algorithms

MDPI

Article

Performance Enhancement of Quadrotor UAVs via Gray Wolf Optimized Algorithm for Sliding Mode Control

Mustafa B. Nidham [1], Khalid Yahya [2,*], Mehdi Safaei [3,*], Nawal Rai [4] and Saleh Al Dawsari [5,6,*]

[1] Department of Aeronautical Engineering, College of Engineering, University of Baghdad, Baghdad 10071, Iraq; heromustafa99@gmail.com
[2] Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul 34310, Turkey
[3] Faculty of Economics, Administrative and Social Sciences, Logistics Management Department, Istanbul Gelisim University, Istanbul 34310, Turkey
[4] Materials, Energy Systems Technology and Environment Laboratory, University of Ghardaia, Ghardaïa 47000, Algeria; rainawel27@gmail.com
[5] School of Engineering, Cardiff University, Cardiff CF24 3AA, UK
[6] Electrical Engineering Department, College of Engineering, Najran University, Najran P.O. Box 1988, Saudi Arabia
* Correspondence: hayahya@gelisim.edu.tr (K.Y.); msafaei@gelisim.edu.tr (M.S.); aldawsarisa@cardiff.ac.uk (S.A.D.)

Abstract

This article is an in-depth analysis of the performance and efficiency of various control systems used in quadrotor unmanned aerial vehicles (UAVs). The study is focused on the comparison of three main control approaches, including Sliding Mode Control (SMC), Fuzzy Logic Control (FLC), and an extended version of Sliding Mode Control with the use of the Gray Wolf Optimizer (SMC-GWO), as well as a supportive validation model the Genetic Algorithm (SMC-GA). Based on the Newton–Euler formulation, the mathematical model of a quadrotor has been developed to provide a true picture of the dynamic behavior of the quadrotor. The model was then implemented in MATLAB/Simulink 2025b to test the performance of the system in its nominal and perturbed conditions. The findings have shown that the hybrid SMC-GWO controller has significant improvement in response speed, accuracy, and stability compared to the other controllers. Precisely, the SMC-GWO demonstrated 78.46 percent decrease in rise time and 23.40 percent decrease in settling time compared to the traditional SMC, as well as a nearly negligible steady-state error (SSE = 0.0008) in the roll channel. The proposed controller in the pitch channel reduced the rise time by 93.65 percent and the settling time by 20.22 percent, with a much smoother and more stable tracking and an effectively negligible steady-state error (SSE = 0.0001). The hybrid controller in the yaw channel had a 77.94 percent better rise time and 23.16 percent better settling time, resulting in a steady-state error of 0.0022. In relation to altitude control, SMC-GWO decreased the rise time by 91.87 percent and settling time by 25.04 percent over classical SMC, yet the steady-state error was almost zero. Under constant, time-varying actuator disturbances, the SMC-GWO controller also demonstrated better system stabilization and trajectory-tracking behavior than both SMC and FLC, as well as slightly better behavior than SMC-GA in the presence of faults and disturbances. These results verify that a UAV control framework based on the combination of the Gray Wolf Optimizer and Sliding Mode Control is more resilient, quick, and significantly more precise.

Keywords: SMC; FLC; GWO; GA; quadrotor; UAV; MATLAB/Simulink

Check for updates

Academic Editors: Ramiro Barbosa, Paulo Moura Oliveira and Isabel Jesus

Received: 31 January 2026
Revised: 20 April 2026
Accepted: 22 April 2026
Published: 24 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

1.1. Background

Unmanned aerial vehicles (UAVs), commonly known as drones, are aircraft that operate without an onboard pilot and can be controlled either remotely or autonomously. In recent years, UAV technology has developed rapidly due to advances in control systems, artificial intelligence, and embedded electronics. UAVs can generally be classified into fixed-wing and rotary-wing platforms according to their aerodynamic structure and flight mechanism. Fixed-wing UAVs provide higher efficiency and longer endurance but require a runway for take-off and landing, whereas rotary-wing UAVs enable vertical take-off and landing (VTOL) and high maneuverability, although with lower flight endurance and payload capability [1,2].

Quadrotor UAVs are among the most widely used rotary-wing platforms due to their simple mechanical structure, hovering capability, and high maneuverability [3–5]. These vehicles generate lift using four rotors arranged symmetrically, where two rotors rotate clockwise and the other two rotate counterclockwise to maintain torque balance and flight stability [6,7]. Despite their advantages, quadrotors are highly nonlinear, underactuated, and strongly coupled systems, which makes the design of robust and accurate controllers a challenging task, especially in the presence of disturbances and model uncertainties [8]. Therefore, the development of advanced control strategies for quadrotor UAVs has become an important research topic in recent years.

1.2. Literature Review

Control systems play a fundamental role in ensuring the stability and trajectory-tracking performance of quadrotor UAVs. Traditional linear controllers, such as Proportional–Integral–Derivative (PID) controllers, are widely used due to their simplicity and ease of implementation. However, linear controllers often exhibit limited performance when dealing with nonlinear dynamics, parameter variations, and external disturbances [9–11].

To overcome these limitations, nonlinear control approaches such as Sliding Mode Control (SMC) and Backstepping have been extensively investigated [12,13]. Among these methods, SMC is well known for its robustness against disturbances and model uncertainties, since it forces the system states to converge toward a predefined sliding surface, ensuring stable and accurate tracking performance [14]. However, classical SMC suffers from the chattering phenomenon caused by the discontinuous control law, which may lead to actuator wear and degraded control performance.

To improve controller performance, intelligent and optimization-based methods have also been introduced in UAV control systems. Fuzzy Logic Control (FLC) has been used to enhance adaptability in nonlinear and uncertain environments [15]. In addition, hybrid and optimization-based approaches have attracted increasing attention because they aim to improve precision, robustness, and dynamic response [16]. Recently, advanced Fuzzy Logic Control (FLC) strategies have been extensively investigated for nonlinear systems subject to uncertainties and communication constraints. For instance, quantized fuzzy feedback control has been proposed for systems with discretization and communication limitations, including applications in electric vehicle lateral dynamics [17]. In addition, adaptive fuzzy control approaches with guaranteed performance have demonstrated strong capability in handling nonlinear systems with uncertainties and actuator faults [18]. These studies highlight the effectiveness and flexibility of Fuzzy Logic Control in dealing with nonlinearities and uncertainties, which supports its use as a comparative control strategy in the present quadrotor UAV study.

Evolutionary optimization techniques such as Genetic Algorithm (GA) have also been applied to nonlinear controllers for quadrotor systems and have shown improved tracking performance [19].

Although previous studies have demonstrated the effectiveness of nonlinear and optimization-based control methods, many of them focus on a single controller or a single optimization technique. As a result, there is still a need for a unified comparative evaluation of conventional, intelligent, and optimized controllers under identical simulation conditions, particularly in the presence of disturbances.

1.3. Research Gap and Contribution

Although SMC provides strong robustness, its performance depends heavily on appropriate parameter tuning, and the classical design may still suffer from chattering and suboptimal transient response [20]. Optimization-based tuning methods can improve controller performance; however, previous studies have often focused on a single optimization strategy without providing a consistent comparison with other controllers under the same quadrotor model and operating conditions. Therefore, this study proposes a hybrid control framework that combines Sliding Mode Control with the Gray Wolf Optimizer (SMC–GWO) based on the hunting behavior of gray wolves [21]. Unlike previous studies, this work presents a unified comparative evaluation including SMC, FLC, SMC–GA, and SMC–GWO using the same quadrotor dynamic model and identical simulation conditions under both nominal and disturbed scenarios. The main contribution of this paper lies in providing a comprehensive and consistent performance comparison of these controllers in terms of response speed, settling time, overshoot, and steady-state error. The results demonstrate that the optimized SMC–GWO controller achieves improved response and robustness compared with the conventional approaches considered in this study.

2. Quadrotor Dynamics and Modeling

The quadrotor mathematical equations are obtained through the Newton–Euler method. Despite the fact that the quadcopter has six degrees of freedom (6-DOF), it is an underactuated system because of the few actuators, that is, four rotors. These rotors are placed orthogonally in an X-configuration, with two rotating clockwise and the remaining two rotating anticlockwise. This counter-rotation is necessary in order to produce balanced torques and provide dynamic stability to the vehicle. The quadcopter rotates and stabilizes using the differential control of the rotating speed of the four propellers. The quadcopter can be controlled to move in all horizontal directions (forward, backward, left, and right) by modulating the velocity of each motor and also execute rotational motions around its three major axes. The basic motion elements of a quadcopter can generally be separated into four elements which include altitude control, roll, pitch and yaw. Holding the altitude is achieved by equally decreasing or accelerating the rotor speeds of all four rotors by throttle control. When the propellers are used with lower speeds, the quadcopter will fall. On the other hand, once the group speed reaches a point that creates an equivalent lift to the pull of gravity, the quadcopter is suspended. The rotational movement along the longitudinal (X) axis, referred to as roll motion, is induced by changing the thrust between the left and right pairs of rotors. As an example, acceleration of the rotors on the right side leads to a tilting to the left, and the other way around. Pitch motion rotation about the lateral (Y) axis is actuated by changing the thrust between the front and rear rotors. An increase in the speed of the forward propellers triggers a forward tilt, thus pushing the vehicle forward. The imbalance of torques caused by the counter-rotating groups of propellers causes yaw movement or rotation about the vertical (Z) axis. When propellers on the right side are rotating at higher speeds than the left ones the vehicle will rotate in a

counterclockwise motion. On the other hand, a faster rotation of the propellers on the left side leads to clockwise rotation [22,23]. Angular variables of roll, pitch, and yaw are usually defined relative to an Earth-fixed reference (A- rame), whereas linear acceleration and body dynamics are determined in a body-fixed reference frame (B frame). The quadcopter layout of the X- layout with the orientation of the A and B frames is shown in Figure 1. The fixed-frame that is represented by A is the earth and the fixed-frame that is represented by B is the body. There are three translational states (x, y, z), and three rotational states (ϕ, θ, ψ), with x, y, and z being the position coordinates. ϕ, θ, and ψ are the roll, pitch and yaw angles [24–26].

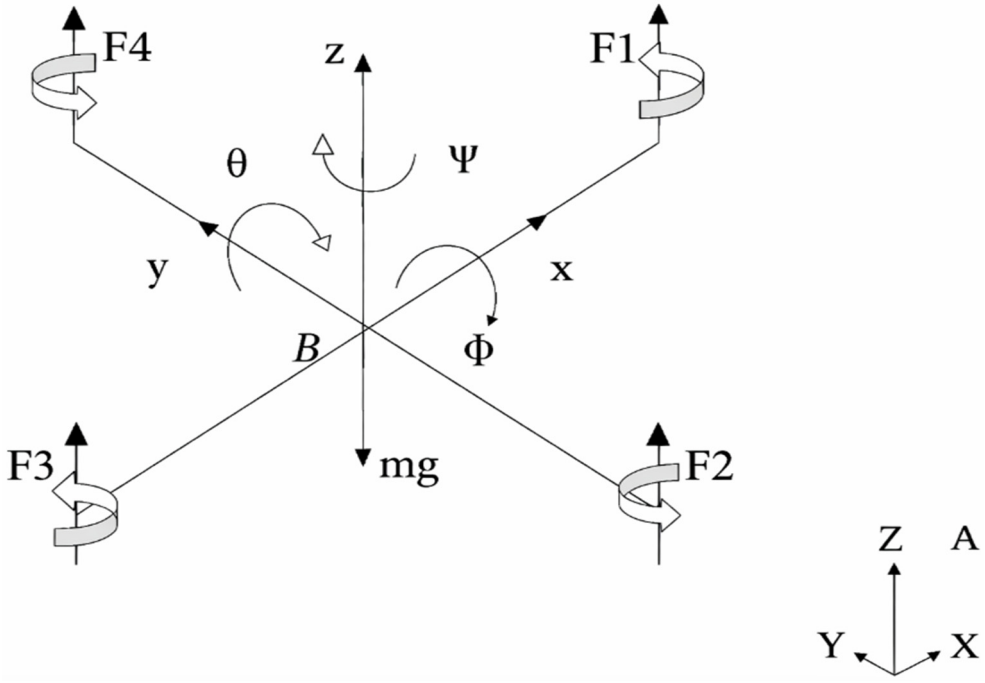


Figure 1. The forces which affect quadrotors [27].

The mathematical equations of the quadrotor are derived using the Newton–Euler formulation, as presented in [27,28]. Under the following assumptions, the kinematics and dynamics models of a quadrotor will be derived.

1. The quadrotor's frame is supposed to be inflexible.
2. The entire configuration is defined as symmetric.
3. The point of origin of the body-fixed coordinate system is the same as that of the center of gravity.
4. The rotors are represented as rigid propellers.
5. The thrust force and reactive drag torque generated by each rotor are assumed to be proportional to the square of the rotor angular velocity.
6. The aerodynamic drag acting on the quadrotor body is neglected to obtain a simplified model suitable for controller design under low-speed flight conditions. The robustness of the proposed controller is evaluated with respect to bounded disturbances and modeling uncertainties.

Quadrotor dynamics are then developed by the Newton–Euler equations of translational and rotational motion, which are:

$$\begin{pmatrix} \ddot{x} = (cos\phi sin\theta cos\psi + sin\phi sin\psi) \cdot \frac{1}{m} U_1 \\ \ddot{y} = (cos\phi sin\theta sin\psi - sin\phi cos\psi) \cdot \frac{1}{m} U_1 \\ \ddot{z} = g - (cos\phi cos\theta) \cdot \frac{1}{m} U_1 \\ \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) + \frac{J_r}{I_x}\dot{\theta} \cdot \Omega_r + \frac{l}{I_x} U_2 \\ \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) - \frac{J_r}{I_y}\dot{\phi} \cdot \Omega_r + \frac{l}{I_y} U_3 \\ \ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z} U_4 \end{pmatrix} \quad (1)$$

In Equation (1), the mass of the quadrotor is denoted by m [kg], the inertia matrix elements in the body system are described by Ix [kg·m^2], Iy [kg·m^2], and Iz [kg·m^2], the angular momentum is denoted by J_r [kg·m^2], and the speed of the propeller is denoted by Ω [rad/s]. Total rotor thrust U_1 is the signal input. Roll, pitch, and yaw are represented by moments U_2, U_3, and U_4, correspondingly. One way to provide the inputs is by

$$\begin{pmatrix} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(-\Omega_2^2 + \Omega_4^2) \\ U_3 = b(\Omega_1^2 - \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{pmatrix} \quad (2)$$

In Equation (2), b [N·s^2], and d [N·m·s^2] represent the lift and drag coefficients, respectively. Ω is the general angular speed, and the angular speed for each rotor is $\Omega_1, \Omega_2, \Omega_3, \Omega_4$.

Ω_r is the total rotor angular speed, defined as

$$\Omega_r = d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right)$$

You may rewrite the control inputs in Equation (2) in the matrix as:

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} \quad \begin{pmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ b & 0 & -b & 0 \\ -d & d & -d & d \end{pmatrix} \quad \begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} \Omega_1^2 = \frac{U_1}{4b} + \frac{U_3}{2b} - \frac{U_4}{4d} \\ \Omega_2^2 = \frac{U_1}{4b} - \frac{U_2}{2b} + \frac{U_4}{4d} \\ \Omega_3^2 = \frac{U_1}{4b} - \frac{U_3}{2b} - \frac{U_4}{4d} \\ \Omega_4^2 = \frac{U_1}{4b} + \frac{U_2}{2b} + \frac{U_4}{4d} \end{pmatrix} \tag{4}$$

The equations of motion were formulated in (1). To optimize the model, thrust and drag coefficients are presumed to be invariant. The quadrotor system can be articulated in state-space form. $\dot{x} = f(x \cdot U)$ where x and U are the state and input vectors, respectively. The state vector's characterization and dynamics are presented in Equation (5) and Table 1, respectively, whilst the input vector is delineated in Equation (2).

$$X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} & x & \dot{x} & y & \dot{y} \end{bmatrix} \tag{5}$$

Table 1. State vector of the system.

$x_1 = \phi$	$x_2 = \dot{x}_1 = \dot{\phi}$
$x_3 = \theta$	$x_4 = \dot{x}_3 = \dot{\theta}$
$x_5 = \psi$	$x_6 = \dot{x}_5 = \dot{\psi}$
$x_7 = z$	$x_8 = \dot{x}_7 = \dot{z}$
$x_9 = x$	$x_{10} = \dot{x}_9 = \dot{x}$
$x_{11} = y$	$x_{12} = \dot{x}_{11} = \dot{y}$

On simplification, the resultant equations are the following:

$$f(X,U) = \begin{pmatrix} \dot{\phi} \\ \dot{\theta}\,\dot{\psi}\,a_1 + \dot{\theta}\,a_2\,\Omega + b^1U^2\,\dot{\theta} \\ \dot{\phi}\dot{\psi}a_3 - \dot{\phi}a_4\Omega + b^2U^3 \\ \dot{\Psi} \\ \dot{\theta}\,\dot{\phi}a_5 + b^3U^4 \\ \dot{z} \\ g - (cos(\phi)cos(\theta))\left(\frac{1}{m}\right)U^1 \\ \dot{x} \\ u_x\left(\frac{1}{m}\right)U^1 \\ \dot{y} \\ u_y\,(1/m)\,U_1 \end{pmatrix} \tag{6}$$

With

$$a_1 = \frac{I_y - I_z}{I_x}, a_2 = \frac{J_R}{I_x}, a_3 = \frac{I_z - I_x}{I_y}, a_4 = \frac{J_R}{I_y}, a_5 = \frac{I_x - I_y}{I_z} b_1 = \frac{l}{I_x}, b_2 = \frac{l}{I_y}, b_3 = \frac{l}{I_z}$$

where

$$\begin{pmatrix} u_x = (cos\phi sin\theta cos\psi + sin\phi sin\psi) \\ u_y = (cos\phi sin\theta sin\psi - sin\phi cos\psi) \end{pmatrix} \tag{7}$$

Robotic systems, akin to quadrotors, can be examined by partitioning them into two primary subsystems: the rotating subsystem, which pertains to angular motion, and the translational subsystem, which concerns linear motion. Figure 2 demonstrates that the

rotating subsystem functions autonomously from the translational subsystem, while the translational subsystem is affected by the dynamics of rotation [13].

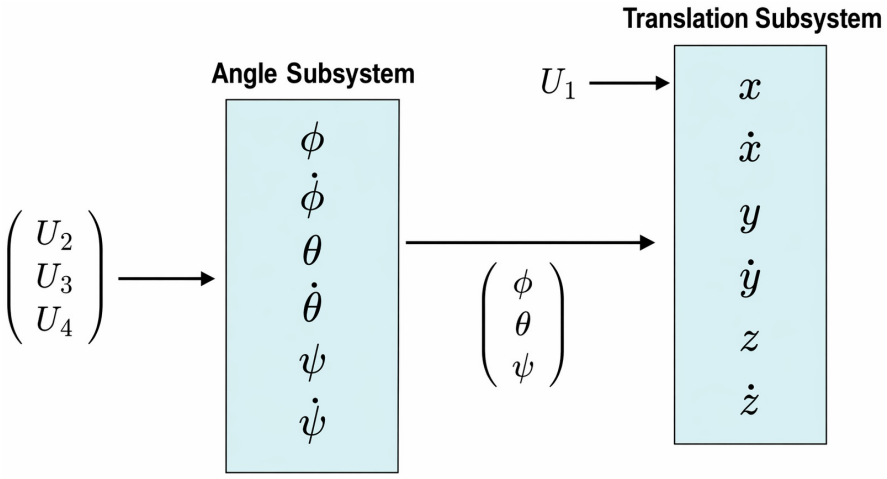


Figure 2. Relation between rotation and translation subsystem [29].

The simulations conducted in this research are based on the system parameters presented in Table 2, which also summarizes the physical specifications of the quadrotor.

Table 2. The quadrotor's parameter values [30].

Parameter	Symbol	Value	Unit
Quadrotor mass	m	6.5×10^{-1}	kg
Moment of inertia about the *x*-axis	I_x	7.5×10^{-3}	kg·m^2
Moment of inertia about the *y*-axis	I_y	7.5×10^{-3}	kg·m^2
Moment of inertia about the *z*-axis	I_z	1.3×10^{-2}	kg·m^2
Thrust coefficient	b	3.13×10^{-5}	N·s^2
Drag coefficient	d	7.5×10^{-7}	N·m·s^2
Rotor inertia	J_r	6.0×10^{-5}	kg·m^2
Arm length	l	2.3×10^{-1}	m

Figure 3 shows the block diagram of the proposed quadrotor control system including the controller, quadrotor model, and feedback signals.

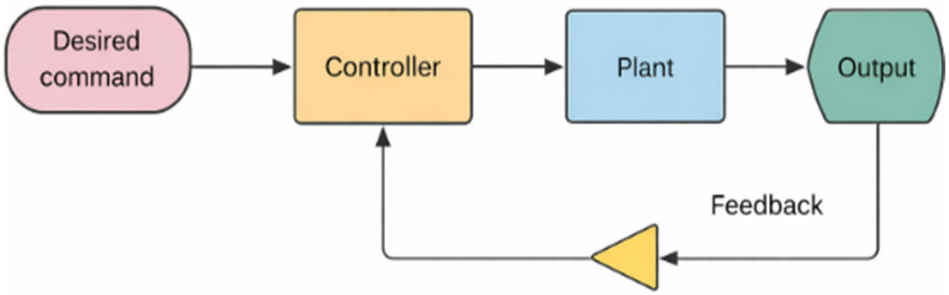


Figure 3. Closed-loop system.

The final open loop Simulink of the quadrotor plant is seen in Figure 4. The Simulink model is structured into three primary subsystems [31]. The Simulink models of each subsystem are depicted as follows:

1. Propeller Velocity Subsystem (Ω) in Figure 5.
2. Angle Subsystem (Yaw, Pitch, Roll) in Figure 6.
3. Translation Subsystem (X, Y, Z) in Figure 7.

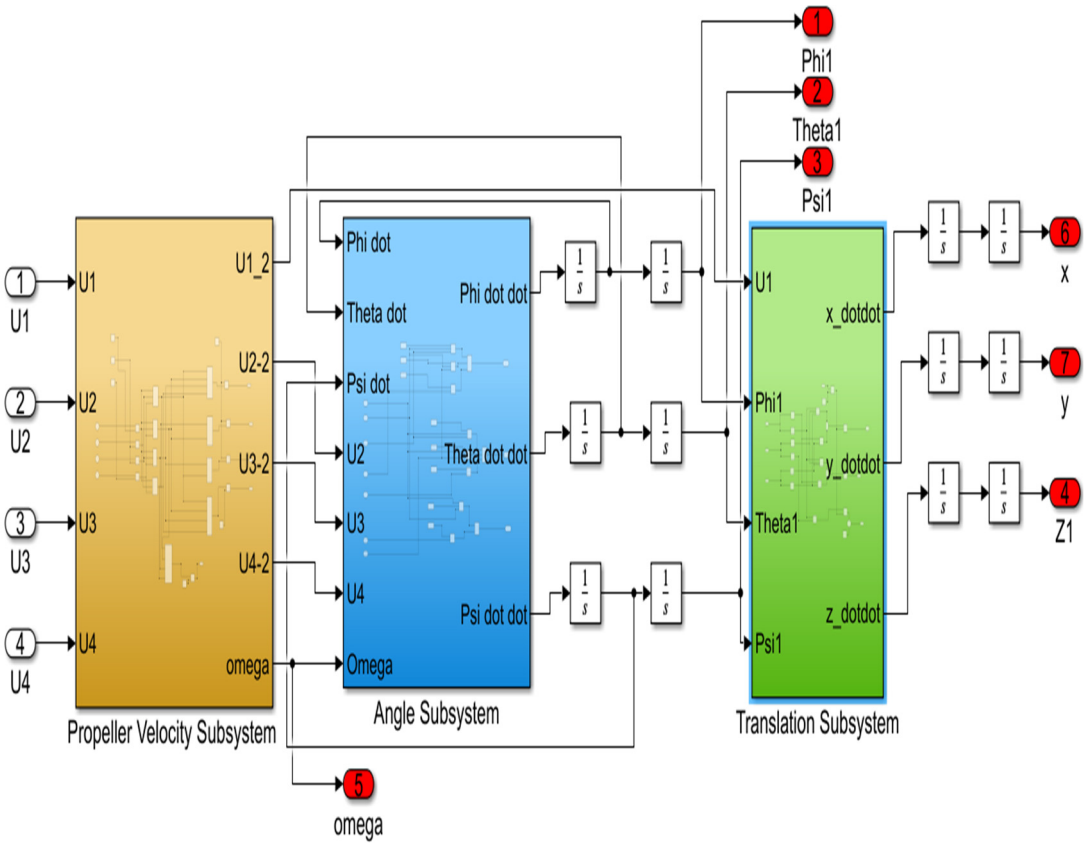


Figure 4. Schematic representation of quadrotor plant.

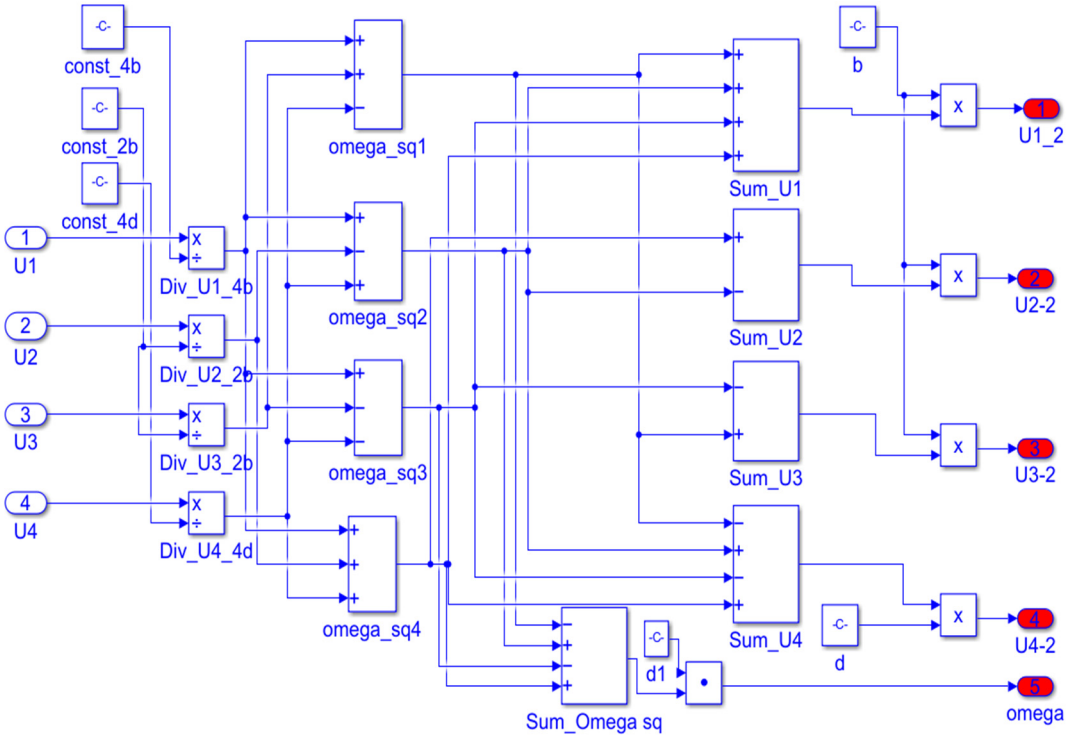


Figure 5. Schematic representation of velocity subsystem.

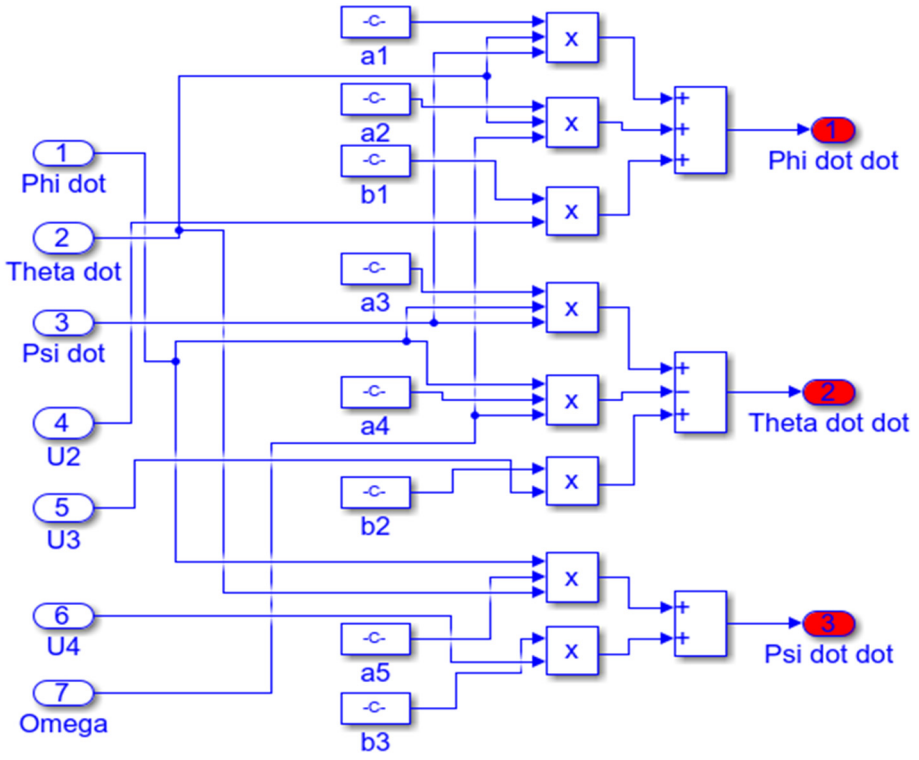


Figure 6. Schematic representation of the angle subsystem.

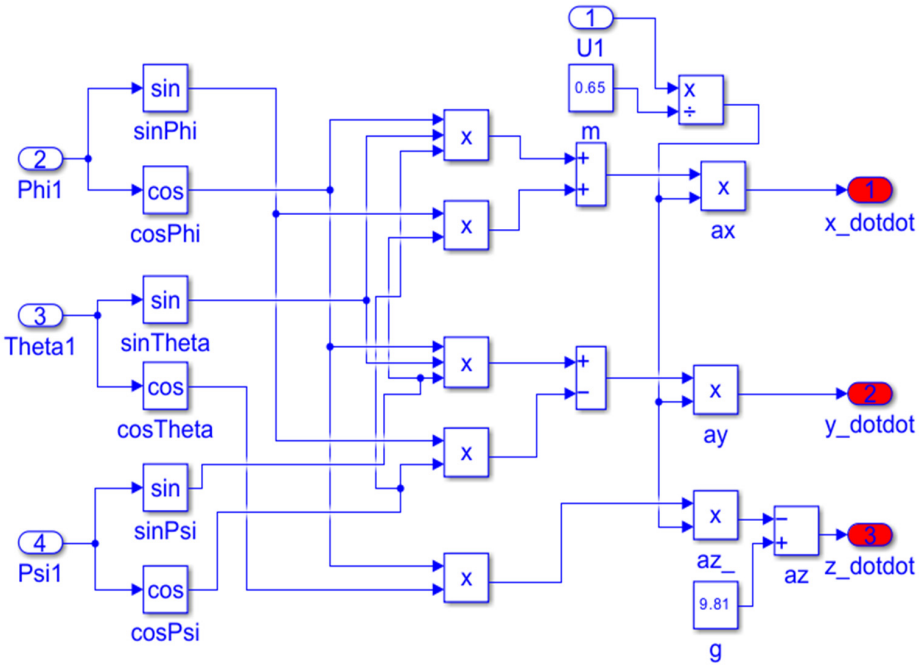


Figure 7. Schematic depiction of the translation subsystem.

3. Control Design

3.1. Sliding Mode Controller (SMC) Design and Implementation

Sliding Mode Control (SMC) is a nonlinear control approach that uses a discontinuous control signal to make the system on a known surface on the state space. In this way, it is easy to alternate between different control structures depending on the current state of the system thus qualifying it as a variable-structure control strategy. The control laws are formulated in a way that causes the system trajectories to move towards the edges of the sliding surface where the system has a sliding motion or as it is also known as the sliding

mode [32]. Furthermore, an advanced quadrotor modeling that relies on the Newton–Euler framework has provided a solid base for the design of SMC [33]. These models show that SMC is successfully able to control not only the attitude (ϕ, θ, ψ) but also the altitude (z), reducing the common effects of chattering and, in turn, providing a smooth and accurate quadrotor control. SMC is also known to resist disturbances and parametric changes, hence ensuring stable operation in the system [14]. The sliding manifold is constructed in the following manner:

$$s(t) = \left(\frac{d}{dt} + \lambda\right)\widetilde{X} \tag{8}$$

With

$$s = \dot{\widetilde{X}} + \lambda\widetilde{X} \tag{9}$$

Define $s(t) \in$ Rn as the equilibrium surface, $X^{\sim}$ as the discrepancy between intended and actual position/orientation, and λ as a positive constant.

The primary function of the SMC is to maintain a constant altitude and attitude for the quadrotor through the process of choosing a candidate function that maps the state variables of the system, which is a Lyapunov function. The system must stay on the predetermined course. Develop the error dynamics as a starting point.

$$z_1 = x_{1d} - x_1 \tag{10}$$

We regard z_1 as positive definite, with its time derivative being negative semi-definite.

$$V(z_1) = \frac{1}{2}z_1^2 \tag{11}$$

$$\dot{v}_2(z_1) = z_1\left(\dot{x_{1d}} - x_2\right) \tag{12}$$

The stabilization z_1 is achieved by including a virtual control input x_2.

$$x_2 = \dot{x_{1d}} + \alpha_1 z_1 \ (\alpha_1 > 0) \tag{13}$$

The second step is to choose the sliding surface. A sliding manifold is designed as below:

$$s_2 = x_2 - \dot{x_{1d}} - \alpha_1 z_1 \tag{14}$$

Likewise, the Lyapunov function is examined for both z_1 and s_2.

$$V(z_1, s_2) = \frac{1}{2}\left(z_1^2 + s_2^2\right) \tag{15}$$

$$\dot{V} = z_1\dot{z_1} + s_2\dot{s_2}$$

$$\dot{V} = z_1(-\alpha_1 z_1 + s_2) + s_2(-k_1 \text{sign}(s_2) - k_2 s_2 - z_1)$$

$$\dot{V} = -\alpha_1 z_1^2 - k_1|s_2| - k_2 s_2^2 \leq 0 \tag{16}$$

The derivative of the Lyapunov function is negative semi-definite. Therefore, the closed-loop system is stable in the Lyapunov sense, and both the tracking error and sliding variable converge to zero.

The convergence condition of the sliding manifold is given by $\dot{S}S \leq -\eta|S|$, $\eta > 0$, where S is the sliding variable that represents the deviation of the system state from the sliding manifold.

The sliding dynamics for the roll channel are defined as

$$\begin{aligned} \dot{s_2} &= -k_1 sign(s_2) - k_2 s_2 \\ &= \dot{x_2} - \ddot{x_{1d}} - \alpha_1 \dot{z_1} \\ &= a_1 x_4 x_6 + a_2 x_4 \Omega_r + b_1 U_2 - \ddot{x_{1d}} + \alpha_1 (z_2 + \alpha_1 z_1) \end{aligned} \tag{17}$$

where k_1 and k_2 are non-negative adaptability constants, and "sign" refers to the signum function. Consequently, the control law for U_2 is achieved as

$$U_2 = \frac{1}{b_1}\left(-a_1 x_4 x_6 - a_2 x_4 \Omega - \alpha_1^2 z_1 - k_1 sign(s_2) - k_2 s_2\right) \tag{18}$$

The corresponding control rules for U_3, U_4, and U_1 are derived using same reasons as stated above.

$$\begin{aligned} U_3 &= \tfrac{1}{b_2}\left(-a_3 x_2 x_6 + a_4 x_2 \Omega - \alpha_2^2 z_3 - k_3 sign(s_3) - k_4 s_3\right) \\ U_4 &= \tfrac{1}{b_3}\left(-a_5 x_2 x_4 - \alpha_3^2 z_5 - k_5 sign(s_4) - k_6 s_4\right) \\ U_1 &= \tfrac{m}{\cos\Phi\cos\theta}\left((-g) - \alpha_7^2 z_7 - k_7 sign(s_z) - k_8 s_z\right) \end{aligned} \tag{19}$$

$$\begin{aligned} z_3 &= x_{3d} - x_3 \\ s_3 &= x_4 - \dot{x_{3d}} - \alpha_2 z_3 \\ z_5 &= x_{5d} - x_5 \\ s_4 &= x_6 - \dot{x_{5d}} - \alpha_3 z_5 \\ z_7 &= x_{7d} - x_7 \\ S_7 &= x_8 - \dot{x_{7d}} - \alpha_7 z_7 \end{aligned} \tag{20}$$

where S_2, S_3, S_4 and S_z illustrate the dynamics of roll, pitch, yaw, and altitude, respectively, and $\alpha_1, \alpha_2, \alpha_3$ and α_7 represent the specifications of the SMC. The parameters utilized for the SMC are presented in Table 3.

Table 3. SMC parameters.

Parameter	Φ (Roll)	θ (Pitch)	ψ (Yaw)	z (Altitude)
α	10	10	10	10
K_1	20	25	20	15
K_2	20	25	20	15

The overall control structure and the developed sliding mode controller are illustrated in Figures 8 and 9, respectively.

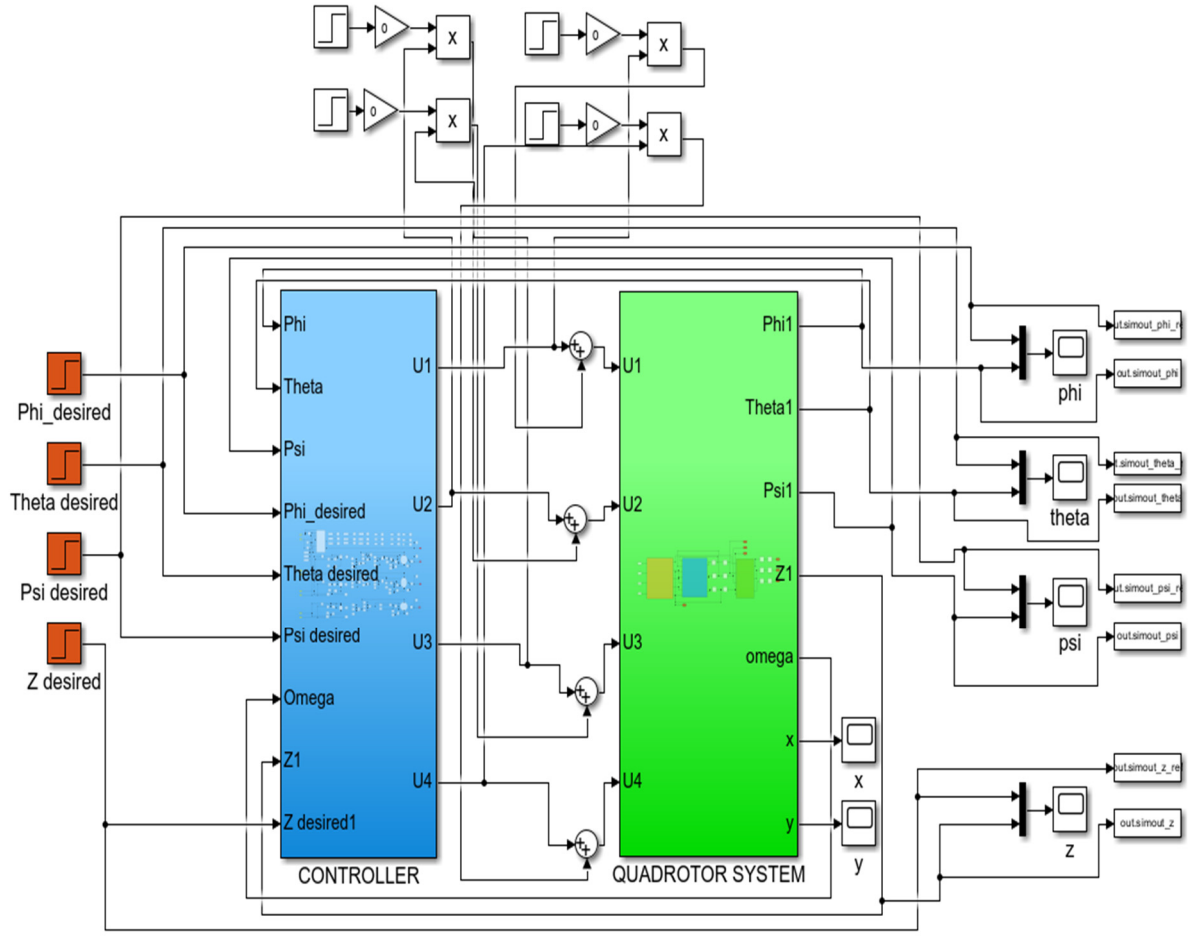


Figure 8. Block diagram of quadrotor system control.

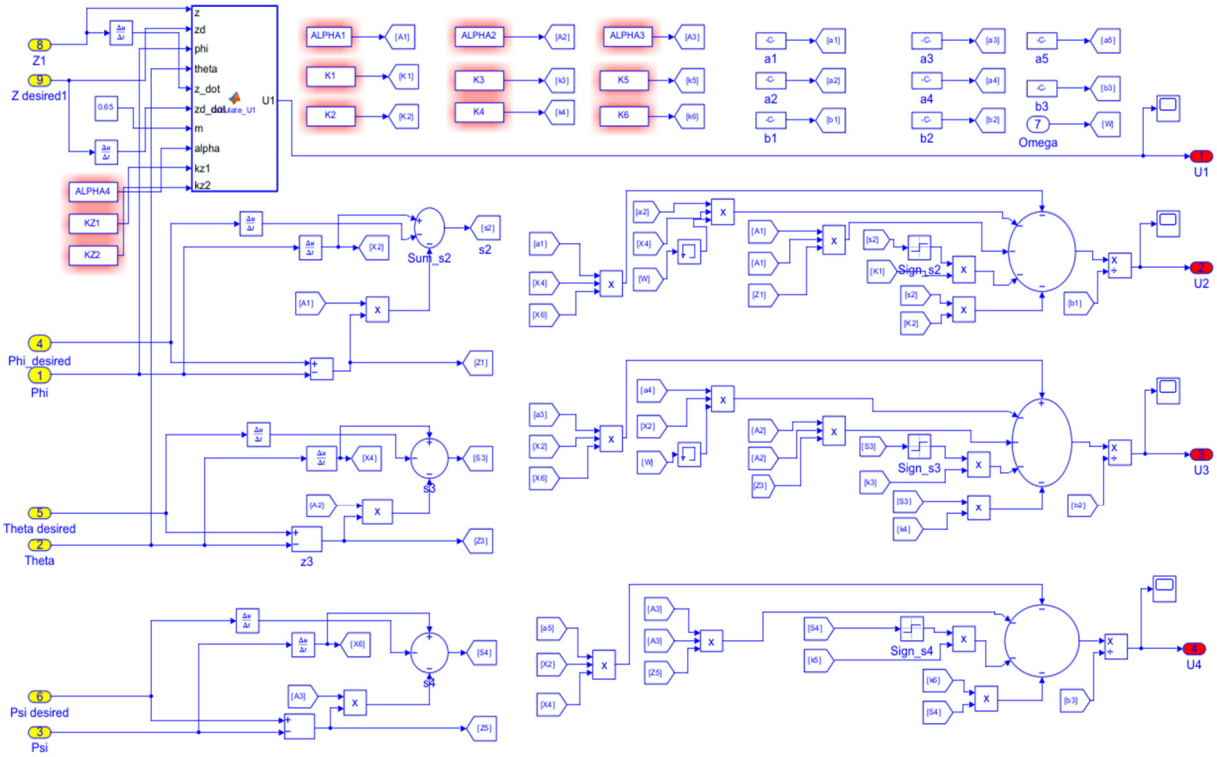


Figure 9. Block schematic of the developed sliding mode control.

3.2. Fuzzy Logic Controller (FLC) Design and Implementation

The Fuzzy Logic Controller (FLC) is based on strict mathematical equations, with the expertise knowledge being captured by a collection of the so-called if–then rules. The controller, in operation, transforms linguistic descriptions of the error, e.g., negative large, zero, positive medium, etc., and change rate of the error into an appropriate control action. This systems-based methodology is able to accommodate nonlinearity and uncertainty in the system without the need to have an actual analytical model [34].

Input Selection and Linguistic Variable Definition: The controller is actuated by two main inputs: (1) Error (e) which is a difference between the desired and observed values of the controlled variables, including the attitude (roll, pitch, yaw) of the quadcopter and its altitude; (2) Error rate (e/dt) which is the discrete time derivative of the error that measures how fast the attitude or altitude is going out of control.

Triangular membership functions (MFs) are used to divide the admissible range into overlapping linguistic labels of each input. The error variable is used in 7 MF sets, i.e., NL (negative large), NM (negative medium), NS (negative small), Z (zero), PS (positive small), PM (positive medium), PL (positive large), which makes a total of 7 unique linguistic terms. In the same way, there are seven MF sets which are defined on the error-rate variable including NL (extremely negative), NM (moderately negative), NS (extremely small), Z (neutral), PS (extremely positive), PM (moderately positive), and PL (extremely large).

These membership functions allow the controller not only to assess the size and direction of the deviation with respect to the setpoint, but also to assess whether the deviation is increasing or decreasing and hence respond more responsively to dynamic changes in system behavior.

Fuzzy Logic Control includes three major phases. This process starts with the fuzzification stage, in which the representative system as the error signal and its rate of change are transformed into the fuzzy sets using linguistic variables. An inference step then makes use of a knowledge base comprising fuzzy if–then rules that relate the input linguistic words to the corresponding controller settings. This step relies on a knowledge base of membership functions and an inference mechanism, which considers the rules of relevance to infer the control action and is hence a replica of human decision-making. The last stage is defuzzification, where the fuzzy signal is converted to an accurate control signal that can be applied in practice. The Mamdani fuzzy inference model was selected following the structure used in the reference adopted for the quadrotor fuzzy controller design, as it is widely used in UAV control due to its simplicity and robustness in nonlinear systems. Triangular membership functions were chosen because of their low computational complexity and ease of implementation in MATLAB/Simulink. Seven membership functions were used for each input to achieve a balance between control accuracy and rule-based complexity, which is a common choice in fuzzy control design. The universe of discourse was determined based on the operating ranges obtained from preliminary simulations and according to the adopted reference to ensure that all expected system states are covered without saturation.

The diagram below (Figure 10) shows the architecture of a standard Fuzzy Logic system [35].

The rule base used is the Mamdani type, but it is specifically created to be used in a multiple-input single-output (MISO) setup. The membership functions defining the fuzzy set of error input are triangular functions over the interval $[-30, 30]$, and the error rate input is also defined by triangular functions over a larger range $[-200, 200]$ (see Figures 11 and 12). Outputs that control the attitude and altitude of the quadcopter fall within the range $[-2.8, 2.8]$; as such, the output variable is described by seven triangular membership functions as illustrated in Figure 13. Mamdani-type Fuzzy Controllers make use of a complete repertoire of if–then rules to map the input fuzzy sets to output control responses,

thus enabling useful alleviation of nonlinearities and uncertainties of quadcopter control systems [36]. Rule-Based Construction: The amount of rules is 49 in total, covering all of the possible combinations of the seven categories of errors and the seven categories of error-rate as shown in Table 4.

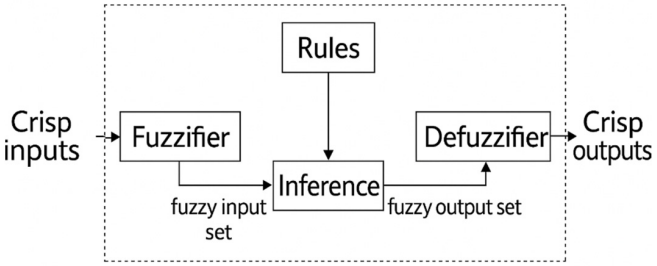


Figure 10. Fuzzy Control system.

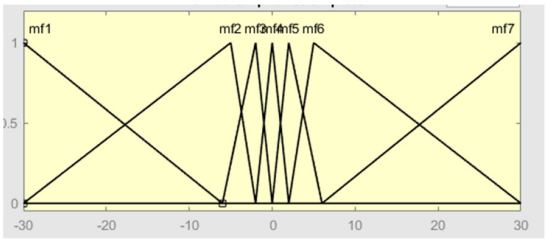


Figure 11. Error input membership function.

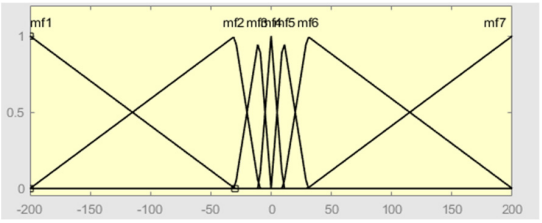


Figure 12. Error rate input membership function.

Table 4. Fuzzy rules.

Δe\e	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

NB (Neg. Big), NM (Neg. Med), NS (Neg. Small), Z (Zero), PS (Pos. Small), PM (Pos. Med), PB (Pos. Big).

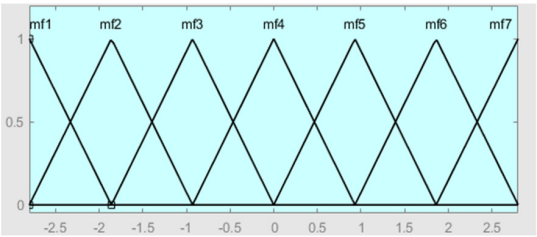


Figure 13. Output membership function.

The rules are represented as follows: IF Error = X and Error-Rate = Y, THEN Output = Z. These seven discrete output values are outlined in the range between the maximum negative and maximum positive corrections between [−2.8, 2.8] in terms of control signal. Example rules include:

IF Error = NB and Error-Rate = NB, then the Output = −2.8 (negative big correction).

IF Error = Z and Error -Rate = Z, THEN output = 0 (hold steady state).

Where IF Error = PB and Error-rate = PB, then Output = 2.8 (positive big correction).

The ensuing control input can be employed to adjust the quadrotor's roll, pitch, and yaw angle [35].

$$\begin{pmatrix} U_2 = K_p \cdot (\phi_d - \phi) + K_d \cdot \left(\dot{\phi}_d - \dot{\phi}\right) \\ U_3 = K_p \cdot (\theta_d - \theta) + K_d \cdot \left(\dot{\theta}_d - \dot{\theta}\right) \\ U_4 = K_p \cdot (\psi_d - \psi) + K_d \cdot \left(\dot{\psi}_d - \dot{\psi}\right) \end{pmatrix} \tag{21}$$

U_2: Roll control Input ϕ_d : Roll Desired ϕ: Actual Roll.
U_3: Pitch control Input θ_d : Pitch Desired θ: Actual Pitch.
U_4: Yaw control Input ψ_d : Yaw Desired ψ: Actual Yaw.
K_p: Proportional Gain K_d: Derivative Gain.

The implementation of the altitude or Z controller diverges from that of roll, pitch, and yaw due to the necessity of canceling the nonlinearities included in the Z dynamics [37]. The control input of the Z controller can be delineated as

$$U_1 = \frac{g + K_{zp} \cdot (z_d - z) + K_{zd} \cdot \left(\dot{z_d} - \dot{z}\right)}{\cos(\theta) \cdot \cos(\phi)} \tag{22}$$

U_1: Altitude control Input; z_d : Altitude Desired; z: Altitude Actual Yaw.

The quadrotor control system used is the same as in Figure 8, but Figure 14 was used instead of Figure 9.

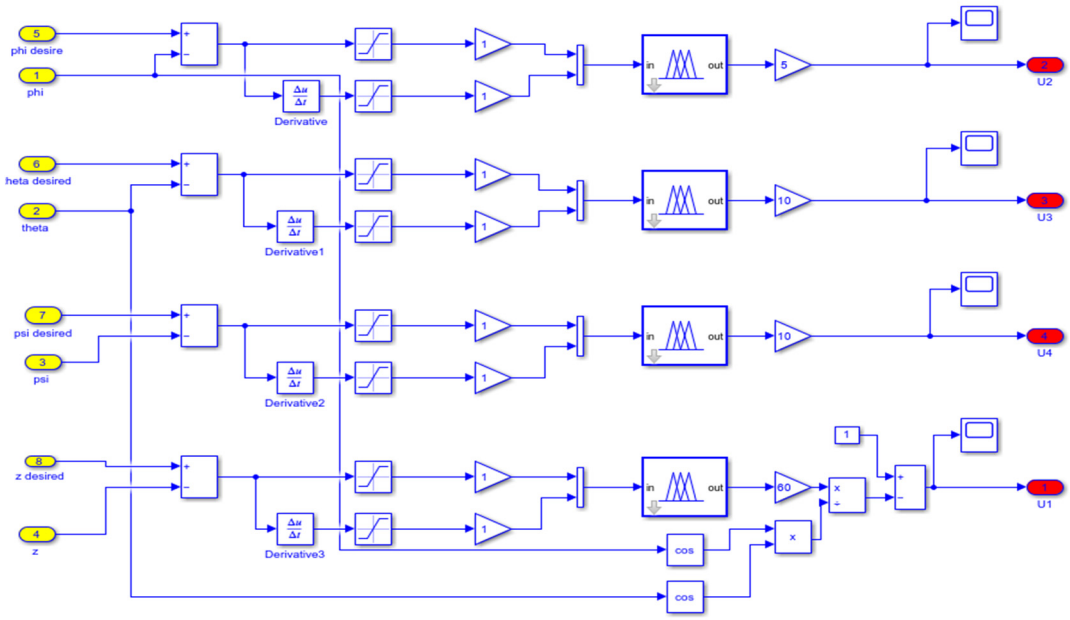


Figure 14. Block schematic of the Fuzzy Logic Controller.

3.3. Sliding Mode Controller (SMC)–Gray Wolf Optimizer (GWO) Design and Implementation

Various hybrid methods are integrated into the Sliding Mode Controller, especially the combination of Sliding Mode Control (SMC) and intelligent control. These integrations

have proved to be more flexible and precise in unmanned aerial vehicle (UAV) control [38]. Still, the traditional SMC approaches continue to have issues with the optimization of parameters and complexity. This is why smart optimization methods, especially the Gray Wolf Optimizer (GWO), are used to automatically regulate SMC parameters, hence enhancing the stability of the systems and the performance of the control systems in the face of uncertainties. Gray Wolf Optimization (GWO) is a fairly new meta-heuristic method which has been presented in [21] and models the social hierarchy and predatory behavior of gray wolves. The algorithm uses simulation of a process of hunting in order to find and achieve the best solutions. The hunting process can be described as comprising three steps: (A) prey monitoring, (B) surrounding and forcing the target to stop, and (C) attacking the target. The algorithm is modeled after the social hierarchy of wolves and will assign people to four positions: Alpha (α), Beta (β), Delta (δ), and Omega (ω). During the design stage, the wolf hierarchy is modeled, where Alpha is the best candidate solution, while Beta and Delta are the second and third most promising solutions. The rest of the solutions are very minute and are represented by Omega [39].

A. Prey acquisition: The gray wolves find their prey by choosing individuals at points of α, β, and δ. After that, they separate and meet back again to hunt the prey.

$$|A| > 1 \tag{23}$$

B. Surrounding the Prey: The surrounding process is mathematically described over the following equations.

$$\vec{D} = j\vec{C}\vec{X}_p(t) - \vec{X}(t) \tag{24}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{25}$$

where

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r_1} - \vec{a} \tag{26}$$

$$\vec{C} = 2 \cdot \vec{r_2} \tag{27}$$

With t = Current iteration $\vec{A}$; $\vec{C}$ = Coefficient Vectors.

$\vec{X_p}$ = Position Vector of the prey; $\vec{X}$ = Position vector of wolf.

The parameter 'a' is reduced from 2 to 0 to highlight exploration and exploitation, respectively. The location of a gray wolf adjusts in relation to the position of its prey. This technique attains the optimal solution (prey) with the previously identified three best solutions (α, β, δ). The subsequent equations are employed to update their locations in the next iteration.

$$\begin{pmatrix} \vec{D_\alpha} = \left| \vec{C_1} \cdot \vec{X_\alpha} - \vec{X} \right| \\ \vec{D_\beta} = \left| \vec{C_2} \cdot \vec{X_\beta} - \vec{X} \right| \\ \vec{D_\delta} = \left| \vec{C_3} \cdot \vec{X_\delta} - \vec{X} \right| \end{pmatrix} \tag{28}$$

$$\begin{pmatrix} \vec{X_1} = \vec{X_\alpha} - \vec{A_1} \cdot \left(\vec{D_\alpha}\right) \\ \vec{X_2} = \vec{X_\beta} - \vec{A_2} \cdot \left(\vec{D_\beta}\right) \\ \vec{X_3} = \vec{X_\delta} - \vec{A_3} \cdot \left(\vec{D_\delta}\right) \end{pmatrix} \tag{29}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{30}$$

C. Attacking the Prey: The wolves attack the prey, and the location of the prey is the ultimate locus of the Alpha.

$$|\,A\,| < 1 \tag{31}$$

The Gray Wolf Optimizer (GWO) enables its search agents to revise their positions with respect to the positions of the top three agents (alpha, beta, gamma) in the process of seeking the optimum after every iteration.

Before the actual process of carrying out the major goal of any meta-heuristic population-based algorithm, two basic parameters should be set up. The main configuration parameter is the total number of search agents. These agents are termed as gray wolves in the GWO algorithm. The number of search agents can vary, depending on the situation of the problem; in the current research, this parameter will be 100. The number of iterations is the second important parameter that depends on the type of application and changes significantly. With fewer iterations, the evaluation time will be reduced. Maximum number of iterations will decide if the algorithm stops at this stage, depending on whether optimal solution is achieved or not. The objective (or fitness) function is defined by first taking into account the desired requirements and constraints. A proper objective function is chosen so as to optimally control the control parameters by considering the overall closed-loop response. There are several domain functions that can be used as objective functions to different systems, which may be broadly categorized into two: (a) functions based on certain attributes of the response; (b) functions based on the overall or holistic response. The reason why integral criteria is usually selected is due to their performance. One of the benefits of using an integral function is that it is easily extended to multiloop systems. The objective function used in this work is defined using the Integral of Squared Error (ISE) of the quadrotor tracking states, given by

$$J = \Sigma \int_0^T e_i^2(t)\, dt \tag{32}$$

where e_i represents the tracking error of roll, pitch, yaw, and altitude, and t is the simulation time. In this work, the optimization problem is formulated as a single-objective cost function based on the Integral of Squared Error (ISE) of the quadrotor states. The total cost is defined as the sum of the ISE (Best Fitness) values of roll, pitch, yaw, and altitude with equal weights.

The Sliding Mode Controller (SMC) parameters were established following the application of the GWO algorithm through a code developed to have fixed parameters of 200–300 iterations and 100 wolves. In this work, the classical Sliding Mode Control law is intentionally retained without modification in order to preserve its simplicity and robustness. The Gray Wolf Optimizer is used only for optimal tuning of the controller parameters, which improves the overall performance and reduces oscillatory behavior in response. The output of the application of the algorithm in both cases where there is no disturbance and disturbance is as follows. Table 5 presents the parameters of the controller in the disturbance-free scenario, whereas Figure 15 demonstrates how the values of the best

fitness change as the number of iterations grows. Table 6 shows the controller parameters obtained in the case of disturbance and Figure 16 shows how the best fitness values change with the number of iterations. The key stages of the Gray Wolf Optimizer (GWO) while seeking the optimum solution are shown in Figure 17.

Table 5. Optimized SMC parameters for quadrotor attitude and altitude control without disturbance.

Parameter	ϕ (Roll)	θ (Pitch)	ψ (Yaw)	z (Altitude)
α	76.7107	71.4167	71.6452	163.0688
K_1	17.6811	3.2796	45.0034	3.7921
K_2	153.3674	159.7310	142.4560	85.8731

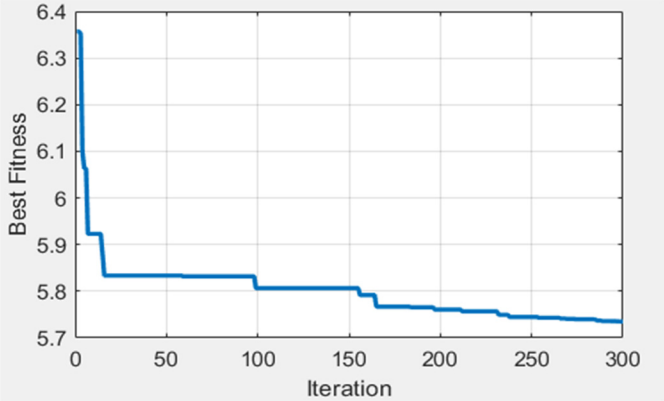


Figure 15. The change in best fitness values, with increasing no. of iterations for the GWO-SMC without disturbance. (Best Fitness = 5.7349).

Table 6. Optimized SMC parameters for quadrotor attitude and altitude control with disturbance.

Parameter	ϕ (Roll)	θ (Pitch)	ψ (Yaw)	z (Altitude)
α	55.9935	55.2580	91.4953	152.7136
K_1	32.1087	13.7840	0.0000	15.1749
K_2	112.0095	104.2681	183.5301	80.6698

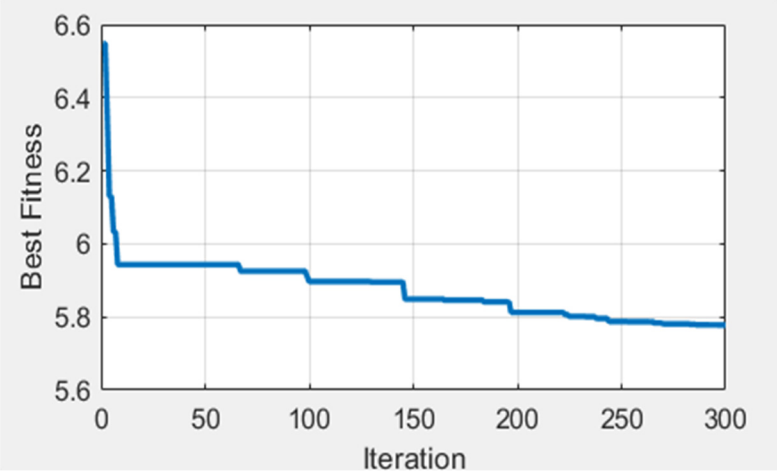


Figure 16. The change in best fitness values with increasing no. of iterations for the GWO-SMC with disturbance. (Best Fitness = 5.7777).

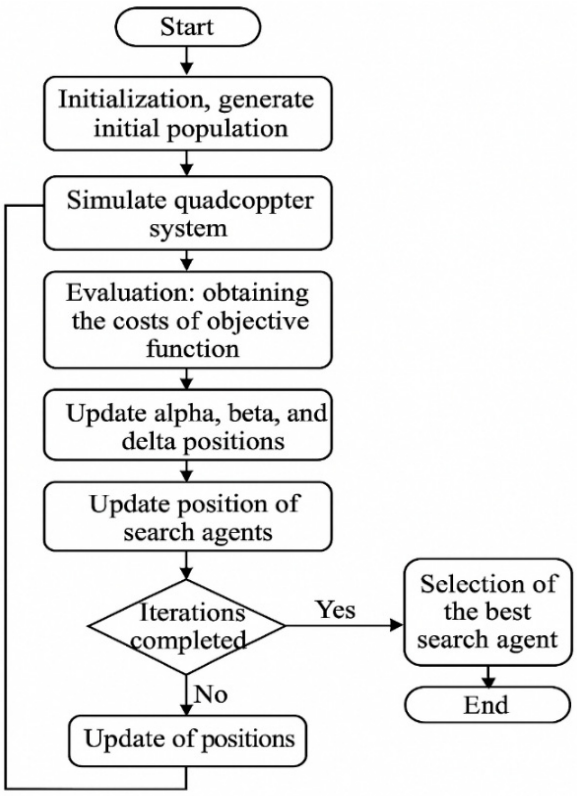


Figure 17. Flow chart of the GWO.

3.4. Sliding Mode Controller (SMC)–Genetic Algorithm (GA) Design and Implementation

To validate the generalizability and reliability of the optimization strategy proposed, an additional validation step, herein referred to as the Sliding Mode Controller (SMC) based on Genetic Algorithm (SMC-GA) was implemented. The meta-heuristic algorithm was realized in this step, which was applied in the same simulation environment in which the SMC-GWO controller was used to guarantee the consistency of the optimization process. The Genetic Algorithm (GA) is aimed at finding the best parameters of the Sliding Mode Controller (SMC) that controls the quadcopter. The optimization is realized through the process of natural selection and spreading of genetic information between individuals. The fitness function is a measure of the performance of the system as compared to the existing values of SMC parameters and is used to differentiate effective and ineffective alternatives [40,41].

A. Initialization and Creation of the Starting Population: The genetic individual is represented as a set of parameter values representing its attributes. The loci is the number of variables to be optimized, and a locus of the model can take the value in a fixed interval. These intervals have been determined after the original calibration of the system based on the Integral of Squared Error (ISE) criterion. Other important parameters of genetic algorithms include the generations, evaluation strategy, selection mechanism, crossover or mating protocol, and mutation rate. The size of the population depends on the complexity of the problem so that the size must be chosen to provide a balance between the search precision and the efficiency of its computation to provide optimal search performance.
B. Evaluation: Each member of the population is evaluated based on a fitness function, which is defined as an average ISE added by each parameter of optimization, which includes roll, pitch, yaw, and altitude.
C. Selection: At the selection stage, the chromosomes, which perform better based on the fitness function, are selected and kept to survive and reproduce. The process of

natural selection is repeated at every step of algorithm implementation in order to guarantee further evolution process across generation. Individuals with the most suitable values of fitness are only reproduced, and unfit chromosomes are scrapped to give way to new ones. This paper uses tournament selection as the selection method.

D. Reproduction or Crossover: The crossover or mating process is the generation of new offspring out of the parents chosen by recombining their genetic materials to increase their population efficacy. There is a single-point crossover method which is used to produce the second generation of people.

E. Mutation: The concept of mutation is the change in the values of genes, and the example is the transformation of a 1 to a 0 and the reverse. The genes that will be changed will be randomly selected.

The objective function used in optimizing the SMC parameters of the quadcopter is the average ISE of each controller, which is optimized using twelve parameters related to it. This experiment used the GA, with a crossover rate of 0.8 and mutation rate of 0.2 being used for the tournament selection method and single-point crossover method. The parameters were also obtained by trial and error using the various trials to determine various effects of population size, search domain, number of generations, and crossover rate on the overall system performance. The SMC parameters were optimized after applying the GA with fixed-bound 200–300 generations and a population of n = 100 individuals. Algorithm results are displayed both in the case of disturbance free and disturbed. Table 7 contains the computed controller parameters in the disturbance-free scenario, whereas Figure 18 shows the change in the best fitness values with the generations. The controller parameters obtained in the disturbed case are given in Table 8, and Figure 19 shows the fitness curve versus generation. The critical stages of the GA as it seeks the optimal solution are summarized in Figure 20.

Table 7. Optimized SMC parameters for quadrotor attitude and altitude control without disturbance.

Parameter	ϕ (Roll)	θ (Pitch)	ψ (Yaw)	z (Altitude)
α	65.3268	71.6298	81.8742	154.9266
K_1	49.5338	15.8283	22.9202	0.1810
K_2	136.7562	142.5595	157.5123	89.1712

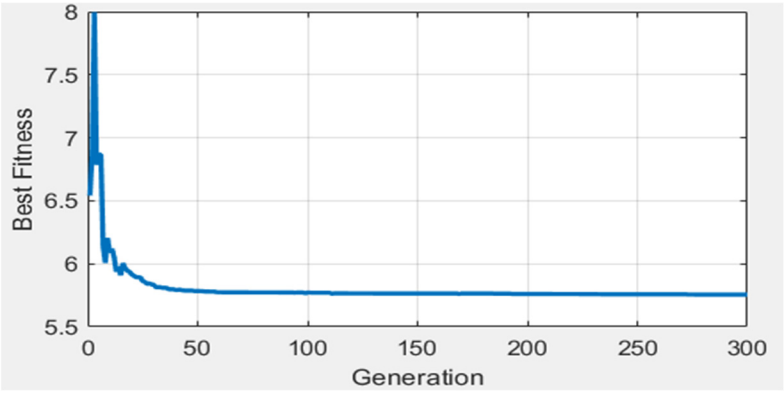


Figure 18. The change in best fitness values with increasing no. of generations for the GA-SMC without disturbance. (Best Fitness = 5.7556).

Table 8. Optimized SMC parameters for quadrotor attitude and altitude control with disturbance.

Parameter	ϕ (Roll)	θ (Pitch)	ψ (Yaw)	z (Altitude)
α	57.3973	54.1478	91.5439	146.7855
K_1	26.7905	0.8908	6.3724	4.9507
K_2	115.3443	109.0496	182.7499	83.6192

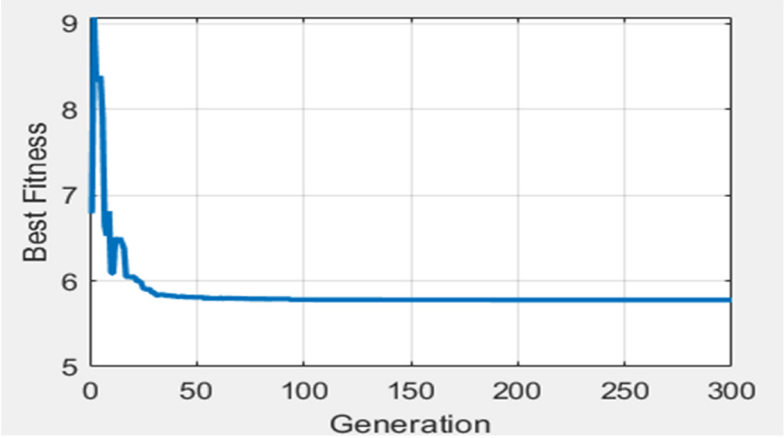


Figure 19. The change in best fitness values with increasing no. of generations for the GA-SMC without disturbance. (Best Fitness = 5.7786).

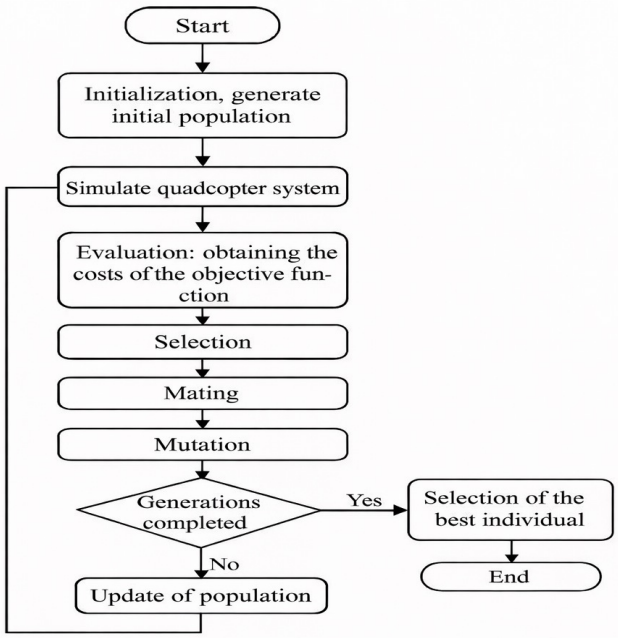


Figure 20. Flow chart of the GA [41].

4. Results and Discussion

Several simulations were performed in MATLAB/Simulink. For numerical implementation in Simulink, a fixed-step solver was used, where the step size represents the sampling time of the controller. The sampling frequency was selected according to the fastest closed-loop dynamics to ensure accurate representation of the system behavior. In practice, the sampling period must be chosen at a much smaller value than the minimum rise time of the system to guarantee stability and proper digital implementation of the proposed control scheme.

The performance indices were compared using the main performance indices, including rise time (T r), overshoot (OS), and settling time (T s), as well as steady-state error (SSE) on the main quadrotor channels of roll, pitch, yaw, and altitude. Figure 21 shows the reaction of altitude and attitude (roll, pitch, and yaw) in the SMC, FLC, SMC-GWO, and SMC-GA controllers, respectively, under nominal (i.e., non-disturbed) conditions, whereas Figure 22 shows the reaction of the same under perturbed conditions. The objective trajectory of the prescribed objective was used to assess and illustrate system responses of the quadrotor states, i.e., the roll (φ) angle, pitch (θ) angle, yaw (ψ) angle, and the altitude (z). The nominal control input was set at t = 1 s. Without any external perturbations, the reference signals would have been defined with the transition between 0 and 1, whereas the control input would gradually change between 0 and its nominal value at t = 1 s. Then, the external disturbances were added to show the strength and adaptability of the four control strategies (SMC, FLC, SMC-GWO, and SMC-GA), and two separate disturbance components were determined.

$$u = \hat{u} + \delta u \tag{33}$$

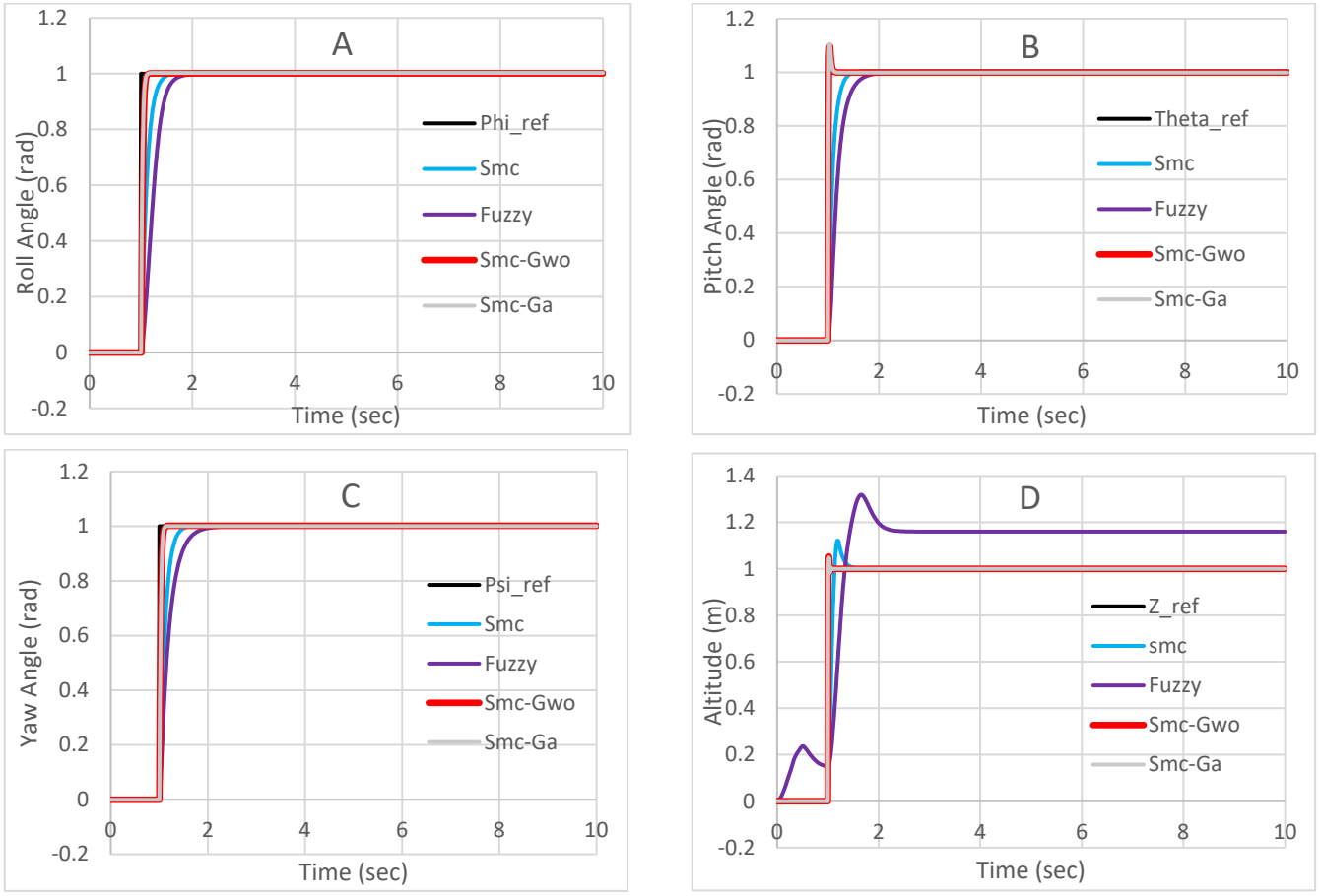


Figure 21. Desired and obtained trajectory simulation using different controllers without disturbance: (**A**) ϕ (roll), (**B**) θ (pitch), (**C**) ψ (yaw), and (**D**) z (altitude).

The normalized nominal control variable which is represented by (û) is obtained relating to the nominal model relating to a given trajectory. A model is made to have an auxiliary control term, (δu), to balance external disturbances and model uncertainties. The quadrotor was perturbed uniformly in all angular velocities—roll, pitch, yaw, and altitude. A constant disturbance of the same value 10% of the nominal control input was imposed at time t = 1 s; that is, δu = 0.1. The attitude responses, roll, pitch, and the yaw are also given in radians, and the altitude response is given in meters with the time axis in seconds.

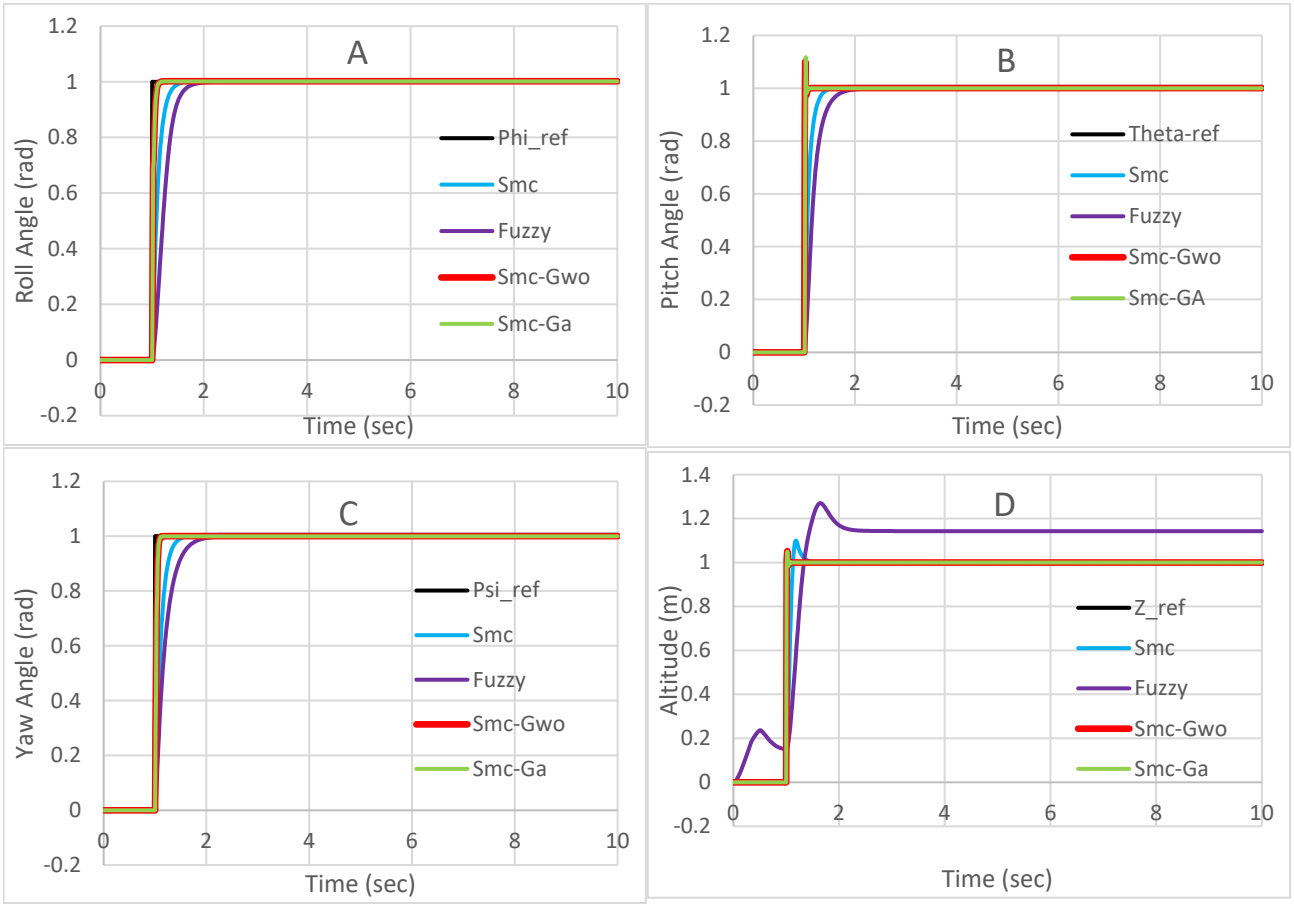


Figure 22. Desired and obtained trajectory simulation using different controllers with disturbance: (**A**) ϕ (roll), (**B**) θ (pitch), (**C**) ψ (yaw), and (**D**) z (altitude).

Table 9A indicates that the SMC-GWO controller has a significant enhancement in roll dynamics. The controller decreases the rise and settling times by 78.46 and 23.40 percent, respectively, compared to the traditional SMC, and the steady-state error (SSE = 0.0008) is almost zero. Relative to the Fuzzy Logic Controller (FLC), the proposed controller attains an 86.57 percent acceleration in rise time, and a 34.71 percent decrease in settling time, which represents significantly high transient performance. When used as a roll controller, a comparison between SMC-GWO and SMC-GA reveals that the GWO-tuned controller has a slightly faster rise time (0.0513 s versus 0.0531 s, a 3.39 percent improvement) and a slightly shorter settling time (1.0853 s versus 1.0890 s, a 0.34 percent improvement). More importantly, the overshoot decreases to 0.08 with GWO compared to 0.31 with GA, which is an improvement by approximately 74.19 percent, which means that the transient response is smoother. Furthermore, the steady-state error is significantly smaller with GWO (0.0008 vs. 0.0031, which is improved by 74.19 percent), which proves greater control accuracy. In general, the application of SMC-GWO is more effective than SMC-GA at roll stabilization with less time delay, better smoothness, and accuracy, which highlights the power of GWO-based optimization to Sliding Mode Control.

As Table 9B indicates, the SMC-GWO controller is able to considerably improve the dynamics of the pitch of the quadrotor. It decreases the rise time by 93.65 percent and the settling time by 20.22 percent compared to the classical SMC, but the steady-state error is very small (SSE = 0.0001). Comparing it to the FLC, the enhancement becomes even more significant, with a 96.98 percent faster rise time and a 39.22 percent shorter settling time, thus showing the high-quality transient performance of the proposed controller. When comparing SMC-GWO and SMC-GA to the pitch control, the GWO-tuned controller has

a small difference in the rise time (0.0128 s vs. 0.0125 s, 2.40% slower) but settles faster (1.0797 s vs. 1.0834 s, 0.34 percent better). More to the point, the overshoot of 10.61% in the case of GA is lowered by a substantial margin to 9.64 in the case of GWO, an improvement of about 9.13 percent, which offers a less noisy transient response. There is also a much lower steady-state error with GWO (0.0001 compared to 0.0007, an 85.71 percent improvement), which means a much more accurate control. In general, SMC-GWO offers improved smooth, accurate, and stable regulation of pitch with significantly smaller overshoot and much better steady-state accuracy, despite GA giving a small but significant increase in initial rise.

Table 9. Performance comparison of non-disturbed controllers.

Variable	Controller	RT (s)	OS (%)	ST (s)	SSE
[A] Roll Angle (ϕ)	SMC	0.2382	0.47	1.4179	0.0047
	FLC	0.3815	0.00	1.6625	0.0000
	SMC-GWO	**0.0513**	**0.08**	**1.0853**	**0.0008**
	SMC-GA	0.0531	0.31	1.0890	0.0031
[B] Pitch Angle (θ)	SMC	0.2015	0.51	1.3531	0.0051
	FLC	0.3543	0.00	1.6856	0.0000
	SMC-GWO	**0.0128**	**9.64**	**1.0797**	**0.0001**
	SMC-GA	0.0125	10.61	1.0834	0.0007
[C] Yaw Angle (ψ)	SMC	0.2389	0.48	1.4136	0.0048
	FLC	0.4241	0.05	1.7762	0.0000
	SMC-GWO	**0.0527**	**0.22**	**1.0867**	**0.0022**
	SMC-GA	0.0505	0.10	1.0837	0.0010
[D] Altitude (z)	SMC	0.0959	12.20	1.3726	0.0012
	FLC	0.6000	30.00	2.1036	0.1602
	SMC-GWO	**0.0078**	**5.32**	**1.0287**	**0.0000**
	SMC-GA	0.0079	4.87	1.0314	0.0000

Table 9C shows that the SMC-GWO controller is characterized by significant improvement of the yaw response. It reduced the rise time by 77.94 percent and the settling time by 23.16 percent compared to the more traditional SMC, which establishes that the transient behavior is distinctly quicker. The difference when contrasted with that of the FLC is also significant: the rise time is reduced by 87.57, and the settling time is reduced by 38.82, thus showing the better dynamic performance of the GWO-based controller on the highly coupled yaw dynamics. The comparison between SMC-GWO and SMC-GA in terms of yaw control shows that the GA-tuned controller has a slightly higher transient performance. Despite a slightly slower rise time (0.0527 s vs. 0.0505 s, 4.35 percent slower), the difference is small. The settling time of both controllers is almost the same, with the GA being slightly quicker (1.0837 s compared to 1.0867 s, 0.28 percent-differentiated). More to the point, the SMC-GA controller has a lower overshoot (0.10% verses 0.22%), a 54.5 percent reduction, and it also has better steady-state accuracy, with SSE changing to 0.0010 (GA) versus 0.0022 (GWO), a reduction of 54.5 percent. On the whole, both the controllers improve the yaw dynamics compared to the traditional approaches, although SMC-GA attains better overshoot, steady-state accuracy, and marginally lower transient speed compared to the SMC-GWO, which is still a competitive response with less accuracy in yaw regulation.

Table 9D indicates that the altitude performance improves significantly through the optimized controllers. Compared to classical SMC, the proposed SMC-GWO slows the rise time by 91.87 percent and the settling time by 25.04 percent, whereas the steady-state error (SSE = 0) is completely removed. The improvement is even greater when compared to the FLC: the rise time is lowered by 98.70, the settling time is lowered by 51.10, and the steady-state error is lowered from 0.1602 to 0, which represents an 82.3 percent improvement in accuracy. An altitude control comparison between SMC-GWO and SMC-GA reveals that the GWO-tuned controller has a marginally faster rise time (0.0078 s versus 0.0079 s, 1.27 percent faster) and slightly lower settling time (1.0287 s versus 1.0314 s, 0.26 percent shorter). The difference between the overshoots is low (5.32% and 4.87%), and both controllers have zero steady-state error. In general, it can be concluded that SMC-GWO provides slightly better transient response and the same high accuracy as SMC-GA, proving that GWO tuning is effective in improving the altitude regulation without causing an extra steady-state error.

Table 10A indicated that the SMC-GWO controller had better roll stability than the other controllers in the event that a disturbance was introduced at t = 1 s. It reached a rise time of 0.0558 s, which is a 76.31 percent decrement as compared to the classical SMC (0.2355 s) and 85.46 percent lower than the FLC (0.3840 s). The maximum overshoot was 0.20% and the system stabilized with a settling time of 1.0975 s, which is a 22.55 percent faster response than SMC (1.4153 s) and 34.71 percent faster than FLC (1.6807 s). Also, SMC-GWO was the lowest steady-state error contestant, with SSE 0.0002. The roll control through SMC-GWO and SMC-GA in the case of disturbance conditions indicates that the GWO-tuned controller has a slightly faster rise time (0.0558 s vs. 0.0565 s, 1.24 percent better) and a smaller steady-state error (0.0002 vs. 0.0017, 88.24 percent smaller). Conversely, the values of the overshoot are very similar (0.20% vs. 0.17%), and the settling times are similar (1.0975 s vs. 1.0983 s, 0.07% difference). In general, SMC-GWO provides faster transient response and much better accuracy than SMC-GA, which shows its strength and effectiveness in terms of disturbance rejection.

Table 10B indicates that the SMC-GWO controller produced much better pitch dynamics than the conventional controllers after application of a disturbance at t = 1 s. It recorded a rise time of 0.0113 s, which represents 94.31 percent of the classical SMC (0.1985 s) and a 96.88 percent improvement compared to the FLC (0.3625 s). This increase in speed was, however, accompanied by an overshoot of 10.12%, which is more than SMC (0.57) and FLC (0%), but still within acceptable transient ranges. The SMC-GWO performed well in settling behavior with a settling time of 1.0469 s, 22.91 percent less than SMC (1.3582 s) and 38.33 percent less than FLC (1.6970 s). In disturbance cases, SMC-GWO and SMC-GA will have slightly different rise times, with GA-tuned controller having a slightly lower rise time (0.0105 s vs. 0.0113 s, 7.08 percent faster). Nevertheless, SMC-GWO has a smaller overshoot (10.12% vs. 11.40%, 11.23 percent reduction) and shorter yet highly close settling time (1.0469 s vs. 1.0296 s), whereas SMC and FLC have a smaller stabilizing error (0.0001), which is smaller than that of SMC-GWO (0.0009). In general, SMC-GWO guarantees a smoother transient response with smaller overshoot.

As illustrated in Table 10C, the SMC-GWO controller gave the fastest response to disturbance at t = 1 s as compared to the other controllers. It decreased the rise time to 0.0469 s, which is an 80.25 percent improvement over the classical SMC (0.2375 s) and an 88.99 percent improvement as compared to the FLC (0.4258 s). The SMC-GWO realized a zero overshoot, which is superior to the SMC (0.53%) and the FLC (0.01%) controllers and exhibited high levels of transient smoothness when disturbed. The SMC-GWO, in its turn, settled within 1.0816 s, which is 23.70 percent lower than that of SMC (1.4173 s) and 39.17 percent lower than that of FLC (1.7776 s). The difference between SMC-GWO and

SMC-GA is not that significant. The rise time of SMC-GWO is slightly shorter (0.0469 s vs. 0.0472 s, 0.63 percent difference), and the settling time is slightly less (1.0816 s vs. 1.0818 s, 0.02 percent difference). For steady-state accuracy, GWO had zero SSE, whereas GA has a very small error (0.0003). On the whole, the SMC-GWO controller has a quicker and smoother yaw response during disturbance and slightly better transient performance when compared with SMC-GA, as well as much better performance when compared with SMC and FLC.

Table 10. Performance comparison of disturbed controllers.

Variable	Controller	RT (s)	OS (%)	ST (s)	SSE
[A] Roll Angle (ϕ)	SMC	0.2355	0.52	1.4153	0.0052
	FLC	0.3840	0.00	1.6807	0.0000
	SMC-GWO	**0.0558**	**0.20**	**1.0975**	**0.0002**
	SMC-GA	0.0565	0.17	1.0983	0.0017
[B] Pitch Angle (θ)	SMC	0.1985	0.57	1.3582	0.0057
	FLC	0.3625	0.00	1.6970	0.0000
	SMC-GWO	**0.0113**	**10.12**	**1.0469**	**0.0009**
	SMC-GA	0.0105	11.40	1.0296	0.0001
[C] Yaw Angle (ψ)	SMC	0.2375	0.53	1.4173	0.0053
	FLC	0.4258	0.01	1.7776	0.0000
	SMC-GWO	**0.0469**	**0.00**	**1.0816**	**0.0000**
	SMC-GA	0.0472	0.00	1.0818	0.0003
[D] Altitude (z)	SMC	0.0940	9.90	1.3434	0.0023
	FLC	0.8000	25.00	2.1901	0.1431
	SMC-GWO	**0.0079**	**5.17**	**1.0303**	**0.0001**
	SMC-GA	0.0080	4.93	1.0320	0.0000

Table 10D shows that when a disturbance was applied at t = 1 s, the SMC-GWO controller exhibited a significant increase in the performance of altitude. It decreased the rise time to 0.0079 s, which was 91.60 percent better than the classical SMC (0.0940 s) and 99.01 percent better than the FLC (0.8000 s). It was also found that the overshoot was decreased to 5.17% with SMC-GWO (47.7 percent more than SMC and 79.3 percent more than FLC). The settling time of SMC-GWO was 1.0303 s and was faster by 23.28 and 52.94 percent as compared to SMC (1.3434 s) and FLC (2.1901 s), respectively. When comparing SMC-GWO and SMC-GA, it is seen that both controllers exhibit extremely similar transient behavior, compared to the classical SMC. The GWO-tuned controller has a slightly higher rise time (0.0079 s compared with 0.0080 s, 1.25 percent higher) and a smaller settling time (1.0303 s compared with 1.0320 s, 0.17 percent lower). Stable-wise accuracy teaches SMC-GA the zero SSE, and SMC-GWO has a negligible error (0.0001). Both are far more accurate in comparison with SMC and FLC. Altogether, the SMC-GWO controller offers the highest rate of response to altitude and significant decrease in rise time, overshoot, and settling period.

Then, faults within actuators are introduced both intentionally and at specific time points, including variances on bias side, time-varying external disturbances, and system uncertainties. These defects can occur at any time in the course of maneuvering. As a consequence, the time-varying Loss of Control Effectiveness (LoCE) actuator faults are included to determine the effectiveness of the proposed control scheme [42].

The external disturbances of high magnitude are determined by the following definition:

$$\begin{cases} D_\phi = 4.0\sin(2\pi t) + 1.0, & 0 \le t \le 10 \\ D_\theta = 3.8\cos(2\pi t) + 1.0, & 0 \le t \le 10 \\ D_\psi = 3.9\sin(2\pi t) + 1.0, & 0 \le t \le 10 \\ D_z = 4.0\,t + 0.1, & 0 \le t \le 10 \end{cases} \tag{34}$$

The time-varying actuator faults are defined as follows:

$$\begin{cases} u_\phi = (1 - 0.60\cos(2\pi t))\,U_2 + 0.80, & 1 \le t \le 3 \\ u_\theta = (1 - 0.70\sin(2\pi t))\,U_3 + 0.90, & 3 \le t \le 5 \\ u_\psi = (1 - 0.75\cos(\pi t))\,U_4 + 0.70, & 5 \le t \le 7 \\ u_z = (1 - 0.65\sin(2\pi t))\,U_1 + 0.85, & 7 \le t \le 10 \end{cases} \tag{35}$$

As Figure 23 shows, the proposed controller SMC-GWO is capable of following the desired trajectory precisely even when actuator faults and external disturbances are present. As a result, the controller is resilient to unexpected faults and disturbances with a small tracking error. Compared to FLC and SMC, the alternative controllers with faults and disturbances do not accurately track the reference trajectory, as indicated by increased tracking errors, and oscillatory action, as observed in Figure. In particular, concerning the altitude, the FLC experiences the highest tracking error, leading the UAV to the wrong direction, but the proposed controller maintains a smooth and stable path.

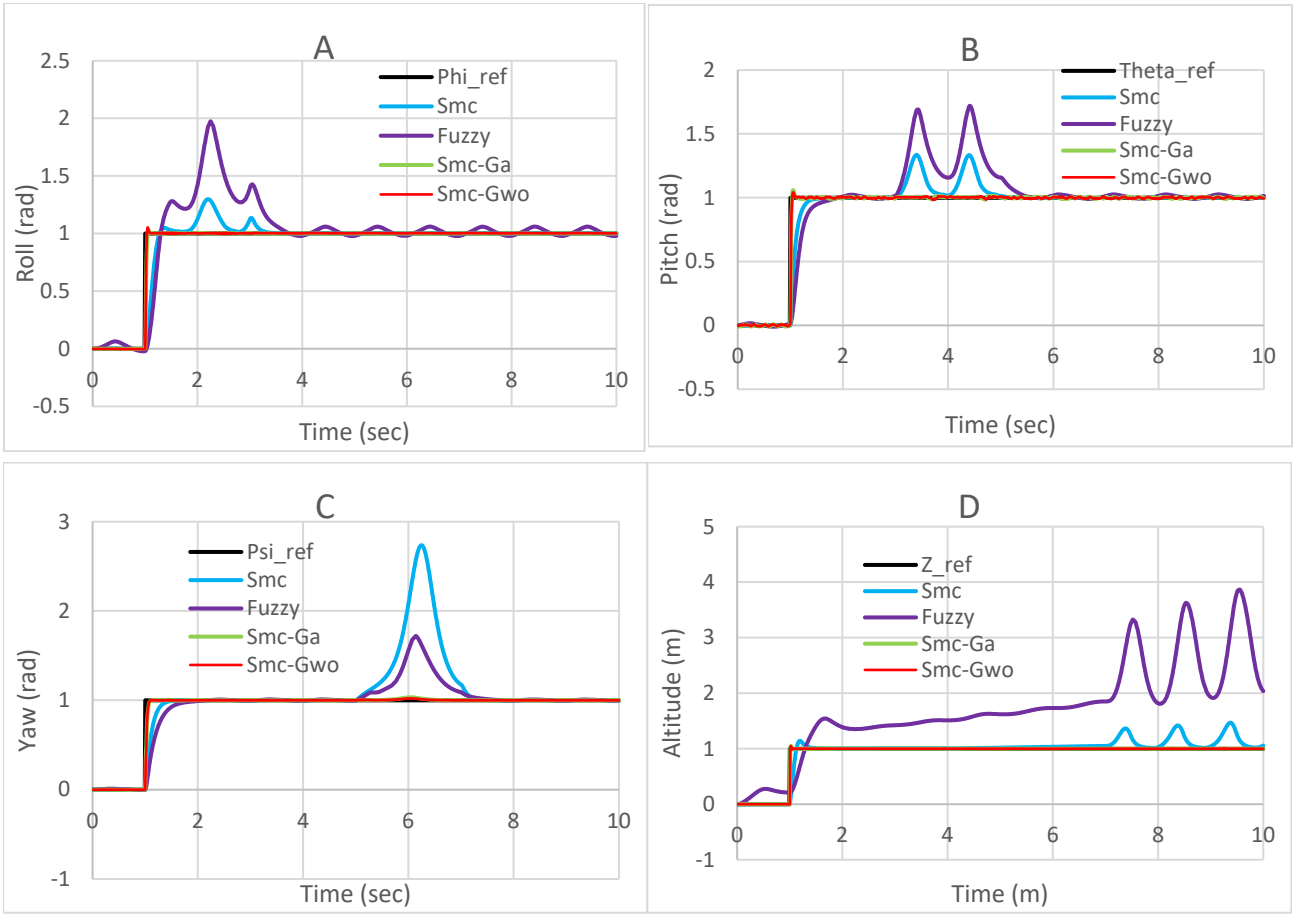


Figure 23. Desired and obtained trajectory simulation using different controllers with disturbance: (**A**) ϕ (roll), (**B**) θ (pitch), (**C**) ψ (yaw), and (**D**) z (altitude).

When compared to SMC-GA, it can be said that the proposed controller is slightly better, achieving smaller tracking errors and a slightly better dynamic response. Therefore, it has a relatively minor efficiency benefit compared to SMC-GA when it comes to actuator regulation of UAVs, and its superiority over FLC and SMC is significantly high and easily identifiable.

The controller performance is measured through Integral of Squared Error (ISE) that is determined as: $ISE_i = \int_0^t e_i^2(\tau)\, dt$.

Thus, the performance outcomes, provided in Table 11, prove that the suggested SMC-GWO controller provides significant tracking and performance ratios with actuator faults, unknown disturbances, and model uncertainties compared to traditional SMC and FLC schemes. Although the SMC-GWO and SMC-GA algorithms have similar performances, with nearly identical ISE values at each state, the SMC-GWO algorithm produces a slightly better total ISE, which is a slight stable increase in the overall tracking performance. Figure 24 below presents the control input responses under the presence of time-varying actuator faults, external disturbances, and model uncertainties.

Table 11. Performance analysis of the ISE on various controls.

States	ISE			
	SMC	**FLC**	**SMC-GA**	**SMC-GWO**
Roll	0.119210	0.569760	0.017126	0.015956
Pitch	0.110043	0.423183	0.013280	0.013274
Yaw	1.478021	0.335249	0.018535	0.019075
Height	0.172217	11.020925	0.010040	0.010040
Total	**1.879491**	**12.349117**	**0.058981**	**0.058345**

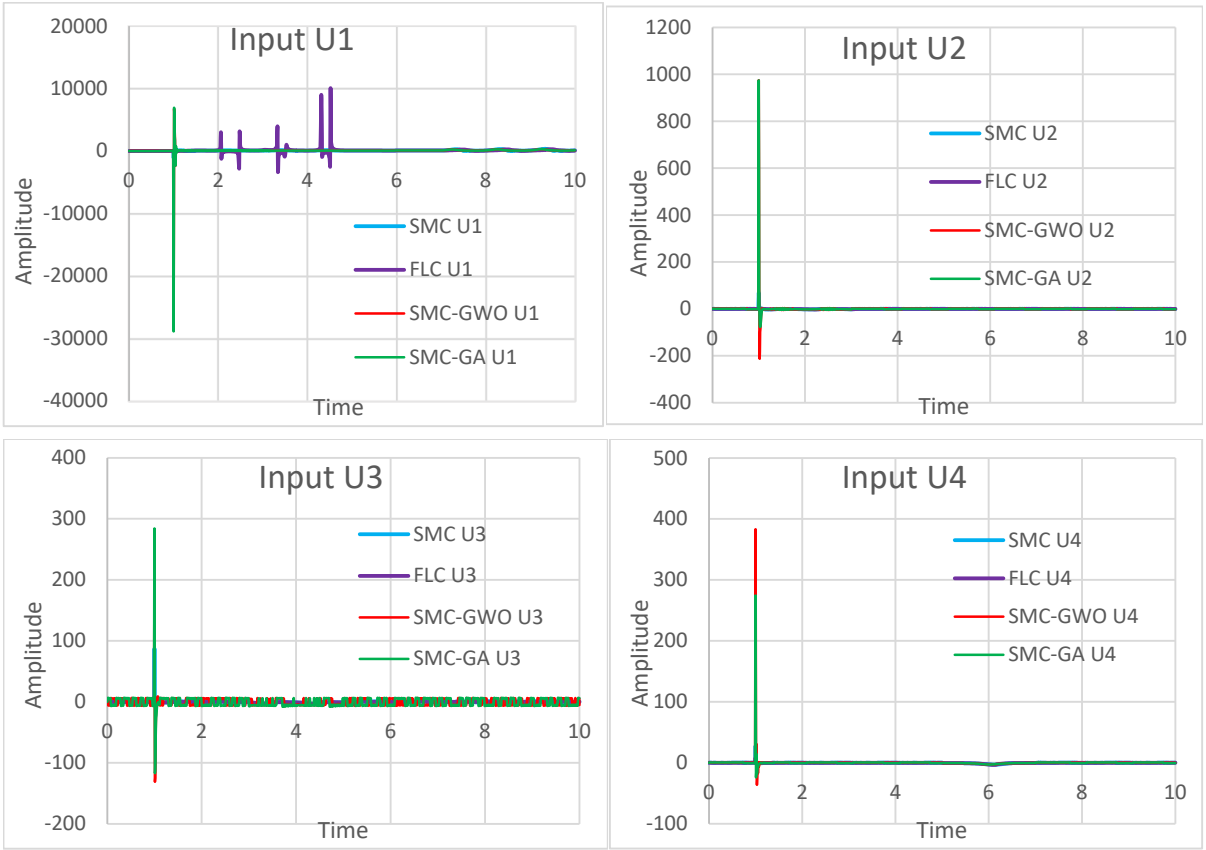


Figure 24. Control input response: (**U1**) ϕ (roll), (**U2**) θ (pitch), (**U3**) ψ (yaw), and (**U4**) z (altitude).

It should be noted that the control action starts at t = 1 s, where the nominal control input is applied and the reference signals begin to change from zero to their desired values. Therefore, transient peaks appearing around this instant are expected due to the sudden activation of the controller and the transition of the reference trajectory. The classical SMC and FLC show smoother responses, while SMC-GA and SMC-GWO generate short spikes due to their faster corrective action at the activation instant. The control inputs U1–U4 show that the classical SMC produces noticeable oscillations due to the discontinuous switching nature of the control law, which increases actuator activity. The FLC generates smoother control signals with smaller amplitudes; however, its robustness becomes weaker when disturbances and actuator faults are introduced. Both SMC-GA and SMC-GWO exhibit transient spikes around t = 1 s, which correspond to the instant where the nominal control input is applied and the system starts responding to the reference command. These peaks are caused by the fast corrective action required to drive the system toward the desired trajectory under changing conditions. Despite these short transients, the control signals remain bounded and quickly settle to stable values. Compared with SMC-GA, the proposed SMC-GWO shows more consistent control behavior with reduced oscillations after the transient period, indicating improved stability of the optimized sliding mode parameters. The presence of short peaks at the activation instant reflects fast response and strong disturbance rejection capability rather than instability. Therefore, the proposed SMC-GWO controller provides a better compromise between control effort, robustness, and fault-tolerant performance, making it more suitable for UAV operation under nominal and disturbed conditions.

The general results of the paper prove that the suggested SMC-GWO controller is a serious and statistically important improvement of the usual SMCs and FLCs in all channels of quadrotor control. It was also found that the proposed system was higher in performance in response time, tracking accuracy, and system stability, as well as dynamical behavior with smoother transitions, faster convergence, and reduced steady-state error. Even though the conventional SMC has a marked stability and near-zero overshoot, it is characterized by slower response and longer settling times. However, the conventional SMC remains superior to the FLC in general, because it offers more stable and predictable behavior under varied operating conditions. By contrast, the FLC performed well in attitude control, but poorly in altitude control, so that the oscillations were larger and steady-state errors were larger than with the traditional SMC. The hybrid SMC-GWO controller, employing both the stability of Sliding Mode Control (SMC) and intelligent tuning of the Gray Wolf Optimizer (GWO), was thus able to balance speed, accuracy in tracking, and stability, and it is clear that the hybrid controller performs better than the conventional approaches. To further confirm its usefulness in a meta-heuristic optimization environment, the proposed controller was also compared with the optimized SMC-GA controller. It was found that both of the optimized controllers achieved significant improvement in the performance of the baseline SMC, but the SMC-GWO had better dynamic behavior, such as reduced response time, reduced transition or fluctuations, better output stability, and an observable reduction in chattering. Even though SMC-GA was better in some respects, it was slower in reaction and takes more time to enter the steady state. The hybrid SMC-GWO controller can be discussed as the most successful and efficient of all the control approaches explored in this paper, based on these results. Therefore, the suggested controller offers a good and prospective solution to quadrotor UAV problems, necessitating clever, rapid, and accurate control within harsh and unpredictable working circumstances. The work thus provides the background for the design of new hybrid, self-adaptive, and intelligent control systems that constantly enhance performance according to new trends in advanced UAV control technologies.

5. Conclusions

This work performs an in-depth comparison of performance and analysis of various control strategies that are used in quadrotor unmanned aerial vehicles (UAVs), which in this case, are Sliding Mode Control (SMC), Fuzzy Logic Control (FLC), and a new hybrid controller known as Sliding Mode Control enhanced with the Gray Wolf Optimizer (SMC-GWO). The simulation uses a Newton–Euler formulation to simulate the nonlinear and coupled dynamics of a quadrotor, and all control algorithms are developed and tested in MATLAB/Simulink under the same operating conditions to provide a fair comparison. The quadrotor nonlinear and coupled dynamics were modeled based on the Newton–Euler model, and all control algorithms were realized and run in the MATLAB/Simulink environment under the same operating conditions to provide a fair comparison. The findings of the experiments support the validity of the optimized SMC-GWO controller in steadying a quadrotor UAV under both nominal and disturbed operational conditions. The proposed hybrid approach evidently outperforms traditional SMC and FLCs in speed of response, accuracy, and stability. The Gray Wolf Optimizer is efficient in optimizing the SMC parameters, resulting in faster convergence, reduced overshoot, as well as tiny steady-state error in all the flight channels phi, theta, psi, and z. The effective validation of SMC-GWO and its comparative superiority over SMC-GA underscores the potential of meta-heuristic optimization of the UAV control tasks. SMC-GA has an adequate response in some of these areas, but its response is comparatively slower and it takes more time to stabilize when compared to SMC-GWO. The suggested SMC-GWO controller exhibits strong and accurate tracking of the trajectory when actuator faults, external disturbances, and system uncertainty vary with time. It has a higher performance over traditional Sliding Mode Control (SMC) and Fuzzy Logic Control (FLC) technologies through many folds in terms of minimized tracking errors and maintained altitude control. Compared to the SMC-GA method, SMC-GWO has a slightly lower total of the square errors (ISE), which implies it can slightly improve the overall performance. These results support the effectiveness and robustness of the SMC-GWO strategy during unmanned aerial vehicle operation in fault-prone and highly uncertain environments. The method shows the better ability to quickly converge to the optimal control parameters, which results in less aggressive control actions, and the chattering can be significantly reduced. To conclude, the hybrid SMC-GWO controller is an efficient and trustworthy system to use in real-time UAVs. It can be taken as a promising and viable direction in the development of hybrid, self-adaptive, and intelligent control systems by combining the strength of Sliding Mode Control with the adaptive intelligence of nature-inspired optimization algorithms. This work provides simulation-level validation only; therefore, real-time experimental verification on embedded hardware will be addressed in future research to confirm the controller performance under real flight conditions.

Author Contributions: Conceptualization, K.Y.; methodology, K.Y. and M.B.N.; software, M.B.N. and K.Y.; validation, K.Y. and M.S.; formal analysis, M.B.N. and N.R.; investigation, K.Y. and M.B.N.; resources, S.A.D.; data curation, M.B.N. and N.R.; writing—original draft preparation, M.B.N. and K.Y.; writing—review and editing, K.Y., M.S., N.R. and S.A.D.; visualization, M.B.N.; supervision, K.Y.; project administration, K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

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

References

1. Zhang, Z.; Shi, H.; Wen, L.; Zhu, X. A novel UAV integrated with fixed-wing aircraft and rotorcraft. *Acad. J. Eng. Technol. Sci.* **2023**, *6*, 9–14. [CrossRef]
2. Sabour, M.H.; Jafary, P.; Nematiyan, S. Applications and classifications of unmanned aerial vehicles: A literature review with focus on multi-rotors. *Aeronaut. J.* **2023**, *127*, 466–490. [CrossRef]
3. Ariante, G.; Del Core, G. Unmanned Aircraft Systems (UASs): Current State, Emerging Technologies, and Future Trends. *Drones* **2025**, *9*, 59. [CrossRef]
4. Alqudsi, Y.; Sulaiman, H.; Pehlül, E. A comprehensive review of VTOL UAVs: Design, technologies, and applications. *Preprints* **2025**.
5. Peksa, J.; Mamchur, D. A review on the state of the art in copter drones and flight control systems. *Sensors* **2024**, *24*, 3349. [CrossRef]
6. Thu, N.T.; Gavrilov, A.I.; Myat Thu, K. Designing and modeling of quadcopter control system using L1 adaptive control. *Procedia Comput. Sci.* **2017**, *103*, 528–535. [CrossRef]
7. Kumar, T.; Hasan, M.I. Physics of quadcopter and its surveillance application: A review. *Int. J. Adv. Res. Eng. Technol.* **2020**, *11*, 570–578.
8. Mahony, R.; Kumar, V.; Corke, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [CrossRef]
9. Abdelmaksoud, S.I.; Mailah, M.; Abdallah, A.M. Control strategies and novel techniques for autonomous rotorcraft unmanned aerial vehicles: A review. *IEEE Access* **2020**, *8*, 183560–183589. [CrossRef]
10. Božić, M.; Vasiljević Toskić, M.; Bodić, M.; Rajs, V. Advantages of combination of PD and PID controller over PID controller in the example of quadcopter control and stabilization. *Int. J. Electr. Eng. Comput.* **2020**, *4*, 43–50. [CrossRef]
11. Bouabdallah, S.; Siegwart, R. Full control of a quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, 29 October 2007–2 November 2007*; IEEE: Piscataway, NJ, USA, 2007; pp. 153–158. [CrossRef]
12. Alyoussef, F.; Kaya, I. A review on nonlinear control approaches: Sliding mode control, back-stepping control and feedback linearization control. In *Proceedings of the International Engineering and Natural Sciences Conference (IENSC 2019), Diyarbakır, Turkey, 20–22 December 2019*; IENSC: Diyarbakır, Turkey, 2019; pp. 608–619.
13. Bouabdallah, S. *Design and Control of Quadrotors with Application to Autonomous Flying*; EPFL: Lausanne, Switzerland, 2007.
14. Yilmaz, M. Motion Control of Quadrotor System with Sliding Mode Control. Master's Thesis, Bursa Teknik Üniversitesi, Bursa, Turkey, 2015.
15. Pang, S.; Tian, L. Control methods for quadrotor's stability. *J. Phys. Conf. Ser.* **2021**, *1948*, 012109. [CrossRef]
16. Abbas, N. Survey of advanced nonlinear control strategies for UAVs. *Sensors* **2024**, *24*, 3286. [CrossRef]
17. Chang, X.H.; Liu, X.M.; Hou, L.W.; Qi, J.H. Quantized fuzzy feedback control for electric vehicle lateral dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 2331–2341. [CrossRef]
18. Wang, H.; Bai, W.; Zhao, X.; Liu, P.X. Finite-time-prescribed performance-based adaptive fuzzy control. *IEEE Trans. Cybern.* **2021**, *52*, 6959–6971. [CrossRef]
19. Rodríguez-Abreo, O.; Garcia-Guendulain, J.M.; Hernández-Alvarado, R.; Flores Rangel, A.; Fuentes-Silva, C. Genetic algorithm-based tuning of backstepping controller for quadrotor UAV. *Electronics* **2020**, *9*, 1735. [CrossRef]
20. Wang, Q.; Namiki, A.; Asignacion, A.; Li, Z.; Suzuki, S. Chattering reduction of sliding mode control for quadrotor UAVs. *Drones* **2023**, *7*, 420. [CrossRef]
21. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
22. Elagib, R.; Karaarslan, A. Sliding mode control-based modeling and simulation of a quadcopter. *J. Eng. Res. Rep.* **2023**, *24*, 32–41. [CrossRef]
23. Idrissi, M.; Annaz, F. Dynamic modelling of quadrotor. *Int. J. Mech. Eng. Robot. Res.* **2020**, *9*, 784–790.
24. Raza, S.A.; Gueaieb, W. Intelligent flight control of quadrotor. In *Proceedings of IEEE SMC, Istanbul, Turkey, 10–13 October 2010*; IEEE: Piscataway, NJ, USA, 2010; pp. 245–264.
25. Harkare, O.; Maan, R. Design and control of quadcopter. *Int. J. Eng. Technol. Res.* **2021**, *10*, 257–261.
26. Nguyen, H.T.T. Quadrotor UAV control techniques. *Int. J. Eng. Appl. Sci.* **2021**, *8*, 42–46.
27. Kırlı, A. İnsansız Dört Rotorlu Hava Araçları İçin Değişken Serbestlik Dereceli Yere Sabit Deney Düzeneği ve Denetleyici Tasarımı. Master's Thesis, İstanbul Teknik 979 Üniversitesi, İstanbul, Turkey, 2010.
28. Kuantama, E.; Tarcă, R. Quadcopter simulation using Simulink. *Robot. Manag.* **2015**, *20*, 1–6.
29. Bouabdallah, S.; Siegwart, R. Backstepping and sliding-mode control for quadrotor. In *Proceedings of ICRA, Barcelona, Spain, 18–22 April 2005*; IEEE: Piscataway, NJ, USA, 2005; pp. 2247–2252.
30. Eltayeb, A.; Rahmat, M.F.A.; Basri, M.A.M. Sliding mode control for quadrotor. *Int. J. Smart Sens. Intell. Syst.* **2020**, *13*, 1–13.
31. Abdelmaksoud, S.I.; Mailah, M. Optimizing flight path accuracy of quadrotor. *Int. J. Dyn. Control* **2024**, *12*, 4120–4137. [CrossRef]

32. Wang, C. Sliding-mode control for complex systems. *Math. Biosci. Eng.* **2022**, *19*, 2741–2764. [CrossRef]
33. Gedefaw, E.A.; Abdissa, C.M.; Lemma, L.N. Improved trajectory tracking control of quadrotor. *PLoS ONE* **2024**, *19*, e0308997.
34. Vilela, R.L.S.M.; Silva, E.C. Stability control using fuzzy logic. In *Proceedings of the International Micro Air Vehicle Conference, Madrid, Spain, 30 September–4 October 2019*; IMAV: Madrid, Spain, 2019; pp. 174–179.
35. Sattar, M.A.; Ismail, A. Modeling and fuzzy control of quadrotor. *Int. Res. J. Eng. Technol.* **2017**, *4*, 1494–1498.
36. Akyüz, S. *Design and Simulation of PD and Fuzzy Controllers*; Ege University: Izmir, Turkey, 2013.
37. Merheb, A.R.; Noura, H. Bioinspired tuning of quadrotor controller. In *Proceedings of the Australian Control Conference, Sydney, Australia, 15–16 November 2012*; IEEE: Piscataway, NJ, USA, 2012; pp. 227–232.
38. Abro, G.E.M.; Zulkifli, S.A.B.M.; Asirvadam, V.S.; Ali, Z.A. Model-free fuzzy SMC design. *Actuators* **2021**, *10*, 191. [CrossRef]
39. Ghoual, B.; Benyssaad, Y.; Meddahi, Y. PID-GWO control of quadcopter. In Proceedings of the CIESD Conference, Relizane, Algeria, 16–17 May 2023.
40. Pazooki, M.; Mazinan, A.H. Hybrid fuzzy SMC optimized by GA. *Complex Intell. Syst.* **2018**, *4*, 79–93. [CrossRef]
41. Orosco, V.; Chávez, D.; Camacho, O.; Aguilar, J. *Parameter Tuning of SMC Using GA*; Escuela Politécnica Nacional: Quito, Ecuador, 2018.
42. Ranjan, S.; Majhi, S. Fault-tolerant control for quadrotor UAV. *Int. J. Adapt. Control Signal Process.* **2024**, *39*, 132–151. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.