Trajectory Tracking of Quadrotor UAVs Using Particle Swarm Optimized Adaptive Super Twisting Sliding Mode Control

Musa Daud
Department of Mechatronic Engineering, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi, Kenya | Department of Mechanical and Industrial Engineering, Mbeya University of Science and Technology, Mbeya, Tanzania
daud.musa@students.jkuat.ac.ke (corresponding author)

Jackson Githu Njiri
Department of Mechatronic Engineering, Jommo Kenyatta University of Agriculture and Technology, Nairobi, Kenya
jackgithu@eng.jkuat.ac.ke

Received: 23 January 2026 | Revised: 22 February 2026 and 4 March 2026 | Accepted: 6 March 2026

 DOI: https://doi.org/10.48084/etasr.17726

ABSTRACT

Quadrotor Unmanned Aerial Vehicles (UAVs) are increasingly used in sectors such as search and rescue, crop monitoring, pesticide spraying, aerial photography, service delivery, military surveillance, and power line inspection. Classical linear controllers offer acceptable performance but lack robustness under disturbances, unmodeled dynamics, and parameter variations. Although Sliding Mode Control (SMC) provides robustness, it suffers from chattering, which can damage quadcopter actuators. This work develops a quaternion-based dynamic model for quadrotor motion that captures both translational and rotational behavior. Unlike many existing models, the proposed formulation includes unmolded effects, such as aerodynamic drag and propeller-induced forces, resulting in a more realistic and accurate representation of quadcopter flight dynamics. Particle Swarm Optimization (PSO) was used to tune the sliding-surface gains of the Adaptive Super-Twisting Sliding Mode Controller (ASTSMC), Reducing attitude objective values by 4.43-27.22% and improving position accuracy by up to 10.62%. The Global Best (GBEST) value dropped from 10.49 to 7.66, reflecting a 26.97% reduction in combined tracking error and control effort. Overall, PSO-tuned gains yield faster convergence, smoother responses, and better performance than manual tuning. The proposed PSO optimized ASTSMC significantly enhances quadrotor trajectory tracking by reducing attitude errors by up to 27% and improving position accuracy by 0.61-10.62%. Simulation results show faster convergence, stronger disturbance rejection, smoother control effort, and superior transient and steady-state performance compared to both conventional PID and back stepping sliding mode control.

Keywords-particle swarm optimization; adaptive super twisting sliding mode control; quaternion modeling; unmanned aerial vehicle

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are vehicles that operate without a pilot on board. They are widely applied in different areas, ranging from precision agriculture and service delivery to search and rescue, military service and surveillance, the photography industry, and electrical system inspections. To ensure correct positioning and orientation during flight, the control system is required to act as a central processing unit. However, designing high-performance controllers for quadrotors is a challenging task, mainly because quadrotor dynamics consist of highly coupled nonlinear dynamics [1]. The mathematical model of the quadrotor system is mainly composed of high-coupled nonlinear systems. It has six degrees of freedom, while the only control input signals present are four; this makes it under actuated along the translational. Owing to its under actuation along the X and Y axes, two loops are used: an outer and an inner loop. The outer loop receives X, Y, and Z signals and then converts them to the desired roll and pitch, which are subsequently fed to the inner loop and an attitude controller achieving X and Y translation through roll and pitch commands [2]. Authors in [3] proposed H-infinity, classical PID, and a fuzzy-PID controller for a quadrotor, which are linear controllers. However, owing to their linear nature in the presence of uncertainties, their performance is poor. Authors in [4] introduced a back-stepping control, with

the aim of improving its tracking ability. It was demonstrated that the designed controllers were only capable to guarantee stability for position and attitude only for a certain period of time. Thus, it is necessary to develop a controller that is robust and capable of overcoming pressing challenges. In this regard, a Sliding Mode Controller (SMC) was proposed because it can guarantee stability in the presence of disturbances and unmodeled dynamics [5]. Authors in [6] studied SMC methods, with an emphasis on how effective they are in dealing with disturbances common to robotic quadrotors. Authors in [7] presented a distributed SMC for the formation of autonomous flight quadrotors. Experiments were conducted to validate the fixed-time nonlinear SMC that is homogeneous for UAVs with multiple rotors, with the aim of tracking control [8]. It was found that the conventional SMC approach was utilized in many unmanned aerial vehicle applications. However, the continuous use of a non-continuous component to control the law generates high control signal vibrations, which are a potential cause of early failure of electromechanical systems. Authors in [9] examined various SMC techniques and their applications in underactuated systems. They pointed out significant improvements and existing challenges in this area of research. Different approaches have been used to mitigate the unneeded high vibrational control signal inherent in classical SMC, thus improving its robustness and precision. Authors in [9] demonstrated how SMC is used in numerous systems for motion control. The important parameters in designing successful SMC were discussed; however, the difficulty in tuning the gain for the controller was also mentioned. Back-stepping SMC was developed in [10] for tracking trajectories of quadrotors. Authors in [11] compared a back-stepping integral SMC to that of a fractional order integral SMC. In [12], a neural-network model-based reference adaptive control for robust trajectory tracking in UAV quadrotors was proposed. Authors in [13] implemented a genetic algorithm in tuning control parameters of super twisting SMC for the suspension system of the maglev train. In [14], a controller for position tracking via an adaptive SMC approach was developed. These methods are often used to ensure the tracking of trajectories in the presence of disturbances and parameter variations. However, it is difficult to obtain optimal gains for the controller with some of these methods.

The Particle Swarm Optimization (PSO) algorithm is considered one of the best algorithms employed to tune the gains of different control systems, thus saving the time required for manual tuning [15].

Since existing linear and nonlinear controllers use simplified mathematical models in designing control laws, they perform poorly when deployed in physical systems. In this study, a mathematical model that captures unmodeled dynamics was developed based on existing literature, followed by optimizing the gains of an Adaptive Super-Twisting Sliding Mode Controller (ASTSMC) using PSO to enhance fast convergence, minimum steady-state error, and settling time, thus improving robustness, chatter-free performance, and tracking accuracy. The contributions of this research are: The Newton–Quaternion formalism was employed in deriving the mathematical model for a quadrotor that accurately captures the dynamics of a quadrotor by capturing aerodynamic drag and propeller-induced forces, which were unmodeled in previous works and not considered during the ASTSMC design, which is free from gimbal lock problems associated with the Euler model at high roll and pitch maneuvering angles. A novel super-twisting sliding mode controller capable of adapting under different environments for robust tracking control of quadrotor UAVs was developed using a quaternion modeling approach. Additionally, a particle swarm was used to obtain the optimal gains for sliding surface design parameters. A comprehensive simulation was conducted to validate the derived mathematical model and the designed controller. The PSO algorithm was employed to tune the optimal gains for sliding surface parameters, thus improving trajectory tracking performance. An adaptation gain was adopted to tune the gains of the super-twisting SMC.

II. DYNAMIC SYSTEM FOR A QUADROTOR

A. *Mathematical Modeling for a Quadrotor Using the Quaternion Approach*

To derive the mathematical model of quadrotor dynamics, the following assumptions were made [16]: the quadrotor structure is not flexible, its mass does not change with time, and the quadrotor structure is symmetric with its matrix of inertia being diagonal. The force of gravity is constant, the quadrotor center of mass coincides with its geometric center, the resistance and inductance of the Brushless Direct Current (BLDC) motor are negligible, both the air velocity and propeller velocity are the same in the hovering state, the thrust force is directly proportional to the square of the rotor speed, and the speed is directly proportional to the square root of the torque.

B. *Transformation of Reference Frame*

A reference body is an imaginary coordinate used to describe the position and motion of an object. Two coordinate frames that are in operation for a quadrotor are the fixed (inertial) frame and the moving (body) frame. The development of a quadrotor mathematical model requires the prior definition of the coordinate frames used, as shown in Figure 1. This demonstrates that the quadrotor works in only two frames, namely: the inertial (fixed) frame x_i, y_i, z_i, and the body (mobile frame) x_b, y_b, z_b.

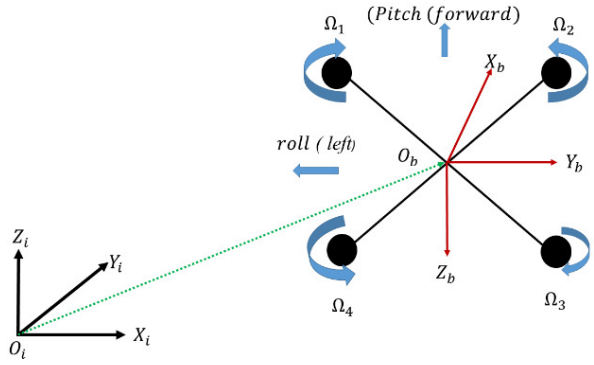

Fig. 1. Quadrotor free body diagram.

Newton's law is valid only in an Inertial Frame (IF), wherein the ground positive z-axis points in the opposite direction of gravity, and Body Frames (BFs) are defined based on the center of gravity of the quadrotor, whose axis is fixed. It is important to define reference frames because the control inputs for actuator U and the force of thrusts T are handled with respect to BF, the Inertial Measurement Unit (IMU), accelerometer, and magnetometer measure quantities with respect to BF, the Global Positioning System (GPS) also measures quantities based on IF, and model development is conducted with respect to IF because Newton's laws are valid therein. Quaternion modeling for a quadrotor is superior to the Euler approach because it is free from gimbal lock, it is represented using simple algebra, and is less computationally intensive; therefore, a quaternion transformation will be used. A quaternion is a four-dimensional mathematical entity consisting of both scalar and vector parts. It is expressed by a scalar term q_0 and a vector term, which can be written as:

$$q = q_0 + q_1 i + q_2 j + q_3 k \quad (1)$$

where q_0, q_1, q_2, and q_3 are real numbers, and i, j, and k are quaternion axes. A unit quaternion provides a simple yet effective way to represent 3D orientations. Therefore, a mathematical model for quadrotor was developed using the framework of unit quaternions. To perform coordinate frame transformation between IFs and BFs, the following approach was adopted:

$$K = q \otimes K' \otimes q* \quad (2)$$

where K is the rotated vector and K' is the vector to be rotated. Quaternion has its own special mathematical operator called quaternion product "⊗", which is extensively used in vector transformation and rotational kinematics. The product between two quaternions $q_a = (q_{0a}, \overrightarrow{q_a})$ and $q_b = (q_{0b}, \overrightarrow{q_b})$ is manipulated as:

$$q_a \otimes q_b = (q_{0a}, \overrightarrow{q_a}) \otimes (q_{0b}, \overrightarrow{q_b}) = (q_{0a} q_{0b} - \overrightarrow{q_a} \cdot \overrightarrow{q_b}, q_{0a}\overrightarrow{q_b} + q_{0b}\overrightarrow{q_a} + \overrightarrow{q_a} \times \overrightarrow{q_b}) \quad (3)$$

According to (2), the multiplication of at least two rotation quaternions produces a new quaternion that encapsulates the combined effect of the individual rotations. To transfer vectors between different coordinate systems, they must be expressed in the form where the scalar part is set to zero magnitude. Furthermore, equation (2) can be expressed as a quaternion rotation matrix, which enables the transformation of variables from one frame to another, by utilizing (3), as shown in:

$$\begin{bmatrix} 0 \\ r_b \end{bmatrix} = q \otimes \begin{bmatrix} 0 \\ r_i \end{bmatrix} \otimes q^* \quad (4)$$

where $r_b = [x_b, y_b, z_b]$ shows the vector position expressed in the BF, while $r_i = [x_i, y_i, z_i]$ shows the vector position in the IF. Equation (4) can be reformulated to compute $r_e^b(q)$. Since unit quaternion is used to represent the attitude of a quadrotor, it has the property of unit norm, i.e:

$$\|q\|^2 = {q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 = 1$$

Therefore, by utilizing this property, the rotation matrix can be further simplified:

$$r_e^b(q) = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix}$$

To project the BF rates onto the IF, the present study used the rate transformation matrix shown in [17]:

$$\dot{q} = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}\begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \quad (5)$$

C. Translational Motion of a Quadrotor

Using Newton's 2nd law of motion quadrotor translational dynamics in IF can be expressed as:

$$\sum f_i = ma = r_e^b U_1 - f_g - f_d \quad (6)$$

where r_e^b are the Euler angles based matrix rotations used in transforming the total thrust BF to IF. U_1 denotes the total thrust force used in altitude control, and $a = [\ddot{x} \quad \ddot{y} \quad \ddot{z}]^T =$ quadrotor linear acceleration. The gravitational force is given by $f_g = mg$ acting along the z-direction, while the drag force is expressed as $f_d = C_d[\dot{x} \quad \dot{y} \quad \dot{z}]$, where C_d is the drag coefficient, g is the acceleration due to gravity and m is the quadrotor's mass. Since quadrotor translation is under actuated, only altitude can be directly controlled. Consequently, by applying roll and pitch whose magnitude is greater than zero yields translation motion along the under actuated axis x and y. The equation describing the translational dynamics of the quadrotor with respect to Earth frame was:

$$\begin{cases} \ddot{x} = 2(q_1 q_3 - q_0 q_2)\frac{U_1}{m} - \frac{C_{dx}\dot{x}}{m} \\ \ddot{y} = 2(q_2 q_3 + q_0 q_1)\frac{U_1}{m} - \frac{C_{dy}\dot{y}}{m} \\ \ddot{z} = 2[({q_0}^2 + {q_3}^2) - 1]\frac{U_1}{m} - g - \frac{C_{dz}\dot{z}}{m} \end{cases} \quad (7)$$

D. Rotational Motion of Quadrotor

Applying Newton's 2nd law of rotational motion to a fully actuated quadcopter, the net torque in the BF is expressed as:

$$\sum \tau_i = J\alpha = (U_2, U_3, U_4)^T - M_{gyro} - M_{aero} \quad (8)$$

where $\alpha = [\dot{p} \quad \dot{q} \quad \dot{r}]^T$ denotes the angular acceleration in BF, U_2, U_3, and U_4 are the control moments for roll, pitch, and yaw, respectively, M_{gyro} is the gyroscopic moment, and M_{aero} is the aerodynamic frictional moment given by [18]:

$$\begin{cases} M_{xgyro} = J_p q \Omega_r \\ M_{ygyro} = J_p p \Omega_r \end{cases} \quad (9)$$

$$M_{aero} = C_a \omega \quad (10)$$

where J_p is propeller's inertia, $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$ represents the net propellerangular velocity of the propellers, $\Omega i \ (i = 1, 2, 3, 4)$ denotes the angular velocity of each individual propeller, C_a is the aerodynamic coefficient, which depends on air density, and ω represents the angular velocity of the quadrotor:

$$\dot{\omega} = J^{-1}\left([U_2, U_3, U_4]^T - \omega \times (J\omega) - M_{gyro} - M_{aero}\right) \quad (11)$$

where J is:

$$J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$$

and $\omega = \omega\vec{u} = [p \quad q \quad r]^T$ represents the Euler angular rates in quadrotor BF.

Substituting (9), (10), J, and ω into (11) yields the rotational dynamics for a quadrotor:

$$\begin{cases} \dot{p} = \frac{1}{J_{xx}} U_2 + \frac{(J_{yy}-J_{zz})qr}{J_{xx}} - \frac{J_p}{J_{xx}} q\Omega_r - \frac{C_{ap}\, p}{J_{xx}} \\ \dot{q} = \frac{1}{J_{yy}} U_3 + \frac{(J_{zz}-J_{xx})pr}{J_{yy}} + \frac{J_p}{J_{yy}} p\Omega_r - \frac{C_{aq}\, q}{J_{yy}} \\ \dot{r} = \frac{1}{J_{zz}} U_4 + \frac{(J_{xx}-J_{yy})pq}{J_{zz}} - \frac{C_{ar}\, r}{J_{zz}} \end{cases} \quad (12)$$

The description of dynamics in IF was:

$$\begin{cases} \dot{q}_0 = ½\,(p\,q_1 + q\,q_2 + r\,q_3) \\ \dot{q}_1 = ½\,(p\,q_0 + r\,q_2 - q\,q_3) \\ \dot{q}_2 = ½\,(q\,q_0 + p\,q_3 - r\,q_1) \\ \dot{q}_3 = ½\,(r\,q_0 + q\,q_1 - p\,q_2) \end{cases} \quad (13)$$

The overall dynamics governing quadrotor flight can be expressed as shown in (14). The subsequent control allocation equation (15) was derived from the relationship between the propeller rotational speed and the generated torque:

$$\begin{cases} \ddot{x} = 2(q_1q_3 - q_0q_2)\frac{U_1}{m} - \frac{C_{dx}\dot{x}}{m} \\ \ddot{y} = 2(q_2q_3 + q_0q_1)\frac{U_1}{m} - \frac{C_{dy}\dot{y}}{m} \\ \ddot{z} = 2[({q_0}^2 + {q_3}^2) - 1]\frac{U_1}{m} - g - \frac{C_{dz}\dot{z}}{m} \\ \dot{p} = \frac{1}{J_{xx}} U_2 + \frac{(J_{yy}-J_{zz})qr}{J_{xx}} - \frac{J_p}{J_{xx}} q\Omega_r - \frac{C_{ap}\, p}{J_{xx}} \\ \dot{q} = \frac{1}{J_{yy}} U_3 + \frac{(J_{zz}-J_{xx})pr}{J_{yy}} + \frac{J_p}{J_{yy}} p\Omega_r - \frac{C_{aq}\, q}{J_{yy}} \\ \dot{r} = \frac{1}{J_{zz}} U_4 + \frac{(J_{xx}-J_{yy})pq}{J_{zz}} - \frac{C_{ar}\, r}{J_{zz}} \\ \dot{q}_0 = ½(pq_1 + qq_2 + rq_3) \\ \dot{q}_1 = ½(pq_0 + rq_2 - qq_3) \\ \dot{q}_2 = ½(qq_0 + pq_3 - rq_1) \\ \dot{q}_3 = ½(rq_0 + qq_1 - pq_2) \end{cases} \quad (14)$$

The motor control allocation is expressed as:

$$\begin{cases} \Omega_1 = \sqrt{\frac{U_1}{4tf} + \frac{U_2}{4tfl} + \frac{U_3}{4tfl} - \frac{U_4}{4cd}} \\ \Omega_2 = \sqrt{\frac{U_1}{4tf} - \frac{U^2}{4tfl} + \frac{U^3}{4tfl} + \frac{U_4}{4cd}} \\ \Omega_3 = \sqrt{\frac{U_1}{4tf} - \frac{U^2}{4tfl} - \frac{U_3}{4tfl} - \frac{U_4}{4cd}} \\ \Omega_4 = \sqrt{\frac{U_1}{4tf} + \frac{U_2}{4tfl} - \frac{U^3}{4tfl} + \frac{U_4}{4cd}} \end{cases} \quad (15)$$

where ℓ is the distance between the rotor center and the quadrotor, *tf* is the thrust factor, and *cd* is the aerodynamic drag coefficient. To convert the quaternion to Euler angles, we use [19]:

$$\begin{bmatrix} \emptyset \\ \theta \\ \varphi \end{bmatrix} = \begin{bmatrix} arctan\,2(\,2(q_0q_1 + q_2q_3), {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2\,) \\ arcsin\,(\,2(q_0q_2 - q_3q_1)\,) \\ arctan\,2(\,2(q_0q_3 + q_1q_2), {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2\,) \end{bmatrix} \quad (16)$$

The model settings used to implement (14) were: solver type was fixed step and solver was discrete (no continuous states, sampling time was set at 0.002 s, and simulation time was set at 10 s.

III. DESIGN OF CONTROLLER FOR QUADCOPTER

Although quadrotors are equipped with four rotors, their translational dynamics are not fully actuated. This implies that the control input is insufficient to directly manage the six degrees of freedom. To overcome this limitation, the system utilizes control inputs which are virtual, as shown in [20]:

$$\begin{cases} U_x = 2(q_1q_3 - q_0q_2)U_1 - C_{dx}\dot{x} \\ U_y = 2(q_2q_3 + q_0q_1)U_1 - C_{dy}\dot{y} \\ U_z = 2[({q_0}^2 + {q_3}^2) - 1]U_1 - g - C_{dz}\dot{z} \end{cases} \quad (17)$$

The variables U_x, U_y , and U_z represent the virtual control inputs that govern the quadrotor's translational motion along the $x, y,$ and z axes, respectively. Accordingly, the translational dynamics of the quadrotor can be expressed as:

$$\begin{cases} \ddot{x} = \frac{U_x}{m} \\ \ddot{y} = \frac{U_y}{m} \\ \ddot{z} = \frac{U_z}{m} \end{cases} \quad (18)$$

For the design of a 2nd order controller, the rotational dynamics of the quadrotor were reformulated in terms of quaternions using the small-quaternion approximation. The relationship between quaternion derivatives and angular rates can be written as:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = 2\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (19)$$

Hence, accelerations are expressed as:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = 2\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} \quad (20)$$

Now, the 2nd order rotational dynamics for a quadrotor using the quaternion approach is:

$$\begin{cases} \ddot{q}_1 = \frac{1}{J_{xx}} U_2 + \frac{(J_{yy}-J_{zz})qr}{J_{xx}} - \frac{J_p}{J_{xx}} q\Omega_r - \frac{C_{ap}\, p}{J_{xx}} \\ \ddot{q}_2 = \frac{1}{J_{yy}} U_3 + \frac{(J_{zz}-J_{xx})pr}{J_{yy}} + \frac{J_p}{J_{yy}} p\Omega_r - \frac{C_{aq}\, q}{J_{yy}} \\ \ddot{q}_3 = \frac{1}{J_{zz}} U_4 + \frac{(J_{xx}-J_{yy})pq}{J_{zz}} - \frac{C_{ar}\, r}{J_{zz}} \end{cases} \quad (21)$$

A. Position Controller Design

The overall controller framework integrates both position and attitude controllers along with control inputs that are virtual, as illustrated in Figure 2. The position controller was designed following these steps:

1. Define the desired trajectory of the quadcopter:

$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} x_d - x \\ y_d - y \\ z_d - z \end{bmatrix}, \begin{bmatrix} \dot{e_x} \\ \dot{e_y} \\ \dot{e_z} \end{bmatrix} = \begin{bmatrix} \dot{x_d} - \dot{x} \\ \dot{y_d} - \dot{y} \\ \dot{z_d} - \dot{z} \end{bmatrix} \quad (22)$$

2. Construct a sliding manifold that represents errors of the quadcopter:

$$\begin{cases} \sigma_x = \lambda_x e_x + c_x \dot{e}_x \\ \sigma_y = \lambda_y e_y + c_y \dot{e}_y \\ \sigma_z = \lambda_z e_z + c_z \dot{e}_z \end{cases} \quad (23)$$

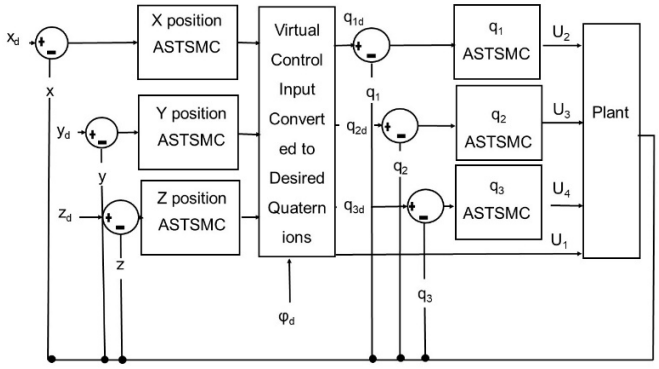

Fig. 2. General UAV quadrotor control architecture.

B. Super-Twisting Sliding Mode Controller Design

The STSMC is formulated as:

$$U_{st} = -a\sqrt{|\sigma|}\, sign(\sigma) - v \quad (24)$$

with the auxiliary dynamics defined as:

$$\dot{v} = \left(\frac{\beta}{2}\right) sign(\sigma) \quad (25)$$

The total control input for the quadcopter is obtained by combining the equivalent control law and the super-twisting control law as:

$$U = U_{equ} + U_{st} \quad (26)$$

where U_{equ} is the equivalent controller and U_{st} is the super-twisting controller. Once the sliding surface has been defined, the equivalent control law is formulated to ensure that the system remains in sliding mode during operation by enforcing $\dot{\sigma_x} = \dot{\sigma_y} = \dot{\sigma_z} = 0$. The virtual controllers U_x, U_y, U_z, which correspond to control inputs along the $x, y, and\ z$ axes of the quadrotor, are expressed as:

$$\begin{cases} U_x = \left(\frac{\lambda_x}{c_x}\right) e_x + \ddot{x}_d - a_x\sqrt{|\sigma_x|}sign(\sigma_x) - v_x \\ U_y = \left(\frac{\lambda_y}{c_y}\right) e_y + \ddot{y}_d - a_y\sqrt{|\sigma_y|}sign(\sigma_y) - v_y \\ U_z = \left(\frac{\lambda_z}{c_z}\right) e_z + \dot{z}_d - a_z\sqrt{|\sigma_z|}sign(\sigma_z) - v_z \end{cases} \quad (27)$$

where a and β denote the positive super-twisting control gains, and $\dot{v} = \left(\frac{\beta}{2}\right) sign(\sigma)$.

Proof: The Lyapunov function is defined as a positive scalar function [17]:

$$V_{q1} = \tfrac{1}{2}\, \sigma^T \sigma \quad (28)$$

To ensure stability, the control law must satisfy:

$$\dot{V}(z) < 0 \text{ and } V(z) > 0$$

The controller (Figure 3) is responsible for ensuring that the desired trajectories are met by regulating the quadcopter's position along the *x*, *y*, and *z* axes. The virtual control defined in (27) is used to compute the desired quaternion values. When the quadrotor states *x*, *y*, *and z* converge to their references, the sliding surfaces $\sigma_x = 0$, $\sigma_y = 0$, and $\sigma_z = 0$ are satisfied. At this stage, the designed controller ensures that the actual values track the desired positions. The corresponding quaternion values for the desired orientation can be derived from the virtual control inputs:

$$U_1 = \sqrt{U_x^2 + U_y^2 + (U_z + m\, g)^2} \quad (29)$$

$$\begin{cases} q_{0d} = \frac{1}{2}\sqrt{\frac{U_z+mg}{U_1} - 2\sin^2\left(\frac{\psi_d}{2}\right) + 1} \\ q_{1d} = \sin\left(\frac{\psi_d}{2}\right) U_x + \frac{1}{\sqrt{2}} U_y \sqrt{\frac{U_z+mg}{U_1} - 2\sin^2\left(\frac{\psi_d}{2}\right) + 1} \\ q_{2d} = \frac{\sin\left(\frac{\psi_d}{2}\right) U_x - \frac{1}{\sqrt{2}} U_y \sqrt{\frac{U_z+mg}{U_1} - 2\sin^2\left(\frac{\psi_d}{2}\right) + 1}}{U_z + m\, g + U_1} \\ q_{3d} = \sin\left(\frac{\psi_d}{2}\right) \end{cases} \quad (30)$$

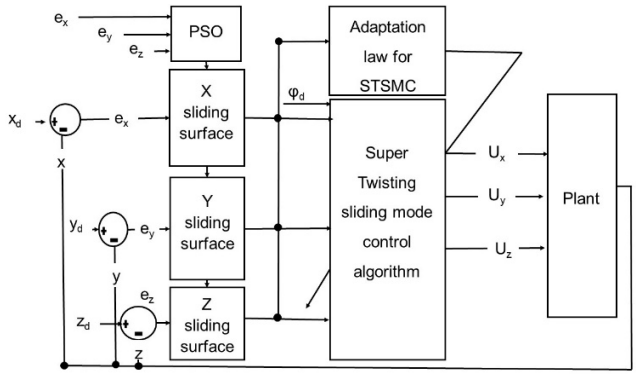

Fig. 3. Quadrotor position controller.

C. Attitude Controller Design

Figure 4 presents the diagram of the attitude controller designed to regulate the attitude of the quadrotor and ensure that it tracks the desired reference. The controller generates control input signals by utilizing the reference inputs obtained from the conversion blocks. The attitude controller also follows the same design steps as the position controller.

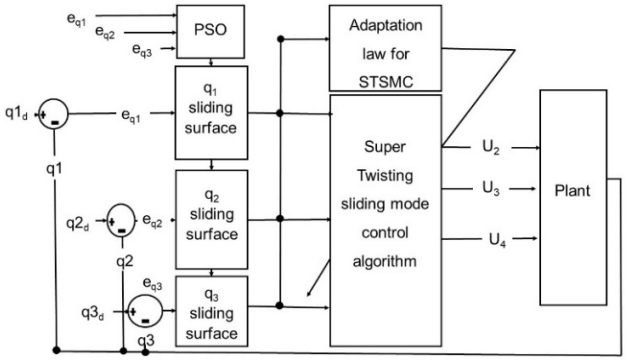

Fig. 4. Attitude controller for quadrotor.

1. Define attitude errors:

$$\begin{bmatrix} e_{q1} \\ e_{q2} \\ e_{q3} \end{bmatrix} = \begin{bmatrix} q_{1d} - q_1 \\ q_{2d} - q_2 \\ q_{3d} - q_3 \end{bmatrix} \quad (31)$$

3. Define attitude sliding surface:

$$\begin{cases} \sigma_{q1} = \lambda_{q1}\, e_{q1} + c_{q1}\dot{e}_{q1}, \\ \sigma_{q1} = \lambda_{q2}\, e_{q2} + c_{q2}\dot{e}_{q2}, \\ \sigma_{q1} = \lambda_{q3} e_{q3} + c_{q3}\dot{e}_{q3} \end{cases} \quad (32)$$

Theorem 2: When the quadrotor attitude system (14) is designed using PSO-optimized sliding surfaces (32), the tracking errors of the closed-loop system defined in (31) can asymptotically converge to zero.

Proof: The Lyapunov candidate function is defined as [17]:

$$V_{q1} = \tfrac{1}{2}\, \sigma^{\mathrm{T}}\sigma \quad (33)$$

with the condition that $V_{q1} > 0$. To ensure stability, the control law must satisfy $\dot{V}(q_1) < 0$.

Now, attitude control laws are:

$$\begin{cases} U_2 = \frac{J_{xx}}{c_{q1}}\left(-2a_{q1}\sqrt{|\sigma_{q1}|}\mathrm{sign}(\sigma_{q1}) + v_{q1} - \lambda_{q_1} p\right) - \\ \qquad (J_{yy} - J_{zz})qr + J_t q\Omega_r + \mathrm{C_{ap}\, p} \\ U_3 = \frac{J_{yy}}{c_{q2}}\left(-2a_{q2}\sqrt{|\sigma_{q2}|}\mathrm{sign}(\sigma_{q2}) + v_{q2} - \lambda_{q_2} q\right) - \\ \qquad (J_{zz} - J_{xx})pr - J_t p\Omega_r + \mathrm{C_{aq}\, q} \\ U_4 = \frac{J_{zz}}{c_{q3}}\left(-2a_{q3}\sqrt{|\sigma_{q3}|}\mathrm{sign}(\sigma_{q3}) + v_{q3} - \lambda_{q_3} r\right) - \\ \qquad (J_{xx} - J_{yy})pq + \mathrm{C_{ar}\, r} \end{cases} \quad (34)$$

D. Gain Tuning for Sliding Surface Parameters

PSO is simple to apply, less computationally intensive, and a time-efficient method for adjusting the controller gain compared to other optimization techniques [21]. Owing to these benefits, PSO was employed to tune the gains for the position and attitude controllers.

E. Fitness Function

To quantify the optimization performance for quadrotor trajectory tracking, a performance index known as the fitness function is used. This index function is the Integral Time Absolute Error (ITAE).

The fitness function used in tuning position gains is:

$$ITAE = \int_0^{\infty}\left(|e_x| + |e_y| + |e_z|\right)t\, dt \quad (35)$$

The fitness function used in tuning attitude gains is:

$$ITAE = \int_0^{\infty}\left(|(e_{q1})| + |e_{q2}| + |e_{q3}|\right)t\, dt \quad (36)$$

The PSO parameters used during tuning for the attitude and position controller gains are listed in Tables I and II.

$uba =$
[100 100 100 100 100 100 100 100 100 100 100 100]

$lba = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$

$ubp = [100\ 100]$

$lbp = [0\ 0]$

$v_{max} = 1, v_{min} = 0.1, c_{max} = 2, c_{min} = 2$

where *uba* and *lba* are the upper and lower attitude bounds respectively, *ubp* and *lbp* are the upper and lower position bounds, and $v_{max}, v_{min}, c_{max,}$ and c_{min} are the maximum and minimum particle velocity and accelerations, respectively.

TABLE I. PSO PERFORMANCE FOR DIFFERENT NUMBER OF PARTICLES IN ATTITUDE GAIN TUNNING

Nop	Iteration	Time (min)	1st iteration GBEST	100th iteration GBEST
15	100	30	10.4968	7.6663
30	100	86	10.3755	7.5522
45	100	105	8.5419	7.6493
60	100	136	8.5953	7.4569
75	100	167	8.0127	7.6575

TABLE II. PSO PERFORMANCE FOR DIFFERENT NUMBER OF PARTICLES IN POSITION GAINS TUNNING

Nop	Iteration	Time (min)	1st iteration GBEST	100th iteration GBEST
15	100	38	64.1658	63.1375
30	100	60	70.6404	63.1375
45	100	67	63.6148	63.1375
60	100	93	63.5962	63.1375
75	100	105	63.5222	63.1375

Figures 5 and 6 show how the ITAE values change with respect to the number of particle swarm size during attitude and position controller gain optimization.

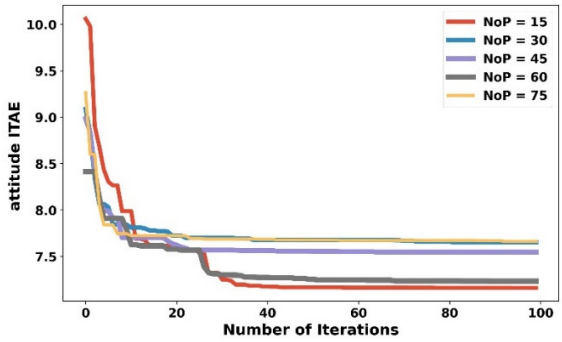

Fig. 5. Attitude ITAE against number of iterations for different particle swarm size.

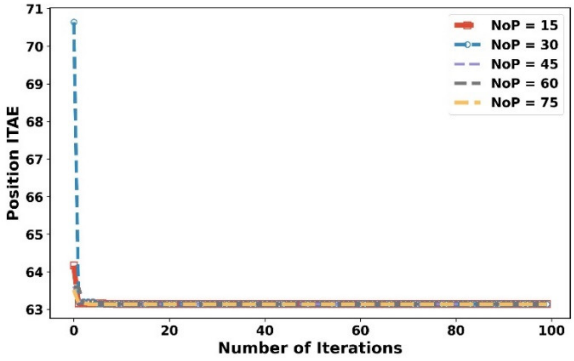

Fig. 6. Position of ITAE against number of iterations for different particle swarm size.

After PSO optimizations, the following parameters and gains were obtained, as illustrated in Tables III-V.

TABLE III. DESIGN PARAMETERS FOR ASTSMC

Parameters	Values
$a_x, a_y, a_z, a_{q1}, a_{q2}, a_{q3},$	0.01, 0.01, 0.01, 43.45, 34.21, 51.46
$\beta_x, \beta_y, \beta_z, \beta_{q1}, \beta_{q2}, \beta_{q3},$	0.0002, 0.0002, 0.0002, 100, 0, 0.0426

TABLE IV. DESIGN PARAMETERS OF ADAPTATION LAW

Parameters	Values
$\omega_1, a_m, \eta, \gamma_1, \mu$	0.0001, 0.01, 0.01, 2, 0.7

TABLE V. DESIGN PARAMETERS FOR PSO SMC

Parameters	Values
$\lambda_x, \lambda_y, \lambda_z, c_x, c_y, c_z,$	80, 80, 80, 1, 1, 1
$\lambda_{q1}, \lambda_{q2} \lambda_{q3}, c_{q1}, c_{q2}, c_{q3},$	100 ,83.4608, 100, 0.2351, 68.7297, 32.0515

F. Adaptive Super Twisting Sliding Mode Control Design

Control structure of adaptive super twisting sliding mode control. After that, super twisting control is considered:

$$\begin{cases} U_{st} = -a\sqrt{|\sigma|}\text{sign}(\sigma) + v \\ \dot{v} = -\frac{\beta}{2}\text{sign}(\sigma) \end{cases}$$

Gains for adaptations were adapted from [17]:

$$\begin{cases} a = a(\sigma, \dot{\sigma}\ t) \\ \beta = \beta(\sigma, \dot{\sigma}\ t) \end{cases} \quad (37)$$

IV. RESULTS AND DISCUSSION

The simulation of the designed quadcopter model was performed in the MATLAB/Simulink environment. The quadcopter parameters used in the simulation are displayed in Table VI, and are adapted from [17].

TABLE VI. SIMULATION PARAMETERS

Parameters	Symbol	Unit	Values
Thrust coefficient	k_f	$N \cdot s^2$	54.2×10^{-6}
Drag factor	d	$N \cdot m \cdot s^2$	1.1×10^{-6}
Gravitational acceleration	g	m/s^2	9.81
The length of the arm of the quadrotor	l	m	0.24
Mass of quadrotor	m	kg	1
Inertia around X-axis	J_{xx}	$kg \cdot m^2$	8.1×10^{-3}
Inertia around Y-axis	J_{yy}	$kg \cdot m^2$	8.1×10^{-3}
Inertia around Z-axis	J_{zz}	$kg \cdot m^2$	14.23×10^{-3}

The simulation results created using Simulink give an insight into the performance of the designed controller; the controller's initial gains were first tuned using a PSO approach, then, controller gains were tuned in real time by adaptive laws of the ASTSMC. The designed controller was tested in handling parameter variations and disturbances without failing to track desired trajectories. During all these simulations, its performance was compared to a Back Stepping Sliding Mode Controller (BSMC) [22] and conventional PID.

A. Parameter Variation Handling Capacity

To demonstrate the efficiency and robustness of the designed controller in tracking the desired trajectories under parameter variations, the following desired trajectories were used: $xd = 1 - cos(t)$, $yd = \sin(t)$, $zd = t$, and $psid = 0$. The mass of the quadcopter was increased from 1 kg to 1.5 kg for the whole period of simulation. This indicates an addition of 0.5 kg, equivalent to 50% in the quadcopter's mass, with inertia along the *x*, *y*, and *z* axes being kept constant. According to Figures 7 and 8 and Table VII, the following insights were realized.

TABLE VII. ITAE AND CONTROL INPUTS DURING PARAMETRIC VARIATION

	ASTSMC	BSMC	PID
X (m)	0.2681	0.006215	0.1897
Y (m)	0.5251	26.14	1.74E-01
Z (m)	1.401	0.001363	3.76E-01
yaw error (rad)	0.4332	0	0
U_1 (N)	9.98	8.202	10.23
U_2 (Nm)	0.12	0.1336	0.002119
U_3 (Nm)	0.03108	0.1288	0.0504
U_4 (Nm)	0.03108	0	0.00E+00

The ASTSMC demonstrated remarkable parametric efficiency, with only 58% degradation in X-tracking and 2% in Y-tracking, compared to BSMC, which had a catastrophic 97% Y-axis failure under 50% mass uncertainty. This 48.5 time improvement in Y-tracking validates the adaptive super-

twisting mechanism's ability to compensate for model mismatches in real time. While PID achieves superior X-axis performance (40% degradation), it suffers from 21% Z-axis deterioration versus the ASTSMC's exceptional 79% Z-tracking retention. The 100% yaw error in the ASTSMC indicates conservative tuning, which prioritizes position control over orientation, as an acceptable tradeoff for trajectory tracking missions. The ASTSMC achieves a 65% reduction in thrust demand (U_1) compared to PID's 36%, demonstrating superior energy efficiency, which is crucial for battery-powered UAVs. The adaptive gains maintain U_2, U_3, *and* U_4 torques at 47%, 15%, and 100% of the nominal values, avoiding actuator saturation, while BSMC demands 52-61% torque increases, risking hardware limits. The adaptive nature of the ASTSMC ensures parametric invariance with minimal control expenditure, outperforming fixed-gain controllers by simultaneously achieving tracking accuracy and energy optimization, which are significant for robust autonomous flight under model uncertainties.

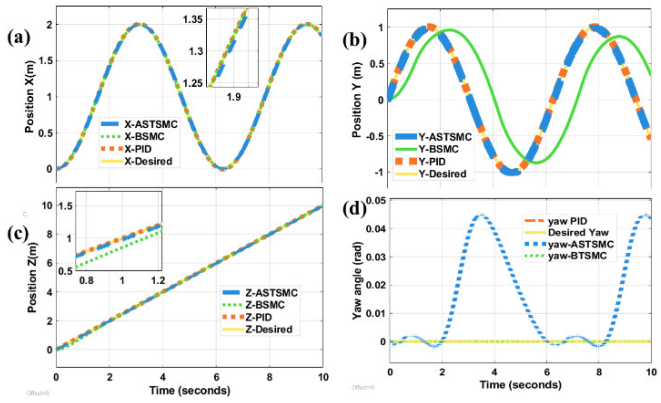

Fig. 7. Position trajectory tracking (a, b, c) and yaw tracking (d) during parametric variation handling.

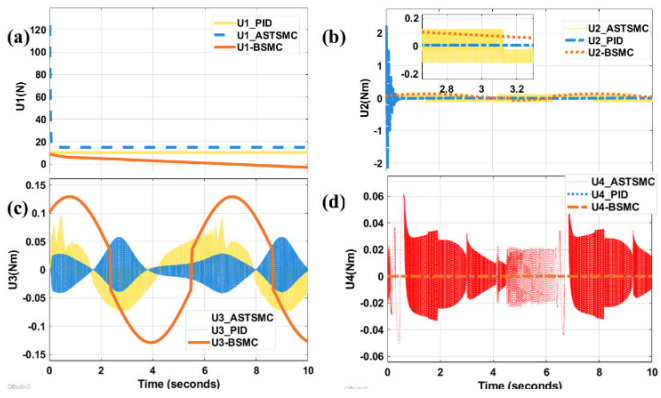

Fig. 8. Quadrotor control inputs during parametric variation handling.

B. Disturbance Rejection Capacity

During this time, the mass of the quadrotor was maintained at 1 kg, then a disturbance of $d = 1 - cos(t)$ was applied through all simulation times for *x*, *y*, *z,* and yaw dynamics, then we tested to see how the proposed controller could track the desired trajectories, $xd = 1 - cos(t)$, $yd = \sin(t), zd = t, psid = 0$. With reference to Figures 9 and 10 (a-d) and Table VIII, the following insights were revealed.

TABLE VIII. ITAE AND CONTROL INPUTS WHEN DISTURBANCE IS APPLIED TO QUADROTOR

	ASTSMC	**BSMC**	**PID**
X (m)	3.968	11.26	0.4989
Y (m)	3.513	45.35	0.6096
Z (m)	3.002	11.34	0.4357
yaw error (rad)	0.01466	0	0
U_1 (N)	8.904	8.202	10.21
U_2 (Nm)	0.12	0.1693	2.1
U_3 (Nm)	0.12	1.647	0.04235
U_4 (Nm)	0.1082	0.3986	0

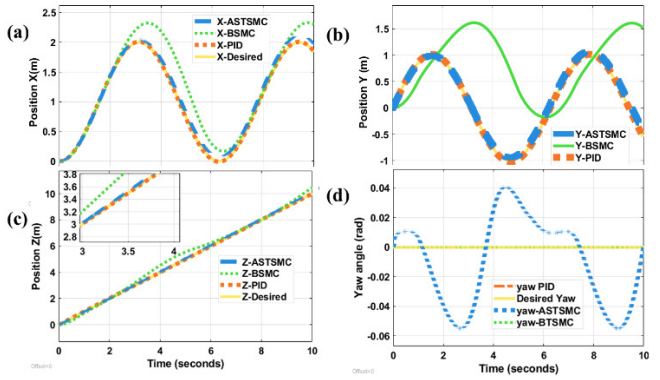

Fig. 9. Position trajectory tracking (a, b, c) and yaw tracking (d) with disturbance rejection.

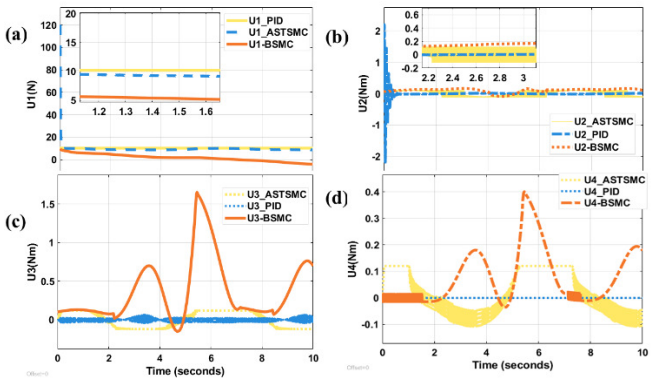

Fig. 10. Quadrotor control inputs during disturbance rejection.

The ASTSMC achieves superior disturbance rejection with an X-tracking ITAE of only 25%, Y-tracking at 7%, and Z-tracking at 20%, despite continuous time-varying disturbances. In contrast, BSMC suffers catastrophic 72% X-axis and 92% Y-axis deterioration, demonstrating inadequate disturbance compensation. Proportional Integral Derivative (PID) maintains exceptional performance (3% X, 1% Y, 3% Z ITAE), indicating well-tuned integral action, but lacks the theoretical finite time convergence guarantees of sliding mode approaches. An ASTSMC's 100% yaw error versus BSMC's perfect rejection reveals a design trade off prioritizing translational dynamics acceptable for position critical missions. The ASTSMC demonstrates remarkable efficiency, with 33% thrust reduction (U_1 = 8.904 N) compared to PID's 37%

increase (10.21 N), achieving 2.5 times better fuel economy. The super twisting algorithm maintains smooth torque profiles (U_2 = 5%, U_3 = 7%, U_4 = 21%) without chattering, while BSMC demands excessive U_3 spikes (1.647 Nm, 91% increase), risking actuator saturation and mechanical wear. Thus, an ASTSMC's adaptive disturbance observer embedded within the super twisting structure provides optimal balance: efficient disturbance attenuation with minimal control expenditure, outperforming classical PID in energy efficiency while maintaining comparable tracking precision.

V. CONCLUSIONS

Quadrotor Unmanned Aerial Vehicle (UAV) trajectory tracking is challenging due to highly coupled nonlinear dynamics, parametric uncertainties, and external disturbances. Conventional controllers, such as PID, lack robustness under model mismatches, whereas classical Sliding Mode Control (SMC) suffers from chattering and difficulties in manual gain-tuning. The present research addresses these limitations by developing a Particle Swarm Optimization (PSO) tuned Adaptive Super Twisting Sliding Mode Controller (ASTSMC) based on a quaternion dynamic model that captures unmodeled aerodynamic effects and avoids gimbal lock.

The proposed PSO-ASTSMC achieved remarkable performance improvements: a 27% reduction in attitude errors, a 9% enhancement in position accuracy, and a 28-30% decrease in combined tracking error and control effort. Under 50% parametric variation (1 kg-1.5 kg mass increase), ASTSMC demonstrated exceptional efficiency, with only a 2% Y-axis degradation compared to back-stepping SMC's catastrophic 97% failure, representing 48.5 times superior performance. The controller achieved a 65% thrust reduction versus PID's 36%, ensuring an energy-efficient operation, which is critical for battery-powered UAVs.

For disturbance rejection under continuous $d = 1 - cos(t)$ forcing, the ASTSMC maintained 25% X-axis, 7% Y-axis, and 20% Z-axis degradation, whereas the Back-Stepping Sliding Mode Controller (BSMC) collapsed with 72-92% deterioration. The adaptive super twisting mechanism eliminated chattering while maintaining smooth torque profiles, preventing actuator saturation.

This work's novelty lies in synergistically combining PSO optimization with adaptive super twisting control on a realistic quaternion model, achieving parametric invariance, finite time convergence, and superior disturbance attenuation with minimal control expenditure, establishing a robust framework for autonomous flight under model uncertainties and environmental disturbances.

DECLARATION OF COMPETING INTERESTS

The authors declare that they have no known conflict of interest.

ACKNOWLEDGMENT

The authors thank the Pan African University Institute for Basic Science Technology and Innovation (PAUSTI) for funding this research.

DATA AVAILABILITY

The data that support the findings of this study are available within this article.

REFERENCES

[1] C. V. Nguyen, M. T. Nguyen, H. T. Tran, M. L. Trinh, H. M. La, and H. T. T. Nguyen, "Trajectory Tracking Control for a Quadcopter under External Disturbances," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17620–17628, Dec. 2024, https://doi.org/10.48084/etasr.8449.

[2] M. B. Anjum *et al.*, "Maximum Power Extraction from a Standalone Photo Voltaic System via Neuro-Adaptive Arbitrary Order Sliding Mode Control Strategy with High Gain Differentiation," *Applied Sciences*, vol. 12, no. 6, Mar. 2022, Art. no. 2773, https://doi.org/10.3390/app12062773.

[3] T. L. Wong, R. R. Khan, and D. Lee, "Model linearization and H∞, controller design for a quadrotor unmanned air vehicle: Simulation study," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, Dec. 2014, pp. 1490–1495, https://doi.org/10.1109/ICARCV.2014.7064536.

[4] T.-W. Ou and Y.-C. Liu, "Adaptive Backstepping Tracking Control for Quadrotor Aerial Robots Subject to Uncertain Dynamics," in *2019 American Control Conference (ACC)*, July 2019, pp. 1–6, https://doi.org/10.23919/ACC.2019.8815025.

[5] X. Shi *et al.*, "Adaptive Fractional-Order SMC Controller Design for Unmanned Quadrotor Helicopter Under Actuator Fault and Disturbances," *IEEE Access*, vol. 8, pp. 103792–103802, 2020, https://doi.org/10.1109/ACCESS.2020.2998698.

[6] T. L. Huu, H. L. Anh, and D. T. Tran, "Applying Sliding Mode Control to a Quadrotor," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16389–16394, Oct. 2024, https://doi.org/10.48084/etasr.8026.

[7] O. Mechali, L. Xu, X. Xie, and J. Iqbal, "Theory and practice for autonomous formation flight of quadrotors via distributed robust sliding mode control protocol with fixed-time stability guarantee," *Control Engineering Practice*, vol. 123, June 2022, Art. no. 105150, https://doi.org/10.1016/j.conengprac.2022.105150.

[8] O. Mechali, L. Xu, X. Xie, and J. Iqbal, "Fixed-time nonlinear homogeneous sliding mode approach for robust tracking control of multirotor aircraft: Experimental validation," *Journal of the Franklin Institute*, vol. 359, no. 5, pp. 1971–2029, Mar. 2022, https://doi.org/10.1016/j.jfranklin.2022.01.010.

[9] J. Rivera, L. Garcia, C. Mora, J. J. Raygoza, and S. Orteg, "Super-Twisting Sliding Mode in Motion Control Systems," in *Sliding Mode Control*, A. Bartoszewicz, Ed. InTech, 2011.

[10] H. Ramirez-Rodriguez, V. Parra-Vega, A. Sanchez-Orta, and O. Garcia-Salazar, "Robust Backstepping Control Based on Integral Sliding Modes for Tracking of Quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1–4, pp. 51–66, Jan. 2014, https://doi.org/10.1007/s10846-013-9909-4.

[11] S. Ullah, A. Mehmood, Q. Khan, S. Rehman, and J. Iqbal, "Robust Integral Sliding Mode Control Design for Stability Enhancement of Under-actuated Quadcopter," *International Journal of Control, Automation and Systems*, vol. 18, no. 7, pp. 1671–1678, July 2020, https://doi.org/10.1007/s12555-019-0302-3.

[12] M. M. Madebo, C. M. Abdissa, L. N. Lemma, and D. S. Negash, "Robust Tracking Control for Quadrotor UAV With External Disturbances and Uncertainties Using Neural Network Based MRAC," *IEEE Access*, vol. 12, pp. 36183–36201, 2024, https://doi.org/10.1109/ACCESS.2024.3374894.

[13] E. A. Teklu and C. M. Abdissa, "Genetic Algorithm Tuned Super Twisting Sliding Mode Controller for Suspension of Maglev Train With Flexible Track," *IEEE Access*, vol. 11, pp. 30955–30969, 2023, https://doi.org/10.1109/ACCESS.2023.3262416.

[14] O. Mofid and S. Mobayen, "Adaptive sliding mode control for finite-time stability of quad-rotor UAVs with parametric uncertainties," *ISA*

Transactions, vol. 72, pp. 1–14, Jan. 2018, https://doi.org/10.1016/j.isatra.2017.11.010.

[15] S. P. Madruga, A. D. H. B. M. Tavares, G F. Basso, T. P. D. Nascimento, and A. Brito, "A Pso-Based Tuning Algorithm for Quadcopter Controllers," presented at the XXII Congresso Brasileiro de Automática, 2018, https://doi.org/10.20906/CPS/CBA2018-0064.

[16] O. Doukhi, A. R. Fayjie, and D. J. Lee, "Intelligent Controller Design for Quad-Rotor Stabilization in Presence of Parameter Variations," *Journal of Advanced Transportation*, vol. 2017, pp. 1–10, 2017, https://doi.org/10.1155/2017/4683912.

[17] N. B. Abera, C. M. Abdissa, and L. N. Lemma, "An improved nonsingular adaptive super twisting sliding mode controller for quadcopter," *PLOS One*, vol. 19, no. 10, Oct. 2024, Art. no. e0309098, https://doi.org/10.1371/journal.pone.0309098.

[18] C. Bensalah, N. K. M'Sirdi, and A. Naamane, "Full modelling and sliding mode control for a quadrotor UAV in visual servoing task," presented at the Proceedings of the 12th International Conference on Integrated Modeling and Analysis in Applied Control and Automation, Sept. 2020, pp. 1–6, https://doi.org/10.46354/i3m.2019.imaaca.007.

[19] P. J. Sanchez-Cuevas, V. Martín, G. Heredia, and A. Ollero, "Aerodynamic Effects in Multirotors Flying Close to Obstacles: Modelling and Mapping," in *Robot 2019: Fourth Iberian Robotics Conference*, vol. 1092, M. F. Silva, J. Luís Lima, L. P. Reis, A. Sanfeliu, and D. Tardioli, Eds. Cham: Springer International Publishing, 2020, pp. 63–74.

[20] Z. Li, X. Ma, and Y. Li, "Model-free control of a quadrotor using adaptive proportional derivative-sliding mode control and robust integral of the signum of the error," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, Sept. 2018, https://doi.org/10.1177/1729881418800885.

[21] N. El Gmili, M. Mjahed, A. El Kari, and H. Ayad, "Particle swarm optimization based proportional-derivative parameters for unmanned tilt-rotor flight control and trajectory tracking," *Automatika*, vol. 61, no. 2, pp. 189–206, Apr. 2020, https://doi.org/10.1080/00051144.2019.1698191.

[22] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding Mode Control Based on Backstepping Approach for an UAV Type-Quadrotor," *International Journal of Mechanical and Mechatronics Engineering*, vol. 1, no. 2, pp. 39–44, 2007.