boldsymbol{R}$ [21].

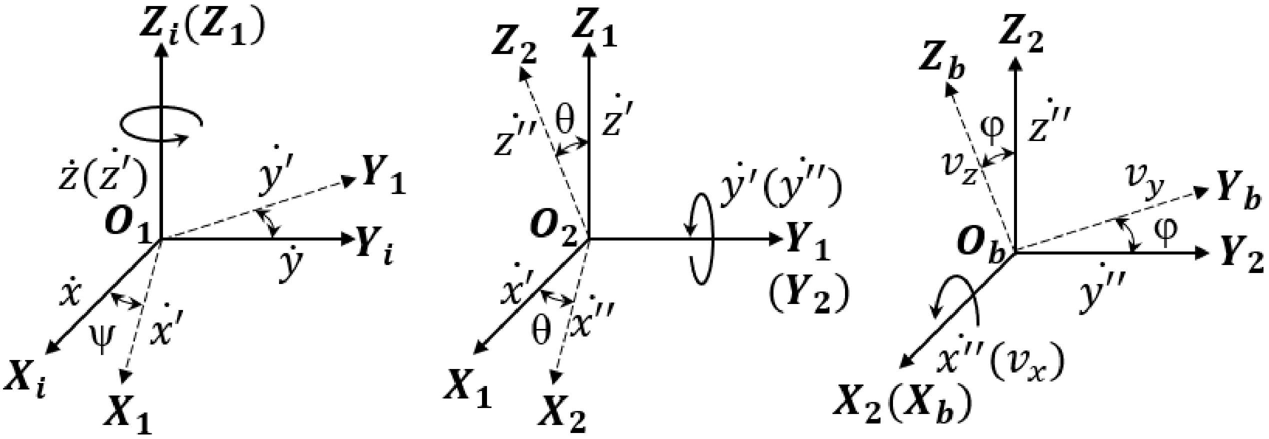

Figure 2. Rotation of hexacopter.

$$\boldsymbol{R} = \begin{bmatrix} cos\theta\ cos\psi & sin\varphi\ sin\theta\ cos\psi - cos\varphi\ sin\psi & cos\varphi\ sin\theta\ cos\psi + sin\varphi\ sin\psi \\ cos\theta\ sin\psi & sin\varphi\ sin\theta\ sin\psi + cos\varphi\ cos\psi & cos\varphi\ sin\theta\ sin\psi - sin\varphi\ cos\psi \\ -sin\theta & sin\varphi\ cos\theta & cos\varphi\ cos\theta \end{bmatrix} \tag{3}$$

The transformation of velocities between the body-fixed and inertial frames is given as:

$$\dot{\boldsymbol{r}} = \boldsymbol{RV} \tag{4}$$

The same angular velocity is expressed differently in different coordinate systems. The vector $\boldsymbol{\omega}$ denotes the angular speed of the drone in body-fixed coordinates where p, q, and r are the angular velocities of hexacopter in the body frame; the vector $\boldsymbol{q}$ is the attitude vector in the earth-fixed frame.

$$\boldsymbol{\omega} = [p \quad q \quad r\ ]^{\boldsymbol{T}} \tag{5}$$

$$\boldsymbol{q} = [\varphi \quad \theta \quad \psi\ ]^{\boldsymbol{T}} \tag{6}$$

The transformation matrix for the angular speed is as follows:

$$\boldsymbol{R_A} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \cos\theta\sin\varphi \\ 0 & -\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \tag{7}$$

The transformation of the angular speed is:

$$\boldsymbol{\omega} = \boldsymbol{R_A}\dot{\boldsymbol{q}} \tag{8}$$

Equation (9) is the Newton–Euler equation of motion, which describe the equilibrium of forces of a drone in body coordinate:

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{N} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{I_3} & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega} \times m\boldsymbol{V} \\ \boldsymbol{\omega} \times \boldsymbol{M\omega} \end{bmatrix} \tag{9}$$

where $\boldsymbol{F}$ is the force vector and $\boldsymbol{N}$ is the torque vector, $\boldsymbol{I_3}$ is a 3 × 3 identity matrix, and $\boldsymbol{M}$ is the inertia matrix:

$$\boldsymbol{M} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{10}$$

The force is determined by the gravity of the drone and the thrust generated by the rotors, the aerodynamic resistance. The equation of force with transformation can be expressed as Equation (11):

$$\boldsymbol{F} = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \boldsymbol{R}^{-1} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \boldsymbol{R}^{-1} \begin{bmatrix} D_x \\ D_y \\ D_z + L \end{bmatrix} \tag{11}$$

where m is the mass of the drone, g is gravitational acceleration, and T is the total thrust generated by the propellers.

$$T = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 + \Omega_5^2 + \Omega_6^2) \tag{12}$$

In Equation (12), b is the thrust coefficient, and Ω_1 to Ω_6 are the angular speeds of the corresponding propellers.

The aerodynamic resistance of the drone cannot be ignored. The direction of the drag is opposite to the direction of the movement, D_x, D_y, and D_z are the aerodynamic resistances or drags in *x*, *y*, and *z* direction. [22,23]

In Equation (13), ρ is the air density; $C_{Dx}, C_{Dy}\ and\ C_{Dz}$ are the drag coefficients, which are dimensionless numbers, depending on the angle of attack (AOA). C_{Lx}, C_{Ly} are dimensionless lift coefficients dependent on the AOA; A is the area of the drone in contact with the air when flying vertically upwards.

$$\begin{cases} D_x = -\frac{1}{2}\rho A C_{Dx}\dot{x}|\dot{x}| \\ D_y = -\frac{1}{2}\rho A C_{Dy}\dot{y}|\dot{y}| \\ D_z = -\frac{1}{2}\rho A C_{Dz}\dot{z}|\dot{z}| \end{cases} \tag{13}$$

$$L = -\frac{1}{2}\rho A\left(C_{Lx}\dot{x}^2 + C_{Ly}\dot{y}^2\right)$$

The torque vector $\boldsymbol{N}$:

$$\boldsymbol{N} = [N_x \quad N_y \quad N_z]^{\boldsymbol{T}} = \boldsymbol{\tau_T} + \boldsymbol{\tau_{P_{Gyro}}} + \boldsymbol{\tau_{counter}} + \boldsymbol{\tau_{drag}} \tag{14}$$

where $\boldsymbol{\tau_T}$ is the torque produced by the propellers; $\boldsymbol{\tau_{P_{Gyro}}}$ is the propeller gyroscopic torque, $\boldsymbol{\tau_{counter}}$ is the inertial counter-torque; $\boldsymbol{\tau_{drag}}$ is the torque due to the aerodynamic drag.

For a hexacopter, the torque produced by the propeller is [24,25]:

$$\begin{cases} \tau_{Tx} = \frac{\sqrt{3}}{2} bl(\Omega_5^2 + \Omega_6^2 - \Omega_2^2 - \Omega_3^2) \\ \tau_{Ty} = bl\left(-\Omega_1^2 - \frac{1}{2}\Omega_2^2 + \frac{1}{2}\Omega_3^2 + \Omega_4^2 + \frac{1}{2}\Omega_5^2 - \frac{1}{2}\Omega_6^2\right) \\ \tau_{Dz} = d(\Omega_1^2 + \Omega_3^2 + \Omega_5^2 - \Omega_2^2 - \Omega_4^2 - \Omega_6^2) \end{cases} \tag{15}$$

where b is the thrust factor, d is the drag torque constant, and l is the arm length; τ_{Tx} and τ_{Ty} are the torque due to thrust about *x*-axis on body-fixed frame; τ_{Dz} is the torque about the *z*-axis, which is dominated by the torque from the drag of the blade.

The propeller gyroscopic effect results from a change of angular momentum of propellers due to the rotation of the hexacopter. It is given by the cross-product of the angular velocity of the hexacopter and the angular momentum of a propeller. The expression is:

$$\boldsymbol{\tau_{P_{Gyro}}} = \begin{bmatrix} -J_r q(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 + \Omega_5 - \Omega_6) \\ J_r p(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 + \Omega_5 - \Omega_6) \\ 0 \end{bmatrix} \tag{16}$$

where J_r is the moment of inertia of the rotor (a propeller and a motor).

The inertial counter-torque is the torque required to balance the torque of rotors. The torque is given by the product of the rotor moment of inertia and angular acceleration. Applying the rotation form of Newton's third law; the counter torques are numerically equal to the rotors torques but in opposite directions. The counter-torque only appears on the *z*-axis (of the body-fixed frame):

$$\boldsymbol{\tau_{counter}} = \begin{bmatrix} 0 \\ 0 \\ J_r(\alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 + \alpha_5 - \alpha_6) \end{bmatrix} \tag{17}$$

where α_1 to α_6 are the angular accelerations of the corresponding propellers.

The drone is also affected by aerodynamics during its rotation. The expression of the torque caused by aerodynamics is:

$$\boldsymbol{\tau_{drag}} = \begin{bmatrix} -C_{Tx}p|p| \\ -C_{Ty}q|q| \\ -C_{Tz}r|r| \end{bmatrix} \tag{18}$$

C_{Tx}, C_{Ty}, and C_{Tz} are the drag torque coefficients.

According to Newton–Euler Equation, the acceleration and angular acceleration of the drone in body-fixed coordinate can be calculated with the following equation:

$$\begin{cases} \dot{\boldsymbol{V}} = \dfrac{1}{m}\boldsymbol{F} - \boldsymbol{\omega} \times \boldsymbol{V} \\ \dot{\boldsymbol{\omega}} = \boldsymbol{M}^{-1}\boldsymbol{N} - \boldsymbol{\omega} \times \boldsymbol{\omega} \end{cases} \tag{19}$$

The drone's velocity can be calculated by integrating the acceleration and then transforming the velocity to inertial coordinate by Equations (3) and (4). Likewise, the position can be calculated by integrating the velocity. Similarly, the attitude angle can be calculated by two integrations and one transformation.

3. Control of Hexacopter

The objective of the control is to make the drone fly autonomously following a desired trajectory. The desired trajectory includes the desired position (*x*, *y*, *z*) and the desired yaw angle. The pitch and roll angles are supposed to be small, but they control the position. A state vector is defined [11]:

$$\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}]^{\boldsymbol{T}} = \left[\varphi\ \dot{\varphi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ z\ \dot{z}\ x\ \dot{x}\ y\ \dot{y}\right]^{\boldsymbol{T}} \tag{20}$$

The 12-state variables are the position in the inertial axis, attitude angles, and their time derivatives (or speed). The 12 state variables correspond to six degrees of freedom. The flight controller can be divided into six sub-controllers, corresponding to the state variables or degrees of freedom. The rotation sub-controllers are roll sub-controller, pitch sub-controller, and yaw sub-controller; the position sub-controllers are z (altitude) sub-controller, x sub-controller, and y sub-controller.

The state of the drone is determined by the thrust and torque provided by the rotors, the control vector of the hexacopter is defined as:

$$\boldsymbol{U} = [U_1\ U_2\ U_3\ U_4\ ]^T = \left[T\ N_x\ N_y\ N_z\right]^T \tag{21}$$

After simplification, the dynamical model for the flight controller, the time derivative of the state vector is:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{U}) = \begin{pmatrix} x_2 \\ a_1x_4x_6 + b_1U_2 + b_1c_1x_2|x_2| \\ x_4 \\ a_2x_2x_6 + b_2U_3 + b_2c_2x_4|x_4| \\ x_6 \\ a_3x_2x_4 + b_3U_4 + b_3c_3x_6|x_6| \\ x_8 \\ \dfrac{1}{m}cos\,x_3\,cos\,x_1\,U_1 - g + \dfrac{1}{m}c_4x_8|x_8| + \dfrac{1}{m}(c_5{x_{10}}^2 + c_6{x_{12}}^2) \\ x_{10} \\ \dfrac{1}{m}u_xU_1 + \dfrac{1}{m}c_7x_{10}|x_{10}| \\ x_{12} \\ \dfrac{1}{m}u_yU_1 + \dfrac{1}{m}c_8x_{12}|x_{12}| \end{pmatrix} \tag{22}$$

With

$$
\begin{array}{lll}
 & & c_1 = -C_{Tx} \\
 & & c_2 = -C_{Ty} \\
 & & c_3 = -C_{Tz} \\
 & b_1 = \dfrac{1}{J_x} & c_4 = -\dfrac{\rho A C_{Dz}}{2} \\
a_1 = \left(J_y - J_z\right)/J_x & & \\
a_2 = \left(J_z - J_x\right)/J_y & b_2 = \dfrac{1}{J_y} & c_5 = -\dfrac{\rho C_{Lx} A}{2} \\
a_3 = \left(J_x - J_y\right)/J_z & & c_6 = -\dfrac{\rho C_{Ly} A}{2} \\
 & b_3 = \dfrac{1}{J_z} & \\
 & & c_7 = -\dfrac{\rho A C_{Dx}}{2} \\
 & & c_8 = -\dfrac{\rho A C_{Dy}}{2}
\end{array}
\tag{23}
$$

$$
\begin{aligned}
u_x &= \cos x_5 \sin x_3 \cos x_1 + \sin x_5 \sin x_1 \\
u_y &= \sin x_5 \sin x_3 \cos x_1 - \cos x_5 \sin x_1
\end{aligned}
$$

The propeller gyro effect and inertial counter-torque are neglected in this equation. U_2 is the control variable of roll sub-controller; U_3 is for pitch sub-controller; U_4 is for yaw sub-controller. u_x and u_y are an expansion of the control variables and are determined by the attitude angles and U_1. They are the control variables of x and y sub-controllers. In addition, conversely, the commanded pitch and roll angle (x_{1c}, x_{3c} in the form of state variables) can be calculated by the control variables u_x and u_y. When the pitch and roll angles are small,

$$
\begin{aligned}
x_{1c} &= u_x \sin x_5 - u_y \cos x_5 \\
x_{3c} &= u_x \cos x_5 + u_y \sin x_5
\end{aligned}
\tag{24}
$$

The flight control scheme for a hexacopter is indicated in Figure 3. The input variables are the desired position and yaw angle. First, input the desired position to the position sub-controllers to get the commanded control variable U_1 and commanded pitch and roll angles. Then, input the desired yaw angle and commanded pitch and roll angles to the rotation sub-controllers to get the control variables U_2, U_3, and U_4. The control variables are inputted into the system. The system estimates the actual position and attitude and then feeds back to the sub-controllers.

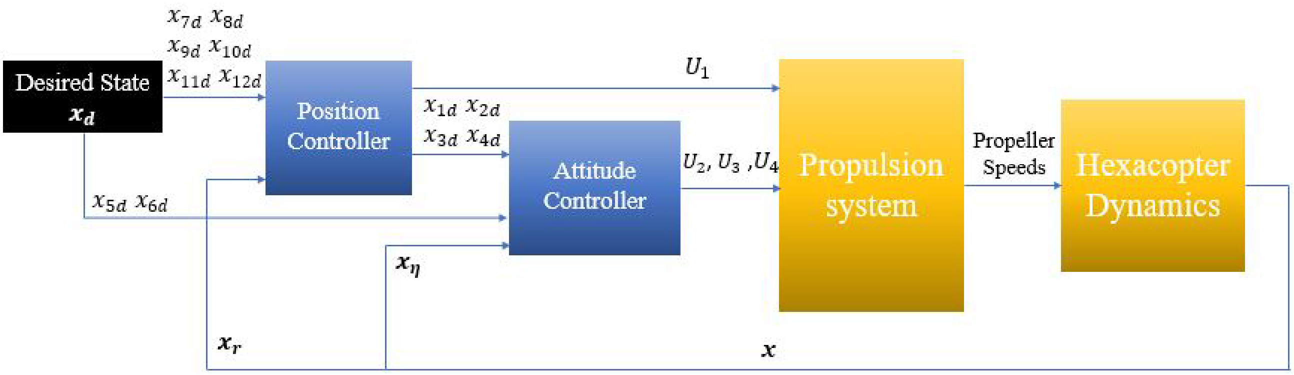

Figure 3. Control scheme.

The control strategy applied in the position and rotation sub-controllers is the sliding mode control approach. It is an iterative method based on Lyapunov stability theory. A sliding surface is designed to drive the state space, and the controller makes the state space reach the sliding surface. When the sliding surface is reached, the controller keeps the state on the neighbourhood of the sliding surface [26]. For a second-order system, such as the roll sub-controller:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = f(x_1, x_2) + h(x_1, x_2)U_2 \end{cases} \tag{25}$$

Define the setpoint x_{1d} and the error z_1:

$$z_1 = x_{1d} - x_1 \tag{26}$$

The sliding surface is defined as follows:

$$s_1(t) = \{(x_1, x_2)|s_1(x_1, x_2, t) = 0\} \tag{27}$$

With $s_1(x_1, x_2, t)$ defined by

$$s_1(x_1, x_2, t) = \left(\frac{d}{dt} + m_1\right) z_1(t), m_1 > 0 \tag{28}$$

where m_1 is a positive constant.

$$s_1 = \dot{x}_{1d} - x_2 + m_1 z_1 \tag{29}$$

To make the state approach, the sliding mode is equivalent with stabilising s to zero, a positive definite Lyapunov candidate function for s is defined as:

$$V(s_1) = \frac{1}{2}{s_1}^2 \tag{30}$$

Differentiating:

$$\dot{s_1} = \ddot{x}_{1d} - (f(x_1, x_2) + h(x_1, x_2)U_2) + m_1(s_1 - m_1 z_1) \tag{31}$$

Define

$$U_2{}^* = \frac{1}{h(x_1, x_2)}\left(\ddot{x}_{1d} + m_1(s_1 - m_1 z_1) - f(x_1, x_2)\right) \tag{32}$$

Then, if $U_2 = U_2{}^*$, $\dot{s_1} = 0$ for all time. Following the exponential reaching law:

$$\dot{s_1} = -n_1 sgn(s_1) - n_2 s_1, with\ U_2 = U_2{}^* + \frac{1}{h(x_1, x_2)}(n_1 sgn(s_1) + n_2 s_1) \tag{33}$$

$$\dot{V}(s_1) = s_1 \dot{s_1} = -n_1|s_1| - n_2{s_1}^2 \tag{34}$$

where n_1 and n_2 are positive constants, the three coefficients of one sliding mode controller. In addition, $\dot{V}(s)$ is a negative definite function. Apply this control approach to a hexacopter with Equation (22), following the exponential law, the rotation sub-controllers:

$$\begin{cases} U_2 = \frac{1}{b_1}(\ddot{x}_{1d} + m_1(s_1 - m_1 z_1) - a_1 x_4 x_6 - c_1 x_2|x_2| + n_1 sgn(s_1) + n_2 s_1) \\ U_3 = \frac{1}{b_2}(\ddot{x}_{3d} + m_2(s_2 - m_2 z_3) - a_2 x_3 x_6 - c_2 x_4|x_4| + n_3 sgn(s_2) + n_4 s_2) \\ U_4 = \frac{1}{b_3}(\ddot{x}_{5d} + m_3(s_3 - m_3 z_5) - a_3 x_2 x_4 - c_3 x_6|x_6| + n_5 sgn(s_3) + n_6 s_3) \end{cases} \tag{35}$$

With

$$\begin{cases} s_1 = \dot{x}_{1d} - x_2 + m_1 z_1 \\ s_2 = \dot{x}_{3d} - x_4 + m_2 z_3 \\ s_3 = \dot{x}_{5d} - x_6 + m_3 z_5 \end{cases} \tag{36}$$

And the position sub-controllers:

$$\begin{cases} U_1 = \dfrac{m}{cos(x_1)\,cos(x_3)}\left(\ddot{x}_{7d} + g + m_4(s_4 - m_4 z_7) - n_7 sgn(s_4) + n_8 s_4 - \dfrac{1}{m}c_4 x_8|x_8| - \dfrac{c_5}{m}{x_{10}}^2 - \dfrac{c_5}{m}{x_{12}}^2\right) \\ u_x = \dfrac{m}{U_1}\left(\ddot{x}_{9d} - m_5(s_5 - m_5 z_9) - n_9 sgn(s_5) + n_{10} s_5 - \dfrac{1}{m}c_7 x_{10}|x_{10}|\right) \\ u_x = \dfrac{m}{U_1}\left(\ddot{x}_{11d} - m_6(s_6 - m_6 z_{11}) - n_{11} sgn(s_6) + n_{12} s_6 - \dfrac{1}{m}c_7 x_{12}|x_{12}|\right) \end{cases} \tag{37}$$

With

$$\begin{cases} s_4 = \dot{x}_{7d} - x_8 + m_4 z_7 \\ s_5 = \dot{x}_{9d} - x_{10} + m_5 z_9 \\ s_6 = \dot{x}_{11d} - x_{12} + m_6 z_{11} \end{cases} \tag{38}$$

z_1 to z_{11} are the difference between the desired value and real value or error. m_1 to m_6 and n_1 to n_{12} are the control coefficients. In the sliding mode flight controller of the drone, each sub-controller has three coefficients; there are 6 sub-controllers with 18 coefficients. These coefficients have been tuned based on previous experience, the requirements for an autonomous flight can be achieved in this way. However, further improvements can be made by means of optimisation.

4. Optimisation

PSO and GWO have been used to optimise the controllers by tuning the coefficients in this study. Both of these optimisation approaches are derived from biomimicry. PSO is inspired by the social behaviour and dynamical movements with the communication applicable in insects, birds, and fish; GWO is inspired by grey wolves' social hierarchy and hunting process.

In PSO, all particles are in search of an optimal position. This position is unknown, but it is possible to compare which particle is in a better position. Each particle decides where to move next based on the best position it has experienced and the best position of all the particles [18].

$$\begin{cases} W(t_i + 1) = w_0 W(t_i) + w_1 r_1\big(P(t_i) - X(t_i)\big) + w_2 r_2\big(G(t_i) - X(t_i)\big) \\ X(t_i + 1) = X(t_i) + W(t_i + 1) \end{cases} \tag{39}$$

Equation (39) expresses the direction and position of a particle in the next iteration. t_i is the iteration number, W is the direction, and X is the position. P is the best position reached by the particle, and G is the best position reached by the swarm. r_1 and r_2 are random numbers between 0 and 1; w_0, w_1, and w_2 are positive constants as the coefficients. w_0 is usually equal to 1 and $w_1 + w_2 = 4$.

In GWO, four groups are defined in the algorithm to simulate the social hierarchy of grey wolves [27]:

- Alpha (α)—the leaders, called alpha, are responsible for decision-making, such as hunting, sleeping place, time to wake, etc.
- Beta (β)—this is the second level. They are the subordinate wolves, and they help the alpha wolves with decision making.
- Delta (δ)—like beta wolves, they must submit to alpha and beta wolves but dominate the other wolves.
- Omega (ω)—the followers.

During the hunting process, the exact location of the prey is unknown. Therefore, the alpha, beta, and delta wolves have more experience and knowledge about hunting to estimate the prey's position more accurately. The hunting is guided by alpha, beta, and delta; the omega wolves are followers.

The following equations represent the mathematical model of the hunting behaviour:

$$\begin{cases} D_\alpha = |C_1 X_\alpha(t_i) - X(t_i)| \\ D_\beta = |C_2 X_\beta(t_i) - X(t_i)| \\ D_\delta = |C_3 X_\delta(t_i) - X(t_i)| \end{cases} \tag{40}$$

In Equation (40), t_i is the current iteration. Thus, X_α, X_β, and X_δ are the three best positions or the positions of alpha, beta, and delta wolves. X is the position of any wolf in the group; C_1, C_2, and C_3 are random numbers.

$$\begin{cases} X_1 = X_\alpha - A_1 D_\alpha \\ X_2 = X_\beta - A_2 D_\beta \\ X_3 = X_\delta - A_3 D_\delta \\ X(t_i+1) = \dfrac{X_1 + X_2 + X_3}{3} \end{cases} \tag{41}$$

In Equation (41), A_1, A_2, and A_3 are adaptive numbers; these two equations define how the wolf moves with the guide of the leaders. The random numbers and adaptive numbers are defined with

$$\begin{cases} C_{1,2,3} = 2r_1 \\ A_{1,2,3} = 2ar_2 - a \\ a = 2 - 2(t_i/max) \end{cases} \tag{42}$$

where r_1 and r_2 are random numbers between 0 and 1, max is the number of iterations. The process of the GWO algorithm is indicated in Figure 4. The three groups of wolves with the highest fitness are alpha, beta, and delta wolves. The position and identity of the wolves are updated in each loop. The system then proceeds to the next iteration (Iter = Iter + 1) until the maximum number of iterations is reached (Iter == max).

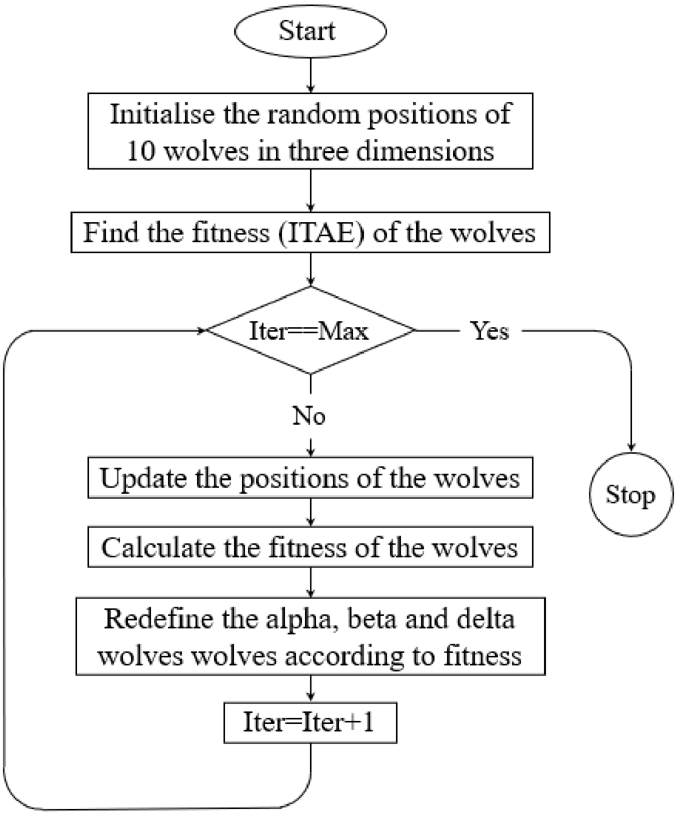
Start
Initialise the random positions of 10 wolves in three dimensions
Find the fitness (ITAE) of the wolves
Iter==Max
Yes
No
Update the positions of the wolves
Calculate the fitness of the wolves
Redefine the alpha, beta and delta wolves wolves according to fitness
Iter=Iter+1
Stop

Figure 4. GWO Algorithm.

At first, the roll sub-controller was used to test the performance of the optimisation algorithms. The 30 degrees step response tested the sets of coefficients of the roll sub-controller. The optimal coefficients are unknown during the optimisation of a sliding mode controller. A total of 10 sets of coefficients are generated randomly, such as ten particles or wolves. The integral of time multiplied absolute error (ITAE, Equation (43)) measured in simulation has been taken as the objective function.

$$\text{ITAE} = \int_0^{\infty} t|e(t)|dt \tag{43}$$

where $e(t)$ represents the error at a certain operation time.

ITEA reflects the accumulated error in the control as an indicator to judge the performance of a controller. So, three coefficients refer to the position of a particle or a group of wolves. The smaller the ITAE, the better the position. Initially, the coefficients m_1, n_1, and n_2 are 20, 4, and 4, respectively, these coefficients are based on some previous experiences and manual tunings. The step time of the input signal is at 2 s, and the final value is 30 degrees. The original ITAE is 8.616. The roll sub-controller has been optimised four times with the number of iterations 3, 5, 10, and 15 with the two optimisation approaches. With the increase in iterations, the ITAE reduced significantly. The detail of the comparisons is in Table 1 for PSO and Table 2 for GWO.

Table 1. PSO controller performance in term of ITAE.

PSO Iterations	m_1	n_1	n_2	**ITAE (*rad s*^{2})**
0	20.0	4.00	4.00	8.616
3	17.96	7.34	22.71	6.056
5	19.66	13.26	24.59	5.540
10	35.77	11.04	36.91	4.153
15	37.91	10.61	38.03	4.097

Table 2. GWO controller performance in term of ITAE.

GWO Iterations	m_1	n_1	n_2	**ITAE (*rad s*^{2})**
0	20.0	4.00	4.00	8.616
3	18.84	8.24	26.82	5.654
5	20.38	9.85	33.14	5.141
10	39.35	15.80	46.31	3.956
15	39.49	15.85	46.47	3.955

As can be seen from Tables 1 and 2, the effectiveness of the two optimisation algorithms can be compared by comparing ITAE. GWO outperformed PSO by 6.6% and 7.2% in three and five iterations, respectively, in terms of optimisation; in 10 and 15 iterations, the advantage reduced to 4.7% and 3.5%. After more than ten iterations, the sub-controllers improved by both optimisation algorithms and were close to optimal. However, the performance of GWO is always slightly better than PSO. The coefficients were set to $w_1 = 2$ and $w_2 = 2$. Other coefficients have been tried but with no better performance. The purpose of using these optimisation methods is to tune the controller. Still, the PSO itself also has coefficients that need to be adjusted. This includes its three coefficients and the initial direction of motion, which all impact the optimisation results. Whereas all the coefficients of the GWO are random numbers or can vary linearly depending on the number of iterations. The GWO shows better performance; therefore, the overall flight controller is optimised using GWO.

In Table 3, the rise time, the peak time, the overshoot, and the settling time of each optimised roll sub-controller have been compared. All these four response parameters have improved after optimisation. The performances of the sub-controllers optimised over 10 and 15 iterations are very similar; there was only a 0.3% difference in the ITAE. The sub-controller with 15 iterations should be selected if only numerical results are considered. However, from Figure 5, in the performance of the sub-controllers with 10 and 15 iterations, when the simulated value exceeds the target value, there was some oscillation before reaching the peak. This oscillation occurs when the sub-controller has undergone more than five iterations of optimisation. When all rotation sub-controllers work in tandem, these oscillations can introduce further instability into the flight of the drone. So finally, the roll sub-controller, optimised through five iterations, was selected for further design. With the optimisation of five iterations, the rise time was reduced by 52.9%; the peak time was reduced by 45.2%, the overshot reduced by 1.9%, and the ITEA reduced by 59.7%. With the exception of the overshot, the improvement in all other aspects is remarkable.

Table 3. Controller performance in terms of rise time, overshoot, and settling time.

GWO Iterations	Rise(*s*)	Peak (*s*)	Overshot (%)	Settle (*s*)
0	0.087	0.259	21.2	1.014
3	0.051	0.154	24.6	0.499
5	0.041	0.142	20.8	0.469
10	0.026	0.120	19.0	0.389
15	0.026	0.120	19.0	0.389

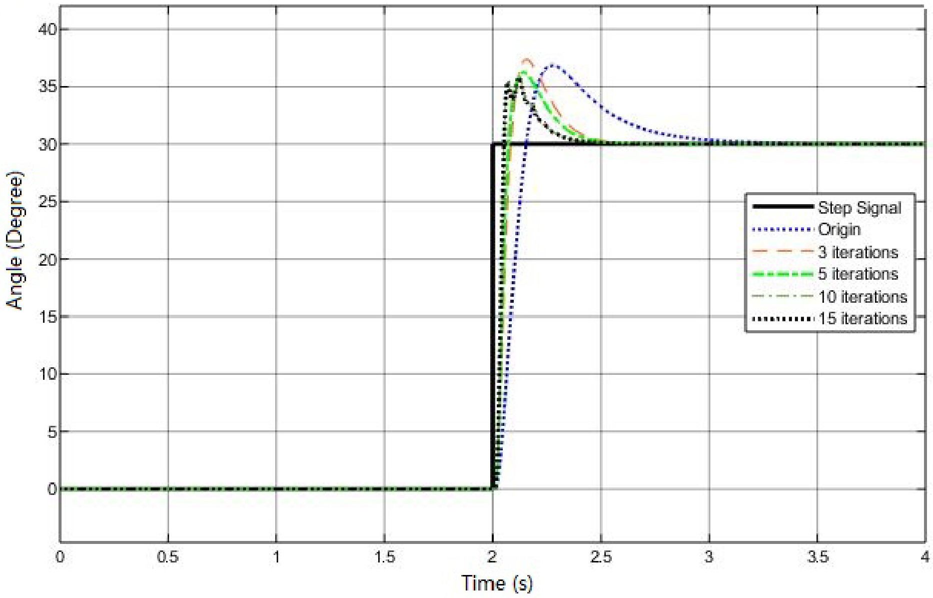

Figure 5. Comparison of the roll sub-controllers optimised by GWO in simulation.

The other sub-controllers have also been optimised by GWO with a similar process. The roll and pitch motions of the drone are very similar, so the pitch sub-controller used the same coefficients as the roll controller. These two sub-controllers were optimised first because they are essential for stable flight and are used to control the translational movement of the drone.

The yaw sub-controller has been optimised secondly. The control of the yaw angle is relatively less important than the other two attitude angles. Thus, when optimising the yaw sub-controller, it is also important to consider its influence on the other two attitude angles. The method used is to simultaneously give a 30 degrees step response command to all three attitude angles. To minimise the sum of the three angles ITAE is used as the optimisation target.

Then, the altitude sub-controller has been optimised. A 2 m step response was given to the system. The optimisation target is to minimise the ITAE of the motion in z-direction. The x and y sub-controllers have been optimised last as the thrust and attitude angles determine the translational motion. They need to be optimised based on the optimised rotation sub-controllers and the altitude sub-controller. The process is the same with the altitude sub-controller. Due to the similarity of the two translational movements, the two sub-controllers use the same coefficients. The original and optimal coefficients are shown in Table 4.

Table 4. The original and optimised coefficients of other sub-controllers.

Coefficient	m_3	n_5	n_6	m_4	n_7	n_8	m_5&m_6	n_9&n_{11}	n_{10}&n_{12}
Original	1.60	0.90	3.20	5.00	10.00	3.00	2..50	1.00	1.00
Optimal	15.14	11.40	12.67	6.91	9.96	6.65	3.81	4.68	3.89
No. of Iterations		5			10			3	

The optimisation effect of the roll angle sub-controller has been certified in a test bench. In addition, the entire flight controller has been verified in simulation with a designed trajectory.

5. Test Bench

A preliminary test bench was carried out to test the flight controller. This test bench is a two-motor system, motors are fixed to the ends of a carbon fibre board, and an axis is fixed in the middle with a bearing, as shown in Figure 6. The system has one degree-of-freedom motion, the roll angle. The test bench helps lock some degrees of freedom to avoid system damage and reduce control complexity [6]. This test bench is equivalent to the roll motion of the drone; it was used to test the roll sub-controller. Using the following equation:

$$I_{test}\ddot{\theta}_t = bl_t(\Omega_1^2 - \Omega_2^2) \tag{44}$$

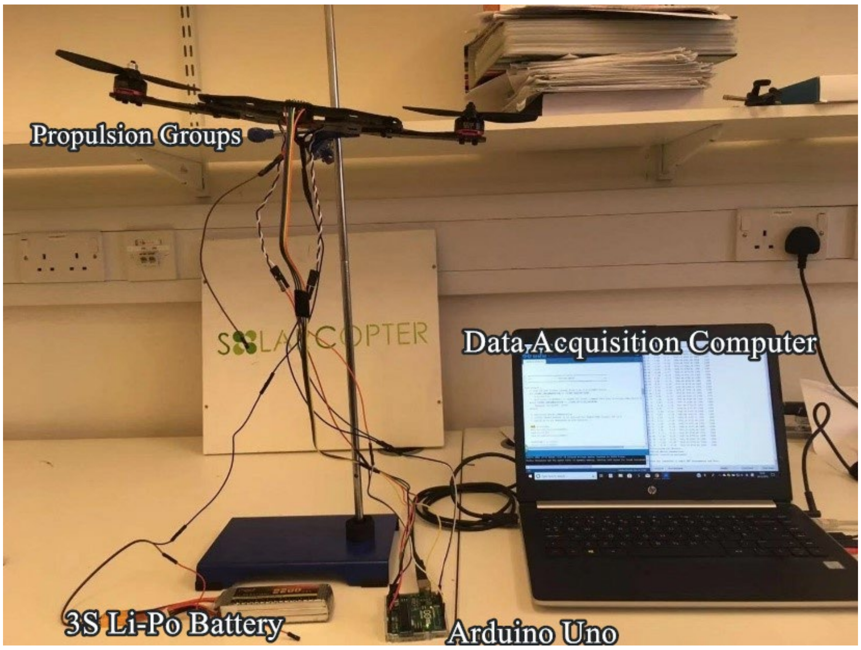

Figure 6. Two motors test bench.

Where θ_t is the angle between the system and horizontal plane, it can be measured by an IMU, l_t is the distance between propeller and axis, which is easy to measure, Ω_1 and Ω_2 are the speeds of the two propellers. I_{test} is the moment of inertia of this testing device. In this experiment, the moment of inertia of the system is measured by the oscillation of a bifilar suspension (Figure 7).

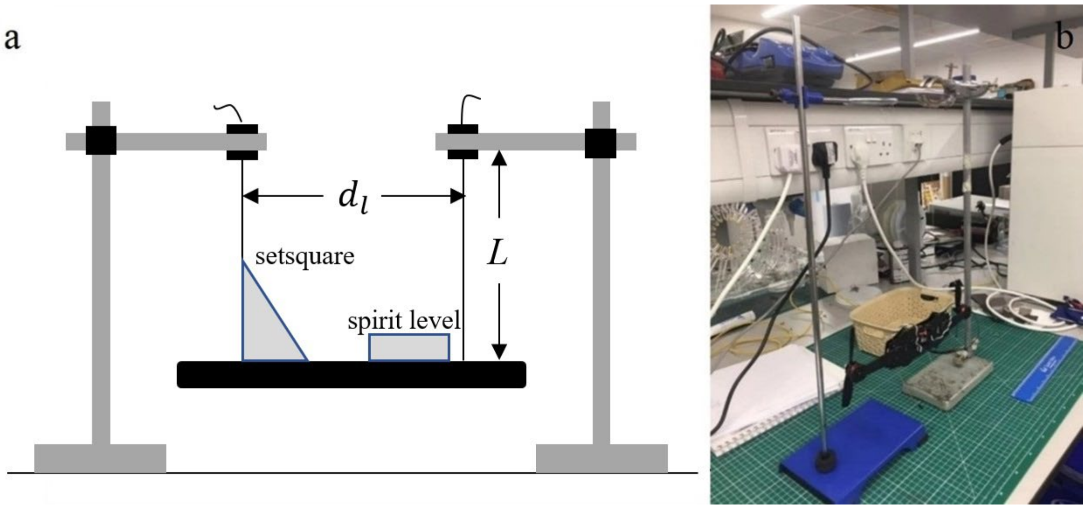

Figure 7. Oscillation of a bifilar Suspension ((**a**): Schematic; (**b**): Physical).

$$I_{test} = \frac{m_t g {t_b}^2 {d_l}^2}{16\pi^2 L} \tag{45}$$

The moment of inertia I_{test}can be calculated using Equation (45). m_t is the mass of the system, g is the acceleration due to gravity. L is the rope length; $d_l$is the distance between the wires; and t_b is the period of the swing, which a stopwatch can measure. This test bench was applied to test the different controllers in the roll angle step response; the control algorithm was translated to codes and implemented to a microcontroller (Arduino Uno).

6. Results

The optimised roll sub-controller was verified in the test bench, as shown in Figure 8 and Table 5. In quantitative analysis, the integration of absolute error (IAE, Equation (46)) is also used in the analysis. This indicator does not take time into account. IAE is more valuable in trajectory tracing. In Table 4, the IAE and ITAE in the simulation, as well as in the experiment are compared; E-ITAE and E-IAE refer to the ITAE and IAE in the experiment. In this step response, the step time is 0, so the ITAE is smaller than the simulation in the optimisation process. In the simulation, the IAE and ITAE were reduced by 39.13 and 58.33, respectively, while in the experiment, the E-IAE and E-ITAE were reduced by 19.35 and 10.39, respectively. The situation is caused by the vibration in the results given by the sensor, nevertheless, the effect of the optimisation is remarkable. Moreover, due to the results' vibration, in Figure 8, it is difficult to visualise the difference between the original system and the optimised system.

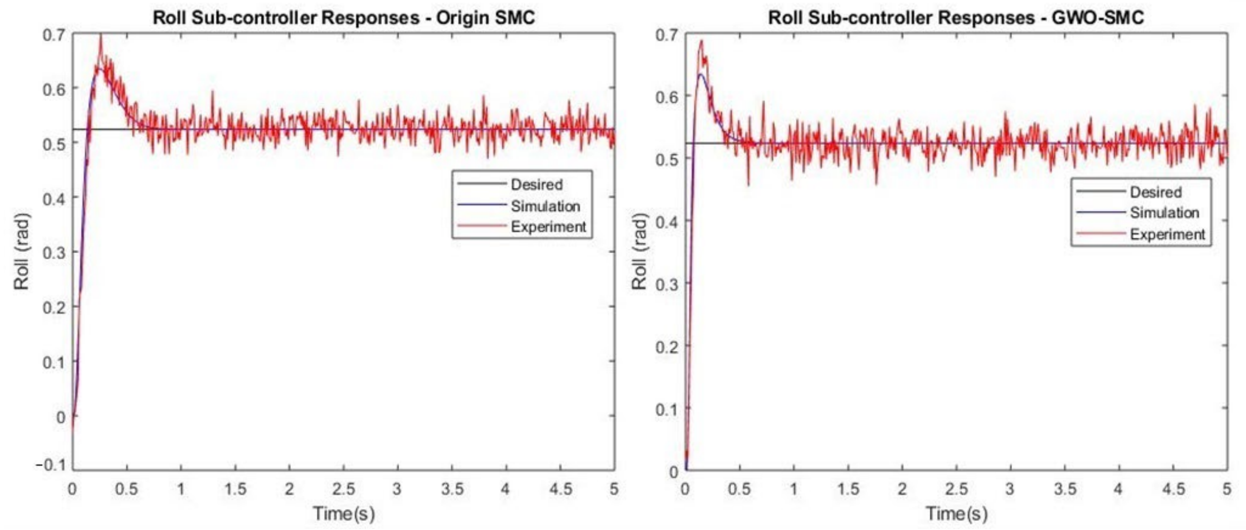

Figure 8. Comparison between original and GWO improved roll sub-controller in the experiment.

$$\text{IAE} = \int_0^{\infty} |e(t)|dt \quad (46)$$

Table 5. Performance of the optimised roll sub-controller.

Indicator	IAE (*rad s*)	ITAE (*rad s*^{2})	E-IAE (*rad s*)	E-ITAE (*rad s*^{2})
Origin	0.069	0.012	0.155	0.231
GWO	0.042	0.005	0.125	0.207

The whole optimised flight controller has been implemented in Simulink for simulation with the mathematical model. First, the inputs of the controller have to be confirmed. During the control of the autonomous flight, the system's inputs are the desired position (x, y, z) and yaw angle (ψ). The outputs of the flight controller are the commanded speeds of the motors; the commanded speeds are inputted into the dynamical model for simulation.

A trajectory was designed to test the flight controller to face extreme conditions. In the test trajectory, the drone rises 2 metres in the first second and then starts moving along a circular trajectory, and simultaneously the drone should rotate 360 degrees; the simulation time is 20 s. All the six degrees of freedom motion are involved in this trajectory. The equation of trajectory is as follows:

$$\begin{gathered} x_d(t) = x_{9d}(t) = \begin{cases} 0, & 0 \le t < \pi \\ -2sin\,(t), & \pi \le t < 3\pi \\ 0, & t \ge 3\pi \end{cases} \\ y_d(t) = x_{11d}(t) = \begin{cases} 0, & 0 \le t < \pi \\ 2\cos(t) + 2, & \pi \le t < 3\pi \\ 0, & t \ge 3\pi \end{cases} \\ z_d(t) = x_{7d}(t) = \begin{cases} 2t, & 0 \le t < 2 \\ 2, & t \ge 2 \end{cases} \\ \psi_d(t) = x_{5d}(t) = \frac{\pi}{10}t \end{gathered} \quad (47)$$

The trajectory is shown graphically in Figure 9. The designed trajectory was used to test both the original and optimised flight controller, and the results are presented in Figures 10 and 11.

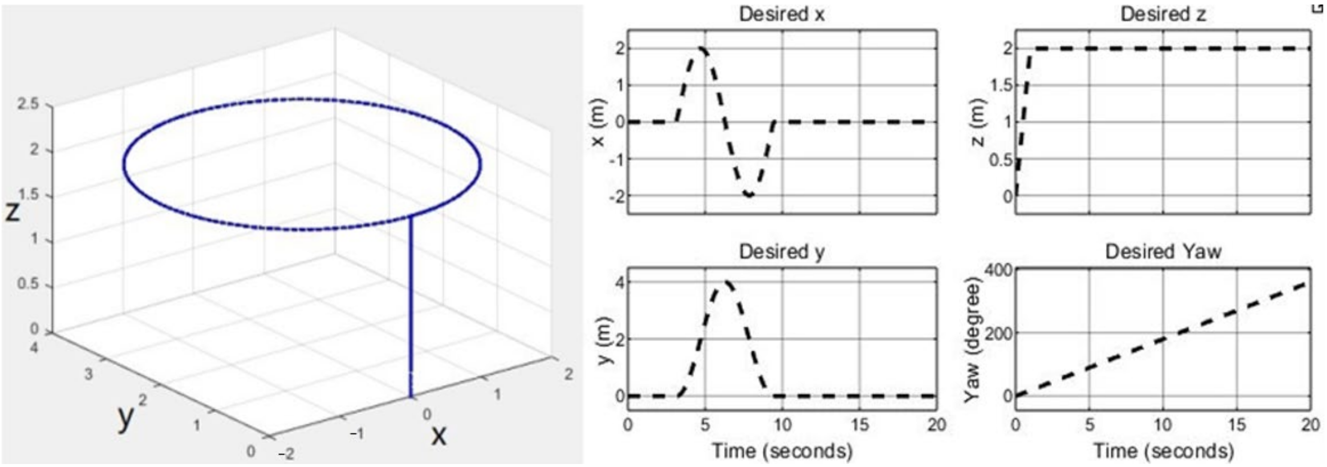

Figure 9. Test trajectory in the 3D coordinate system.

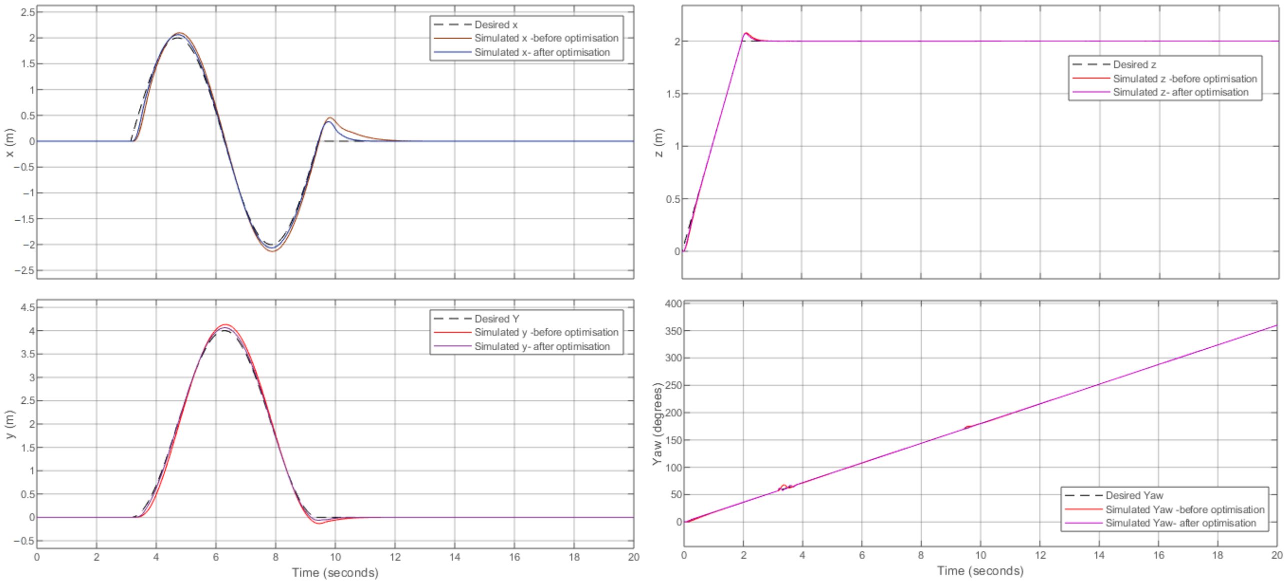

Figure 10. Control performance in the desired trajectory.

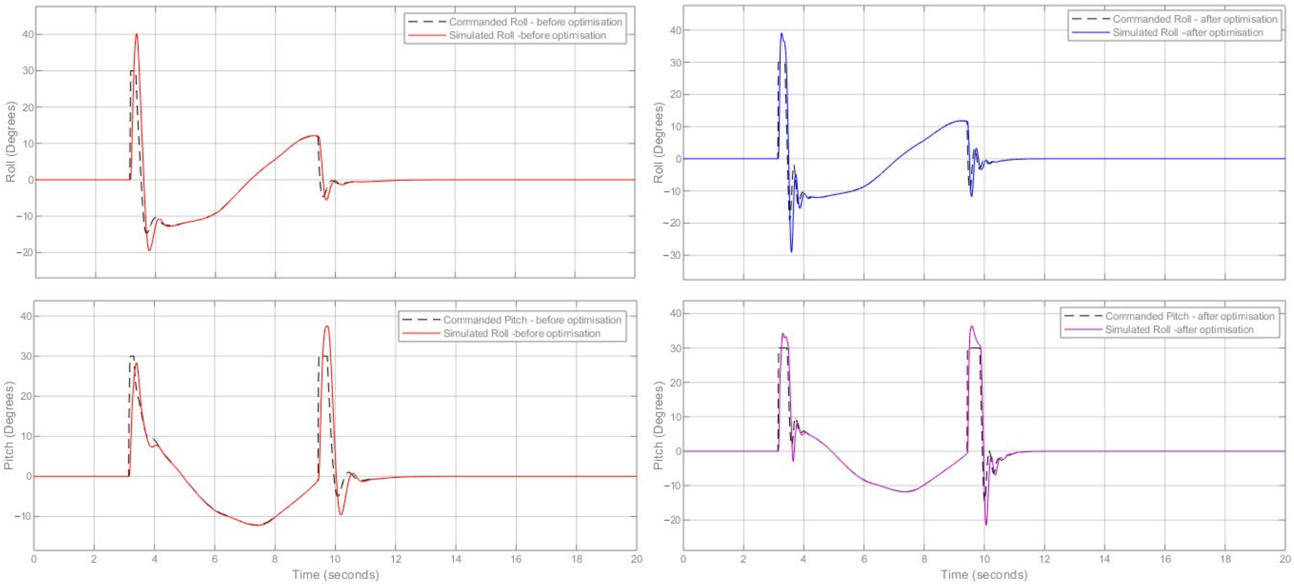

Figure 11. Control performance in roll and pitch.

From 0 to 2 s, the drone started to rotate and rise; there was no translational movement or roll and pitch. From the πth second, the drone was required to move towards both x and y directions, so the commanded roll and pitch angle appeared. The commanded roll and pitch angle is limited to +/− 30 degrees in both original and optimised controllers to maintain a stable flight. The two figures clearly show that the drone with the optimised controller follows the designed trajectory more closely. This result does not only come from intuition but is confirmed numerically.

Quantitative analysis is more convincing. In Table 6, both ITAE and IAE in six degrees of freedom motion have been compared. ITAE is a kind of time-weighted measurement. Applying ITAE on step response is necessary, since it is important to consider a steady-state error. However, for this trajectory, some significant errors appear in the late stage due to the complexity of the desired value; it is not reasonable to compare ITAE in this case. So, the measurement without time weighted or IAE has been compared too. However, all the error measurements of the drone with optimised sub-controllers are smaller.

Table 6. Original and optimised comparison in IAE and ITAE.

	IAE-Roll (*deg s*)	**ITAE-Roll (*deg s*^{2})**	**IAE-Pitch (*deg s*)**	**ITAE-Pitch (*deg s*^{2})**	**IAE-Yaw (*deg s*)**	**ITAE-Yaw (*deg s*^{2})**
Origin	11.79	55.24	14.51	112.40	5.185	23.04
GWO	9.96	51.7	11.26	81.36	1.39	5.68
Improvement (%)	*15.6*	*6.4*	*22.4*	*27.6*	*73.2*	*73.4*
	IAE-*x* (*ms*)	**ITAE-*x* (*ms*^{2})**	**IAE-*y* (*ms*)**	**ITAE-*y* (*ms*^{2})**	**IAE-*z* (*ms*)**	**ITAE-*z* (*ms*^{2})**
Origin	1.15	8.73	0.80	4.53	0.06	0.09
GWO	0.60	4.37	0.29	1.92	0.04	0.06
Improvement (%)	*47.8*	*50.0*	*63.8*	*57.6*	*33.3*	*33.3*

Table 6 also gives the percentage of the improvement. The most improved is the yaw sub-controller with 73.2% error reduction. The performance of the x and y sub-controllers has also improved by close to or more than 50%. Relatively small improvements were made to the altitude controller. Mainly the coefficients of the original z sub-controller are closer to optimal compared to the other sub-controllers. The above four sub-controllers, which receive direct commands regarding the desired trajectory, have all been significantly optimised. The roll and pitch controllers receive instructions from the position sub-controllers. When the position sub-controllers are optimised, the angle commands are given more aggressively to achieve a fast response. This poses a greater challenge to the roll and pitch sub-controllers. Despite this, the cumulative control error of the roll and pitch sub-controllers for the same trajectory is reduced by 15.6% and 22.4%. This is a further indication of the success of the optimisation.

7. Conclusions

The nonlinear dynamical model of a hexacopter drone has been simulated, and a flight controller based on sliding mode control theory has been developed and applied to the drone. The drone can fly autonomously following a desired trajectory. GWO was chosen to optimise the entire SMC flight controller on the basis of a comparative performance study of PSO and GWO on a sub-controller. The optimised sub-controller has improved in rise time, peak time, overshot, settle time, and ITAE. In theory, optimisation for a single angle sub-controller could result in a 58.3% reduction in ITAE. However, due to equipment limitations, only a 10.4% improvement was measured in the experiment. The entire flight controller has also been optimised. For the same trajectory tracking, the optimised controller shows a significant reduction in the IAE in each degree of freedom compared to the original controller. The optimisation is most evident in the yaw angle, where the cumulative error is reduced by 73.2%. In practice, these optimisations enable the drone to get to the desired location faster and more accurately and could make surveillance or information collection via cameras more accurate.

Author Contributions: Conceptualisation, C.S. and M.H.S.; methodology, C.S. and M.H.S.; software, C.S.; validation, C.S.; formal analysis, C.S.; investigation, C.S.; resources, C.S. and M.H.S.; data curation, C.S. and S.A.A.; writing—original draft preparation, C.S. and S.A.A.; writing—review and editing, M.H.S., Z.M. and S.A.A.; visualization, C.S; supervision, M.H.S. and Z.M. project administration, C.S. and M.H.S.; funding acquisition, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The study did not report any data.

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

References

1. Manarvi, I.A.; Aqib, M.; Ajmal, M.; Usman, M.; Khurshid, S.; Sikandar, U. Design and development of a quad copter (UMAASK) using CAD/CAM/CAE. In Proceedings of the 2013 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2013; pp. 1–10. https://doi.org/10.1109/aero.2013.6497379.
2. Sun, C. Design, Modelling and Control of a Solar-powered Rotary Aerial Vehicle. Ph.D. Thesis, Queen Mary University of London, London, UK, 2021.
3. Salih, A.L.; Moghavemmi, M.; Mohamed, H.A.F.; Gaeid, K.S. Modelling and PID Controller Design for a Quadrotor Unmanned Air Vehicle. In Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 28–30 May 2010; pp. 1–5.
4. Mustapa, Z.; Saat, S.; Husin, S.H.; Abas, N. Altitude controller design for multi-copter UAV. In Proceedings of the 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2–4 September 2014; pp. 382–387. https://doi.org/10.1109/i4ct.2014.6914210.
5. Agha, S.A. A Systematic Approach to Design and Control of a Solar-Powered Multi-Rotor Unmanned Aerial Vehicle. Ph.D. Thesis, Queen Mary University of London, London, UK, 2019.
6. Abidali, A. Design, Fabrication, and Control of a Solar-Powered Multi-Rotor Micro Aerial Vehicle. Ph.D. Thesis, Queen Mary University of London, London, UK, 2018.
7. Bouabdallah, S. Design and Control of Quadrotors with Application to Autonomous Flying. Ph.D. Thesis, École Polytechnique Fédéral de Lausanne, Lausanne, Switzerland, 2007.
8. Bouabdallah, S.; Siegwart, R. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In Proceedings of the International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2247–2252. https://doi.org/10.1109/robot.2005.1570447.
9. Bouabdallah, S.; Siegwart, R. Full control of a quadrotor. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 153–158. https://doi.org/10.1109/iros.2007.4399042.
10. Ricardo, J.A., Jr.; Santos, D.A. Smooth second-order sliding mode control for fully actuated multirotor aerial vehicles. *ISA Trans.* **2022**, 1–10. https://doi.org/10.1016/j.isatra.2022.01.028.
11. Moya, V.; Espinosa, V.; Chavez, D.; Leica, P.; Camacho, O. Trajectory tracking for quadcopter's formation with two control strategies. In Proceedings of the 2016 IEEE Ecuador Technical Chapters Meeting (ETCM), Guayaquil, Ecuador, 12–14 October 2016; pp. 1–6. https://doi.org/10.1109/etcm.2016.7750839.
12. Chiew, T.H.; Lee, H.-E.; Lee, Y.K.; Chang, K.M.; Ong, J.J.; Eu, K.S. Second Order Sliding Mode Controller for Altitude and Yaw Control of Quadcopter. In Proceedings of the 2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 27–28 August 2021; pp. 97–102. https://doi.org/10.1109/iccsce52189.2021.9530850.
13. Abrougui, H.; Nejim, S.; Charrada, A.; Dallagi, H. Sliding Mode Control for Hexacopter Autopilot design. In Proceedings of the 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia, 22–25 March 2022; pp. 445–449. https://doi.org/10.1109/ic_aset53395.2022.9765926.
14. Concha, A.; Thenozhi, S.; Betancourt, R.J.; Gadi, S. A tuning algorithm for a sliding mode controller of buildings with ATMD. *Mech. Syst. Signal Process.* **2021**, *154*, 107539. https://doi.org/10.1016/j.ymssp.2020.107539.
15. Nekoukar, V.; Dehkordi, N.M. Robust path tracking of a quadrotor using adaptive fuzzy terminal sliding mode control. *Control Eng. Pr.* **2021**, *110*, 104763. https://doi.org/10.1016/j.conengprac.2021.104763.
16. Liu, M.; Chen, Z.; Sun, Z. Sliding mode control based on particle swarm optimization and support vector machine. In Proceedings of the 2011 9th World Congress on Intelligent Control and Automation, Taipei, Taiwan, 21–25 June 2011; pp. 260–264. https://doi.org/10.1109/wcica.2011.5970739.

17. Kumar, A.; Anwar, N.; Kumar, S. Sliding mode controller design for frequency regulation in an interconnected power system. *Prot. Control Mod. Power Syst.* **2021**, *6*, 6. https://doi.org/10.1186/s41601-021-00183-1.
18. Noordin, A.; Basri, M.A.M.; Mohamed, Z.; Abidin, A.F.Z. Modelling and PSO Fine-tuned PID Control of Quadrotor UAV. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2017**, *7*, 1367. https://doi.org/10.18517/ijaseit.7.4.3141.
19. Haitao, Y.; Yan, L. Research on Attitude Control for UAV Based on Improved PSO Algorithm. In Proceedings of the 2021 6th International Conference on Robotics and Automation Engineering (ICRAE), Guangzhou, China, 19–22 November 2021; pp. 46–49. https://doi.org/10.1109/icrae53653.2021.9657828.
20. Qu, C.; Gai, W.; Zhang, J.; Zhong, M. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. *Knowl.-Based Syst.* **2020**, *194*, 105530. https://doi.org/10.1016/j.knosys.2020.105530.
21. Patel, K.; Barve, J. Modeling, simulation and control study for the quad-copter UAV. In Proceedings of the 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, India, 15–17 December 2014; pp. 1–6. https://doi.org/10.1109/iciinfs.2014.7036590.
22. Dong, W.; Gu, G.; Zhu, X.; Ding, H. *Modeling and Control of a Quadrotor UAV with Aerodynamic Concepts*; World Academy of Science: Shanghai, China, 2013.
23. Ye, J.; Wang, J.; Song, T.; Wu, Z.; Tang, P. Nonlinear Modeling the Quadcopter Considering the Aerodynamic Interaction. *IEEE Access* **2021**, *9*, 134716–134732. https://doi.org/10.1109/access.2021.3116676.
24. Baranek, R.; Solc, F. Modelling and control of a hexa-copter. In Proceedings of the 13th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 28–31 May 2012; pp. 19–23. https://doi.org/10.1109/carpathiancc.2012.6228609.
25. Alaimo, A.; Artale, V.; Milazzo, C.; Ricciardello, A.; Trefiletti, L. Mathematical modeling and control of a hexacopter. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 1043–1050. https://doi.org/10.1109/icuas.2013.6564793.
26. Garcia-Gabin, W.; Dorado, F.; Bordons, C. Real-time implementation of a sliding mode controller for air supply on a PEM fuel cell. *J. Process Control* **2010**, *20*, 325–336. https://doi.org/10.1016/j.jprocont.2009.11.006.
27. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software*. **2014**, *69*, 46–61.