International Journal of Dynamics and Control (2025) 13:188
https://doi.org/10.1007/s40435-025-01710-w

Twistor-based finite-time dynamic pose tracking control for coaxial octorotor UAV systems

Jiaxing Zhou[1,2] · Wei Chen[1,2] · Yifan Deng[3] · Zhao Deng[1,2]

Received: 21 January 2025 / Revised: 14 April 2025 / Accepted: 16 April 2025 / Published online: 3 May 2025

Abstract

Coaxial octorotors have gained widespread application due to their superior control performance and robustness compared to quadrotors. However, the introduction of four more rotors significantly increases the complexity of modeling and control, as the intricate rotor interactions demand a more comprehensive representation of the UAV's dynamics. Traditional approaches, such as the Newton–Euler method, often separate position and attitude dynamics, leading to redundant operations and increased computational load, which becomes even more problematic for coaxial octorotors. Twistor, as a novel rigid-body representation parameter, is particularly well suited for modeling coaxial octorotors, offering a unified, non-redundant framework that simplifies the description of UAV motion. To address the computational efficiency issues associated with dynamics modeling using the Newton–Euler method and dual quaternions, this study proposes a novel approach based on twistor, which enables a more concise and efficient unified description of UAV pose dynamics. To address the challenges posed by the strong dynamic performance of coaxial octorotors, we design a finite-time double-hyperbolic sliding mode controller for dynamic pose tracking, rigorously proven to ensure global finite-time stability using Lyapunov finite-time criterion and LaSalle invariance principles. The effectiveness and practicality of twistor-based modeling as well as twistor-based controller are demonstrated through hardware-in-the-loop simulations on the RT-Links platform, highlighting its potential for real-world applications in complex environments.

Keywords Twistor · Finite-time control · Coaxial octorotor · Dynamic pose tracking

1 Introduction

In recent years, multirotor UAVs, particularly quadrotors, have attracted considerable attention due to their advantages such as simplicity of operation, ease of control, and cost-effectiveness [1]. However, despite these benefits, quadrotors have certain limitations, especially in scenarios involving heavy payloads, harsh operating conditions, and fault-tolerant flight. In contrast, coaxial octorotors, with their eight rotors configured coaxially, can effectively address these limitations by offering superior stability, redundancy, and dynamic performance [2]. This configuration enhances payload capacity, improves flight stability, and provides better fault tolerance, especially in cases of rotor failure. Despite these advantages, the increased rotor count introduces significant complexities in the design of control systems. The interactions between the rotors and the increased demand for advanced control algorithms are more pronounced in coaxial octorotors compared to their quadrotor counterparts [3].

✉ Wei Chen
2322131007@stu.xmut.edu.cn

Jiaxing Zhou
2021000082@xmut.com

Yifan Deng
ivandunne@xjtu.edu.cn

Zhao Deng
2023000093@xmut.edu.cn

[1] School of Electric Engineering and Automation, Xiamen University of Technology, 600 Ligong Road, Xiamen 361024, Fujian, China

[2] Xiamen Key Laboratory of Frontier Electric Power Equipment and Intelligent Control, 600 Ligong Road, Xiamen 361024, Fujian, China

[3] Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, No.28 Xianning West, Xi'an 710075, Shaanxi, China

The dynamic response capability of a UAV pose control system is partly determined by the accuracy and computational speed of its dynamic model. In existing research on the modeling and control of coaxial octorotors, most studies model the UAV's position and attitude separately. In the research of quadrotor UAVs, the Newton–Euler method is commonly used for the dynamic description of the UAV [4–7]. Modeling is typically performed separately using the Newton–Euler method [8], where position is described by position vectors [9] and attitude is represented by parameters such as Euler angles [10] or quaternions [11]. In attitude representation, Euler angles are effective for small-angle maneuvers but introduce trigonometric complexity and computational overhead for large-angle rotations. Attitude quaternions address these limitations but suffer from parameter redundancy, requiring normalization and additional resources. For complex systems like coaxial octorotors, an integrated modeling approach is essential to optimize position and attitude representation, eliminating redundancy and computational burden while providing a more efficient and scalable solution. A unified pose modeling approach overcomes the limitations of separate modeling. In quadrotor research, dual quaternions have been widely used for UAV modeling and control [12, 13], providing a unified description of position and attitude. However, dual quaternions employ an eight-dimensional parameter set for six-degree-of-freedom motion, leading to redundancy and requiring normalization constraints in data filtering and state fusion [13]. Moreover, many dual quaternion-based controllers simplify the dual part to a position vector, neglecting the full dual quaternion form and introducing additional calculations.

To overcome these limitations, [14] introduces the twistor, a pose description parameter that extends the modified Rodrigues parameters (MRPs) within the dual-number algebra. The twistor uses a six-dimensional parameter set to describe six-degree-of-freedom pose motion without parameter redundancy. Furthermore, controllers based on twistor can be directly derived from MRP-based controllers. Therefore, twistor-based controllers eliminate the need for additional computational steps and normalization This is particularly advantageous for control systems of complex multirotor UAVs, such as coaxial octorotors, where accurate and efficient pose modeling is essential. To the best of our knowledge, this study represents the first application of twistor in the research of coaxial octorotors. We employ MATLAB for pose dynamic model simulation, which shares conceptual parallels with numerical methods in other dynamic system analysis [15–18]. The twistor's potential to simplify modeling and enhance control performance in such systems is demonstrated, offering a novel and efficient framework for future research and applications.

Coaxial octorotors are widely used in demanding environments like military reconnaissance, search-and-rescue, and disaster response due to their high payload capacity, fault tolerance, and stability. These missions often involve dynamic, time-sensitive tasks, requiring UAVs to adapt quickly to changing targets or unpredictable conditions. A robust and responsive pose control system is essential for dynamic pose tracking, ensuring real-time position and orientation accuracy in challenging scenarios. Traditional control methods, such as PD, PID, and linear sliding mode controllers, are widely used but often fall short in meeting the high dynamic response demands of complex tasks. Their limited precision and stability in fast-changing environments can result in suboptimal performance or instability. To address these issues, robust control strategies have been proposed in several studies. For example, a six-degree-of-freedom coaxial octorotor controller was developed to ensure robustness in the presence of actuator faults [19], while a higher-order integral dynamic sliding mode controller was designed for fault-tolerant control of position and attitude dynamics [20]. Additionally, a robust fault-tolerant control law was introduced to address attitude controllability issues in multirotor UAVs with actuator failures [8]. However, these approaches did not address finite-time control, which is critical for achieving time-optimal performance [21]. Finite-time control offers enhanced robustness and dynamic response compared to conventional closed-loop systems by incorporating fractional power terms [22, 23]. For coaxial octorotors, finite-time position and heading control has been proposed using back-stepping-based fast terminal sliding mode control [20]. Finite-time sliding mode control is particularly effective for UAV pose tracking, yet traditional linear reaching laws fail to deliver the dynamic responsiveness required for tracking dynamic pose targets. To address this, a double-hyperbolic reaching law (DHRL) was introduced, significantly improving dynamic response capabilities while avoiding oscillations [24]. This approach has been validated to exhibit excellent dynamic response and robustness, fundamentally resolving oscillation issues in quadrotor attitude control [25].

The main contribution lies in:

(1) This paper presents the first application of twistor in the modeling and control of coaxial octorotor systems. By leveraging twistor for dynamic description, the proposed approach eliminates parameter redundancy and computational overhead, addressing key limitations of dual quaternion-based methods, especially the inefficiencies in dual part transformation.
(2) A twistor-based finite-time double-hyperbolic sliding mode (FDSM) pose integration controller is designed and validated through hardware-in-the-loop simulation. The proposed FDSM controller demonstrates exceptional performance, achieving finite-time convergence and robust dynamic pose tracking, even in challenging scenarios.

This paper is organized as follows. Section 1 presents the current state of research and the significance of coaxial octorotor UAV modeling and control. Section 2 establishes octorotor dynamics and kinematics equations based on twistor. Section 3 designs the FDSM and verifies its stability using the Lyapunov finite-time stability criterion and the LaSalle invariance principle. Section 4 performs hardware-in-the-loop simulation validation of the proposed method and analyzes the results. The conclusion of this paper is given in Sect. 5.

2 Problem statement

2.1 Mathematical preliminaries

To provide a unified description, the following definitions of the sets related to real numbers and dual quaternions are presented. The real number set is defined as $\mathbb{R}$. The quaternion set is defined as $\mathbb{H} = \{q = q_0 + q_1 i + q_2 j + q_3 k, q_1, q_2, q_3 \in \mathbb{R}\}$, where i, j, k are imaginary unit satisfying $i^2 = j^2 = k^2 = ijk = -1$. The dual-number set is defined as $\mathbb{D}^n = \{\hat{a} = a_r + \varepsilon a_d \mid a_r, a_d \in \mathbb{R}^n\}$, where ε denotes the dual unit, which satisfies $\varepsilon^2 = 0, \varepsilon \neq 0$. The dual quaternion set is defined as $\mathbb{D}^{\mathcal{Q}} = \{\hat{q} = q_r + \varepsilon q_d \mid q_r, q_d \in \mathbb{H}\}$.

The dual quaternion multiply operation is defined as

$$\hat{p}\hat{q} = p_r q_r + \varepsilon(p_d q_r + p_r q_d) \tag{1}$$

where $\hat{p} = p_r + \varepsilon p_d, \hat{q} = q_r + \varepsilon q_d$. It is worth noting that the dual quaternions mentioned here are not normalized. Only dual quaternions that satisfy the normalization constraint can be used to represent the rigid-body pose. The unit dual quaternion set is defined as $\mathbb{D}^{\mathcal{Q}n} = \{\hat{q} \in \mathbb{D}^{\mathcal{Q}} \mid \|\hat{q}\| = 1\}$. When both real parts of the quaternion are zero, the set is denoted as $\mathbb{D}^{\mathcal{Q}v} = \{\hat{q} \in \mathbb{D}^{\mathcal{Q}} \mid q_{r0} = q_{d0} = 0\}$.

The vector mapping of quaternion can be written as

$$(q)_v = [q_1\ q_2\ q_3]^T = \boldsymbol{q} \tag{2}$$

and the quaternion mapping of vector can be written as

$$\overline{\boldsymbol{a}} = [0\ a_1\ a_2\ a_3]^T = [0\ \boldsymbol{a}]^T \tag{3}$$

It should be noted that the generalized dual quaternion cannot be directly used for representing rigid-body motion, as it has not undergone normalization. The normalization constraint of dual quaternion can be written as $\|\hat{q}\| = \hat{q}^*\hat{q} = \hat{q}\hat{q}^* = 1$, where $\hat{q}^*$ is the conjugate dual quaternion which can be expressed as $\hat{q}^* = q_r^* + \varepsilon q_d^*$.

Assume two frames $\mathcal{F}_O$ and $\mathcal{F}_A$. The rigid kinematic equation based on dual quaternion can be expressed as

$$\dot{\hat{q}} = \frac{1}{2}\hat{q}\hat{V}_{AO}^A = \frac{1}{2}\hat{V}_{AO}^O\hat{q} \tag{4}$$

where $\hat{V}_{AO}^A, \hat{V}_{AO}^O \in \mathbb{D}^{\mathcal{Q}v}$ are dual quaternion mapping of their relative vector and $\boldsymbol{\Psi}_{AO}^A$ is the dual velocity of frame $\mathcal{F}_O$ with respect to frame $\mathcal{F}_A$, expressed in $\mathcal{F}_A$. The relative frames for the other parameters in this paper are defined in the same manner and will not be elaborated upon further. The dual velocity can be expanded as

$$\begin{aligned}
\hat{V}_{AO}^A &= \bar{\boldsymbol{V}}_{AO}^A + \varepsilon\bar{\boldsymbol{v}}_{AO}^A \\
\hat{\boldsymbol{V}}_{AO}^A &= \boldsymbol{\omega}_{AO}^A + \varepsilon\boldsymbol{v}_{AO}^A \\
\hat{V}_{AO}^O &= \bar{\boldsymbol{\omega}}_{AO}^O + \varepsilon\bar{\boldsymbol{v}}_{AO}^O \\
\hat{\boldsymbol{V}}_{AO}^O &= \boldsymbol{\omega}_{AO}^O + \varepsilon\boldsymbol{v}_{AO}^O
\end{aligned} \tag{5}$$

where $\boldsymbol{v}_{AO}^A = \dot{\boldsymbol{r}}_{AO}^A + \boldsymbol{\omega}_{AO}^A \times \boldsymbol{r}_{AO}^A, \boldsymbol{v}_{AO}^O = \dot{\boldsymbol{r}}_{AO}^O + \boldsymbol{\omega}_{AO}^O \times \boldsymbol{r}_{AO}^O$. The twistor is defined as

$$\hat{B} = (\hat{q} + 1)^{-1}(\hat{q} - 1) \in \mathbb{D}^{\mathcal{Q}v} \tag{6}$$

In [14] and [26], the set of twistor has not been explicitly defined. This paper provides a detailed definition of the set of twistor as

$$\mathbb{B} = \left\{\hat{B} = (\hat{q} + 1)^{-1}(\hat{q} - 1)\ \in \mathbb{D}^{\mathcal{Q}v} \mid \forall \hat{q} \in \mathbb{D}^{\mathcal{Q}}\right\} \tag{7}$$

Naturally, $\mathbb{B} \subset \mathbb{D}^{\mathcal{Q}}$. The dual form of twistor can be written as

$$\begin{aligned}
\hat{B} &= \bar{\boldsymbol{p}} + \varepsilon\bar{\boldsymbol{b}} \\
\boldsymbol{b} &= \boldsymbol{\Pi}\boldsymbol{r}_{AO}^A = \boldsymbol{\Pi}^T\boldsymbol{r}_{AO}^O \in \mathbb{R}^3
\end{aligned} \tag{8}$$

where

$$\boldsymbol{\Pi} = \frac{1}{4}(1 - |\boldsymbol{p}|^2)\boldsymbol{I}_3 - \frac{1}{2}\boldsymbol{p}^\times + \frac{1}{2}\boldsymbol{p}\boldsymbol{p}^T \tag{9}$$

where $\boldsymbol{I}_3$ is three-dimensional identical matrix, $\boldsymbol{p} = [p_1\ p_2\ p_3]^T$ is relative modified Rodrigues parameter (MRP), and $\boldsymbol{p}^\times$ represents the skew-symmetric matrix of a three-dimensional vector. According to Eq. (8), it is obvious that the real part of twistor is MRP. MRP and unit quaternion double covering the three-dimensional rotational group SO(3), which result in unwinding phenomenon. The unwinding phenomenon can lead to unnecessary maneuvers in the UAV when the attitude maneuver exceeds π. Therefore, it is crucial to resolve the issue of unwinding. For dual quaternion, a pose can be represented by both $\hat{q}$ and $-\hat{q}$. Therefore, Eq. (6) can be rewritten as

$$\hat{\Gamma}(-\hat{q}) = (-\hat{q} + 1)^{-1}(-\hat{q} - 1) = \Gamma(\hat{q})^{-1} \tag{10}$$

Eq. (10) indicates that a pose can be represented by both $\hat{\Gamma}$ and $\hat{\Gamma}^{-1}$. To avoid the occurrence of the unwinding phenomenon,

the numerical range can be artificially constrained, that is

$$\|\boldsymbol{p}\| \leq 1 \tag{11}$$

When $\|\boldsymbol{p}\| \geq 1$, employing $\hat{\Gamma}^{-1}$ for the pose computation, which can be expressed as

$$\hat{\Gamma}^{-1} = \overline{\boldsymbol{p}}^{-1} - \varepsilon \overline{\boldsymbol{p}}^{-1} \overline{\boldsymbol{b}} \overline{\boldsymbol{p}}^{-1} \tag{12}$$

For any value of twistor, the corresponding dual quaternion will automatically satisfy the normalization constraint, which can be expressed as

$$\hat{q}^* \hat{q} = (1+\hat{\Gamma})^{-1}(1-\hat{\Gamma})(1-\hat{\Gamma})^{-1}(1+\hat{\Gamma}) = 1 \tag{13}$$

Therefore, rigid pose representation is based on twistor without normalization constraint.

The relationship between the twistor, the dual angle $\hat{\alpha}$, and the screw axis $\hat{\boldsymbol{n}}$ can be expressed as:

$$\hat{\Gamma} = \tan\frac{\hat{\alpha}}{4}\hat{n} \tag{14}$$

where $\hat{n}$ is the dual quaternion mapping of dual vector $\hat{\boldsymbol{n}}$. According to Eq. (14), it can be observed that $\hat{\Gamma}$ exhibits a singularity at $\alpha_r = 2\pi$. However, its singularity can be addressed by equivalently treating it as the original pose, which does not affect the pose description.

2.2 Twistor-based octorotor kinematics and dynamics

Dynamic pose tracking of the UAV can be viewed as the relative pose motion of its body frame with respect to the desired coordinate system, with the latter changing over time. Therefore, the coordinate systems applied in this paper are the ground inertial frame $\mathcal{F}_I$, the UAV body-fixed frame $\mathcal{F}_B$, and the desired frame $\mathcal{F}_D$.

The kinematics of dynamic pose with twistor representation can be obtained according to the derivation of the mentioned above dual quaternion equation.

$$\dot{\hat{\Gamma}} = \frac{1}{4}(1+\hat{\Gamma})\hat{V}_{BD}^{B}(1-\hat{\Gamma}) \tag{15}$$

where $\hat{V}_{BD}^{B}$ denotes the dual quaternion mapping of dual vector $\hat{\boldsymbol{V}}_{BD}^{B}$ of the frame $\mathcal{F}_B$ relative to the frame $\mathcal{F}_D$ expressed in frame $\mathcal{F}_B$, which can be calculated as

$$\hat{V}_{BD}^{B} = \hat{V}_{BI}^{B} - (1+\hat{\Gamma})^{-1}(1-\hat{\Gamma})\hat{V}_{DI}^{D}(1-\hat{\Gamma})^{-1}(1+\hat{\Gamma}) \tag{16}$$

where $\hat{V}_{BI}^{B}$ denotes the dual quaternion mapping of dual vector $\hat{\boldsymbol{V}}_{BI}^{B}$ of the frame $\mathcal{F}_B$ relative to the frame $\mathcal{F}_I$ expressed in frame $\mathcal{F}_B$, $\hat{V}_{DI}^{D}$ denotes the dual quaternion mapping of dual vector $\hat{\boldsymbol{V}}_{DI}^{D}$ of the frame $\mathcal{F}_D$ relative to the frame $\mathcal{F}_I$ expressed in frame $\mathcal{F}_D$.

The dynamics of the dynamic pose with twistor representation can be defined as

$$\begin{aligned}\dot{\hat{\boldsymbol{V}}}_{BD}^{B} &= \hat{\boldsymbol{M}}^{-1}\hat{\boldsymbol{F}}_{u}^{B} + \hat{\boldsymbol{M}}^{-1}\hat{\boldsymbol{F}}_{g}^{B} + \hat{\boldsymbol{M}}^{-1}\hat{\boldsymbol{F}}_{d}^{B} \\ &-\hat{\boldsymbol{M}}^{-1}[\hat{\boldsymbol{V}}_{BI}^{B} \times \left(\hat{\boldsymbol{M}}\hat{\boldsymbol{V}}_{BI}^{B}\right)] - \dot{\hat{\boldsymbol{V}}}_{DI}^{B}\end{aligned} \tag{17}$$

where

$$\begin{aligned}\hat{\boldsymbol{M}} &= m\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\boldsymbol{I} + \varepsilon\boldsymbol{J}^{B} \\ \hat{\boldsymbol{F}}_{u}^{B} &= \boldsymbol{f}_{u}^{B} + \varepsilon\boldsymbol{\tau}_{u}^{B} \\ \hat{\boldsymbol{F}}_{g}^{B} &= \boldsymbol{f}_{g}^{B} + \varepsilon\boldsymbol{0} \\ \hat{\boldsymbol{F}}_{d}^{B} &= \boldsymbol{f}_{d}^{B} + \varepsilon\boldsymbol{\tau}_{d}^{B}\end{aligned} \tag{18}$$

where m is the total mass of UAV, $\boldsymbol{J}$ is the inertia matrix of UAV, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{f}$ and $\boldsymbol{\tau}$ are the resultant external force vector and torque vector. Subscripts g, u and d denote wrench of gravity, wrench of control, wrench of disturbance respectively. The operator $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}$ satisfies $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}(a+\varepsilon b) = b$. The inverse of dual mass operator $\hat{\boldsymbol{M}}$ is defined as

$$\hat{\boldsymbol{M}}^{-1} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\boldsymbol{J}^{-1} + \varepsilon m^{-1}\boldsymbol{I} \tag{19}$$

which satisfies

$$\hat{\boldsymbol{M}}^{-1}(\hat{\boldsymbol{M}}\hat{\boldsymbol{V}}) = \hat{\boldsymbol{V}} \tag{20}$$

For the coaxial octorotor UAV, its six-dimensional control force screw is generated by the eight rotors, as shown in Fig. 1. The total thrust can be calculated as

$$T_{all} = C_T \rho D_P^4(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2+\Omega_5^2+\Omega_6^2+\Omega_7^2+\Omega_8^2) \tag{21}$$

where C_T is thrust coefficient, ρ is air density of the environment that UAV located, D_P is diameter of rotor, $\Omega_i, i = 1, 2, ..., 8$ are angular velocity of i-th rotor. The torque of UAV can be calculated as

$$\tau_a = \begin{bmatrix} lC_d(\Omega_3^2+\Omega_4^2-\Omega_7^2-\Omega_8^2) \\ lC_d(\Omega_1^2+\Omega_2^2-\Omega_5^2-\Omega_6^2) \\ C_d(\Omega_1^2+\Omega_4^2+\Omega_5^2+\Omega_8^2-\Omega_2^2-\Omega_3^2-\Omega_6^2-\Omega_7^2) \end{bmatrix} \tag{22}$$

where l is the arm length and C_d is torque coefficient. Define the thrust of each axis is written as $F_i, i = 1, 2, 3, 4$, which

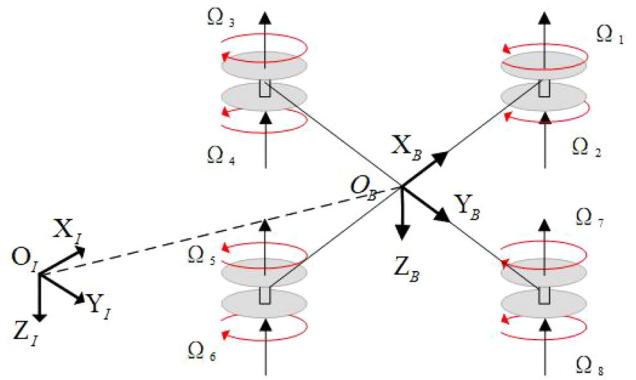


Fig. 1 Schematic diagram of the UAV system

can be expressed as

$$\begin{aligned} F_1 &= C_T \rho D_P^4 (\Omega_1^2 - \Omega_2^2) \\ F_2 &= C_T \rho D_P^4 (\Omega_3^2 - \Omega_4^2) \\ F_3 &= C_T \rho D_P^4 (\Omega_5^2 - \Omega_6^2) \\ F_4 &= C_T \rho D_P^4 (\Omega_7^2 - \Omega_8^2) \end{aligned} \tag{23}$$

As the UAV adopts "X" configuration, the transformation of the three-dimensional forces and the thrust generated by four rotors can be expressed as

$$\boldsymbol{f}_u^B = \frac{1}{\sqrt{2}} \begin{bmatrix} l & -l & l & -l \\ l & l & -l & -l \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{24}$$

The τ_a can be written as

$$\boldsymbol{\tau}_u^B = \begin{bmatrix} 0 & l & 0 & -l \\ l & 0 & -l & 0 \\ C_M & -C_M & C_M & -C_M \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{25}$$

where C_M denotes the moment coefficient. Therefore, the matrix integration of Eqs. (24) and (25) can be obtained as

$$\begin{bmatrix} \boldsymbol{f}_u^B \\ \boldsymbol{\tau}_u^B \end{bmatrix} = \begin{bmatrix} \frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} & \frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} \\ \frac{L}{\sqrt{2}} & \frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} & -\frac{L}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & L & 0 & -L \\ L & 0 & -L & 0 \\ c_M & -c_M & c_M & -c_M \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{26}$$

By performing the inverse solution of Eq. (26), the thrust generated by the four axes can be obtained, which further allows for the determination of the rotational speed of each of the eight rotors.

In summary, the relative dynamic and kinematics can be obtained as

$$\begin{cases} \dot{\hat{\Gamma}}_{BD} = \frac{1}{4}(1 + \hat{\Gamma}_{BD}) \hat{V}_{BD}^B \left(1 - \hat{\Gamma}_{BD}\right) \\ \dot{\hat{V}}_{BD}^B = \hat{\boldsymbol{M}}^{-1} \hat{\boldsymbol{F}}_u^B + \hat{\boldsymbol{M}}^{-1} \hat{\boldsymbol{F}}_g^B \\ + \hat{\boldsymbol{M}}^{-1} \hat{\boldsymbol{F}}_d^B - \hat{\boldsymbol{M}}^{-1} [\hat{V}_{BI}^B \times \left(\hat{\boldsymbol{M}} \hat{V}_{BI}^B\right)] - \dot{\hat{V}}_{DI}^B \end{cases} \tag{27}$$

To compare the impact of different algebraic systems on runtime average memory usage and peak memory usage, numerical simulations were conducted for controllers based on the Newton–Euler method, dual quaternions, and twistor algebra under identical control parameters. The simulation results are presented in Table1.

As can be seen from Table 1, due to the presence of iterative computations, the Newton–Euler method exhibits the highest memory usage, while twistor consumes the least. Since the operational functions based on dual quaternions are all custom-defined, their runtime optimization is insufficient. Consequently, the controller under the dual quaternion algebra requires a longer runtime compared to the Newton–Euler-based method. In contrast, the twistor approach adopted in this study eliminates redundancy and normalization constraints, resulting in a shorter runtime than the other methods.

3 Finite-time controller design

This paper adopts dynamic desired pose tracking as the task objective and considers external disturbances and measurement noise. Therefore, it is necessary to design a controller that can quickly converge within a finite time and possesses a certain level of robustness.

As indicated in [27], hyperbolic function-based sliding mode control can effectively mitigate the chattering caused by switching sliding surfaces and improve the dynamic response performance of the controller. Therefore, this paper combines hyperbolic functions for the design of a finite-time controller.

The linear sliding mode (LSM) surface reaching law [26] can be expressed as

$$\hat{\boldsymbol{s}} = \hat{\boldsymbol{V}}^B + \hat{4} \hat{\boldsymbol{K}}_1 \hat{\boldsymbol{E}} ((1 - \hat{\boldsymbol{\Gamma}}^2)^{-1} \hat{\boldsymbol{\Gamma}})_v \tag{28}$$

$$\dot{\hat{\boldsymbol{s}}} = -\hat{\boldsymbol{K}}_2 \mathrm{sgn}(\hat{\boldsymbol{s}}) \tag{29}$$

The operation of $\hat{\boldsymbol{E}}$ can be expressed as

$$\hat{\boldsymbol{E}}(\boldsymbol{a}_r + \varepsilon \boldsymbol{a}_d) = \boldsymbol{a}_d + \varepsilon \boldsymbol{a}_r \tag{30}$$

In order to compare with the LSM, the finite-time double-hyperbolic sliding mode (FDSM) controller proposed in this

Table 1 The runtime and memory of different algebraic systems

Algebraic system	Time (ms)	Avg Mem (KB)	Peak Mem (KB)
Dual quaternion	239.810	0.8	4
Newton–Euler	238.127	1.6	22
Twistor	237.411	0.2	4

paper adopts the same sliding surface. The reaching law of FDSM is designed as

$$\dot{\hat{s}} = -\hat{\boldsymbol{K}}_2|\hat{s}|^{\alpha}\tanh(a\hat{s}) - \hat{\boldsymbol{K}}_3|\hat{s}|^{\alpha}\cosh(b\hat{s}) \tag{31}$$

where $0 < \alpha < 1$; the hyperbolic tangent function is $\tanh(a\hat{s}) = (e^{a\hat{s}} - e^{-a\hat{s}})/(e^{a\hat{s}} + e^{-a\hat{s}})$, hyperbolic cosine function is $\cosh(b\hat{s}) = (e^{b\hat{s}} - e^{-b\hat{s}})/2$; $\hat{\boldsymbol{K}}_1$, $\hat{\boldsymbol{K}}_2$ and $\hat{\boldsymbol{K}}_3$ are dual coefficient operator which defined as

$$\hat{\boldsymbol{K}}_i = k_i \boldsymbol{I}\frac{\mathrm{d}}{\mathrm{d}\varepsilon} + k_i \boldsymbol{I}\varepsilon, i = 1, 2, 3 \tag{32}$$

By substituting Eq. (17) in Eq. (32), the control wrench can be written as

$$\begin{aligned}\hat{\boldsymbol{F}}_u^B &= -\hat{\boldsymbol{M}}\hat{\boldsymbol{K}}_2\hat{\boldsymbol{E}}|\hat{s}|^{\alpha}\tanh(a\hat{s}) - \hat{\boldsymbol{M}}\hat{\boldsymbol{K}}_3\hat{\boldsymbol{E}}|\hat{s}|^{\alpha}\cosh(b\hat{s}) \\ &+\hat{\boldsymbol{V}}_{BI}^B \times \left(\hat{\boldsymbol{M}}\hat{\boldsymbol{V}}_{BI}^B\right) - 4\hat{\boldsymbol{M}}\hat{\boldsymbol{K}}_1\hat{\boldsymbol{E}}\left(1-\hat{\boldsymbol{\Gamma}}^2\right)^{-2}\left(\dot{\hat{\boldsymbol{\Gamma}}} + \hat{\boldsymbol{\Gamma}}\dot{\hat{\boldsymbol{\Gamma}}}\hat{\boldsymbol{\Gamma}}\right)_v \\ &-\hat{\boldsymbol{F}}_g^B - \hat{\boldsymbol{F}}_d^B - \hat{\boldsymbol{M}}\hat{\boldsymbol{V}}_{BI}^B + \hat{\boldsymbol{M}}\dot{\hat{\boldsymbol{V}}}_{DI}^B\end{aligned} \tag{33}$$

Define a new operation as follows

$$\hat{\boldsymbol{a}} \odot \hat{\boldsymbol{b}} = \boldsymbol{a}_r \cdot \boldsymbol{b}_d + \boldsymbol{a}_d \cdot \boldsymbol{b}_r \tag{34}$$

To prove that the controller can achieve global finite-time stability, it is necessary to demonstrate that the controller can converge within a finite time during both the reaching phase and the sliding phase. For reaching phase, the Lyapunov function is chosen as

$$V_1 = \frac{1}{2}\hat{\boldsymbol{s}} \odot \left(\hat{\boldsymbol{E}}\hat{\boldsymbol{s}}\right) \tag{35}$$

The dual part form of Eq. (35) can be written as

$$V_1 = \frac{1}{2}(|\boldsymbol{s}_r|^2 + |\boldsymbol{s}_d|^2) \geq 0 \tag{36}$$

According to Eq. (36), it can be seen that it is positive semi-definite and exhibits radial unboundedness. Furthermore, $V_1 = 0$ if and only when $\hat{\boldsymbol{s}} = 0$, which indicates that the closed-loop system attains the equilibrium point of the reaching phase under the condition that $\hat{\boldsymbol{s}} = 0$. By taking the derivative and substituting it in Eq. (31), we obtain

$$\begin{aligned}\dot{V}_1 &= \dot{\hat{\boldsymbol{s}}} \odot (\hat{\boldsymbol{E}}\hat{\boldsymbol{s}}) \\ &= -(k_2|\hat{s}|^{\alpha}\tanh(a\hat{s}) + k_3\left|\hat{\boldsymbol{s}}\right| \cdot |\hat{s}|^{\alpha}\cosh(b\hat{s})) \odot (\hat{\boldsymbol{E}}\hat{\boldsymbol{s}}) \\ &= -k_2|\hat{s}|^{\alpha}\tanh(a\boldsymbol{s}_r)\cdot\boldsymbol{s}_r - k_2|\hat{s}|^{\alpha}\tanh(a\boldsymbol{s}_d)\cdot\boldsymbol{s}_d \\ &\quad - k_3|\hat{s}|^{\alpha}\cosh(b\boldsymbol{s}_r)\cdot\boldsymbol{s}_r - k_3|\hat{s}|^{\alpha}\cosh(b\boldsymbol{s}_d)\cdot\boldsymbol{s}_d\end{aligned} \tag{37}$$

According to Eq. (37), $\dot{V}_1 = 0$ if and only when $\boldsymbol{s} = 0$. In accordance with LaSalle's invariance principle, the reaching phase is proven to be globally asymptotically stable at the given equilibrium point. Equation (37) satisfies

$$\dot{V}_1 \leq -\lambda\left|\widehat{s}\right|^{1+\alpha} = -\lambda\left(|s_r|^{1+\alpha} + \varepsilon|s_d|^{1+\alpha}\right) \tag{38}$$

which can be written as

$$\dot{V}_1 + \lambda\dot{V}_1^{\frac{1+\alpha}{2}} \leq 0 \tag{39}$$

According to the Lyapunov finite-time stability criterion, it can be concluded that the system can converge to the sliding surface within a finite time during the reaching phase. The upper bound of the reaching phase stability time can be derived from the following integral equation

$$\int_{V_1}^{0}\frac{dV_1}{V_1^{\frac{1+\alpha}{2}}} \leq -\lambda\int_{t_0}^{T}dt \tag{40}$$

By evaluating the integral of the above expression, the following result can be obtained

$$\frac{2}{1-\alpha}\left(V_1^{\frac{1-\alpha}{2}}(t_0)\right) \leq \lambda\,(T - t_0) \tag{41}$$

Therefore, the reaching phase stability time upper bound is

$$T \leq \frac{2}{\lambda\,(1-\alpha)}V_1^{\frac{1-\alpha}{2}}(t_0) \tag{42}$$

For sliding phase, the Lyapunov function is chosen as

$$V_2 = \frac{1}{2}\hat{\boldsymbol{\Gamma}} \odot \left(\hat{\boldsymbol{E}}\hat{\boldsymbol{\Gamma}}\right) \tag{43}$$

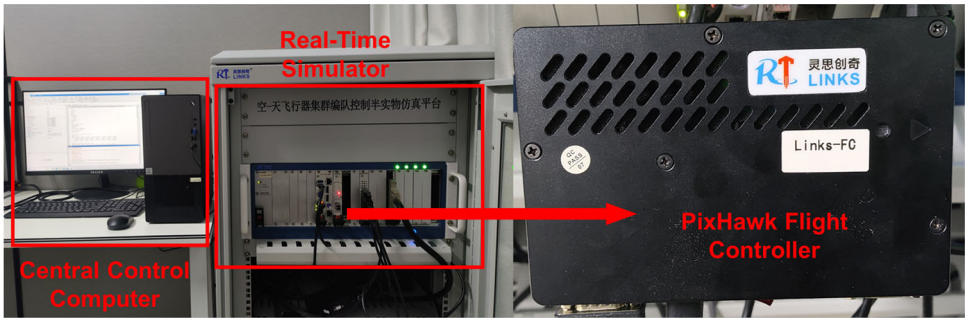


Fig. 2 Links-RT hardware-in-the-loop simulation platform

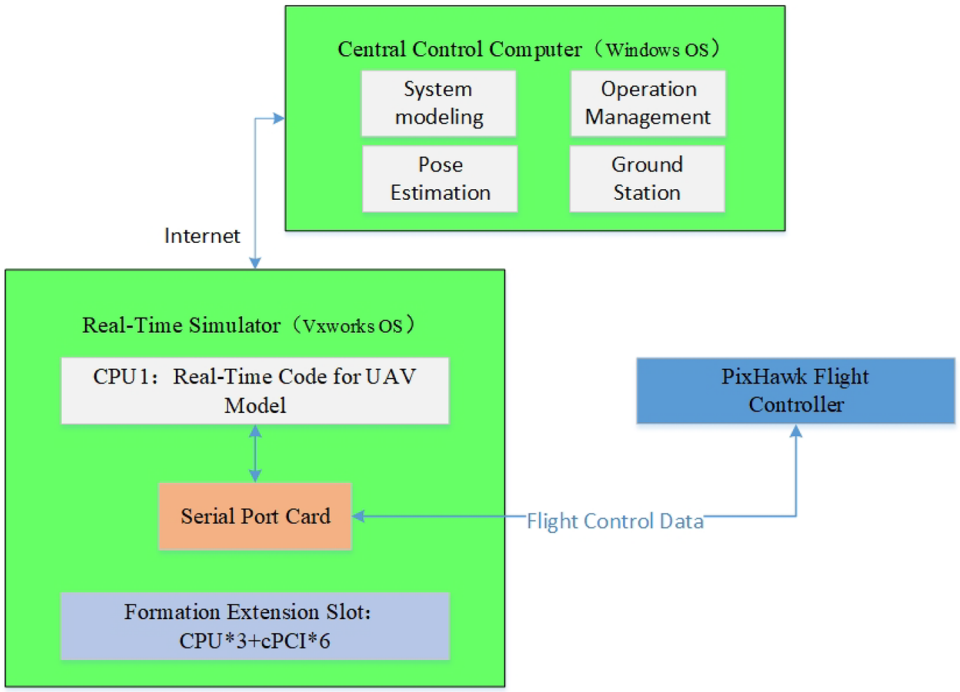


Fig. 3 Links-RT System Overall Architecture Diagram

According to Equation (43), $V_2 = 0$ if and only when $\hat{\boldsymbol{B}} = 0$. This implies that the closed-loop system reaches the equilibrium point of the reaching phase when $\hat{\boldsymbol{B}} = 0$. By taking the derivative, we obtain

$$\begin{aligned}\dot{V}_2 &= \dot{\hat{\boldsymbol{\Gamma}}} \odot \left(\hat{\boldsymbol{E}}\hat{\boldsymbol{\Gamma}}\right) \\ &= -\left(\frac{k_1}{1-\hat{B}\hat{B}}(1+\hat{B})\hat{B}(1-\hat{B})\right)_v \odot \left(\hat{\boldsymbol{E}}\hat{\boldsymbol{B}}\right) \\ &= -k_1(|\sigma|^2 + |b|^2) \leq -\kappa k_1 2^{\frac{\alpha-3}{2}} V_2^{\frac{\alpha+1}{2}} \leq 0\end{aligned} \tag{44}$$

where $0 < \kappa < 1$. Based on Equation (44), $\dot{V}_2 = 0$ if and only when $\hat{\boldsymbol{B}} = 0$. At this point, only $\hat{\boldsymbol{B}} = 0$, $\hat{\boldsymbol{V}}^B = 0$ are the solutions of the closed-loop system. According to LaSalle's invariance principle, it can be concluded that the sliding phase exhibits global asymptotic stability at the point $\hat{\boldsymbol{B}} = 0$, $\hat{\boldsymbol{V}}^B = 0$. Consequently, the global asymptotic stability of the closed-loop system's pose motion can be ensured by implementing hyperbolic sliding mode control utilizing the twistor framework. Equation (44) can be written as

$$\dot{V}_2 + \kappa k_1 2^{\frac{\alpha-3}{2}} V_2^{\frac{\alpha+1}{2}} \leq 0 \tag{45}$$

According to the Lyapunov finite-time stability criterion, it can be concluded that the system can converge to zero within a finite time during the sliding phase. The upper bound of the sliding phase stability time can be calculated as

$$T_2 \leq \frac{V_2^{\frac{\alpha+1}{2}}(t_0)}{\kappa k_1 2^{\frac{\alpha-3}{2}}(1-\alpha)} \tag{46}$$

Therefore, the total converge time taken to reach the origin of the system can be obtained as

$$T_{all} = \frac{2}{\lambda(1-\alpha)} V_1^{\frac{1-\alpha}{2}}(t_0) + \frac{V_2^{\frac{\alpha+1}{2}}(t_0)}{\kappa k_1 2^{\frac{\alpha-3}{2}}(1-\alpha)} \tag{47}$$

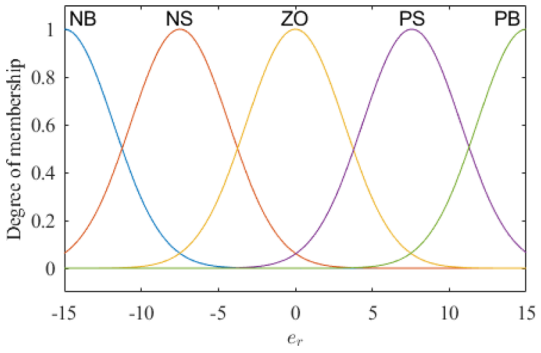


Fig. 4 Membership function

The above derivation indicates that, regardless of the initial state, the system will converge to the equilibrium state within a finite time.

4 Simulation

4.1 Initial parameters

The simulation in this paper is conducted using the semi-hardware-in-the-loop platform Links-RT from Beijing Links Co., Ltd., as shown in Fig. 2.

The Links-RT hardware-in-the-loop simulation platform adopts an upper-lower computer system architecture, as shown in Fig. 3. The upper computer is a Windows-based development and simulation host, while the lower computer is a real-time simulation machine with an integrated CPU. It runs real-time code for the aircraft dynamics and atmospheric environment models and connects to the flight control board through I/O channels.

To validate the effectiveness of the proposed method, simulations are conducted using Links-RT to compare the FDSM, LSM controller, and PD controller, with the simulation parameters listed in Table 3. The details of the PD control law can be found in [14]. To compare with more advanced controllers, we have also included a fuzzy sliding mode (FSM) controller for comparison. The FSM reaching law can be written as

$$\dot{\hat{s}} = -\hat{\boldsymbol{K}}_2 \kappa \mathrm{sgn}(\hat{s}) \tag{48}$$

The membership function and fuzzy adjustment rules are shown in Fig. 2 and Table 4.

The desired dynamic position is set as

$$\boldsymbol{r}_{BD} = \begin{bmatrix} \cos(0.3t) \\ \sin(0.3t) \\ 0.1 + t \end{bmatrix} \mathrm{m} \tag{49}$$

Table 2 Fuzzy adjustment rules

λ		e_r				
		NB	NS	ZO	PS	PB
$\hat{\boldsymbol{V}}_{BD}^{B}$	NB	PB	PB	PS	PS	PB
	NS	PB	PS	ZO	PS	PB
	ZO	PB	PS	ZO	PS	PB
	PS	PB	PS	ZO	NS	PB
	PB	PB	PS	ZO	NS	PB

The desired dynamic attitude quaternion is set as

$$q_{BD} = \begin{bmatrix} \cos\left(\frac{\pi t}{10}\right) \\ \sin\left(\frac{\pi t}{10}\right) \\ 0 \\ 0 \end{bmatrix} \tag{50}$$

The attitude quaternion can be converted to MRP using the following equation:

$$\boldsymbol{p} = \left((q+1)^{-1}(q-1)\right)_v \tag{51}$$

To validate the rationality of the parameters selected in Table 3, a parameter sensitivity analysis was conducted for the FDSM. In the simulations, only one parameter was altered at a time from the original set, and the resulting overshoot (OS), settling time (ST), steady-state error (SSE), average control force variance (AFV), and average control torque variance(ATV) for both position and attitude were recorded. The simulation results are presented in Table 4. The"N/A" in Table 4 indicates that the position/attitude did not reach a stable state.

According to Table 4, it can be seen that the original parameters demonstrate superior performance compared to other configurations, as it achieves an optimal balance between OS, ST, SSE, AFV and ATV. Specifically, the original parameters result in zero overshoot for both position and attitude control, a relatively short settling time, and a low steady-state error. Additionally, the average control force variance and control torque variance remain within acceptable ranges, indicating minimal fluctuations in control inputs. In contrast, increasing k_1 reduces steady-state error but introduces higher overshoot and control force variance; higher k_2 values improve some aspects of control but lead to incomplete performance data, suggesting potential instability; larger k_3 values maintain zero overshoot but significantly prolong settling time, indicating slower response. Adjusting a has minimal impact on performance, while lower values of b slightly reduce control torque variance at the cost of increased settling time, and higher values degrade torque control performance. These trade-offs highlight the robustness of the original parameters, which deliver balanced performance

Table 3 Simulation conditions

Variable name	Initial value
Octorotor mass	$m_c = 1.282\text{kg}$
Octorotor rotational inertia	$\boldsymbol{J}_c = \text{diag}[\,4.586\ 4.856\ 8.801\,]^{-3}\text{kg}\cdot\text{m}^2$
Torque coefficient	$C_M = 0.01129$
Initial position	$\boldsymbol{r}_{B0} = [\,0\ 0\ 0\,]^T\text{m}$
Initial attitude MRP	$\boldsymbol{p}_{B0} = \left[\,00.1346\ -0.1567\ 0.0951\,\right]^T$
FDSM parameter	$k_1 = 5, k_2 = 35, k_3 = 10, a = 10, b = 0.1$
LSM parameters	$k_1 = 5\ k_2 = 35, \sigma = 0.07$
PD parameters	$k_p = 20, k_d = 10$
Control force limitation	$f_{max} = \pm 100\ \text{N}$
Control torque limitation	$\tau_{max} = \pm 1\ \text{N}\cdot\text{m}$

Table 4 Performance comparison under different parameters

Parameters	Position				Attitude		
	OS (m)	ST (s)	SSE (m)	AFV	OS	ST (s)	ATV ($\times 10^{-3}$)
Original	0	0.79	0.097	183.8312	0	1.928	0.95311
$k_1 = 1$	0	N/A	N/A	71.8077	0	N/A	0.47499
$k_1 = 10$	0.8127	0.74	0.0195	344.1537	0	1.464	2.7732
$k_2 = 10$	0.2528	N/A	N/A	152.0546	0	N/A	0.58381
$k_2 = 50$	0	0.83	0.095	198.3017	0	2.28	1.2533
$k_3 = 1$	0	0.79	0.097	183.7017	0	2.707	0.92161
$k_3 = 10$	0	0.79	0.098	183.9842	0	4.14	0.99528
$a = 20$	0	0.79	0.099	184.6602	0	1.753	0.95424
$a = 1$	0	N/A	N/A	177.943	0.0221	N/A	0.89316
$b = 0.01$	0	0.79	0.097	183.6855	0.0326	4.14	0.91777
$b = 1$	0	0.79	0.098	184.6391	0.0234	4.14	2.8477

across all metrics without significant compromises, making them a well-performing configuration for stable and precise control in the tested system. However, further exploration is needed to identify potential parameter sets that may achieve even better performance.

4.2 Results and analysis

This section compares the proposed FDSM with the LSM and PD controllers. To accurately simulate real-world conditions, measurement noise is introduced into the simulation. The simulation results are shown from Fig. 5 to Fig. 9. The variance of the control outputs for the three controllers is computed to assess their stability, with the results shown in Tables 5 and 6. A smaller variance in control output signifies reduced control cost and improved control efficiency, as it indicates more precise regulation around the desired output, which is ideally zero.

The horizontal coordinate in the above figures represents time, while the vertical coordinate represents the relative position and velocity of the tracking octorotor with respect to the measured pose point at the current moment in the tracking

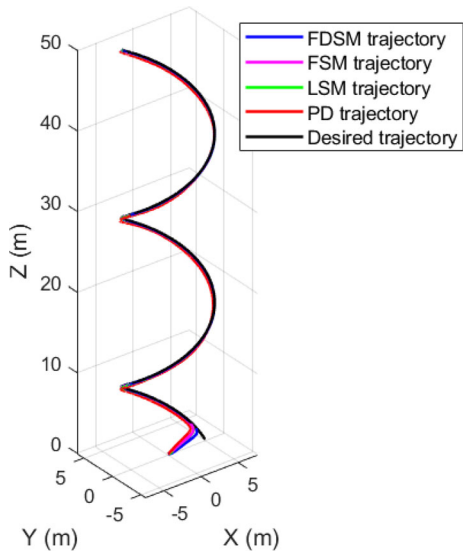


Fig. 5 Flight trajectory

octorotor UAV frame. Figures 6 and 7 illustrate the relative position and relative attitude MRP controlled by different controllers. The UAV's position control is achieved through

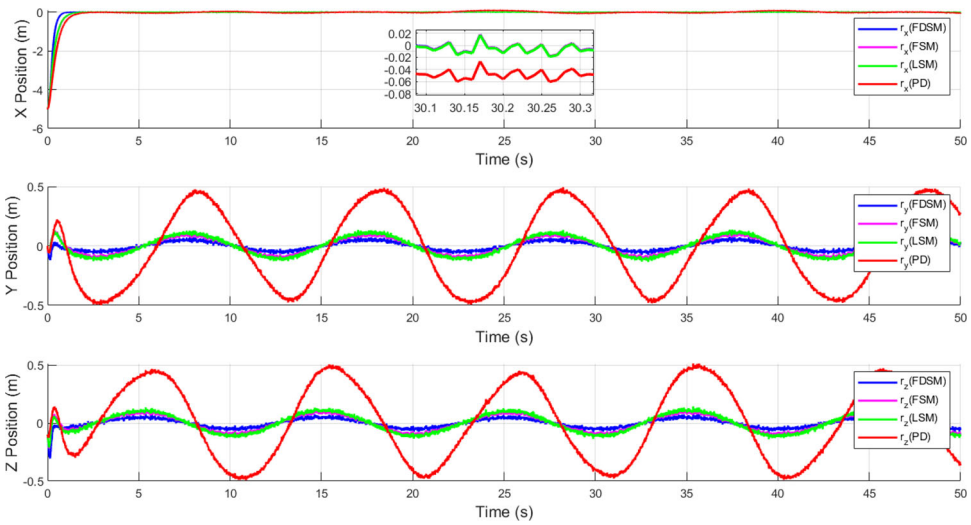


Fig. 6 Relative position expressed in frame $\mathcal{F}_B$

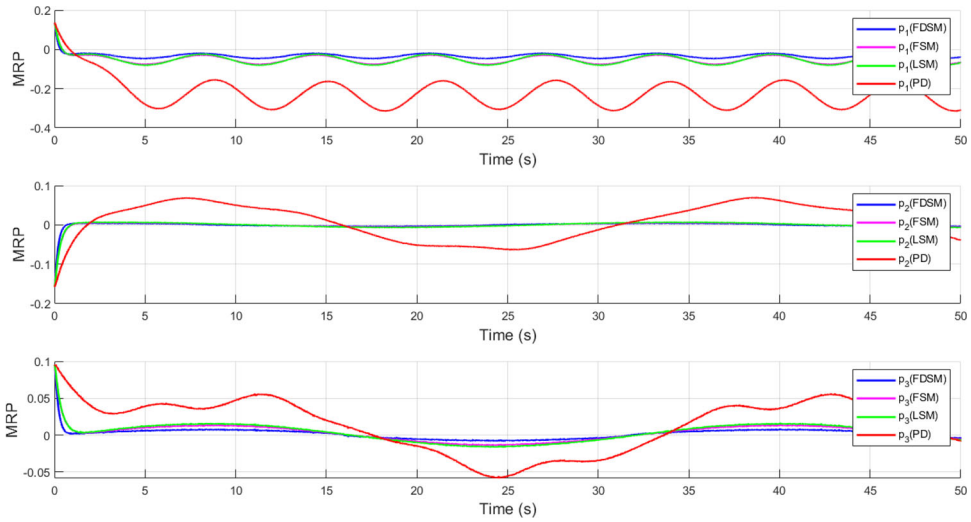


Fig. 7 Relative MRP expressed in frame $\mathcal{F}_B$

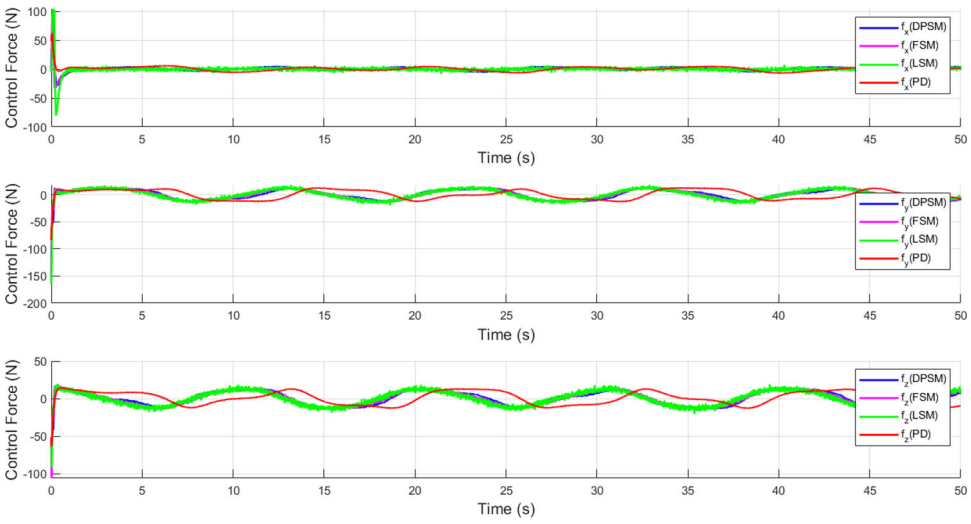


Fig. 8 Control force expressed in frame $\mathcal{F}_B$

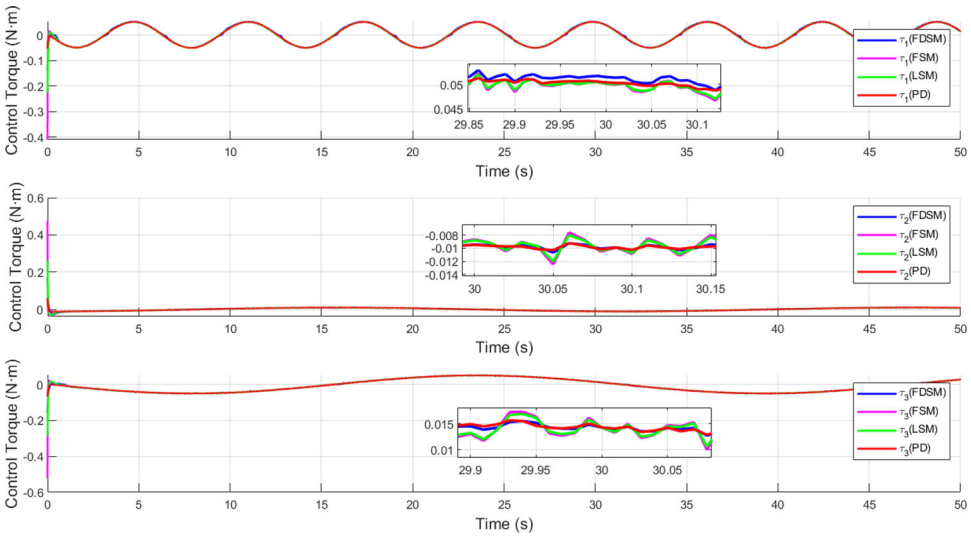
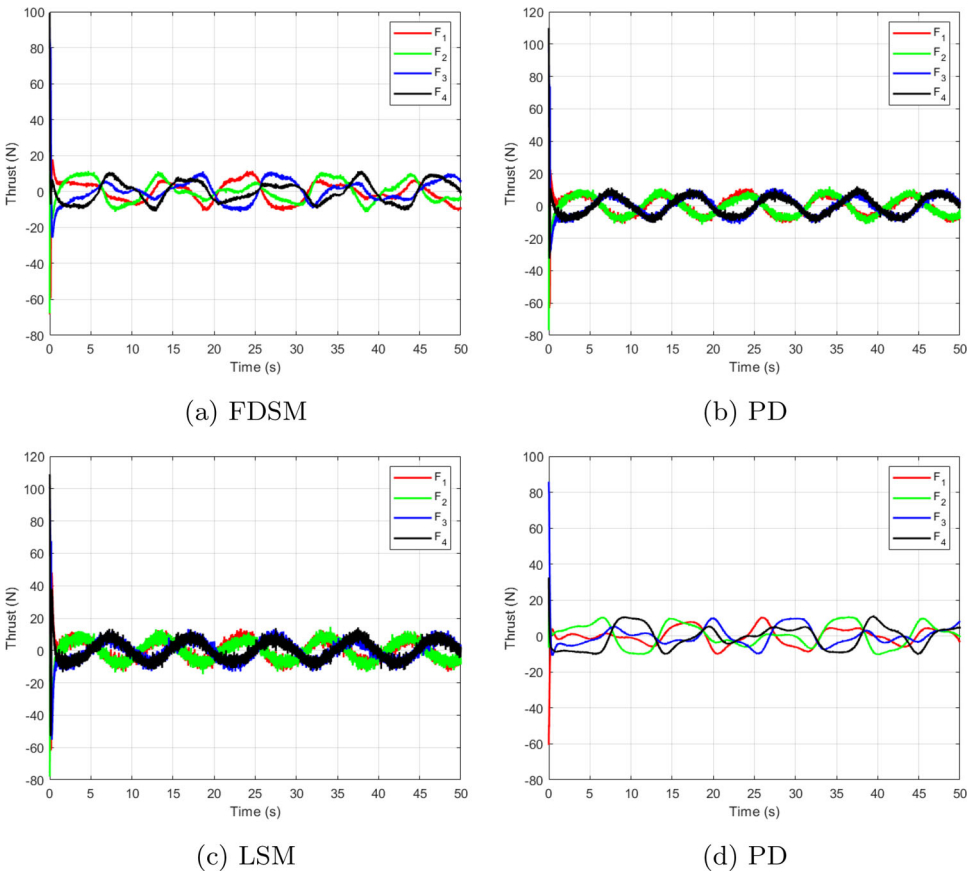


Fig. 9 Control torque expressed in frame $\mathcal{F}_B$



(a) FDSM (b) PD

(c) LSM (d) PD

Fig. 10 Comparison of thrust on each axis

Table 5 Variance of control force

Controller	f_x	f_y	f_z
FDSM	28.8702	87.7253	86.0709
FSM	27.0050	94.1056	89.7324
LSM	55.3390	105.1723	92.0300
PD	28.4895	63.7053	83.8370

Table 6 Variance of control torque

Controller	τ_1	τ_2	τ_3
FDSM	0.00132	0.00007	0.00112
FSM	0.00132	0.00011	0.00118
LSM	0.00134	0.00015	0.00115
PD	0.00129	0.00006	0.00006

the combination of attitude and thrust. Thus, dynamic attitude tracking has a significant impact on position tracking performance. It can be observed that the PD controller exhibits poor control performance, resulting in the largest oscillations. The LSM controller and FSM controller are more stable than the PD controller, though oscillations still occur in the position tracking curve. Although the FSM improves robustness and control performance to some extent by optimizing parameters using fuzzy methods, it still exhibits oscillation issues similar to the LSM due to the limitations of its reaching law. In comparison to the PD, LSM and FSM controllers, the proposed FDSM controller achieves faster position convergence with minimal oscillation. Regarding attitude, the dynamic response performance of the controllers determines the magnitude of the steady-state error. Notably, the proposed FDSM controller yields the smallest steady-state error, demonstrating superior dynamic performance. Based on Fig. 5 and the analysis above, the flight trajectory achieved under the FDSM controller proposed in this paper aligns more closely with the expected trajectory. Furthermore, the FDSM controller demonstrates faster convergence speed and achieves finite-time convergence.

Figures 8 and 9 show the time-varying control outputs of the three controllers. Figure 10 shows the variation curves of lift on each axis of the coaxial octorotor UAV under different controllers. As observed from the figures, the control outputs of all three controllers are similar. Due to measurement noise, the controller outputs exist oscillation. According to Tables 5 and 6, the PD controller's output is the smoothest. Nonetheless, the PD controller demonstrates insufficient dynamic response capability, and its output fails to adjust promptly for effective tracking. The output fluctuation of the FSM is smaller than that of the LSM but larger than that of the FDSM. In comparison with the FDSM controller proposed in this paper, the LSM controller and FSM controller exhibit more pronounced oscillations in its control output. This is because the LSM controller and FSM controller utilize linear reaching law, which does not adjust its parameter size in real time according to the system's state. Consequently, while the boundary layer method is employed to reduce oscillations, the inherent nature of its reaching law still leads to some oscillations in the control output. In contrast, the proposed FDSM controller employs the hyperbolic tangent function to ensure pseudo-linear convergence of the sliding mode variable to zero, thus preventing oscillations and offering enhanced robustness and dynamic response performance. Based on the analysis and simulation results, it can be concluded that the FDSM controller better balances sliding mode invariance and disturbance rejection, achieving faster dynamic response with smoother control outputs.

In conclusion, the FDSM proposed in this paper can effectively track dynamic pose and converge within a finite time. Furthermore, compared to the LSM and FSM, the proposed controller, which is based on a hyperbolic trigonometric function reaching law, effectively avoids oscillations, demonstrates better dynamic response, and exhibits certain universality.

5 Conclusion

This paper introduces the novel application of twistor for the dynamic modeling of coaxial octorotors, effectively addressing parameter redundancy and normalization constraints. Furthermore, a finite-time double-hyperbolic sliding mode controller is designed based on twistor for dynamic pose tracking. The RT-Links hardware-in-the-loop simulation platform is employed to validate the proposed method. The simulation results demonstrate that the twistor-based FDSM controller for coaxial octorotors effectively tracks dynamic pose targets, converging within a finite time and exhibiting robust disturbance rejection and dynamic response performance.

However, the simulation results do not fully represent the practical performance of the proposed method. The hardware limitations of the UAV have not been strictly considered in this study; therefore, the simulation results should be regarded as a reference for future physical experiments.

Acknowledgements Not applicable.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [Jiaxing Zhou], [Wei Chen],[Yifan Deng] and [Qing Li]. The first draft of the manuscript was written by [Wei Chen] and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript. Conceptualization: [Jiaxing Zhou], [Wei Chen],[Yifan Deng] and [Qing Li]; Methodology: [Yifan Deng] ; Formal analysis and investigation: [Wei Chen] ; Writing - original draft preparation: [Wei Chen]; Writing-review and editing: [Jiaxing Zhou], [Wei Chen] ; Funding acquisition: [Jiaxing Zhou], [Qing Li]; Resources:[Yifan Deng] ; Supervision: [Jiaxing Zhou], [Yifan Deng].

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

Code availability Not applicable. https://github.com/joonhap/spMCMC.

Declarations

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

References

1. Singh BK, Kumar A (2023) Attitude and position control with minimum snap trajectory planning for quadrotor UAV. Int J Dynam Control 11:2342–2353

2. Saied M, Shraim H, Francis C (2024) A review on recent development of multirotor uav fault-tolerant control systems. IEEE Aerosp Electron Syst Mag 39:146–180
3. Marshall JA, Sun W, L'Afflitto A (2021) A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems. Annu Rev Control 52:390–427
4. Abera NB, Abdissa CM, Lemma LN (2024) An improved non-singular adaptive super twisting sliding mode controller for quadcopter. PLoS ONE 19:1–28
5. Madebo MM, Abdissa CM, Lemma LN, Negash DS (2024) Robust tracking control for quadrotor uav with external disturbances and uncertainties using neural network based mrac. IEEE Access 12:36183–36201
6. Gedefaw EA, Abdissa CM, Lemma LN (2024) An improved trajectory tracking control of quadcopter using a novel sliding mode control with fuzzy pid surface. PLoS ONE 19:1–36
7. Wanore Madebo N, Merga Abdissa C, Negash Lemma L (2024) Enhanced trajectory control of quadrotor UAV using fuzzy pid based recurrent neural network controller. IEEE Access 12:190454–190469
8. Saied M, Shraim H, Lussier B, Fantoni I, Francis C (2017) Local controllability and attitude stabilization of multirotor uavs: validation on a coaxial octorotor. Robot Auton Syst 91:128–138
9. Derakhshan RE, Danesh M, Moosavi H (2024) Disturbance observer-based model predictive control of a coaxial octorotor with variable centre of gravity. IET Control Theory Appl 18:764–783
10. Hassani H, Mansouri A, Ahaitouf A (2023) Backstepping-based supertwisting sliding mode attitude control for a quadrotor aircraft subjected to wind disturbances: experimental validation. Int J Dynam Control 11:1285–1296
11. Ali R et al (2019) Adaptive backstepping sliding mode control of coaxial octorotor unmanned aerial vehicle. IEEE Access 7:27526–27534
12. Michieletto G, Lissandrini N, Antonello A, Antonello R, Cenedese A (2020) Dual quaternion delay compensating maneuver regulation for fully actuated uavs. IFAC-Papers Line 53:9316–9321
13. Lee B, Sung S (2024) A study on dual quaternion based cooperative relative navigation of multiple uavs with monocular vision-inertial integration. Chin J Aeronaut 37:335–354
14. Deng Y, Wang Z (2016) Modeling and control for spacecraft relative pose motion by using twistor representation. J Guid Control Dyn 39:1147–1154
15. Abbas N, Rehman KU (2022) Theoretical study of non-newtonian micropolar nanofluid flow over an exponentially stretching surface with free stream velocity. Adv Mech Eng 14(7):16878132221107790
16. Shatanawi W, Abbas N, Shatnawi TAM, Hasan F (2023) Heat and mass transfer of generalized fourier and fick's law for second-grade fluid flow at slendering vertical riga sheet. Heliyon 9:e14250
17. Shatnawi N (2024) Velocity and thermal slips impact on boundary layer flow of micropolar nanofluid over a vertical nonlinear stretched riga sheet. Proceed Institut Mech Eng Part N J Nanomater Nanoeng Nanosyst 238:107–117
18. Nazir A, Abbas N, Shatanawi W (2023) On stability analysis of a mathematical model of a society confronting with internal extremism. Int J Mod Phys B 37:2350065
19. Zeghlache S, Mekki H, Bouguerra A, Djerioui A (2018) Actuator fault tolerant control using adaptive rbfnn fuzzy sliding mode controller for coaxial octorotor uav. ISA Trans 80:267–278
20. Sanwale J, Dahiya S, Trivedi P, Kothari M (2023) Robust fault-tolerant adaptive integral dynamic sliding mode control using finite-time disturbance observer for coaxial octorotor uavs. Control Eng Pract 135:105495
21. Huang D, Huang T, Qin N, Li Y, Yang Y (2022) Finite-time control for a uav system based on finite-time disturbance observer. Aerosp Sci Technol 129:107825
22. Lv Z, Zhao Q, Sun X-M, Wu Y (2024) Finite-time control design for a coaxial tilt-rotor uav. IEEE Trans Industr Electron 71:16132–16142
23. Boning Li MC, Qi S (2024) Finite-time trajectory tracking control of quadrotor uavs based on neural network disturbance observer and command filter. Int J Syst Sci 56:1–15
24. Tao L, Chen Q, Nan Y, Wu C (2018) Double hyperbolic reaching law with chattering-free and fast convergence. IEEE Access 6:27717–27725
25. Xu L, Shao X, Zhang W (2021) Usde-based continuous sliding mode control for quadrotor attitude regulation: method and application. IEEE Access 9:64153–64164
26. Li Q, Liu L, Deng Y, Tang S, Zhao Y (2018) Twistor-based synchronous sliding mode control of spacecraft attitude and position. Chin J Aeronaut 31:1153–1164
27. Xu L, Shao X, Zhang W (2021) Usde-based continuous sliding mode control for quadrotor attitude regulation: method and application. IEEE Access 9:64153–64164