Trajectory Generation and Tracking Control for Aggressive Tail-Sitter Flights

Journal Title
XX(X):1–36
©The Author(s) 2022
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Guozheng Lu, Yixi Cai, Nan Chen, Fanze Kong, Yunfan Ren and Fu Zhang

Abstract

We address the theoretical and practical problems related to the trajectory generation and tracking control of tail-sitter UAVs. Theoretically, we focus on the differential flatness property with full exploitation of actual UAV aerodynamic models, which lays a foundation for generating dynamically feasible trajectory and achieving high-performance tracking control. We have found that a tail-sitter is differentially flat with accurate (not simplified) aerodynamic models within the entire flight envelope, by specifying coordinate flight condition and choosing the vehicle position as the flat output. This fundamental property allows us to fully exploit the high-fidelity aerodynamic models in the trajectory planning and tracking control to achieve accurate tail-sitter flights. Particularly, an optimization-based trajectory planner for tail-sitters is proposed to design high-quality, smooth trajectories with consideration of kinodynamic constraints, singularity-free constraints and actuator saturation. The planned trajectory of flat output is transformed into state trajectory in real-time with optional consideration of wind in environments. To track the state trajectory, a global, singularity-free, and minimally-parameterized on-manifold MPC is developed, which fully leverages the accurate aerodynamic model to achieve high-accuracy trajectory tracking within the whole flight envelope. The proposed algorithms are implemented on our quadrotor tail-sitter prototype, "Hong Hu", and their effectiveness are demonstrated through extensive real-world experiments in both indoor and outdoor field tests, including agile SE(3) flight through consecutive narrow windows requiring specific attitude and with speed up to $10\,\mathrm{m/s}$, typical tail-sitter maneuvers (transition, level flight and loiter) with speed up to $20\,\mathrm{m/s}$, and extremely aggressive aerobatic maneuvers (Wingover, Loop, Vertical Eight and Cuban Eight) with acceleration up to $2.5\,\mathrm{g}$. The video demonstration is available at https://youtu.be/2x_bLbVuyrk.

Keywords

Differential flatness, trajectory generation, motion control, tail-sitter UAVs

1 Introduction

A tail-sitter unmanned aerial vehicle (UAV) is a type of vertical takeoff and landing (VTOL) flying machine that takes off and lands vertically on its tail while tilts the entire airframe in a near horizontal attitude for forward flight. Its hybrid fixed-wing and rotary-wing design combines advantages of the VTOL capability, aerodynamic efficiency, and hence extends the power endurance and flight range. Compared to other hybrid designs of VTOL UAVs, like tilt-rotors (Carlson 2014; Ozdemir et al. 2014), tilt-wings (Çetinsoy et al. 2011), rotor-wing (McKenna 2007), and dual-systems (Park 2014; Gu et al. 2017), tail-sitters have rotors fixed to the wing and use their thrust in all flight conditions, leading to a mechanically simple, lightweight and efficient airframe configuration, which is particularly important for small-scale, low-cost, portable UAVs. Such UAVs hold immense potentials for a wide range of industrial and civil applications, such as infrastructure inspection, geological surveying, environment mapping, and post-disaster search and rescue. These exciting opportunities have attracted intensive research interests and led to the development of a variety of tail-sitter UAV prototypes, such as the single-propeller configuration (Frank et al. 2007; Wang et al. 2017b; De Wagter et al. 2018), the shoulder-mounted twin-engine configuration (Bapst et al. 2015; Ritz and D'Andrea 2017; Sun et al. 2018), and the quadrotor configuration (Oosedo et al. 2013; Gu et al. 2018).

To accommodate the escalating demand of real-world applications, tail-sitter UAVs must be able to execute highly aggressive maneuvers, including forward transition to level flight, back transition break, and quickly bank turns. The agile maneuverability is crucial for the UAV to navigate at high-speed through obstacle-dense environments. Unlike conventional airplanes that fly in open space, tail-sitter UAVs are subjected to more challenging flight conditions of fast-varying speed and attitude, asking a holistic design of the trajectory generation and tracking control, where the former aims to plan a smooth, dynamically-feasible, and collision-free trajectory and the latter should track the planned trajectory with small errors.

While the planning and control of multicopter UAVs have been comprehensively studied by leveraging the differential flatness property of the systems (Mellinger and Kumar 2011; Faessler et al. 2017), thus stimulating a wealth of practical

All authors are with the Department of Mechanical Engineering, The University of Hong Kong.

Corresponding author:
Fu Zhang, Department of Mechanical Engineering, The University of Hong Kong, HW 7-18, Pokfulam, Hong Kong.
Email: fuzhang@hku.hk

applications, like flying through narrow gaps (Mellinger et al. 2012; Falanga et al. 2017; Ren et al. 2023), perching on structures (Mellinger et al. 2012; Hang et al. 2019), autonomous safe navigation (Shen et al. 2011; Zhou et al. 2019; Zhang et al. 2020), and drone racing (Foehn et al. 2021), the equivalent techniques for tail-sitter UAVs are relatively underdeveloped. The differential flatness for tail-sitters, which resolves the full states and inputs of the system from finite flat outputs and their derivatives, has not been rigorously investigated. A significant hurdle confronting this task lies in the complex nonlinear aerodynamics inherent to tail-sitter UAVs. While the wings of a tail-sitter can produce the desired lift force to enhance power efficiency, they also introduce highly nonlinear aerodynamic forces into the system dynamics. Unlike fixed-wing airplanes that are primarily confined to a conservative level flight regime where the wing aerodynamics are well understood as being linear, tail-sitters usually operate within a large flight envelope with a wide range of angle of attack (AoA), where the wing aerodynamics exhibit extreme nonlinearity. Consequently, the study on differential flatness, as well as high-precision planning and control of tail-sitter UAVs that fully exploit aerodynamic models are significantly complicated and still remains an open question.

Besides the theoretical difficulty, trajectory generation and tracking of tail-sitter UAVs are also confronted with practical challenges. For example, during outdoor long-range missions, a tail-sitter UAV often suffers from model uncertainties and considerable wind disturbances. Other constraints such as actuator saturation, sensor noise and limited onboard computation resource also ask for high robustness and computation efficiency of the designed algorithms.

1.1 Contributions

In this work, we address the challenge of high-quality trajectory generation and high-performance tracking control of tail-sitters by leveraging the differential flatness property, aiming to enable agile tail-sitter flights within the whole envelope in real-world environments. Specifically, our contributions are outlined as follows.

1) We show that the tail-sitter is differentially flat in the coordinated flight condition, in considering the actual aerodyanamic model without any simplifications.
2) Based on the differential flatness, we develop an optimization-based trajectory generation method enabling aggressive flights while taking account of actuator constraints, flight time, dynamical feasibility, and singularity conditions in coordinated flight.
3) We propose a two-stage control architecture. The first stage transforms the planned flat-output trajectory into a state-input trajectory while compensating wind effect and treating singularities. The second stage is a real-time state trajectory tracking controller.
4) For the second stage, we develop a global, model-based, minimally-parameterized and singularity-free model predictive control (MPC) for trajectory tracking within the entire tail-sitter flight envelope.
5) We demonstrate and validate our algorithms via extensive real-world experiments on an actual quadrotor tail-sitter prototype in both indoor and outdoor environments. To our best knowledge, it is the first tail-sitter demonstration of flying through narrow tilted windows and outdoor aerobatics.

1.2 Outline

The outline of the rest of the paper is as follows. Section 2 reviews the related work. The system dynamics including the aerodynamic model are introduced in Section 3. The fundamental property of differential flatness is proved in Section 4. Section 5 describes the system architecture including high-level trajectory generation and tracking, and low-level control. Section 6 presents the optimization-based trajectory generation and its solver. Section 7 derives the error-state dynamics along the reference trajectory leading to an on-manifold MPC. Section 8 presents real-world experiments validating our approach. Finally, Section 9 concludes the paper with extensions and limitations.

2 Related work

2.1 Tail-sitter control

There is a wealth of research on tail-sitter control which can be generally categorized into two main strategies: the separated control strategy, which consists of several isolated controllers designed for respective flight phases, and the global control that regulates the vehicle maneuvers within the entire envelope under a unified controller. We will discuss these control approaches in the following content.

Since the tail-sitter dynamics reduce to a rotary-wing model and a fixed-wing model in low-speed vertical flight and high-speed level flight respectively, separated control methods (Frank et al. 2007; Oosedo et al. 2013; Lyu et al. 2017b) usually divide the flight process into three phases – vertical flight (including takeoff, landing and hovering), transition and level flight – and design controllers separately for each phase. The vertical flight dynamics are linearized at the stationary hovering equilibrium (Frank et al. 2007; Matsumoto et al. 2010; Lyu et al. 2017b), and controlled by means of established control methods for quadrotors, such as loop-shaping, (Zhou et al. 2018), robust control (Lyu et al. 2018b), and MPC (Li et al. 2018). The level flight controllers are usually borrowed from the fixed-wing airplanes and UAVs, such as the total energy control system (Lambregts 1983) which is widely used in the open-source autopilot PX4 (Meier et al. 2015).

Transition control is a key challenge for the separated control strategy and there is rich literature addressing this issue. The aerodynamics become highly nonlinear during the transition due to the dramatic change of AoA. An intuitive linear control method is to feed a pre-designed profile of linearly decreasing or increasing pitch angle to the attitude controller with a constant altitude command (Verling et al. 2016; Lyu et al. 2017a), forcing the vehicle to pitch down or up until triggering the mode-switching condition. Because of the nonlinear dynamics, gain-scheduling techniques (Kita et al. 2010; Jung and Shim 2012) could be applied to enhance the stability margin. However, this linear method is not always dynamically feasible and usually results in undesired altitude deviation. The altitude control performance can be

Tab. 1. Comparison of the stat-of-the-art global control methods for tail-sitter UAVs.

Study	Methodology	Aerodynamic Model	Singularity	Flight Condition	Wind Compensation	Demo Flights
Ours	MPC & Differential Flatness	Classic	Specific Airspeed	Coordinated	On Reference	$*, \dagger, \ddagger$
Tal and Karaman (2022)	Cascaded PD & INDI & Differential Flatness	ϕ-Theory	Specific Airspeed	No Restriction	On Input	$*, \ddagger$
Lustosa (2017)	Scheduled LQR	ϕ-Theory	None	No Restriction	None	$*$
Ritz and D'Andrea (2017)	Cascaded PID	Classic	Specific Airspeed	Coordinated	None	$*$
Smeur et al. (2020)	INDI	Quasi-Static	Euler Angle	Not Specified	On Input	$*$
Cheng and Pei (2022)	Adaptive Control	-	Euler Angle	Not Specified	None	$*$

Symbols $*, \dagger, \ddagger$ indicate three different demonstrated maneuvers: the typical maneuvers $*$ include common tail-sitter flights such as transition, level flight and loiter; the $SE(3)$ maneuvers $\dagger$ denote a whole-body flying motion with specified pose and velocity; the aerobatic maneuvers $\ddagger$ denotes aggressive maneuvers with large attitude variation and flight speeds.

improved either by a well-designed transition planner (Naldi and Marconi 2011; Oosedo et al. 2017; Wang et al. 2017a; Li et al. 2020a) using accurate aerodynamic models or a sophisticated altitude controller, such as iterative learning control (Xu et al. 2019b). A limitation of these methods is their focus on the altitude and pitch control to transit a tail-sitter to the level flight phase, often neglecting the lateral motion or any maneuvers (e.g., bank turns) during the transition, which are necessary for obstacle avoidance in low-altitude cluttered environments.

To sum up, although the separated strategy eases the controller design and has widespread use in practice, the controller switching usually causes unexpected transient response, thereby degrading control performance. Given that a tail-sitter would frequently enter and exit the transition regime (i.e., a specified range of pitch angle and airspeed) when performing aggressive maneuvers, a global control strategy that uses a unified system model and control law serving for the whole envelope without switching among different flight phases (e.g., hovering, transition, and level flight) is more preferable. This direction has prompted a significant amount of research.

Model-free global control methods for tail-sitters do not rely on vehicle aerodynamic models, but manage to approximate the aircraft dynamics locally and stabilize the local approximation by using linear theory. For example, Barth et al. (2020) proposed a cascaded model-free global control framework based on quasi-static assumptions. The vehicle system is decoupled, approximated and estimated locally as a group of second-order piece-wise linear systems, and thus the reference thrust and attitude can be solved from the desired body velocity. Similarly, Cheng and Pei (2022) employed an adaptive control law with an IMU-based thrust-attitude decoupling method, assuming zero gradient for the aerodynamic forces. Although model-free methods can estimate and compensate the unmodeled aerodynamics, they apply small control input at each step to maintain the effective region of the state-input linearization. These approaches are not ideal to agile flights requiring more aggressive control inputs. Consequently, the control performance degrades (i.e., altitude error exceeds $1\,\mathrm{m}$ during transition) during highly agile maneuvers as demonstrated in Barth et al. (2020); Cheng and Pei (2022).

To further improve the control performance, model-based global controllers of varying sophistication have been proposed. For instance, Ritz and D'Andrea (2017) used a classic aerodynamics model to derive the desired attitude and thrust from the acceleration commands, by specifying the coordinated flight condition. To enable real-time implementation on a low-cost microcontroller, the aerodynamic model was simplified based on first-principles derivations, leading to considerable tracking errors. Zhou et al. (2017) also calculated the desired attitude but by solving a non-convex optimization using an accurate aerodynamic model obtained from wind tunnel test. However, this controller is computational demanding, which precludes real-time implementation. When the airspeed is zero, the definitions of angle of attack and sideslip angle become invalid, introducing singularity into the classic aerodynamic model used by these two research. There are studies employing alternative aerodynamic characterizations to avoid this singularity. Pucci et al. (2013) transformed a 2-dimension (2-D) planner VTOL (PVTOL) vehicle into an orientation-independent model, separating the computation of the vehicle thrust and orientation, thereby leading to a unified controller design (Pucci 2012). The author also derived the conditions, *spherical equivalency*, that airfoil aerodynamic characteristics must satisfy for the transformation to hold. Lustosa (2017) proposed a polynomial-like global aerodynamic parameterization, termed as the ϕ-theory model, and developed a linear quadratic regulator (LQR) based on the model. Their experiment results show that the LQR gain must be scheduled during the transition to avoid the instability in pitch angle caused by the model errors of ϕ-theory. Alternatively, Smeur et al. (2020) designed a global incremental nonlinear dynamic inversion (INDI) controller by linearizing the system at the current control inputs. To design the INDI controller, it requires knowing the current aerodynamic force (and moment) applied to the UAV and its

gradient with respect to (w.r.t.) the control input increment (pitch angle and velocity): the former one is obtained from inertial measurement units (IMUs), which suffer from either large measurement noise caused by constant propeller rotation or considerable filter delay; the latter is derived from a simple, heuristic aerodynamic model at a quasi-static condition where the flight path angle is zero. More recently, Tal and Karaman (2021) integrated the aforementioned ϕ-theory model and INDI technique into a global cascade PD controller applicable to both coordinated and uncoordinated flight. They also introduced feedforward jerk and yaw rate to improve the tracking performance and demonstrated indoor aerobatics (Tal and Karaman 2022). Compared to the previous INDI method (Smeur et al. 2020) with an over-simplified aerodynamic model, the ϕ-theory model used in Tal and Karaman (2021, 2022) can significantly increase the control accuracy. However, similar issues, i.e., significant measurement noise or filter delay still persist in INDI-based methods. Moreover, the ϕ-theory models have limited fitting capability, leading to larger model errors compared with the classic model, as confirmed by the authors in Lustosa et al. (2019). To sum up, the existing mode-based global control methods typically make compromise between model fidelity and computational load. While high-fidelity models are costly and impractical for real-time implementation, simplified models are relatively easy to be estimated from limited experimental data, but tend to degrade the control accuracy to varying extents.

Our method aims to fully exploit the UAV's actual aerodynamics to achieve high-accuracy and real-time control performance. Compared to the existing works, our proposed control scheme has the following advantages. **1)** Existing works either give up the vehicle aerodynamic model (e.g., model-free methods (Barth et al. 2020; Cheng and Pei 2022)) or compromise to simplified models (e.g., simplified classic model (Ritz and D'Andrea 2017), spherical equivalence model (Pucci et al. 2013), steady-level-flight model (Smeur et al. 2020) and ϕ-theory model (Lustosa 2017; Tal and Karaman 2022)), while our proposed controller leverages classic aerodynamic models without any simplification on its aerodynamic coefficients. The use of high-fidelity aerodynamic model is crucial to achieve higher control accuracy. **2)** Existing controllers either ignore wind effect in the environment (e.g. Ritz and D'Andrea (2017); Lustosa (2017)), or compensate the disturbance through incremental control updates from increased control error (e.g. (Smeur et al. 2020; Tal and Karaman 2022)), while our proposed approach incorporates wind effect by adjusting the reference trajectory (e.g., attitude) to maintain coordinated flight based on the differential flatness, and then tracks the adjusted trajectory in real time. Given the considerable aerodynamic efficiency loss of tail-sitter in windy conditions (Vourtsis et al. 2023), our proposed feedforward strategy compensates the wind effect in an pre-emptive way before the control error actually accumulates. **3)** Existing controllers (Ritz and D'Andrea 2017; Tal and Karaman 2022) simultaneously track trajectories and process singularities, while our work decouples singularities from the tracking controller, by the two-statge architecture. Such separation isolates the singularity treatment from the state tracking controller. **4)** An on-manifold MPC is proposed for trajectory tracking in high accuracy. MPC tracks full states by solving a finite-horizon optimization at each step to yield the best future behavior based on the system model (Borrelli et al. 2017). Its predictive nature that exploits the information of the future reference trajectory, contributes to a higher control bandwidth for trajectory tracking. In contrast, existing works (Ritz and D'Andrea 2017; Barth et al. 2020; Smeur et al. 2020; Tal and Karaman 2022) usually track the position trajectory in a cascaded control structure (e.g., a position controller followed by an attitude controller), which simplifies the outer loop design, but simultaneously constrains the outer loop's bandwidth. Admittedly, MPC is more computationally demanding and its convergence is challenging to guarantee, but its predictive nature and constraint handling capability have led to a wealth of successful robotic applications, such as the leading-edge Boston Dynamics Atlas humanoid robot (Marion 2021), drone racing (Foehn et al. 2021) and aerobatics (Kaufmann et al. 2020; Lu et al. 2022). In summary, a comparison of our work with those existing state-of-the-art global controllers is presented in Table 1.

2.2 Tail-sitter trajectory generation

Depending on the control strategy reviewed above, there are different trajectory generation algorithms for tail-sitters in literature. For separated control strategies, trajectories are generated separately for each phase. When the tail-sitter dynamics reduce to a rotary-wing model in low-speed vertical flight, well-established trajectory generation methods for quadrotors (or multicopters) (Mellinger and Kumar 2011; Mueller et al. 2015) are applicable directly. Trajectory planners for quadrotors can be also applied for high-level autonomy, such as obstacle avoidance and autonomous navigation. Similarly, traditional fixed-wing planners (Park et al. 2004; Chitsaz and LaValle 2007) can be adapted for tail-sitter in level flights. For example, the L1 guidance proposed by Park et al. (2004) has been widely used in prototype verification (Frank et al. 2007; Jung and Shim 2012; Verling et al. 2016) and commercial Autopilots (Meier et al. 2015) for tail-sitter level flights.

Generating a transition trajectory between vertical and level flights is relatively challenging due to the nonlinear aerodynamics during the transition. The intuitive linear transition method, which designs linearly increasing or decreasing pitch angle and constant altitude command (Verling et al. 2016; Lyu et al. 2017a) as mentioned before, does not consider the dynamical feasibility, thus requiring a lot of empirical trials and errors. To incorporate dynamic feasibility, trajectory generation is usually formulated into nonlinear optimization problems subject to different control objectives and constraints. For instance, Kita et al. (2010) calculated a pitch angle and thrust profile achieving the shortest transition time. Naldi and Marconi (2011) considered a minimum-time and minimum-energy optimal transition problem, while Oosedo et al. (2017) and Li et al. (2020a) respectively minimized the flight time and energy to maintain a constant altitude during the transition flight. However, solving these non-convex optimization problems is computationally expensive, preventing from onboard implementation and online replanning. These methods are also confined to straight-line transition that cannot be

extended to other maneuvers like transition with banked turns to avoid obstacles. Simplified dynamic models like the point-mass model (McIntosh and Mishra 2022) can be used to expedite obstacle-free planning, but again, the dynamical feasibility is omitted. Overall, existing separated trajectory generation approaches generate simple trajectories with limited maneuverability, making them only suitable for flights in open areas. The aerodynamic simplification and kinodynamic limitation prevent them from being extended to dynamically feasible and agile flights in cluttered environments.

Compared to the separated trajectory generation, designing a dynamically feasible trajectory that spans the entire envelope is a significantly more complex task because the tail-sitter is an under-actuated system with extremely nonlinear aerodynamics. For under-actuated mechanical systems, such as tail-sitters, the differential flatness is an essential property that can significantly ease trajectory generation. If a dynamic system is differentially flat, its full states and inputs can be determined by algebraic functions of flat outputs and their derivatives (Fliess et al. 1995; Murray et al. 1995). This property simplifies the trajectory generation problem to a set of algebraic operations in the flat-output space. This is a significant reduction in complexity compared to the state-space planning, which usually has to take into account on-manifold kinematic constraints. For example, the differential flatness property of quadrotors (Mellinger and Kumar 2011; Faessler et al. 2017) has been thoroughly studied and enabled a variety of applications in trajectory planning.

Research on the differential flatness of tail-sitter UAVs is scarce due to the extremely complicated, nonlinear aerodynamics mentioned above. Early research based on simplified models can trace back to 1990s. Hauser et al. (1992); Martin et al. (1996) studied the differential flatness and control of a simple 2-D PVTOL aircraft. Van Nieuwstadt and Murray (1998) simply considered the transition dynamics as a nominal flat system where the aerodynamics are treated as perturbations. Recently, Tal and Karaman (2021) showed the differential flatness based on the ϕ-theory aerodynamic model. The vehicle position and yaw angle are chosen as flat outputs, which allow for a global framework of trajectory optimization (Tal et al. 2023). The optimization could then be solved efficiently in the flat-output space, and the flatness transformation provides state projections (e.g., mapping acceleration to attitude) in a cascaded controller. Yet, this framework has certain theoretical limitations. First, the differential flatness is built on the coarse ϕ-theory aerodynamic model, the model errors of which degrade the trajectory quality and the resultant control performance. Second, the ϕ-theory model assumes a windless condition that only considers the vehicle attitude and velocity w.r.t. the fixed inertial frame, rather than the aerodynamic angles and airspeed. Lastly, the method must assume that the vehicle has no body or vertical rudder that produce side forces. Hence, this differential flatness property is not applicable to outdoor environments commonly with external winds or more general tail-sitter airframes.

Contrasted with early studies based on simplified 2-D models (Hauser et al. 1992; Martin et al. 1996), our work considers the full 3D model of a real tail-sitter UAVs. Furthermore, in comparison to recent research that used simplified aerodynamic model, such as the spherical equivalence model (Pucci et al. 2013) and the ϕ-theory model (Lustosa 2017), or that required particular airframe, such as configurations without vertical surfaces necessitated by (Tal et al. 2023), we prove the differential flatness property on accurate aerodynamic models and more general tail-sitter airframes. Based on the proved differential flatness, we propose a systematic trajectory generation framework for tail-sitter UAVs. High-quality trajectories are optimized subjecting to actuator constraints, flight time and dynamical feasibility.

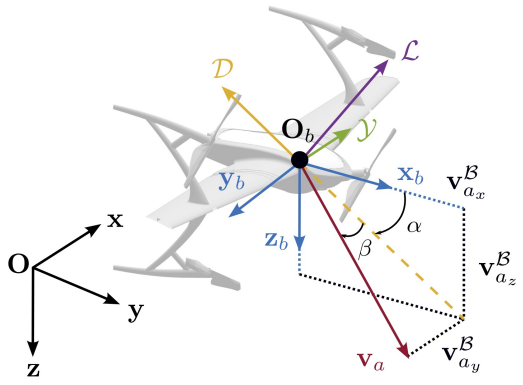


Fig. 1. Coordinate frames: the world frame, body frame and aerodynamic forces.

It is interesting to note that, both Tal and Karaman (2022), which assumes no vertical surfaces but with uncoordinated flight, and ours, which assumes coordinated flight, eventually lead to the same effect of avoiding lateral forces. The lateral force would dramatically complicate the solving of the UAV state (i.e., attitude and thrust) due to the highly nonlinear aerodynamic forces. Zhou et al. (2017) solves these highly nonlinear constraints by leveraging numerical approach, leading to high computational complexity not suitable for real-time implementation. Instead, avoiding such lateral force could effectively isolate and solve the angle of attack in our work (or Pitch angle in Tal and Karaman (2022)), hence the rest UAV states.

3 Flight dynamics

This section introduces the dynamic models that describe the motion of tail-sitters. We define coordinate frames for tail-sitter modeling, trajectory generation, and tracking control in Section 3.1. The dynamic model of the tail-sitter is presented in Section 3.2 and Section 3.3 introduces the classic aerodynamic models.

3.1 Coordinate frames

As shown in Fig. 1, the definition of coordinate frames follows the convention of traditional fixed-wing aircraft. The world frame $\{\mathbf{O}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$ denoted North-East-Down (NED), is considered as the inertial frame. The body frame $\{\mathbf{O}_b, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b\}$ is defined as Forward-Right-Down where the body axis $\mathbf{x}_b$ points along the nose of the aircraft and $\mathbf{O}_b$ is the vehicle center of gravity.

3.2 Airframe dynamics

We view the whole body of the tail-sitter as a rigid body. Referring to the Newton-Euler equations, the translational and rotational dynamics of the aircraft is modeled as follows:

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

$$\dot{\mathbf{v}} = \mathbf{g} + a_T \mathbf{R}\mathbf{e}_1 + \frac{1}{m}\mathbf{R}\mathbf{f}_a \tag{1b}$$

$$\dot{\mathbf{R}} = \mathbf{R}\lfloor\boldsymbol{\omega}\rfloor \tag{1c}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} + \mathbf{M}_a - \boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega} \tag{1d}$$

where $\mathbf{p}$ and $\mathbf{v}$ are respectively the vehicle position and velocity in the inertial frame, $\boldsymbol{\omega}$ is the angular velocity in the body frame, $\mathbf{R}$ denotes the rotation from the inertial frame to the body frame, m is the total mass of the aircraft, $\mathbf{J}$ is the inertia matrix and $\mathbf{g} = [0\ 0\ 9.8]^T$ is the gravity vector in the inertial frame. a_T and $\boldsymbol{\tau}$ denote the thrust acceleration scalar and control moment vector produced by actuators (e.g., four motors for a quadrotor tail-sitter). $\mathbf{f}_a$ and $\mathbf{M}_a$ are the aerodynamic force and moment in the body frame, respectively. The notation $\lfloor\mathbf{a}\rfloor$ converts a 3-D vector $\mathbf{a}$ into a skew-symmetric matrix such that $\mathbf{a}\times\mathbf{b} = \lfloor\mathbf{a}\rfloor\mathbf{b}, \forall\mathbf{a},\mathbf{b}\in\mathbb{R}^3$. $\mathbf{e}_1 = [1\ 0\ 0]^T, \mathbf{e}_2 = [0\ 1\ 0]^T, \mathbf{e}_3 = [0\ 0\ 1]^T$ are unit vectors used in the remaining of the paper.

Collecting all the state and input elements of the dynamics (1) leads to the system state and input below:

$$\mathbf{x}_{\text{full}} = (\mathbf{p},\mathbf{v},\mathbf{R},\boldsymbol{\omega}) \in \mathbb{R}^3\times\mathbb{R}^3\times SO(3)\times\mathbb{R}^3 \tag{2a}$$

$$\mathbf{u}_{\text{full}} = (a_T,\boldsymbol{\tau}) \in \mathbb{R}\times\mathbb{R}^3 \tag{2b}$$

Note that in the above model, we assume that the thrust direction is aligned to the body X axis $\mathbf{x}_b$. For cases where the thrust has a fixed installation angle, it can be trivially handled by re-defining the body frame.

3.3 Aerodynamics

Referring to (Etkin and Reid 1959), the aerodynamic force $\mathbf{f}_a$ is modeled in the body frame as follows:

$$\mathbf{f}_a = \begin{bmatrix}\mathbf{f}_{a_x}\\ \mathbf{f}_{a_y}\\ \mathbf{f}_{a_z}\end{bmatrix} = \begin{bmatrix}-\cos\alpha & 0 & \sin\alpha\\ 0 & 1 & 0\\ -\sin\alpha & 0 & -\cos\alpha\end{bmatrix}\begin{bmatrix}\mathcal{D}\\ \mathcal{Y}\\ \mathcal{L}\end{bmatrix} \tag{3}$$

where α is the angle of attack. The force components $\mathcal{L},\mathcal{D},\mathcal{Y}$ are the lift, drag, and side force, respectively. The aerodynamic moment vector $\mathbf{M}_a$ consists of rolling L, pitching M and yawing N moment along the body axis $\mathbf{x}_b,\mathbf{y}_b,\mathbf{z}_b$:

$$\mathbf{M}_a = \begin{bmatrix}L & M & N\end{bmatrix}^T \tag{4}$$

The force and moment components $\mathcal{L},\mathcal{D},\mathcal{Y},L,M,N$ can be written as products of non-dimensional coefficients, dynamic pressure $\frac{1}{2}\rho V^2$, the reference area S (e.g., the wing area), and the characteristic length $\bar{c}$ (e.g., the mean aerodynamic chord), as follows:

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}\rho V^2 S C_L(\alpha,\beta) & M &= \frac{1}{2}\rho V^2 S\bar{c} C_l(\alpha,\beta)\\ \mathcal{D} &= \frac{1}{2}\rho V^2 S C_D(\alpha,\beta) & N &= \frac{1}{2}\rho V^2 S\bar{c} C_m(\alpha,\beta)\\ \mathcal{Y} &= \frac{1}{2}\rho V^2 S C_Y(\alpha,\beta) & L &= \frac{1}{2}\rho V^2 S\bar{c} C_n(\alpha,\beta)\end{aligned} \tag{5}$$

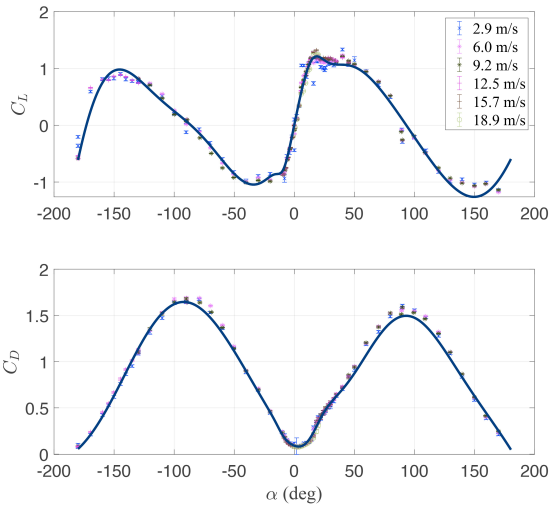


Fig. 2. Longitudinal aerodynamic coefficients C_L and C_D of our previous quadrotor tail-sitter UAV prototype, identified by wind tunnel tests (Lyu et al. 2018a).

where ρ is the air density and $V = \|\mathbf{v}_a\|$ is the norm of the airspeed. C_L, C_D, C_Y are the lift, drag, and side force coefficients, while C_l, C_m, C_n are the rolling, pitching, and yawing moment coefficients. The aerodynamic coefficients are functions of the angle of attack α and the sideslip angle β, depending on the design of the airfoil profile and the overall airframe. The accurate aerodynamic coefficients are usually identified by wind tunnel tests (Lyu et al. 2018a). For readability, the total aerodynamic force $\mathbf{f}_a$ in (3) can be rewritten as

$$\mathbf{f}_a = \frac{1}{2}\rho V^2 S\mathbf{c}(\alpha,\beta) \tag{6}$$

where

$$\mathbf{c}(\alpha,\beta) = \begin{bmatrix}\mathbf{c}_x(\alpha,\beta) & \mathbf{c}_y(\alpha,\beta) & \mathbf{c}_z(\alpha,\beta)\end{bmatrix}^T \tag{7a}$$

$$\mathbf{c}_x(\alpha,\beta) = -C_D(\alpha,\beta)\cos\alpha + C_L(\alpha,\beta)\sin\alpha \tag{7b}$$

$$\mathbf{c}_y(\alpha,\beta) = C_Y(\alpha,\beta) \tag{7c}$$

$$\mathbf{c}_z(\alpha,\beta) = -C_D(\alpha,\beta)\sin\alpha - C_L(\alpha,\beta)\cos\alpha \tag{7d}$$

Given the vehicle ground velocity $\mathbf{v}$ and wind speed $\mathbf{w}$ defined in the inertial frame, the airspeed $\mathbf{v}_a$, the angle of attack α and the sideslip angle β are calculated as follows:

$$\mathbf{v}_a = \mathbf{v} - \mathbf{w},\ \mathbf{v}_a^{\mathcal{B}} = \mathbf{R}^T\mathbf{v}_a = \begin{bmatrix}\mathbf{v}_{a_x}^{\mathcal{B}} & \mathbf{v}_{a_y}^{\mathcal{B}} & \mathbf{v}_{a_z}^{\mathcal{B}}\end{bmatrix}^T, \tag{8}$$

$$V = \|\mathbf{v}_a\|,\ \alpha = \tan^{-1}\left(\frac{\mathbf{v}_{a_z}^{\mathcal{B}}}{\mathbf{v}_{a_x}^{\mathcal{B}}}\right),\ \beta = \sin^{-1}\left(\frac{\mathbf{v}_{a_y}^{\mathcal{B}}}{V}\right) \tag{9}$$

We further assume that the airframe is symmetric to the body X-Z plane, which implies

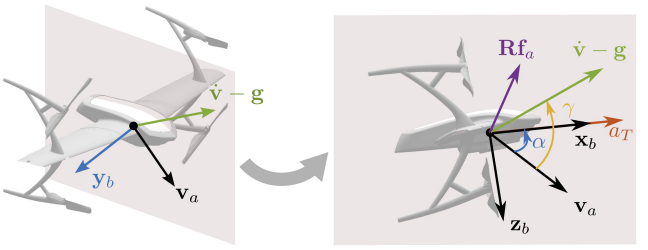
$$C_L(\alpha,\beta) = C_L(\alpha,-\beta),\ \forall\alpha,\beta \tag{10a}$$

$$C_D(\alpha,\beta) = C_D(\alpha,-\beta),\ \forall\alpha,\beta \tag{10b}$$

$$C_Y(\alpha,\beta) = -C_Y(\alpha,-\beta),\ \forall\alpha,\beta \tag{10c}$$

and hence $\forall\alpha$

$$C_Y(\alpha,0) = 0,\ \left.\frac{\partial C_L(\alpha,\beta)}{\partial\beta}\right|_{\beta=0} = \left.\frac{\partial C_D(\alpha,\beta)}{\partial\beta}\right|_{\beta=0} = 0 \tag{11a}$$



(a) Determining $\mathbf{y}_b$ (b) The plane of $\dot{\mathbf{v}}-\mathbf{g}$ and $\mathbf{v}_a$

Fig. 3. A tail-sitter UAV in coordinated flight: (a) the axis $\mathbf{y}_b$ is perpendicular to both of $\mathbf{v}_a$ and $\dot{\mathbf{v}}-\mathbf{g}$; (b) the angle of attack α and thrust acceleration a_T are determined on the longitudinal plane by the fact that the total acceleration comprising the drag acceleration $\mathcal{D}/m$, the lift acceleration $\mathcal{L}/m$, the a_T, and gravity $\mathbf{g}$ is equal to $\dot{\mathbf{v}}$.

$$\frac{\partial \mathbf{c}(\alpha,\beta)}{\partial \beta}|_{\beta=0} = \begin{bmatrix} 0 & \frac{\partial \mathbf{c}_y(\alpha,\beta)}{\partial \beta}\Big|_{\beta=0} & 0 \end{bmatrix}^T \tag{11b}$$

$$\frac{\partial \mathbf{c}(\alpha,\beta)}{\partial \alpha}|_{\beta=0} = \begin{bmatrix} \frac{\partial \mathbf{c}_x(\alpha,0)}{\partial \alpha} & 0 & \frac{\partial \mathbf{c}_z(\alpha,0)}{\partial \alpha} \end{bmatrix}^T \tag{11c}$$

4 Differential flatness in coordinated flight

In this section, we aim to investigate the fundamental differential flatness property which is the theoretical foundation for trajectory generation and tracking control. We prove that the tail-sitter is differentially flat in a flight condition known as the coordinated flight.

4.1 The coordinated flight

An aircraft in coordinated flight indicates a flight condition without sideslip (e.g., $\beta = 0, \mathbf{v}_{a_y}^{\mathcal{B}} = 0$) (Clancy 1975). This flight condition does not restrict the degree-of-freedom of the tail-sitter, which is still able to reach any position in the entire 3-D space. Moreover, the coordinated flight is usually preferred over uncoordinated flight (Stevens et al. 2015) for several practical reasons: **1**) the coordinated flight condition ideally achieves maximum aerodynamic efficiency and also minimizes undesirable aerodynamic moment that could cause spins. **2**) it is naturally required when the navigation sensors (e.g., cameras) mounted on the vehicle's nose have a limited FoV. **3**) restricting the sideslip angle around zero reduces the efforts for aerodynamic model identification by only requiring the longitudinal aerodynamic coefficients around $\beta = 0$ (see Fig. 2).

4.2 The differential flatness

Definition 1. (Fliess et al. 1995) A system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}), \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \text{rank}\left(\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right) = m$, is differentially flat, if there exists a flat output $\mathbf{y} \in \mathbb{R}^m$ of the form

$$\mathbf{y} = \mathbf{y}\left(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \cdots, \mathbf{u}^{(p)}\right) \tag{12}$$

such that the system state can be expressed explicitly by functions of the flat output and a finite number of its derivatives:

$$\mathbf{x} = \mathbf{x}\left(\mathbf{y}, \dot{\mathbf{y}}, \cdots, \mathbf{y}^{(q)}\right) \tag{13}$$

$$\mathbf{u} = \mathbf{u}\left(\mathbf{y}, \dot{\mathbf{y}}, \cdots, \mathbf{y}^{(q)}\right) \tag{14}$$

The definition of differential flatness formally requires an equal dimension of the control input and the selected flat output for a system with independent inputs. However, the control input $\mathbf{u}_{\text{full}}$ defined in (2) is not independent, due to the coordinated flight condition.

Theorem 1. Given the system dynamics in (1) and definition of state and input in (2), when the UAV performs coordinated flight, it holds that

$$\text{rank}\left(\frac{\partial \mathbf{f}(\mathbf{x}_{\text{full}}, \mathbf{u}_{\text{full}})}{\partial \mathbf{u}_{\text{full}}}\right) = 3 \tag{15}$$

Proof. The proof is given in Appendix A.

It is seen in Theorem 1 and its proof that two elements of the body angular velocity and consequently the control moment $\boldsymbol{\tau}$ are coupled, and the control input $\mathbf{u}_{\text{full}}$ reduced by one degree-of-freedom to maintain the coordinated flight condition. The reduced input dimension decreases the rank of derivative $\frac{\partial \mathbf{f}(\mathbf{x}_{\text{full}}, \mathbf{u}_{\text{full}})}{\partial \mathbf{u}_{\text{full}}}$ by one, resulting in a flat output vector with a dimension of three only.

Our choice of the flat output is the vehicle position $\mathbf{p} \in \mathbb{R}^3$ in the inertial frame. In the following, we prove that all of the vehicle states $\mathbf{x}_{\text{full}}$ and inputs $\mathbf{u}_{\text{full}}$ can be expressed by functions of $\mathbf{p}$ and its derivatives.

The position $\mathbf{p}$ and velocity $\mathbf{v}$ are simply $\mathbf{p}$ itself and its first-order derivatives, respectively. To express the attitude $\mathbf{R}$ as a function of $\mathbf{p}$ and its derivatives, we observe that in the coordinate flight, **1**) there is no airspeed along the body Y axis, implying that $\mathbf{y}_b$ is perpendicular to the airspeed $\mathbf{v}_a$; and **2**) because the aerodynamic sideslip force $\mathcal{Y}$ is zero (due to coordinated flight and symmetric airframe) and the thrust is in the body X-Z plane, there is no force (and hence acceleration) except gravity along the body Y axis. That is being said, the total acceleration excluding gravity, $\dot{\mathbf{v}} - \mathbf{g}$, has no projection on the body Y axis (i.e., $\mathbf{y}_b$ is perpendicular to $\dot{\mathbf{v}} - \mathbf{g}$). As shown in Fig. 3 (a), being perpendicular to both $\dot{\mathbf{v}} - \mathbf{g}$ and $\mathbf{v}_a$, $\mathbf{y}_b$ can only be in one of two opposite directions. We choose the one closest to the body Y axis determined at the previous time step, denoted as $\mathbf{y}_b^{\text{prev}}$, to prevent drastic attitude change:

$$r = \text{sign}\left((\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})) \cdot \mathbf{y}_b^{\text{prev}}\right) \tag{16}$$

$$\mathbf{y}_b = r\frac{\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})}{\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\|}, \quad \text{if } \|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| \neq 0 \tag{17}$$

where $\text{sign}(a)$ denotes the sign of $a \in \mathbb{R}$ and the scalar r denotes the direction of the body Y axis, ensuring that $\mathbf{y}_b \cdot \mathbf{y}_b^{\text{prev}} \geqslant 0$ (the angle between $\mathbf{y}_b$ and $\mathbf{y}_b^{\text{prev}}$ is always less than $90°$). $\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| = 0$ is a singularity condition that will be discussed in Section 4.3.

Next, we show how to solve the body Z axis $\mathbf{z}_b$ and body X axis $\mathbf{x}_b$. We note that the sideslip force is zero due to the coordinated flight, hence the aerodynamic force $\mathbf{f}_a$ reduces to $\mathbf{f}_a = \begin{bmatrix} \mathbf{f}_{a_x} & 0 & \mathbf{f}_{a_z} \end{bmatrix}^T$ and $\mathbf{Rf}_a = \mathbf{x}_b\mathbf{f}_{a_x} + \mathbf{z}_b\mathbf{f}_{a_z}$. Substituting $\mathbf{Rf}_a$ into (1b) leads to:

$$a_T\mathbf{x}_b + \frac{\mathbf{f}_{a_x}}{m}\mathbf{x}_b + \frac{\mathbf{f}_{a_z}}{m}\mathbf{z}_b + \mathbf{g} = \dot{\mathbf{v}} \tag{18}$$

Decomposing the equation along the direction of $\mathbf{x}_b$ and $\mathbf{z}_b$ respectively, we have (see Fig. 3 (b))

$$a_T = \mathbf{x}_b^T(\dot{\mathbf{v}} - \mathbf{g}) - \mathbf{f}_{a_x}/m \quad \mathbf{z}_b^T(\dot{\mathbf{v}} - \mathbf{g}) = \mathbf{f}_{a_z}/m \tag{19a}$$

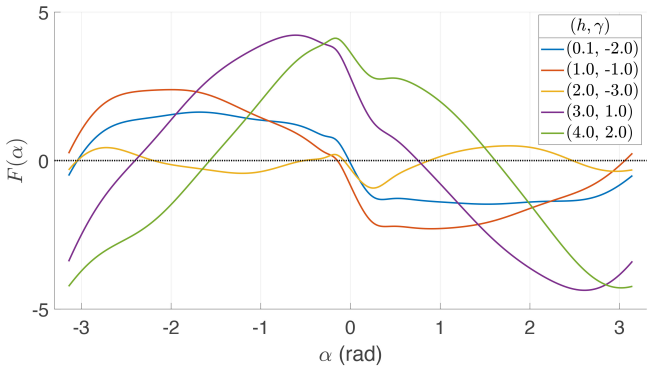


Fig. 4. Numerical examples of the root-finding problem of $F(\alpha)=0$ in (22) for five pairs of (h,γ) and the longitudinal aerodynamic coefficients shown in Fig. 2.

Since $\mathbf{x}_b$, $\mathbf{z}_b$, $\dot{\mathbf{v}}-\mathbf{g}$ and $\mathbf{v}_a$ are all perpendicular to $\mathbf{y}_b$, they should lie in the same plane (see Fig. 3 (b)). Hence we have $\mathbf{x}_b^T(\dot{\mathbf{v}}-\mathbf{g}) = \|\dot{\mathbf{v}}-\mathbf{g}\|\cos(\gamma-\alpha)$, $\mathbf{z}_b^T(\dot{\mathbf{v}}-\mathbf{g}) = \|\dot{\mathbf{v}}-\mathbf{g}\|\sin(\gamma-\alpha)$, and

$$a_T = \|\dot{\mathbf{v}}-\mathbf{g}\|\cos(\gamma-\alpha) - \mathbf{f}_{a_x}/m \tag{20a}$$
$$\|\dot{\mathbf{v}}-\mathbf{g}\|\sin(\gamma-\alpha) = -\mathbf{f}_{a_z}/m \tag{20b}$$

where

$$\gamma = r\cdot\operatorname{atan2}\left(\|(\dot{\mathbf{v}}-\mathbf{g})\times\mathbf{v}_a\|, (\dot{\mathbf{v}}-\mathbf{g})\cdot\mathbf{v}_a\right), \text{if } \|\mathbf{v}_a\|\neq 0 \tag{21}$$

and r denotes the angle direction (the positive direction of γ and α is defined such that rotating $\mathbf{v}_a$ along $\mathbf{y}_b$ will reach $\dot{\mathbf{v}}-\mathbf{g}$ and $\mathbf{x}_b$, respectively), while $\|\mathbf{v}_a\|\neq 0$ has been specified in $\|\mathbf{v}_a\times(\dot{\mathbf{v}}-\mathbf{g})\|\neq 0$ above.

It is noticed that (20b) only involves the known flat derivatives and the angle of attack α, which can hence be solved. Specifically, (20b) can be written as a nonlinear root-finding problem in terms of α:

$$F(\alpha) = h\sin(\gamma-\alpha) + \mathbf{c}_z(\alpha,0) = 0 \tag{22}$$

where

$$h = \frac{2m\|\dot{\mathbf{v}}-\mathbf{g}\|}{\rho V^2 S} \tag{23}$$

In the function of $F(\alpha)$, the variables h and γ is completely determined by the flight trajectory (and wind gust), while $\mathbf{c}_z(\alpha,0)$ is the third element of $\mathbf{c}$ in (7), which is completely determined by the actual aerodynamic configuration of the UAV. It should be also noted that γ and h are independent because they are respectively the angle and length ratio between $\dot{\mathbf{v}}-\mathbf{g}$ and $\mathbf{v}_a$. These properties allow us to investigate the shape of $F(\alpha)$, hence the solution of α, for a given pair of (h,γ). An example of such function $F(\alpha)$ is presented in Fig. 4. As can be seen, the equation $F(\alpha)=0$ is highly nonlinear due to the nonlinear aerodynamic model $\mathbf{c}_z(\alpha,0)$, hence no closed-form solution can be found in general. In practice, the equation can be solved numerically, such as Newton–Raphson method using $\mathbf{c}_z(\alpha)$ and $\partial\mathbf{c}_z(\alpha,0)/\partial\alpha$ identified in advance. Moreover, the extreme nonlinearity in $F(\alpha)$ also results in multiple solutions of α in most cases. To avoid the ambiguity and prevent drastic change of α, α^{prev}, the value of α determined at the previous time step, could be used as the initial guess for the numerical solver, to find a solution close to α^{prev}.

With the solved angle of attack α, the body X axis $\mathbf{x}_b$, and hence the rotation matrix $\mathbf{R}$, can be determined as

$$\mathbf{x}_b = \operatorname{Exp}(\alpha\mathbf{y}_b)\frac{\mathbf{v}_a}{\|\mathbf{v}_a\|}, \quad \text{if } \|\mathbf{v}_a\|\neq 0, \tag{24a}$$
$$\mathbf{R} = \begin{bmatrix}\mathbf{x}_b & \mathbf{y}_b & \mathbf{z}_b\end{bmatrix}, \quad \mathbf{z}_b = \mathbf{x}_b\times\mathbf{y}_b. \tag{24b}$$

where $\operatorname{Exp}(\cdot)$ is the exponential map on $SO(3)$ and $\|\mathbf{v}_a\|\neq 0$ has been specified in the singularity condition $\|\mathbf{v}_a\times(\dot{\mathbf{v}}-\mathbf{g})\|\neq 0$ above. With the solved α and $\beta=0$, the aerodynamic force $\mathbf{f}_a$ and system input a_T are determined by (6) and (20a), respectively.

Next, to show that the body angular velocity $\boldsymbol{\omega}$ is a function of the flat output, we take the time derivative of the translational dynamics (1b) as follows:

$$\begin{aligned}\ddot{\mathbf{v}} &= (\dot{a}_T\mathbf{R} + a_T\mathbf{R}\lfloor\boldsymbol{\omega}\rfloor)\mathbf{e}_1 \\ &\quad + \frac{1}{m}\mathbf{R}\left(\lfloor\boldsymbol{\omega}\rfloor\mathbf{f}_a + \frac{\partial\mathbf{f}_a}{\partial(\mathbf{R}^T\mathbf{v}_a)}\frac{d}{dt}(\mathbf{R}^T\mathbf{v}_a)\right) \\ &= \frac{1}{m}\mathbf{R}\frac{\partial\mathbf{f}_a}{\partial\mathbf{v}_a^{\mathcal{B}}}\mathbf{R}^T\dot{\mathbf{v}}_a + \dot{a}_T\mathbf{R}\mathbf{e}_1 \\ &\quad + \mathbf{R}\left(-\left\lfloor\left(a_T\mathbf{e}_1 + \frac{\mathbf{f}_a}{m}\right)\right\rfloor + \frac{1}{m}\frac{\partial\mathbf{f}_a}{\partial\mathbf{v}_a^{\mathcal{B}}}\lfloor\mathbf{v}_a^{\mathcal{B}}\rfloor\right)\boldsymbol{\omega}\end{aligned} \tag{25}$$

where $\partial\mathbf{f}_a/\partial\mathbf{v}_a^{\mathcal{B}}$ is evaluated at $\beta=0$ and can be obtained by taking derivative of (6) as below.

Theorem 2. Given the aerodynamic coefficients $\mathbf{c}(\alpha,\beta)$ of a symmetric airframe configuration satisfying (11), the partial derivative $\partial\mathbf{f}_a/\partial\mathbf{v}_a^{\mathcal{B}}$ at $\beta=0$ is

$$\frac{\partial\mathbf{f}_a}{\partial\mathbf{v}_a^{\mathcal{B}}} = \frac{\rho S}{2}\left(2\mathbf{c}\mathbf{v}_a^{\mathcal{B}^T} + \frac{\partial\mathbf{c}}{\partial\alpha}\mathbf{v}_a^{\mathcal{B}^T}\lfloor\mathbf{e}_2\rfloor + V\frac{\partial\mathbf{c}}{\partial\beta}\mathbf{e}_2^T\right) \tag{26}$$

Proof. The proof is given in Appendix B.

With the $\ddot{\mathbf{v}}, \dot{\mathbf{v}}_a, \mathbf{R}, \mathbf{f}_a, a_T$ and $\partial\mathbf{f}_a/\partial\mathbf{v}_a^{\mathcal{B}}$ solved above, the equation (25) forms three linear functions for $\dot{a}_T$ and $\boldsymbol{\omega}$. To solve $\dot{a}_T$ and $\boldsymbol{\omega}$ uniquely, we need to find one more equation. Recall that in coordinated flight the tail-sitter has no lateral airspeed: the condition requires zero lateral airspeed:

$$\mathbf{v}_{a_y}^{\mathcal{B}} = \mathbf{e}_2^T\mathbf{R}^T\mathbf{v}_a \equiv 0 \tag{27}$$

which leads to the derivative on the both sides:

$$\begin{aligned}&-\mathbf{e}_2^T\lfloor\boldsymbol{\omega}\rfloor\mathbf{R}^T\mathbf{v}_a + \mathbf{e}_2^T\mathbf{R}^T\dot{\mathbf{v}}_a = 0 \\ \Rightarrow\quad &-\mathbf{v}_a^T\mathbf{R}\lfloor\mathbf{e}_2\rfloor\boldsymbol{\omega} + \mathbf{y}_b^T\dot{\mathbf{v}}_a = 0\end{aligned} \tag{28}$$

Combing (28) and (25), we obtain four linear equations in terms of the $\dot{a}_T$ and $\boldsymbol{\omega}$, which can hence be solved as:

$$\begin{bmatrix}\dot{a}_T \\ \boldsymbol{\omega}\end{bmatrix} = \mathbf{N}^{-1}\mathbf{h} = \begin{bmatrix}\mathbf{N}_1 \\ \mathbf{N}_2\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{h}_1 \\ \mathbf{h}_2\end{bmatrix}, \text{ if } \operatorname{rank}(\mathbf{N}) = 4 \tag{29}$$

where $\operatorname{rank}(\mathbf{N})<4$ is the second singularity condition that will be discussed in Section 4.3, and

$$\mathbf{h}_1 = \mathbf{y}_b^T\dot{\mathbf{v}}_a \tag{30a}$$

$$\mathbf{h}_2 = \ddot{\mathbf{v}} - \frac{1}{m}\mathbf{R}\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}}\mathbf{R}^T\dot{\mathbf{v}}_a \tag{30b}$$

$$\mathbf{N}_1 = \begin{bmatrix} 0 & \mathbf{v}_a^T\mathbf{R}\lfloor \mathbf{e}_2 \rfloor \end{bmatrix} \tag{30c}$$

$$\mathbf{N}_2 = \begin{bmatrix} \mathbf{R}\mathbf{e}_1 & \mathbf{R}\left(-\lfloor \left(a_T\mathbf{e}_1 + \frac{\mathbf{f}_a}{m}\right)\rfloor + \frac{1}{m}\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}}\lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor\right) \end{bmatrix} \tag{30d}$$

Furthermore, the angular acceleration $\dot{\boldsymbol{\omega}}$ can be attained by further taking the derivative of (29):

$$\begin{bmatrix} \ddot{a}_T \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \frac{d}{dt}\left(\mathbf{N}^{-1}\mathbf{h}\right) = -\mathbf{N}^{-1}\dot{\mathbf{N}}\mathbf{N}^{-1}\mathbf{h} + \mathbf{N}^{-1}\dot{\mathbf{h}} \tag{31}$$

where the matrix derivative $\dot{\mathbf{N}}$ and $\dot{\mathbf{h}}$ are given in Appendix C. It is noted that the coefficient gradients $\partial^2\mathbf{c}_z(\alpha,0)/\partial\alpha^2$ (hence $\partial^2 C_L(\alpha,0)/\partial\alpha^2$ and $\partial^2 C_D(\alpha,0)/\partial\alpha^2$) should be further provided. Then the control moment $\boldsymbol{\tau}$, is solved from (1d) as

$$\boldsymbol{\tau} = \mathbf{J}\dot{\boldsymbol{\omega}} - \mathbf{M}_a + \boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega} \tag{32}$$

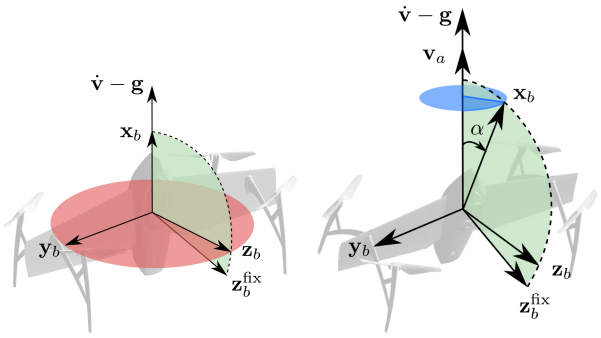
where the aerodynamic moments $\mathbf{M}_a$ is calculated from (5) based on $\beta = 0$ and the α solved above.

Remark 1. Formally, the flatness functions are real-analysis by the classic definition. However, when deriving the flatness function of angle of attack α, we cannot find its closed-form solution for a general aerodynamic model due to the extreme nonlinearity. Fortunately, we reduce this problem into a one-dimensional root-finding problem as shown in Fig. 4, that can be solved efficiently by numerical methods in real-time computation. Except α, the remaining flatness functions are all given explicitly.

Remark 2. In the aerodynamic model (6) and the differential flatness derivation above, we assumed that the aerodynamic force $\mathbf{f}_a$ depends only on the vehicle states (i.e., airspeed and attitude) but not the control inputs (i.e., moment $\boldsymbol{\tau}$ and thrust a_T). This is generally true for quadrotor tail-sitter VTOL UAVs where no extra flaps are used and the propellers are distant from wing hence the wing aerodynamic force $\mathbf{f}_a$ not depending on the propeller airflow. For tail-sitter UAVs whose moment $\boldsymbol{\tau}$ is produced by flaps at the trailing edge of wings, such as the twin-rotor tail-sitter UAV in Tal and Karaman (2022), the flaps deflection and propeller airflow would change the aerodynamic force $\mathbf{f}_a$, causing the aerodynamic force $\mathbf{f}_a$ to depend on the control inputs and preventing the solving of (22). This issue could be overcome practically by a strategy similar to Tal and Karaman (2022), which assumes very small changes of control inputs (i.e., flap deflections and propeller thrust) at each step, so that aerodynamic force $\mathbf{f}_a$ can be evaluated at the last flap angle and propeller thrust, and then used to solve $\boldsymbol{\omega}$ and $\boldsymbol{\tau}$ as detailed above.

4.3 Singularity conditions

We discuss the two conditions that singularities occur in the above flatness functions, one is $\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| = 0$ as specified in (17) and the other is $\text{rank}(\mathbf{N}) < 4$ as specified in (29). We first investigate the possible singularity condition where $\text{rank}(\mathbf{N}) < 4$, by calculating the determinant of $\mathbf{N}$ as follows:



(a) Singularity case $\|\mathbf{v}_a\| = 0$ (b) Singularity case $\gamma = 0$

Fig. 5. Determination of the vehicle body Y axis $\mathbf{y}_b$ (or body Z axis $\mathbf{z}_b$) under two singular conditions (a) $\|\mathbf{v}_a\| = 0$ (e.g., near hovering flights) and (b) $\gamma = 0$ (e.g., vertical takeoff or landing). In both figures, the green plane denotes the plane of $\dot{\mathbf{v}} - \mathbf{g}$ and $\mathbf{z}_b^{\text{fix}}$. In (a), the red circular plane perpendicular to $\mathbf{x}_b$ denotes all possible directions of $\mathbf{z}_b$. To minimize the yaw effort, $\mathbf{z}_b$ should the intersecting line of the green and red plane. In (b), the blue disk denotes all possible directions of $\mathbf{x}_b$. For each direction of $\mathbf{x}_b$, $\mathbf{z}_b$ could further rotate along $\mathbf{x}_b$ freely. To minimize the yaw effort, both $\mathbf{x}_b$ and $\mathbf{z}_b$ should be within the green plane.

Theorem 3. Given the aerodynamic coefficients $\mathbf{c}(\alpha,\beta)$ of a symmetric airframe configuration satisfying (11), the determinant of $\mathbf{N}$ defined in (30) is calculated as follows.

$$\det(\mathbf{N}) = -\frac{\rho S V^2}{2m}\frac{\partial F(\alpha)}{\partial \alpha}\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| \tag{33}$$

Proof. The proof is given in Appendix D.

As can be seen in (33), there are two cases that make $\mathbf{N}$ singular, one is $\frac{\partial F(\alpha)}{\alpha} = 0$ and the other is $\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| = 0$. Because the angle of attack α is solved from $F(\alpha) = 0$ in (22), the former condition essentially requires $F(\alpha)$ passing trough zero with a zero slope, a condition that rarely occurs for actual aerodynamic configuration $\mathbf{c}_z(\alpha,\beta)$ (see Fig.4). Therefore, the singularity condition $\text{rank}(\mathbf{N}) < 4$ reduces to the first singularity condition $\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| = 0$, which has to be considered. This singularity condition breaks into the following three sub-conditions:

$$\|\dot{\mathbf{v}} - \mathbf{g}\| = 0 \tag{34a}$$

$$\|\mathbf{v}_a\| = 0 \tag{34b}$$

$$\gamma = 0, \pm\pi \tag{34c}$$

We investigate the corresponding flight status for these three condition as follows.

4.3.1 Singularity sub-conditions 1. $\|\dot{\mathbf{v}} - \mathbf{g}\| = 0$ The sub-condition $\|\dot{\mathbf{v}} - \mathbf{g}\| = 0$ is the case where the vehicle is free falling, which is undesired in usual flights and should be avoided in the trajectory planning. Therefore, this sub-condition would not be encountered in practice.

4.3.2 Singularity sub-conditions 2. $\|\mathbf{v}_a\| = 0$

The second sub-condition $\|\mathbf{v}_a\| = 0$ corresponds to zero airspeed, which occurs when the vehicle hovers in windless environments such as indoor places, or flies in the same velocity as the wind in outdoor environments. When the airspeed $\mathbf{v}_a$ is zero, (17) becomes singular and hence cannot determine $\mathbf{y}_b$. Actually, even when $\mathbf{v}_a$ is close to zero,

(17) will be ill-conditioned, where a small change in $\mathbf{v}_a$ may cause drastic orientation change in $\mathbf{y}_b$. To avoid this ill condition, we choose a small velocity threshold $v_{\min}$ (e.g. $v_{\min}$ =0.5 m/s). When $\|\mathbf{v}_a\| < v_{\min}$, the aerodynamic force $\mathbf{f}_a$, which is quadratic to $\|\mathbf{v}_a\|$, can be safely ignored. Substituting $\mathbf{f}_a = \mathbf{0}$ into (18) leads to

$$\mathbf{x}_b = \frac{\dot{\mathbf{v}} - \mathbf{g}}{\|\dot{\mathbf{v}} - \mathbf{g}\|}, \quad a_T = \|\dot{\mathbf{v}} - \mathbf{g}\| \tag{35}$$

For the axis $\mathbf{y}_b$ (or equivalently, $\mathbf{z}_b$), it could be any direction perpendicular to $\mathbf{x}_b$ without affecting the solution in (35) (see Fig. 5(a)). To minimize the unnecessary efforts for yawing control, we fix the vehicle yaw angle at the value of yaw angle just before $\|\mathbf{v}_a\| < v_{\min}$ took place (e.g., when the vehicle decelerates to hover) or the value of yaw angle at initial time (e.g., when the vehicle just took off from the ground). Since the yaw angle is represented by the body Z axis, we hope to find a $\mathbf{z}_b$ that has the smallest angle with $\mathbf{z}_b^{\text{fix}}$, the vehicle body Z axis just before $\|\mathbf{v}_a\| < v_{\min}$ took place or at initial time. This essentially causes $\mathbf{z}_b$ to lie on the plane formed by $\mathbf{x}_b$ and $\mathbf{z}_b^{\text{fix}}$ (see Fig. 5(a)), which, in return, leads $\mathbf{y}_b$ to be perpendicular to $\mathbf{x}_b$ (i.e., $\dot{\mathbf{v}} - \mathbf{g}$) and $\mathbf{z}_b^{\text{fix}}$:

$$\mathbf{y}_b = \frac{\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})}{\|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\|} \tag{36}$$

With $\mathbf{x}_b$ and $\mathbf{y}_b$, the vehicle attitude can be determined by (24b).

Next, to determine the body angular velocity, we notice $\left(\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\right)^T \mathbf{z}_b \equiv 0$ always holds. Taking time derivative on both sides and recalling that $\mathbf{z}_b^{\text{fix}}$ is a prescribed constant vector, we have

$$\begin{aligned}\left(\lfloor \mathbf{z}_b^{\text{fix}} \rfloor \ddot{\mathbf{v}}\right)^T \mathbf{z}_b &= \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \mathbf{y}_b^T \mathbf{R} \lfloor \mathbf{e}_3 \rfloor \boldsymbol{\omega} \\ &= \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \mathbf{e}_1^T \boldsymbol{\omega}\end{aligned} \tag{37}$$

Moreover, neglecting the aerodynamics, the derivative of translational dynamics in (25) can be rewritten as

$$\ddot{\mathbf{v}} = \left(\dot{a}_T \mathbf{R} + a_T \mathbf{R} \lfloor \boldsymbol{\omega} \rfloor\right) \mathbf{e}_1 \tag{38}$$

Combining (37) and (38), both the $\dot{a}_T$ and $\boldsymbol{\omega}$ can be solved from the a 4-D linear equations in the same form as (29) with sub-matrices of $\mathbf{h}$ and $\mathbf{N}$ are rewritten as follows:

$$\mathbf{h}_1 = \left(\lfloor \mathbf{z}_b^{\text{fix}} \rfloor \ddot{\mathbf{v}}\right)^T \mathbf{z}_b \tag{39a}$$

$$\mathbf{h}_2 = \ddot{\mathbf{v}} \tag{39b}$$

$$\mathbf{N}_1 = \begin{bmatrix} 0 & \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \mathbf{e}_1^T \end{bmatrix} \tag{39c}$$

$$\mathbf{N}_2 = \begin{bmatrix} \mathbf{R}\mathbf{e}_1 & -a_T \mathbf{R} \lfloor \mathbf{e}_1 \rfloor \end{bmatrix} \tag{39d}$$

Theorem 4. The determinant of $\mathbf{N}$ defined in (39) is calculated as

$$\det(\mathbf{N}) = -a_T^2 \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \tag{40}$$

Proof. The proof is given in Appendix E.1.

From (35), we have $a_T = \|\dot{\mathbf{v}} - \mathbf{g}\|$, which is not zero in practice (see Section 4.3.1). Therefore, the only requirement for both (36) and $\det(\mathbf{N}) \neq 0$ is $\|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \neq 0$, a condition that is always true because at the moment $\|\mathbf{v}_a\| \approx v_{\min}$, the body X axis $\mathbf{x}_b^{\text{fix}}$ is almost aligned with $\dot{\mathbf{v}} - \mathbf{g}$ (the aerodynamic force is negligible and the thrust must provide most of the special acceleration $\dot{\mathbf{v}} - \mathbf{g}$), meaning that $\mathbf{z}_b^{\text{fix}}$ cannot be parallel to $\dot{\mathbf{v}} - \mathbf{g}$.

Finally, the angular acceleration and control moment are also solved from (31) and (32), where the derivatives $\dot{\mathbf{h}}$ and $\dot{\mathbf{N}}$ are recalculated in Appendix E.2.

4.3.3 *Singularity sub-conditions 3.* $\gamma = 0, \pm\pi$

When the airspeed $\mathbf{v}_a$ and the acceleration $\dot{\mathbf{v}} - \mathbf{g}$ is parallel, the singularity sub-condition $\gamma = 0$ or $\gamma = \pm\pi$ occurs. A common possible case is that the vehicle performs vertical takeoff and landing when the wind speed is zero. In this case, the angle of attack α and thrust acceleration a_T can still be solved from (22) and (20a) respectively, but the body Y axis cannot be determined from (17), which is singular. Actually, even when γ is close to zero, (17) will be ill-conditioned, where a small change in $\mathbf{v}_a$ or $\dot{\mathbf{v}} - \mathbf{g}$ may cause drastic orientation change in $\mathbf{y}_b$. To avoid this ill condition, we choose a small angle threshold $\gamma_{\min}$ (e.g. $\gamma_{\min} = 5^\circ$) that defines a set $\Gamma = (-\pi, -\pi + \gamma_{\min}) \cup (-\gamma_{\min}, \gamma_{\min}) \cup (\pi - \gamma, \pi]$. When $\gamma \in \Gamma$, we minimize the unnecessary yaw control efforts by restricting the axes $\mathbf{x}_b$ and $\mathbf{z}_b$ within the plane formed by $\mathbf{v}_a$ (or $\dot{\mathbf{v}} - \mathbf{g}$) and $\mathbf{z}_b^{\text{fix}}$, the vehicle body Z axis just before $\gamma \in \Gamma$ occurs or at the initial time. As a result, the body Y axis is perpendicular to $\mathbf{v}_a$ (or $\dot{\mathbf{v}} - \mathbf{g}$) and $\mathbf{z}_b^{\text{fix}}$ and is hence determined from (36). With $\mathbf{y}_b$, the body X axis and the vehicle attitude are determined from (24).

To solve the body angular velocity, we take the time derivative to the constraint $\left(\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\right)^T \mathbf{z}_b \equiv 0$ which is identical to (37). Combing this constraints with the derivative of the translational dynamics in (25), the body angular velocity is solved in the same form as (29) where the sub-matrices are given from (30) for $\mathbf{h}_2, \mathbf{N}_2$ and (39) for $\mathbf{h}_1, \mathbf{N}_1$:

$$\mathbf{h}_1 = \left(\lfloor \mathbf{z}_b^{\text{fix}} \rfloor \ddot{\mathbf{v}}\right)^T \mathbf{z}_b \tag{41a}$$

$$\mathbf{h}_2 = \ddot{\mathbf{v}} - \frac{1}{m} \mathbf{R} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \mathbf{R}^T \dot{\mathbf{v}}_a \tag{41b}$$

$$\mathbf{N}_1 = \begin{bmatrix} 0 & \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \mathbf{e}_1^T \end{bmatrix} \tag{41c}$$

$$\mathbf{N}_2 = \begin{bmatrix} \mathbf{R}\mathbf{e}_1 & \mathbf{R}\left(-\left\lfloor \left(a_T \mathbf{e}_1 + \frac{\mathbf{f}_a}{m}\right) \right\rfloor + \frac{1}{m} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor\right) \end{bmatrix} \tag{41d}$$

Theorem 5. The determinant of $\mathbf{N}$ defined in (41) is calculated as

$$\det(\mathbf{N}) = \frac{\rho S V^2}{2m} \frac{\partial F(\alpha)}{\partial \alpha} \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \psi_{23} \tag{42}$$

where

$$\psi_{23} = \|\dot{\mathbf{v}} - \mathbf{g}\| \cos(\gamma - \alpha) - \frac{\rho S V^2}{2m} \frac{\partial \mathbf{c}_y}{\partial \beta} \cos\alpha \tag{43}$$

Proof. The proof is given in Appendix F.1.

It is seen in (33) that three possible cases making $\mathbf{N}$ singular, $\frac{\partial F(\alpha)}{\alpha} = 0$, $\|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| = 0$, or $\psi_{23} = 0$. As the discussion to Theorem 3, since the angle of attack α is solved from $F(\alpha) = 0$, the former condition requires

$F(\alpha)$ to pass through zero with a zero slope, which rarely occurs for actual aerodynamic configuration $\mathbf{c}_z(\alpha,\beta)$. For the second condition $\|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}}-\mathbf{g})\| = 0$, since the current singularity case occurred at the vertical ascending or descending flights, the thrust should provide the major special acceleration $\dot{\mathbf{v}}-\mathbf{g}$. Since the thrust is aligned with body X axis, the direction $\dot{\mathbf{v}}-\mathbf{g}$ should be most similar to $\mathbf{x}_b^{\text{fix}}$, not $\mathbf{z}_b^{\text{fix}}$, which rules out the condition $\|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}}-\mathbf{g})\| = 0$. For the third condition $\psi_{23}=0$, it requires a special $\frac{\partial \mathbf{c}_y(\alpha,\beta)}{\partial\beta}|_{\beta=0}$ that satisfies both of $F(\alpha)=0$ and (43), which generally does not hold in actual aerodynamic configurations. Therefore, the matrix $\mathbf{N}$ is non-singular in practice.

Finally, the angular acceleration and control moment are also solved from (31) and (32), but the derivatives $\dot{\mathbf{h}}$ and $\dot{\mathbf{N}}$ are recalculated in Appendix F.2.

Remark 3. Singularity conditions $||\mathbf{v}_a|| = 0$ and $\gamma = 0$, is resolved in a unified manner of assigning $\mathbf{z}_b$ closest to a fixed direction $\mathbf{z}_b^{\text{fix}}$, which is equivalent to fixing the yaw angle. If the singularity conditions are caused by vehicles at low speed vertical flights (e.g., hovering, vertical take-off and landing), such fixing of yaw angle is unnecessary. For example, an extra yaw angle can be specified by assigning $\mathbf{z}_b^{\text{fix}}$, to achieve sensor-pointing and sideways maneuvering.

4.4 Differential flatness transform

In this section, we present a complete differential flatness transform that maps a flat-output trajectory to a state-input trajectory, based on the flatness functions in Section 4.2 with treatments for singularity conditions presented in Section 4.3. Since the flatness functions and singularity treatments are all based on $\mathbf{v}_a$, the airspeed, they could naturally incorporate the wind speed $\mathbf{w}$ into the inertial speed $\mathbf{v}$. In practice, we compute the airspeed as $\mathbf{v}_a = \mathbf{v} - \bar{\mathbf{w}}$, where $\bar{\mathbf{w}}$ is a surrogate wind speed to be compensated. In case of full wind speed compensation, we set $\bar{\mathbf{w}} = \mathbf{w}$ or else $\bar{\mathbf{w}} = \mathbf{0}$.

Combining all elements above, the complete differential flatness transform can be obtained as shown in Algorithm 1. With the transform, any flat-output trajectories can be mapped to the system state $\mathbf{x}_{\text{full}}$ and control input $\mathbf{u}_{\text{full}}$ as below:

$$\mathbf{x}_{\text{full}} = \mathcal{X}_{\text{full}}(\mathbf{p}^{(0:3)}), \quad \mathbf{u}_{\text{full}} = \mathcal{U}_{\text{full}}(\mathbf{p}^{(1:4)}), \tag{44}$$

where $\mathcal{X}_{\text{full}}(\mathbf{p}^{(0:3)})$ denotes the state flatness function of the flat output and its derivatives up to the third order, and $\mathcal{U}_{\text{full}}(\mathbf{p}^{(1:4)})$ denotes the input flatness function of the flat-output derivatives up to the fourth order.

Algorithm 1: Differential flatness transform

Given: a trajectory of flat output $\mathbf{p}^{(0:4)}$ avoiding $\|\dot{\mathbf{v}}-\mathbf{g}\| = 0$, current surrogate wind speed $\bar{\mathbf{w}}$, previous body Y axis $\mathbf{y}_b^{\text{prev}}$, velocity threshold $v_{\min}$, angle threshold $\gamma_{\min}$, and the body Z axis $\mathbf{z}_b^{\text{fix}}$ fixed just before $\|\mathbf{v}_a\| < v_{\min}$ or $\gamma \in \Gamma$.
Calculate the airspeed:
$\mathbf{v}_a = \mathbf{v} - \bar{\mathbf{w}}$;
Calculate the states and inputs:
if $\|\mathbf{v}_a\| < v_{\min}$ **then**
  Assign $\mathbf{y}_b$ perpendicular to $\dot{\mathbf{v}}-\mathbf{g}$ and $\mathbf{z}_b^{\text{fix}}$ (36);
  Determine $\mathbf{x}_b$ (35), a_T (35) and $\mathbf{R}$ (24b);
  Calculate $\mathbf{h}$ and $\mathbf{N}$ (39);
else
  Calculate γ (21);
  Solve α (22) and a_T (20a);
  if $\gamma \in \Gamma$ **then**
    Assign $\mathbf{y}_b$ perpendicular to $\dot{\mathbf{v}}$–$\mathbf{g}$ and $\mathbf{z}_b^{\text{fix}}$ (36);
    Determine $\mathbf{x}_b$ (24a) and $\mathbf{R}$ (24b);
    Calculate $\mathbf{h}$ and $\mathbf{N}$ (41);
  else
    Determine $\mathbf{y}_b$ (17), $\mathbf{x}_b$ (24a) and $\mathbf{R}$ (24b);
    Calculate $\mathbf{h}$ and $\mathbf{N}$ (30);
    Set $\mathbf{z}_b^{\text{fix}} = \mathbf{z}_b$;
  end
end
Solve $\boldsymbol{\omega}$ (29);
Solve $\dot{\boldsymbol{\omega}}$ (31) and $\boldsymbol{\tau}$ (32);
Set $\mathbf{y}_b^{\text{prev}} = \mathbf{y}_b$;

5 System overview

In this section, we present the entire framework of trajectory generation and tracking control for aggressive flights based on the fundamental differential flatness of the tail-sitter vehicle presented previously.

5.1 System reduction

The full system presented in (1) is of dimension 12, comprising the vehicle position, velocity, attitude and angular velocity. Note that the system has a cascaded structure, where the input torque $\boldsymbol{\tau}$ solely affects the angular velocity, and then the angular velocity determines the attitude, hence the velocity and position of the vehicle. Enabled by this cascaded dynamics, we propose to control the angular velocity dynamics (1d) separately (referred to as the "low-level control"). In the low-level control, the Coriolis term $\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}$ and aerodynamic moment $\mathbf{M}_a$ can be compensated in a feed forward way, while the remaining dynamics are first order linear systems that can be controlled by linear feedback controller (e.g., PID controller). More systematic and advanced control techniques could also be deployed, such as μ-synthesis (Noormohammadi-Asl et al. 2020), H_∞ loop shaping (Li et al. 2020b), Notch filters Xu et al. (2019a), to improve the controller bandwidth and robustness to model uncertainties (e.g., unknown flexible modes) and possible vibrations. Details of our low-level controller is shown in Section 8.1.

With a well-designed low-level controller, we assume that the vehicle angular velocity can be instantaneously achieved hence it can be viewed as the control input to the rest vehicle dynamics consisting of (1a, 1b, 1c) (the "high-level system"). As a result, the state and input of the high-level system are

$$\mathbf{x} = (\mathbf{p} \;\; \mathbf{v} \;\; \mathbf{R}) \tag{45a}$$
$$\mathbf{u} = (a_T \;\; \boldsymbol{\omega}) \tag{45b}$$

which are subject to the following system model

$$\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u}) = \begin{cases} \dot{\mathbf{p}} = \mathbf{v}, \\ \dot{\mathbf{v}} = \mathbf{g} + a_T\mathbf{R}\mathbf{e}_1 + \frac{1}{m}\mathbf{R}\mathbf{f}_a \\ \dot{\mathbf{R}} = \mathbf{R}\lfloor\boldsymbol{\omega}\rfloor \end{cases} \tag{46}$$

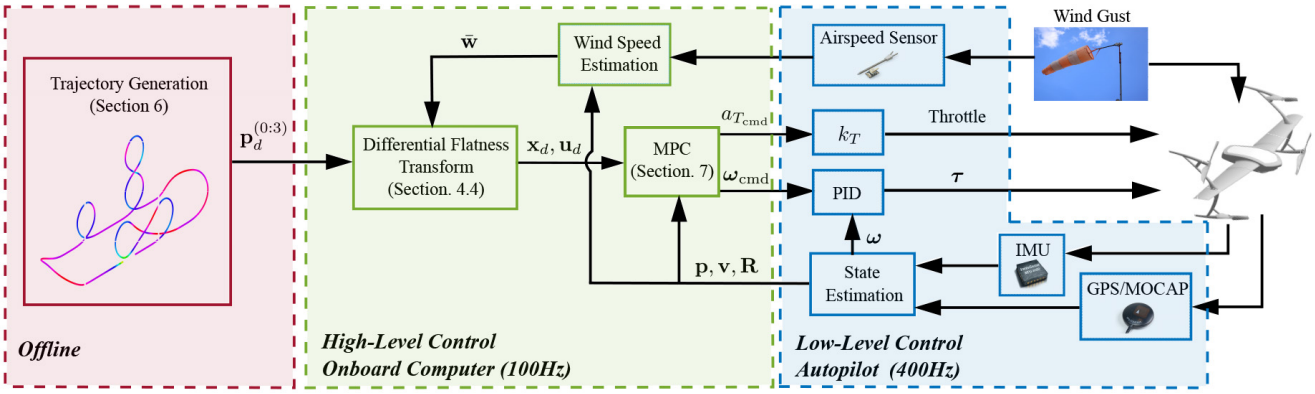


Fig. 6. System overview.

Since the state in (46) is a reduced set of the original one in (1), the reduced system is still differentially flat. Specifically, the state and input of the high-level system can be written as

$$\mathbf{x} = \mathcal{X}(\mathbf{p}^{(0:2)}), \quad \mathbf{u} = \mathcal{U}(\mathbf{p}^{(1:3)}) \tag{47}$$

where $\mathcal{X}(\cdot)$ and $\mathcal{U}(\cdot)$ are subsets of $\mathcal{X}_{\text{full}}(\cdot)$ and $\mathcal{U}_{\text{full}}(\cdot)$ from (44) and the corresponding state-input trajectory $(\mathbf{x}, \mathbf{u})$ will satisfy the high-level system model (46) subject to the surrogate wind speed $\bar{\mathbf{w}}$. The high-level system in (46) is of lower dimension and will be used for our trajectory planning and tracking control.

This system reduction presented above has both advantages and disadvantages. One advantage is the reduction of computation complexity in trajectory generation, avoiding the cumbersome derivative of moment $\boldsymbol{\tau}$ with respect to the flat-output and the knowledge of dynamic parameters. Another advantage is decoupling the low-level angular velocity control, which are highly related to the vehicle dynamics (e.g., flexible modes, motor delay, etc.), from the high-level system planning and tracking control. A disadvantage arises that the original dynamical feasibility (i.e., thrust and moment) is approximated as constraints on inputs of the reduced high-level system (i.e., thrust and angular velocity). This approximation becomes less accurate when the vehicle is maneuvering with rapidly varying angular velocity that necessitates large control moment. Despite the rough approximation, practical quadrotor applications in drone racing (Romero et al. 2022) and aerobatics (Kaufmann et al. 2020; Lu et al. 2022) demonstrate that the feasibility can be sufficiently guaranteed by constraining inputs of the reduced system in most cases.

5.2 System framework

With the system reduction above, the overview of our proposed approach is shown in Fig. 6. A flat-output trajectory up to the third-order smoothness (i.e, $\mathbf{p}_d^{(0:3)}$) is planned offline for the high-level system by a trajectory generation module (Section 6). For online trajectory tracking, we propose a two-stage control strategy. The first stage is differential flatness transform (Section 4.4) that maps the flat-out trajectory to the desired state and input trajectory $\mathbf{x}_d, \mathbf{u}_d$. This transform also incorporates environment wind (if enabled) and fixes the singularity conditions presented in Section 4.3. The computed state-input trajectories are then tracked in the second stage by an unified global on-manifold MPC, which computes the optimal control inputs $a_{T_{\text{cmd}}}$ and $\boldsymbol{\omega}_{\text{cmd}}$ (Section 7). These commands are then sent as reference to the low-level controller.

6 Trajectory generation

Since the vehicle dynamics is differentially flat in coordinated flight as proved in Section 4, all states and inputs can be expressed by flatness functions of the flat output and its derivatives. As a result, the trajectory generation problem reduces to low-dimensional algebra in the flat-output space (i.e., the vehicle position), without any integration of the under-actuated system dynamics in (46). We parameterize the vehicle position as polynomials (Bry et al. 2015; Mueller et al. 2015; Ding et al. 2019) and minimize the flight time and control efforts computed from the flatness functions (47), subject to necessary constraints.

6.1 Trajectory optimization

We formulate the trajectory planning as an optimization problem that finds a dynamically-feasible, smooth trajectory $\mathbf{p}(t) : \mathbb{R} \in [0, T_f] \mapsto \mathbb{R}^3$ with the minimum flight time T_f, control effort $\mathbf{u}$, and passing through a sequence of waypoints $\mathbf{Q} = (\mathbf{q}_0, \ \cdots \ \mathbf{q}_M)$. The purpose of waypoints are two-fold: 1) it could be used to obtain a collision-free trajectory when using with a front-end flight corridor (e.g., Liu et al. (2017); Gao et al. (2019)); and 2) specifying the location of the waypoints could change the shape of the flight trajectory, so that the desired aerobatic flight trajectories can be obtained. Given the initial state $\mathbf{s}_0$, terminal state $\mathbf{s}_f$ and waypoints $\mathbf{Q}$, the trajectory optimization is formulated as:

$$\min_{\mathbf{p}(t), T_f} \int_0^{T_f} \|\mathbf{u}\|_{\mathbf{W}}^2 dt + \rho T_f \tag{48a}$$

$$\text{s.t.} \quad \mathbf{x}(t) = \mathcal{X}\left(\mathbf{p}^{(0:2)}(t)\right), \ \mathbf{u}(t) = \mathcal{U}\left(\mathbf{p}^{(1:3)}(t)\right) \tag{48b}$$

$$\mathbf{p}^{(0:3)}(0) = \mathbf{s}_0, \ \mathbf{p}^{(0:3)}(T_f) = \mathbf{s}_f \tag{48c}$$

$$\mathbf{p}(t_{\mathbf{q}_i}) = \mathbf{q}_i, \ 0 \leqslant t_{\mathbf{q}_0} < \cdots < t_{\mathbf{q}_M} \leqslant T_f \tag{48d}$$

$$\mathbf{x}(t) \in \mathbb{X}, \ \mathbf{u}(t) \in \mathbb{U} \tag{48e}$$

$$\mathcal{S}(\mathbf{x}) \geqslant \boldsymbol{\epsilon} \tag{48f}$$

where $\mathbf{W} \in \mathbb{R}^{4\times4}$ is a positive diagonal matrix penalizing the total control effort and $\rho > 0$ is the flight time penalty. $\mathbb{X}$ denotes the kinodynamic constraint that ensures the vehicle

to operate within a safe workspace. The state constraints (48e) in this paper is the velocity condition

$$\|\mathbf{v}(t)\| \leqslant v_{\max} \tag{49}$$

where $v_{\max}$ is the maximum velocity for safe flight. $\mathbb{U} = \{\mathbf{u} \in \mathbb{R}^4 | \ \mathbf{u}_{\min} \leqslant \mathbf{u} \leqslant \mathbf{u}_{\max}\}$ is the boundary of the system inputs (i.e., the thrust acceleration a_T and angular velocity $\boldsymbol{\omega}$). $\mathcal{S}(\mathbf{x})$ denotes the singularity condition. Among the three singularity sub-conditions in Section 4.3, the conditions $\|\mathbf{v}_a\| < v_{\min}$ and $|\gamma| < \gamma_{\min}$ have been well treated, hence $\mathcal{S}(\mathbf{x})$ needs only to consider the first sub-condition:

$$\mathcal{S}(\mathbf{x}) = \|\dot{\mathbf{v}} - \mathbf{g}\|^2 \geqslant \epsilon^2 \tag{50}$$

where ϵ is a small positive value for numerical stability on implementation ($\epsilon = 0.1\ \mathrm{m/s^2}$ in this paper).

The optimization problem in (48) optimizes both the flat-output trajectory $\mathbf{p}(t)$ and the flight time T_f, to minimize the total control efforts and time in (48a). The minimization of control efforts tend to find smooth trajectories that are easier to track and the minimization of total time T_f tends to produce high-speed trajectories. Hence, the optimization (48) promises both trajectory smoothness and agility. The system state and control input in (48b) are characterized as the flatness functions that explicitly exploits the vehicle dynamic and kinematic models. The initial and terminal conditions of the trajectory are specified in (48c). The dynamical feasibility which indicates the actuation capability of the aircraft (or the tracking capability of the low-level control system) is guaranteed by the boundary constraints in (48e). The collision-free and the shape constraints of the trajectory could be achieved by satisfying the waypoint constraints in (48d). Finally, singularity conditions are incorporated into the constraint in (48f).

6.2 Trajectory optimization solving

The trajectory optimization (48) is a nonlinear constrained optimization problem. We leverage a state-of-the-art flight trajectory planning framework, MINCO (Wang et al. 2022), to parameterize and solve the trajectory. Referring to Wang et al. (2022), we insert N free control points $\mathbf{d}_i = (\mathbf{d}_{i_1}, \cdots, \mathbf{d}_{i_N}) \in \mathbb{R}^{N\times 3}$ between each two consecutive waypoints $\mathbf{q}_i, \mathbf{q}_{i+1}$ and create a waypoint sequence $\mathbf{r} = (\mathbf{q}_0, \mathbf{d}_0, \mathbf{q}_1, \cdots, \mathbf{q}_{M-1}, \mathbf{d}_{M-1}, \mathbf{q}_M) \in \mathbb{R}^{(M(N+1)+1)\times 3}$. The corresponding passing time for the waypoint sequence is $\mathbf{T} = [t_{\mathbf{q}_0}, t_{\mathbf{d}_{0_1}}, \cdots, t_{\mathbf{d}_{0_N}}, t_{\mathbf{q}_1}, \cdots, t_{\mathbf{q}_M}] \in \mathbb{R}^{M(N+1)+1}$. Then we characterize the trajectory by a multi-stage polynomial trajectory, where a 7th-order polynomial trajectory with C^4 continuity is used to connect to two consecutive points $\mathbf{r}_j, \mathbf{r}_{j+1} \in \mathbf{r}$ at their respective passing time $\mathbf{T}_j, \mathbf{T}_{j+1} \in \mathbf{T}$. The entire trajectory is therefore uniquely determined by all points $\mathbf{r}$ and respective passing time $\mathbf{T}$, having the endpoint constraint (48c) and waypoint constraint (48d) naturally satisfied. To deal with remaining boundary constraint (48e) and the singularity condition (48f), we relax these constraints to soft penalties in the objective function, hence transforming the constrained nonlinear optimization (48) into an unconstrained nonlinear optimization problem. The decision variables of the resultant optimization problem consist of control points $\mathbf{D} = (\mathbf{d}_0, \cdots, \mathbf{d}_{M-1}) \in \mathbb{R}^{MN\times 3}$ and passing time $\mathbf{T}$, which are solved by a quasi-Newton method (Wang et al. 2022).

To solve the unconstrained nonlinear optimization with a quasi-Newton method, gradients of the objective and constraints with respect to the decision variables $\mathbf{D}$ and $\mathbf{T}$ are needed. The gradients of flat-output (i.e, $\partial \mathbf{p}^{(1:3)}(t)/\partial \mathbf{D}$, $\partial \mathbf{p}^{(1:3)}(t)/\partial \mathbf{T}$) have been derived in detail in Wang et al. (2022), with which the gradients of the control input $\mathbf{u}$ and singularity condition in (48f) can be calculated by the chain rule:

$$\frac{\partial \mathcal{S}(\mathbf{x})}{\partial \mathbf{D}} = 2(\dot{\mathbf{v}} - \mathbf{g})^T \frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{D}} \tag{51a}$$

$$\frac{\partial \mathcal{S}(\mathbf{x})}{\partial \mathbf{T}} = 2(\dot{\mathbf{v}} - \mathbf{g})^T \frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{T}} \tag{51b}$$

$$\frac{\partial \mathbf{u}(t)}{\partial \mathbf{D}} = \frac{\partial \mathcal{U}\left(\mathbf{p}^{(1:3)}(t)\right)}{\partial \mathbf{p}^{(1:3)}(t)} \frac{\partial \mathbf{p}^{(1:3)}(t)}{\partial \mathbf{D}} \tag{51c}$$

$$\frac{\partial \mathbf{u}(t)}{\partial \mathbf{T}} = \frac{\partial \mathcal{U}\left(\mathbf{p}^{(1:3)}(t)\right)}{\partial \mathbf{p}^{(1:3)}(t)} \frac{\partial \mathbf{p}^{(1:3)}(t)}{\partial \mathbf{T}} \tag{51d}$$

where $\partial \mathcal{U}\left(\mathbf{p}^{(1:3)}(t)\right)/\partial \mathbf{p}^{(1:3)}(t)$ is gradients of the flatness functions in Section 4. The calculation is provided in Appendix G.

7 Global control for trajectory tracking

In this section, we develop a global tracking controller that allows a tail-sitter to accurately follow aggressive reference trajectories in real-world environments. Unlike conventional tail-sitter controllers operating in separate flight modes or existing global controllers considering a simplified aerodynamic model, the proposed global controller fully exploits the vehicle aerodynamics, contributing to accurate, agile flights within the entire envelope without encountering control switching or singularity.

7.1 The error-state system

The goal of the tracking controller is to drive the vehicle state to follow the desired reference state trajectory $\mathbf{x}_d$, which is computed from the trajectory planned in Section 6 via the flatness function (47). Equivalently, the error between the actual and reference state trajectory should converge to zero. Therefore, we only need to control the error state $\delta\mathbf{x}$.

7.1.1 Definition of the error state

Considering the tail-sitter model in (46), the system state evolves on a compound manifold below

$$\mathcal{M} = \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3), \quad \dim(\mathcal{M}) = 9 \tag{52}$$

$$\mathbf{x} = \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{R} \end{pmatrix} \in \mathcal{M}, \quad \mathbf{u} = \begin{bmatrix} a_T \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^4 \tag{53}$$

We assume that the trajectory planner generates a full reference trajectory, including the state $\mathbf{x}_d = (\mathbf{p}_d\ \mathbf{v}_d\ \mathbf{R}_d) \in \mathcal{M}$ and input $\mathbf{u}_d = \begin{bmatrix} a_{T_d} & \boldsymbol{\omega}_d^T \end{bmatrix}^T \in \mathbb{R}^4$. Note that the state-input trajectory $(\mathbf{x}_d, \mathbf{u}_d)$ satisfies the model (46) subject to the surrogate wind speed $\bar{\mathbf{w}}$.

Defining the error between the actual state $\mathbf{x}$ and the reference one $\mathbf{x}_d$, both lie on the state manifold $\mathcal{M}$, is

not trivial. We adopt the definition in our prior work (Lu et al. 2022), which defines the error state in the local homeomorphic space (an open set in Euclidean space) around each point $\mathbf{x}_d$. This particular error definition on manifold is denoted as $\boxminus$ (Hertzberg et al. 2013) detailed as below:

$$\delta\mathbf{x} \triangleq \mathbf{x}_d \boxminus \mathbf{x} = \begin{bmatrix} \delta\mathbf{p}^T & \delta\mathbf{v}^T & \delta\mathbf{R}^T \end{bmatrix}^T \in \mathbb{R}^9 \tag{54a}$$

$$\delta\mathbf{p} \triangleq \mathbf{p}_d \boxminus \mathbf{p} = \mathbf{p}_d - \mathbf{p} \in \mathbb{R}^3 \tag{54b}$$

$$\delta\mathbf{v} \triangleq \mathbf{v}_d \boxminus \mathbf{v} = \mathbf{v}_d - \mathbf{v} \in \mathbb{R}^3 \tag{54c}$$

$$\delta\boldsymbol{\theta} \triangleq \mathbf{R}_d \boxminus \mathbf{R} = \mathrm{Log}(\mathbf{R}^T\mathbf{R}_d) \in \mathbb{R}^3 \tag{54d}$$

where $\mathrm{Log}(\cdot)$ is the logarithmic map of the manifold $SO(3)$ and also the inverse of the exponential map $\mathrm{Exp}(\cdot)$. The control inputs are in the Euclidean space, so their errors can be defined directly:

$$\delta\mathbf{u} \triangleq \mathbf{u}_d - \mathbf{u} = \begin{bmatrix} \delta a_T & \delta\boldsymbol{\omega}^T \end{bmatrix}^T \in \mathbb{R}^4 \tag{55a}$$

$$\delta a_T \triangleq a_{T_d} - a_T \in \mathbb{R} \tag{55b}$$

$$\delta\boldsymbol{\omega} \triangleq \boldsymbol{\omega}_d - \boldsymbol{\omega} \in \mathbb{R}^3 \tag{55c}$$

7.1.2 The error-state system dynamics

To control the error state $\delta\mathbf{x}$ (54) to converge to zero, we need to obtain its dynamic model. To do so, we take the derivative of the error state with respect to time.

Theorem 6. Given the error state defined in (54), where the actual trajectory $(\mathbf{x}, \mathbf{u})$ satisfies (46) with the actual wind speed $\mathbf{w}$ and the reference trajectory $(\mathbf{x}_d, \mathbf{u}_d)$ satisfies (46) with the surrogate wind speed $\bar{\mathbf{w}}$, then the dynamics of the error-state system is:

$$\delta\dot{\mathbf{x}} = \begin{bmatrix} \delta\dot{\mathbf{p}}^T & \delta\dot{\mathbf{v}}^T & \delta\dot{\boldsymbol{\theta}}^T \end{bmatrix}^T \tag{56a}$$

$$\delta\dot{\mathbf{p}} = \delta\mathbf{v} \tag{56b}$$

$$\delta\dot{\mathbf{v}} = \left(a_{T_d}\mathbf{R}_d\mathbf{e}_1 + \frac{1}{m}\mathbf{R}_d\mathbf{f}_{a_d}\right) - \left(a_T\mathbf{R}\mathbf{e}_1 + \frac{1}{m}\mathbf{R}\mathbf{f}_a\right) \tag{56c}$$

$$\delta\dot{\boldsymbol{\theta}} = \mathbf{A}^T(\delta\boldsymbol{\theta})\left(-\mathbf{R}_d^T\mathbf{R}\boldsymbol{\omega} + \boldsymbol{\omega}_d\right) \tag{56d}$$

where $\mathbf{f}_{a_d}$ and $\mathbf{f}_a$ are the aerodynamic forces in terms of the desired and actual state, respectively

$$\mathbf{f}_{a_d} = \mathbf{f}_a\left(\mathbf{v}_{a_d}^{\mathcal{B}}\right), \quad \mathbf{v}_{a_d}^{\mathcal{B}} = \mathbf{R}_d^T\left(\mathbf{v}_d - \bar{\mathbf{w}}\right) \tag{57a}$$

$$\mathbf{f}_a = \mathbf{f}_a\left(\mathbf{v}_a^{\mathcal{B}}\right), \quad \mathbf{v}_a^{\mathcal{B}} = \mathbf{R}\left(\mathbf{v} - \mathbf{w}\right) \tag{57b}$$

$\mathbf{A}(\cdot)$ denotes the Jacobian of the exponential coordinates of $SO(3)$ (Bullo and Murray 1995):

$$\mathbf{A}(\delta\boldsymbol{\theta}) = \mathbf{I}_3 + \left(\frac{1-\cos\|\delta\boldsymbol{\theta}\|}{\|\delta\boldsymbol{\theta}\|}\right)\frac{\lfloor\delta\boldsymbol{\theta}\rfloor}{\|\delta\boldsymbol{\theta}\|} + \left(1 - \frac{\sin\|\delta\boldsymbol{\theta}\|}{\|\delta\boldsymbol{\theta}\|}\right)\frac{\lfloor\delta\boldsymbol{\theta}\rfloor^2}{\|\delta\boldsymbol{\theta}\|^2} \tag{58}$$

Proof. The proof is given in Appendix H.

Lemma 1. The first-order linearization of the error-state dynamics given in (56) is:

$$\delta\dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}}\delta\mathbf{x} + \mathbf{F}_{\mathbf{u}}\delta\mathbf{u} + \mathbf{F}_{\mathbf{w}}\delta\mathbf{w} \tag{59}$$

where

$$\mathbf{F}_{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathbf{v}} & \mathbf{M}_{\mathbf{R}} \\ \mathbf{0} & \mathbf{0} & -\lfloor\boldsymbol{\omega}_d\rfloor \end{bmatrix}, \mathbf{F}_{\mathbf{u}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{M}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}, \mathbf{F}_{\mathbf{w}} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}_{\mathbf{v}} \\ \mathbf{0} \end{bmatrix} \tag{60a}$$

$$\mathbf{M}_T = \mathbf{R}_d\mathbf{e}_1 \tag{60b}$$

$$\mathbf{M}_{\mathbf{v}} = \frac{1}{m}\mathbf{R}_d\frac{\partial\mathbf{f}_{a_d}}{\partial\mathbf{v}_{a_d}^{\mathcal{B}}}\mathbf{R}_d^T \tag{60c}$$

$$\mathbf{M}_{\mathbf{R}} = \mathbf{R}_d\left(-a_{T_d}\lfloor\mathbf{e}_1\rfloor - \lfloor\frac{\mathbf{f}_{a_d}}{m}\rfloor + \frac{\partial\mathbf{f}_{a_d}}{\partial\mathbf{v}_{a_d}^{\mathcal{B}}}\lfloor\frac{\mathbf{v}_{a_d}^{\mathcal{B}}}{m}\rfloor\right) \tag{60d}$$

$$\frac{\partial\mathbf{f}_{a_d}}{\partial\mathbf{v}_{a_d}^{\mathcal{B}}} = \left.\frac{\partial\mathbf{f}_a}{\partial\mathbf{v}_a^{\mathcal{B}}}\right|_{\mathbf{v}_{a_d}^{\mathcal{B}}} \quad \text{(see Equation (26))} \tag{60e}$$

$$\delta\mathbf{w} = \mathbf{w} - \bar{\mathbf{w}} \tag{60f}$$

Proof. The proof is given in Appendix I.

Remark 4. The error system in (59) is valid for any desired state-input trajectory $(\mathbf{x}_d, \mathbf{u}_d)$. This is because the system matrix $\mathbf{F}_{\mathbf{x}}$, input matrix $\mathbf{F}_{\mathbf{u}}$ and $\mathbf{F}_{\mathbf{w}}$ can always be calculated properly without encountering any singularities at any desired state $\mathbf{R}_d$ and $\mathbf{v}_d$. For the calculation of $\frac{\partial\mathbf{f}_{a_d}}{\partial\mathbf{v}_{a_d}^{\mathcal{B}}}$, as shown in (26), it involves the calculation of $\frac{\partial\mathbf{c}}{\partial\alpha}$ and $\frac{\partial\mathbf{c}}{\partial\beta}$, which are invalid when $\mathbf{v}_{a_d} = \mathbf{0}$. Fortunately, regardless of the values of $\frac{\partial\mathbf{c}}{\partial\alpha}$ and $\frac{\partial\mathbf{c}}{\partial\beta}$, we always have $\lim_{\mathbf{v}_a\to\mathbf{0}}\frac{\partial\mathbf{f}_{a_d}}{\partial\mathbf{v}_{a_d}^{\mathcal{B}}} = \mathbf{0}$ according to (26). Consequently, the linearized error-state system in (59) has no singularities within the entire flight envelope.

7.2 On-manifold MPC for trajectory tracking

With the error-state dynamics (59), which is a standard linear time varying system, a MPC that minimizes the state $\delta\mathbf{x}$ and $\delta\mathbf{u}$ is utilized for trajectory tracking. Setting the unknown disturbance $\delta\mathbf{w}$ in (59) to zero, the MPC is an optimization problem as follows:

$$\begin{aligned} \delta\mathbf{u}^* = \arg\min_{\delta\mathbf{u}_k} & \sum_{k=0}^{N-1}\left(\|\delta\mathbf{x}_k\|_{\mathbf{Q}_k}^2 + \|\delta\mathbf{u}_k\|_{\mathbf{R}_k}^2\right) + \|\delta\mathbf{x}_N\|_{\mathbf{P}_N}^2 \\ \text{s.t.} \quad & \delta\mathbf{x}_{k+1} = \left(\mathbf{I}_9 + \Delta t\mathbf{F}_{\mathbf{x}_k}\right)\delta\mathbf{x}_k + \Delta t\mathbf{F}_{\mathbf{u}_k}\delta\mathbf{u}_k \\ & \delta\mathbf{x}_0 = \delta\mathbf{x}_{\text{init}} \\ & \delta\mathbf{u}_k \in \delta\mathbb{U}_k, \quad k = 0, \cdots, N-1 \end{aligned} \tag{61}$$

where N is the predictive horizon, and $\mathbf{Q}_k, \mathbf{R}_k, \mathbf{P}_N$ are positive-definite diagonal matrices, denoting the penalty of the stage state, stage input and terminal state, respectively. $\mathbb{U}_k = \{\delta\mathbf{u} \in \mathbb{R}^m | \mathbf{u}_{\min} - \mathbf{u}_k^d \leqslant \delta\mathbf{u} \leqslant \mathbf{u}_{\max} - \mathbf{u}_k^d\}$ is the constraints for the input error that is derived from the actual input constraints $\mathbf{u}_{\min} \leqslant \mathbf{u}_k \leqslant \mathbf{u}_{\max}$. The optimization in (61) is a standard quadratic programming (QP) problem, which can be solved efficiently by existing QP solvers. Finally, the optimal control command at the current step is

$$\mathbf{u}_{\text{cmd}} = \mathbf{u}_{d_0} + \delta\mathbf{u}_0^* \tag{62}$$

Remark 5. The MPC in (61) is minimally parameterized and singularity-free. The minimal parameterization results

from the use of error state $\delta\mathbf{x}$ in the controlled system (59), which parameterizes the original state $\mathbf{x}$ on the state manifold $\mathcal{M}$ in its homeomorphic space. This space, being a normal Euclidean space, has the same dimension (i.e., 9) as the state manifold $\mathcal{M}$. The resultant MPC formulation (61) does not have any redundant parameters when compared with existing quaternion-based MPC for UAV control (Falanga et al. 2018; Sun et al. 2022). The singularity-free property of the MPC is two-folds. First, the MPC is not singular to the flight trajectory because the error system (59) is always valid in the entire flight envelope. Second, the MPC is not singular to the parameterization $\delta\mathbf{x}$. Common minimal parameterization of manifolds, such as Euler angles (Kamel et al. 2017; Nguyen et al. 2021), parameterizes the manifold with respect to a fixed point on the manifold, the resultant parameterization is singular at certain configurations. In contrast, our error state $\delta\mathbf{x}$ parameterizes the state manifold with respect to each point on the reference trajectory (as opposed to a fixed point). If the feedback MPC controller is stable (as it always needs to be), the error state is stabilized around zero and hence avoids the singularity effectively. The minimally-parameterized, yet singularity-free nature of our MPC, avoids any switching in parameterization or control scheme, and eventually leads to a global trajectory tracking controller.

Remark 6. The MPC in (61) is a model-based controller, where the computation of $\mathbf{F_x}$ and $\mathbf{F_u}$ requires the knowledge of the aerodynamic model $\mathbf{f}_a$ and its derivative (see (60)). This enables the MPC to exploit a high-fidelity aerodynamic model of the vehicle to achieve high-accuracy tracking control while effectively admits practical constraints, such as the input saturation.

Remark 7. The error-state dynamics formally derived in Section. 7.1.2 are globally equivalent to the original system. This equivalence allows to treat the tail-sitter as a formal nonlinear system. The system is further linearized along the reference state-input trajectory at each point, leading to a linear time-varying system in (59). The consequent MPC design is standard, and its convergence analysis can be studied using established techniques in existing literature like (Mayne et al. 2000), and hence will not be further discussed in the rest of paper.

8 Real-world experimental results

In this section, we validate the key ideas of the approach presented in this paper via real-world experiments on a quadrotor tail-sitter UAV. The algorithms of trajectory generation, flatness transform and global tracking controller are implemented to enable the vehicle to perform aggressive agile flights. Extensive challenging indoor and outdoor field tests are demonstrated, including agile $SE(3)$ flight through consecutive narrow windows, typical tail-sitter maneuvers (transition, level flight and loiter), and extremely aggressive aerobatics (Wingover, Loop, Vertical Eight, Cuban Eight, and their combo). All experiments are successfully tested at least three times for initial verification, data collection, and video record.

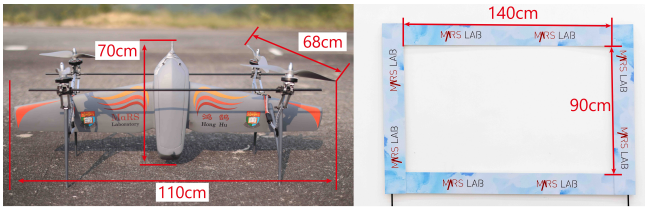
Fig. 7. Our quadrotor tail-sitter UAV prototype: Hong Hu.

8.1 Tail-sitter UAV platform

We validate the presented algorithms on a quadrotor tail-sitter prototype, named "Hong Hu", based on our previous airframe design (Gu et al. 2018). As shown in Fig. 7, Hong Hu is manufactured out of carbon fiber, weighs $2.4\,\mathrm{kg}$, and has a wingspan of $90\,\mathrm{cm}$. The cruise airspeed is $18\,\mathrm{m/s}$. It is powered by four T-MOTOR[1] MN5006 KV450 motors and APC[2] 13×10 propellers, achieving a hovering throttle at 43% of the full throttle. The tail-sitter UAV is equipped with an onboard computer DJI Manifold 2-C[3](1.8 GHz quad-core Intel i7 CPU) and an autopilot PX4 Mini[4] with a global positioning system (GPS) receiver module. A uni-axial airspeed sensor is mounted on the nose of the airframe. An action camera DJI Action 2[5] is fixed on a carbon rod for first-person-view (FPV) video capturing.

The presented algorithms of trajectory generation and high-level tracking control (i.e., MPC) are implemented on the onboard computer, and communicated via the Robot Operating System (ROS). An open-source QP solver OOQP (Gertz and Wright 2003) is deployed to solve the MPC problem in (61) at $100\,\mathrm{Hz}$. The predictive horizon is set to 12 in all experiments and the MPC takes 0.85ms in average to compute the optimal commands of thrust acceleration a_T and angular velocity $\boldsymbol{\omega}$, which are then sent to the autopilot PX4 Mini via MAVROS[6]. In the autopilot, the thrust acceleration command a_T is mapped to the throttle command by $\mathrm{thr} = k_T a_T$, where the coefficient k_T is computed as $\mathrm{thr_h}/9.81$ with $\mathrm{thr_h}$ being the throttle at hovering. The angular velocity command $\boldsymbol{\omega}$ is tracked by three PID controllers, each compares the respective angular velocity command with its onboard IMU measurements and calculate a normalized control torque $\boldsymbol{\tau}$ at $400\,\mathrm{Hz}$. The three PID controllers, one for each channel, are decoupled, where the coupled Coriolis term $\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}$ and aerodynamic moment $\mathbf{M}_a$ are all viewed as unknown disturbances and hence ignored in the controller. In the experiments, we found the vehicle exhibited a severe vibration caused by the propeller rotation and attenuate this vibration by a Notch filter added on each PEED controller (Xu et al. 2019a). The throttle and normalized torque are finally mixed into the four motor pulse-width

[1] https://uav-en.tmotor.com/
[2] https://www.apcprop.com/
[3] https://www.dji.com/manifold-2/specs
[4] https://px4.io/
[5] https://www.dji.com/dji-action-2
[6] http://wiki.ros.org/mavros



(a) (b)

Fig. 8. Sizes of the quadrotor tail-sitter UAV (a) and the narrow window (b).

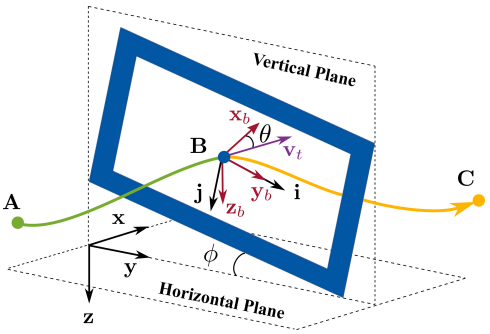


Fig. 9. The traverse trajectory is divided into two pieces: the first one (the green line) connects the start position A and the center of the window B, and the second one (the yellow line) connects B and the target position C. The specified traverse velocity $\mathbf{v}_t$ is perpendicular to the window plane. The body axis $\mathbf{y}_b$ is along the long side of the window, and the angle between $\mathbf{x}_b$ and $\mathbf{v}_t$ is specified as $\theta = 30^\circ$.

modulation (PWM) commands using the standard quadrotor configuration. The vehicle state is estimated by an extended Kalman Filter (EKF) also running on the autopilot. External position and heading measurements are obtained by a motion capture system for indoor experiments or the GPS module with magnetometer for outdoor experiments.

The aerodynamic model is identified by wind tunnel tests in our previous work (Lyu et al. 2018a) and refined by real flight tests due to the new propulsion system and manufacturing. For model refining, we conduct a series of normal and inverted level flight tests in different speeds (and angle of attack), and collect the flight data of motor PWM, vehicle velocity and attitude. To ensure the sideslip angle is zero during the flights, we measure the wind speed using an anemometer, and manually set the vehicle heading along the wind direction prior to each level or inverted flights. We calculate the rotor speed from the motor PWM, the incoming airflow consisting of the measured wind speed and the vehicle's inertial speed, and then obtain the total thrust according to the open-source APC propeller model[7]. Excluding the propeller thrusts leads to the lift and drag forces exerted on the vehicle and hence the values of C_L and C_D at different angle of attack α. We conduct the flight tests from low speed to high speed and iteratively refine the aerodynamic model, to achieve stable flights. For the side force coefficient C_Y, we use the model of (Lyu et al. 2018a) without any modification.

In all outdoor experiments without otherwise specified, the wind speed is estimated and compensated in the differential flatness transform (by setting the surrogate wind $\bar{\mathbf{w}}$). Referring to Johansen et al. (2015), only the wind speed components in the world frame X-Y plane is estimated by an EKF that propagates a constant wind speed model based on the airspeed sensor measurement and the vehicle inertial velocity and attitude. To avoid unstable wind speed estimation due to degraded airspeed measurements at low flight speeds, the estimated wind speed is compensated in the differential flatness transform only when the airspeed magnitude $||\mathbf{v}_a|| > 5\,\mathrm{m/s}$. When the vehicle speed is below this threshold or in all indoor experiments, no wind speed is compensated in the differential flatness transform (i.e., setting $\bar{\mathbf{w}} = \mathbf{0}$). In all results that follow, the angle of attack and side slip angle are computed based on $\bar{\mathbf{w}}$ used in the flatness transform, regardless of the actually estimated wind speed.

8.2 SE(3) flight through narrow windows

Flying through narrow windows is a challenging but potentially worthwhile scenario that a UAV can navigate in obstacle-dense environments, such as searching through thick forest or collapsed buildings after disasters. The main challenge of the problem is that the vehicle can fly through the narrow window only when its body is aligned with the window orientation to fit the limited traversing space as shown in Fig. 8. This task requires the UAV to execute a precise, aggressive full body motion on $SE(3)$ (i.e., $SE(3)$ flight). For the sake of flight agility and tracking accuracy, dynamical feasibility of the trajectory should be guaranteed rigorously in planning, such that the tracking error can be reduced when the vehicle executes the maneuver.

To generate a collision-free and dynamically feasible trajectory through a narrow window, we divide the trajectory into two pieces (i.e., before and after passing through the window), and optimize them by (48) separately. As shown in Fig. 9, the first trajectory (the green line) connects the UAV start position to a traversing position fixed at the center of the window, and the second trajectory (the yellow line) connects the traversing position to the target position. To determine the boundary conditions for these two trajectories, the speed, acceleration, and jerk at the start and target positions are all set to zeros (i.e., stationary hovering). For the traversing position, the position is the center of the window, velocity $\mathbf{v}_t$ is normal to the window plane with magnitude manually specified. To determine the acceleration at the traversing position, we specify the body Y axis to be along the window long edge and set the body X axis to form a $\theta = 30^\circ$ angle with the traversing velocity $\mathbf{v}_t$ (i.e., AoA is $\alpha = 30^\circ$ at the traversing position). Then, we choose the thrust a_T by minimizing the total acceleration $\mathbf{a}_t$ at the traversing position:

$$\begin{aligned}\min_{a_T} \|\mathbf{a}_t\| &= \min_{a_T} \|\mathbf{R}(a_T\mathbf{e}_1 + \mathbf{R}^T\mathbf{g} + \frac{1}{m}\mathbf{f}_a)\| \\ &= \min_{a_T} \|a_T\mathbf{e}_1 + \mathbf{R}^T\mathbf{g} + \frac{1}{m}\mathbf{f}_a\| \\ &= \min_{a_T} \|a_T + \mathbf{e}_1^T(\mathbf{R}^T\mathbf{g} + \frac{1}{m}\mathbf{f}_a)\|\end{aligned} \tag{63}$$

[7] https://www.apcprop.com/technical-information/performance-data/

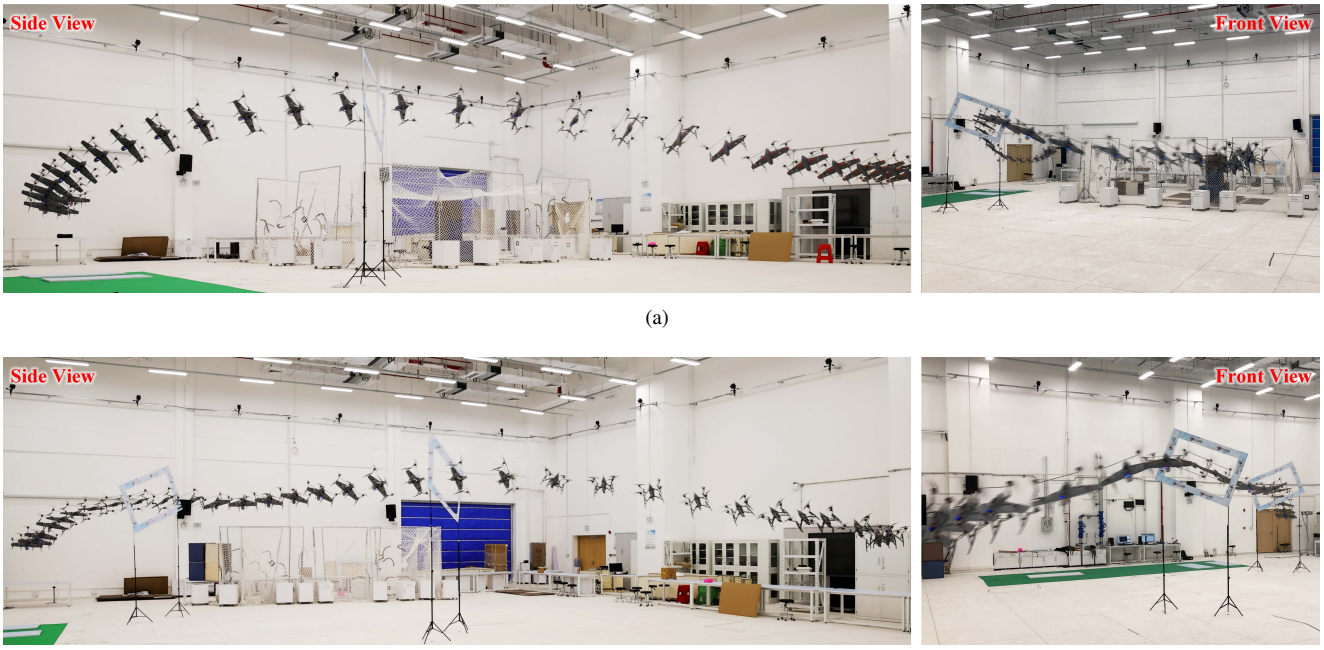


Fig. 10. Snapshot sequences of agile tail-sitter flight through narrow windows. (a) Flying through a single window in $10\,\mathrm{m/s}$. (b) Flying through two consecutive windows in $8\,\mathrm{m/s}$.

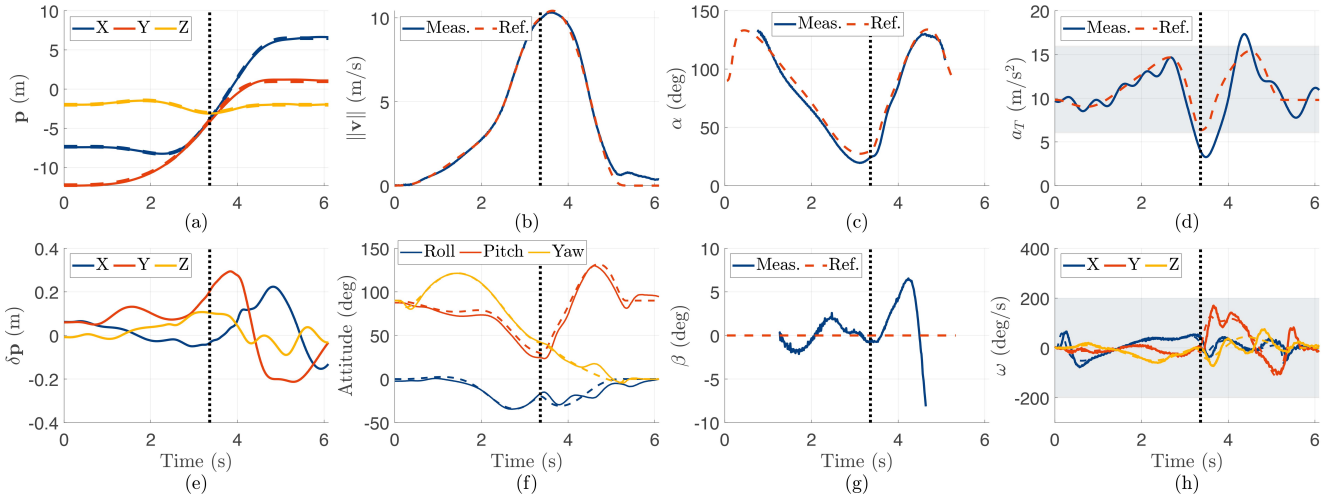


Fig. 11. Flight data of the SE(3) flight through a single window with roll 20° and traversing speed $10\,\mathrm{m/s}$: (a) position, (b) flight speed, (c) angle of attack, (d) thrust acceleration, (e) position tracking errors, (f) attitude in Euler angles, (g) sideslip angle, (h) angular velocity. In all subplots where applicable, the solid and dashed lines denote the measurement and reference, respectively. For the thrust acceleration, the measurement is obtained from the accelerometer X axis. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $1\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. The vertical dotted lines denote the moment the vehicle passes the window, and the shaded areas in (d) and (h) denote the feasible region of the actuation in trajectory optimization (48).

Taking the thrust constraint into consideration, (63) leads to a constrained linear optimization:

$$\begin{aligned} a_T^* &= \arg\min_{a_T} \|a_T + \mathbf{e}_1^T(\mathbf{R}^T\mathbf{g} + \frac{1}{m}\mathbf{f}_a)\| \\ &\text{s.t.} \quad a_{T_{\min}} \leqslant a_T \leqslant a_{T_{\max}} \end{aligned} \tag{64}$$

where $a_{T_{\min}}$ and $a_{T_{\max}}$ are the boundaries of thrust acceleration. Then, the traversing acceleration can be obtained by substituting the optimal thrust acceleration and the determined attitude into the translational dynamics in (1b). Finally, the traversing jerk is set to zeros for simplicity.

We validate the algorithms in real-world experiments as shown in Fig. 10. The kinodynamic and control input constraints of the planner are $v_{\max} = 12\,\mathrm{m/s}$, $a_{T_{\min}} = 6\,\mathrm{m/s^2}$, $a_{T_{\max}} = 16\,\mathrm{m/s^2}$, and $\omega_{\max} = 200\,\mathrm{deg/s}$. To increase the tracking accuracy for position and attitude, which is crucial for the UAV to pass the window, parameters of the MPC are set as $\mathbf{Q}_k = \mathrm{diag}([1800, 1800, 1800, 5, 5, 5, 50, 50, 50])$, $\mathbf{R}_k = \mathrm{diag}([0.3, 0.4\ 0.4, 0.4])$, and $\mathbf{P}_N = \mathbf{Q}_k$. All poses of the windows and UAV are measured by a motion capture system. The flying volume is about $15 \times 15 \times 4\,\mathrm{m}^3$.

In the first scenario, the tail-sitter performs aggressive $SE(3)$ flights to fly through a single window. Fig. 10(a)

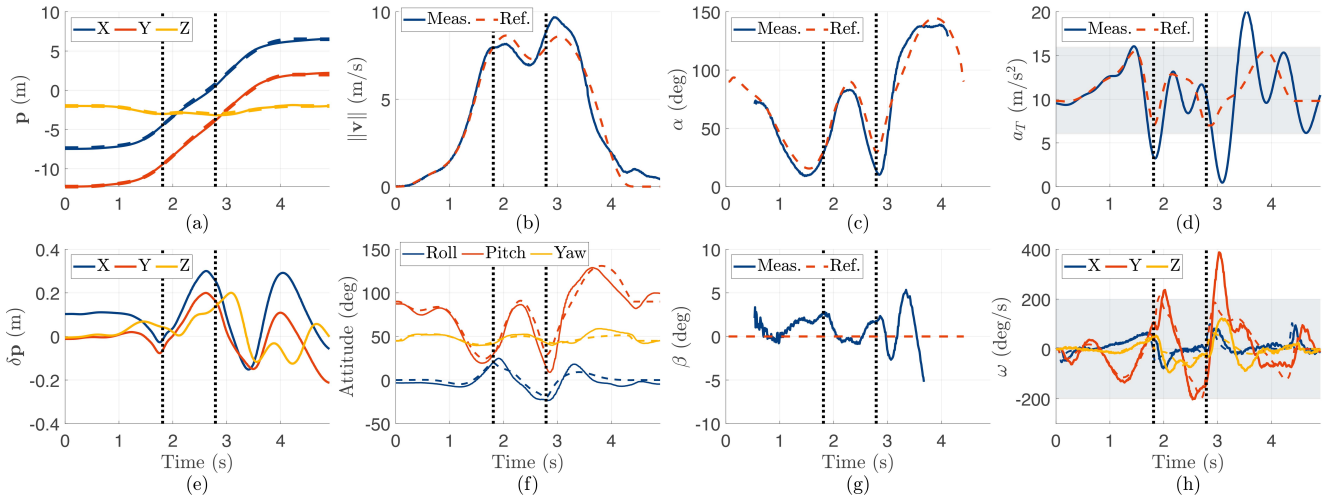


Fig. 12. Flight data of the SE(3) flight through two consecutive windows with roll angles 20° and -20° and traversing speeds both at $8\,\mathrm{m/s}$: (a) position, (b) flight speed, (c) angle of attack, (d) thrust acceleration, (e) position tracking errors, (f) attitude in Euler angles, (g) sideslip angle, (h) angular velocity. In all subplots where applicable, the solid and dashed lines denote the measurement and reference, respectively. For the thrust acceleration, the measurement is obtained from the accelerometer X axis. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $1\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. The vertical dotted lines denote the moments when the vehicle passes the windows, and the shaded areas in (d) and (h) denote the feasible region of the actuation in trajectory optimization (48).

and Fig. 11 respectively show the snapshot sequence and experimental data of a successful flight passing through a window with roll angle $\phi = 20^\circ$ and in a traversing speed of $\|\mathbf{v}_t\| = 10\,\mathrm{m/s}$. As can be seen in Fig. 11 (a-b), to fly through the window with the specified speed, the UAV must accelerate from stationary hovering to the traversing speed (i.e., $10\,\mathrm{m/s}$) in a time less than 3.4 seconds and a space within $3.6 \times 8.1 \times 1\,\mathrm{m}^3$. To achieve this, the UAV performs transition and a banked turn simultaneously (see Fig. 11 (f)). In fact, the planner and controller are not even aware of the transition, but treats the entire flights uniformly. Then the UAV traverses the window with the required pose and velocity at $3.36\,\mathrm{s}$ (the vertical black dotted line) and finally recover to the hovering status again within a very limited flight space. During the flight, the angle of attack varies up to 113° in merely two seconds (see Fig. 11 (c)), indicating a large envelope of angle of attack. Despite this, the overall position error as shown in Fig. 11 (e) is less than $0.3\,\mathrm{m}$ and the slideslipe angle as shown in Fig. 11 (g) is well stabilized around zero. The seemly large sideslip angle at the beginning and end of the flight is due to the unstable airspeed measurements at very low speeds. Fig. 11 (d) and (h) show that the trajectory planner effectively bounds the thrust acceleration and angular velocity of the reference trajectory within the nominal actuator constraints (the shaded area).

In the second scenario, the tail-sitter performs more aggressive $SE(3)$ flights to fly through two consecutive windows. Fig. 10(b) and Fig. 12 respectively show the snapshot sequence and experimental data of a successful flight with window roll angles -20° and 20° and traversing speeds both at $8\,\mathrm{m/s}$. As shown in Fig. 10(b) and Fig. 12 (a), (b) and (f), the UAV traverses the first window at $1.81\,\mathrm{s}$, then immediately pulls up the pitch angle, which slows down the speed, to gain sufficient lift maintaining the height. After this, the UAV pitches down and accelerates again to fly through the second window safely at $2.79\,\mathrm{s}$. The

Tab. 2. Average Pose tracking error of $SE(3)$ flights.

Roll Angle ϕ	$\|\mathbf{v}_t\|$ (m/s)	$\delta\mathbf{p}_{\mathrm{RMS}}$(cm)	$\delta\boldsymbol{\theta}_{\mathrm{RMS}}(^\circ)$
0°	8	10.8	4.7
20°	8	13.5	6.4
40°	8	9.5	4.0
20°	3	10.7	4.5
20°	5	12.8	5.2
20°	10	9.5	4.0
0° & 20°	8	11.6	6.2
20° & -20°	8	10.0	5.5
40° & 20°	8	12.0	6.6

fact that the maneuver in this scenario is more aggressive than the former, is also shown in Fig. 12 (d) and (h) where the IMU measurements of thrust acceleration and angular velocity reach $20\,\mathrm{m/s^2}$ and $400\,\mathrm{deg/s}$, respectively. The position tracking error in Fig. 12 (e) is consequently larger, but the overall position error remains less than $0.3\,\mathrm{m}$. Other phenomenons, such as the large envelope of angle of attack, simultaneous bank turn and transition, and stabilization of the sideslip angles, are all similar to the previous experiment.

To provide more convincing results, we conduct two test groups of experiments demonstrating the flights through single and double windows, respectively. The first group consists of six different flight tests with a window roll angle $\phi \in \{0^\circ, 20^\circ, 40^\circ\}$ and a traversing speed $\|\mathbf{v}\| \in \{3, 5, 8, 10\}$ m/s. The second group consists of three $SE(3)$ flights with window angles combinations drawn from $\{-20^\circ, 0^\circ, 20^\circ, 40^\circ\}$ and a traversing speed of $8\,\mathrm{m/s}$. All nine experiments are successfully conducted with results summarized in Tab. 2. The first group results demonstrate a sufficiently high control accuracy to avoid collision (i.e., the maximum average position and attitude error are 13.5

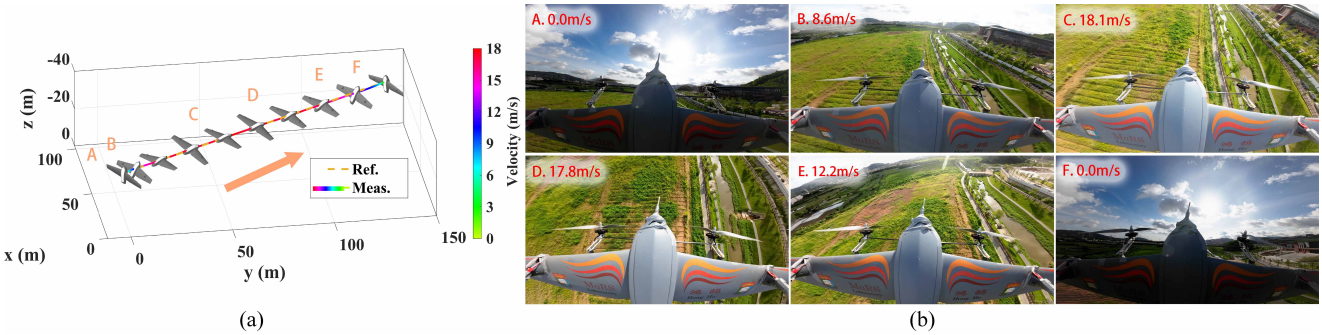


Fig. 13. Forward flight of the straight-line path in $18\,\mathrm{m/s}$: (a) trajectory illustration, (b) images from the FPV camera. Labels A-F denote different flight phases of the vehicle: A. hovering, B. forward transition, C. and D. level flight, E. backward transition, F. hovering.

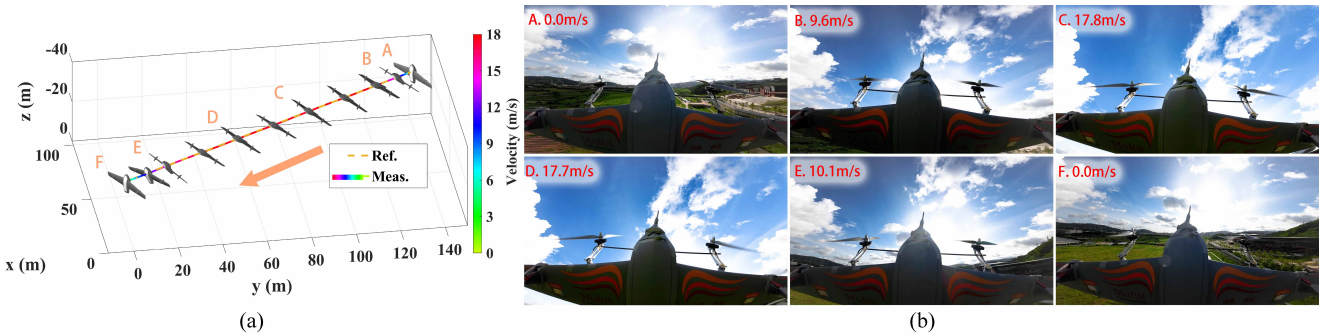


Fig. 14. Inverted flight of the straight-line path in $18\,\mathrm{m/s}$: (a) trajectory illustration, (b) images from the FPV camera. Labels A-F denote different flight phases of the vehicle: A. hovering, B. inverted forward transition, C. and D. inverted level flight, E. interted backward transition, F. hovering.

cm and 6.4°, respectively). It also shows that the proposed trajectory generation in coordinated flight is applicable to low-speed $SE(3)$ flights (down to $3\,\mathrm{m/s}$). For the second group results, the pose tracking error slightly increases at the second window due to the dramatic attitude and velocity variations as mentioned before, but still small enough for the UAV to pass through the window. To sum up, the varioius agile flights through narrow windows demonstrate that the proposed trajectory generation and control framework is capable to execute accurate $SE(3)$ flights, which shows a promising application to aggressive autonomous flight with obstacle avoidance in cluttered environments. Readers are encouraged to watch the accompanying videos for better visualization of the experiments.

8.3 Typical maneuvers in field environments

In this task, we examine the effectiveness and performance of the proposed algorithms for typical maneuvers in field environments. We test a straight-line maneuver (including hovering, transition and level flight) and loiter flights with speed ranging from $5\,\mathrm{m/s}$ to $20\,\mathrm{m/s}$, and make comparisons to conventional tail-sitter controllers (with details supplied later). We reserve the same parameters of the planner and controller, except decreasing the MPC position penalty (i.e., the first three diagonal elements of $\mathbf{Q}_k$) to $[1200, 1200, 1200]$ to increase the robustness to uncertainties like unmeasured wind disturbance, and noisy GPS measurement in outdoor environments.

8.3.1 Straight-line flight

Transition and level flights are two crucial maneuvers for tail-sitter UAVs and are commonly tested for tail-sitter controllers. We demonstrate the proposed framework on these maneuvers via a forward flight trajectory, which involves three maneuvers: forward transition, level flight, and backward transition (see Fig. 13 (a), and an inverted flight trajectory, which involves another three maneuvers: inverted forward transition, inverted level flight, and inverted backward transition (see Fig. 14 (a). We present the tracking performance on these trajectories with different level-flight speed ranging from $5\,\mathrm{m/s}$ to $20\,\mathrm{m/s}$, and make comparisons against existing works in terms of transition accuracy.

We design the forward and inverted flight trajectories along the same straight-line path, where the vehicle first flies forward along the path to a target position and then flies in an inverted pose along the same path back to the origin, as shown in Fig. 15. Both forward and inverted flight trajectories have the same level-flight phase, which is manually specified as a constant-velocity trajectory (speed ranges from $5\,\mathrm{m/s}$ to $20\,\mathrm{m/s}$) lasting for 4-5 seconds. The trajectories from the initial hovering position to the constant-velocity trajectory and that from the constant-velocity trajectory to the target hovering position are designed by the proposed trajectory optimization method in (48), for both forward and inverted trajectories.

Fig. 13 and Fig. 14 show the 3-D trajectory and FPV images of the test with level-flight speed of $18\,\mathrm{m/s}$, and the corresponding flight data are detailed in Fig. 15. As shown in Fig. 15, the tail-sitter first performs a forward transition (phase I) from hovering to level flight with speed $18\,\mathrm{m/s}$ (phase II) while the pitch angle decreases from 90° to 13°. After flying $112\,\mathrm{m}$ over 7.7 seconds, the vehicle performs a backward transition (phase III) to hovering (phase IV). Subsequently, the vehicle performs an inverted forward transition (phase V), where the pitch angle increases from

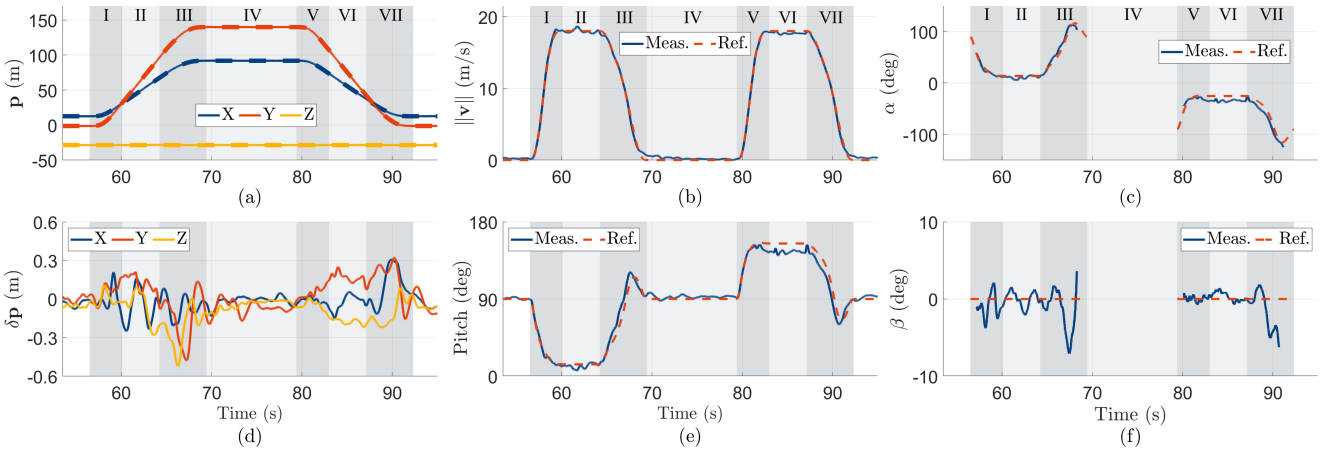


Fig. 15. Flight data of the straight-line flight test consisting of both forward and inverted flights: (a) position, (b) flight speed, (c) angle of attack, (d) position tracking errors, (e) pitch angle, (f) sideslip angle. Flight stages from I to VII divided by shaded areas indicate the I. forward transition, II. level flight, III. backward transition, IV. hovering, V. inverted forward transition, VI. inverted level flight and VII. inverted backward transition. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $2\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. In all subplots where applicable, the solid and dashed lines respectively denote the measurement and reference.

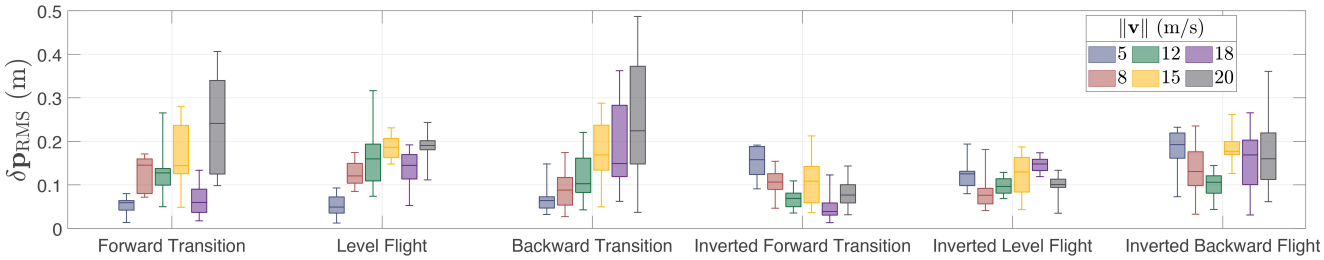


Fig. 16. The position tracking error in six different speeds when the tail-sitter flies in different phases of a straight-line path.

90° to 146°, reaching the inverted level flight with speed of $18\,\mathrm{m/s}$ (phase VI). Finally the vehicle performs an inverted backward transition (phase VII) to return to the initial hovering position. It is seen that the vehicle feedback trajectory of position, velocity and pitch angle tracks the reference state trajectory precisely throughout the flight. Fig. 15 (d) shows the position tracking error. As can be seen, the overall tracking error is $0.13\,\mathrm{m}$ in average and $0.52\,\mathrm{m}$ at most, which is incredibly small considering that the flight speed is up to $18\,\mathrm{m/s}$, and the angle of attack varies over 230° (see Fig. 15 (c)).

To provide a more convincing result and demonstrate the effectiveness of the proposed framework in full-envelope flight, we conduct a group of straight-line flights with six different level-flight speeds of $\|\mathbf{v}\| \in \{5, 8, 12, 15, 18, 20\}$ m/s. The position tracking error in each flight phase of each flight is statistically analyzed in Fig. 16. As can be seen, the errors at all times in all 36 groups of data across different flight speeds or phases are less than $0.5\,\mathrm{m}$, showing that the proposed framework enables a tail-sitter to fly within the whole envelope in high accuracy. Notably, the tracking error during inverted flight is as low as $0.2\,\mathrm{m}$. The increased tracking accuracy in the inverted flight are due to better fitting of the aerodynamic coefficients in negative AoA regions. It is also noted that existing methods based on separated trajectory planners and controllers Frank et al. (2007); Oosedo et al. (2017); Lyu et al. (2017b); Xu et al. (2019a) did not demonstrate such inverted flights, because the required AoA is out of the designed envelope.

Moreover, we make a comparison on the transition accuracy with a traditional linear transition controller (Lyu et al. 2017b), which is the same strategy used by the autopilot PX4. To ensure a fair comparison, we implement both our MPC and the traditional controller with the same low-level angular velocity controller, on the same vehicle. In addition, the linear transition controller are tuned to the best extent. The cascaded attitude and altitude PID controllers of the linear transition controller are tuned by Ziegler-Nichols method, while the linear Pitch reference is determined by a transition duration and angle span. The angle span indicates the Pitch change between hovering and level flight at cruise speed, and is obtained as $13^\circ - 90^\circ$ in the former straight-line flight experiment (see Fig 15(e)). The transition duration is initially set to the same transition time of our method but it failed the transition flight due to the too short transition time. Then, we gradually increase the transition duration until successful flight is achieved. We iteratively refine the above attitude and altitude controller, achieving comparable performance demonstrated in existing research (Oosedo et al. 2017; Lyu et al. 2017b; Xu et al. 2019a).

Because the linear transition controllers usually focus on pitch and altitude control, and have no position control in other directions, we focus on the comparison of longitudinal state variables only. Fig. 17 (1a)-(1c) shows the comparison in forward transition. Our method controls the pitch angle to decrease from 90° to 13° smoothly and speeds up from hovering to $18\,\mathrm{m/s}$ in merely 3 seconds with altitude error peaking at $0.11\,\mathrm{m}$, while the linear method has good

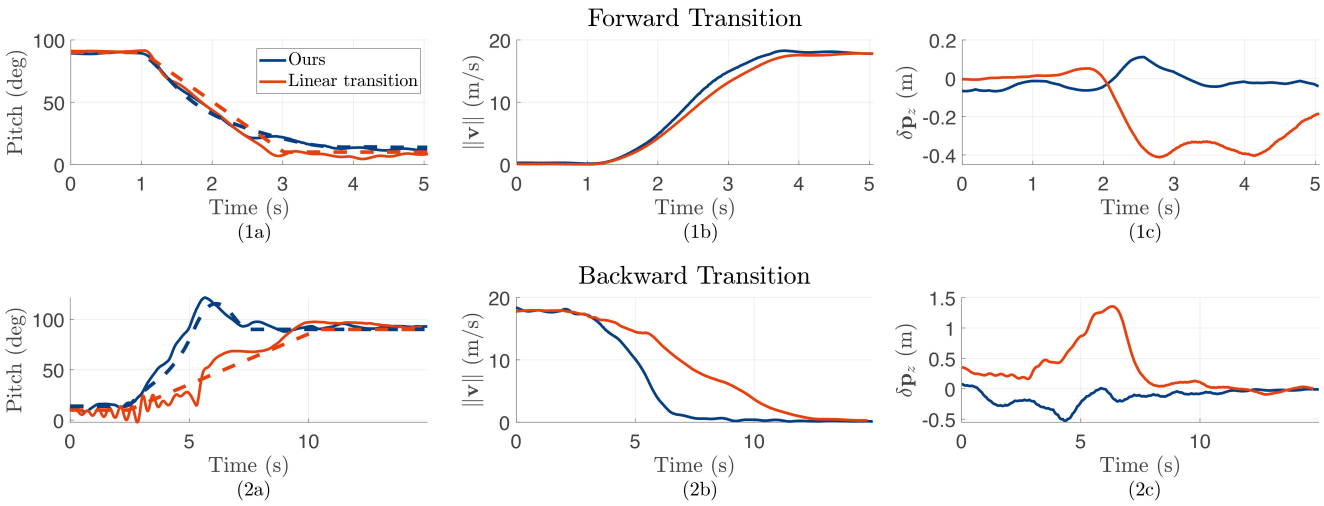


Fig. 17. Comparison on transition control performance between our presented approach and the linear transition method (Lyu et al. 2017b). (1a)-(1c) are respectively the pitch angle, flight speed and altitude tracking error for the forward transition, and (2a)-(2c) are those for backward transition. The solid and dashed lines in (1a) and (2a) respectively denote the measurement and reference.

performance in pitch control and accelerating but the altitude drops $0.41\,\mathrm{m}$. Similarly, in backward transition shown in Fig. 17 (2a)-(2c), our method tracks the reference pitch angle smoothly and has maximum altitude error of $0.52\,\mathrm{m}$ only, while the linear method tracks the linear pitch trajectory with significant pitch fluctuations and has large altitude deviation of $1.35\,\mathrm{m}$. It can be also noticed that our method pulls up the pitch angle over 120° and then returns to 90° for fast deceleration as shown in Fig. 17 (2b), the resultant backward transition is 4 seconds shorter than the linear method. In this comparison, our model-based framework shows its advantages in tracking accuracy and flight aggressiveness, which outperforms the model-free linear transition control.

8.3.2 *Loiter flight*

Loiter flight is another typical trajectory that validates the cruise performance of tail-sitters. As shown in Fig. 18, the trajectory consists of three phases: banked forward transition from hovering to loiter, loiter flight in constant speed, and banked backward transition from loiter to hovering. The loiter trajectory is designed in three steps. The constant-speed circular trajectory is first determined manually. Then the banked forward transition trajectory is optimized by (48) with initial condition as the hovering state and terminal condition as the first point on the circular trajectory. Similarly, the banked backward transition trajectory is optimized by (48) to perform a loiter-to-hovering maneuver.

Fig. 18 and Fig. 19 respectively show the trajectory and flight data in the loiter test with a flight radius of 50m and speed of $18\,\mathrm{m/s}$. As shown in Fig. 18 and Fig. 19 (b-c), after a while of stationary hovering, the tail-sitter first performs a coupled roll and pitch rotation to smoothly transition into the circular trajectory, and similarly transitions out of the circular trajectory with coupled roll and pitch rotations. Compared to traditional control methods (Verling et al. 2016; Lyu et al. 2017b) where a loiter trajectory is separated into straight-line transition followed by a bank turn in level flight, our maneuver is more elegant and time-saving due to less extra flight distance. It is seen in Fig. 19 that during the entire flight, the feedback of position, velocity and attitude tracks the reference closely. More specifically, Fig. 19 (e) illustrates the position tracking error, which is less than $0.26\,\mathrm{m}$ during the 45-second constant-speed loiter and slightly increases to $0.42\,\mathrm{m}$ and $0.56\,\mathrm{m}$ in the two transition phases, respectively. Moreover, we conduct this test with different loiter speed $\|\mathbf{v}\| \in \{8, 12, 18\}$ m/s. The tracking error statics of each phase of the three tests are summarized in Fig. 20. Banked transitions in the largest speed $18\,\mathrm{m/s}$ have the largest worst-case tracking errors (i.e., $0.31\,\mathrm{m}$ for the forward transition and $0.33\,\mathrm{m}$ for the backward transition), while all loiter flights have similarly small errors less than $0.21\,\mathrm{m}$. The above experimental results demonstrate that the proposed trajectory generation and tracking control framework promises high-accuracy flights in real outdoor environments.

In order to demonstrate the effectiveness and significance of wind speed compensation in the controller, we conduct a loiter fight in $18\,\mathrm{m/s}$ with wind speed in the flatness transform enabled and disabled online. As shown in Fig. 21 (f), the wind speed is estimated during the entire flight test, but the control framework only compensates the wind speed after 89s, indicated by the shaded background. When the wind speed is not compensated, the reference pitch angle (and angle of attack) maintains at a constant value 25° due to the constant loitering speed (see Fig. 21 (d, e)). In contrast, the actual vehicle pitch angle climbs to about 50° to increase the lift due to the smaller airspeed when following the wind, and drops to around 10° to decrease the lift due to the larger airspeed when against the wind. Moreover, due to the uncompensated wind speed, the vehicle actually does not perform coordinated flight, causing a side force that is then compensated by the vehicle roll angles (see Fig. 21 (a)). Furthermore, the uncompensated wind speed contributes to an extra disturbance as shown in (59), which causes the control error of the measured sideslip angle (which is computed without considering the estimated wind velocity and should be equal to the reference sideslip angle) to fluctuate between 12.5° and -7° (see Fig. 21 (b)). On the other hand, when the estimated wind speed is used in the differential flatness transform for the calculation of the state-input trajectory and the subsequent trajectory tracking controller, the reference pitch angle is recalculated to fluctuate according to the wind speed, similarly the

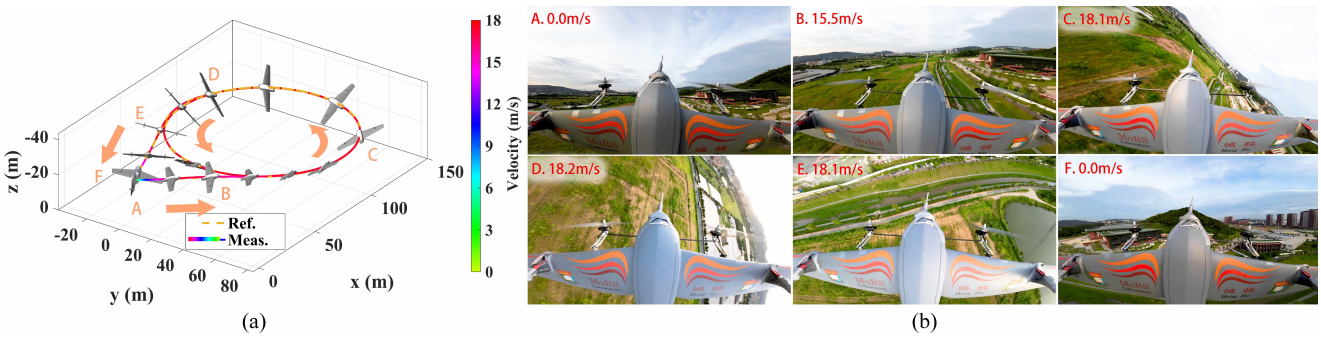


Fig. 18. Loiter flight with flight radius of 50m and velocity of $18\,\mathrm{m/s}$: (a) trajectory illustration, (b) images from the FPV camera. Labels A-F denote different states of the vehicle: A. hovering, B. banked forward transition, C and D. loiter flight, E. banked backward transition, F. hovering.

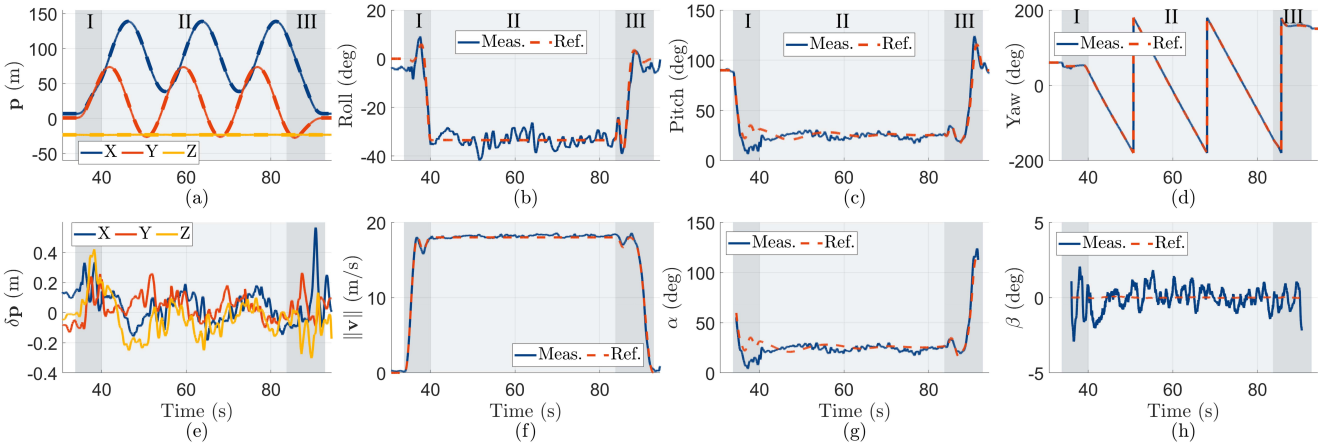


Fig. 19. Flight data of the loiter flight in $18\,\mathrm{m/s}$: (a) position, (b-d) attitude in Euler angles, (e) position tracking errors, (f) flight speed, (g) angle of attack, (h) sideslip angle. Flight stages from I to III divided by shaded areas indicate the banked forward transition, loiter, banked backward transition, respectively. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $2\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. In all subplots where applicable, the solid and dashed lines respectively denote the measurement and reference.

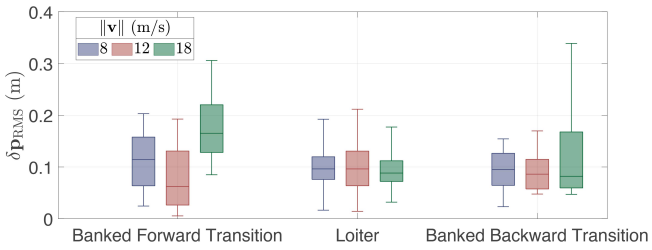


Fig. 20. Position tracking error in three different phases when the tail-sitter flies the loiter trajectory in different speeds.

reference yaw angle is also adjusted to keep the sideslip angle at zero (i.e., ensuring the coordinated flight condition). As a result, the control errors in pitch, slideslip angle, and flight speed are significantly reduced.

Finally, a comparison between our method and the total energy control system (TECS) is conducted on the loiter flight of $18\,\mathrm{m/s}$. As a mature technique for fixed-wing aircraft flight control, TECS also has been widely used in tail-sitter level flights. Due to the approximately linear aerodynamic force in low AoA, TECS employs a proportional and integral (PI) control scheme to regulate the airspeed and altitude by controlling the error of the total energy (i.e., the sum of potential and kinetic energy) to zero (Lambregts 1983). We use the TECS implemented in the PX4 autopilot and tune its parameters to the best extent. Similar to the previous transition control comparison, both the proposed MPC and TECS utilize the same low-level controller for tracking the angular velocity command. The inner attitude loop, middle energy balance loop, and outer total energy loop of the TECS, which compute the commands for angular velocity, pitch angle, and thrust respectively, are tuned in sequence using the Ziegler-Nichols method. The resulting control performance achieved in the experiment is on par with those demonstrated in related works (Lyu et al. 2017b; Gu et al. 2017). As shown in Fig. 22, the vehicle altitude drops around $1.5\,\mathrm{m}$ in average and $2.5\,\mathrm{m}$ in maximum when using TECS for the loiter flight. In comparison, there is no obvious steady-state error for our approach and the maximum altitude error is less than $0.25\,\mathrm{m}$. The results are reasonable since TECS does not make use of any aerodynamic models of the vehicle, while our approach fully exploits these information.

8.4 Aerobatics

In this task, we push the tail-sitter to its physical limits to perform extremely aggressive aerobatics, which further demonstrates the effectiveness and robustness of our proposed methods. Our approach is the first to enable an autonomous tail-sitter to perform a series of aerobatic

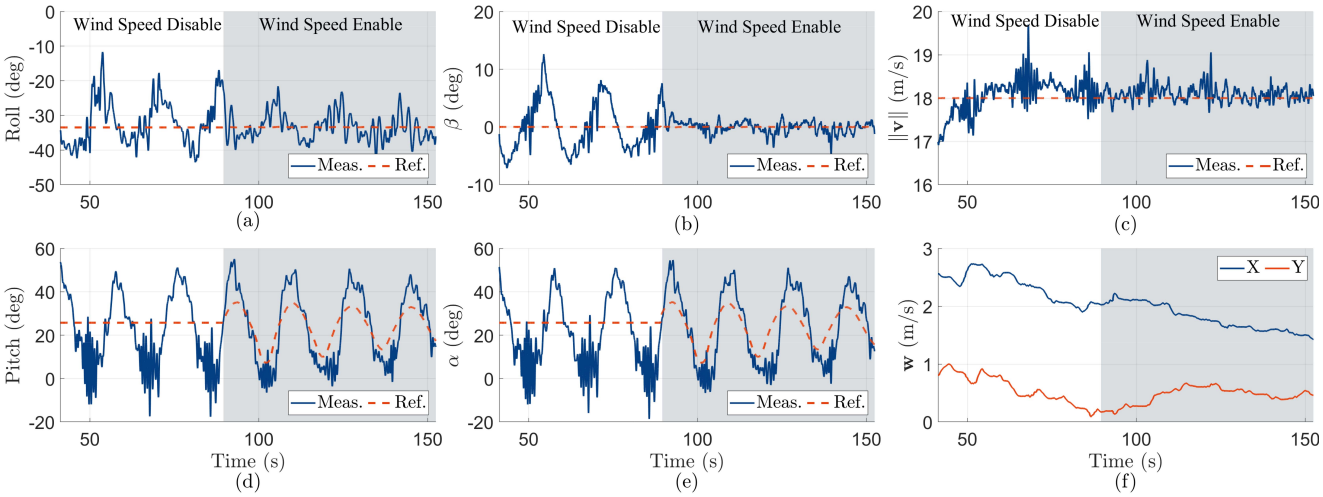


Fig. 21. Loiter flight test in $18\,\mathrm{m/s}$ with and without wind speed compensation in the controller: (a) roll angle, (b) sideslip angle, (c) flight speed, (d) pitch angle, (e) angle of attack, (f) estimated wind speed. In all subplots, the white area denotes the flight when setting $\bar{\mathbf{w}} = \mathbf{0}$ in the flatness function (Algorithm 1), while the shaded area denotes the duration when the online-estimated wind speed is used as the $\bar{\mathbf{w}}$ in the flatness transform. Note that both the measured and reference angle of attack and sideslip angle are computed based on $\bar{\mathbf{w}}$, regardless of the actually estimated wind speed.

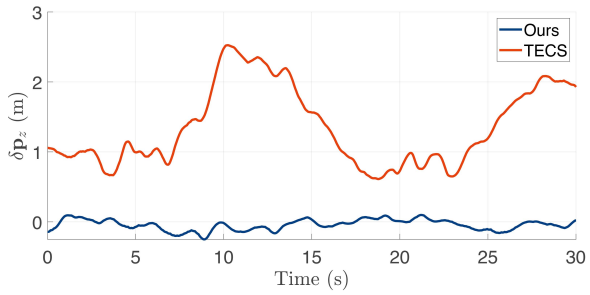


Fig. 22. Comparison on the altitude control performance between our presented approach and the total energy control system (TECS) in a $18\,\mathrm{m/s}$ loiter flight.

maneuvers with such agility in real outdoor environments. These maneuvers are highly challenging even for expert human pilots and are listed by increasing difficulty as follows:

1) Wingover: the vehicle makes a 180° turn in heading by executing a fast climb and turn, during which the wing swings over the top of the turn (i.e., the roll angle reaches 90°), as shown in Fig. 23.
2) Loop: the vehicle enters a vertical circle and makes a 360° flip in pitch angle, as shown in Fig. 24.
3) Vertical Eight: the vehicle performs a vertical figure-"8" trajectory with pitch angle pulled up and down over 180°, as shown in Fig. 25.
4) Cuban Eight: similarly to the Vertical Eight, the vehicle performs a "∞"-shape trajectory with pitch angle pulled up and down over 180°, as shown in Fig. 26.
5) Combo: the vehicle starts with Cuban Eight, followed by Wingover, Vertical Eight, Loop, and ends with another Wingover to fly back to the origin, as shown in Fig. 27. The entire maneuver is executed consecutively without any breaks.

As shown in Fig. 23-26, to specify the shape of the trajectory and the vehicle pose at certain position on the trajectory, we separate the entire trajectory by multiple pieces by boundary points (i.e., the black squares). At the boundary points, the full vehicle states (i.e., position, velocity, and attitude) are specified and transformed to trajectory boundary conditions $\mathbf{p}^{(0:3)}$. With these boundary conditions, trajectories within two consecutive boundary points are optimized by our trajectory optimization framework (48). To further specify the shape of each trajectory segment, we specify some waypoints (i.e., the black dots) that the trajectory must pass through, which is naturally supported by the optimization framework in (48). All the trajectories begins with a forward transition (i.e., the origin to the first black square) and ends with a backward transition to hovering (i.e., the last black square to the destination). Taking the Wingover in Fig. 23 (a) as example, the trajectory consists of four segments: forward transition, climbing up with 90° rotation in both roll and yaw, diving down with reverse heading, and backward transition. The design of the Loop trajectory in Fig. 24 (a) is similar, except that the top boundary point is designed to drive the vehicle upside down (i.e., -180° in pitch angle) and further inserting two waypoints to guarantee the shape of Loop. The Vertical Eight and Cuban Eight trajectories are generated by connecting two Loop trajectories. The boundary points of the connecting trajectories are obtained from the original Loop trajectories and a waypoint in the middle is used to serve the intersection point of the two connecting trajectories.

For all aerobatics above, we use the same parameters of the planner and controller as in the indoor $SE(3)$ flights and outdoor typical flights (i.e., Section 8.2 and 8.3, respectively), except further decreasing the MPC position penalty (i.e., the first three diagonal elements of $\mathbf{Q}_k$) to $[900, 900, 900]$ to increase the system robustness in consideration of the highly aggressive maneuvers being executed.

Fig. 28 details the flight data of the Wingover. The vehicle first transits from hovering to level flight with speed of $15\,\mathrm{m/s}$, then performs the Wingover maneuver in 14-$18\,\mathrm{m/s}$ and finally ends with a backward transition to hovering. The

Fig. 23. Wingover: (a) illustration of the trajectory that is divided in two segments by three boundary points (black squares). Each segment is optimized by (48), (b) images from the FPV camera. Labels A-F denote different flight phases of the vehicle.

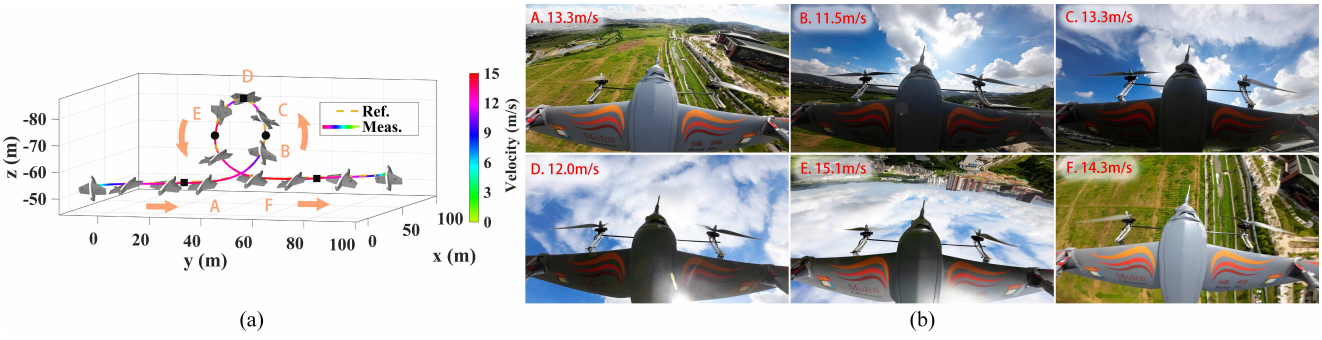


Fig. 24. Loop: (a) illustration of the trajectory, boundary points (black squares), and intermediate waypoints (black dots) that are constrained in (48), (b) images from the FPV camera. Labels A-F denote different flight phases of the vehicle.

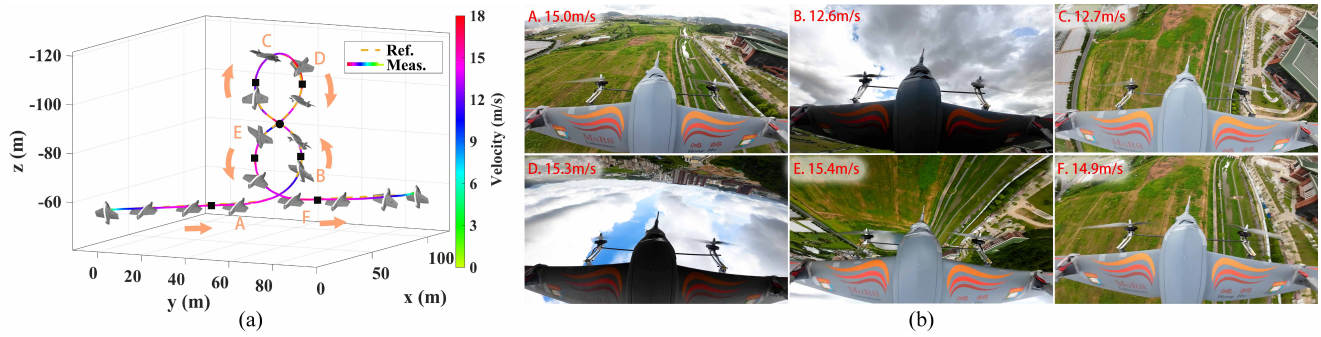


Fig. 25. Vertical Eight: (a) illustration of the trajectory, boundary points (black squares), and intermediate waypoints (black dots), (b) images from the FPV camera. Labels A-F denote different flight phases of the vehicle.

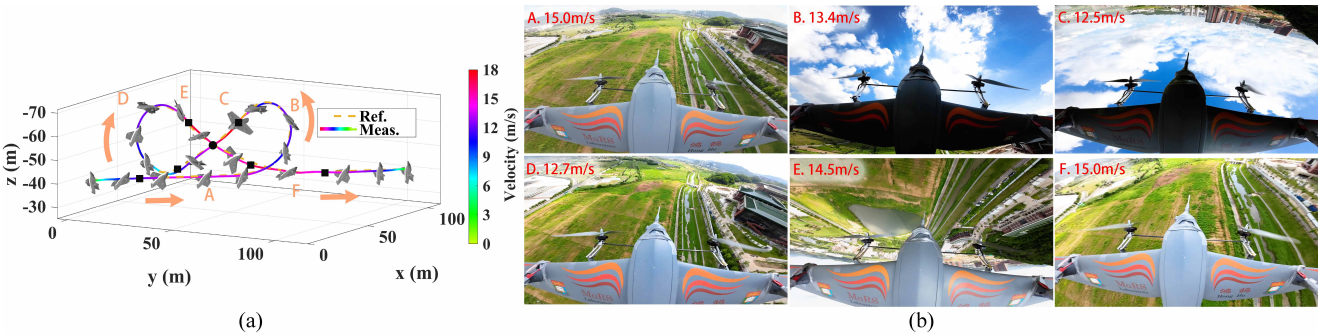


Fig. 26. Cuban Eight: (a) illustration of the trajectory, boundary points (black squares), and intermediate waypoints (black dots), (b) images from the FPV camera. Labels A-F denote different flight phases of the vehicle.

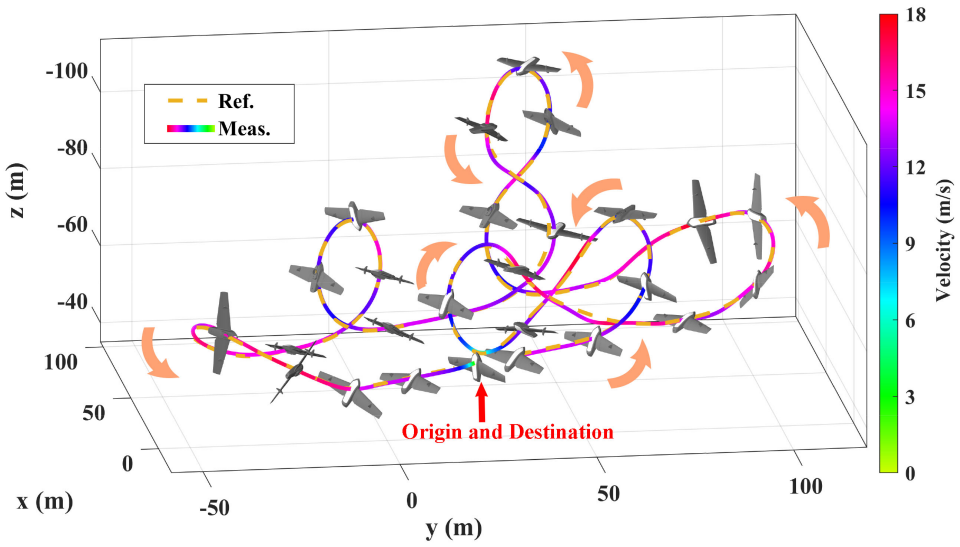


Fig. 27. Combo flight trajectory illustration.

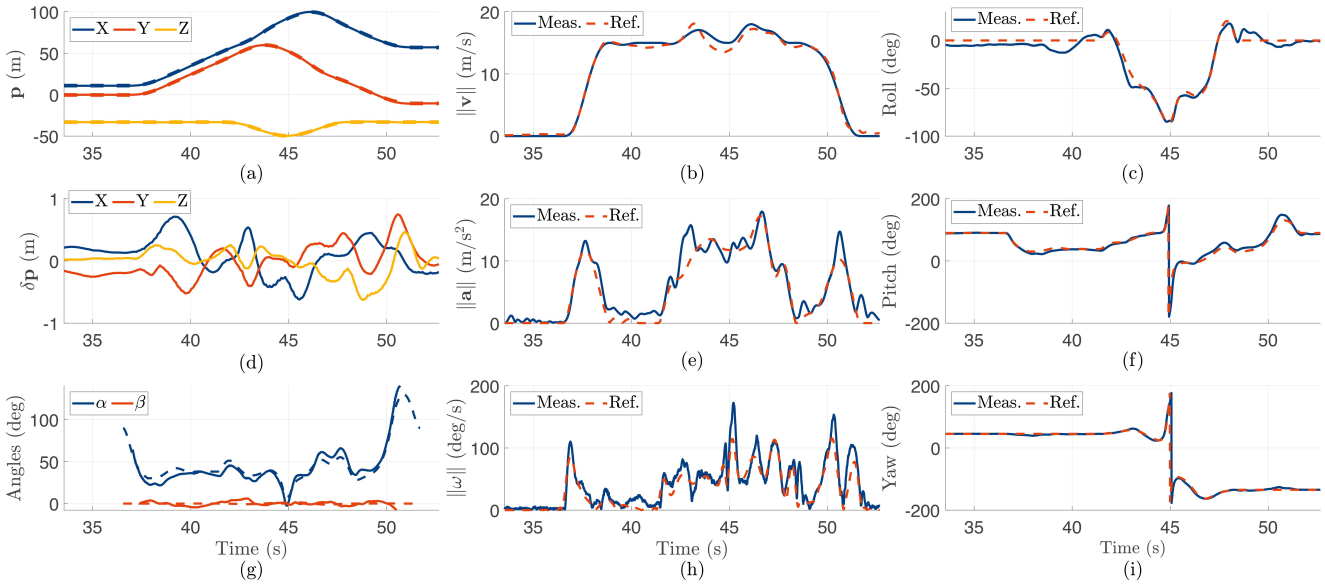


Fig. 28. Flight results of the aerobatic maneuver Wingover: (a) position, (b) flight speed, (c),(f) and (i) attitude in Euler angles, (d) position tracking errors, (e) acceleration. (g) angle of attack and sideslip angle, (h) angular velocity. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $2\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. In all subplots, the solid and dashed lines respectively denote the measurement and reference. Note that the ZXY Euler angle representation incur singularity when roll angle reaches 90° at $45\,\mathrm{s}$.

vehicle climbs $16.5\,\mathrm{m}$ at the top and achieves the specified 90° roll and yaw angle at $45\,\mathrm{s}$. Note that the ZXY Euler angle incurs singularity in the visualization, but our global on-manifold MPC has no such singularity as shown in the FPV image in Fig. 23 (b)D. Throughout the flight, the vehicle tracks all of the state trajectories closely: the position error is less than $0.75\,\mathrm{m}$ in all time and the sideslip angle is well stabilized around zero. This tracking accuracy is not trivial for outdoor UAV aerobatics with such large large span of angle of attack (up to 130°), acceleration (up to $18\,\mathrm{m/s^2}$), and angular velocity (up to $175\,\mathrm{deg/s}$).

The flight results of the Loop is shown in Fig. 29. The vehicle transits to $15\,\mathrm{m/s}$ and successfully finishes a Loop with radius of around $15\,\mathrm{m}$ in $10\,\mathrm{s}$. It is seen that the pitch angle rises to 180° at $44.6\,\mathrm{s}$, when the vehicle is totally upside-down at the top of the Loop as designed, which is also shown in Fig. 24 (b)D. The vehicle also tracks all of the state trajectories closely in the coordinated flight condition (i.e., the sideslip angle is shown around zero). The position error also remains below $1\,\mathrm{m}$ in all directions, even though the maximum acceleration and angular velocity increase to $25.5\,\mathrm{m/s^2}$ and $370\,\mathrm{deg/s}$, respectively.

As shown in Fig. 30, the vehicle finishes a more aggressive aerobatic maneuver of Vertical Eight also in high tracking accuracy. From $58\,\mathrm{s}$ in level flight, the vehicle begins to pull up the pitch angle to 145° and quickly lowers it to zero at $65\,\mathrm{s}$, meanwhile the vehicle simultaneously gains $60\,\mathrm{m}$ altitude by following a "S"-shape trajectory (i.e., position A-B-C in Fig. 25 (a)). After that, the vehicle flies another "S"-shape trajectory to decrease to the original altitude when the pitch angle continues to decrease to -260° (i.e, nearly free falling as shown in Fig. 25 (b)E) and quickly increases to 35°

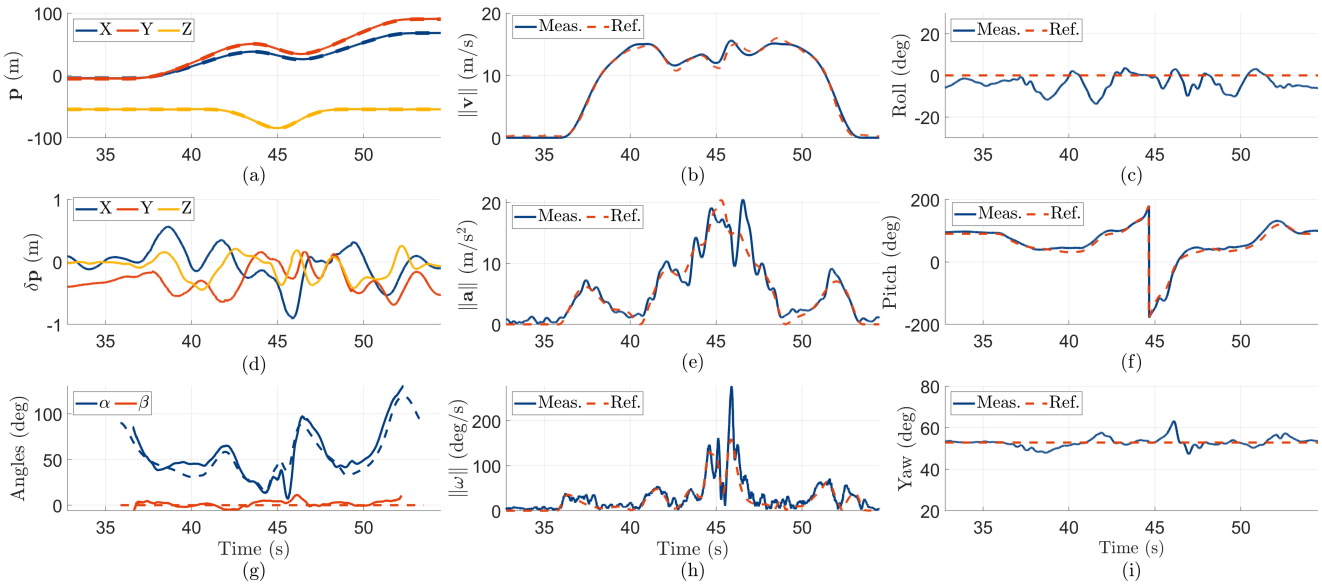


Fig. 29. Flight results of the aerobatic maneuver Loop: (a) position, (b) flight speed, (c),(f) and (i) attitude Euler angles, (d) position tracking errors, (e) acceleration. (g) angle of attack and sideslip angle, (h) angular velocity. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $2\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. In all subplot, the solid and dashed lines respectively denote the measurement and reference.

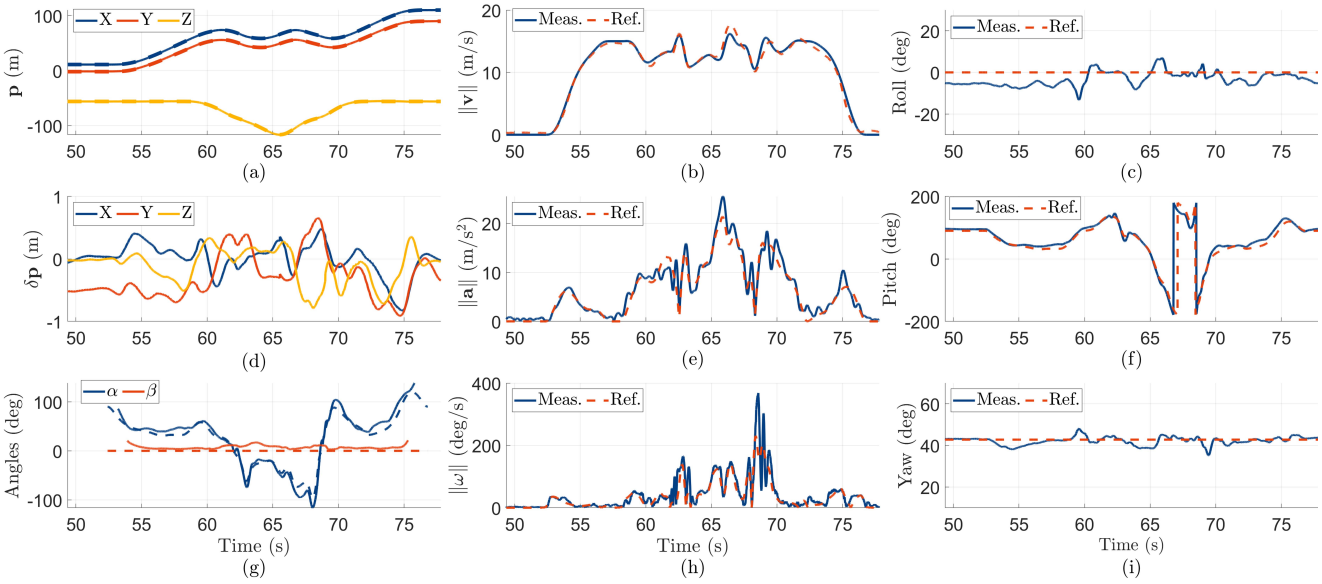


Fig. 30. Flight results of the aerobatic maneuver Vertical Eight: (a) position, (b) flight speed, (c),(f) and (i) attitude Euler angles, (d) position tracking errors, (e) acceleration. (g) angle of attack and sideslip angle, (h) angular velocity. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $2\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. In all subplot, the solid and dashed lines respectively denote the measurement and reference.

to perform a $15\,\mathrm{m/s}$ level flight again. It is seen that the angle of attack ranges from -115° to 120°, the largest span among all the demonstrated aerobatics. Moreover, the acceleration and angular velocity respectively peak at $25.5\,\mathrm{m/s^2}$ and $400\,\mathrm{deg/s}$. Despite such large span of angle of attack and high acceleration and angular velocity, the overall position error still remains less than $1\,\mathrm{m}$.

Similarly, the vehicle executes the Cuban Eight maneuver in high tracking performance despite the extremely high aggressiveness. The vehicle tracks the "∞"-shape with a width of $65\,\mathrm{m}$, a height of $30\,\mathrm{m}$, and a time duration of $15\,\mathrm{s}$. The pitch angle increases from 36° in level flight to 225° at position C in Fig 26 (a), then it decreases to -88° at position E and recovers to 36° at position F for level flight. The resulting span of angle of attack is about 220°. The acceleration and angular velocity peaks at $22\,\mathrm{m/s^2}$ and $400\,\mathrm{deg/s}$, respectively. The maximum position error slightly rises to $1.38\,\mathrm{m}$ due to the large control actuation but the overall tracking performance for the other state trajectories are still as good as other aerobatic maneuvers.

Moreover, we demonstrate a Combo trajectory by connecting the above aerobatic maneuvers in sequence, as shown in Fig. 27. The vehicle performs the Combo maneuver which requires extremely large control actuation over the entire $62\,\mathrm{s}$ flight. In Fig. 32, it is seen that the thrust acceleration and angular velocity commands computed by

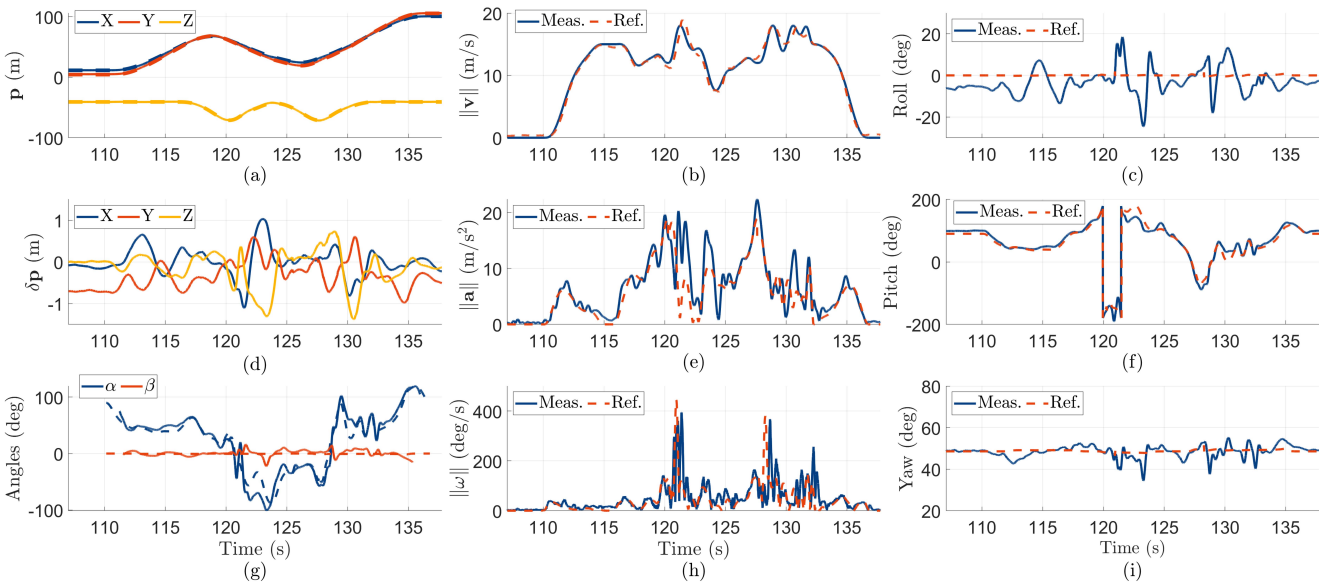


Fig. 31. Flight results of the aerobatic maneuver Cuban Eight: (a) position, (b) flight speed, (c),(f) and (i) attitude Euler angles, (d) position tracking errors, (e) acceleration. (g) angle of attack and sideslip angle, (h) angular velocity. For the angle of attack and sideslip angle, their measurements are displayed only when the airspeed exceeds $2\,\mathrm{m/s}$ due to the unstable airspeed measurements at low speeds. In all subplot, the solid and dashed lines respectively denote the measurement and reference.

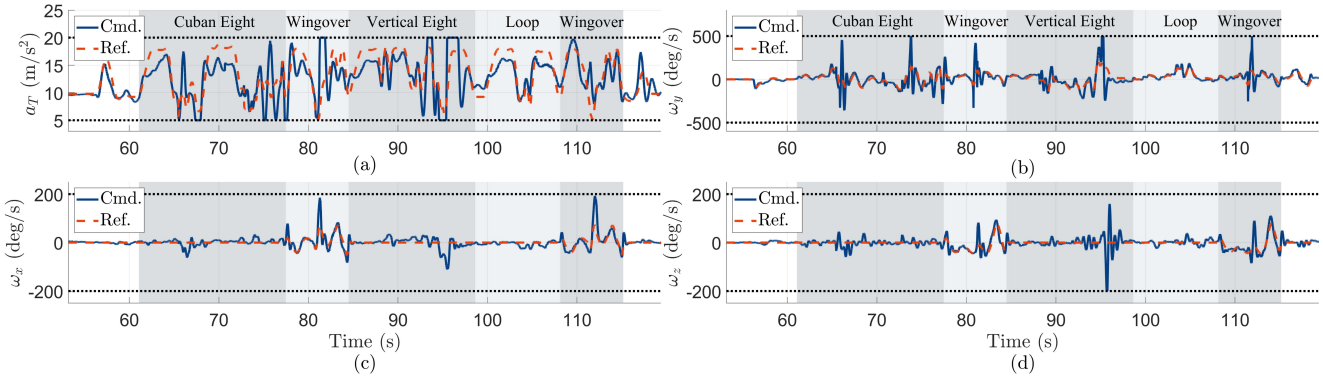


Fig. 32. Control efforts of the aerobatic maneuver Combo. The shaded areas indicate the comprising maneuvers of Cuban Eight, Wingover, Vertical Eight, Loop and Wingover. The dotted lines denote input constraints in the MPC optimization (61).

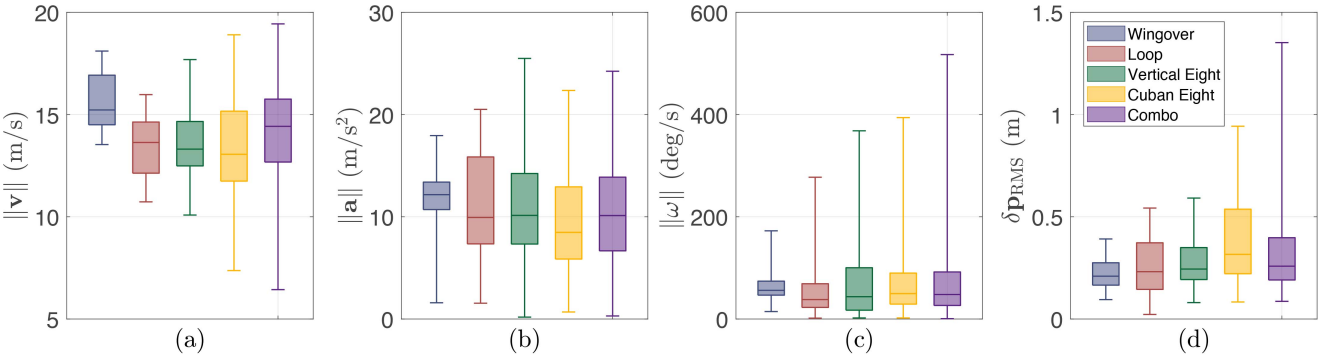


Fig. 33. The aggressiveness and control performance of five aerobatic maneuvers in the Combo flight. (a-d) show the norm of velocity, acceleration, angular velocity and position control error, respectively.

the MPC frequently touch their limits, but the controller still manages to stabilize the vehicle under such control saturation.

The trajectory aggressiveness and tracking accuracy of the five aerobatic maneuvers in the Combo flight are statistically analyzed in Fig. 33. The maximum velocity, acceleration and angular velocity reach $19.4\,\mathrm{m/s}$, $25.5\,\mathrm{m/s}$ and $520\,\mathrm{deg/s}$, respectively. Still, the proposed global controller shows a remarkable tracking performance that the average position tracking error is $0.33\,\mathrm{m}$ and largest position error is only $1.35\,\mathrm{m}$. Readers are highly recommended to watch the accompanying videos for better visualization of the experiments.

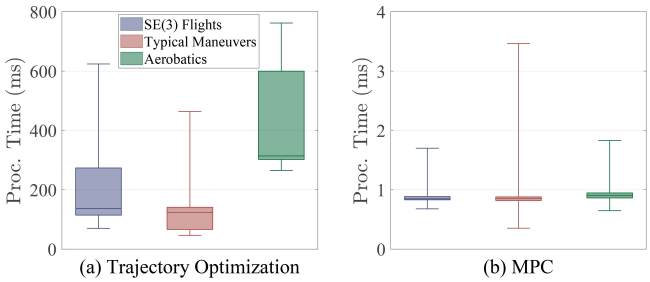


Fig. 34. Time consumption of (a) solving the proposed trajectory optimization to generating a trajectory segment, (b) solving the proposed MPC at each control step.

8.5 Time consumption

The statistical time consumption of the trajectory planner and MPC in all of the above flight tests including $SE(3)$ flights, typical maneuvers, and aerobatic maneuvers is summarized in Fig. 34. For the trajectory generation in (48) which runs offline, the computation time is about 45-750 ms to generate one trajectory segment with a length of about 2.5-50 m. The average and maximum time consumption to solve the MPC problem in (61) in total is 0.84 ms and 3.46 ms, respectively, showing a high computational efficiency ensuring online implementation at 100 Hz.

9 Conclusion

In this section, we discuss the limitation and extension of the proposed framework, and then draw the conclusion.

9.1 Limitation

Our proposed framework is a model-based approach. Higher tracking accuracy requires a more precise dynamic model, especially the aerodynamic model. However, identifying a high-fidelity aerodynamic model generally requires high-cost and time-consuming wind tunnel tests. The cost and time escalate for tail-sitter UAVs where the envelope of angle of attack is large. In this paper, we leveraged the wind tunnel test data in (Lyu et al. 2018a). For general tail-sitter UAVs, such aerodynamic model could be identified from onboard sensor data collected in real flights, which would be a promising future research to pursue.

Another limitation lies in the robustness and computation efficiency of the trajectory planner. In this paper, we adopted the MINCO trajectory optimization framework (Wang et al. 2022), which parameterizes the trajectory by a multi-stage polynomial and penalizes the constraints in the objective function as soft constraints. Softly penalizing the constraints in objective functions could reduce the optimization time by eliminating the hard constraints. However, due to the extremely nonlinear objective function, the solver could easily converge to local minimum violating the constraints. This phenomenon occasionally occurred in the planning of the outdoor aerobatic trajectories when the waypoints locations are poorly specified. Moreover, the optimization time is still quite long, 40-750 ms, preventing it from real-time implementation on current tail-sitter onboard computing devices.

9.2 Extension

Firstly, the proposed trajectory optimization could potentially be solved more efficiently by leveraging state-of-the-art nonlinear optimization techniques (e.g., Schulman et al. (2014); Gill et al. (2005)), the availability of higher-performance onboard computing devices, and the parallelization of the optimization based on Graphic Processing Units (GPUs). With an efficient solution, the proposed trajectory generation could serve as a reliable back-end planner for on-line trajectory planning. Equipped with onboard sensors such as cameras and lidars, and the corresponding front-end corridor generation techniques (e.g., Liu et al. (2017); Gao et al. (2019)), the tail-sitter could perform autonomous obstacle avoidance in cluttered environments.

Secondly, the tracking accuracy can be further improved by augmenting a low-level controller to the thrust acceleration a_T. In the present implementation, we directly mapped the thrust acceleration command a_T to the collective throttle of the four motors. However, the actual propeller thrust is also affected by various other factors, such as the propeller inflow (Brandt and Selig 2011; Gill and D'andrea 2017) and motor internal dynamics. These factors have caused significant errors between the actual and commanded thrust acceleration as shown in our experiment results. This issue could be mitigated by tracking the thrust acceleration command a_T with a low-level controller based on accelerometer measurements.

Thirdly, other than the model predictive controller in the present implementation, the flatness function provides a possibility to design a more light-weight cascaded PID controller that runs on low-cost micro processors. The cascaded control architecture could be similar to that of a multicopter: an outer-loop position controller first computes the desired acceleration, then our differential flatness function maps the desired acceleration to the desired attitude and thrust, finally the attitude is tracked by an inner-loop attitude controller. Such a cascaded control structure is also used in existing works (Ritz and D'Andrea 2017; Cheng and Pei 2022), but based on an over-simplified aerodynamic model.

Finally, the proposed framework can be extended to other configurations of tail-sitter UAVs, such as the single-propeller configuration (Frank et al. 2007; Wang et al. 2017b; De Wagter et al. 2018) and the shoulder-mounted twin-engine configuration (Bapst et al. 2015; Ritz and D'Andrea 2017; Sun et al. 2018). All of he trajectory generation, flatness transform and global control for the high-level system can be directly applied to the other configurations, while the low-level controller could be re-designed according to the specific vehicle dynamic parameters and actuator performances.

9.3 Conclusion

In this paper, we proposed a trajectory generation and global tracking controller for aggressive agile tail-sitter flights. The foundation of the framework is the differential flatness property that is proved in coordinated flights. The singularity conditions occurred in the flatness function were fully investigated and resolved in the framework. Based on these theoretical results, we developed a trajectory optimization

framework for trajectory generation and a model predictive controller for trajectory tracking. The entire approach is tested on a quadrotor tail-sitter prototype in extensive real-world flights. Notably, we demonstrated agile $SE(3)$ flights in indoor environments and aerobatic maneuvers in windy outdoor environments, which were rarely shown in any prior literature works. Extensive flight tests on typical maneuvers of transition, level flight and loiter, have also shown a superior tracking accuracy compared to existing methods.

Acknowledgment

This work was supported in part by Hong Kong RGC ECS under grant 27202219 and in part by DJI donation. The authors appreciate Dr. Wei Xu and Dr. Haowei Gu for the help on the initial set up of the prototype, and Miss Huirong Cheng for discussion. The authors especially thank Dr. Ximin Lyu of Sun Yat-sen University for providing the space and equipment for conducting the indoor experiments.

References

Bapst R, Ritz R, Meier L and Pollefeys M (2015) Design and implementation of an unmanned tail-sitter. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 1885–1890.

Barth JM, Condomines JP, Bronz M, Moschetta JM, Join C and Fliess M (2020) Model-free control algorithms for micro air vehicles with transitioning flight capabilities. *International Journal of Micro Air Vehicles* 12: 1756829320914264.

Borrelli F, Bemporad A and Morari M (2017) *Predictive control for linear and hybrid systems*. Cambridge University Press.

Brandt J and Selig M (2011) Propeller performance data at low reynolds numbers. In: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. p. 1255.

Bry A, Richter C, Bachrach A and Roy N (2015) Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *The International Journal of Robotics Research* 34(7): 969–1002.

Bullo F and Murray RM (1995) Proportional derivative (pd) control on the euclidean group .

Carlson S (2014) A hybrid tricopter/flying-wing vtol uav. In: *52nd Aerospace Sciences Meeting*. p. 0016.

Çetinsoy E, Sirimoğlu E, Öner KT, Hancer C, Ünel M, Akşit MF, Kandemir I and Gülez K (2011) Design and development of a tilt-wing uav. *Turkish Journal of Electrical Engineering and Computer Sciences* 19(5): 733–741.

Cheng ZH and Pei HL (2022) Transition analysis and practical flight control for ducted fan fixed-wing aerial robot: Level path flight mode transition. *IEEE Robotics and Automation Letters* 7(2): 3106–3113.

Chitsaz H and LaValle SM (2007) Time-optimal paths for a dubins airplane. In: *2007 46th IEEE conference on decision and control*. IEEE, pp. 2379–2384.

Clancy LJ (1975) *Aerodynamics*. John Wiley & Sons.

De Wagter C, Ruijsink R, Smeur EJ, van Hecke KG, van Tienen F, van der Horst E and Remes BD (2018) Design, control, and visual navigation of the delftacopter vtol tail-sitter uav. *Journal of Field Robotics* 35(6): 937–960.

Ding W, Gao W, Wang K and Shen S (2019) An efficient b-spline-based kinodynamic replanning framework for quadrotors. *IEEE Transactions on Robotics* 35(6): 1287–1306.

Etkin B and Reid LD (1959) *Dynamics of flight*, volume 2. Wiley New York.

Faessler M, Franchi A and Scaramuzza D (2017) Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robotics and Automation Letters* 3(2): 620–626.

Falanga D, Foehn P, Lu P and Scaramuzza D (2018) Pampc: Perception-aware model predictive control for quadrotors. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1–8.

Falanga D, Mueggler E, Faessler M and Scaramuzza D (2017) Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 5774–5781.

Fliess M, Lévine J, Martin P and Rouchon P (1995) Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control* 61(6): 1327–1361.

Foehn P, Romero A and Scaramuzza D (2021) Time-optimal planning for quadrotor waypoint flight. *Science Robotics* 6(56): eabh1221.

Frank A, McGrew J, Valenti M, Levine D and How J (2007) Hover, transition, and level flight control design for a single-propeller indoor airplane. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. p. 6318.

Gao F, Wu W, Gao W and Shen S (2019) Flying on point clouds: Online trajectory generation and autonomous navigation for quadrotors in cluttered environments. *Journal of Field Robotics* 36(4): 710–733.

Gertz EM and Wright SJ (2003) Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software (TOMS)* 29(1): 58–81.

Gill PE, Murray W and Saunders MA (2005) Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review* 47(1): 99–131.

Gill R and D'andrea R (2017) Propeller thrust and drag in forward flight. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, pp. 73–79.

Gu H, Cai X, Zhou J, Li Z, Shen S and Zhang F (2018) A coordinate descent method for multidisciplinary design optimization of electric-powered winged uavs. In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 1189–1198.

Gu H, Lyu X, Li Z, Shen S and Zhang F (2017) Development and experimental verification of a hybrid vertical take-off and landing (vtol) unmanned aerial vehicle (uav). In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 160–169.

Hang K, Lyu X, Song H, Stork JA, Dollar AM, Kragic D and Zhang F (2019) Perching and resting—a paradigm for uav maneuvering with modularized landing gears. *Science Robotics* 4(28): eaau6637.

Hauser J, Sastry S and Meyer G (1992) Nonlinear control design for slightly non-minimum phase systems: Application to v/stol aircraft. *Automatica* 28(4): 665–679.

Hertzberg C, Wagner R, Frese U and Schröder L (2013) Integrating generic sensor fusion algorithms with sound

state representations through encapsulation of manifolds. *Information Fusion* 14(1): 57–77.

Johansen TA, Cristofaro A, Sørensen K, Hansen JM and Fossen TI (2015) On estimation of wind velocity, angle-of-attack and sideslip angle of small uavs using standard sensors. In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 510–519.

Jung Y and Shim DH (2012) Development and application of controller for transition flight of tail-sitter uav. *Journal of Intelligent & Robotic Systems* 65(1): 137–152.

Kamel M, Burri M and Siegwart R (2017) Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles. *IFAC-PapersOnLine* 50(1): 3463–3469.

Kaufmann E, Loquercio A, Ranftl R, Müller M, Koltun V and Scaramuzza D (2020) Deep drone acrobatics. *RSS: Robotics, Science, and Systems* .

Kita K, Konno A and Uchiyama M (2010) Transition between level flight and hovering of a tail-sitter vertical takeoff and landing aerial robot. *Advanced Robotics* 24(5-6): 763–781.

Lambregts A (1983) Vertical flight path and speed control autopilot design using total energy principles. In: *Guidance and Control Conference*. p. 2239.

Li B, Sun J, Zhou W, Wen CY, Low KH and Chen CK (2020a) Transition optimization for a vtol tail-sitter uav. *IEEE/ASME transactions on mechatronics* 25(5): 2534–2545.

Li B, Zhou W, Sun J, Wen CY and Chen CK (2018) Development of model predictive controller for a tail-sitter vtol uav in hover flight. *Sensors* 18(9): 2859.

Li Y, Qin Y, Xu W and Zhang F (2020b) Modeling, identification, and control of non-minimum phase dynamics of bi-copter uavs. In: *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, pp. 1249–1255.

Liu S, Watterson M, Mohta K, Sun K, Bhattacharya S, Taylor CJ and Kumar V (2017) Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robotics and Automation Letters* 2(3): 1688–1695.

Lu G, Xu W and Zhang F (2022) On-manifold model predictive control for trajectory tracking on robotic systems. *IEEE Transactions on Industrial Electronics* .

Lustosa LR (2017) *The Phi-theory approach to flight control design of hybrid vehicles*. PhD Thesis, PhD thesis, ISAE-SUPAERO.

Lustosa LR, Defaÿ F and Moschetta JM (2019) Global singularity-free aerodynamic model for algorithmic flight control of tail sitters. *Journal of Guidance, Control, and Dynamics* 42(2): 303–316.

Lyu X, Gu H, Wang Y, Li Z, Shen S and Zhang F (2017a) Design and implementation of a quadrotor tail-sitter vtol uav. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 3924–3930.

Lyu X, Gu H, Zhou J, Li Z, Shen S and Zhang F (2017b) A hierarchical control approach for a quadrotor tail-sitter vtol uav and experimental verification. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5135–5141.

Lyu X, Gu H, Zhou J, Li Z, Shen S and Zhang F (2018a) Simulation and flight experiments of a quadrotor tail-sitter vertical take-off and landing unmanned aerial vehicle with wide flight envelope. *International Journal of Micro Air Vehicles* 10(4): 303–317.

Lyu X, Zhou J, Gu H, Li Z, Shen S and Zhang F (2018b) Disturbance observer based hovering control of quadrotor tail-sitter vtol uavs using H_∞ synthesis. *IEEE Robotics and Automation Letters* 3(4): 2910–2917.

Marion P (2021) Flipping the script with atlas. URL https://www.bostondynamics.com/resources/blog/flipping-script-atlas. Accessed on 08 23, 2021.

Martin P, Devasia S and Paden B (1996) A different look at output tracking: Control of a vtol aircraft. *Automatica* 32(1): 101–107.

Matsumoto T, Kita K, Suzuki R, Oosedo A, Go K, Hoshino Y, Konno A and Uchiyama M (2010) A hovering control strategy for a tail-sitter vtol uav that increases stability against large disturbance. In: *2010 IEEE international conference on robotics and automation*. IEEE, pp. 54–59.

Mayne DQ, Rawlings JB, Rao CV and Scokaert PO (2000) Constrained model predictive control: Stability and optimality. *Automatica* 36(6): 789–814.

McIntosh K and Mishra S (2022) Transition trajectory planning and control for quadrotor biplanes in obstacle cluttered environments .

McKenna J (2007) One step beyond, rotor wing.

Meier L, Honegger D and Pollefeys M (2015) Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 6235–6240.

Mellinger D and Kumar V (2011) Minimum snap trajectory generation and control for quadrotors. In: *2011 IEEE international conference on robotics and automation*. IEEE, pp. 2520–2525.

Mellinger D, Michael N and Kumar V (2012) Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research* 31(5): 664–674.

Mueller MW, Hehn M and D'Andrea R (2015) A computationally efficient motion primitive for quadrocopter trajectory generation. *IEEE transactions on robotics* 31(6): 1294–1310.

Murray RM, Rathinam M and Sluis W (1995) Differential flatness of mechanical control systems: A catalog of prototype systems. In: *ASME international mechanical engineering congress and exposition*. Citeseer.

Naldi R and Marconi L (2011) Optimal transition maneuvers for a class of v/stol aircraft. *Automatica* 47(5): 870–879.

Nguyen H, Kamel M, Alexis K and Siegwart R (2021) Model predictive control for micro aerial vehicles: A survey. In: *2021 European Control Conference (ECC)*. IEEE, pp. 1556–1563.

Noormohammadi-Asl A, Esrafilian O, Arzati MA and Taghirad HD (2020) System identification and H_∞-based control of quadrotor attitude. *Mechanical Systems and Signal Processing* 135: 106358.

Oosedo A, Abiko S, Konno A, Koizumi T, Furui T and Uchiyama M (2013) Development of a quad rotor tail-sitter vtol uav without control surfaces and experimental verification. In: *2013 IEEE international conference on robotics and automation*. IEEE, pp. 317–322.

Oosedo A, Abiko S, Konno A and Uchiyama M (2017) Optimal transition from hovering to level-flight of a quadrotor tail-sitter uav. *Autonomous Robots* 41(5): 1143–1159.

Ozdemir U, Aktas YO, Vuruskan A, Dereli Y, Tarhan AF, Demirbag K, Erdem A, Kalaycioglu GD, Ozkol I and Inalhan G (2014) Design of a commercial hybrid vtol uav system. *Journal of Intelligent & Robotic Systems* 74(1): 371–393.

Park R (2014) Arcturus uav upgrades the jump15 vtol uav. *Airlines & Aviation, Aerospace & Defense* .

Park S, Deyst J and How J (2004) A new nonlinear guidance logic for trajectory tracking. In: *AIAA guidance, navigation, and control conference and exhibit*. p. 4900.

Pucci D (2012) Flight dynamics and control in relation to stall. In: *2012 American Control Conference (ACC)*. IEEE, pp. 118–124.

Pucci D, Hamel T, Morin P and Samson C (2013) Nonlinear control of aerial vehicles subjected to aerodynamic forces. In: *52nd IEEE Conference on Decision and Control*. IEEE, pp. 4839–4846.

Ren Y, Liang S, Zhu F, Lu G and Zhang F (2023) Online whole-body motion planning for quadrotor using multi-resolution search. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 1594–1600. DOI:10.1109/ICRA48891.2023.10160767.

Ritz R and D'Andrea R (2017) A global controller for flying wing tailsitter vehicles. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 2731–2738.

Romero A, Penicka R and Scaramuzza D (2022) Time-optimal online replanning for agile quadrotor flight. *IEEE Robotics and Automation Letters* 7(3): 7730–7737.

Schulman J, Duan Y, Ho J, Lee A, Awwal I, Bradlow H, Pan J, Patil S, Goldberg K and Abbeel P (2014) Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research* 33(9): 1251–1270.

Shen S, Michael N and Kumar V (2011) Autonomous multi-floor indoor navigation with a computationally constrained mav. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 20–25.

Smeur EJ, Bronz M and de Croon GC (2020) Incremental control and guidance of hybrid aircraft applied to a tailsitter unmanned air vehicle. *Journal of Guidance, Control, and Dynamics* 43(2): 274–287.

Stevens BL, Lewis FL and Johnson EN (2015) *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons.

Sun J, Li B, Wen CY and Chen CK (2018) Design and implementation of a real-time hardware-in-the-loop testing platform for a dual-rotor tail-sitter unmanned aerial vehicle. *Mechatronics* 56: 1–15.

Sun S, Romero A, Foehn P, Kaufmann E and Scaramuzza D (2022) A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight. *IEEE Transactions on Robotics* : 1–17DOI:10.1109/TRO.2022.3177279.

Tal E and Karaman S (2022) Global incremental flight control for agile maneuvering of a tailsitter flying wing. *Journal of Guidance, Control, and Dynamics* 45(12): 2332–2349.

Tal E, Ryou G and Karaman S (2023) Aerobatic trajectory generation for a vtol fixed-wing aircraft using differential flatness. *IEEE Transactions on Robotics* .

Tal EA and Karaman S (2021) Global trajectory-tracking control for a tailsitter flying wing in agile uncoordinated flight. In: *AIAA AVIATION 2021 FORUM*. p. 3214.

Van Nieuwstadt MJ and Murray RM (1998) Rapid hover-to-forward-flight transitions for a thrust-vectored aircraft. *Journal of Guidance, Control, and Dynamics* 21(1): 93–100.

Verling S, Weibel B, Boosfeld M, Alexis K, Burri M and Siegwart R (2016) Full attitude control of a vtol tailsitter uav. In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 3006–3012.

Vourtsis C, Rochel VC, Müller NS, Stewart W and Floreano D (2023) Wind defiant morphing drones. *Advanced Intelligent Systems* 5(3): 2200297.

Wang K, Ke Y, Lai S, Gong K, Tan Y and Chen BM (2017a) Model-based optimal auto-transition and control synthesis for tail-sitter uav kh-lion. In: *2017 13th IEEE International Conference on Control & Automation (ICCA)*. IEEE, pp. 541–547.

Wang W, Zhu J and Kuang M (2017b) Design, modelling and hovering control of a tail-sitter with single thrust-vectored propeller. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5971–5976.

Wang Z, Zhou X, Xu C and Gao F (2022) Geometrically constrained trajectory optimization for multicopters. *IEEE Transactions on Robotics* .

Xu W, Gu H, Qing Y, Lin J and Zhang F (2019a) Full attitude control of an efficient quadrotor tail-sitter vtol uav with flexible modes. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 542–550.

Xu W, Gu H and Zhang F (2019b) Acceleration based iterative learning control for pugachev's cobra maneuver with quadrotor tailsitter vtol uavs. *work* 7: 12.

Zhang J, Hu C, Chadha RG and Singh S (2020) Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation. *Journal of Field Robotics* 37(8): 1300–1313.

Zhou B, Gao F, Wang L, Liu C and Shen S (2019) Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters* 4(4): 3529–3536.

Zhou J, Lyu X, Cai X, Li Z, Shen S and Zhang F (2018) Frequency domain model identification and loop-shaping controller design for quadrotor tail-sitter vtol uavs. In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 1142–1149.

Zhou J, Lyu X, Li Z, Shen S and Zhang F (2017) A unified control method for quadrotor tail-sitter uavs in all flight modes: Hover, transition, and level flight. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4835–4841.

Appendix A Proof of theorem 1

Given the vehicle dynamics in (1), the rank of system dynamics derivative w.r.t input can be given by eliminating unrelated items:

$$\begin{aligned}\operatorname{rank}\left(\frac{\partial \mathbf{f}(\mathbf{x}_{\text{full}}, \mathbf{u}_{\text{full}})}{\partial \mathbf{u}_{\text{full}}}\right) &= \operatorname{rank}\left(\frac{\partial\left(\dot{\mathbf{v}}, \dot{\boldsymbol{\omega}}\right)}{\partial\left(a_T, \boldsymbol{\tau}\right)}\right) \\ &= \operatorname{rank}\left(\begin{bmatrix} \frac{\partial \dot{\mathbf{v}}}{\partial a_T} & \frac{\partial \dot{\mathbf{v}}}{\partial \boldsymbol{\tau}} \\ \frac{\partial \dot{\boldsymbol{\omega}}}{\partial a_T} & \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\tau}} \end{bmatrix}\right)\end{aligned} \tag{65}$$

where the following elements can be computed directly from the system dynamics in (1):

$$\frac{\partial \dot{\mathbf{v}}}{\partial a_T} = \mathbf{x}_b, \quad \frac{\partial \dot{\mathbf{v}}}{\partial \boldsymbol{\tau}} = \mathbf{0}, \quad \frac{\partial \dot{\boldsymbol{\omega}}}{\partial a_T} = \mathbf{0} \tag{66}$$

and $\frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\tau}}$ has coupling effect due to the coordinated flight condition that the vehicle has no lateral airspeed:

$$\mathbf{v}_{a_y}^{\mathcal{B}} = \mathbf{e}_2^T \mathbf{R}^T \mathbf{v}_a \equiv 0 \tag{67}$$

which leads to the derivative on the both sides:

$$-\mathbf{e}_2^T \lfloor \boldsymbol{\omega} \rfloor \mathbf{R}^T \mathbf{v}_a + \mathbf{e}_2^T \mathbf{R}^T \dot{\mathbf{v}}_a = 0 \tag{68a}$$

$$\Rightarrow \quad \mathbf{y}_b^T \dot{\mathbf{v}}_a = \mathbf{v}_a^T \mathbf{R} \lfloor \mathbf{e}_2 \rfloor \boldsymbol{\omega} \tag{68b}$$

$$\Rightarrow \quad \mathbf{y}_b^T \dot{\mathbf{v}}_a = \mathbf{v}_{a_x}^{\mathcal{B}} \boldsymbol{\omega}_z - \mathbf{v}_{a_z}^{\mathcal{B}} \boldsymbol{\omega}_x \tag{68c}$$

It is seen that the body angular velocity elements $\boldsymbol{\omega}_x$ and $\boldsymbol{\omega}_z$ are coupled. Without loss of generality, we consider $\boldsymbol{\omega}_z$ as a function of $\boldsymbol{\omega}_x$. Then we have

$$\frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\tau}} = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{\omega}}_x}{\partial \boldsymbol{\tau}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_y}{\partial \boldsymbol{\tau}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_z}{\partial \boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{\omega}}_x}{\partial \boldsymbol{\tau}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_y}{\partial \boldsymbol{\tau}} \\ \frac{\partial \dot{\boldsymbol{\omega}}_z}{\partial \dot{\boldsymbol{\omega}}_x} \frac{\partial \dot{\boldsymbol{\omega}}_x}{\partial \boldsymbol{\tau}} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\partial \dot{\boldsymbol{\omega}}_z}{\partial \dot{\boldsymbol{\omega}}_x} & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \mathbf{J}^{-1} \tag{69}$$

With (65), (66) and (69), the derivative is finally computed and transformed based on the fact of non-zero vector $\mathbf{x}_b$ and the full-rank inertia matrix:

$$\begin{bmatrix} \frac{\partial \dot{\mathbf{v}}}{\partial a_T} & \frac{\partial \dot{\mathbf{v}}}{\partial \boldsymbol{\tau}} \\ \frac{\partial \dot{\boldsymbol{\omega}}}{\partial a_T} & \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{-1} \end{bmatrix} \sim \begin{bmatrix} \mathbf{x}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \tag{70}$$

Therefore, we have its rank

$$\operatorname{rank}\left(\frac{\partial \mathbf{f}(\mathbf{x}_{\text{full}}, \mathbf{u}_{\text{full}})}{\partial \mathbf{u}_{\text{full}}} \right) = \operatorname{rank}\left(\begin{bmatrix} \mathbf{x}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \right) = 3 \tag{71}$$

Appendix B Proof of theorem 2

Reminding the aerodynamic force in (6) and the coordinated flight condition that there is no lateral airspeed in (27) (i.e., $\mathbf{e}_2^T \mathbf{v}_a^{\mathcal{B}} = 0$) , we have

$$\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} = \frac{\rho S}{2} \left(\mathbf{c} \frac{\partial V^2}{\partial \mathbf{v}_a^{\mathcal{B}}} + V^2 \frac{\partial \mathbf{c}}{\partial \alpha} \frac{\partial \alpha}{\partial \mathbf{v}_a^{\mathcal{B}}} + V^2 \frac{\partial \mathbf{c}}{\partial \beta} \frac{\partial \beta}{\partial \mathbf{v}_a^{\mathcal{B}}} \right) \tag{72}$$

where

$$\frac{\partial V^2}{\partial \mathbf{v}_a^{\mathcal{B}}} = \frac{\partial \|\mathbf{v}_a^{\mathcal{B}}\|^2}{\partial \mathbf{v}_a^{\mathcal{B}}} = 2\mathbf{v}_a^{\mathcal{B}^T} \tag{73a}$$

$$\begin{aligned} \frac{\partial \alpha}{\partial \mathbf{v}_a^{\mathcal{B}}} &= \frac{\partial}{\partial \mathbf{v}_a^{\mathcal{B}}} \tan^{-1} \frac{\mathbf{e}_3^T \mathbf{v}_a^{\mathcal{B}}}{\mathbf{e}_1^T \mathbf{v}_a^{\mathcal{B}}} \\ &= \frac{1}{1 + \left(\frac{\mathbf{e}_3^T \mathbf{v}_a^{\mathcal{B}}}{\mathbf{e}_1^T \mathbf{v}_a^{\mathcal{B}}} \right)^2} \frac{\mathbf{e}_1^T \mathbf{v}_a^{\mathcal{B}} \mathbf{e}_3^T - \mathbf{e}_3^T \mathbf{v}_a^{\mathcal{B}} \mathbf{e}_1^T}{\left(\mathbf{e}_1^T \mathbf{v}_a^{\mathcal{B}} \right)^2} \\ &= \frac{\begin{bmatrix} -\mathbf{v}_{a_z}^{\mathcal{B}} & 0 & \mathbf{v}_{a_x}^{\mathcal{B}} \end{bmatrix}}{\mathbf{v}_{a_x}^{\mathcal{B}^2} + \mathbf{v}_{a_z}^{\mathcal{B}^2}} \\ &= \frac{\mathbf{v}_a^{\mathcal{B}^T} \lfloor \mathbf{e}_2 \rfloor}{V^2} \end{aligned} \tag{73b}$$

$$\begin{aligned} \frac{\partial \beta}{\partial \mathbf{v}_a^{\mathcal{B}}} &= \frac{\partial}{\partial \mathbf{v}_a^{\mathcal{B}}} \left(\sin^{-1} \frac{\mathbf{e}_2^T \mathbf{v}_a^{\mathcal{B}}}{\|\mathbf{v}_a^{\mathcal{B}}\|} \right) \\ &= \frac{1}{\sqrt{1 - \left(\frac{\mathbf{e}_2^T \mathbf{v}_a^{\mathcal{B}}}{\|\mathbf{v}_a^{\mathcal{B}}\|} \right)^2}} \frac{\|\mathbf{v}_a^{\mathcal{B}}\| \mathbf{e}_2^T - \mathbf{e}_2^T \mathbf{v}_a^{\mathcal{B}} \frac{\mathbf{v}_a^{\mathcal{B}^T}}{\|\mathbf{v}_a^{\mathcal{B}}\|}}{\|\mathbf{v}_a^{\mathcal{B}}\|^2} \\ &= \frac{\mathbf{e}_2^T}{V} \end{aligned} \tag{73c}$$

and the aerodynamic coefficient gradients $\frac{\partial \mathbf{c}}{\partial \alpha}$ and $\frac{\partial \mathbf{c}}{\partial \beta}$ of an axially symmetric airframe satisfies (11). Substituting (73) into (72), we have

$$\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} = \frac{\rho S}{2} \left(2\mathbf{c}\mathbf{v}_a^{\mathcal{B}^T} + \frac{\partial \mathbf{c}}{\partial \alpha} \mathbf{v}_a^{\mathcal{B}^T} \lfloor \mathbf{e}_2 \rfloor + V \frac{\partial \mathbf{c}}{\partial \beta} \mathbf{e}_2^T \right) \tag{74}$$

Appendix C Calculation of matrices $\dot{\mathbf{N}}$ and $\dot{\mathbf{h}}$

As the matrices $\mathbf{N}$ and $\mathbf{h}$ are broken into block matrices in (29), their time derivatives can be taken in block matrices as follows:

$$\dot{\mathbf{h}} = \begin{bmatrix} \dot{\mathbf{h}}_1 \\ \dot{\mathbf{h}}_2 \end{bmatrix} \quad , \dot{\mathbf{N}} = \begin{bmatrix} \dot{\mathbf{N}}_1 \\ \dot{\mathbf{N}}_2 \end{bmatrix} \tag{75}$$

where

$$\begin{aligned} \dot{\mathbf{h}}_1 &= \frac{d}{dt} \left(\mathbf{y}_b^T \dot{\mathbf{v}}_a \right) \\ &= \frac{d}{dt} \left(\mathbf{e}_2^T \mathbf{R}^T \dot{\mathbf{v}}_a \right) \\ &= \mathbf{e}_2^T \left(-\lfloor \boldsymbol{\omega} \rfloor \mathbf{R}^T \dot{\mathbf{v}}_a + \mathbf{R}^T \ddot{\mathbf{v}}_a \right) \end{aligned} \tag{76a}$$

$$\begin{aligned} \dot{\mathbf{h}}_2 &= \frac{d}{dt} \left(\ddot{\mathbf{v}} - \frac{1}{m} \mathbf{R} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \mathbf{R}^T \dot{\mathbf{v}}_a \right) \\ &= \dddot{\mathbf{v}} - \frac{1}{m} \mathbf{R} \left(\lfloor \boldsymbol{\omega} \rfloor \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \mathbf{R}^T \dot{\mathbf{v}}_a + \frac{d}{dt} \left(\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \right) \mathbf{R}^T \dot{\mathbf{v}}_a \right. \\ &\quad \left. - \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \lfloor \boldsymbol{\omega} \rfloor \mathbf{R}^T \dot{\mathbf{v}}_a + \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \mathbf{R}^T \ddot{\mathbf{v}}_a \right) \end{aligned} \tag{76b}$$

$$\dot{\mathbf{N}}_1 = \begin{bmatrix} 0 & \dot{\mathbf{N}}_{12} \end{bmatrix} \tag{76c}$$

$$\dot{\mathbf{N}}_{12} = \frac{d}{dt} \left(\mathbf{v}_a^T \mathbf{R} \lfloor \mathbf{e}_2 \rfloor \right) = \left(\dot{\mathbf{v}}_a^T \mathbf{R} + \mathbf{v}_a^T \mathbf{R} \lfloor \boldsymbol{\omega} \rfloor \right) \lfloor \mathbf{e}_2 \rfloor \tag{76d}$$

$$\dot{\mathbf{N}}_2 = \begin{bmatrix} \dot{\mathbf{N}}_{21} & \dot{\mathbf{N}}_{22} \end{bmatrix} \tag{76e}$$

$$\dot{\mathbf{N}}_{21} = \frac{d}{dt} \left(\mathbf{R} \mathbf{e}_1 \right) = \mathbf{R} \lfloor \boldsymbol{\omega} \rfloor \mathbf{e}_1 \tag{76f}$$

$$\begin{aligned} \dot{\mathbf{N}}_{22} &= \frac{d}{dt} \left(\mathbf{R} \left(- \left\lfloor \left(a_T \mathbf{e}_1 + \frac{\mathbf{f}_a}{m} \right) \right\rfloor + \frac{1}{m} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor \right) \right) \\ &= \mathbf{R} \lfloor \boldsymbol{\omega} \rfloor \left(- \left\lfloor \left(a_T \mathbf{e}_1 + \frac{\mathbf{f}_a}{m} \right) \right\rfloor + \frac{1}{m} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor \right) \\ &\quad + \mathbf{R} \left(- \left\lfloor \left(\dot{a}_T \mathbf{e}_1 + \frac{1}{m} \left(\frac{\partial \mathbf{f}_a}{\partial V} \dot{V} + \frac{\partial \mathbf{f}_a}{\partial \alpha} \dot{\alpha} \right) \right) \right\rfloor \right. \\ &\quad \left. + \frac{1}{m} \left(\left(\frac{d}{dt} \left(\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \right) \right) \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor + \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \lfloor \dot{\mathbf{v}}_a^{\mathcal{B}} \rfloor \right) \right) \end{aligned} \tag{76g}$$

$$\frac{\partial \mathbf{f}_a}{\partial V} = \rho V S \mathbf{c} \tag{76h}$$

$$\frac{\partial \mathbf{f}_a}{\partial \alpha} = \frac{1}{2}\rho V^2 S \frac{\partial \mathbf{c}}{\partial \alpha} \tag{76i}$$

$$\dot{\mathbf{v}}_a^{\mathcal{B}} = \frac{d}{dt}\left(\mathbf{R}^T \mathbf{v}_a\right) = -\lfloor \boldsymbol{\omega} \rfloor \mathbf{R}^T \mathbf{v} + \mathbf{R}^T \dot{\mathbf{v}}_a \tag{76j}$$

$$\begin{aligned} \dot{\alpha} &= \frac{1}{1+\left(\frac{\mathbf{v}_{a_z}^{\mathcal{B}}}{\mathbf{v}_{a_x}^{\mathcal{B}}}\right)^2} \frac{\dot{\mathbf{v}}_{a_z}^{\mathcal{B}} \mathbf{v}_{a_x}^{\mathcal{B}} - \mathbf{v}_{a_z}^{\mathcal{B}} \dot{\mathbf{v}}_{a_x}^{\mathcal{B}}}{\mathbf{v}_{a_x}^{\mathcal{B}^2}} \\ &= \frac{\dot{\mathbf{v}}_{a_z}^{\mathcal{B}} \mathbf{v}_{a_x}^{\mathcal{B}} - \mathbf{v}_{a_z}^{\mathcal{B}} \dot{\mathbf{v}}_{a_x}^{\mathcal{B}}}{V^2} \end{aligned} \tag{76k}$$

$$\dot{\mathbf{v}}_a = \dot{\mathbf{v}} - \dot{\mathbf{w}} \tag{76l}$$

$$\ddot{\mathbf{v}}_a = \ddot{\mathbf{v}} - \ddot{\mathbf{w}} \tag{76m}$$

$$\dot{V} = \mathbf{v}_a^T \dot{\mathbf{v}}_a / V \tag{76n}$$

and with (74), we have

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}}\right) = \frac{\rho S}{2}\Bigg(2\left(\frac{\partial \mathbf{c}}{\partial \alpha}\dot{\alpha}\mathbf{v}_a^{\mathcal{B}^T} + \mathbf{c}\dot{\mathbf{v}}_a^{\mathcal{B}^T}\right) + \left(\frac{\partial^2 \mathbf{c}}{\partial \alpha^2}\dot{\alpha}\mathbf{v}_a^{\mathcal{B}^T} \\ + \frac{\partial \mathbf{c}}{\partial \alpha}\dot{\mathbf{v}}_a^{\mathcal{B}^T}\right)\lfloor \mathbf{e}_2 \rfloor + \left(\dot{V}\frac{\partial \mathbf{c}}{\partial \beta} + V\frac{\partial^2 \mathbf{c}}{\partial \beta \partial \alpha}\dot{\alpha}\right)\mathbf{e}_2^T\Bigg) \end{aligned} \tag{77}$$

Appendix D Proof of Theorem 3 (determinant of $\mathbf{N}$)

We first denote

$$\boldsymbol{\Psi} = -\left\lfloor \left(a_T \mathbf{e}_1 + \frac{\mathbf{f}_a}{m}\right)\right\rfloor + \frac{1}{m}\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}}\lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor \tag{78}$$

With (20), we have

$$a_T \mathbf{e}_1 + \frac{\mathbf{f}_a}{m} = \|\dot{\mathbf{v}} - \mathbf{g}\| \begin{bmatrix} \cos(\gamma - \alpha) & 0 & -\sin(\gamma - \alpha) \end{bmatrix}^T \tag{79}$$

With (26) and (11), we have

$$\begin{aligned} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}}\lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor &= \frac{\rho S}{2}\left(-\frac{\partial \mathbf{c}}{\partial \alpha}\mathbf{e}_2^T \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor^2 + V\frac{\partial \mathbf{c}}{\partial \beta}\mathbf{e}_2^T \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor\right) \\ &= \frac{\rho S V^2}{2}\begin{bmatrix} 0 & \frac{\partial \mathbf{c}_x}{\partial \alpha} & 0 \\ \frac{\partial \mathbf{c}_y}{\partial \beta}\sin\alpha & 0 & -\frac{\partial \mathbf{c}_y}{\partial \beta}\cos\alpha \\ 0 & \frac{\partial \mathbf{c}_z}{\partial \alpha} & 0 \end{bmatrix} \end{aligned} \tag{80}$$

Therefore, combining (79) and (80), (78) can be rewritten as

$$\boldsymbol{\Psi} = \begin{bmatrix} 0 & \psi_{12} & 0 \\ \psi_{21} & 0 & \psi_{23} \\ 0 & \psi_{32} & 0 \end{bmatrix} \tag{81}$$

where

$$\psi_{12} = -\|\dot{\mathbf{v}} - \mathbf{g}\|\sin(\gamma - \alpha) + \frac{\rho S V^2}{2m}\frac{\partial \mathbf{c}_x}{\partial \alpha} \tag{82a}$$

$$\psi_{21} = \|\dot{\mathbf{v}} - \mathbf{g}\|\sin(\gamma - \alpha) + \frac{\rho S V^2}{2m}\frac{\partial \mathbf{c}_y}{\partial \beta}\sin\alpha \tag{82b}$$

$$\psi_{23} = \|\dot{\mathbf{v}} - \mathbf{g}\|\cos(\gamma - \alpha) - \frac{\rho S V^2}{2m}\frac{\partial \mathbf{c}_y}{\partial \beta}\cos\alpha \tag{82c}$$

$$\psi_{32} = -\|\dot{\mathbf{v}} - \mathbf{g}\|\cos(\gamma - \alpha) + \frac{\rho S V^2}{2m}\frac{\partial \mathbf{c}_z}{\partial \alpha} \tag{82d}$$

Now we calculate the determinant of $\mathbf{N}$. With (30c) and (30d), $\mathbf{N}$ can be factorized as

$$\mathbf{N} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & \mathbf{v}_a^{\mathcal{B}^T}\lfloor \mathbf{e}_2 \rfloor \\ \mathbf{e}_1 & \boldsymbol{\Psi} \end{bmatrix}}_{\bar{\mathbf{N}}} \tag{83}$$

which implies $\det(\mathbf{N}) = \det(\bar{\mathbf{N}})$. Performing elementary row and column operations on $\bar{\mathbf{N}}$ produces

$$\bar{\mathbf{N}} = \begin{bmatrix} 0 & -\mathbf{v}_{a_z}^{\mathcal{B}} & 0 & \mathbf{v}_{a_x}^{\mathcal{B}} \\ 1 & 0 & \psi_{12} & 0 \\ 0 & \psi_{21} & 0 & \psi_{23} \\ 0 & 0 & \psi_{32} & 0 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \psi_{32} & 0 & 0 \\ 0 & 0 & -\mathbf{v}_{a_z}^{\mathcal{B}} & \mathbf{v}_{a_x}^{\mathcal{B}} \\ 0 & 0 & \psi_{21} & \psi_{23} \end{bmatrix} \tag{84}$$

By substituting (82) into (84), the determinant of $\mathbf{N}$ hence can be calculated as follows:

$$\begin{aligned} \det(\mathbf{N}) = \det(\bar{\mathbf{N}}) &= -\psi_{32}\left(\mathbf{v}_{a_z}^{\mathcal{B}}\psi_{23} + \mathbf{v}_{a_x}^{\mathcal{B}}\psi_{21}\right) \\ &= -\psi_{32}\|\mathbf{v}_a\|\left(\psi_{23}\sin\alpha + \psi_{21}\cos\alpha\right) \\ &= -\psi_{32}\|\mathbf{v}_a\|\left(\|\dot{\mathbf{v}} - \mathbf{g}\|\cos(\gamma - \alpha)\sin\alpha\right. \\ &\quad \left. + \|\dot{\mathbf{v}} - \mathbf{g}\|\sin(\gamma - \alpha)\cos\alpha\right) \\ &= -\psi_{32}\|\mathbf{v}_a\|\|\dot{\mathbf{v}} - \mathbf{g}\|\sin\gamma \\ &= -\psi_{32}\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| \end{aligned} \tag{85}$$

It is noted that the derivative of (22) w.r.t. α is given as

$$\begin{aligned} \frac{\partial F(\alpha)}{\partial \alpha} &= -\frac{2m\|\dot{\mathbf{v}} - \mathbf{g}\|}{\rho S V^2}\cos(\gamma - \alpha) + \frac{\mathbf{c}_z(\alpha, 0)}{\partial \alpha} \\ &= \frac{2m}{\rho S V^2}\psi_{32} \end{aligned} \tag{86}$$

Therefore, the determinant of $\mathbf{N}$ is finally arrived at

$$\det(\mathbf{N}) = -\frac{\rho S V^2}{2m}\frac{\partial F(\alpha)}{\partial \alpha}\|\mathbf{v}_a \times (\dot{\mathbf{v}} - \mathbf{g})\| \tag{87}$$

Appendix E Singularity $\|\mathbf{v}_a\| = 0$

E.1 Proof of Theorem 4: determinant of $\mathbf{N}$

With (39c) and (39d), $\mathbf{N}$ can be factorized as

$$\mathbf{N} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\|\mathbf{e}_1^T \\ \mathbf{e}_1 & -a_T\lfloor \mathbf{e}_1 \rfloor \end{bmatrix}}_{\bar{\mathbf{N}}} \tag{88}$$

which implies $\det(\mathbf{N}) = \det(\bar{\mathbf{N}})$. Performing elementary row and column operations on $\bar{\mathbf{N}}$ produces

$$\bar{\mathbf{N}} \sim \operatorname{diag}\left(\begin{bmatrix} 1 & \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| & a_T & -a_T \end{bmatrix}\right) \tag{89}$$

Hence, the determinant of $\mathbf{N}$ can be calculated as:

$$\det(\mathbf{N}) = -a_T^2\|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \tag{90}$$

E.2 Calculation of $\dot{\mathbf{h}}$ and $\dot{\mathbf{N}}$

As $\mathbf{h}$ and $\mathbf{N}$ break into block matrices, their derivatives $\dot{\mathbf{h}}$ and $\dot{\mathbf{N}}$ can be presented as like (75), where each block is calculated as follows:

$$\dot{\mathbf{h}}_1 = \left(\lfloor \mathbf{z}_b^{\text{fix}} \rfloor \ddot{\mathbf{v}}\right)^T \mathbf{z}_b + \left(\lfloor \mathbf{z}_b^{\text{fix}} \rfloor \ddot{\mathbf{v}}\right)^T \mathbf{R}\lfloor \boldsymbol{\omega} \rfloor \mathbf{e}_3 \tag{91a}$$

$$\dot{\mathbf{h}}_2 = \dddot{\mathbf{v}} \tag{91b}$$

$$\dot{\mathbf{N}}_1 = \begin{bmatrix} 0 & \frac{-(\dot{\mathbf{v}}-\mathbf{g})^T \lfloor \mathbf{z}_b^{\text{fix}} \rfloor^2 \ddot{\mathbf{v}} \mathbf{e}_1^T}{\|\lfloor \mathbf{z}_b^{\text{fix}} \rfloor (\dot{\mathbf{v}}-\mathbf{g})\|} \end{bmatrix} \tag{91c}$$

$$\dot{\mathbf{N}}_2 = \begin{bmatrix} \mathbf{R}\lfloor \boldsymbol{\omega} \rfloor \mathbf{e}_1 & -\left(\frac{(\dot{\mathbf{v}}-\mathbf{g})^T \ddot{\mathbf{v}}}{a_T}\mathbf{R} + a_T \mathbf{R}\lfloor \boldsymbol{\omega} \rfloor\right) \lfloor \mathbf{e}_1 \rfloor \end{bmatrix} \tag{91d}$$

Appendix F Singularity $\gamma = 0, \pm\pi$

F.1 Proof of Theorem 5: determinant of $\mathbf{N}$

Because $\mathbf{y}_b$ is perpendicular to $\mathbf{v}_a$, so it still holds the lateral airspeed condition $\mathbf{y}_b^T \mathbf{v}_a = 0$. We can leverage the results in (81), (82) and (86) in Appendix D to factorize $\det(\mathbf{N})$ as follows:

$$\mathbf{N} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \mathbf{e}_1^T \\ \mathbf{e}_1 & \boldsymbol{\Psi} \end{bmatrix}}_{\bar{\mathbf{N}}} \tag{92}$$

which implies $\det(\mathbf{N}) = \det(\bar{\mathbf{N}})$. Performing elementary row and column operations on $\bar{\mathbf{N}}$ produces

$$\bar{\mathbf{N}} \sim \operatorname{diag}\left(\begin{bmatrix} 1 & \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| & \psi_{23} & \psi_{32} \end{bmatrix}\right) \tag{93}$$

where $\psi_{23} = \left(\|\dot{\mathbf{v}} - \mathbf{g}\| - \frac{\rho S V^2}{2m} \frac{\partial \mathbf{c}_y}{\partial \beta}\right) \cos\alpha$ is from (82c) by setting $\gamma = 0$, and $\psi_{32} = \frac{\rho S V^2}{2m} \frac{\partial F(\alpha)}{\partial \alpha}$ is from (86). Finally, the determinant of $\mathbf{N}$ is

$$\det(\mathbf{N}) = \frac{\rho S V^2}{2m} \frac{\partial F(\alpha)}{\partial \alpha} \|\mathbf{z}_b^{\text{fix}} \times (\dot{\mathbf{v}} - \mathbf{g})\| \psi_{23} \tag{94}$$

F.2 Calculation of $\dot{\mathbf{h}}$ and $\dot{\mathbf{N}}$

Similarly, the derivatives $\dot{\mathbf{h}}$ and $\dot{\mathbf{N}}$ can be presented as like (75), in which block matrices $\dot{\mathbf{h}}_1$ and $\dot{\mathbf{N}}_1$ are given by (91a) and (91c) in Appendix E.2, while $\dot{\mathbf{h}}_2$ and $\dot{\mathbf{N}}_2$ are given by (76).

Appendix G Gradients of the flatness functions

We denote $\mathcal{P} = \begin{bmatrix} \mathbf{v}^T & \dot{\mathbf{v}}^T & \ddot{\mathbf{v}}^T \end{bmatrix}^T$, $\mathbf{b} = \begin{bmatrix} \dot{a}_T & \boldsymbol{\omega}^T \end{bmatrix}^T$ for simplicity. We also split the matrix $\mathbf{h}$ and $\mathbf{N}_2$ in (30), (39), (41) into $\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2^T \end{bmatrix}^T$ and $\mathbf{N}_2 = \begin{bmatrix} \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix}$, respectively.

G.1 When in coordinated flight

The flatness functions are presented in Section 4.2, and the corresponding gradients are given as follows:

$$\begin{aligned} \frac{\partial a_T}{\partial \mathcal{P}} &= \frac{\partial \|\dot{\mathbf{v}} - \mathbf{g}\| \cos(\gamma - \alpha) - \mathbf{f}_{a_x}/m}{\partial \mathcal{P}} \\ &= \frac{\partial \|\dot{\mathbf{v}} - \mathbf{g}\|}{\partial \mathcal{P}} \cos(\gamma - \alpha) - \|\dot{\mathbf{v}} - \mathbf{g}\| \sin(\gamma - \alpha) \\ &\quad \left(\frac{\partial \gamma}{\partial \mathcal{P}} - \frac{\partial \alpha}{\partial \mathcal{P}}\right) - \frac{\mathbf{e}_1^T}{m} \frac{\partial \mathbf{f}_a}{\partial \mathcal{P}} \end{aligned} \tag{95a}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial \mathcal{P}} = \begin{bmatrix} \mathbf{0}_{3\times 1} & \mathbf{I}_3 \end{bmatrix} \mathbf{N}^{-1} \left(\frac{\partial \mathbf{h}}{\partial \mathcal{P}} - \frac{\partial \mathbf{N}}{\partial \mathcal{P}} \mathbf{b}\right) \tag{95b}$$

where

$$\frac{\partial \|\dot{\mathbf{v}} - \mathbf{g}\|}{\partial \mathcal{P}} = \frac{(\dot{\mathbf{v}} - \mathbf{g})^T}{\|\dot{\mathbf{v}} - \mathbf{g}\|} \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \tag{96a}$$

$$\frac{\partial \mathbf{f}_a}{\partial \mathcal{P}} = \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \frac{\partial \mathbf{v}_a^{\mathcal{B}}}{\partial \mathcal{P}} \tag{96b}$$

$$\begin{aligned} \frac{\partial \gamma}{\partial \mathcal{P}} &= r \frac{\partial}{\partial \mathcal{P}} \operatorname{atan2}\left(\|(\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a\|, (\dot{\mathbf{v}} - \mathbf{g})^T \mathbf{v}_a\right) \\ &= \frac{r}{\|(\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a\|^2 + \left((\dot{\mathbf{v}} - \mathbf{g})^T \mathbf{v}_a\right)^2} \\ &\quad \left(\left(\frac{(\dot{\mathbf{v}} - \mathbf{g})^T \mathbf{v}_a ((\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a)^T \lfloor \mathbf{v}_a \rfloor}{\|(\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a\|} \right.\right. \\ &\quad \left.\left. - \|(\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a\| \mathbf{v}_a^T \right) \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \right. \\ &\quad \left(\frac{(\dot{\mathbf{v}} - \mathbf{g})^T \mathbf{v}_a ((\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a)^T \lfloor \dot{\mathbf{v}} - \mathbf{g} \rfloor}{\|(\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a\|} \right. \\ &\quad \left. - \|(\dot{\mathbf{v}} - \mathbf{g}) \times \mathbf{v}_a\| (\dot{\mathbf{v}} - \mathbf{g})^T \right) \frac{\partial \mathbf{v}}{\partial \mathcal{P}} \Big) \\ &= \frac{r}{|\sin\gamma|} \left(\left(\frac{\cos\gamma (\dot{\mathbf{v}} - \mathbf{g})^T}{\|\dot{\mathbf{v}} - \mathbf{g}\|^2} - \frac{\mathbf{v}_a^T}{\|\dot{\mathbf{v}} - \mathbf{g}\| \|\mathbf{v}_a\|} \right) \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \right. \\ &\quad \left. + \left(\frac{\cos\gamma \mathbf{v}_a^T}{\|\mathbf{v}_a\|^2} - \frac{(\dot{\mathbf{v}} - \mathbf{g})^T}{\|\dot{\mathbf{v}} - \mathbf{g}\| \|\mathbf{v}_a\|} \right) \frac{\partial \mathbf{v}}{\partial \mathcal{P}} \right) \end{aligned} \tag{96c}$$

$$\frac{\partial \alpha}{\partial \mathcal{P}} = \frac{\frac{\partial h}{\partial \mathcal{P}} \sin(\gamma - \alpha) + h \cos(\gamma - \alpha) \frac{\partial \gamma}{\partial \mathcal{P}}}{h \cos(\gamma - \alpha) - \frac{\partial \mathbf{c}_z}{\partial \alpha}} \tag{96d}$$

$$\frac{\partial \alpha}{\partial \mathcal{P}} = \frac{\frac{\partial h}{\partial \mathcal{P}} \sin(\gamma - \alpha) + h \cos(\gamma - \alpha) \frac{\partial \gamma}{\partial \mathcal{P}}}{h \cos(\gamma - \alpha) - \frac{\partial \mathbf{c}_z}{\partial \alpha}} \tag{96e}$$

$$\frac{\partial h}{\partial \mathcal{P}} = \frac{2m}{\rho S} \left(\frac{(\dot{\mathbf{v}} - \mathbf{g})^T}{\|\dot{\mathbf{v}} - \mathbf{g}\| \|\mathbf{v}_a\|^2} \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} - \frac{2\|\dot{\mathbf{v}} - \mathbf{g}\| \mathbf{v}_a^T}{\|\mathbf{v}_a\|^4} \frac{\partial \mathbf{v}}{\partial \mathcal{P}} \right) \tag{96f}$$

$$\frac{\partial \mathbf{h}}{\partial \mathcal{P}} = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \mathcal{P}}^T & \frac{\partial \mathbf{h}_2}{\partial \mathcal{P}}^T \end{bmatrix}^T \tag{96g}$$

$$\frac{\partial \mathbf{h}_1}{\partial \mathcal{P}} = \dot{\mathbf{v}}_a^T \frac{\partial \mathbf{y}_b}{\partial \mathcal{P}} + \mathbf{y}_b^T \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \tag{96h}$$

$$\begin{aligned} \frac{\partial \mathbf{h}_2}{\partial \mathcal{P}} &= \frac{\partial \ddot{\mathbf{v}}}{\partial \mathcal{P}} - \frac{1}{m} \left(\frac{\partial \mathbf{R}\boldsymbol{\xi}}{\partial \mathcal{P}} \Big|_{\boldsymbol{\xi} = \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \mathbf{R}^T \dot{\mathbf{v}}_a} + \mathbf{R} \frac{\partial \left(\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \boldsymbol{\xi} \right)}{\partial \mathcal{P}} \Big|_{\boldsymbol{\xi} = \mathbf{R}^T \dot{\mathbf{v}}_a} \right. \\ &\quad \left. + \mathbf{R} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \frac{\partial \mathbf{R}^T \boldsymbol{\xi}}{\partial \mathcal{P}} \Big|_{\boldsymbol{\xi} = \dot{\mathbf{v}}_a} + \mathbf{R} \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \mathbf{R}^T \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \right) \end{aligned} \tag{96i}$$

$$\frac{\partial \mathbf{R}\boldsymbol{\xi}}{\partial \mathcal{P}} = \boldsymbol{\xi}_1 \frac{\partial \mathbf{x}_b}{\partial \mathcal{P}} + \boldsymbol{\xi}_2 \frac{\partial \mathbf{y}_b}{\partial \mathcal{P}} + \boldsymbol{\xi}_3 \frac{\partial \mathbf{z}_b}{\partial \mathcal{P}} \tag{96j}$$

$$\frac{\partial \mathbf{R}^T \boldsymbol{\xi}}{\partial \mathcal{P}} = \begin{bmatrix} \left(\boldsymbol{\xi}^T \frac{\partial \mathbf{x}_b}{\partial \mathcal{P}}\right)^T & \left(\boldsymbol{\xi}^T \frac{\partial \mathbf{y}_b}{\partial \mathcal{P}}\right)^T & \left(\boldsymbol{\xi}^T \frac{\partial \mathbf{z}_b}{\partial \mathcal{P}}\right)^T \end{bmatrix}^T \tag{96k}$$

$$\begin{aligned} \frac{\partial \mathbf{y}_b}{\partial \mathcal{P}} &= r \frac{\|\lfloor \mathbf{v}_a \rfloor (\dot{\mathbf{v}} - \mathbf{g})\|^2 \mathbf{I}_3 - \lfloor \mathbf{v}_a \rfloor (\dot{\mathbf{v}} - \mathbf{g}) (\dot{\mathbf{v}} - \mathbf{g})^T \lfloor \mathbf{v}_a \rfloor}{\|\lfloor \mathbf{v}_a \rfloor (\dot{\mathbf{v}} - \mathbf{g})\|^3} \\ &\quad \frac{\partial \left(\lfloor \mathbf{v}_a \rfloor (\dot{\mathbf{v}} - \mathbf{g})\right)}{\partial \mathcal{P}} \end{aligned} \tag{96l}$$

$$\frac{\partial \left(\lfloor \mathbf{v}_a \rfloor (\dot{\mathbf{v}} - \mathbf{g})\right)}{\partial \mathcal{P}} = -\lfloor \dot{\mathbf{v}} - \mathbf{g} \rfloor \frac{\partial \mathbf{v}}{\partial \mathcal{P}} + \lfloor \mathbf{v}_a \rfloor \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \tag{96m}$$

$$\frac{\partial \mathbf{x}_b}{\partial \mathcal{P}} = \lfloor \mathbf{y}_b \rfloor \mathrm{Exp}(\alpha \mathbf{y}_b) \frac{\mathbf{v}_a}{\|\mathbf{v}_a\|} \frac{\partial \alpha}{\partial \mathcal{P}} + \left. \frac{\partial \mathrm{Exp}(\alpha \mathbf{y}_b)\boldsymbol{\xi}}{\partial \mathbf{y}_b} \right|_{\boldsymbol{\xi}=\frac{\mathbf{v}_a}{\|\mathbf{v}_a\|}} + \mathrm{Exp}(\alpha \mathbf{y}_b) \frac{\|\mathbf{v}_a\|^2 \mathbf{I}_3 - \mathbf{v}_a \mathbf{v}_a^T}{\|\mathbf{v}_a\|^3} \frac{\partial \mathbf{v}}{\partial \mathcal{P}} \tag{96n}$$

$$\frac{\partial \mathrm{Exp}(\alpha \mathbf{y}_b)\boldsymbol{\xi}}{\partial \mathbf{y}_b} = \frac{\partial}{\partial \mathbf{y}_b} \left(\mathbf{I}_3 + \lfloor \mathbf{y}_b \rfloor \sin\alpha + \lfloor \mathbf{y}_b \rfloor^2 (1-\cos\alpha) \right) \boldsymbol{\xi} = -\lfloor \boldsymbol{\xi} \rfloor \sin\alpha - \left(\lfloor \lfloor \mathbf{y}_b \rfloor \boldsymbol{\xi} \rfloor + \lfloor \mathbf{y}_b \rfloor \lfloor \boldsymbol{\xi} \rfloor \right) (1-\cos\alpha) \tag{96o}$$

$$\frac{\partial \mathbf{z}_b}{\mathcal{P}} = \lfloor \mathbf{x}_b \rfloor \frac{\partial \mathbf{y}_b}{\partial \mathcal{P}} - \lfloor \mathbf{y}_b \rfloor \frac{\partial \mathbf{x}_b}{\partial \mathcal{P}} \tag{96p}$$

$$\frac{\partial}{\partial \mathcal{P}} \left(\frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \boldsymbol{\xi} \right) = \frac{\rho S}{2} \left(\left(2 \frac{\partial \mathbf{c}}{\alpha} \mathbf{v}_a^{\mathcal{B}^T} + \frac{\partial^2 \mathbf{c}}{\partial \alpha^2} \mathbf{v}_a^{\mathcal{B}^T} \lfloor \mathbf{e}_2 \rfloor + V \frac{\partial^2 \mathbf{c}}{\partial \beta \partial \alpha} \mathbf{e}_2^T \right) \boldsymbol{\xi} \frac{\partial \alpha}{\partial \mathcal{P}} + \left(2\mathbf{c}\boldsymbol{\xi}^T - \frac{\partial \mathbf{c}}{\partial \alpha} \boldsymbol{\xi}^T \lfloor \mathbf{e}_2 \rfloor + \frac{\partial \mathbf{c}}{\partial \beta} \frac{\mathbf{e}_2^T \boldsymbol{\xi} \mathbf{v}_a^{\mathcal{B}^T}}{V} \right) \frac{\partial \mathbf{v}_a^{\mathcal{B}}}{\partial \mathcal{P}} \right) \tag{96q}$$

$$\frac{\partial \mathbf{v}_a^{\mathcal{B}}}{\partial \mathcal{P}} = \left. \frac{\partial \mathbf{R}^T \boldsymbol{\xi}}{\partial \mathcal{P}} \right|_{\boldsymbol{\xi}=\mathbf{v}_a} + \mathbf{R}^T \frac{\partial \mathbf{v}_a}{\partial \mathcal{P}} \tag{96r}$$

$$\frac{\partial \mathbf{N}\mathbf{b}}{\partial \mathcal{P}} = \begin{bmatrix} \mathbf{b}^T \frac{\partial \mathbf{N}_1^T}{\partial \mathcal{P}} \\ \mathbf{b}_1 \frac{\partial \mathbf{N}_{21}}{\partial \mathcal{P}} + \sum_{i=2}^{4} \left(\mathbf{b}_i \frac{\partial \mathbf{N}_{22}\mathbf{e}_{i-1}}{\partial \mathcal{P}} \right) \end{bmatrix} \tag{96s}$$

$$\frac{\partial \mathbf{N}_1^T}{\partial \mathcal{P}} = \begin{bmatrix} \mathbf{0} & \left(-\lfloor \mathbf{e}_2 \rfloor \frac{\partial \mathbf{v}_a^{\mathcal{B}}}{\partial \mathcal{P}} \right)^T \end{bmatrix}^T \tag{96t}$$

$$\frac{\partial \mathbf{N}_{21}}{\partial \mathcal{P}} = \frac{\partial \mathbf{x}_b}{\partial \mathcal{P}} \tag{96u}$$

$$\frac{\partial \mathbf{N}_{22}\mathbf{e}_j}{\partial \mathcal{P}} = \frac{\partial \mathbf{R}}{\partial \mathcal{P}} \mathbf{R}^T \mathbf{N}_{22}\mathbf{e}_j + \mathbf{R} \left(-\lfloor \mathbf{e}_1 \rfloor \mathbf{e}_j \frac{\partial a_T}{\partial \mathcal{P}} + \frac{1}{m} \left(\lfloor \mathbf{e}_j \rfloor \frac{\partial \mathbf{f}_a}{\partial \mathcal{P}} + \frac{\partial^2 \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}} \partial \mathcal{P}} \lfloor \mathbf{v}_a^{\mathcal{B}} \rfloor \mathbf{e}_j - \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \lfloor \mathbf{e}_j \rfloor \frac{\partial \mathbf{v}_a^{\mathcal{B}}}{\partial \mathcal{P}} \right) \right) \tag{96v}$$

G.2 When in singularity condition $\|\mathbf{v}_a\| = 0$

The flatness functions are rewritten in Section 4.3.2 when $\|\mathbf{v}_a\| = 0$. The corresponding modified gradients that are different from Appendix G.1 are given as follows:

$$\frac{\partial a_T}{\partial \mathcal{P}} = \frac{\partial \|\dot{\mathbf{v}} - \mathbf{g}\|}{\partial \mathcal{P}} \tag{97a}$$

$$\frac{\partial \mathbf{x}_b}{\partial \mathcal{P}} = \frac{\|\dot{\mathbf{v}} - \mathbf{g}\|^2 \mathbf{I}_3 - (\dot{\mathbf{v}} - \mathbf{g})(\dot{\mathbf{v}} - \mathbf{g})^T}{\|\dot{\mathbf{v}} - \mathbf{g}\|^3} \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \tag{97b}$$

$$\frac{\partial \mathbf{y}_b}{\partial \mathcal{P}} = \frac{\|\lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor (\dot{\mathbf{v}} - \mathbf{g})\|^2 \mathbf{I}_3 - (\lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor (\dot{\mathbf{v}} - \mathbf{g}))(\lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor (\dot{\mathbf{v}} - \mathbf{g}))^T}{\|\lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor (\dot{\mathbf{v}} - \mathbf{g})\|^3} \lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \tag{97c}$$

$$\frac{\partial \mathbf{h}_1}{\partial \mathcal{P}} = \left(\lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor \ddot{\mathbf{v}} \right)^T \frac{\partial \mathbf{z}_b}{\partial \mathcal{P}} + \mathbf{z}_b^T \lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor \frac{\partial \ddot{\mathbf{v}}}{\partial \mathcal{P}} \tag{97d}$$

$$\frac{\partial \mathbf{h}_2}{\partial \mathcal{P}} = \frac{\partial \ddot{\mathbf{v}}}{\partial \mathcal{P}} \tag{97e}$$

$$\frac{\partial \mathbf{N}_1^T}{\partial \mathcal{P}} \begin{bmatrix} \mathbf{0} & \left(-\mathbf{e}_1 \frac{(\dot{\mathbf{v}} - \mathbf{g})^T \lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor^2}{\|\lfloor \mathbf{z}_b^{\mathrm{fix}} \rfloor (\dot{\mathbf{v}} - \mathbf{g})\|} \frac{\partial \dot{\mathbf{v}}}{\partial \mathcal{P}} \right)^T \end{bmatrix}^T \tag{97f}$$

$$\frac{\partial \mathbf{N}_{22}\mathbf{e}_j}{\partial \mathcal{P}} = -a_T \frac{\partial \mathbf{R}}{\partial \mathcal{P}} \lfloor \mathbf{e}_1 \rfloor \mathbf{e}_j + \mathbf{R} \lfloor \mathbf{e}_1 \rfloor \mathbf{e}_j \frac{\partial a_T}{\partial \mathcal{P}} \tag{97g}$$

G.3 When in singularity condition $\gamma = 0, \pm\pi$

The flatness functions are rewritten in Section 4.3.2 when $|\gamma| = 0$. Since a_T, $\mathbf{h}_2$ and $\mathbf{N}_2$ are the same as those presented in Section 4.2, while $\mathbf{h}_1$ and $\mathbf{N}_1$ are the same as those in Section 4.3.2, their gradients are the identical to those given respectively in Appendix G.1 and G.2.

Appendix H Proof of Theorem 6 (the error-state dynamics)

The dynamics of (56b) and (56c) simply take the time derivative to (54b) and (54c), respectively. Denoting $\boldsymbol{\theta} = \mathrm{Log}(\mathbf{R})$, the exponential map holds $\dot{\boldsymbol{\theta}} = \mathbf{A}^T(\boldsymbol{\theta})\boldsymbol{\omega}$, where $\boldsymbol{\omega} = (\mathbf{R}^T\dot{\mathbf{R}})^\vee$, $(\cdot)^\vee$ the inverse of $\lfloor\cdot\rfloor$ that maps a skew-symmetric matrix to a vector, and $\mathbf{A}(\cdot)$ denotes the Jacobian of the exponential coordinates of $SO(3)$ (Bullo and Murray 1995):

$$\mathbf{A}(\boldsymbol{\theta}) = \mathbf{I}_3 + \left(\frac{1-\cos\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|} \right) \frac{\lfloor \boldsymbol{\theta} \rfloor}{\|\boldsymbol{\theta}\|} + \left(1 - \frac{\sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|} \right) \frac{\lfloor \boldsymbol{\theta} \rfloor^2}{\|\boldsymbol{\theta}\|^2} \tag{98}$$

By substituting (54b) into the above rules, we have

$$\begin{aligned} \delta\dot{\boldsymbol{\theta}} &= \mathbf{A}^T(\delta\boldsymbol{\theta}) \left(\left(\mathbf{R}^T\mathbf{R}_d \right)^T \frac{d}{dt} \left(\mathbf{R}^T\mathbf{R}_d \right) \right)^\vee \\ &= \mathbf{A}^T(\delta\boldsymbol{\theta}) \left(\mathbf{R}_d^T\mathbf{R} \left(-\lfloor \boldsymbol{\omega} \rfloor \mathbf{R}^T\mathbf{R}_d + \mathbf{R}^T\mathbf{R}_d \lfloor \boldsymbol{\omega}_d \rfloor \right) \right)^\vee \\ &= \mathbf{A}^T(\delta\boldsymbol{\theta}) \left(-\mathbf{R}_d^T\mathbf{R}\boldsymbol{\omega} + \boldsymbol{\omega}_d \right) \end{aligned} \tag{99}$$

which is the error attitude dynamics in (56d).

Appendix I Proof of Lemma 1 (the linearized error-state dynamics)

The position error dynamics in (56b) is linear, and the velocity error dynamics in (56c) can be linearized along the reference trajectory. Specifically, since $\mathbf{R}^T\mathbf{R}_d = \mathrm{Exp}(\delta\boldsymbol{\theta}) \approx \mathbf{I}_3 + \lfloor \delta\boldsymbol{\theta} \rfloor$, (54) implies:

$$\begin{aligned} \delta\dot{\mathbf{v}} &= \left(a_{T_d}\mathbf{R}_d\mathbf{e}_1 + \frac{1}{m}\mathbf{R}_d\mathbf{f}_{a_d} \right) - \left(a_T\mathbf{R}\mathbf{e}_1 + \frac{1}{m}\mathbf{R}\mathbf{f}_a \right) \\ &\approx \left(a_{T_d}\mathbf{R}_d - (a_{T_d} - \delta a_T)\mathbf{R}_d(\mathbf{I}_3 + \lfloor \delta\boldsymbol{\theta} \rfloor)^T \right) \mathbf{e}_1 \\ &\quad + \frac{1}{m}\mathbf{R}_d \left(\mathbf{f}_{a_d} - (\mathbf{I}_3 + \lfloor \delta\boldsymbol{\theta} \rfloor)^T (\mathbf{f}_{a_d} - \delta\mathbf{f}_a) \right) \\ &\approx \mathbf{R}_d\mathbf{e}_1\delta a_T - \mathbf{R}_d \left(a_{T_d} \lfloor \mathbf{e}_1 \rfloor + \lfloor \frac{\mathbf{f}_{a_d}}{m} \rfloor \right) \delta\boldsymbol{\theta} + \mathbf{R}_d \frac{\delta\mathbf{f}_a}{m} \end{aligned} \tag{100}$$

where

$$\delta\mathbf{f}_a = \mathbf{f}_{a_d} - \mathbf{f}_a \approx \frac{\partial \mathbf{f}_{a_d}}{\partial \mathbf{v}_{a_d}^{\mathcal{B}}} \delta\mathbf{v}_a^{\mathcal{B}} \tag{101a}$$

$$\begin{aligned} \delta\mathbf{v}_a^{\mathcal{B}} &= \mathbf{v}_{a_d}^{\mathcal{B}} - \mathbf{v}_a^{\mathcal{B}} = \mathbf{R}_d^T\mathbf{v}_{a_d} - \mathbf{R}^T\mathbf{v}_a \\ &\approx \mathbf{R}_d^T\mathbf{v}_{a_d} - (\mathbf{I}_3 + \lfloor \delta\boldsymbol{\theta} \rfloor)\mathbf{R}_d^T(\mathbf{v}_{a_d} - \delta\mathbf{v}_a) \\ &\approx -\lfloor \delta\boldsymbol{\theta} \rfloor \mathbf{R}_d^T\mathbf{v}_{a_d} + \mathbf{R}_d^T\delta\mathbf{v}_a \\ &\approx \lfloor \mathbf{v}_{a_d}^{\mathcal{B}} \rfloor \delta\boldsymbol{\theta} + \mathbf{R}_d^T\delta\mathbf{v}_a \end{aligned} \tag{101b}$$

and the partial derivative $\partial\mathbf{f}_{a_d}/\partial\mathbf{v}_{a_d}^{\mathcal{B}}$ in (101a) is given in (74):

$$\frac{\partial \mathbf{f}_{a_d}}{\partial \mathbf{v}_{a_d}^{\mathcal{B}}} = \left. \frac{\partial \mathbf{f}_a}{\partial \mathbf{v}_a^{\mathcal{B}}} \right|_{\mathbf{v}_{a_d}^{\mathcal{B}}} \quad \text{(see Equation (74))} \tag{102}$$

In (101b), $\mathbf{v}_a = \mathbf{v} - \mathbf{w}$ is the actual air velocity and $\mathbf{v}_{a_d} = \mathbf{v}_d - \bar{\mathbf{w}}$ is the air velocity used to calculate the reference trajectory. Hence,

$$\delta\mathbf{v}_a = \mathbf{v}_a - \mathbf{v}_{a_d} = \delta\mathbf{v} - \delta\mathbf{w}; \quad \delta\mathbf{w} = \mathbf{w} - \bar{\mathbf{w}} \tag{103a}$$

and

$$\delta\mathbf{f}_a \approx \frac{\partial \mathbf{f}_{a_d}}{\partial \mathbf{v}^{\mathcal{B}}_{a_d}} \left(\lfloor \mathbf{v}^{\mathcal{B}}_{a_d} \rfloor \delta\boldsymbol{\theta} + \mathbf{R}_d^T (\delta\mathbf{v} - \delta\mathbf{w}) \right) \tag{104a}$$

By substituting (101a) and (104) into (100), the velocity error dynamics can be given by

$$\delta\dot{\mathbf{v}} = \mathbf{M}_T \delta a_T + \mathbf{M}_{\mathbf{v}} \delta\mathbf{v} + \mathbf{M}_{\mathbf{R}} \delta\boldsymbol{\theta} + \mathbf{M}_{\mathbf{w}} \mathbf{w} \tag{105}$$

where

$$\mathbf{M}_T = \mathbf{R}_d \mathbf{e}_1 \tag{106a}$$

$$\mathbf{M}_{\mathbf{v}} = \frac{1}{m} \mathbf{R}_d \frac{\partial \mathbf{f}_{a_d}}{\partial \mathbf{v}^{\mathcal{B}}_{a_d}} \mathbf{R}_d^T \tag{106b}$$

$$\mathbf{M}_{\mathbf{R}} = \mathbf{R}_d \left(-a_{T_d} \lfloor \mathbf{e}_1 \rfloor - \lfloor \frac{\mathbf{f}_{a_d}}{m} \rfloor + \frac{\partial \mathbf{f}_{a_d}}{\partial \mathbf{v}^{\mathcal{B}}_{a_d}} \lfloor \frac{\mathbf{v}^{\mathcal{B}}_{a_d}}{m} \rfloor \right) \tag{106c}$$

$$\mathbf{M}_{\mathbf{w}} = -\frac{1}{m} \mathbf{R}_d \frac{\partial \mathbf{f}_{a_d}}{\partial \mathbf{v}^{\mathcal{B}}_{a_d}} \mathbf{R}_d^T \tag{106d}$$

To linearize the attitude error dynamics, we substitute (54d) in (56d) and approximate $\mathbf{A}(\delta\mathbf{R}) \approx \mathbf{I}_3$. Thus we have

$$\begin{aligned} \delta\dot{\mathbf{R}} &\approx -\mathbf{R}_d^T \mathbf{R} \boldsymbol{\omega} + \boldsymbol{\omega}_d \\ &\approx -\left(\mathbf{I} + \lfloor \delta\mathbf{R} \rfloor\right)^T (\boldsymbol{\omega}_d - \delta\boldsymbol{\omega}) + \boldsymbol{\omega}_d \\ &\approx \delta\boldsymbol{\omega} - \lfloor \boldsymbol{\omega}_d \rfloor \delta\mathbf{R} \end{aligned} \tag{107}$$