AEROSPACE RESEARCH COMMUNICATIONS
ORIGINAL RESEARCH
published: 27 November 2025
doi: 10.3389/arc.2025.15466

A Minimum Snap Flight Transition Strategy for Quadrotor Tail-Sitter UAVs: Altitude-Hold Transition

*Mingyue Fan and Yifan Xia**

School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, Zhejiang, China

This study examines the altitude-hold flight mode transition problem for tail-sitter unmanned aerial vehicles (UAVs) between hovering and level flight. The problem is significantly challenging because of the complicated nonlinear aerodynamics of tail-sitter UAVs during the transition process. To address this problem, this article proposes a minimum snap trajectory generation method and a model predictive control (MPC)-based tracking strategy. First, the generated trajectories are highly applicable, satisfying the dynamic constraint. Second, MPC tracks full states by solving a finite-horizon optimization problem at each step, yielding the optimal future behavior based on the system model. A numerical simulation was conducted in which the altitude changed by less than 1.4 m during the entire transition process.

Keywords: autonomous aerial vehicles, altitude-hold transition, vehicle dynamics, flight mode transition control, tail-sitter UAVs

INTRODUCTION

Unmanned aerial vehicles (UAVs) with Vertical Take-off and Landing (VTOL) differ from fixed-wing UAVs in that they do not require a runway for takeoff and landing, and possess the ability to hover in mid-air. This versatility enables their wide application across various fields [1–4]. A tail-sitter is a hybrid UAV that combines the agility of a VTOL with the endurance of a fixed-wing aircraft, allowing for longer flights and heavier payloads. In tail-sitter UAVs, thrust is provided exclusively by propellers. The predominant propulsion architectures comprise single- [5, 6], dual- [7, 8], and quad-propeller [9, 10] configurations, which exhibit significant differences in flight-dynamics characteristics, control strategies, and system complexity. The single-propeller configuration minimizes the number of propulsion units, thereby reducing structural mass and maintenance costs. However, it must generate additional anti-torque in hover to counter the main propeller's reaction torque; moreover, flaps and other aerodynamic control surfaces are largely ineffective at such low dynamic pressure, so low-speed control authority is limited, making it difficult to reject gusts and other external disturbances. The dual-propeller configuration employs two counter-rotating propulsion units that cancel reaction torque in hover and place the propellers ahead of the wing so that the propeller slipstream energizes the control surfaces to produce pitch and yaw control moments. Nevertheless, design analyses often do not fully account for how slipstream characteristics vary with rotational speed and their impact on control-surface effectiveness, potentially leading to degraded controllability across a complex flight envelope. In contrast, the quad-propeller configuration, arranged as two counter-rotating pairs, not only inherently cancels reaction torque in hover but also enables direct control of all three body-axis moments via differential thrust, thereby providing unified control during hovering, transition, and level flight, reducing the complexity of attitude-control algorithms, and improving the robustness of the flight-control system.

OPEN ACCESS

***Correspondence**
Yifan Xia,
xiayifan@zju.edu.cn

Received: *21 August 2025*
Revised: 14 September 2025
Accepted: *06 November 2025*
Published: *27 November 2025*

Citation:
Fan M and Xia Y (2025) A Minimum Snap Flight Transition Strategy for Quadrotor Tail-Sitter UAVs: Altitude-Hold Transition. Aerosp. Res. Commun. 3:15466. doi: 10.3389/arc.2025.15466

The transition from hovering to level flight has consistently posed a challenge for VTOL UAVs with wings. Traditionally, an aircraft with wings has to conduct the transition from vertical to level flight by "jumping" [5]. Even the most experienced pilots will inevitably encounter an obvious increase in altitude, as with the famous Pugachev's Cobra Maneuver [11]. As our understanding of aerodynamics deepens and control technology develops, it becomes possible for an autonomous UAV to maintain a certain altitude during the transition from hovering to level flight. Maintaining a constant altitude during the mode transition has several advantages: (1) The less altitude rises, the less energy is required to land; (2) the smaller space required for the transition makes it suitable for working in cluttered environments; (3) for certain tasks (e.g., surveillance), the UAV needs to operate at a constant altitude.

However, the altitude-hold transition is still a challenging subject. Many studies have been conducted to reduce the altitude variation during the transition process through nonlinear optimization [12–14]. The challenges of altitude-hold transitions for tail-sitters arise from the highly nonlinear aerodynamics caused by significant variations in the angle of attack (AoA). The large flight envelope makes the modeling of tail-sitters a difficult and expensive task. Moreover, even when a model is established, finding the necessary inputs to operate the tail-sitter during flight is challenging due to the range of forces that affect the aircraft's acceleration within the flight envelope. For this reason, tail-sitter control usually adopts a cascaded structure [7, 12, 13, 15]—the altitude and speed are controlled separately from the attitude, which divides the transition flight into two phases: transition and level flight. This structure simplifies the design of the outer loop, but limits its bandwidth [9]. In contrast, this study employs a manifold-based model predictive controller for trajectory tracking, wherein the UAV's rotation matrix in 3-D space is mapped from the manifold to a Euclidean space. Compared with prevailing cascaded control architectures, this approach yields higher trajectory-tracking control bandwidth.

The main contributions of this work are as follows.

1. A minimum snap altitude-hold transition method is proposed for quadrotor tail-sitter UAVs. Guaranteeing minimum snap during the transition saves more energy and makes it easier for the vehicle to meet the control input constraints.
2. Compared with the existing works, we adopt an MPC-based global controller to track the full dynamic model. Its predictive nature, which exploits the information about the future reference trajectory, contributes to a higher control bandwidth for trajectory tracking.
3. The numerical simulation validates the effectiveness of the proposed method for the tail-sitter UAVs.

The rest of this article is organized as follows. The flight dynamics of the tail-sitter are presented in Section *Flight Dynamics*. Section *Trajectory Generation and Tracking of Altitude-Hold Transition* provides a thorough explanation of our strategy for addressing the altitude-hold transition problem for quadrotor tail-sitter UAVs. Section *Simulation Results* presents the numerical simulations and discusses the results. Finally, Section *Conclusion* concludes this article.

FLIGHT DYNAMICS

This section presents the flight dynamics model of the tail-sitter, which serves as the foundation of our trajectory generation method. In Section *Vehicle Equations of Motion*, the translational and rotational dynamics of the tail-sitter are introduced. The aerodynamic model is presented in Section *Modeling of Aerodynamics*.

Vehicle Equations of Motion

In the present work, the tail-sitter UAV (SWAN K1 PRO) is utilized to validate our proposed altitude-hold transition method, as shown in **Figure 1**. The SWAN K1 PRO is developed by HEQ UAV Technical Company in Shenzhen, China. The vehicle's mass and wingspan are 1.3328 kg and 1.085 m, respectively. It lacks control surfaces, relying solely on the four propellers on the fuselage to generate all torques and forces. The inertial frame $\{\mathbf{Oxyz}\}$ is defined as North-East-Down (NED). The body frame $\{\mathbf{O}_b\mathbf{x}_b\mathbf{y}_b\mathbf{z}_b\}$ is defined as Front-Right-Down (FRD). $\mathbf{O}$ is the origin of the world coordinate frame, while $\mathbf{O}_b$ is the center of gravity of the vehicle.

The states of the tail-sitter UAV are denoted by $\mathbf{x}_{full} = \{\mathbf{p}, \mathbf{v}, \mathbf{R}, \boldsymbol{\omega}\}$. $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ are the vehicle position and velocity in the world-fixed reference frame, respectively; $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity in the body frame. $\mathbf{R} \in SO(3)$ denotes the rotation matrix from the body frame to the inertial frame; The inputs for the tail-sitter in terms of force and torques are $\mathbf{u}_{full} = \{f, \boldsymbol{\tau}\}$, where f and $\boldsymbol{\tau} \in \mathbb{R}^3$ denote the thrust and control moment vector produced by four propellers, respectively.

Within these definitions, the translational and rotational dynamics of the quadrotor tail-sitter UAV can be described by the following equations:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1}$$

$$\dot{\mathbf{v}} = \mathbf{g} + \frac{1}{m}\left(f\mathbf{R}\mathbf{e}_1 + \mathbf{R}\mathbf{f}_a\right) \tag{2}$$

$$\dot{\mathbf{R}} = \mathbf{R}\lfloor\boldsymbol{\omega}\rfloor \tag{3}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} + \mathbf{M}_a - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \tag{4}$$

Where $\mathbf{g} = (0, 0, 9.8)^{\mathrm{T}}$ is the gravitational acceleration; and m is the vehicle mass. $\mathbf{e}_1 = (1, 0, 0)^{\mathrm{T}}$, $\mathbf{e}_2 = (0, 1, 0)^{\mathrm{T}}$ and $\mathbf{e}_3 = (0, 0, 1)^{\mathrm{T}}$ are the unit vectors. $\mathbf{f}_a \in \mathbb{R}^3$ and $\mathbf{M}_a \in \mathbb{R}^3$ are the aerodynamic force and moment in the body frame, respectively, and will be introduced in Section *Modeling of Aerodynamics*. $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the inertia tensor matrix of the vehicle about the body frame with the center of gravity as the origin. The notation $\lfloor\cdot\rfloor$ converts a 3D vector into an antisymmetric matrix. In this study, the direction of thrust is assumed to align with the $\mathbf{x}_b$. In a situation where propellers are installed at a fixed angle, it only requires a simple transformation using a constant matrix.

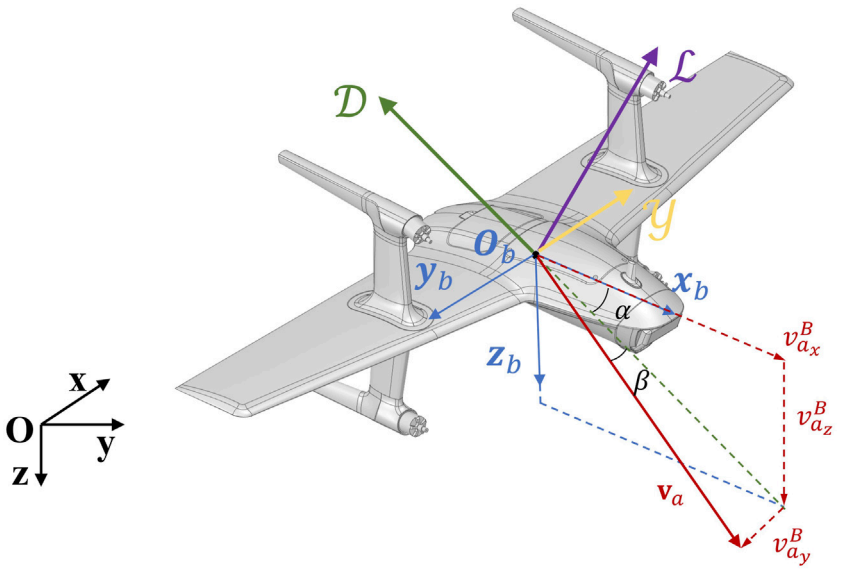

FIGURE 1 | The tail-sitter UAV SWAN K1 PRO and coordinate frames.

Modeling of Aerodynamics

The present study uses an analytical aerodynamic model to determine the aerodynamic forces and moments of an aircraft. Generally, these forces and moments can be described as a set of functions depending on the AoA α and the sideslip angle β.

The aerodynamic force $\mathbf{f}_a$ is modeled in the body frame as follows:

$$\mathbf{f}_a = \begin{bmatrix} -\cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & -\cos\alpha \end{bmatrix} \begin{bmatrix} \mathcal{D} \\ \mathcal{Y} \\ \mathcal{L} \end{bmatrix} \tag{5}$$

where $\mathcal{L}$, $\mathcal{D}$, and $\mathcal{Y}$ are the lift, drag, and side force produced by the fuselage and wings, respectively. L, M, and N are the rolling, pitching, and yawing moments, respectively, along the body axes $\mathbf{x}_b$, $\mathbf{y}_b$, and $\mathbf{z}_b$. Hence, the aerodynamic moment vector $\mathbf{M}_a$ can be modeled as:

$$\mathbf{M}_a = (L, M, N)^{\mathrm{T}} \tag{6}$$

According to Etkin and Reid [16], the aerodynamic forces and moments can be parameterized as:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}\rho V^2 S C_L, \quad \mathcal{D} = \frac{1}{2}\rho V^2 S C_D, \quad \mathcal{Y} = \frac{1}{2}\rho V^2 S C_Y \\ L &= \frac{1}{2}\rho V^2 S b C_l, \quad M = \frac{1}{2}\rho V^2 S \bar{c} C_m, \quad N = \frac{1}{2}\rho V^2 S b C_n \end{aligned} \tag{7}$$

where $V = \|\mathbf{v}_a\|$ is the magnitude of the air speed, ρ is the density of the air, S is the reference area of the wing, $\bar{c}$ is the mean aerodynamic chord, and b is the span of airplane. It should be noted that $\mathbf{v}_a = \mathbf{v} - \mathbf{w}$, where $\mathbf{w}$ is the wind velocity. C_L, C_D, and C_Y are the lift coefficients, drag coefficients, and side force coefficients of the vehicle, respectively, while C_l, C_m, and C_n are the rolling, pitching, and yawing moment coefficients, respectively. Each set of non-dimensional numbers is exclusively linked to the aircraft's shape and attitude.

The total aerodynamic force $\mathbf{f}_a$ in **Equation 5** can be rewritten as:

$$\mathbf{f}_a = \frac{1}{2}\rho V^2 S \mathbf{C}(\alpha, \beta) \tag{8}$$

where

$$\mathbf{C}(\alpha,\beta) = \begin{bmatrix} \mathbf{C}_x(\alpha,\beta) & \mathbf{C}_y(\alpha,\beta) & \mathbf{C}_z(\alpha,\beta) \end{bmatrix}^{\mathrm{T}} \tag{9}$$

$$\mathbf{C}_x(\alpha,\beta) = -C_D(\alpha,\beta)\cos\alpha + C_L(\alpha,\beta)\sin\alpha \tag{10}$$

$$\mathbf{C}_y(\alpha,\beta) = C_Y(\alpha,\beta) \tag{11}$$

$$\mathbf{C}_z(\alpha,\beta) = -C_D(\alpha,\beta)\sin\alpha - C_L(\alpha,\beta)\cos\alpha \tag{12}$$

Based on the air velocity with respect to the body frame $\mathbf{v}_a^B = \mathbf{R}^T\mathbf{v}_a = (\mathbf{v}_{a_x}^B, \mathbf{v}_{a_y}^B, \mathbf{v}_{a_z}^B)^{\mathrm{T}}$, the AoA α and the sideslip angle β can be calculated as:

$$\alpha = \arctan\left(\frac{\mathbf{v}_{a_z}^B}{\mathbf{v}_{a_x}^B}\right) \tag{13}$$

$$\beta = \arcsin\left(\frac{\mathbf{v}_{a_y}^B}{V}\right) \tag{14}$$

Since the airframe is symmetric with respect to the $\mathbf{x}_b\mathbf{O}_b\mathbf{z}_b$ plane, it can be concluded that:

$$C_L(\alpha,\beta) = C_L(\alpha,-\beta), \forall\alpha,\beta \tag{15}$$

$$C_D(\alpha,\beta) = C_D(\alpha,-\beta), \forall\alpha,\beta \tag{16}$$

$$C_Y(\alpha,\beta) = -C_Y(\alpha,-\beta), \forall\alpha,\beta \tag{17}$$

Computational fluid dynamics (CFD) are used to obtain the aerodynamic parameters. Details of the CFD method and simulation results can be found in our previous work [17].

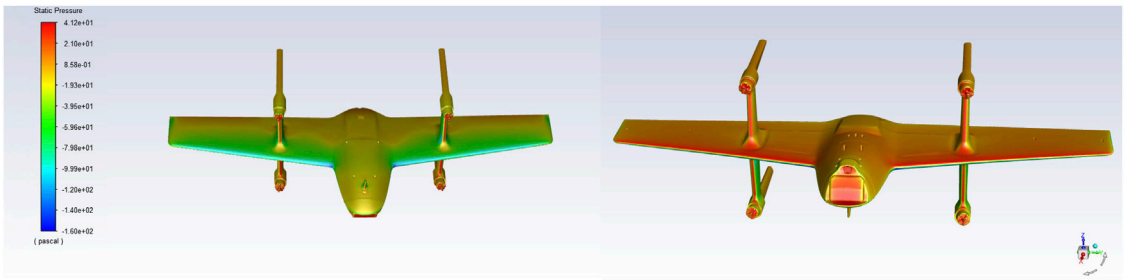

FIGURE 2 | The surface pressure distribution of the tail-sitter UAV calculated by CFD. Left is the top; right is the bottom. (AoA = 10° and airspeed = 8 m/s).

Figure 2 shows an example of the surface pressure on the tail-sitter.

TRAJECTORY GENERATION AND TRACKING OF ALTITUDE-HOLD TRANSITION

In this section, we introduce a strategy that addresses the altitude-hold transition problem for quadrotor tail-sitter UAVs. Section *Minimum Snap Trajectory Generation* provides a method to handle the altitude-hold transition problem as a trajectory generation problem. To track the trajectory, Section *Global Controller for the Tail-Sitter* presents a controller, which is a global controller for the tail-sitter based on MPC.

Minimum Snap Trajectory Generation

The essence of the altitude-hold transition can be regarded as an optimal boundary value problem (OBVP) that considers only the initial and final states, which can be expressed as follows:

$$\min \int_0^T \mathbf{p}^{(s)}(t)^{\mathrm{T}} \mathbf{p}^{(s)}(t) dt \tag{18}$$

$$\mathbf{s.t.}\ t \in [0, T] \tag{19}$$

$$\mathbf{p}^{[s-1]}(0) = \mathbf{s}_0,\ \mathbf{p}^{[s-1]}(T) = \mathbf{s}_f \tag{20}$$

$$\dot{\mathbf{p}}(t) < V_{\max},\quad \ddot{\mathbf{p}}(t) < a_{\max} \tag{21}$$

Where T is used to denote the temporal duration of the trajectory, and $\mathbf{p}^{(s)}(t)$ means that we minimize the s-th derivative of position $\mathbf{p}$. $V_{\max}$ and $a_{\max}$ are the maximum speed constraint and maximum acceleration constraint, respectively. $\mathbf{s}_0$, $\mathbf{s}_f \in \mathbb{R}^{s\times 3}$ are the constant boundary conditions. $\mathbf{p}^{[s-1]} \in \mathbb{R}^{s\times 3}$ is defined by:

$$\mathbf{p}^{[s-1]} = \left(\mathbf{p}, \dot{\mathbf{p}}, \ldots, \mathbf{p}^{(s-1)}\right)^{\mathrm{T}} \tag{22}$$

In practice, minimizing snap ($s = 4$) optimization roughly corresponds to reducing the required control moment [18]. This increases the likelihood that the control input limits are satisfied and that the trajectory is feasible. Thus, the minimum snap trajectory is utilized in this work. Traditionally, an unconstrained OBVP for trajectory generation can be solved using Pontryagin's Maximum Principle (PMP). The MINCO (minimum control) trajectory class [19] utilizes optimality conditions to directly calculate an OBVP. If the transfer time between the two points is prescribed *a priori*, the minimum control effort trajectory can be derived in closed form from the structure of the problem, without solving an optimal control problem. Specifically, for a system controlled by the s-th derivative of position, the trajectory is a polynomial of degree $2s-1$ with respect to time, where the expression for a 3-D trajectory is denoted by:

$$\mathbf{p}(t) = \mathbf{c}^{\mathrm{T}} \rho(t), t \in [0, T] \tag{23}$$

where $\mathbf{c} \in \mathbb{R}^{2s\times 3}$ are the coefficients of $(2s-1)$th-degree polynomials, $\rho(x) = (1, x, x^2, \ldots, x^{(2s-1)})^{\mathrm{T}}$. To acquire the matrix $\mathbf{c}$, we establish a linear system:

$$\mathbf{A}\mathbf{c} = \mathbf{b} \tag{24}$$

where $\mathbf{A} \in \mathbb{R}^{2s\times 2s}$ and $\mathbf{b} \in \mathbb{R}^{2s\times 3}$ are:

$$\mathbf{A} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_M \end{pmatrix} \tag{25}$$

$$\mathbf{b} = \left(\mathbf{s}_0{}^{\mathrm{T}}, \mathbf{s}_f{}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{26}$$

$\mathbf{H}_0$, $\mathbf{G}_M \in \mathbb{R}^{s\times 2s}$ are defined as:

$$\mathbf{H}_0 = \left(\rho(0), \dot{\rho}(0), \ldots, \rho^{(s-1)}(0)\right)^{\mathrm{T}} \tag{27}$$

$$\mathbf{G}_M = \left(\rho(T), \dot{\rho}(T), \ldots, \rho^{(s-1)}(T)\right)^{\mathrm{T}} \tag{28}$$

The matrix $\mathbf{A}$ is nonsingular and banded for any $T > 0$. Using the MINCO trajectory class **Equations 18–21**, can be recast as an unconstrained optimization problem with respect to the decision variable T. Substituting the optimal $T*$ into **Equation 24** and solving then yields a minimum-snap trajectory.

Leveraging the differential flatness property, several recent studies [20, 21] for multirotor UAVs have explicitly incorporated attitude, angular-velocity, and control-input constraints already at the trajectory generation phase. The underlying reason is that, for multirotors, both the system states and the control inputs can be expressed as closed-form functions of position and a finite number of its time derivatives, thereby enabling the trajectory optimization to be cast as a second-order cone program (SOCP) and solved efficiently. In contrast, in the differential flatness mapping of a tail-sitter UAV, the attitude must be obtained by numerically solving nonlinear algebraic equations, making it difficult to derive a closed-form expression in terms of the

decision variable T. Consequently, attitude cannot be included directly in the optimization as a function of T, nor can angular-velocity and control-input constraints be explicitly handled as convex constraints at the planning stage, which hinders the formulation of the problem as an SOCP. Nevertheless, this does not imply a deficiency in the feasibility of the proposed trajectory-generation method: we implicitly bound the states by imposing a maximum-speed constraint and a maximum-acceleration constraint, and we implicitly constrain variations in the control inputs by minimizing the second time derivative of acceleration (snap). A substantial body of prior work [9, 22, 23] has shown that enforcing constraints on velocity and its finite-order derivatives (equivalently, on higher-order derivatives of position) at the planning stage suffices, to a large extent, to ensure dynamic feasibility. In addition, during the trajectory tracking phase, control-input constraints will be explicitly enforced (see Section *Global Controller for the Tail-Sitter*).

Global Controller for the Tail-Sitter

The system **Equations 1–4** is a cascaded structure; that is, the input torque, $\boldsymbol{\tau}$, only affects the angular velocity. The attitude of the UAV is then calculated through the angular velocity, and the current speed and position can be determined through the attitude of the UAV. Due to this cascaded dynamics, the angular velocity dynamics **Equation 4** can be tracked separately. In this work, the angular velocity of the tail-sitter is tracked by three decoupled proportional–integral–derivative (PID) controllers in the autopilot. The Coriolis term, $\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}$, and the aerodynamic moment, $\mathbf{M}_a$, are both regarded as unknown disturbances and are compensated for in a feed-forward way. With a well-designed PID controller, the vehicle's angular velocity can be achieved instantaneously. Thus, in terms of controller design, the tail-sitter's states and inputs can be simplified as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^{\mathrm{T}} & \mathbf{v}^{\mathrm{T}} & \mathbf{R}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{29}$$

$$\mathbf{u} = \begin{bmatrix} f & \boldsymbol{\omega}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{30}$$

Based on **Equations 29**, **30**, we utilize the same controller as in our previous work [17], which is derived from Model Predictive Control (MPC). MPC is a common and powerful trajectory tracking method in robotics. Generally, we can design MPC to track the trajectory of a model whose states are all in Euclidean space (such as the unicycle model). This is because all system states are flat vectors, and by calculating their error's norm, we can quantify the tracking error. However, for the dynamic model shown in **Equations 1–4**, the tail-sitter's states contain the rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$, which is not in Euclidean space but on a non-vector, curved manifold. This makes it difficult to quantify the rotation matrix error $\delta\mathbf{R}$. Inspired by [24], the rotation matrix $\mathbf{R}$ can be projected into Euclidean space via mapping:

$$\boldsymbol{\theta} = Log(\mathbf{R}) \in \mathbb{R}^3 \tag{31}$$

where $Log(\cdot)$ is the logarithmic map proposed by [25], which is defined as:

$$Log_{SO(3)}(\mathbf{R}) = \frac{\phi}{2\sin\phi}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \tag{32}$$

where ϕ satisfies $\cos(\phi) = \frac{1}{2}(tr(\mathbf{R}) - 1)$.

Furthermore, $\delta\mathbf{R}$ can be formulated as:

$$\delta\mathbf{R} = \mathbf{R}_r \boxminus \mathbf{R} = Log\left(\mathbf{R}^{\mathrm{T}}\mathbf{R}_r\right) \tag{33}$$

where $\mathbf{R}_r$ is the rotation matrix on the reference trajectory, and $\mathbf{R}$ is the actual rotation matrix. The error of the rotation matrix is represented as a 3D vector, and calculating its norm can track the attitude of the vehicle.

Next, we construct the MPC objective function to ensure optimal performance of our control strategy. Benefiting from the differential flatness of quadrotor tail-sitter UAVs, the UAV's reference 12-state can be determined from a time differentiable curve at any given moment. Therefore, according to **Equations 29**, **30**, **33**, the error of the states and inputs can be represented as:

$$\begin{cases} \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p}^{\mathrm{T}} & \delta\mathbf{v}^{\mathrm{T}} & \delta\mathbf{R}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^9 \\ \delta\mathbf{u} = \begin{bmatrix} \delta f & \delta\boldsymbol{\omega}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^4 \end{cases} \tag{34}$$

$$\delta\mathbf{p} = \mathbf{p}_r - \mathbf{p} \in \mathbb{R}^3 \tag{35}$$

$$\delta\mathbf{v} = \mathbf{v}_r - \mathbf{v} \in \mathbb{R}^3 \tag{36}$$

$$\delta\mathbf{R} = \mathbf{R}_r \boxminus \mathbf{R} = Log\left(\mathbf{R}^{\mathrm{T}}\mathbf{R}_r\right) \in \mathbb{R}^3 \tag{37}$$

$$\delta f = f_r - f \in \mathbb{R} \tag{38}$$

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega}_r - \boldsymbol{\omega} \in \mathbb{R}^3 \tag{39}$$

Where the subscript r represents the value on the reference trajectory, and no subscript represents the actual value. Therefore, we can construct an MPC as an optimization problem:

$$\begin{aligned} \delta\mathbf{u}_k^{\star} = \arg\min_{\delta\mathbf{u}_k} & \sum_{k=0}^{N-1}\left(\|\mathbf{x}_{rk+1} - \mathbf{x}_{k+1}\|_{\mathrm{Q}_k}^2 + \|\delta\mathbf{u}_k\|_{\mathrm{P}_k}^2\right) \\ \text{s.t.} \quad & \delta\mathbf{x}_{k+1} = \left(\mathbf{I} + \Delta t\mathbf{F}_{\mathbf{x}_k}\right)\delta\mathbf{x}_k + \Delta t\mathbf{F}_{\mathbf{u}_k}\delta\mathbf{u}_k + \Delta t\mathbf{F}_{\mathbf{w}_k}\delta\mathbf{w}_k \\ & \mathbf{u}_{\min} \le \mathbf{u}_{rk} - \delta\mathbf{u}_k \le \mathbf{u}_{\max} \end{aligned} \tag{40}$$

where N is the receding horizon, Q_k and P_k are positive diagonal matrices, $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ denote the actual input constraints, $\mathbf{F}_{\mathbf{x}}$, $\mathbf{F}_{\mathbf{u}}$, and $\mathbf{F}_{\mathbf{w}}$ are the Jacobians of the error-state dynamic model $\dot{\delta\mathbf{x}}$ with respect to $\delta\mathbf{x}$, $\delta\mathbf{u}$ and $\delta\mathbf{w}$, respectively. The $\delta\mathbf{w} = \mathbf{w}_r - \mathbf{w}$, where $\mathbf{w}$ is the wind velocity in the environment, and $\mathbf{w}_r$ is the wind velocity used in trajectory optimization. The objective function should be as small as possible so that the states of the trajectory are well tracked and the trajectory inputs are as smooth as possible. Finally, the optimal control of a trajectory is:

$$\mathbf{u}^{\star} = \mathbf{u}_r - \delta\mathbf{u}^{\star} \tag{41}$$

To obtain $\mathbf{F}_{\mathbf{x}}$, $\mathbf{F}_{\mathbf{u}}$, and $\mathbf{F}_{\mathbf{w}}$, we take the derivative of the state error with respect to time to obtain the error dynamic model:

$$\dot{\delta\mathbf{p}} = \delta\mathbf{v} \tag{42}$$

$$\dot{\delta\mathbf{v}} = \frac{1}{m}\left(f_r\mathbf{R}_r\mathbf{e}_1 + \mathbf{R}_r\mathbf{f}_{ar} - \left(f\mathbf{R}\mathbf{e}_1 + \mathbf{R}\mathbf{f}_a\right)\right) \tag{43}$$

$$\dot{\delta\mathbf{R}} = \mathbf{A}^{-T}(\delta\mathbf{R})\left(-\mathbf{R}_r{}^{T}\mathbf{R}\boldsymbol{\omega} + \boldsymbol{\omega}_r\right) \tag{44}$$

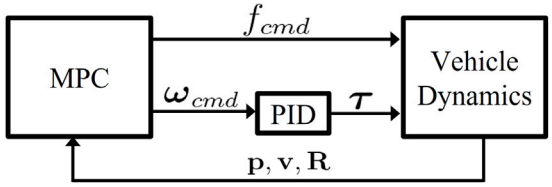

FIGURE 3 | The global controller structure.

Where $\mathbf{f}_{ar}$ and $\mathbf{f}_a$ are the aerodynamic forces on the reference trajectory and actual trajectory, respectively. $\mathbf{A}(\cdot)$: $\mathbb{R}^3 \rightarrow \mathbb{R}^{3\times3}$ represents a map [25]:

$$\mathbf{A}(\delta\mathbf{R}) = \mathbf{I} + \left(\frac{1-\cos\|\delta\mathbf{R}\|}{\|\delta\mathbf{R}\|}\right)\frac{\lfloor\delta\mathbf{R}\rfloor}{\|\delta\mathbf{R}\|} + \left(1-\frac{\sin\|\delta\mathbf{R}\|}{\|\delta\mathbf{R}\|}\right)\frac{\lfloor\delta\mathbf{R}\rfloor^2}{\|\delta\mathbf{R}\|^2} \tag{45}$$

See **Supplementary Appendix SA** for the proof of **Equation 44**.

Linearizing **Equations 42–44**:

$$\dot{\delta\mathbf{x}} = \mathbf{F}_\mathbf{x}\delta\mathbf{x} + \mathbf{F}_\mathbf{u}\delta\mathbf{u} + \mathbf{F}_\mathbf{w}\delta\mathbf{w} \tag{46}$$

where

$$\mathbf{F}_\mathbf{x} \approx \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{m}\mathbf{R}_r\frac{\partial\mathbf{f}_{ar}}{\partial\mathbf{v}_a^B r}\mathbf{R}_r{}^{\mathrm{T}} & \frac{\partial\delta\dot{\mathbf{v}}}{\partial\delta\mathbf{R}} \\ \mathbf{0} & \mathbf{0} & -\lfloor\boldsymbol{\omega}\rfloor - \frac{1}{2}\lfloor\delta\boldsymbol{\omega}\rfloor + \frac{1}{2}\mathbf{K} \end{bmatrix} \tag{47}$$

$$\mathbf{F}_\mathbf{u} \approx \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{R}_r\mathbf{e}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} + \frac{1}{2}\lfloor\delta\mathbf{R}\rfloor \end{bmatrix} \tag{48}$$

$$\mathbf{F}_\mathbf{w} \approx \begin{bmatrix} \mathbf{0} \\ -\frac{1}{m}\mathbf{R}_r\frac{\partial\mathbf{f}_{ar}}{\partial\mathbf{v}_a^B r}\mathbf{R}_r{}^{\mathrm{T}} \\ \mathbf{0} \end{bmatrix} \tag{49}$$

$$\frac{\partial\delta\dot{\mathbf{v}}}{\partial\delta\mathbf{R}} \approx \mathbf{R}_r\left(-a_T\lfloor\mathbf{e}_1\rfloor - \frac{1}{m}\lfloor\mathbf{f}_a\rfloor + \frac{1}{m}\frac{\partial\mathbf{f}_{ar}}{\partial\mathbf{v}_a^B r}\lfloor\mathbf{R}_r{}^{\mathrm{T}}\mathbf{v}_a\rfloor\right) \tag{50}$$

Where the matrix $\mathbf{K}$ and the proof of Section *Global Controller for the Tail-Sitter* are given in **Supplementary Appendix SB**.

In conclusion **Equation 40**, can be seen as an optimization problem with an inequality constraint that can be solved by the Powell-Hestenes-Rockafellar augmented Lagrangian method (PHR-ALM) [26]. The controller structure is shown in **Figure 3**.

SIMULATION RESULTS

This section demonstrates our methods' ability to generate and track an altitude-hold trajectory. The Ryzen R5-5600G CPU (3.9 GHz) and 8 GB of RAM are utilized to run the simulation.

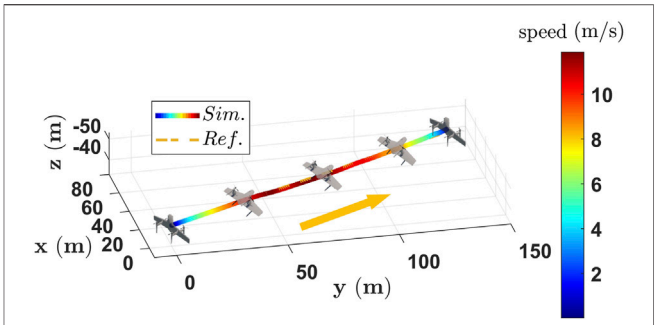

FIGURE 4 | An altitude-hold transition trajectory.

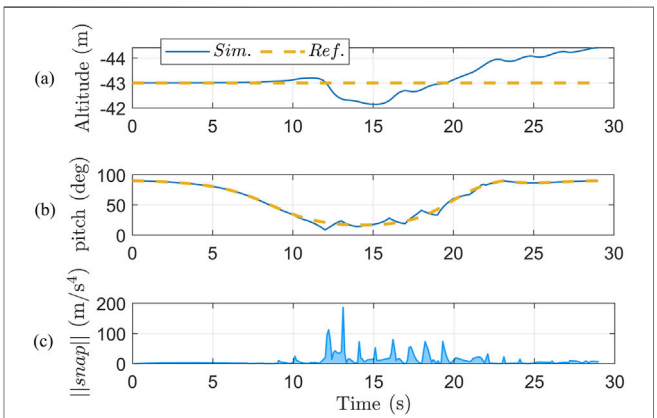

FIGURE 5 | States of the altitude-hold transition simulation. **(a)** altitude, **(b)** pitch, **(c)** integration of snap.

In terms of the MPC controller, horizon N is set to 10 and step size Δt is set to 0.1. In practice, the MPC takes 0.15 s on average to compute the optimal commands. The angular velocity command $\boldsymbol{\omega}$ is tracked by three PID controllers, with each comparing the respective angular velocity command with its actual values and calculating a normalized control torque $\boldsymbol{\tau}$ at 400 Hz.

Figure 4 shows the 3-D trajectory with a maximum speed of 12 m/s, and the corresponding flight states are detailed in **Figure 5**. First, the vehicle transitions from hovering to level flight and accelerates to a maximum speed of 12 m/s. Then, the vehicle performs a backward transition by decelerating from level flight to a hovering state. As can be seen in **Figure 5**, the altitude error remained within 1.4 m during the entire flight, with the entire path being approximately 150 m. The maximum altitude error occurred at the end of the trajectory, as the vehicle transitioned from level flight to a hovering state. The pitch angle of the UAV increased, and the direction of the thrust f_{cmd} became gradually perpendicular to the UAV's velocity, resulting in a gradual climb of the UAV. Overall, the tracking performance is satisfactory for both altitude and pitch angles.

Figure 6 shows the simulated control inputs for the altitude-hold transition. **Figure 6a** shows the angular velocity time-domain response when tracking the command. It is easy to

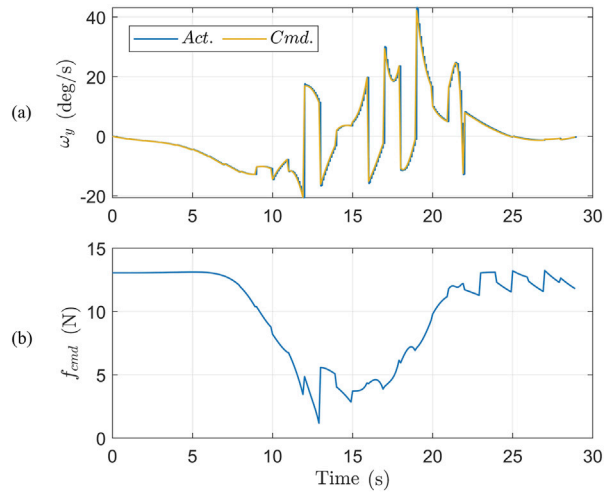

FIGURE 6 | Control inputs of the altitude-hold transition. **(a)** angular velocity calculated from MPC and actual angular velocity, **(b)** thrust calculated from MPC.

observe that the PID controller exhibits no overshoot and has fast tracking. **Figure 6b** shows the thrust command to the autopilot in the real world.

CONCLUSION

In this study, an altitude-hold flight mode transition strategy for quadrotor tail-sitter UAVs is proposed. For the proposed strategy, the altitude-hold transition is transformed into a trajectory generation and tracking problem. An optimal condition is utilized to generate a minimum snap and altitude-hold trajectory. Then, we take advantage of the differential flatness property of the tail-sitter UAVs to mitigate the complex nonlinear dynamics of the wings and fuselage. Moreover, a global controller based on MPC is adopted to track the trajectory. Simulation results indicate that the altitude error remained within 1.4 m during the entire flight, with the entire path being approximately 150 m. The AoA variation is approximately 80°, and the overall strategy performance is acceptable. The proposed strategy is universally suitable for all tail-sitter UAVs that utilize a quadrotor to control their attitude.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

MF: Writing – original draft, Conceptualization, Investigation, Methodology, Software, Validation, Visualization; YX: Writing – review and editing, Investigation, Validation, Supervision. All authors contributed to the article and approved the submitted version.

FUNDING

The authors declare that no financial support was received for the research and/or publication of this article.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

GENERATIVE AI STATEMENT

The authors declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontierspartnerships.org/articles/10.3389/arc.2025.15466/full#supplementary-material

REFERENCES

1. Huang J, Du B, Zhang Y, Quan Q, Wang B, Mu L. A Pesticide Spraying Mission Allocation and Path Planning with Multicopters. *IEEE Trans Aerospace Electron Syst* (2024) 60:2277–91. doi:10.1109/taes.2024.3355028
2. Duan D, Zu R, Yu T, Zhang C, Li J. Differential Flatness-based real-time Trajectory Planning for Multihelicopter Cooperative Transportation in Crowded Environments. *AIAA J* (2023) 61:4079–95. doi:10.2514/1.j062854
3. Prajapati P, Vashista V. Aerial Physical Human Robot Interaction for Payload Transportation. *IEEE Robotics Automation Lett* (2023) 8:4903–10. doi:10.1109/lra.2023.3290410
4. Vasconcelos JVR, Hudson TM, dos Santos AG, Saska M, Brandão AS. Uav-Based Optimization for Fruit Counting in Greenhouses. In: *2024 International Conference on Unmanned Aircraft Systems (ICUAS) (IEEE)* (2024). p. 436–41.
5. Cheng ZH, Pei HL. Transition Analysis and Practical Flight Control for Ducted Fan fixed-wing Aerial Robot: Level Path Flight Mode Transition. *IEEE Robotics Automation Lett* (2022) 7:3106–13. doi:10.1109/LRA.2022.3145087
6. De Wagter C, Ruijsink R, Smeur EJ, van Hecke KG, van Tienen F, van der Horst E, et al. Design, Control, and Visual Navigation of the Delftacopter Vtol tail-sitter Uav. *J Field Robotics* (2018) 35:937–60. doi:10.1002/rob.21789
7. Tal E, Karaman S. Global Incremental Flight Control for Agile Maneuvering of a Tailsitter Flying Wing. *J Guidance, Control Dyn* (2022) 45:2332–49. doi:10.2514/1.g006645

8. Zhong J, Chen W, Zhang H. Transition Control of a tail-sitter Unmanned Aerial Vehicle with l1 Neural Network Adaptive Control. *Chin J Aeronautics* (2023) 36:460–75.
9. Lu G, Cai Y, Chen N, Kong F, Ren Y, Zhang F. Trajectory Generation and Tracking Control for Aggressive tail-sitter Flights. *The Int J Robotics Res* (2024) 43:241–80. doi:10.1177/02783649231207655
10. Li B, Sun J, Zhou W, Wen CY, Low KH, Chen CK. Transition Optimization for a Vtol tail-sitter Uav. *IEEE/ASME Transactions Mechatronics* (2020) 25: 2534–45. doi:10.1109/tmech.2020.2983255
11. Xu W, Zhang F. Learning Pugachev's Cobra Maneuver for tail-sitter Uavs Using Acceleration Model. *IEEE Robotics Automation Lett* (2020) 5:3452–9. doi:10.1109/LRA.2020.2976323
12. Yang Y, Zhu J, Yuan X, Wang X, Kuang M, Shi H. Dynamic Characteristics Analysis and Robust Transition Control of tail-sitter Vtol Uavs. *Aerospace Sci Technology* (2024) 145:108868. doi:10.1016/j.ast.2024.108868
13. Cheng Z, Pei H. A Corridor-based Flight Mode Transition Strategy for Agile ducted-fan tail-sitter Uav: Altitude-Hold Transition. *Chin J Aeronautics* (2023) 36:330–45. doi:10.1016/j.cja.2023.05.015
14. McIntosh KF, Mishra S, Reddinger JP. Aerodynamic feedforward-feedback Architecture for Tailsitter Control in Hybrid Flight Regimes. *J Guidance, Control Dyn* (2024) 47:2073–84. doi:10.2514/1.g008002
15. Yang Y, Zhang S, Deng J. Dynamics Modeling and Flight Control of a dual-rotor tail-sitter Uav. In: *2023 35th Chinese Control and Decision Conference (CCDC)*. IEEE (2023). p. 2751–6.
16. Etkin B, Reid LD. *Dynamics of Flight: Stability and Control*. John Wiley & Sons (1995).
17. Fan M, Xie F, Ji T, Zheng Y. Three-Dimensional Trajectory Optimization for Quadrotor tail-sitter Aavs: Traversing Through Given Waypoints. *IEEE Trans Aerospace Electron Syst* (2025) 61:6987–7005. doi:10.1109/TAES.2025.3531845
18. Tal E, Ryou G, Karaman S. Aerobatic Trajectory Generation for a Vtol fixed-wing Aircraft Using Differential Flatness. *IEEE Trans Robotics* (2023) 39: 4805–19. doi:10.1109/TRO.2023.3301312
19. Wang Z, Zhou X, Xu C, Gao F. Geometrically Constrained Trajectory Optimization for Multicopters. *IEEE Trans Robotics* (2022) 38:3259–78. doi:10.1109/tro.2022.3160022
20. Akbari B, Greeff M. A Computationally Efficient Learning-based Model Predictive Control for Multirotors Under Aerodynamic Disturbances. In: *2024 International Conference on Unmanned Aircraft Systems (ICUAS)* (2024). p. 185–92. doi:10.1109/ICUAS60882.2024.10557089
21. Freire V, Xu X. Flatness-Based Quadcopter Trajectory Planning and Tracking with continuous-time Safety Guarantees. *IEEE Trans Control Syst Technology* (2023) 31:2319–34. doi:10.1109/tcst.2023.3250954
22. Lu G, Ren Y, Zhu F, Li H, Xue R, Cai Y, et al. Autonomous tail-sitter Flights in Unknown Environments. *IEEE Trans Robotics* (2025) 41:1098–117. doi:10.1109/TRO.2025.3526102
23. Ren Y, Zhu F, Lu G, Cai Y, Yin L, Kong F, et al. Safety-Assured high-speed Navigation for Mavs. *Sci Robotics* (2025) 10:eado6187. doi:10.1126/scirobotics.ado6187
24. Lu G, Xu W, Zhang F. On-manifold Model Predictive Control for Trajectory Tracking on Robotic Systems. *IEEE Trans Ind Electronics* (2022) 70:9192–202. doi:10.1109/tie.2022.3212397
25. Murray RM. Proportional Derivative (Pd) Control on the Euclidean Group. *Eur Control Conf* (1995) 2:1091.
26. Rockafellar RT. A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization. *Math Programming* (1973) 5: 354–73. doi:10.1007/bf01580138