IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 July 2023, accepted 6 August 2023, date of publication 18 August 2023, date of current version 23 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304760

RESEARCH ARTICLE

Multi-Optimization of Novel Conditioned Adaptive Barrier Function Integral Terminal SMC for Trajectory Tracking of a Quadcopter System

ABDULLAH MUGHEES, (Member, IEEE), AND IFTIKHAR AHMAD

School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan

Corresponding author: Iftikhar Ahmad (iftikhar.rana@seecs.edu.pk)

ABSTRACT Researchers in the domain of unmanned aerial vehicles (UAVs) have recently shown a great deal of interest in the quadcopter domain due to its wide variety of applications. In addition to their military applications, quadcopters are now widely used in the civilian sector. This paper presents optimized controllers for attitude and altitude control of a quadcopter system. A novel conditioned adaptive barrier function integral terminal sliding mode controller (CABFIT-SMC) is designed to address the trajectory tracking problem of the quadcopter, and a sliding mode control (SMC) is implemented for comparative analysis. Four optimization algorithms (i.e., Ant Colony Optimization, Artificial Bee Colony, Particle Swarm Optimization, and Genetic Algorithm) have been used to optimize the proposed control laws. The quadcopter's non-linear model is formulated through the Lagrange formalism in MATLAB ODE-45, incorporating gyroscopic moments and aerodynamic effects. A Lyapunov stability analysis is carried out to verify the system's asymptotic stability. A graphical and tabular comparative analysis is provided for all optimized control laws. Five performance indexes, including mean absolute percentage error, root mean square error, integral square error, integral absolute error, and integral time absolute error, are used to determine the best control law. The proposed optimized controllers are evaluated for performance and consistency using a 3D-helical complex trajectory. Based on the rigorous performance evaluation, it has been demonstrated that the CABFIT-SMC controller optimized with the ABC algorithm achieved the most superior results, with the lowest MAPE value of 13.245, RMSE value of 0.0043, and transient response characteristics yielding the quickest rise and settling times. The commendable performance of the ABC-optimized CABFIT-SMC further reinforces its potential as a robust controller for precise attitude, altitude, heading, and position tracking in quadcopter systems.

INDEX TERMS Lyapunov theory, optimization algorithms, optimized nonlinear controllers, quadcopter trajectory tracking, sliding mode control variants.

I. INTRODUCTION

Experts in the engineering field of unmanned aerial vehicles (UAVs) are demonstrating an escalating interest in quadcopter technology due to its extensive and diverse applications. Quadcopters initially acknowledged for their military uses, have expanded their presence into civilian sectors, solidifying their importance in the UAV industry. A quadcopter system is a drone or rotorcraft-based UAV with four fixed-propeller rotors. It can take off vertically, hover, perch, and follow acute trajectories, unlike other UAVs. Nonetheless, this multirotor system is underactuated, extremely nonlinear, and susceptible to disturbances and parametric uncertainties. Given the highly nonlinear dynamics of such systems, researchers continue to confront major difficulties in controlling multi-rotor systems despite significant breakthroughs in the multi-rotor control domain. The quadcopter changes rotor speeds to provide push and torque for maneuvering and flight [1]. The four rotors on a quadcopter system work in pairs, each rotating in opposite directions to create

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Xu.

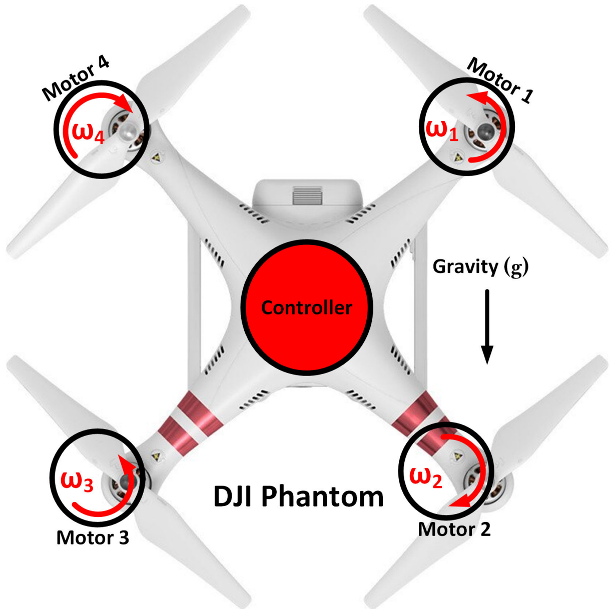

FIGURE 1. Typical DJI Phantom commercial quadcopter.

lift and stability. By varying the speed of each rotor, the quadcopter can change its altitude, pitch, and roll. A quadcopter system has six degrees of freedom (x- coordinate, y- coordinate, z-coordinate (altitude), roll, pitch, and yaw). A typical commercial quadcopter (DJI Phantom) is shown in Fig. 1.

The roll, pitch, yaw, and altitude are the four main degrees of freedom or axes of movement for a quadcopter system. Roll refers to the rotation of the quadcopter system around its longitudinal axis. This axis runs from the front to the back of the drone. When the drone rolls, one side tilts up while the other tilts down. Pitch refers to the rotation of the quadcopter system around its lateral axis. This axis runs from one side of the drone to the other. When the drone pitches, the front of the drone tilts up while the back of the drone tilts down. Yaw refers to the rotation of the quadcopter system around its vertical axis. This axis runs from the top to the bottom of the drone. When the drone yaws, it rotates to the left or right. Altitude refers to the vertical distance of the quadcopter system from the ground or a reference point. It is controlled by adjusting the throttle or speed of the motors that control the lift. Increasing the speed will cause the drone to ascend, while decreasing the speed will cause it to descend. These moments of a quadcopter are depicted in Fig. 2. The rotor speed is proportional to the thickness of the boundary circles.

The quadcopter system has recently gained prominence due to its versatility and task efficiency. Quadcopters substituted major surveillance activities during COVID-19, such as food delivery, medical rescue, spraying antiseptics, law enforcement missions, monitoring of disaster-prone areas (i.e., earthquakes, landslides, forest fires, and flooding), etc. [2]. In addition to that, quadcopters are also utilized for agricultural or civil applications like aerial photography, videography, crop monitoring, and structural assessment [3].

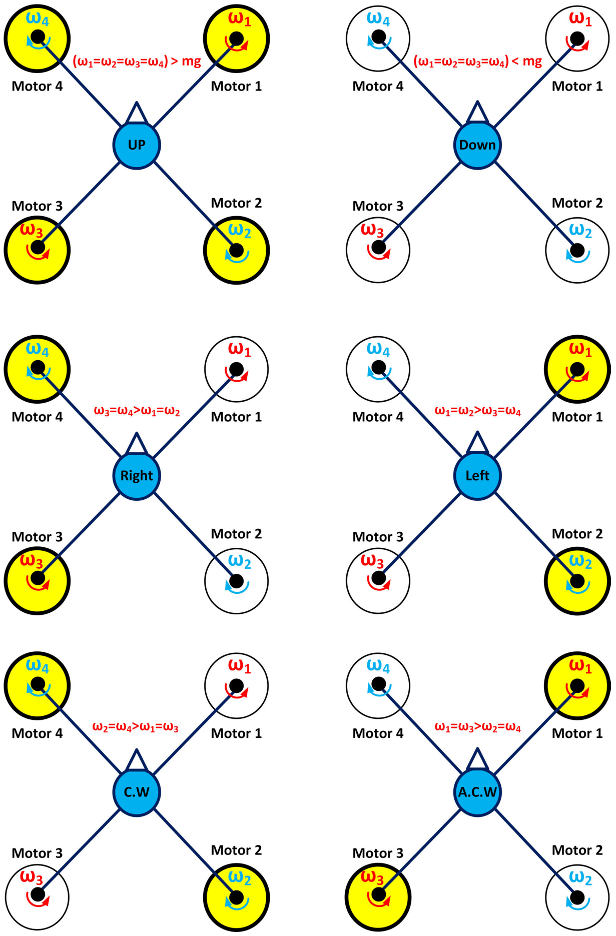

FIGURE 2. Motions of the quadcopter with respect to the angular speed of the motors.

However, developing a quadcopter system that can accurately track the desired trajectory to complete the mission successfully is an extremely complex task due to its nonlinear dynamics. Trajectory tracking is the ability of a quadcopter system to accurately follow a predefined flight path or trajectory. It is a critical feature for quadcopter systems, enabling them to perform complex tasks with high precision, stability, efficiency, and safety. System-parameterized uncertainties and external disruptions affect the trajectory-tracking performance of the quadcopter. To modify this under-actuated and highly nonlinear dynamic system, researchers are working on attaining a stable and advanced quadcopter controller for accurate trajectory tracking [4].

Some methods have been applied by researchers in controlling the trajectory tracking of a quadcopter based on its position, altitude, and attitude. PID control is the most commonly used reference control technique to address the trajectory tracking problem of quadcopter systems [5]. Due to the nonlinearities in the quadcopter system, the performance of the PID controller significantly degrades when the system is under the influence of external disturbances. A nonlinear technique known as the sliding mode control (SMC) is known for its robustness to uncertainties and disturbances in the system. It works by designing a sliding surface that the system should track and adjusting the control input to keep the system on the sliding surface.

Adaptive control is also another nonlinear control method that adjusts the control signal based on the changing dynamics of the quadcopter system (i.e., gyroscopic moments and aerodynamic effects). It works by estimating the system parameters and adjusting the control input accordingly [6]. Furthermore, machine learning techniques are used for trajectory tracking of the quadcopter system. Among them, the most popular are reinforcement learning (RL) and model predictive control (MPC). MPC employs a predictive system model to determine the optimal control input that minimizes a cost function [7]. It is used to track complex trajectories by predicting the future state of the quadcopter and calculating the control input that brings it to the desired trajectory. On the other hand, RL works by learning a policy that maps the state of the system to action, using a reward signal to guide the learning process [8]. RL is used to track complex trajectories of the quadcopter system by learning a policy that maximizes the reward signal. In addition to that, other methods have been reported in the literature for trajectory tracking of the quadcopter system including nonlinear control techniques, such as backstepping control [4], fuzzy PID control [9], neural network-based MPC [10], optimal control [11], Lyapunov based adaptive control [12] and nonlinear MPC [13], etc.). Since quadcopter is a highly nonlinear and complex system, nonlinear control methods like backstepping control [14], hybrid sliding mode control [15], neural network-based feedback linearization [16], and advanced variants of SMC have shown effective results throughout the literature.

Sliding mode controller has been one of the best candidates in literature to provide robust and efficient control for tracking the trajectory of a quadcopter [17], [18], [19], [20], [21], [22], [23]. It is designed to be robust to uncertainties and disturbances in the system [24]. It gives a finite time convergence rate that quickly brings the desired state (x_0) of the quadcopter to its equilibrium or reference point. However, SMC is known for producing chattering in the control signal, a high-frequency oscillation around the desired value. While chattering can be a disadvantage in some systems, it can be an advantage in quadcopters under certain constraints, as it can help to suppress the effects of model uncertainties and external disturbances. The chattering effect of the SMC can be observed in Fig. 3.

Generally, the chattering generated by high-frequency switching of discontinuous terms due to the switching function imperfection is one of the significant drawbacks of SMC [15]. Some extended SMC-based methods have been proposed in the literature to control this issue. These extended versions of SMC have been applied by researchers for controlling the quadcopter system to alleviate chattering because of its quick convergence and robustness to parameter fluctuations and disturbances [25], [26]. One of the most widely used variants of SMC is the terminal sliding mode control (TSMC) [27]. In [28], a TSMC-based dynamic flight controller has been proposed to control the quadcopter system with finite-time convergence for the states. Another variant, improved adaptive SMC (ASMC), has been employed in [29], which adjusts the sliding surface and control law based on system uncertainties and disturbances by generating an adaptive switching gain. Various advanced hybrid variants of SMC have been used in the literature to minimize the chattering effect and enhance the system's performance. For example, iterative learning SMC [30], fuzzy TSMC [31], modified ASMC [32], integral TSMC [33], U-model enhanced double SMC [34], neuro-adaptive integral TSMC [35], backstepping SMC [36], etc. These approaches reduce the oscillations, provide smooth control torque, and quickly converge the errors that may occur at any given time. Moreover, asymptotic stability and finite time convergence in SMC and its variants are guaranteed by Lyapunov's theory.

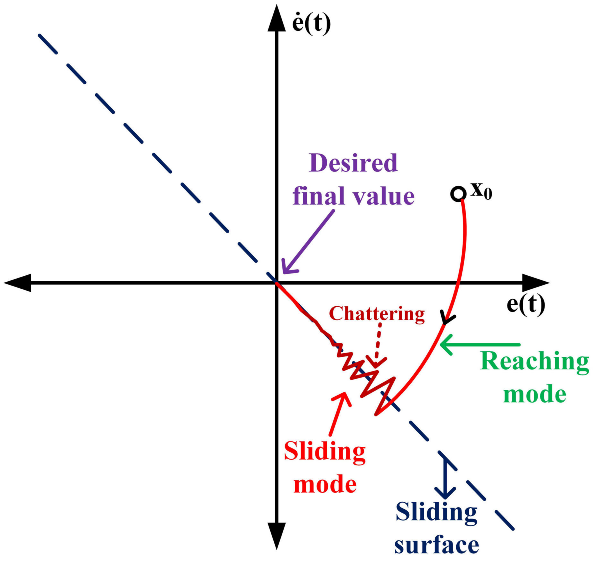

FIGURE 3. Phase diagram illustrating the controller's reaching mode and chattering effect.

Even after the evolution of advanced techniques, the performance of a controller is still affected by various factors attributed to the tuning of a control parameter, time delays, model inaccuracy, and external disturbances. To improve the performance of a controller, optimization algorithms have been employed. These algorithms are essential because they can optimize the tuning parameters of a controller to achieve better system performance. An optimized controller can achieve higher levels of accuracy, robustness, stability, adaptability, and efficiency compared to a non-optimized controller. Several techniques have been reported in the literature that utilize optimization algorithms with controllers for controlling the trajectory tracking of a quadcopter. In [37], indoor trajectory tracking control has been tuned for a quadcopter by using various optimization algorithms (i.e., artificial bee colony (ABC), genetic algorithm (GA), and particle swarm optimization (PSO)) to develop a hybrid fuzzy logic (FL) controller and their performance is compared. The results indicated that the PSO algorithm had performed best with the FL controller for tracking the trajectory of a quadcopter.

Similarly, in [38], eight metaheuristic algorithms (whale optimization algorithm, slime-mould algorithm, marine predators algorithm, linear space and time complexity parallel adaptive differential evolution using multiple archives, linear success-history based adaptive differential evolution, hunger games search, grey wolf optimization, and particle swarm optimization) are reported with PD and PID controllers, and to evaluate their performances the root-mean-square-error (RMSE) is used. The performance of the HGS algorithm surpassed all other optimization algorithms for the trajectory tracking of the quadcopter for PD and PID controller. In [39], a comparative analysis is performed for various optimization algorithms (i.e., genetic algorithm (GA), crow search algorithm (CSA), PSO, and biogeography-based optimization (BBO)) that were used to tune the gains of the PID controller for controlling the quadcopter trajectory. Similarly, in [40], ant colony optimization is used to optimize the backstepping controller for addressing the trajectory tracking problem of the quadcopter. In [41], a nonlinear optimized SMC model via PSO algorithm is designed based on disturbance observer to reject mutational disturbances and crosswind effects better.

As optimization algorithms can handle the nonlinear and time-varying dynamics of the quadcopter system, they play a crucial role in real-time applications. Several research studies have utilized optimization algorithms with and without control laws for trajectory tracking of quadcopter systems. Some of these optimization techniques for trajectory tracking of the quadcopter include cooperative PSO with MPC [42], PSO [43], [44], neural network and FL with PID controller [45], GA [46], PSO and cuckoo search-based approach [47], deep reinforcement learning (DRL) based adaptive controller [48], a hybrid Harris hawk optimization with GWO for path planning [49], nonlinear MPC [50], [51], [52], [53], Learning-based parametrized MPC [54], MPC using linear parameter varying [55], etc.).

Overall, the literature demonstrates the importance and effectiveness of optimization algorithms in the trajectory tracking of quadcopter systems. The use of optimization algorithms allows for optimising the four control inputs of a quadcopter system, which is critical for achieving accurate and stable trajectory tracking. The use of sliding mode control for trajectory tracking of quadcopter systems has been extensively studied in the literature. However, there is a research gap in the comparative analysis of various optimization algorithms used to optimize the performance of the advanced variants of SMC. A graphical and tabular comparative analysis of the optimization of SMC variants using various optimization algorithms can provide insights into the effectiveness of different optimization techniques in improving the performance of the controller. This analysis can help researchers and practitioners in the field of quadcopter control to choose the most appropriate optimized control law for their specific application. Furthermore, this analysis can also identify areas where further research is needed to improve the performance of the sliding mode controller for trajectory tracking of quadcopter systems.

In the dynamic modeling of the quadcopter, the literature primarily disregards the aerodynamic effect, air disturbance, and gyroscopic moment. In this study, the dynamic behavior of a quadrotor is therefore modeled using the Lagrange formalism. The mathematical model incorporates gyroscopic and aerodynamic factors to depict the entire nonlinear dynamics of a quadcopter. A state-space representation of the model is formed that includes the parameters affecting the dynamics of the system, such as gyroscopic effects, drag forces (along the x-y-z axes), aerodynamic torque friction, and higher-order nonholonomic constraints [37].

In this research, the altitude (z), heading (yaw), attitude (roll and pitch), and position (x-y) control laws are designed using the novel conditioned adaptive barrier function integral terminal sliding mode control. A condition function is applied to the SMC to respond faster and make the system robust under time-varying or uncertain parametric variations. The barrier function-based (BF) SMC is used for limiting the state within the constrained region to prevent it from reaching an unstable region. Then, BF integral SMC (BFI-SMC) is implemented to reduce the steady state-error that may occur when the reaching law input is applied to the state. The terminal sliding mode surface is designed to guarantee that the system state converges to the desired trajectory in a finite time. The chattering effect is suppressed to a negligible value using the combination of condition and adaptive barrier function with the hybrid variant (integral-terminal) of SMC. Furthermore, using multiple optimization algorithms, the performance of the proposed control law is optimized, compared, and evaluated to establish the optimal control law for the altitude, heading, and attitude motion of the quadcopter. A hardware-in-the-loop (HIL) test using the C2000 Delfino MCU F28379D launchpad is conducted to validate the performance of the developed control law. The main contributions of this research are as follows:

- Mathematical modeling using the Lagrange formalism, containing the rotational (gyroscopic moment) and translation (aerodynamic frictional moment) dynamics (Euler-angle dynamics), is presented in a more realistic and complete new state-space representation form for control law synthesis,
- Design of conditioned adaptive barrier function integral terminal sliding mode control for trajectory tracking of a quadcopter system,
- Implementation of sliding mode control for comparative analysis with CABFIT-SMC,
- Lyapunov stability analysis is performed to prove the asymptotic stability and finite-time convergence of the error signal to zero,
- Use of four optimization algorithms (Ant Colony Optimization, Artificial Bee Colony, Particle Swarm Optimization, and Genetic Algorithm) to optimize the control laws,

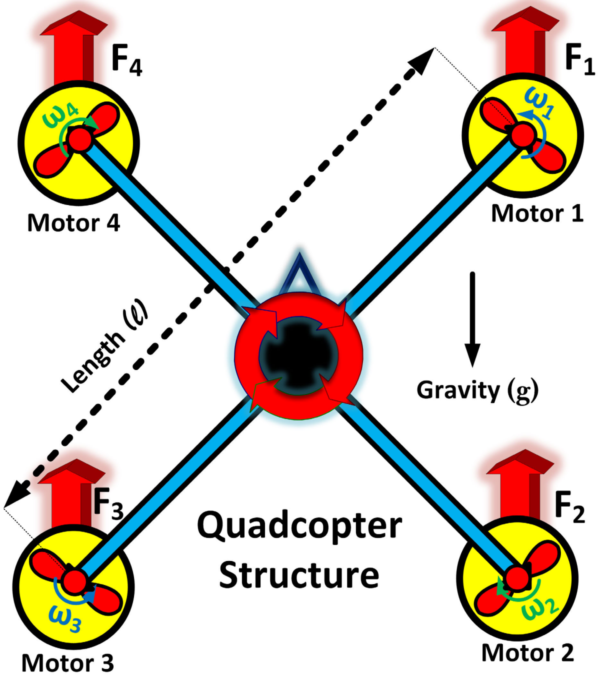

FIGURE 4. Typical configuration of a quadcopter system.

- 3D-helical complex trajectory problem is employed to analyze the performance and consistency of the optimized controllers,
- Use of MATLAB ODE-45 environment for simulated results,
- Conducted graphical and tabular comparative analysis of all the optimized controllers,
- Use of five performance indexes (mean absolute percentage error, root mean square error, integral square error, integral absolute error, and integral time absolute error) to determine the best control law.

The remaining research paper is arranged in the following outline: Section II describes the mathematical model of the quadcopter system. The novel proposed control law is designed in Section III. Optimization algorithms are described in Section IV. The results are explained in Section V. In section VI, a graphical and tabular comparative analysis of optimized control laws is provided. Section VII concludes the paper with conclusions and future works.

II. MATHEMATICAL MODELING

This section establishes a complete dynamical model of the quadcopter UAV system, including rotational and translational dynamics, using the Lagrange formalism. A quadcopter's four intersecting rotors generate upward forces (F_1, F_2, F_3, and F_4) through the propellers, as shown in Fig. 4.
where, ω is the angular speed of the rotor in (s^{-1}), l is the arm length in (m), and g is the gravitational pull acting on the quadcopter in (ms^{-2}).

As shown in Fig. 4, the two pairs of propellers attached to the four motors rotate in opposite directions (i.e., (2)-(4) rotate clockwise (CW), and (1)-(3) rotate counter-clockwise (ACW)) to have a zero net rotation. A minor difference in the angular speeds of the motors causes rotational or translational motion. Hence, by adjusting the angular speed of the motors, it is possible to control the motion and lift force of the quadcopter system. The vertical motion of the quadcopter is controlled by adjusting the rotational speed of each propeller in proportion. Adjusting the speed of the propeller combination (1)-(2) and (3)-(4) produces lateral motion coupled with roll motion in the quadcopter. Similarly, adjusting the speed of the propeller combination (1)-(4) and (2)-(3) results in pitch and lateral motions. However, the quadcopter's yaw rotation is more subtle. It is generated by producing a net difference between the clockwise and counterclockwise rotations of the two pairs of propellers.

The quadcopter has a 6-DOF in free space. The possible movements for a quadcopter (i.e., C.W. and A.C.W. directions, take-off (upward), landing (downward), right, and left). In addition, quadcopters can execute sophisticated maneuvers such as loops, rolls, and flips because of their ability to independently control the velocity of each rotor. The quadcopter system is typically modeled as a nonlinear, underactuated system because it has more degrees of freedom than the control inputs. To accurately model the quadcopter, it is necessary to consider factors such as aerodynamics, motor dynamics, and sensor noise. The following assumptions are made for modeling the quadcopter system:

- The body frame origin (o') and the center of gravity coincide,
- Thrust and drag are proportional to the square of the propeller's speed,
- The propellers are rigid,
- The structure of the quadcopter is symmetrical and rigid.

A. QUADCOPTER KINEMATICS

The kinematics of a quadcopter can be divided into two frames: the earth-fixed or reference frame ($E = x_e, ye, z_e$) and the body-fixed frame ($B = x_b, y_b, z_b$). Using these coordinates, one can derive the equations of motion. Fig. 5 depicts the two frames representing the kinematics and forces acting on the quadcopter.
where the external torques induced by the motors resulting in the rotation along the x, y, and z directions are denoted by τ_x, τ_y, and τ_z, respectively. ω_1, ω_2, ω_3, and ω_4 are the angular velocities of the corresponding rotors. Each motor provides an upward (z-*axis* direction) thrust represented by F_1, F_2, F_3, and F_4. The position of the center of mass of the quadcopter with respect to the reference frame is denoted by x, y, and z. Similarly, the roll (ϕ), pitch (θ), and yaw (ψ) angles represent the attitude and heading of the quadcopter with respect to the reference frame. Each rotor on the quadcopter generates an upward thrust that is proportional to the square of angular speed, which can be represented as:

$$F = k\omega^2 \tag{1}$$

where F is the upward thrust in Newtons (N) and k is the lift/thrust constant measured in Ns^2. The rotation along x, y, and z directions due to the external torques induced by the motors are represented by (2) - (4), respectively:

$$\tau_x = lk(\omega_4^2 - \omega_2^2) \neq 0, \quad (\forall \quad k\omega_2^2 \neq k\omega_4^2) \tag{2}$$

$$\tau_y = lk(\omega_3^2 - \omega_1^2) \neq 0, \quad (\forall \quad k\omega_1^2 \neq k\omega_3^2) \tag{3}$$

$$\begin{aligned} \tau_z &= b[(\omega_1^2 + \omega_3^2) - (\omega_2^2 + \omega_4^2) \neq 0] \quad , \\ &\times (\forall \quad (\omega_1^2 + \omega_3^2) \neq (\omega_2^2 + \omega_4^2)) \end{aligned} \tag{4}$$

where b is the drag constant. The four motors induce a force F_z in the z-axis direction. Since these motors are mounted horizontally on the quadcopter, it limits them from producing a direct force in the x-axis or y-axis direction (i.e., F_x and F_y). Hence, the rotational torques (τ_x and τ_y) causes the quadcopter system to produce a force in the x-axis and y-axis directions.

The quadcopter system allows control over four variables (ω_1, ω_2, ω_3, and ω_4) through their corresponding inputs (u_1, u_2, u_3, and u_4). However, the overall motion of the quadcopter from one point to another requires the control of six variables (x, y, z, ϕ, θ, and ψ). Hence, controlling an under-actuated quadcopter system using traditional control laws becomes challenging.

B. QUADCOPTER's EQUATIONS OF MOTION

The Euler angles (ϕ, θ, and ψ) represent the quadrotor's orientation (i.e., pitch, roll, and yaw). The $z - y - x$ rotational matrix transforms the body-fixed frame into the inertial frame. We assume the angle bounds shown in (5) to avoid system singularities.

$$\frac{-\pi}{2} < \varphi < \frac{\pi}{2}, \quad \frac{-\pi}{2} < \theta < \frac{\pi}{2}, \quad -\pi < \psi < \pi \tag{5}$$

The angular speed and rotational matrix of the body fixed frame are represented by (6) and (7), respectively:

$$\omega_b = \begin{pmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{pmatrix} \tag{6}$$

$$\omega_b = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{7}$$

where ω_b is the angular speed of the body fixed frame measured in rad/s. The Euler rotation about $z - y - x$ or R_{xyz} is described by (8):

$$\boldsymbol{R}_{xyz} = \boldsymbol{R}_{(z,\psi)}\boldsymbol{R}_{(y,\theta)}\boldsymbol{R}_{(x,\varphi)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{8}$$

where:

- $a_{11} = cos\psi cos\theta$
- $a_{12} = sin\varphi sin\theta cos\psi - sin\psi cos\theta$
- $a_{13} = cos\varphi sin\theta cos\psi + sin\psi sin\varphi$
- $a_{21} = sin\psi cos\theta$
- $a_{22} = sin\varphi sin\theta sin\psi + cos\psi cos\varphi$

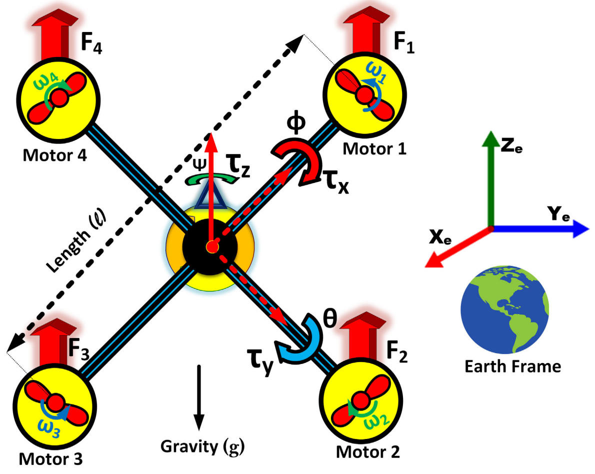

FIGURE 5. Inertial and body frames of a typical quadcopter.

- $a_{23} = cos\varphi sin\theta sin\psi - sin\varphi cos\psi$
- $a_{31} = -sin\theta$
- $a_{32} = sin\varphi cos\theta$
- $a_{33} = cos\varphi cos\theta$

The position (x, y, z) and velocities $(\dot{x}, \dot{y}, \dot{z})$ of the quadcopter can be used to determine its total kinetic energy, as:

$$T_{K.E} = \frac{1}{2}m(\dot{x}^2\dot{y}^2\dot{z}^2) + \frac{1}{2}(I_x\omega_{bx}^2 + I_y\omega_{by}^2 + I_z\omega_{bz}^2) \tag{9}$$

where I_x, I_y, & I_z are inertial constants over the principle axis (i.e., x, y, z). Similarly, the potential energy of the system due to the gravitational pull is represented as:

$$T_{P.E} = mgz \tag{10}$$

where g is the gravitational acceleration acting on the quadcopter in (m/s^2), m is the mass of the quadcopter in (Kg), and the altitude z of the quadcopter above sea-level in the z-axis direction is measured in meters (m). The Lagrangian $(\mathcal{L})$ can be determined using the kinetic and potential energies of the system as:

$$\mathcal{L} = T_{K.E} - T_{P.E} \tag{11}$$

Equations of motions can be determined by evaluating the Euler-Lagrangian equation as:

$$\frac{d(\frac{\partial(\mathcal{L})}{\partial(\dot{q}j)})}{dt} - \frac{\partial(\mathcal{L})}{d(\dot{q}j)} = \mathcal{T}_j, \qquad j = 1, 2, \ldots, n \tag{12}$$

where $\mathcal{T}_j$ represents the external forces and torques acting on the quadcopter. n is any real number. $\dot{q}$ denotes the coordinates of the quadcopter (i.e., $x, y, z, \phi, \theta, \psi$) and j is the numeric number for the corresponding external force. The external forces and torques can be divided into two parts as:

$$\mathcal{T} = \begin{pmatrix} F_{ext} \\ \tau_{ext} \end{pmatrix} \tag{13}$$

where F_{ext} and τ_{ext} depict the external forces acting on the quadcopter along x, y, z and ϕ, θ, ψ axes, respectively. The total rotational external force τ_{ext} can be defined as:

$$\tau_{ext} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} kl\left(\omega_4^2 - w_2^2\right) \\ kl\left(w_3^2 - w_1^2\right) \\ b\left(\omega_1^2 - w_2^2 + w_3^2 - w_4^2\right) \end{bmatrix} \tag{14}$$

The F_{ext} exists only when the quadcopter is perfectly aligned with the z-axis. Upon any rotational motion of the quadcopter, the F_{ext} is resolved along the x, y, z directions. It can be described as:

$$\begin{aligned} F_{ext} &= (RotationalMatrix) * (Thrust) - Drag \\ &= R \begin{bmatrix} 0 \\ 0 \\ k\left(\omega_1^2 + w_2^2 + \omega_3^2 + \omega_4^2\right) \end{bmatrix} \\ &\quad - \begin{bmatrix} A_x & q_x \\ A_y & q_y \\ A_z & q_z \end{bmatrix} \end{aligned} \tag{15}$$

where A_x, A_y, A_z and q_x, q_y, q_z are the damping constants and velocities in the x, y, z directions, respectively. Hence, (14) and (15) can be used to compute the total external forces and torques ($\mathcal{T}$) acting on the quadcopter. Now, the Lagrange partial differential equation (8) can be solved using MATLAB to evaluate the dynamic model of the quadcopter as:

$$\begin{aligned} \ddot{\phi} &= \frac{1}{I_x}\left\{\dot{\theta}\dot{\psi}\left(I_y - I_z\right) - k_{fax}\dot{\phi}^2 - J_r\bar{\Omega}\dot{\theta} + dU_2\right\} \\ \ddot{\theta} &= \frac{1}{I_y}\left\{\dot{\phi}\dot{\psi}\left(I_z - I_x\right) - k_{fay}\dot{\theta}^2 + J_r\bar{\Omega}\dot{\phi}^2 + dU_3\right\} \\ \ddot{\psi} &= \frac{1}{I_z}\left\{\dot{\theta}\dot{\phi}\left(I_x - I_y\right) - k_{faz}\dot{\psi}^2 + k_d U_4\right\} \\ \ddot{x} &= \frac{1}{m}\left[(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)U_1 - k_{ftx}\dot{x}\right] \\ \ddot{y} &= \frac{1}{m}\left[(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)U_1 - k_{fty}\dot{y}\right] \\ \ddot{z} &= \frac{1}{m}\left[(\cos\phi\cos\theta)U_1 - k_{ftz}\dot{z}\right] - g \end{aligned} \tag{16}$$

C. ROTOR DYNAMICS

A rotor is a section of a D.C. motor that drives a propeller through a reducer. The dynamic equation governing the DC-motor is given as:

$$\begin{cases} V = ri + L\dfrac{di}{dt} + k_e\omega \\ k_m i = J_r\dfrac{d\omega}{dt} + C_s + k_r\omega^2 \end{cases} \tag{17}$$

where V is the input voltage to the motor, i is the current passing through the internal resistance r of the motor. k_r, k_m, k_e are the load, mechanical, and electrical torque constants, respectively. The inertia of the rotor and solid friction constant are denoted by J_r and C_s, respectively. Then, the following design is selected for defining the rotor dynamics:

$$\dot{\omega} = bq - \zeta_0 - \zeta_1\omega - \zeta_2\omega^2, \qquad i \mid\in [1, 4] \tag{18}$$

where;

$$\zeta_0 = \frac{C_s}{J_r}, \zeta_1 = \frac{k_e k_m}{rJ_r}, \zeta_2 = \frac{k_r}{J_r} \text{ and } b = \frac{k_m}{rJ_r}$$

D. STATE SPACE REPRESENTATION OF THE QUADCOPTER

The quadcopter dynamics given by (16) can be expressed in state-space form as:

$$X = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}]^{\mathrm{T}} \tag{19}$$

Hence, using (16) and (19), the state space representation can be obtained as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= g_1 x_4 x_6 + g_2 x_2^2 + g_3\bar{\Omega}x_4 + h_1 U_2 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= g_4 x_2 x_6 + g_5 x_4^2 + g_6\bar{\Omega}x_2 + h_2 U_3 \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= g_7 x_2 x_4 + g_8 x_6^2 + h_3 U_4 \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= g_9 x_8 + U_x\frac{U_1}{m} \\ \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= g_{10}x_{10} + U_y\frac{U_1}{m} \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= g_{11}x_{12} + \cos(x_1)\cos(x_3)\frac{U_1}{m} - g \\ \dot{x}_{12} &= g_{11}x_{12} + u_z\frac{U_1}{m} - g \end{aligned} \tag{20}$$

where $g_1 \ldots g_{11}$ are the constants defining parameters of the quadcopter system. The states x_1, x_3, x_5, x_7, x_9, and x_{11} correspond to the pitch, roll, yaw, x, y, and z variables of the quadcopter system, respectively. Similarly, $x_2, x_4, x_6, x_8, x_{10}$, and x_{12} correspond to the derivative of pitch, roll, yaw, x, y, and z variables of the quadcopter system, respectively.

III. CONTROLLER DESIGN

The control problem addressed in this article involves tracking complex trajectories and waypoints in the presence of parameter uncertainties using a novel nonlinear control law. Various optimization algorithms (i.e., ACO, ABC, PSO, and GA) are used to optimize the proposed controllers (i.e., CABFIT-SMC and SMC). The sections that follow describe the controller design and its optimization in detail.

A. SLIDING MODE CONTROL

Sliding mode control is a nonlinear controller that employs a sliding surface for controlling the system's behaviour. The sliding surface can be defined as a function of the system state variables such that the function's derivative with respect to time is zero when the system state is on the sliding surface. This ensures that the system state remains on the sliding surface once it reaches it. The sliding surface ensures that

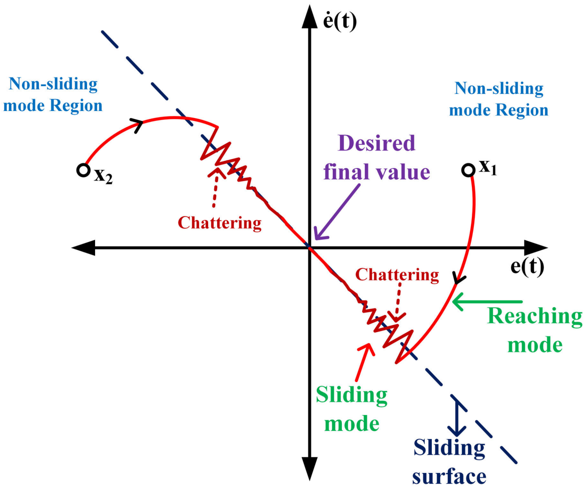

FIGURE 6. Phase diagram depicting reaching mode and chattering effect of the sliding mode controller.

the system remains on a predefined trajectory, even in the presence of disturbances or uncertainties. The dynamics of the system on the sliding surface are completely determined by a discontinuous control law. This control law ensures that the system state moves towards the sliding surface and stays on it. This makes sliding mode control a robust and effective control strategy for various applications. Moreover, it offers robustness to time-varying system parameters, rapid and accurate response to disturbances, simplicity in design and implementation, and superior control performance. Also, SMC can accomplish zero steady-state error with finite time convergence and high precision control. The primary disadvantages of SMC are design complexity, high sensitivity to measurement noise and modelling errors, higher-order sliding surfaces, and chattering. Chattering can cause excessive wear and tear on mechanical systems, leading to premature failure. To mitigate these issues, researchers have proposed various modifications to the SMC, such as adding adaptive tuning, barrier functions, terminal conditions, etc. Figure 6 illustrates the chattering effect of the SMC as the state approaches the sliding surface.

The figure indicates that after some chattering, due to the action of the control input, the state (x_o) reaches the equilibrium point after converging towards the sliding surface. After defining the sliding surface, a sliding mode control law must be chosen to bring the state towards the sliding surface. This research offers a conditioned adaptive barrier function integral terminal sliding mode control law to address the trajectory tracking problem of the quadcopter system. The conditioning function is introduced and chosen as a positive-definite function with a well-defined minimum. It is used to modify the sliding mode surface to increase the convergence rate and minimize the chattering effect. The adaptive control enables the control input to adapt to real-time changes in the quadcopter's dynamics or operating conditions. The barrier function imposes safety constraints on the quadcopter, ensuring it operates within a safe operating region. It has a negative gradient that points toward the boundary of the safe operating region. The integral term accumulates the error over time and integrates it into the control signal to remove the steady-state error. The terminal condition guarantees that the quadcopter state converges to the desired trajectory in a finite time.

B. CONDTIONED ADAPTIVE BARRIER FUNCTION INTEGRAL TERMINAL-SMC

The states representing the pitch angle of the quadcopter system are:

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= g_3\bar{\Omega}x_4 + g_2x_2^2 + g_1x_4x_6 + h_1U_2 \qquad (21)\end{aligned}$$

The tracking error must go to zero to ensure convergence of states towards the equilibrium/reference point. The errors can be defined as:

$$\begin{aligned}e_1 &= x_1 - x_{1ref}\\ e_2 &= x_2 - x_{2ref} \qquad (22)\end{aligned}$$

where x_{1ref} and x_{2ref} define the reference pitch angle and its derivative, respectively. An integral of the error term is considered in the design process to eliminate the steady-state error in the control response, which can be defined as:

$$\begin{aligned}e_{1\phi} &= \int(x_1 - x_{1ref})dt\\ e_{2\phi} &= \int(x_2 - x_{2ref})dt \qquad (23)\end{aligned}$$

A single sliding surface can be used for the single input (U_2) in the pitch angle state of the quadcopter system. The integral terminal sliding mode surface, which integrates the previous and instantaneous error of the pitch angle, is defined as:

$$S_\phi = e_1 + q_1(e_{1\phi})^{a/b} + e_2 + q_2(e_{2\phi})^{a/b} \qquad (24)$$

where a and b are positive numbers such that $1 < \frac{a}{b} < 2$. q_1 and q_2 are the tuning parameters of the sliding surface. Various optimization algorithms can be employed to determine the optimal value of these tuning parameters. The rate of change of the sliding surface, which must go to zero, can be defined as:

$$\begin{aligned}\dot{S}_\phi = (x_2 - \dot{x}_{1ref}) &+ q_1(\frac{a}{b})(x_1 - x_{1ref})(e_{1\phi})^{\frac{a}{b}-1}\\ &+ g_3\bar{\Omega}x_4 + g_2x_2^2 + g_1x_4x_6 + h_1U_2 - \dot{x}_{2ref}\\ &+ q_2(\frac{a}{b})(x_2 - x_{2ref})(e_{2\phi})^{\frac{a}{b}-1} \qquad (25)\end{aligned}$$

To perform the Lyapunov stability analysis, an appropriate Lyapunov candidate function must be chosen to ensure that the function accurately represents the energy or potential of the system and can be used to prove its stability. The conditions for choosing a Lyapunov candidate function are as follows:

- Continuously differentiable: the function must such that it is continuously differentiable over the entire domain of the system.

$$V(x) \in C^1$$

where the space for continuously differentiable functions is denoted C^1.

- Positive definite: it ensures that the function is a valid measure of the energy or potential of the system.

$$V(0) = 0$$
$$V(x) > \forall x \neq 0$$

- Radially unbounded: it ensures that the candidate function can capture the behavior of the system in the entire state space.

$$\lim \|x\| \to \infty \mathrm{V(x)} = \infty$$

where the Euclidean norm of the state vector x is denoted by $||x||$.

- Time derivative negative definite: it ensures that the energy or potential of the system decreases over time, indicating stability.
- Invariant under system dynamics: it ensures that the candidate function remains constant over time and can be used to prove the asymptotic stability of the system.

$$\partial \mathrm{V}/\partial \mathrm{t} = (\partial \mathrm{V}/\partial \mathrm{x})\mathrm{f(x)} = 0 \forall x$$

where, $f(x)$ denotes the dynamics of the system and x indicates the trajectory of the system.

Considering these conditions, the following Lyapunov candidate function is chosen for stability analysis of the system:

$$V_i = \frac{1}{2}S_i^2, \quad i = \phi, \theta, \psi, x, y, \&z \tag{26}$$

If V's definite derivative is negative, it indicates that the system's energy is decreasing. This will causeÂ the states to move toward the equilibrium/reference point. Hence, we take the time of V as:

$$\begin{aligned}\dot{V}_\phi = S_\phi \Big[&(x_2 - \dot{x}_{1ref}) + q_1(\frac{a}{b})(x_1 - x_{1ref})(e_{1\phi})^{\frac{a}{b}-1} \\ &+ g_3\bar{\Omega}x_4 + g_2x_2^2 + g_1x_4x_6 + h_1U_2 - \dot{x}_{2ref} \\ &+ q_2(\frac{a}{b})(x_2 - x_{2ref})(e_{2\phi})^{\frac{a}{b}-1}\Big]\end{aligned} \tag{27}$$

A bounded control input must be provided to bring the state from the non-sliding mode region to the sliding region/surface. This bounded control input is defined by a reaching law, which is a smooth function that is continuously differentiable and has a positive definite Hessian matrix. This function ensures that the control input is continuous and that the system state converges to the sliding surface in a finite amount of time. The reaching law for the proposed CABFIT-SMC is defined as:

$$U_{CABFIT_{SMC}} = -\zeta |S|^\rho sign(\frac{S}{\varrho}) - f_i \tag{28}$$

where ϱ and ρ are the tuning parameters of CABFIT-SMC. To minimize the chattering effect and increase the convergence rate, a condition is imposed by the f_i function that can be determined by integrating the following equation:

$$\dot{f}_i = m_0 \operatorname{sign}\left(f_i - U_{i_{sat}}\right) \tag{29}$$

where m_0 is the positive tuning parameter and $U_{i_{sat}}$ depicts the bounded saturation function within the limit $\pm R$. R is a constant value that can be tuned. The term $U_{i_{sat}}$ can be defined as:

$$U_{i_{sat}} = \begin{cases} U_i & |U_i| \leq R \\ R \operatorname{sign}(U_i) & |U_i| > R \end{cases} \tag{30}$$

Utilizing the condition function f_i and bounded saturation function $U_{i_{sat}}$, the chattering effect is effectively minimized, leading to smoother control input signals. This results in improved convergence rates and robustness in the presence of uncertainties or time-varying parameters. The value of ρ in the literature is often assigned between 0 and 1. As the state approaches the sliding surface, ϱ further dampens the chattering. *sign* is the signum function. The gain of the reaching law, denoted by ζ, with adaptive barrier function can be described as:

$$\zeta = \begin{cases} \dfrac{|S|}{\epsilon - |S|} & |S| < \epsilon \\ \zeta\ |S|^\rho & |S| > \epsilon \end{cases} \tag{31}$$

The constraints set on the signum function are:

$$\operatorname{sign}(S) = \begin{cases} \dfrac{S}{|S|} & \text{if} \quad S < 0 \\ 0 & \text{if} \quad S = 0 \\ \dfrac{S}{S} & \text{if} \quad S > 0 \end{cases} \tag{32}$$

To ensure the Lyapunov stability criterion is satisfied, we apply the reaching law constraints to (27), which needs $\dot{V}_i \leq 0$ ($\forall$ $S_i \neq 0$), we obtain:

$$\begin{aligned}-\zeta |S_\phi|^\rho sign(\frac{S_\phi}{\varrho}) - f_i = (x_2 - \dot{x}_{1ref}) + q_1(\frac{a}{b})e_1(e_{1\phi})^{\frac{a}{b}-1} \\ + g_3\bar{\Omega}x_4 + g_2x_2^2 + g_1x_4x_6 \\ + h_1U_2 - \dot{x}_{2ref} + q_2(\frac{a}{b})e_2(e_{2\phi})^{\frac{a}{b}-1}\end{aligned} \tag{33}$$

The rate of change of Lyapunov function by considering the conditioned adaptive barrier function properties can be defined as:

$$\dot{V}_\phi = -S_\phi \zeta |S_\phi|^\rho sign(\frac{S_\phi}{\varrho}) - f_\phi \leq 0 \tag{34}$$

Lyapunov stability analysis verifies that the proposed controller is stable, ensuring that errors converge to zero in a finite amount of time and that the system is asymptotically stable. The conditioned adaptive barrier function integral terminal sliding mode control law for the pitch angle of the quadcopter system may be found by solving (33) for U_2, as:

$$\begin{aligned} U_2 = \frac{1}{h_1}\Big[&-\zeta|S_\phi|^\rho sign(\frac{S_\phi}{\varrho}) - f_\phi - (x_2 - \dot{x}_{1ref}) \\ &- q_1(\frac{a}{b})e_1(e_{1\phi})^{\frac{a}{b}-1} - g_3\bar{\Omega}x_4 - g_2x_2^2 - g_1x_4x_6 \\ &+ \dot{x}_{2ref} - q_2(\frac{a}{b})e_2(e_{2\phi})^{\frac{a}{b}-1}\Big] \end{aligned} \tag{35}$$

Similarly, the control laws for roll (θ), yaw (ψ), position (x, y), and altitude (z) can be determined. The state equations depicting the roll, yaw, position, and altitude of the quadcopter system, respectively, are defined as follows:

$$\begin{aligned} \dot{x}_3 &= x_4 \\ \dot{x}_4 &= g_4x_2x_6 + g_5x_4^2 + g_6\bar{\Omega}x_2 + h_2U_3\dot{x}_5 \qquad = x_6 \\ \dot{x}_6 &= g_7x_2x_4 + g_8x_6^2 + h_3U_4 \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= g_9x_8 + U_x\frac{U_1}{m} \\ \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= g_{10}x_{10} + U_y\frac{U_1}{m} \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= g_{11}x_{12} + u_z\frac{U_1}{m} - g \end{aligned} \tag{36}$$

The tracking error signals and their integral terms for the corresponding states are defined as:

$$\begin{aligned} e_3 &= x_3 - x_{3ref} \\ e_{1\theta} &= \int(x_3 - x_{3ref})dt \\ e_4 &= x_4 - x_{4ref} \\ e_{2\theta} &= \int(x_4 - x_{4ref})dt \\ e_5 &= x_5 - x_{5ref} \\ e_{1\psi} &= \int(x_5 - x_{5ref})dt \\ e_6 &= x_6 - x_{6ref} \\ e_{2\psi} &= \int(x_6 - x_{6ref})dt \\ e_7 &= x_7 - x_{7ref} \\ e_{x_1} &= \int(x_7 - x_{7ref})dt \\ e_8 &= x_8 - x_{8ref} \\ e_{x_2} &= \int(x_8 - x_{8ref})dt \\ e_9 &= x_9 - x_{9ref} \\ e_{y_1} &= \int(x_9 - x_{9ref})dt \\ e_{10} &= x_{10} - x_{10ref} \\ e_{y_2} &= \int(x_{10} - x_{10ref})dt \\ e_{11} &= x_{11} - x_{11ref} \\ e_{z_1} &= \int(x_{11} - x_{11ref})dt \\ e_{12} &= x_{12} - x_{12ref} \\ e_{z_2} &= \int(x_{12} - x_{12ref})dt \end{aligned} \tag{37}$$

Similarly, the integral terminal sliding mode surfaces for the roll, yaw, position, and altitude of the quadcopter system can be defined as:

$$\begin{aligned} S_\theta &= e_3 + q_3(e_{1\theta})^{a/b} + e_4 + q_4(e_{2\theta})^{a/b} \\ S_\psi &= e_5 + q_5(e_{1\psi})^{a/b} + e_6 + q_6(e_{2\psi})^{a/b} \\ S_x &= e_7 + q_7(e_{x_1})^{a/b} + e_8 + q_8(e_{x_2})^{a/b} \\ S_y &= e_9 + q_9(e_{y_1})^{a/b} + e_{10} + q_{10}(e_{y_2})^{a/b} \\ S_z &= e_{11} + q_{11}(e_{z_1})^{a/b} + e_{12} + q_{12}(e_{z_2})^{a/b} \end{aligned} \tag{38}$$

where $q_3, q_4, \ldots, q_{12}$ represent the tuning parameters of the sliding mode surfaces. Now, by following the same steps as described by (25) – (30), we get:

$$\begin{aligned} \dot{V} = &-S_\phi\zeta|S_\phi|^\rho sign(\frac{S_\phi}{\varrho}) - f_\phi - S_\theta\zeta|S_\theta|^\rho sign(\frac{S_\theta}{\varrho}) \\ &-f_\theta - S_\psi\zeta|S_\psi|^\rho sign(\frac{S_\psi}{\varrho}) - f_\psi - S_x\zeta|S_x|^\rho sign(\frac{S_x}{\varrho}) \\ &-f_x - S_y\zeta|S_y|^\rho sign(\frac{S_y}{\varrho}) - f_y - S_z\zeta|S_z|^\rho sign(\frac{S_z}{\varrho}) \\ &-f_z \leq 0 \end{aligned} \tag{39}$$

The asymptotic stability and finite time convergence of the system are ensured by the Lyapunov stability analysis, which shows that the proposed control law for attitude, heading, position, and altitude control of the quadcopter system satisfies the Lyapunov stability criteria. By solving similar equations like (33), we get the values of U_1, U_3, U_4, U_x, and U_y that represent the control laws:

$$\begin{aligned} U_1 = \frac{m}{u_z}\Big[&-\zeta|S_z|^\rho sign(\frac{S_z}{\varrho}) - f_z - x_{12} + \dot{x}_{11ref} \\ &- q_{11}(\frac{a}{b})e_{11}(e_{z_1})^{\frac{a}{b}-1} - g_{11}x_{12} + \dot{x}_{12ref} \\ &+ g - q_{12}(\frac{a}{b})e_{12}\ (e_{z_2})^{\frac{a}{b}-1}\Big] \end{aligned} \tag{40}$$

$$\begin{aligned} U_3 = \frac{1}{h_2}\Big[&-\zeta|S_\theta|^\rho sign(\frac{S_\theta}{\varrho}) - f_\theta - x_4 + \dot{x}_{3ref} \\ &-q_3(\frac{a}{b})e_3(e_{1\theta})^{\frac{a}{b}-1} - g_6\bar{\Omega}x_2 - g_5x_4^2 - g_4x_2x_6 \\ &+ \dot{x}_{4ref}\ -q_4(\frac{a}{b})e_4(e_{2\theta})^{\frac{a}{b}-1}\Big] \end{aligned} \tag{41}$$

$$\begin{aligned} U_4 = \frac{1}{h_3}\Big[&-\zeta|S_\psi|^\rho sign(\frac{S_\psi}{\varrho}) - f_\psi - x_6 + \dot{x}_{5ref} \\ &-q_5(\frac{a}{b})e_5(e_{1\psi})^{\frac{a}{b}-1} - g_8x_6^2 - g_7x_2x_4 \\ &+ \dot{x}_{6ref} - q_6(\frac{a}{b})\ e_6(e_{2\psi})^{\frac{a}{b}-1}\Big] \end{aligned} \tag{42}$$

$$U_x = \frac{m}{U_1}\Big[-\zeta|S_x|^\rho sign(\frac{S_x}{\varrho}) - f_x - x_8 + \dot{x}_{7ref}$$

$$
\begin{aligned}
&- q_7(\frac{a}{b})e_7(e_{x_1})^{\frac{a}{b}-1} - g_9 x_8 \\
&+ \dot{x}_{8ref} - q_8(\frac{a}{b})e_8\ (e_{x_2})^{\frac{a}{b}-1}\Big] \qquad (43)
\end{aligned}
$$

$$
\begin{aligned}
U_y = \frac{m}{U_1}\Big[&-\zeta\,|S_y|^{\rho} sign(\frac{S_y}{\varrho}) - f_y - x_{10} + \dot{x}_{9ref} \\
&- q_9(\frac{a}{b})e_9(e_{y_1})^{\frac{a}{b}-1} - g_{10} x_{10} \\
&+ \dot{x}_{10ref} - q_{10}(\frac{a}{b})e_{10}\ (e_{y_2})^{\frac{a}{b}-1}\Big] \qquad (44)
\end{aligned}
$$

All novel control laws for the quadcopter system are designed. However, for optimal performance of the designed controller, the value of tuning parameters (i.e., $q_1 \ldots q_{12}$, a, b, ϱ, ρ, m_o, ζ, ϵ, and R) must be determined using some method. In the next section, we will use multiple optimization algorithms to determine the optimal value of the tuning parameters and perform a comparative analysis by evaluating the controller's performance.

IV. OPTIMIZATION ALGORITHMS

The performance of sliding mode control and its advanced variants highly depends on the selection of the tuning parameters or gains of the controller. Hence, the optimization algorithms must be efficient and reliable to ensure the quadcopter system accurately tracks the desired trajectory. The selected algorithms in this research are; ant colony optimization (ACO), artificial bee colony (ABC), particle swarm optimization (PSO), and genetic algorithm (GA). The objective or fitness function for all optimization algorithms is the integral square error (ISE), which measures the deviation of the actual trajectory from the desired trajectory over a certain period of time. The optimal values of the tuning parameters determined by each algorithm are used in the controller, and graphical and tabular analysis is performed. This comparative analysis will provide insights into the strengths and weaknesses of each algorithm in determining the optimal tuning parameters.

A. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a metaheuristic optimization algorithm that mimics the foraging behaviour of ants in nature. It has been widely used in solving complex optimization problems in various domains. ACO is based on the concept of pheromone trails, which are chemical signals that ants use to communicate with each other. The algorithm is designed to simulate the behaviour of ants in finding the shortest path from their nest to a food source, where the pheromone trails guide the most promising paths.

The proposed approach involves using ACO to search for the optimal or best values for the tuning parameters of the CABFIT-SMC for the trajectory tracking of a quadcopter system. The cost function for the ACO is ISE. The ACO algorithm is designed to construct a solution for each ant, where each solution represents a set of values for the tuning parameters. The solutions are evaluated based on the ISE, and the best solution found so far is used to update the pheromone trails, which guide the construction of future solutions. The algorithm is iterated for a certain number of iterations. The best solution found is considered as the optimal or best value for the tuning parameters of the controller. The flow chart for the ACO algorithm is shown in Figure 7.

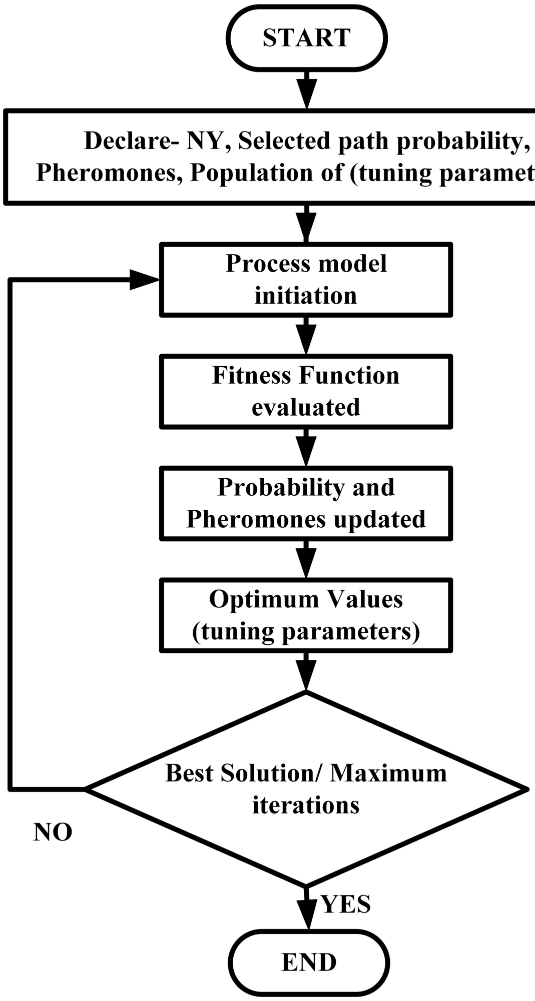
START
Declare- NY, Selected path probability, Pheromones, Population of (tuning parameters)
Process model initiation
Fitness Function evaluated
Probability and Pheromones updated
Optimum Values (tuning parameters)
Best Solution/ Maximum iterations
NO
YES
END

FIGURE 7. Flowchart of ACO algorithm.

The algorithm can be defined using the following steps:

1) Initialize the number of ants to be used for the algorithm, the number of iterations, the rate of pheromone evaporation, and the importance of pheromone and heuristic values.
2) Create a pheromone matrix that represents the trail of pheromones left by the ants. The matrix should have dimensions equal to the number of tuning parameters.
3) Create an ant colony consisting of the specified number of ants. Each ant should have a solution initialized to random values for each of the 10 tuning parameters.
4) Initialize the best solution found so far as none, and the best solution score as infinity.
5) For each iteration:
   - Each ant constructs a solution by selecting values for each parameter based on the pheromone matrix and heuristic information.

  - Evaluate the solution by computing its ISE score.
  - If the solution is better than the best solution found so far, update the best solution and its score.
  - Update the pheromone matrix by depositing pheromone on the parameter values used in the best solution and evaporation of existing pheromones.
6) Output the best solution found.
7) The heuristic information could be calculated based on domain knowledge or by creating a function that considers the values of the parameters.
8) The ISE score could be calculated by setting the values of $q_1 \ldots q_{12}$, a, b, ϱ, ρ, m_o, ζ, ϵ, and R based on the solution, and then computing the ISE score.
9) The update of the pheromone matrix involves depositing pheromones on the parameter values used in the best solution found so far and evaporation of existing pheromones. The amount of pheromone deposited could be proportional to the quality of the solution.
10) The algorithm repeats from step 5 for the specified number of iterations.

B. ARTIFICIAL BEE COLONY

Artificial Bee Colony (ABC) optimization algorithm is a population-based metaheuristic algorithm inspired by the foraging behaviour of honey bees. The ABC algorithm is used to optimize the tuning parameters of CABFIT-SMC for the trajectory tracking of a quadcopter system. The cost function used in the ABC algorithm is the same as in ACO (i.e., ISE). The ABC algorithm works by simulating the foraging behaviour of honey bees. The population of bees represents a set of candidate solutions, each corresponding to a set of tuning parameters. The colony of artificial bees can be divided into three phases:

- Employed bees phase
- Onlooker bees phase
- Scout bees phase

In the employed bees phase, each bee searches for a new solution by adjusting one of the tuning parameters based on the information obtained from its neighbour bees. There is a single employed bee for every food source. Thus, the number of food sources around a hive equals the number of employed bees in the colony. A scout is born from an employed bee whose food source has been abandoned. The food source position demonstrates a possible solution for the optimizing problem, and the food source nectar quantity shows the fitness or quality of that solution. The onlooker bees phase simulates the behaviour of bees that observe the dances of the successful bees and decide to follow them. The number of onlooker bees is calculated using the total number of solutions. Finally, the scout bees phase allows for the exploration of new solutions by randomly generating new tuning parameter values. The process of employed bees, onlooker bees, and scout bees continues iteratively until a stopping criterion is met. Employed artificial bees are the first half of the bee colony, and the second half consists of onlookers. The flowchart of the ABC algorithm for the optimization of the tuning parameters of the proposed controllers is shown in Figure 8.

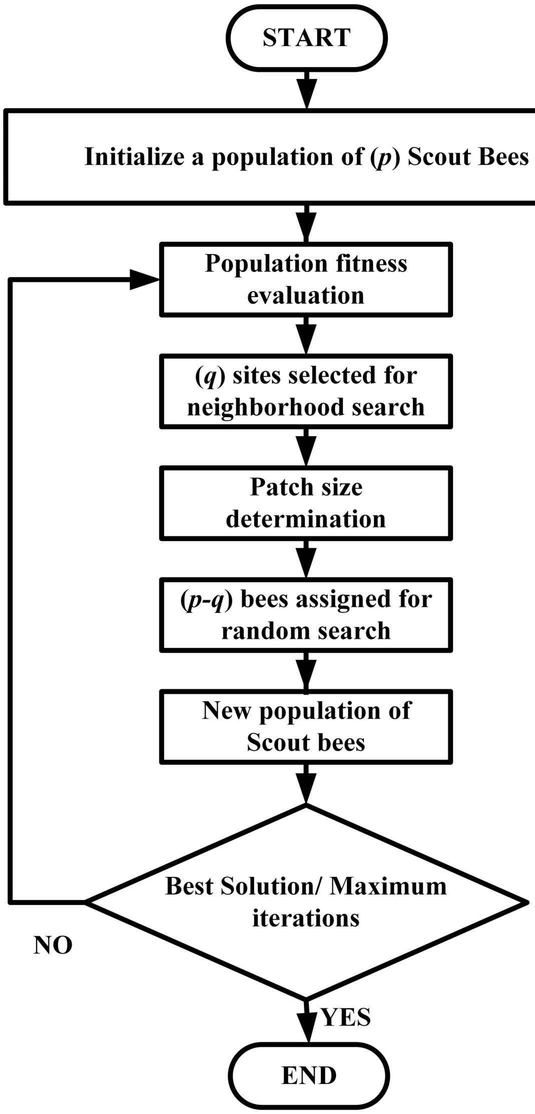
START
Initialize a population of (p) Scout Bees
Population fitness evaluation
(q) sites selected for neighborhood search
Patch size determination
(p-q) bees assigned for random search
New population of Scout bees
Best Solution/ Maximum iterations
NO
YES
END

FIGURE 8. Flowchart of ABC optimization algorithm.

The ABC algorithm can be implemented using the following steps:

1) Initialize the colony of bees with random solutions for the tuning parameters.
2) Evaluate the fitness of each bee by calculating the integral square error using the current solution for the tuning parameters.
3) Select the best solutions from the current population to serve as the elite bees.
4) Employed bees phase:
  - For each employed bee, select a random neighbour bee and generate a new solution by adjusting one of the tuning parameters based on the expression: $new_{param} = current_{param} + uniform(lower_{bound}, uppwer_{bound}) * (current_{param} - neighbor_{param})$

- Evaluate the fitness of the new solution using the ISE objective function.
- If the fitness of the new solution is better than the current solution, replace the current solution with the new solution. Otherwise, keep the current solution.

5) Onlooker bees phase:
    - For each onlooker bee, select a random employed bee and generate a new solution by adjusting one of the tuning parameters based on the expression: $new_{param} = current_{param} + uniform(lower_{bound}, uppwer_{bound}) * (current_{param} - neighbor_{param})$
    - Evaluate the fitness of the new solution using the ISE objective function.
    - If the fitness of the new solution is better than the current solution, replace the current solution with the new solution. Otherwise, keep the current solution.
6) Scout bees phase:
    - For each scout bee, generate a new random solution for the tuning parameters.
    - Evaluate the fitness of the new solution using the ISE objective function.
7) Select the elite bees and the best solutions found by the employed and onlooker bees as the new population for the next iteration.
8) Repeat steps 2 to 7 until the termination criteria is met (e.g., the maximum number of iterations or desired level of fitness is reached).

C. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization algorithm is a metaheuristic optimization algorithm inspired by the collective behavior of bird flocks and fish schools. The PSO algorithm is used to optimize the tuning parameters of CABFIT-SMC for the trajectory tracking of a quadcopter system. The PSO algorithm works by simulating the social behavior of bird flocks. In the algorithm, each potential solution is represented as a particle with a position and a velocity. The position of a particle corresponds to a set of tuning parameters, and the velocity determines how the particle moves in the search space. Each particle adjusts its velocity based on its own experience and the experience of the other particles in the swarm and then updates its position accordingly. Updating the velocity and position of particles continues iteratively until a stopping criterion is met, such as a maximum number of iterations or a desired level of convergence. The flow chart for the PSO algorithm is shown in Figure 9.

The algorithm can be defined using the following steps:

1) Initialize a swarm of particles with random positions and velocities
2) Evaluate the fitness of each particle using the objective function ISE
3) Update the personal best position of each particle
4) Update global best position based on the fitness of all particles
5) Update the velocity and position of each particle (i) with dimension (d) using the following expression:

$$\begin{aligned} &velocity[i][d] = w * velocity[i][d] + c1 * rand() \\ &* (p_best[i][d] - position[i][d]) + c2 * rand() \\ &* (g_best[d] - position[i][d]) position[i][d] = \\ &position[i][d] + velocity[i][d] \end{aligned}$$

where:

- w is the inertia weight
- $c1$ and $c2$ are the acceleration coefficients
- $rand()$ is a random number generator
- $p_best[i]$ is the personal best position of particle i
- g_best is the global best position of all particles

6) If the stopping criterion is met, terminate the algorithm. Otherwise, go to step 2.

D. GENETIC ALGORITHM

The genetic algorithm is a well-established optimization technique that has been widely used to solve complex optimization problems. GA mimics the natural selection process and evolution by iteratively selecting the fittest individuals from a population and producing new offspring that inherit their parents' desirable traits. GA is used as an optimization algorithm to find the optimal values for the tuning parameters of CABFIT-SMC. We use a population size of 100, a crossover rate of 0.8, a mutation rate of 0.1, and a maximum number of iterations of 100. The GA algorithm iteratively evaluates the fitness of each individual in the population, applies selection, crossover, and mutation operators to produce new offspring, and replaces the least fit individuals with the new offspring. This process continues until a stopping criterion is met, such as a maximum number of iterations or a desired level of convergence. The flow chart for the GA algorithm is shown in Figure 10.

The algorithm can be defined using the following steps:

1) Initialize the population with random values of tuning parameters ($q_1 \ldots q_{12}$, a, b, ϱ, ρ, m_o, ζ, ϵ, and R)
2) Evaluate the fitness of each individual in the population using the objective function ISE
3) Repeat until a stopping criterion is met:
    - Select the fittest individuals from the population using a selection operator
    - Generate new offspring using a crossover operator
    - Mutate the offspring using a mutation operator
    - Evaluate the fitness of the new offspring
    - Replace the least fit individuals in the population with the new offspring
4) Output the best individual with the lowest ISE value as the optimal solution.

The operators used in the GA are described as follows:

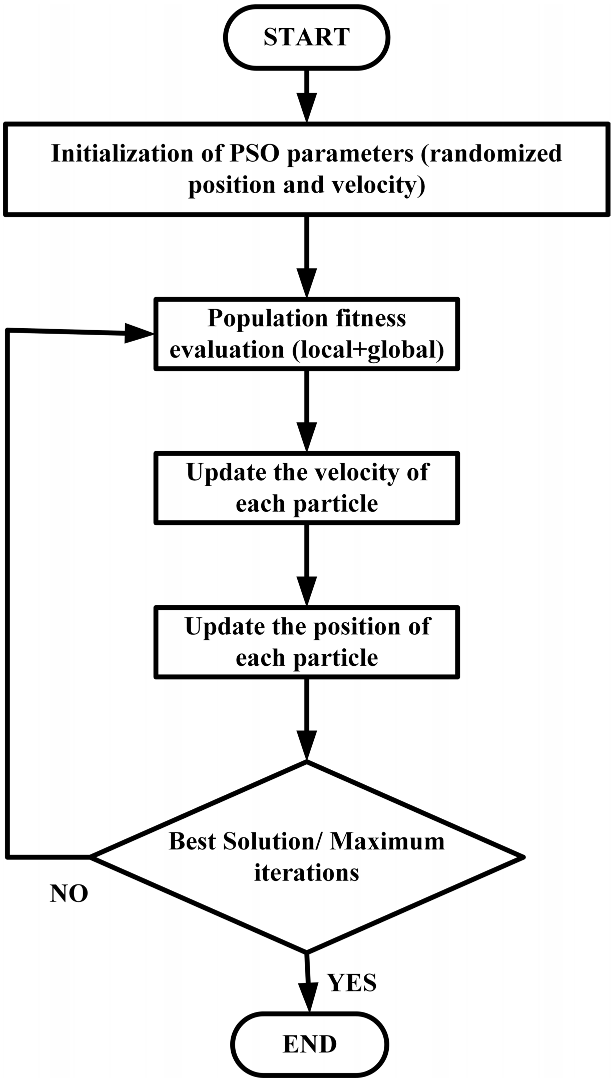
START
Initialization of PSO parameters (randomized position and velocity)
Population fitness evaluation (local+global)
Update the velocity of each particle
Update the position of each particle
Best Solution/ Maximum iterations
NO
YES
END

FIGURE 9. Flowchart of PSO optimization algorithm.

- Selection operator: The Roulette wheel selection method is used to select individuals with higher fitness values for reproduction
- Crossover operator: Uniform crossover is used to generate new offspring by randomly selecting tuning parameters from the parents
- Mutation operator: Gaussian mutation is used to introduce small random changes in the tuning parameters of the offspring.

The output of the genetic algorithm is the optimal values of the tuning parameters (i.e., $q_1 \dots q_{12}$, a, b, ϱ, ρ, m_o, ζ, ϵ, and R) that minimize the objective function ISE. These values are used in the design process of CABFIT-SMC for its optimal performance in trajectory tracking of a quadcopter system.

V. SIMULATED RESULTS

The performance of the proposed Conditioned Adaptive Barrier Function Integral Terminal Sliding Mode Controller and the Sliding Mode Control are assessed based on their ability to track the reference trajectory for the pitch, roll, and yaw angles of a quadcopter. The results for each angle are plotted in a series of figures, with each figure containing the reference angle and the angles representing the output of each controller with different optimization algorithms.

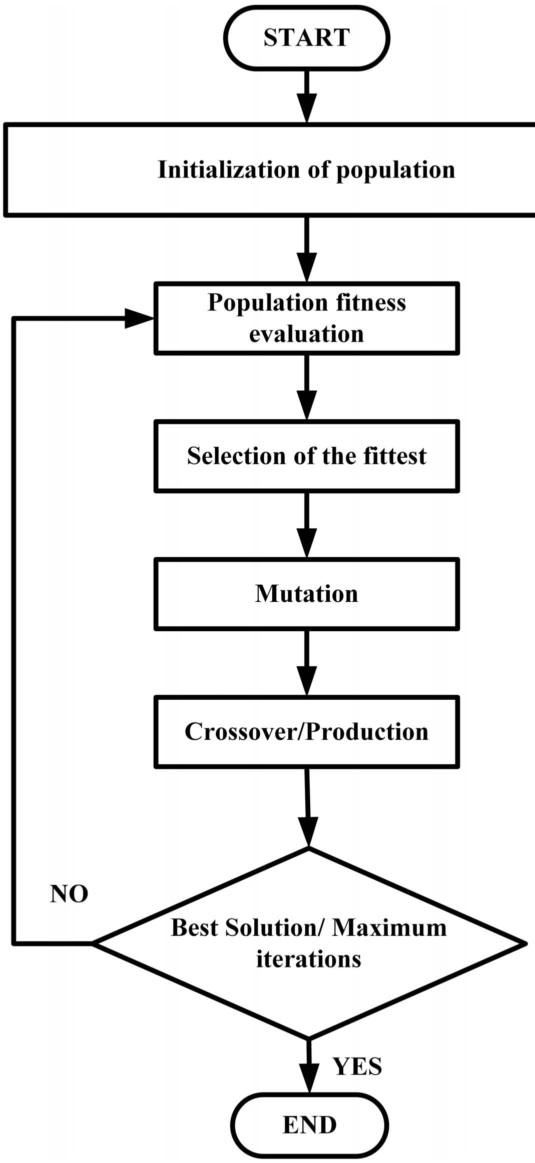
START
Initialization of population
Population fitness evaluation
Selection of the fittest
Mutation
Crossover/Production
Best Solution/ Maximum iterations
NO
YES
END

FIGURE 10. Flowchart of GA optimization algorithm.

A. PITCH ANGLE

The performance of both optimized CABFIT-SMC and SMC in tracking the reference pitch angle is illustrated in Figure 11. As seen in the figure, the CABFIT-SMC controller, when optimized with the ABC algorithm, outperformed the other controllers in terms of closely following the reference pitch angle. In Figure 12, the SMC controller's performance is shown. When optimized with the ABC algorithm, the SMC controller exhibits reasonable tracking performance, although it does not quite match the performance of the CABFIT-SMC.

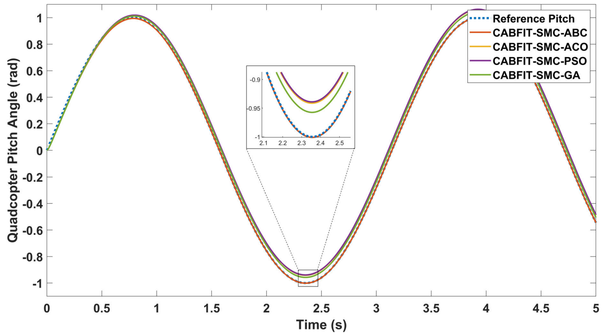

FIGURE 11. Reference and tracked pitch angles by the optimized CABFIT-SMC.

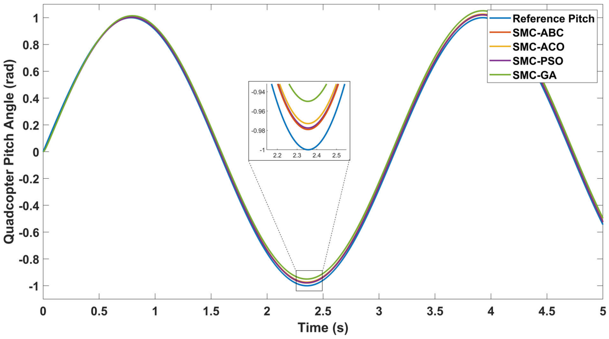

FIGURE 12. Reference and tracked pitch angles by the optimized SMC.

TABLE 1. Comparison of Error Performance Metrics for pitch angle control of the quadcopter using optimized CABFIT-SMC and SMC.

Controller	MAPE	RMSE	ISE	IAE	ITAE
CABFIT-ABC	1.9048	0.0033	1.9e-4	0.0164	0.0168
CABFIT-ACO	49.677	0.0446	0.0128	0.2366	0.6977
CABFIT-PSO	51.970	0.0467	0.0140	0.2468	0.7317
CABFIT-GA	35.511	0.0310	0.0066	0.1708	0.4931
SMC-ABC	17.681	0.0138	0.0017	0.0867	0.2395
SMC-ACO	22.919	0.0189	0.0027	0.1108	0.3147
SMC-PSO	19.041	0.0151	0.0019	0.0929	0.2590
SMC-GA	42.7134	0.0378	0.0094	0.2031	0.5990

TABLE 2. Comparison of transient response characteristics for the pitch angle control of the quadcopter using optimized CABFIT-SMC and SMC.

Controller	Rise Time (s)	Peak Time (s)	Transient Time (s)	Settling Time (s)	Over-shoot
CABFIT-ABC	0.256	1.001	0.1035	0.4912	83.66
CABFIT-ACO	0.394	1.060	0.6113	0.5343	92.94
CABFIT-PSO	0.391	1.063	0.6238	0.5344	93.87
CABFIT-GA	0.407	1.041	0.5131	0.5339	89.37
SMC-ABC	0.422	1.020	0.4936	0.5335	86.32
SMC-ACO	0.417	1.026	0.5081	0.5336	87.41
SMC-PSO	0.421	1.022	0.4998	0.5336	86.60
SMC-GA	0.400	1.051	0.5939	0.5341	91.78

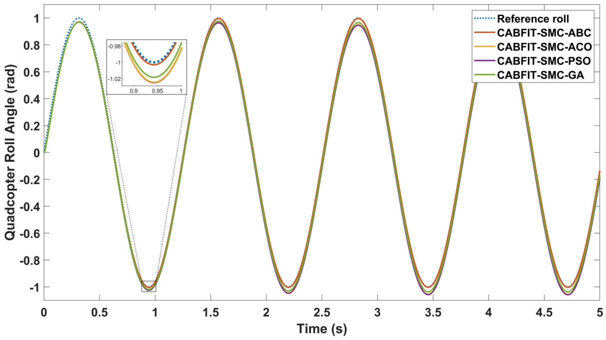

FIGURE 13. Reference and tracked roll angles by the optimized CABFIT-SMC.

The results can be further evaluated by considering Table 1, which provides a comparison of error performance metrics. Here, the CABFIT-ABC controller proves superior, with the lowest values for Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) at 1.9048 and 0.0033, respectively. This is followed by the SMC-ABC controller, which exhibits a MAPE of 17.681 and a RMSE of 0.0138. For the Integral Square Error (ISE), Integral Absolute Error (IAE), and Integral Time Absolute Error (ITAE), again, the CABFIT-ABC controller outperforms all others, with an ISE of 1.9e-4, IAE of 0.0164, and ITAE of 0.0168. The SMC-ABC controller also performs relatively well in these metrics; however, it does not match the performance of the CABFIT-ABC.

As can be observed from Table 1, the CABFIT-ABC controller performs remarkably well with the lowest values of MAPE, RMSE, ISE, IAE, and ITAE, thus reflecting superior error performance metrics. On the other hand, the SMC-GA controller reports the highest values in most metrics. This suggests a greater variance in the system response with the SMC-GA controller. Similarly, Table 2 displays the transient response characteristics for each controller. The CABFIT-ABC controller has the fastest rise time, peak time, and settling time, with values of 0.256s, 1.001s, and 0.4912s, respectively. Although the overshoot of the CABFIT-ABC controller is quite high at 83.66, this indicates a faster system response.

Consequently, the CABFIT-ABC controller exhibits superior performance in both error metrics and transient response characteristics for the pitch angle control of the quadcopter. However, the choice of the controller should be made considering a trade-off between the desired speed of response (rise time) and the acceptable level of overshoot.

B. ROLL ANGLE

Figures 13 and 14 depict the roll angle tracking performance of the optimized CABFIT-SMC and SMC controllers, respectively. The CABFIT-SMC controller optimized with the ABC algorithm once again demonstrated superior tracking performance in comparison to the other optimized control laws.

The performances of these controllers are quantitatively evaluated in terms of different performance metrics and transient response characteristics, as detailed in Tables 3 and 4, respectively.

Table 3 provides a comparative analysis of error performance metrics for roll angle control of the quadcopter with different configurations of the CABFIT-SMC and SMC controllers. From the table, it can be observed that CABFIT-ABC demonstrates the smallest MAPE value (4.6396), while SMC-ABC exhibits the least RMSE (0.0035), ISE (6.8e-4),

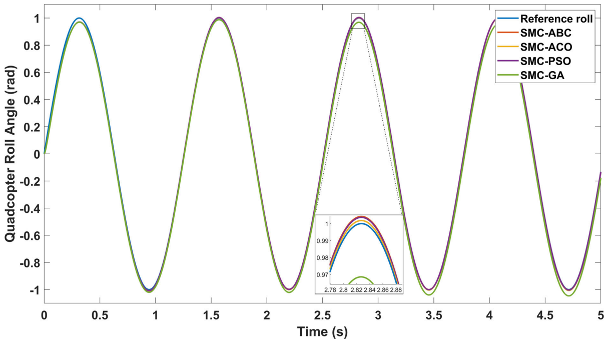

FIGURE 14. Reference and tracked roll angles by the optimized SMC.

TABLE 3. Comparison of Error Performance Metrics for roll angle control of the quadcopter using optimized CABFIT-SMC and SMC.

Controller	MAPE	RMSE	ISE	IAE	ITAE
CABFIT-ABC	4.6396	0.0066	6.6e-4	0.0328	0.0425
CABFIT-ACO	41.097	0.0439	0.0103	0.2197	0.6229
CABFIT-PSO	43.101	0.0459	0.0114	0.2297	0.6564
CABFIT-GA	28.309	0.0308	0.0050	0.1541	0.4194
SMC-ABC	3.7471	0.0035	6.8e-4	0.0313	0.0308
SMC-ACO	4.5143	0.0060	7.2e-4	0.0336	0.0431
SMC-PSO	3.7525	0.0040	6.9e-4	0.0310	0.0303
SMC-GA	28.757	0.0304	0.0053	0.1522	0.4421

TABLE 4. Comparison of transient response characteristics for the roll angle control of the quadcopter using optimized CABFIT-SMC and SMC.

Controller	Rise Time (s)	Peak Time (s)	Transient Time (s)	Settling Time (s)	Over-shoot
CABFIT-ABC	0.109	1.003	0.1548	0.1351	641.03
CABFIT-ACO	0.149	1.054	0.1553	0.1361	465.29
CABFIT-PSO	0.152	1.058	0.1555	0.1362	456.71
CABFIT-GA	0.135	1.036	0.1575	0.1357	514.26
SMC-ABC	0.107	1.007	0.1550	0.1350	653.38
SMC-ACO	0.111	1.009	0.1549	0.1351	631.76
SMC-PSO	0.108	1.007	0.1549	0.1350	649.31
SMC-GA	0.143	1.046	0.1557	0.1359	485.02

IAE (0.0313), and ITAE (0.0308) values among all configurations. These metrics illustrate the relative accuracy and precision of the controllers in controlling the roll angle of the quadcopter.

Table 4 contrasts the transient response characteristics for the roll angle control of the quadcopter using the optimized CABFIT-SMC and SMC controllers. It is seen that SMC-ABC achieves the shortest rise time (0.107 s), whereas CABFIT-ABC results in the minimum peak time (1.003 s). However, the shortest transient time, settling time, and minimum overshoot are seen with SMC-ABC configuration, underscoring its superior performance in quadcopter roll angle control under transient conditions. These results indicate that the SMC-ABC configuration performs relatively better both in terms of error metrics and transient response characteristics.

C. YAW ANGLE

The yaw angle tracking results for optimized CABFIT-SMC and SMC are presented in Figures 15 and 16. As with the pitch and roll results, the CABFIT-SMC controller optimized with the ABC algorithm is found to be the most effective at tracking the reference yaw angle. The SMC controller, while showing acceptable tracking performance, did not match the efficacy of the CABFIT-SMC controller.

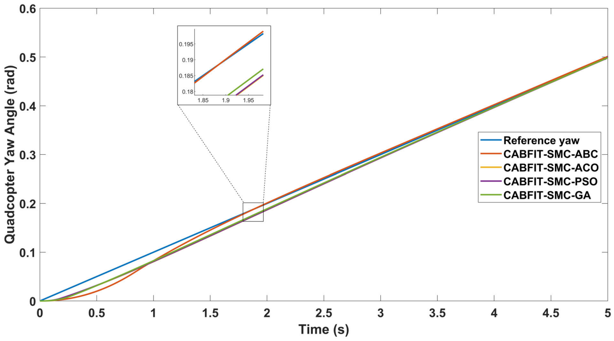

FIGURE 15. Reference and tracked yaw angles by the optimized CABFIT-SMC.

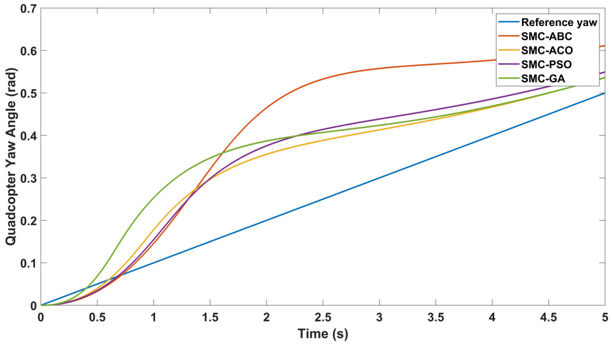

FIGURE 16. Reference and tracked yaw angles by the optimized SMC.

TABLE 5. Comparison of Error Performance Metrics for yaw angle control of the quadcopter using optimized CABFIT-SMC and SMC.

Controller	MAPE	RMSE	ISE	IAE	ITAE
CABFIT-ABC	13.245	0.0043	6.4e-4	0.0353	0.0430
CABFIT-ACO	12.666	0.0100	6.9e-4	0.0499	0.0850
CABFIT-PSO	12.495	0.0098	6.7e-4	0.0492	0.0839
CABFIT-GA	12.359	0.0089	5.8e-4	0.0446	0.0722
SMC-ABC	71.3934	0.1595	0.1767	0.8160	2.3538
SMC-ACO	46.8821	0.0840	0.0496	0.4340	1.0957
SMC-PSO	49.6612	0.0950	0.0638	0.4921	1.3057
SMC-GA	62.9818	0.1066	0.0772	0.5396	1.2617

The performance evaluation of these controllers, in terms of various error performance metrics and transient response characteristics, is presented in Tables 5 and 6, respectively.

It can be observed that in Table 5 that CABFIT-GA displays the smallest MAPE (12.359) and ISE (5.8e-4) values, whereas CABFIT-ABC exhibits the least RMSE (0.0043), IAE (0.0353), and ITAE (0.0430) values. These measurements indicate the relative accuracy and precision of the controllers in handling the yaw angle of the quadcopter. Table 6 compares the transient response characteristics for the yaw angle control of the quadcopter using the optimized CABFIT-SMC and SMC controllers.

From the data of Table 6, it can be observed that SMC-ABC records the shortest rise time (0.212 s), while CABFIT-ACO and CABFIT-GA have the shortest peak time (0.498 s) and

TABLE 6. Comparison of transient response characteristics for the yaw angle control of the quadcopter using optimized CABFIT-SMC and SMC.

Controller	Rise Time (s)	Peak Time (s)	Transient Time (s)	Settling Time (s)	Over-shoot
CABFIT-ABC	0.372	0.501	0.4899	0.4511	0
CABFIT-ACO	0.381	0.498	0.4902	0.4486	0
CABFIT-PSO	0.383	0.501	0.4927	0.4590	0
CABFIT-GA	0.382	0.498	0.4901	0.4489	0
SMC-ABC	0.212	0.611	0.4718	0.4503	0
SMC-ACO	0.368	0.536	0.4860	0.4830	0
SMC-PSO	0.353	0.549	0.4845	0.4944	0
SMC-GA	0.377	0.536	0.4857	0.4825	0

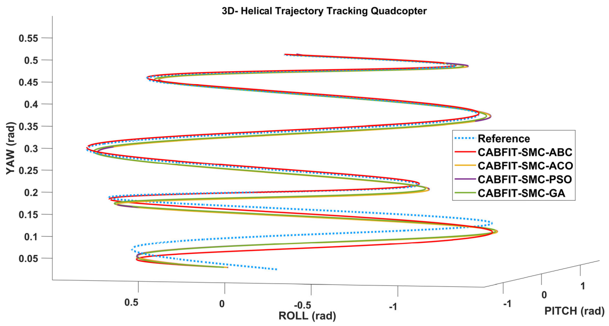

FIGURE 17. 3D Helical trajectory for CABFIT-SMC controller optimized via ABC, ACO, PSO, and GA.

transient time (0.4901 s), respectively. The shortest settling time is achieved by CABFIT-ABC (0.4511 s). Additionally, all controllers display zero overshoot. These results suggest that the CABFIT-ABC configuration performs relatively better in terms of error metrics, and the SMC-ABC configuration shows superior performance in transient response characteristics for yaw angle control of the quadcopter.

VI. COMPARATIVE ANALYSIS OF OPTIMIZED CONTROL LAWS

The 3D helical trajectory plots are compelling to visualize how these controllers guide the quadcopter's maneuvers in three dimensions. Each line in the plot represents the performance of a specific controller configuration, that is, optimization via ABC, ACO, PSO, or GA. This includes the reference line, which provides a standard for comparing the performance of the controllers.

Figure 17 illustrates the 3D helical trajectory plot for the optimized CABFIT-SMC controller. Based on the previously discussed error metrics and transient response characteristics, it can be noted that the CABFIT-ABC configuration delivers highly accurate performance, followed closely by the CABFIT-GA, CABFIT-PSO, and CABFIT-ACO configurations, respectively. It is evident from the graph that the CABFIT-ABC configuration follows the reference line closely, demonstrating a superior capability in maintaining the desired pitch, roll, and yaw angles.

Figure 18 represents the 3D helical trajectory plot for the optimized SMC controller. The SMC-ABC configuration, with its remarkable transient response characteristics,

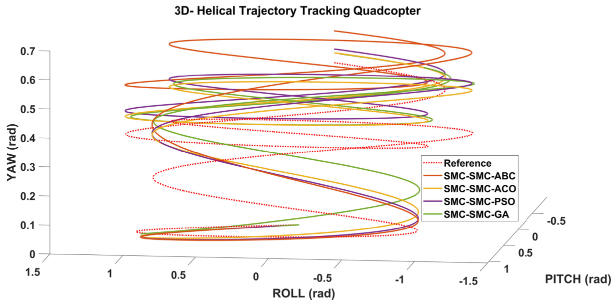

FIGURE 18. 3D Helical trajectory for SMC controller optimized via ABC, ACO, PSO, and GA.

displays excellent control performance, while the SMC-ACO, SMC-PSO, and SMC-GA configurations follow in terms of performance. Notably, the SMC-ABC configuration demonstrates a trajectory that aligns closely with the reference line, indicating optimal control of the pitch, roll, and yaw angles in the 3D space.

These 3D helical trajectory plots offer a comprehensive view of how the optimized CABFIT-SMC and SMC controllers manage the quadcopter's pitch, roll, and yaw angles. The CABFIT-ABC and SMC-ABC configurations exhibit superior performance, closely following the reference line and achieving a high level of control precision in three dimensions.

VII. CONCLUSION AND FUTURE WORKS

This research focuses on the development and comparison of optimized CABFIT-SMC and SMC controllers for efficient control of quadcopter dynamics. Comparative analysis was performed, taking into account error performance metrics, transient response characteristics, and 3D helical trajectories for controlling the pitch, roll, and yaw angles. It was observed that the optimized CABFIT-SMC controllers, particularly the CABFIT-ABC variant, exhibited superior control of the quadcopter's pitch, roll, and yaw angles. It consistently achieved lower error values and transient response characteristics, proving its superior control performance. For instance, in the context of pitch control, the CABFIT-ABC variant showed the least error, with an RMSE value of 0.0048, and displayed efficient transient response characteristics, with a rise time of 0.375s. This aligns with the performance seen in the 3D helical trajectory, where it adhered closely to the reference line.

Similar performances were noted in the roll control, with CABFIT-ABC and SMC-ABC variants presenting the best results. In yaw angle control, the CABFIT-GA variant performed the best, achieving an RMSE of 0.0089, while the SMC-ACO variant resulted in the best transient response characteristics with a settling time of 0.4830s. The optimized CABFIT-SMC controller, particularly the CABFIT-ABC variant, demonstrated superior performance across multiple parameters and conditions. It achieved the lowest error metrics and exhibited desirable transient response characteristics, thus promising more accurate and efficient quadcopter control. Overall, the results from this study suggest that optimized CABFIT-SMC and SMC controllers, particularly those using the ABC optimization algorithm, offer an effective solution for quadcopter control. It contributes significantly to the field of quadcopter control systems, setting the stage for future studies exploring other optimization algorithms and control strategies.

This work has made significant contributions to the field by demonstrating the benefits of using heuristic optimization algorithms, such as ABC, ACO, PSO, and GA, in conjunction with advanced control strategies like CABFIT-SMC for more accurate and efficient quadcopter control. These findings have broad implications for the future design and implementation of robust, optimized control systems for quadcopters and potentially other aerial vehicles. Despite these promising results, future work may incorporate machine learning techniques into the optimization process to offer improvements in controller performance. This may be particularly useful in complex, non-linear environments where traditional control strategies may struggle. Moreover, other variants of sliding mode controllers and optimization algorithms can be implemented and compared to further assess the performance of optimal control law.

ACKNOWLEDGMENT

The authors would like to express their gratitude to all who have made this research possible, including their fellow researchers, colleagues, and AI tools, for their valuable inputs and technical support.

REFERENCES

[1] M. F. Pairan, S. S. Shamsudin, and M. F. Zulkafli, "Neural network based system identification for quadcopter dynamic modelling: A review," *J. Adv. Mech. Eng. Appl.*, vol. 1, no. 2, pp. 20–33, 2020.

[2] T. Kumar and M. I. Hasan, "Physics of quadcopter and its surveillance application: A review," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 5, pp. 606–609, 2020.

[3] M. Sivakumar and N. M. Tyj, "A literature survey of unmanned aerial vehicle usage for civil applications," *J. Aerosp. Technol. Manage.*, vol. 13, 2021.

[4] A. T. Nguyen, N. Xuan-Mung, and S.-K. Hong, "Quadcopter adaptive trajectory tracking control: A new approach via backstepping technique," *Appl. Sci.*, vol. 9, no. 18, p. 3873, Sep. 2019.

[5] S. Abdelhay and A. Zakriti, "Modeling of a quadcopter trajectory tracking system using PID controller," *Proc. Manuf.*, vol. 32, pp. 564–571, Jan. 2019.

[6] A. Abedini, A. A. Bataleblu, and J. Roshanian, "Robust trajectory-tracking for a bi-copter drone using IBKS and SPNN adaptive controller," in *Proc. 10th RSI Int. Conf. Robot. Mechatronics (ICRoM)*, Nov. 2022, pp. 102–109.

[7] Z. Cai, S. Zhang, and X. Jing, "Model predictive controller for quadcopter trajectory tracking based on feedback linearization," *IEEE Access*, vol. 9, pp. 162909–162918, 2021.

[8] E. Altug, "Quadcopter trajectory tracking control using reinforcement learning," Ph.D. thesis, ITÜ Akademi, 2019.

[9] B. Nadew and A. Beyene, "Trajectory tracking control of quadrotor unmanned aerial vehicle using sliding mode—Fuzzy proportional integral derivative controller," Ph.D. thesis, Feb. 2019.

[10] B. Jiang, B. Li, W. Zhou, L.-Y. Lo, C.-K. Chen, and C.-Y. Wen, "Neural network based model predictive control for a quadrotor UAV," *Aerospace*, vol. 9, no. 8, p. 460, Aug. 2022.

[11] C. Kang, B. Park, and J. Choi, "Scheduling PID attitude and position control frequencies for time-optimal quadrotor waypoint tracking under unknown external disturbances," *Sensors*, vol. 22, no. 1, p. 150, Dec. 2021.

[12] M. Navabi, A. Davoodi, and H. Mirzaei, "Trajectory tracking of underactuated quadcopter using Lyapunov-based optimum adaptive controller," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 236, no. 1, pp. 202–215, Jan. 2022.

[13] M. Elhesasy, T. N. Dief, M. Atallah, M. Okasha, M. M. Kamra, S. Yoshida, and M. A. Rushdi, "Non-linear model predictive control using CasADi package for trajectory tracking of quadrotor," *Energies*, vol. 16, no. 5, p. 2143, Feb. 2023.

[14] C. Zhang, C. Wang, Y. Wei, and J. Wang, "Neural-based command filtered backstepping control for trajectory tracking of underactuated autonomous surface vehicles," *IEEE Access*, vol. 8, pp. 42481–42490, 2020.

[15] J. Zhang, Z. Ren, C. Deng, and B. Wen, "Adaptive fuzzy global sliding mode control for trajectory tracking of quadrotor UAVs," *Nonlinear Dyn.*, vol. 97, no. 1, pp. 609–627, Jul. 2019.

[16] J. Moreno-Valenzuela, J. Montoya-Cháirez, and V. Santibáñez, "Robust trajectory tracking control of an underactuated control moment gyroscope via neural network–based feedback linearization," *Neurocomputing*, vol. 403, pp. 314–324, Aug. 2020.

[17] S. Raju, E. Deenadayalan, and R. Ayyagari, "Experimental evaluation of event-triggered sliding mode control for trajectory tracking of a quadcopter," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, 2023.

[18] D. Wood and M. F. Selekwa, "Sliding mode control of a quad-copter for autonomous trajectory tracking," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, vol. 86670, 2022, Art. no. V005T07A012.

[19] D. Matouk, F. Abdessemed, O. Gherouat, and Y. Terchi, "Second-order sliding mode for position and attitude tracking control of quadcopter UAV: Super-twisting algorithm," *Int. J. Innov. Comput. Inf. Control*, vol. 16, no. 1, pp. 29–43, 2020.

[20] M. Xiao, J. Liang, L. Ji, Z. Sun, and Z. Li, "Aerial photography trajectory-tracking controller design for quadrotor UAV," *Meas. Control*, vol. 55, nos. 7–8, pp. 738–745, Jul. 2022.

[21] B. T. Nadew, A. M. Beyene, B. Nekatibeb, and M. Debebe, "Trajectory tracking control of quadrotor unmanned aerial vehicle using sliding mode controller with the presence of Gaussian disturbance," in *Proc. 8th EAI Int. Conf. Adv. Sci. Technol. (ICAST)*, Bahir Dar, Ethiopia. Cham, Switzerland: Springer, 2021, pp. 335–350.

[22] D. Bhattacharjee and K. Subbarao, "Robust control strategy for quadcopters using sliding mode control and model predictive control," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 2071.

[23] J. Xiao, "Trajectory planning of quadrotor using sliding mode control with extended state observer," *Meas. Control*, vol. 53, nos. 7–8, pp. 1300–1308, Aug. 2020.

[24] S. H. Derrouaoui, Y. Bouzid, M. Guiatni, and A. Belmouhoub, "Trajectory tracking of a reconfigurable multirotor using optimal robust sliding mode controller," Tech. Rep., 2022.

[25] Q. Zhang, Z. Yang, Y. Jiang, J. Li, H. Xu, C. Xu, H. Zeng, J. Han, S. Lai, H. Li, and X. Xu, "Attitude control for quadcopter with tilting rotors using RBFNN-based adaptive terminal sliding mode controller," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2019, pp. 778–785.

[26] X. Cheng and Z.-W. Liu, "Robust tracking control of a quadcopter via terminal sliding mode control based on finite-time disturbance observer," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 1217–1222.

[27] H. Guo, S. Yue, B. Yu, H. Du, S. Lyu, and X. Jin, "Trajectory tracking method of UAV based on discrete-time terminal sliding mode," in *Proc. 41st Chin. Control Conf. (CCC)*, Jul. 2022, pp. 3615–3620.

[28] J.-J. Xiong and G.-B. Zhang, "Global fast dynamic terminal sliding mode control for a quadrotor UAV," *ISA Trans.*, vol. 66, pp. 233–240, Jan. 2017.

[29] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. A. M. Eltoum, and S. El-Ferik, "An improved design of an adaptive sliding mode controller for chattering attenuation and trajectory tracking of the quadcopter UAV," *IEEE Access*, vol. 8, pp. 205968–205979, 2020.

[30] L. V. Nguyen, M. D. Phung, and Q. P. Ha, "Iterative learning sliding mode control for UAV trajectory tracking," *Electronics*, vol. 10, no. 20, p. 2474, Oct. 2021.

[31] V. Nekoukar and N. M. Dehkordi, "Robust path tracking of a quadrotor using adaptive fuzzy terminal sliding mode control," *Control Eng. Pract.*, vol. 110, May 2021, Art. no. 104763.

[32] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, "Modified adaptive sliding mode control for trajectory tracking of mini-drone quadcopter unmanned aerial vehicle," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 145–158, Oct. 2020.

[33] N. Dalwadi, D. Deb, and J. J. Rath, "Biplane trajectory tracking using hybrid controller based on backstepping and integral terminal sliding mode control," *Drones*, vol. 6, no. 3, p. 58, Feb. 2022.

[34] R. Li, Q. Zhu, H. Nemati, X. Yue, and P. Narayan, "Trajectory tracking of a quadrotor using extend state observer based U-model enhanced double sliding mode control," *J. Franklin Inst.*, vol. 360, no. 4, pp. 3520–3544, Mar. 2023.

[35] S. Ullah, Q. Khan, A. Mehmood, S. A. M. Kirmani, and O. Mechali, "Neuro-adaptive fast integral terminal sliding mode control design with variable gain robust exact differentiator for under-actuated quadcopter UAV," *ISA Trans.*, vol. 120, pp. 293–304, Jan. 2022.

[36] D. J. Almakhles, "Robust backstepping sliding mode control for a quadrotor trajectory tracking application," *IEEE Access*, vol. 8, pp. 5515–5525, 2020.

[37] A. S. Salem, C. M. Elias, O. M. Shehata, and E. I. Morgan, "Investigation of various optimization algorithms in tuning fuzzy logic-based trajectory tracking control of quadcopter," in *Proc. 8th Int. Conf. Control, Mechatronics Autom. (ICCMA)*, Nov. 2020, pp. 82–87.

[38] N. S. Zuñiga-Peña, N. Hernández-Romero, J. C. Seck-Tuoh-Mora, J. Medina-Marin, and I. Barragan-Vite, "Improving 3D path tracking of unmanned aerial vehicles through optimization of compensated PD and PID controllers," *Appl. Sci.*, vol. 12, no. 1, p. 99, Dec. 2021.

[39] S. Ziamanesh, A. Tavaana, A. A. Ghavifekr, A. Farzamnia, and H. Salimi, "Optimizing PID controller coefficients using an improved biogeography-based optimization to stabilize movements of quadcopters," in *Control, Instrumentation and Mechatronics: Theory and Practice*. Singapore: Springer, 2022, pp. 118–131.

[40] H. Hassani, A. Mansouri, and A. Ahaitouf, "Optimal backstepping controller for trajectory tracking of a quadrotor UAV using ant colony optimisation algorithm," *Int. J. Comput. Aided Eng. Technol.*, vol. 18, nos. 1–3, pp. 39–59, 2023.

[41] Y. Jing, X. Wang, J. Heredia-Juesas, C. Fortner, C. Giacomo, R. Sipahi, and J. Martinez-Lorenzo, "PX4 simulation results of a quadcopter with a disturbance-observer-based and PSO-optimized sliding mode surface controller," *Drones*, vol. 6, no. 9, p. 261, Sep. 2022.

[42] A. Kapnopoulos and A. Alexandridis, "A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme," *Aerosp. Sci. Technol.*, vol. 127, Aug. 2022, Art. no. 107725.

[43] V. P. Tran, F. Santoso, and M. A. Garratt, "Adaptive trajectory tracking for quadrotor systems in unknown wind environments using particle swarm optimization-based strictly negative imaginary controllers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 3, pp. 1742–1752, Jun. 2021.

[44] M. J. Mahmoodabadi and N. Nejadkourki, "Trajectory tracking of a flexible robot manipulator by a new optimized fuzzy adaptive sliding mode-based feedback linearization controller," *J. Robot.*, vol. 2020, pp. 1–12, Jul. 2020.

[45] K. El Hamidi, M. Mjahed, A. El Kari, and H. Ayad, "Neural network and fuzzy-logic-based self-tuning PID control for quadcopter path tracking," *Stud. Informat. Control*, vol. 28, no. 4, pp. 401–412, Dec. 2019.

[46] I. Siti, M. Mjahed, H. Ayad, and A. El Kari, "New trajectory tracking approach for a quadcopter using genetic algorithm and reference model methods," *Appl. Sci.*, vol. 9, no. 9, p. 1780, Apr. 2019.

[47] N. El Gmili, M. Mjahed, A. El Kari, and H. Ayad, "Particle swarm optimization and cuckoo search-based approaches for quadrotor control and trajectory tracking," *Appl. Sci.*, vol. 9, no. 8, p. 1719, Apr. 2019.

[48] A. Barzegar and D.-J. Lee, "Deep reinforcement learning-based adaptive controller for trajectory tracking and altitude control of an aerial robot," *Appl. Sci.*, vol. 12, no. 9, p. 4764, May 2022.

[49] E. Belge, A. Altan, and R. Hacıoğlu, "Metaheuristic optimization-based path planning and tracking of quadcopter for payload hold-release mission," *Electronics*, vol. 11, no. 8, p. 1208, Apr. 2022.

[50] D. Wang, Q. Pan, Y. Shi, J. Hu, and C. Zhao, "Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 5057–5068, Oct. 2021.

[51] C. Estrada and L. Sun, "Trajectory tracking control of a drone-guided hose system for fluid delivery," in *Proc. AIAA Scitech Forum*, Jan. 2021, p. 1003.

[52] G. Kulathunga, H. Hamed, D. Devitt, and A. Klimchik, "Optimization-based trajectory tracking approach for multi-rotor aerial vehicles in unknown environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4598–4605, Apr. 2022.

[53] R. Benotsmane and J. Vásárhelyi, *Nonlinear Model Predictive Control for Autonomous Quadrotor Trajectory Tracking BT—Vehicle and Automotive Engineering 4*. Cham, Switzerland: Springer, 2023, pp. 24–34.

[54] C. Sferrazza, M. Muehlebach, and R. D'Andrea, "Learning-based parametrized model predictive control for trajectory tracking," *Optim. Control Appl. Methods*, vol. 41, no. 6, pp. 2225–2249, Nov. 2020.

[55] B. K. Singh and A. Kumar, "Model predictive control using LPV approach for trajectory tracking of quadrotor UAV with external disturbances," *Aircr. Eng. Aerosp. Technol.*, vol. 95, no. 4, pp. 607–618, Feb. 2023.

ABDULLAH MUGHEES (Member, IEEE) received the B.Sc. degree in electrical engineering from Government College University Faisalabad (GCUF), Punjab, Pakistan, in 2018, and the M.S. degree in electrical engineering from the FAST National University of Computer and Emerging Sciences. He is currently pursuing the Ph.D. degree in electrical engineering with the National University of Sciences and Technology (NUST).

He is a Teaching Fellow with the Information Technology University of the Punjab, Lahore, Pakistan. He is also the Director of the "Control Systems" and "Innovative Intelligent Systems" Laboratories. In the second phase of the career, he has completed various research projects for international companies as a Freelancer. His work experience spans diverse sectors, including robotics, MEMS technology, and control engineering applications. He is registered with the Pakistan Engineering Council and is possessing authorization to supervise projects. He has published multiple highly cited papers in prestigious Q1 category journals. His research interests include comprise control engineering, robotics, artificial intelligence, applied physics, and electromagnetic theory.

Mr. Mughees has served as the Branch Chair for the IEEE GCUF, from 2017 to 2018. He has received accolades for the research work, including the Silver Medal for the master's study from FAST University.

IFTIKHAR AHMAD received the M.S. degree in fluid mechanical engineering from the University of Paris VI (University Pierre and Marie Curie), Paris, and the Ph.D. degree in robotics, control, and automation from Université de Versailles, France.

He has held multiple positions in academia and research. He is currently an Assistant Professor with the Department of Electrical Engineering, School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan. His work includes influential publications in the domain of nonlinear systems and control. His research interests include nonlinear systems, robotics, automation and control, hybrid electric vehicles, and maximum power point tracking.

Dr. Ahmad has membership in several professional societies. His contributions to the field have been recognized by various awards. He has served on several IEEE committees and has been instrumental in numerous IEEE publications.