

Model-Based Planning and Control for Terrestrial-Aerial Bimodal Vehicles with Passive Wheels

Ruixin Zhang, Junxiao Lin, Yuze Wu, Yuman Gao, Chi Wang, Chao Xu, Yanjun Cao, and Fei Gao

Abstract—Terrestrial and aerial bimodal vehicles have gained widespread attention due to their cross-domain maneuverability. Nevertheless, their bimodal dynamics significantly increase the complexity of motion planning and control, thus hindering robust and efficient autonomous navigation in unknown environments. To resolve this issue, we develop a model-based planning and control framework for terrestrial-aerial bimodal vehicles. This work begins by deriving a unified dynamic model and the corresponding differential flatness. Leveraging differential flatness, an optimization-based trajectory planner is proposed, which takes into account both solution quality and computational efficiency. Moreover, we design a tracking controller using nonlinear model predictive control based on the proposed unified dynamic model to achieve accurate trajectory tracking and smooth mode transition. We validate our framework through extensive benchmark comparisons and experiments, demonstrating its effectiveness in terms of planning quality and control performance.

I. INTRODUCTION

Mobile robots play a prominent role in many aspects of human society in recent years. Among them, unmanned aerial vehicles are widely used in unstructured and cluttered environments due to their high maneuverability, but they are not suitable for long-distance missions due to poor energy efficiency. In contrast, while unmanned ground vehicles enjoy much higher energy efficiency, their ability to maneuver over terrains with unavoidable obstacles is limited. In order to combine the advantages of both types of vehicles, researchers have devoted considerable effort to developing terrestrial-aerial bimodal vehicles (TABVs), which feature cross-domain locomotion [1]–[10]. Moreover, some previous works present autonomous navigation frameworks for TABVs, trying to make them competent for autonomous tasks in complex scenarios.

TABVs have different dynamic characteristics in different locomotion modes, which pose significant challenges to trajectory planning and motion control. First, regarding trajectory planning, it is vital to ensure dynamical feasibility. However, due to the TABVs' bimodal nature, it is difficult to find a proper formulation to guarantee the dynamic feasibility of both modes, as well as the continuity and smoothness during mode transition. In addition, it is also necessary to improve planning efficiency for real-time performance.

This work was supported by the Fundamental Research Funds for the Central Universities and the National Natural Science Foundation of China under grant no.62003299. All authors are with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute of Zhejiang University, Huzhou 313000, China.

Corresponding author: Fei Gao.

E-mail:{ruixin_zhang, fgaoaa}@zju.edu.cn



Fig. 1: The real-world experiments. a) An autonomous navigation test in an unknown dense environment. b) A Terrestrial-aerial hybrid trajectory tracking test in which the maximal commanded velocity and acceleration reach $3m/s$ and $2.5m/s^2$, respectively.

Previous works [11]–[16] either oversimplify the dynamics of TABV or suffer from high computational consumption. In terms of motion control, previous efforts [14, 15, 17, 18] mainly aim at developing a unified control framework for both modes. Nonetheless, the nonlinear dynamics of both modes are partially ignored for the convenience of modeling, thus restricting control performance in aggressive scenarios where the linearization or approximation substantially limits the solution space.

In our previous work [15], we propose a navigation framework for passive-wheeled TABV configurations (originally conceived by Kalantari et al. [1]), which achieves state-of-the-art planning performance and controlling accuracy. However, our previous work is based on the dynamics and differential flatness of general quadrotors, while the dynamics of the TABV in terrestrial locomotion are overly simplified. As a result, the planning results are dynamically infeasible in certain cases. Furthermore, the control error considerably increases when tracking aggressive trajectories.

To address the issues mentioned above, we propose a unified model-based planning and control framework for passive-wheeled TABVs, as illustrated in Fig.2. Chiefly, we model the nonlinear bimodal dynamics of passive-wheeled TABVs and their differential flatness in a unified manner. In trajectory planning, we present an optimization-based formulation utilizing the proposed TABV differential flatness

map. It allows the complete states and constraints of both locomotion modes to be analytically derived from the flat output. This formulation simplifies the planning problem while ensuring dynamic feasibility. Furthermore, the unified modeling naturally guarantees continuity and smoothness during mode transitions. In motion control, we propose a bimodal tracking controller based on nonlinear model predictive control (NMPC), which exploits the full capability of the TABV without violating dynamic constraints. It tracks the reference state recovered from the planned flat output while satisfying the control limits and state constraints. Seamless mode transition is achieved thanks to proposed the unified modeling. We adopt incremental nonlinear dynamic inversion (INDI) as the inner-loop angular controller for robustification.

We conduct extensive experiments and benchmark comparisons on a customized TABV platform in challenging real-world environments to verify the proposed methods. The results demonstrate that our method outperforms our previous work [15] in planning quality and control performance. The contributions of this paper are as the followings:

- 1) A unified dynamic model for a passive-wheeled TABV configuration, including the corresponding differential flatness that benefits trajectory planning and tracking control.
- 2) A flatness-based TABV trajectory planner that guarantees dynamical feasibility, continuity, and smoothness for bimodal locomotion while maintaining real-time performance.
- 3) An NMPC-based TABV tracking controller that achieves accurate and robust trajectory tracking and seamless mode transition without violating dynamic constraints.
- 4) A set of real-world tests and benchmark comparisons that validate the the planning quality and control performance of the proposed methods.

II. RELATED WORK

A. TABV Trajectory Planning

Many of the previous works on TABV trajectory planning aim to find a safe and efficient geometric path while considering the extra energy consumption in aerial locomotion [11]–[13]. However, since no high-order information such as velocity or acceleration is available from a geometric path, these methods can hardly be applied to real-world scenarios. Fan et al. [14] propose a primitive-based local planner that generates minimum-snap trajectories under the guidance of geometric path searching. Nevertheless, this method does not take the terrestrial dynamics into account, thus cannot guarantee high-fidelity feasibility. In our previous work [15], we base trajectory planning on the differential flatness of general quadrotors and handle the nonholonomic dynamics in terrestrial locomotion by constraining the trajectory curvature. However, this indirect modeling bypasses terrestrial dynamics and cannot ensure physical feasibility as well. Wu et al. [16] propose a model-based planning approach

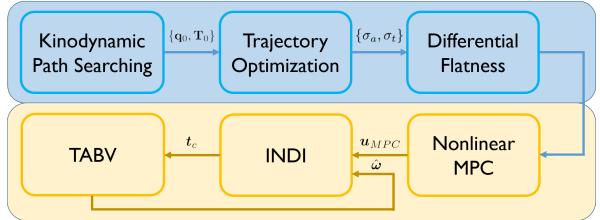


Fig. 2: A diagram of the proposed planning and control framework. that enables optimizing the full-state trajectories considering both terrestrial and aerial dynamics. The bimodal dynamics are modeled in a unified manner utilizing complementary constraints. However, this approach is time-consuming due to the integration of differential equations introduced by the nonlinear dynamics. In order to improve the computational efficiency, this method requires warm starting of the previously generated trajectories. However, as reported by Wu et al. [16], when the newly observed obstacles intersect with the previously generated trajectory which results in the failure of the warm start mechanism, the computation time greatly increases and paused behaviors happen. To sum up, the computation burden limits this approach's application in realistic settings.

In this work, we encode feasibility constraints in trajectory planning utilizing the proposed terrestrial-aerial bimodal differential flatness. The trajectory planning problem is transformed into optimizing continuous trajectories of the flat output and avoids explicit integration of system dynamics, thus ensuring both solution quality and computational efficiency.

B. TABV Tracking Control

Several approaches have been proposed to achieve bimodal trajectory tracking for passive-wheeled TABVs. A common approach is to use a cascaded control framework similar to a general quadrotor controller based on a simplified dynamic model [14, 15, 17, 18]. In such a cascaded control architecture, the resulting control command is either too conservative or too aggressive because the nonlinear dynamics are not precisely captured [19]. Moreover, the state and input limit cannot be properly handled. Atay et al. [20] derive the complete nonlinear dynamics and differential flatness of passive-wheeled TABVs and propose a model-based control system considering control allocation. However, the map from the flatness output to the system state is not considered, making this approach not applicable to trajectory tracking.

In this work, we adopt NMPC, a control method that optimizes the behavior of a nonlinear system based on the system dynamics and constraints over a receding horizon, for TABV tracking control. Leveraging the unified dynamic model (as demonstrated in Sect.III), NMPC enables both accurate trajectory tracking and smooth locomotion mode transition. Moreover, the predictive nature of NMPC allows the TABV to react in advance to aggressive maneuvers that will occur in the near future. Although NMPC is more computationally demanding than non-predictive methods, previous works have shown that it can be applied to trajectory tracking control [19, 21] even using onboard computers with relatively low computing power.

III. PRELIMINARIES

In this section, we will introduce the unified TABV dynamics model and its corresponding differential flatness. The reference frames are defined in Fig.3.

A. Unified TABV Dynamics

We define the state and input vectors of the TABV as $\mathbf{x} = [\mathbf{p}_I^T, \mathbf{q}^T, \mathbf{v}_I^T, \boldsymbol{\omega}_B^T]^T$ and $\mathbf{u} = [T, \boldsymbol{\tau}_B^T]^T$, respectively. Within the full state vector \mathbf{x} , \mathbf{p}_I and \mathbf{v}_I represent the three-dimensional position and velocity in the inertial frame. $\mathbf{q} = [q_w, q_x, q_y, q_z]^T$ is the robot attitude expressed by unit quaternion. $\boldsymbol{\omega}_B = p\mathbf{x}_B + q\mathbf{y}_B + r\mathbf{z}_B$ is the three-dimensional angular velocity in the body frame. Within the input vector \mathbf{u} , T denotes the collective rotor thrust and $\boldsymbol{\tau}_B$ is the rotor torque expressed in the body frame. We model the TABV as a rigid body with total mass m and inertia matrix \mathbf{M} . We consider the situation where the TABV moves on flat ground. The robot dynamics and motor dynamics can then be expressed as:

$$m\ddot{\mathbf{p}}_I = T\mathbf{z}_B - mg\mathbf{z}_I + \mathbf{F}_e, \quad (1)$$

$$\mathbf{M}\dot{\boldsymbol{\omega}}_B = \boldsymbol{\tau}_B - \boldsymbol{\omega}_B \times \mathbf{M}\boldsymbol{\omega}_B + \boldsymbol{\tau}_e, \quad (2)$$

$$\begin{bmatrix} T \\ \boldsymbol{\tau}_B \end{bmatrix} = \mathcal{M}\mathbf{t} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -L/\sqrt{2} & L/\sqrt{2} & L/\sqrt{2} & -L/\sqrt{2} \\ -L/\sqrt{2} & L/\sqrt{2} & -L/\sqrt{2} & L/\sqrt{2} \\ -c_m/c_t & -c_m/c_t & c_m/c_t & c_m/c_t \end{bmatrix} \mathbf{t}, \quad (3)$$

where $g = 9.81 m/s^2$ is the magnitude of gravity, \mathbf{F}_e is the external force, $\boldsymbol{\tau}_e$ is the external torque. $\mathbf{t} = [t_1, t_2, t_3, t_4]^T$ is the rotor thrust vector, \mathcal{M} is the allocation matrix of a standard X -shape configuration, in which L is the arm length, c_m and c_t are the rotor torque and thrust coefficient, respectively.

Note that (1-3) are applicable to both terrestrial and aerial modes because the TABV is driven by rotor thrusts in both modes. The only difference lies in the additional contact forces from the ground in terrestrial mode. To be specific, $\mathbf{F}_e = \mathbf{F}_a$ in aerial mode, while $\mathbf{F}_e = \mathbf{F}_a + \mathbf{F}_c$ in terrestrial mode, where \mathbf{F}_a is the exogenous aerodynamic force (rotor drag for e.g.), $\mathbf{F}_c = \mathbf{q}_\psi \odot [F_d, F_l, F_n]^T$ is the ground contact force that (\mathbf{q}_ψ denotes the quaternion that only includes yaw rotation). Among the components of \mathbf{F}_c , F_d is the rolling resistance, F_n is the normal force, F_l is the lateral force that prevents side slipping. The corresponding nonholonomic constraint is written as:

$$(\mathbf{q}^{-1} \odot \mathbf{v}_I) \cdot \mathbf{e}_2 = 0, \quad (4)$$

where $\mathbf{e}_2 = [0, 1, 0]^T$, \odot denotes the rotation of a vector using a quaternion. In addition, because of the normal force, the TABV keeps in contact with the ground:

$$\mathbf{p}_I \cdot \mathbf{e}_3 = 0, \quad (5)$$

where $\mathbf{e}_3 = [0, 0, 1]^T$.

As the effect of the aerodynamic force only becomes noticeable during highly aggressive locomotion, we neglect \mathbf{F}_a in this work. As long as the wheel is in contact with the

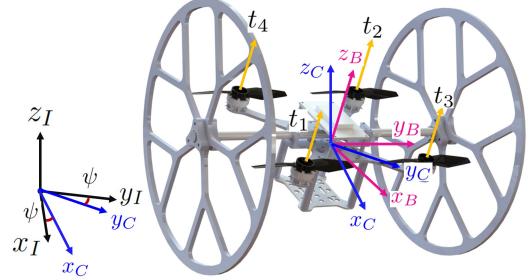


Fig. 3: Illustration of the reference frames. Three frames are introduced: inertial frame ($\mathbf{x}_I - \mathbf{y}_I - \mathbf{z}_I$) with \mathbf{Z}_I pointing in the opposite direction of the gravity vector, body frame ($\mathbf{x}_B - \mathbf{y}_B - \mathbf{z}_B$) with \mathbf{z}_B aligned with the rotor thrust vector, and intermediate frame ($\mathbf{x}_C - \mathbf{y}_C - \mathbf{z}_C$) which is separated from the inertial frame by the yaw rotation ψ .

ground, the rolling friction force F_d is always present and almost maintains a small constant value. Therefore, we omit F_d as well. The external torque $\boldsymbol{\tau}_e$ in both modes is identical, and we leave out this term because it is hard to model in real-world environments [19]. It is compensated later by the INDI inner-loop controller (introduced in Sect.V-B).

Finally, the system dynamics $f(\mathbf{x}, \mathbf{u})$ are given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}}_I \\ \dot{\mathbf{q}} \\ \dot{\mathbf{v}}_I \\ \dot{\boldsymbol{\omega}}_B \end{bmatrix} = \begin{bmatrix} \mathbf{v}_I \\ \frac{1}{2}\mathbf{q} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_B \end{bmatrix} \\ (\mathbf{F}_n + T\mathbf{z}_B)/m + \mathbf{g} \\ \mathbf{M}^{-1}[\boldsymbol{\tau}_B - \boldsymbol{\omega}_B \times \mathbf{M}\boldsymbol{\omega}_B] \end{bmatrix} \quad (6a)$$

$$\text{s.t. } F_n[(\mathbf{q}^{-1} \odot \mathbf{v}_I) \cdot \mathbf{e}_2] = 0, \quad (6b)$$

$$F_n(\mathbf{p}_I \cdot \mathbf{e}_3) = 0, \quad (6c)$$

where $\mathbf{F}_n = [0, 0, F_n]^T$. Since F_n is equal to zero in aerial mode, (6a-6c) hold in both locomotion modes.

B. Differential Flatness

This section introduces the differential flatness of TABV bimodal dynamics. In aerial locomotion, a TABV is no different from a general quadrotor whose differential flatness has been well studied by previous works [22]–[25]. The position and yaw are typically taken as the flat output, denoted as $\sigma_a = [\mathbf{p}_{I,x}, \mathbf{p}_{I,y}, \mathbf{p}_{I,z}, \psi]^T$. In practice, yaw planning is independent of the three-dimensional position planning and does not affect the quadrotor's translational motion.

Regarding terrestrial locomotion, we demonstrate that the TABV dynamics remain differentially flat with the flat output given by

$$\sigma_t = [\mathbf{p}_{I,x}, \mathbf{p}_{I,y}, \mathbf{p}_{I,z}, \theta]^T, \quad (7)$$

where θ is the pitch. The derivation of the differential flatness draws on the work of Mellinger et. al. [22].

First, the three-dimensional position (\mathbf{p}_I), velocity (\mathbf{v}_I), and acceleration (\mathbf{a}_I) are the first three terms of σ_t , $\dot{\sigma}_t$, and $\ddot{\sigma}_t$, respectively. Afterward, we show that the attitude, body angular velocity, and control input can also be derived from σ_t and its derivatives.

Since the heading of the TABV is parallel with the direction of \mathbf{v}_I , it can be expressed as

$$\psi = \arctan 2(\eta \dot{\mathbf{p}}_{I,y}, \eta \dot{\mathbf{p}}_{I,x}), \quad (8)$$

where $\eta \in \{-1, 1\}$ is the motion direction of TABV ($\eta = 1$ means forward moving and $\eta = -1$ means backward moving). Then we can write the unit vector $\mathbf{x}_C = [\cos \psi, \sin \psi, 0]^T$ as shown in Fig.3.

Because the horizontal components of \mathbf{F}_n are not considered, left multiplying (1) by \mathbf{x}_C gives:

$$T = \frac{m \ddot{\mathbf{p}}_I \cdot \mathbf{x}_C}{\sin \theta_e}. \quad (9)$$

Thus, the collective thrust T is recovered from the flat output.

As for the attitude, note that no roll motion occurs since we assume that the TABV moves on flat ground, then \mathbf{z}_B can be determined as follows:

$$\begin{aligned} \mathbf{z}_B &= \frac{\mathbf{t}}{\|\mathbf{t}\|}, \mathbf{t} = \left[\mathbf{a}_l \cos \psi, \mathbf{a}_l \sin \psi, \sqrt{T^2 - \mathbf{a}_l^2} \right]^T, \\ \mathbf{a}_l &= (\mathbf{v}_{I,x} \mathbf{a}_{I,x} + \mathbf{v}_{I,y} \mathbf{a}_{I,y}) / \sqrt{\mathbf{v}_{I,x}^2 + \mathbf{v}_{I,y}^2}, \end{aligned} \quad (10)$$

where \mathbf{a}_l is the longitudinal acceleration. Then, \mathbf{x}_B , \mathbf{y}_B , and the attitude ${}^I R_B$ (expressed by rotation matrix here) can be calculated by

$$\begin{aligned} \mathbf{y}_B &= \frac{\mathbf{z}_B \times \mathbf{x}_C}{\|\mathbf{z}_B \times \mathbf{x}_C\|}, \mathbf{x}_B = \mathbf{y}_B \times \mathbf{z}_B, \\ {}^I R_B &= [\mathbf{x}_B \ \mathbf{y}_B \ \mathbf{z}_B]. \end{aligned} \quad (11)$$

Therefore, the attitude of TABV can be determined from σ_t and its derivatives as well.

We then take the first derivative of (1):

$$m \ddot{\mathbf{p}}_I = \dot{T} \mathbf{z}_B + \boldsymbol{\omega}_B \times T \mathbf{z}_B + \dot{\mathbf{F}}_n \mathbf{z}_I. \quad (12)$$

We define the vector $\mathbf{h}_\omega = \boldsymbol{\omega}_B \times \mathbf{z}_B$ as

$$\mathbf{h}_\omega = \frac{1}{T} (m \ddot{\mathbf{p}}_I - \dot{T} \mathbf{z}_B - \dot{\mathbf{F}}_n \mathbf{z}_I) = f(\sigma_t, \dot{\sigma}_t, \ddot{\sigma}_t, \ddot{\sigma}_t). \quad (13)$$

Afterwards, the components of $\boldsymbol{\omega}_B$ are found as

$$p = -\mathbf{h}_\omega \cdot \mathbf{y}_B, q = \mathbf{h}_\omega \cdot \mathbf{x}_B, r = \dot{\psi} \mathbf{z}_I \cdot \mathbf{z}_B. \quad (14)$$

Finally, the rotor torque τ_B can be calculated from (2) since the other terms in (2) are functions of σ_t and its derivatives.

We find that \mathbf{p}_I exists in both σ_a and σ_t . In trajectory planning, \mathbf{p}_I is planned at first. Then, ψ is set to be parallel with the direction of \mathbf{v}_I in aerial locomotion as well for the continuity of trajectories. In terrestrial locomotion, θ is calculated from (9) in which the reference thrust is set to be constant.

IV. TABV TRAJECTORY PLANNING

In this section, we elaborate on the proposed flatness-based planner. Utilizing the differential flatness derived in Sect.III-B, the trajectory planning problem can be reduced to planning the three-dimensional position. We follow the typical hierarchical framework with a kinodynamic path searching front-end and a spatial-temporal trajectory optimization back-end.

A. Kinodynamic Path Searching

Kinodynamic path searching method aims at finding a safe and dynamically feasible path based on a simplified dynamic model. We adopt the hybrid-state A* algorithm [26] for path searching. An energy cost is exerted on aerial motion primitives to penalize flying locomotion. We refer the readers to our previous works [15] for more details. To deal with the nonholonomic dynamics in terrestrial locomotion which is neglected in our previous work, we use the following differential wheeled robot dynamics for terrestrial motion primitives generation.

$$\begin{cases} \dot{p}_x = v \cos \varphi, \\ \dot{p}_y = v \sin \varphi, \\ \dot{v} = a, \\ \dot{\varphi} = \omega, \end{cases} \quad (15)$$

where the state vector includes two-dimensional position $[p_x, p_y]^T$, longitudinal velocity v , and heading angle φ . Longitudinal acceleration a and heading angular velocity Ω are taken as the control input, i.e., $\mathbf{u}_t = [v, \varphi]^T$. This modeling method prohibits lateral movement, which accords with the nonholonomic dynamics of TABV.

In terms of aerial locomotion, the TABV dynamics can be simplified into a linear system written as

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{a}, \quad (16)$$

where \mathbf{p} , \mathbf{v} , and \mathbf{a} are the three-dimensional position, velocity, and acceleration, respectively. $\mathbf{u}_a = \mathbf{a}$ is taken as the control input. \mathbf{u}_t and \mathbf{u}_a is sampled simultaneously to generate both terrestrial and aerial motion primitives, leading to a terrestrial-aerial path that serves as the initial guess for trajectory optimization.

B. Optimization Problem Formulation

In trajectory optimization, we adopt MINCO trajectory class [27] to generate minimum control effort spatial-temporal trajectories. It achieves linear-complexity spatial-temporal deformation of the flat-output trajectories. The full state defined in (6a) can then be recovered from the flat output. For an m -dimensional M -piece polynomial trajectory with degree $N = 2s-1$, where s is the order of the integrator chain (we choose $s = 3$ in this work), a linear-complexity map is constructed as

$$\mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T}), \quad (17)$$

where $\mathbf{c} = (\mathbf{c}_1^T, \dots, \mathbf{c}_M^T)^T \in \mathbb{R}^{2Ms \times m}$ is the polynomial coefficient matrix and $\mathcal{M}(\mathbf{q}, \mathbf{T})$ is the smooth map from

intermediate waypoints $\mathbf{q} = (q_1, \dots, q_{M-1})^T \in \mathbb{R}^{m \times (M-1)}$ and a time allocation $\mathbf{T} = (T_1, \dots, T_M)^T \in \mathbb{R}_{>0}^M$ to \mathbf{c} . The optimization variables are then converted to $\{\mathbf{q}, \mathbf{T}\}$ through $\mathcal{M}(\mathbf{q}, \mathbf{T})$. The i -th trajectory piece $p_i(t)$ can be expressed as

$$p(t) = p_i(t - t_{i-1}), \forall t \in [t_{i-1}, t_i], \quad (18)$$

$$p_i(t) = \mathbf{c}_i^T \beta(t), \quad \forall t \in [0, T_i], \quad (19)$$

where $\beta(t) := [1, t, \dots, t^N]^T$ is the natural basis. We formulate the TABV trajectory generation problem as unconstrained nonlinear programming written as

$$\min_{\mathbf{q}, \mathbf{T}} [J_t, J_s, J_c, J_n] \cdot \lambda, \quad (20)$$

where λ is a weighting vector to trade off each cost function. In trajectory optimization, we sample three-dimensional constraint points on each trajectory piece denoted as $\tilde{\mathbf{p}}_{i,j} = p_i((j/\kappa_i) \cdot T_i), j = 0, 1, \dots, \kappa_i - 1$ for discretizing the above cost functions. From path searching, the initial guess \mathbf{q}_0 and \mathbf{T}_0 is obtained, along with each piece's locomotion mode vector denoted as $\mathbf{l} = (l_1, \dots, l_{M-1})^T$, in which $l_i = 1$ means terrestrial locomotion and $l_i = 0$ means aerial locomotion.

1) *Cost of Total time J_t* : To accelerate the navigation progress, we first minimize the total time through $J_t = \sum_{i=1}^M T_i$.

2) *Cost of State Limit J_s* : We then limit the magnitude of linear velocity and acceleration to ensure the feasibility. The cost function is formulated as follows:

$$J_s = J_{s,v} + J_{s,a}, \quad (21)$$

$$J_{s,v} = \sum_{i=1}^M \sum_{j=0}^{\kappa_i-1} \max \left\{ \left\| \dot{\tilde{\mathbf{p}}}_{i,j} \right\|_2^2 - v_{max}^2, 0 \right\}, \quad (22)$$

$$J_{s,a} = \sum_{i=1}^M \sum_{j=0}^{\kappa_i-1} \max \left\{ \left\| \ddot{\tilde{\mathbf{p}}}_{i,j} \right\|_2^2 - a_{max}^2, 0 \right\}, \quad (23)$$

where v_{max} and a_{max} are the linear velocity and acceleration thresholds, respectively. Note that J_s is applicable to both terrestrial and aerial locomotion by confining the height of $\tilde{\mathbf{p}}_{i,j}$ to be constant if $l_i = 1$.

3) *Cost of Obstacle Avoidance J_c* : To ensure a collision-free trajectory is generated, we use Euclidean Sign Distance Field (ESDF) map to obtain the distance and the corresponding gradient information of adjacent obstacles. The collision avoidance cost function J_c is defined as follows:

$$J_c = \sum_{i=1}^M \sum_{j=0}^{\kappa_i-1} \max \left\{ \|d_s - \mathcal{E}(\tilde{\mathbf{p}}_{i,j})\|_1, 0 \right\}, \quad (24)$$

where $\mathcal{E}(\cdot)$ is the distance between TABV and the nearest obstacle obtained from ESDF, d_s is the safety distance threshold.

4) *Cost of Nonholonomic Dynamics J_n* : In terrestrial locomotion, the TABV's velocity is limited to be parallel with the yaw heading due to the nonholonomic dynamics. To ensure this in trajectory optimization, we first align the directions of velocities with the desired heading angles in both initial and final states, then limit the heading angular velocity and acceleration to further guarantee dynamical feasibility along the whole trajectory. According to (8), the heading angular velocity and acceleration can be expressed as

$$\dot{\varphi}(\mathbf{p}) = \eta \frac{\dot{\mathbf{p}}^T \mathbf{B} \dot{\mathbf{p}}}{\dot{\mathbf{p}}^T \dot{\mathbf{p}}}, \mathbf{B} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad (25)$$

$$\ddot{\varphi}(\mathbf{p}) = \eta \left(\frac{\ddot{\mathbf{p}}^T \mathbf{B} \dot{\mathbf{p}}}{\dot{\mathbf{p}}^T \dot{\mathbf{p}}} - \frac{2\ddot{\mathbf{p}}^T \mathbf{B} \dot{\mathbf{p}} \dot{\mathbf{p}}^T \dot{\mathbf{p}}}{(\dot{\mathbf{p}}^T \dot{\mathbf{p}})^2} \right), \quad (26)$$

where \mathbf{p} is short for $\tilde{\mathbf{p}}_{i,j}$. In practice, we only consider $\eta = 1$ because the customized TABV platform uses a limited-FOV stereo camera for sensing (details are available in Sect.VI-A), and backward movements are not safe. The cost of nonholonomic dynamics J_n is defined as

$$J_n = J_{s,\Omega} + J_{s,\alpha}, \quad (27)$$

$$J_{s,\Omega} = \sum_{i=1}^M \sum_{j=0}^{\kappa_i-1} l_i \cdot \max \left\{ \|\dot{\varphi}(\tilde{\mathbf{p}}_{i,j})\|_2^2 - \Omega_{max}^2, 0 \right\}, \quad (28)$$

$$J_{s,\alpha} = \sum_{i=1}^M \sum_{j=0}^{\kappa_i-1} l_i \cdot \max \left\{ \|\ddot{\varphi}(\tilde{\mathbf{p}}_{i,j})\|_2^2 - \alpha_{max}^2, 0 \right\}, \quad (29)$$

where Ω_{max} and α_{max} are the heading angular velocity and acceleration threshold, respectively. In (28) and (29), if $l_i = 0$, the corresponding aerial trajectory piece is not affected by this cost.

After the final trajectory is generated, we sample points and their high-order derivatives along the trajectory and restore the desired full state and input from the differential flatness derived in Sect.III-B as the reference for trajectory tracking control.

V. TABV TRAJECTORY TRACKING CONTROL

A. Nonlinear Model Predictive Control Formulation

Given the TABV system dynamics $f(\mathbf{x}, \mathbf{u})$ (6a) and the reference trajectories generated by the TABV trajectory planner, NMPC finds the control commands by solving an optimal control problem (OCP) in a receding horizon manner. The NMPC formulation minimizes the error between predicted states and reference states in a finite look-ahead time horizon $[t_0, t_0 + h]$. For numerical calculation, the formulation is discretized with a fixed time step dt and the corresponding horizon length $N = h/dt$. Then, the NMPC problem is transcribed into constrained nonlinear programming that generates the optimal control command sequence $\mathbf{U}^* \in \mathbb{R}^{4 \times N}$:

$$\begin{aligned} \mathbf{U}^* = \operatorname{argmin}_{\mathbf{u}} \sum_{i=k}^{k+N-1} (\tilde{\mathbf{x}}_i^T \mathbf{W}_x \tilde{\mathbf{x}}_i + \tilde{\mathbf{u}}_i^T \mathbf{W}_u \tilde{\mathbf{u}}_i) + \tilde{\mathbf{x}}_N^T \mathbf{W}_x \tilde{\mathbf{x}}_N, \\ \text{s.t. } \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \mathbf{u}_{min} < \mathbf{u}_i < \mathbf{u}_{max}, \end{aligned} \quad (30)$$

where k is the current time step, $\tilde{\mathbf{x}}_i = \mathbf{x}_{ref,i} - \mathbf{x}_i$ and $\tilde{\mathbf{u}}_i = \mathbf{u}_{ref,i} - \mathbf{u}_i$ are the state and input error, respectively. $\tilde{\mathbf{x}}_N = \mathbf{x}_{ref,N} - \mathbf{x}_N$ is the end state error. \mathbf{W}_x and \mathbf{W}_u are the state and input weight matrix written as:

$$\begin{aligned}\mathbf{W}_x &= \text{diag}([\mathbf{W}_{x,p} \quad \mathbf{W}_{x,q} \quad \mathbf{W}_{x,v} \quad \mathbf{W}_{x,\omega}]), \\ \mathbf{W}_u &= \text{diag}([\mathbf{W}_{u,T} \quad \mathbf{W}_{u,\tau_{xy}} \quad \mathbf{W}_{u,\tau_z}]).\end{aligned}\quad (31)$$

To combine both terrestrial and aerial dynamics, We define $\mu_g \in \{0, 1\} \in \mathbb{N}$ and introduce two equivalent constraints as follows:

$$\mu_g[(\mathbf{q}^{-1} \odot \mathbf{v}_I) \cdot \mathbf{e}_2] = 0, \quad (32)$$

$$\mu_g(\mathbf{p}_I \cdot \mathbf{e}_3) = 0, \quad (33)$$

where $\mu_g = 0$ for aerial mode and $\mu_g = 1$ for terrestrial mode.

B. Incremental Nonlinear Dynamic Inversion

As mentioned in Sect.III-A, the external torque τ_e is omitted in TABV dynamics. As validated in recent works [19], INDI reveals distinct advantages for serving as a low-level controller by virtue of its simplicity and robustness. To be specific, INDI estimates τ_e from instantaneous sensor measurements rather than a precise dynamic model. We adopt the INDI version proposed by Sun et al. [19]. According to the robot rotational dynamics (2), we can estimate τ_e as:

$$\tau_e = M\dot{\omega} - \hat{\tau} + \hat{\omega} \times M\dot{\omega}, \quad (34)$$

where the superscript $\hat{\cdot}$ means the corresponding term is obtained from sensor measurements and then low-pass filtered with the same cut-off frequency. Angular acceleration $\dot{\omega}$ is obtained from numerical differentiation of the measured angular rate. Assuming that τ_e is slow-changing compared with the low-pass filter dynamics, the delay of τ_e is negligible [19]. Then, substitution of (34) into (2) gives:

$$\begin{aligned}M\dot{\omega}_B &= (\tau_B - \omega_B \times M\omega_B) + (M\dot{\omega} - \hat{\tau} + \hat{\omega} \times M\dot{\omega}) \\ &\approx (\tau_B - \hat{\tau}) + M\dot{\omega}.\end{aligned}\quad (35)$$

With the desired control input $\mathbf{u}_{NMP} = [T, \tau_B^T]^T$ solved by NMPC, we can acquire the desired angular acceleration $\dot{\omega}_B^d$ from (6a):

$$\dot{\omega}_B^d = M^{-1}[\tau_B - \omega \times M\omega]. \quad (36)$$

From (35) and (36), the commanded body torque τ_B^d can be obtained as:

$$\tau_B^d = \hat{\tau} + M(\dot{\omega}_B^d - \dot{\omega}). \quad (37)$$

Finally, the commanded rotor thrust vector \mathbf{t}_c can be calculated from (3) for low-level motor control.

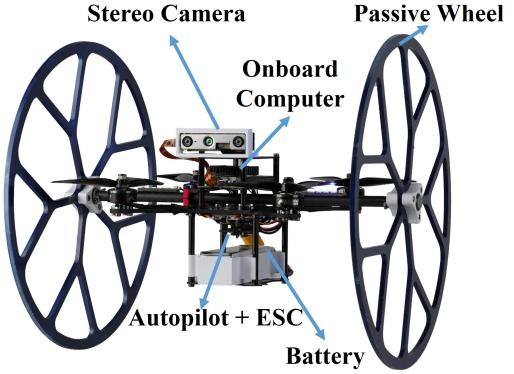


Fig. 4: Illustration of the customized TABV platform.

VI. RESULTS

A. Implementation Details

To demonstrate and verify the proposed approaches in real-world environments, we customize a TABV platform, as shown in Fig.4. The corresponding physical parameters are listed in Tab.I. The 6D localization of the TABV and the environmental information are obtained from a NOKOV motion capture system¹ and a Realsense D430 stereo camera, respectively. The mapping, planning, and control algorithms run on a Jetson Xavier NX onboard computer. In motion planning, the proposed unconstrained optimization problem is solved by LBFGS-Lite². The weighting vector λ is set to be $[5, 6, 100, 5]^T$ in experiments. In motion control, the NMPC problem is solved by ACADO [28] with qpOASES [29] at 200 Hz. The horizon length N and the time step dt is set to be 20 and 70ms, respectively. The NMPC weights are shown in Tab.II. The commanded rotor thrust (converted to rotor speed) obtained from the motion controller is sent to the ESCs via a Kakute H7 Mini autopilot.

B. Benchmark Comparisons

This section presents the benchmark comparisons against our previous work [15]. The tests are conducted in real-world environments for more convincing results. We refer readers to the attached video for more details.

1) Comparison of TABV Trajectory Planning: Firstly, we compare the proposed TABV trajectory planner with our previous one. As shown in Fig.5a, The TABV needs to navigate a series of goals which include the desired heading angles in terrestrial locomotion. These goals are specially arranged so that the TABV cannot proceed in a straight line to the next goal due to the nonholonomic dynamics. This is common in practice, for example, a vehicle needs to turn around when driving toward a rear goal. The planning results of both trajectory planners are tracked by the proposed controller for fair comparison. It turns out that the TABV safely complete the navigation task with the proposed planner, but does not even reach the first goal with our previous planner. As can be seen in Fig.5b, the main reason is that the previous planner ignores the nonholonomic

¹<https://www.nokov.com>

²<https://github.com/ZJU-FAST-Lab/LBFGS-Lite>

TABLE I: TABV Physical Parameters.

Parameter	Value
mass(m) [kg]	0.91
Inertia Matrix(M) [gm^2]	diag(7.7, 3.4, 7.3)
arm length(L) [m]	0.23
rotor thrust coefficient(c_t) [N]	$1.7e^{-8}$
rotor torque coefficient(c_m) [Nm^2]	$3.7e^{-10}$

TABLE II: NMPC Weights.

Parameter	Value
$\mathbf{W}_{x,p}$	diag(8000, 8000, 300)
$\mathbf{W}_{x,q}$	diag(400, 400, 400, 400)
$\mathbf{W}_{x,v}$	diag(100, 100, 100)
$\mathbf{W}_{x,\omega}$	diag(10, 10, 50)
$\mathbf{W}_{u,T}$	0.5
$\mathbf{W}_{u,\tau xy}$	0.1
$\mathbf{W}_{u,\tau z}$	0.2

dynamics and generates a straight-line trajectory for the sake of smoothness, which is not physically feasible in terrestrial locomotion and eventually leads to the divergence of the controller. In contrast, the proposed planner prevents lateral movement and confines the turning speed within the dynamic limit, thus ensuring the dynamic feasibility.

2) **Comparison of TABV Tracking Control:** We then compare the proposed TABV tracking controller with our previous one [15]. We choose the root-mean-square-error (RMSE) as the criteria for trajectory tracking performance, which is calculated by

$$E_r = \sqrt{\frac{1}{N} \sum_{k=1}^N \|\tilde{\mathbf{p}}^k - \mathbf{p}_{\text{ref}}^k\|^2}, \quad (38)$$

where $\tilde{\mathbf{p}}^k$ and $\mathbf{p}_{\text{ref}}^k$ are the k -th sampled actual and desired position, respectively.

The test is to track a two-dimensional lemniscate trajectory when the maximal commanded velocity and acceleration limit reach 2.0m/s and 1.8m/s^2 , respectively. It can be seen from Fig.6 that when the TABV turns at a high translational speed, the proposed controller still maintains accurate trajectory tracking, while our previous controller does not. The reason is that NMPC considers the TABV's full dynamics, and the predicting nature allows the TABV to react to the turning ahead of time.

C. Experiments

We also conduct experimental validation of our framework in an unknown complex environment cluttered with obstacles and an insurmountable fence, as shown in Fig.1a. More details are available in the attached video. As a result, the TABV manages to navigate this environment with terrestrial-aerial hybrid locomotion, revealing the robustness of the proposed framework. To further validate the proposed trajectory planner, we test it in a $50\text{m} \times 50\text{m}$ dense simulation environment on an Intel Core i7-10700 CPU. 500 tests are conducted with randomly arranged obstacles, and an instance of planning results is shown in Fig.8. As a result, 94% success rate is achieved. The average trajectory length is 76.8m , while the average computation times are 14.6ms and

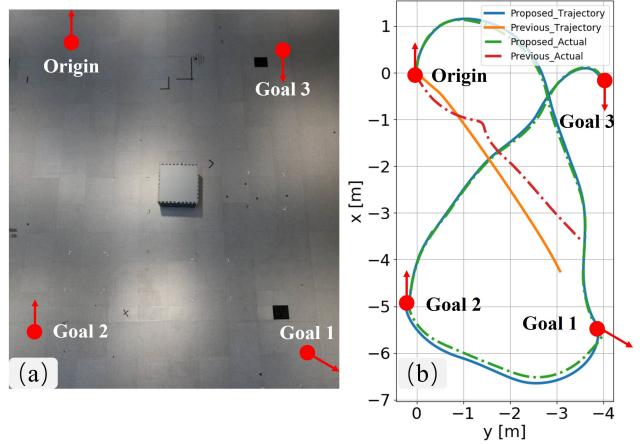


Fig. 5: The trajectory planning benchmark comparison. a) The experimental setting and the given goals. b) The planned and the actual trajectories with the proposed and the previous trajectory planners [15], respectively.

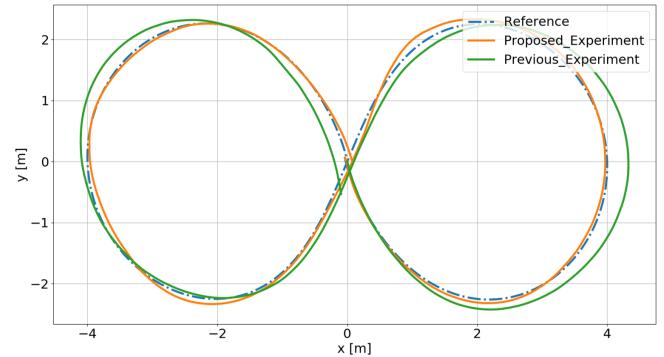


Fig. 6: Comparison of the proposed and the previous controllers [15] when tracking a two-dimensional lemniscate trajectory. The RMSE with our proposed and previous controller are 0.11m and 0.25m , respectively.

70.8ms , respectively. The above results reveal the proposed methods' robustness and real-time performance.

In addition, to further demonstrate the proposed controller's tracking performance, we conduct a three-dimensional lemniscate trajectory tracking test that requires terrestrial-aerial hybrid locomotion, as illustrated in Fig.1b. The velocity and acceleration upper bounds reach 3m/s and 2.5m/s^2 , respectively. The result shown in Fig.7 reveals that the proposed controller manages to track the trajectory and achieves smooth mode transition during both takeoff and landing. The RMSE of three-dimensional motion is 0.12m , which validates the proposed controller's performance when tracking aggressive terrestrial-aerial hybrid trajectories.

VII. CONCLUSION

In this work, we focus on the autonomous navigation of TABVs and present a model-based planning and control framework. In order to guarantee the physical feasibility and smooth locomotion transition, we first derive a unified TABV dynamic model and its differential flatness to encode the bimodal dynamics in both motion planning and control. Based on the unified dynamic model, a flatness-based trajectory planner and an NMPC-based tracking controller

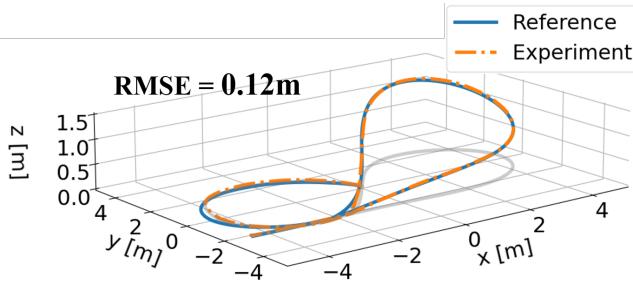


Fig. 7: The control result when tracking an aggressive terrestrial-aerial hybrid trajectory. The transparent grey curve is the projection of the three-dimensional trajectory on the ground.

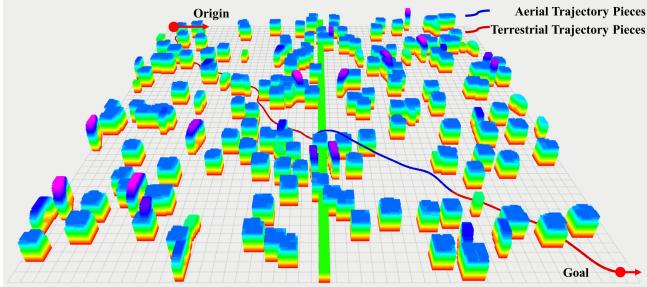


Fig. 8: An instance of the large-scale terrestrial-aerial trajectory planning test. The average computation time is 75.4ms when generating terrestrial-aerial trajectories with an average length of 76.8m.

are then proposed. Real-world benchmark comparisons and experiments validate the proposed methods' planning quality and control accuracy.

In the future, we plan to develop a more comprehensive planning and control framework that consider terresrial maneuvres on rough terrain. In addition, we will extend the proposed methods to more TABV configurations, so as to make TABVs more widely and effectively used in practice.

REFERENCES

- [1] A. Kalantari and M. Spenko, "Design and experimental validation of hytaq, a hybrid terrestrial and aerial quadrotor," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4445–4450.
- [2] C. J. Dudley, A. C. Woods, and K. K. Leang, "A micro spherical rolling and flying robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5863–5869.
- [3] S. Morton and N. Papanikopoulos, "A small hybrid ground-air vehicle concept," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5149–5154.
- [4] S. Mintchev and D. Floreano, "A multi-modal hovering and terrestrial robot with adaptive morphology," in *Proceedings of the 2nd International Symposium on Aerial Robotics*, no. CONF, 2018.
- [5] S. Sabet, A.-A. Agha-Mohammadi, A. Tagliabue, D. S. Elliott, and P. E. Nikravesh, "Rollocopter: An energy-aware hybrid aerial-ground mobility for extreme terrains," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–8.
- [6] B. Li, L. Ma, D. Wang, and Y. Sun, "Driving and tilt-hovering—an agile and manoeuvrable aerial vehicle with tiltable rotors," *IET Cyber-Systems and Robotics*, 2021.
- [7] H. C. Choi, I. Wee, M. Corah, S. Sabet, T. Kim, T. Touma, D. H. Shim, and A.-a. Agha-mohammadi, "Baxter: Bi-modal aerial-terrestrial hybrid vehicle for long-endurance versatile mobility: Preprint version," *arXiv preprint arXiv:2102.02942*, 2021.
- [8] N. B. David and D. Zarrouk, "Design and analysis of feststar, a hybrid flying and climbing sprawl tuned robot," *IEEE Robotics and Automation Letters*, 2021.
- [9] K. Kim, P. Spieler, E.-S. Lupu, A. Ramezani, and S.-J. Chung, "A bipedal walking robot that can fly, slackline, and skateboard," *Science Robotics*, vol. 6, no. 59, p. eabf8136, 2021.
- [10] H. Jia, S. Bai, R. Ding, J. Shu, Y. Deng, B. L. Khoo, and P. Chirarattananon, "A quadrotor with a passively reconfigurable airframe for hybrid terrestrial locomotion," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4741–4751, 2022.
- [11] B. Araki, J. Strang, S. Pohorecky, C. Qiu, T. Naegeli, and D. Rus, "Multi-robot path planning for a swarm of robots that can both fly and drive," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5575–5582.
- [12] A. Sharif, H. Lahiru, S. Herath, and H. Roth, "Energy efficient path planning of hybrid fly-drive robot (hyfdr) using a* algorithm," in *ICINCO (2)*, 2018, pp. 211–220.
- [13] S. Choudhury, J. P. Knickerbocker, and M. J. Kochenderfer, "Dynamic real-time multimodal routing with hierarchical hybrid planning," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2397–2404.
- [14] D. D. Fan, R. Thakker, T. Bartlett, M. B. Miled, L. Kim, E. Theodorou, and A.-a. Agha-mohammadi, "Autonomous hybrid ground/aerial mobility in unknown environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3070–3077.
- [15] R. Zhang, Y. Wu, L. Zhang, C. Xu, and F. Gao, "Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3008–3015, 2022.
- [16] T. Wu, Y. Zhu, L. Zhang, J. Yang, and Y. Ding, "Unified terrestrial/aerial motion planning for hytaqs via nmpc," *IEEE Robotics and Automation Letters*, 2023.
- [17] J. Colmenares-Vázquez, P. Castillo, N. Marchand, and D. Huerta-García, "Nonlinear control for ground-air trajectory tracking by a hybrid vehicle: theory and experiments," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 19–24, 2019.
- [18] J. Yang, Y. Zhu, L. Zhang, Y. Dong, and Y. Ding, "Sytab: A class of smooth-transition hybrid terrestrial/aerial bicopters," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9199–9206, 2022.
- [19] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.
- [20] S. Atay, M. Bryant, and G. Buckner, "Control and control allocation for bimodal, rotary wing, rolling-flying vehicles," *Journal of Mechanisms and Robotics*, vol. 13, 2021.
- [21] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, "Nonlinear mpc for quadrotor fault-tolerant control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5047–5054, 2022.
- [22] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, Shanghai, China, May 2011, pp. 2520–2525.
- [23] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.
- [24] M. Watterson and V. Kumar, "Control of quadrotors using the hopf fibration on $so(3)$," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2019, pp. 199–215.
- [25] Z. Wang, C. Xu, and F. Gao, "Robust trajectory planning for spatial-temporal multi-drone coordination in large scenes," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 182–12 188.
- [26] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.
- [27] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.
- [28] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [29] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.