2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Trajectory Planning for the Bidirectional Quadrotor as a Differentially Flat Hybrid System

Katherine Mao, Jake Welde, M. Ani Hsieh, and Vijay Kumar

***Abstract*— The use of bidirectional propellers provides quadrotors with greater maneuverability which is advantageous in constrained environments. This paper addresses the development of a trajectory planning algorithm for quadrotors with bidirectional motors. Previous work has shown that the property of differential flatness can be leveraged for efficient trajectory planning. However, planners that leverage flatness for quadrotors fail at points where the acceleration of the center of mass is equal to gravity, *i.e.*, when the vehicle experiences free fall. The central contribution of this paper is a flatness-based trajectory planning method that allows quadrotors to use bidirectional propellers *and* pass through the so-called free-fall singularity. We model our system as a differentially flat hybrid system with the aid of coordinate charts derived from the Hopf fibration and develop an algorithm that computes forward and reverse thrusts for each propeller, resulting in smooth trajectories everywhere in $SE(3)$. We demonstrate the planner's versatility by planning knife-edge maneuvers and trajectories passing through the free-fall singularity, while transitioning from forward to reverse thrust.**

I. Introduction

Recent hardware developments in symmetric fixed-pitched aerial vehicle propellers and reversible electronic speed controllers have led to the introduction of bidirectional thrusters, giving aerial vehicles the ability to generate both positive and negative thrust. The application of such thrusters can significantly expand the potential flight envelope of a quadrotor, enabling sustained inverted flight and greater agility. The ability to transition between sustained upright and inverted flight, referred to as flight orientations, also enables the use of sensors or manipulators in the workspace above or below the vehicle [1], [2], and safe landings on moving or inclined surfaces [3], [4], increasing the vehicle's versatility.

Existing quadrotor trajectory planning methods, such as Mellinger's minimum snap planner [5] [6] and Mueller's motion primitive planner [7] face one major limitation when planning trajectories for bidirectional quadrotors: the avoidance of zero thrust, or the condition where acceleration is only due to gravity. This limitation arises due to the differentially flat characteristic of quadrotors these planners leverage, where the orientation of the quadrotor aligns the thrust vector with the translational acceleration minus the acceleration due to gravity. For a unidirectional quadrotor, passing through this zero thrust, *i.e.* free-fall state, would require a discontinuity in the vehicle attitude. Yet, it is apparent that for a bidirectional quadrotor to reverse motor direction, the net thrust must reach zero at some intermediary time.

Recent works addressing bidirectional quadrotors have focused on control methods that can account for the high forces and moments experienced by the quadrotor during aggressive maneuvers such as flips [8], [9], and accommodate the range of expected orientations the quadrotor will traverse [10]. Trajectory planning for the greater flight envelope of a bidirectional quadrotor has been limited to simple, often planar, trajectories designed to demonstrate a flipping maneuver, starting and ending at hover [11], [12], [1]. In addition, these existing planners often fail or disregard key requirements that ensure smoothness, generating dynamically infeasible trajectories which are only approximately tracked by a closed loop controller [11], [1].

In our work, we eliminate the artificial zero thrust avoidance constraint and show how to plan motions that go directly through this free-fall singularity. The main contribution of this paper is a novel trajectory planning algorithm for bidirectional quadrotors that guarantees smooth, dynamically feasible trajectories throughout the full flight envelope of the vehicle. Our approach builds upon [5], utilizing a minimum snap cost function to plan piece-wise polynomial trajectories with continuity constraints. Our method ensures an efficient and smooth state trajectory while satisfying constraints. We accomplish this by modeling the bidirectional quadrotor as a differentially flat hybrid system, where distinct modes correspond to the local yaw parameterizations determined with the Hopf fibration [13] and the sign of the net thrust, ensuring smooth transitions between coordinate charts. We demonstrate how the proposed planning strategy is used to maneuver a bidirectional quadrotor through environments with dense obstacles without constraining the quadrotor's orientation and acceleration a priori.

This paper is organized as follows: we begin with a summary of our notation in Section II and describe our system model in Section III. Section IV describes the development of the hybrid flatness diffeomorphisms, and the development of the trajectory planning algorithm is covered in Section V. Finally, Sections VI,VII present the experimental results and some concluding remarks.

II. On Symbol Convention

We represent scalar variables and quaternions in unmodified typeface (x, a, q) and vectors with a bolded typeface

We gratefully acknowledge the support of ARL DCIST CRA W911NF-17-2-0181, NSF Grants CCR-2112665, C-BRIC, a Semiconductor Research Corporation Joint University Microelectronics program cosponsored by DARPA and Qualcomm Research.

Katherine Mao, Jake Welde, M. Ani Hsieh, and Vijay Kumar are with the Department of Mechanical Engineering and Applied Mechanics and the GRASP Laboratory, University of Pennsylvania, PA, 19104, USA {maokat, jwelde, mya, kumar}@seas.upenn.edu

$(\mathbf{r}, \mathbf{v}, \boldsymbol{\omega})$. For quaternions, we follow the Hamilton Convention [14] of $ijk = -1$, where $q = q_0 + iq_1 + jq_2 + kq_3$. Superscripts are specified to denote the frame in which a vector's components are expressed ($\mathbf{b}^W$). Table I gives the definitions of symbols used throughout the paper.

TABLE I
DEFINITIONS OF VARIABLES

$\mathcal{W}, \mathcal{A}, \mathcal{B}$	World frame, intermediate body frame, and body frame of quadrotor
$\mathbf{I}^3 \in \mathbb{R}^3$	3×3 Identity Matrix
$g \in \mathbb{R}$	Magnitude of the acceleration due to gravity
$\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 \in \mathbb{R}^3$	Body Frame Vectors of Quadrotor in $\mathcal{W}$
$\mathbf{b}_3 = [a, b, c]^T \in \mathbb{R}^3$	Unit components of $\mathbf{b}_3$ in $\mathcal{W}$
$[\dot{a}, \dot{b}, \dot{c}] \in \mathbb{R}^3$	Time Derivative of $[a, b, c]$ in $\mathcal{W}$
$\mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}}, \dddot{\mathbf{r}} \in \mathbb{R}^3$	Position, Velocity, Acceleration, Jerk vectors of quadrotor in $\mathcal{W}$
$\psi \in \mathbb{S}^1$	Yaw of quadrotor $\mathcal{B}$ in $\mathcal{A}$
$f \in \mathbb{R}$	Net thrust of quadrotor
$m_1, m_2, m_3 \in \mathbb{R}$	Body moments of quadrotor
$\boldsymbol{\omega} \in \mathbb{R}^3$	Angular Velocity of Quad in $\mathcal{W}$
$I \in \mathbb{R}^{3\times3}$	Inertia Matrix of the Quad in $\mathcal{B}$

III. System Modeling

We begin by defining the quadrotor body frame $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ that resides in a world frame world frame $\mathcal{W} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ such that $\mathbf{e}_3$ is antiparallel to gravity, as shown in Fig. 1. We also define an intermediary body frame $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$, introduced in Sec IV-B. The dynamical model of the quadrotor is given by:

$$m\ddot{\mathbf{r}} = -mg\mathbf{e}_3 + f\mathbf{b}_3^W \tag{1}$$

$$\dot{\boldsymbol{\omega}}^B = I^{-1}\left(-\boldsymbol{\omega}^B \times I\boldsymbol{\omega}^B + \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}\right) \tag{2}$$

where the quadrotor applies thrust along the $\mathbf{b}_3$ direction. We define $\mathbf{b}_1$ and $\mathbf{b}_2$ along the perpendicular arms of the quadrotor. From [5], we can determine a relation between f, m_i and the motor speeds of the quadrotor, given knowledge of the characteristic length of the quadrotor L, thrust coefficient k_f, and drag coefficient k_m:

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ f \end{bmatrix} = \begin{bmatrix} 0 & k_f L & 0 & k_f L \\ -k_f L & 0 & -k_f L & 0 \\ k_m & -k_m & k_m & -k_m \\ k_f & k_f & k_f & k_f \end{bmatrix} \begin{bmatrix} sgn(s_1)s_1^2 \\ sgn(s_2)s_2^2 \\ sgn(s_3)s_3^2 \\ sgn(s_4)s_4^2 \end{bmatrix} \tag{3}$$

where $sgn(s_i)$ denotes the direction of spin of motor i, given by

$$sgn(s_i) = \begin{cases} 1, & \text{normal spin} \\ -1, & \text{reverse spin} \end{cases}. \tag{4}$$

IV. Differential Flatness

A differentially flat system is defined as one where the system's state and inputs can be smoothly parameterized in terms of a set of so-called flat outputs and their derivatives. Per [15], [16], a differentially flat hybrid system is one where multiple sets of differentially flat mappings exist, and the transition functions between each pair of mappings, as well as the conditions triggering a transition, can also be written as a function of the flat outputs. We use 'guard' to refer to the transition condition between two mappings and 'reset' for the transition function between a pair of mappings corresponding to a guard. For the quadrotor, we determine a diffeomorphism between the following variables:

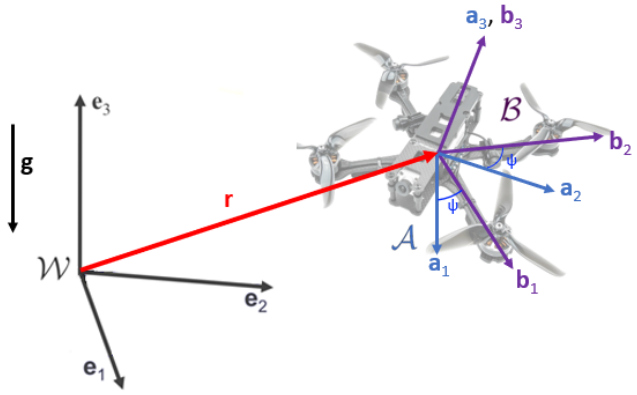

Fig. 1. The quadrotor, with its body frame denoted by $\mathcal{B}$, with some orientation ψ and displacement $\mathbf{r}$ with respect to the world frame $\mathcal{W}$.

$$\text{Flat Outputs: } \mathbf{r}, \psi$$

$$\text{States and Inputs: } \mathbf{r}, q, \dot{\mathbf{r}}, \boldsymbol{\omega}, s_1^2, s_2^2, s_3^2, s_4^2.$$

Both early [6] and more recent [5] work on differential flatness in quadrotors has been limited to a single flatness diffeomorphism on a local region of the state space. For the first time, we describe the quadrotor as a differentially flat hybrid system, enabling operation over the full performance envelope. In the remainder of this section, we describe the necessary extensions to the standard model.

A. Positive and Negative Thrust

In the traditional flat output derivation for quadrotors [5], it is assumed that the thrust vector and $\mathbf{b}_3$ are parallel. For bidirectional quadrotors, this approach introduces an ambiguity to the direction of $\mathbf{b}_3$, as thrust can be generated antiparallel to $\mathbf{b}_3$. We resolve this ambiguity by introducing an additional sign parameter to the original formulation.

We can rewrite (1) to find the desired thrust vector $\mathbf{F}$ for some desired positional acceleration.

$$\mathbf{F} = m\ddot{\mathbf{r}} + mg\mathbf{e}_3 \tag{5}$$

We use η to account for the quadrotor's ability to reverse spin and generate inverted thrust. We use $\uparrow\!\uparrow$ to denote parallel vectors and $\uparrow\!\downarrow$ for antiparallel vectors. With these two values, we can fully determine $\mathbf{b}_3$ as

$$\mathbf{b_3} = [a, b, c]^T = \eta\frac{\mathbf{F}}{||\mathbf{F}||} \tag{6}$$

where η is given by

$$\eta = \begin{cases} 1, & \text{if } \mathbf{b}_3 \uparrow\!\uparrow \mathbf{F} \\ -1, & \text{if } \mathbf{b}_3 \uparrow\!\downarrow \mathbf{F} \end{cases}. \tag{7}$$

For the remainder of this paper, we will refer to the $\mathbf{b}_3 \uparrow\Uparrow \mathbf{F}$ and $\mathbf{b}_3 \uparrow\Downarrow \mathbf{F}$ as the $(+)$ and $(-)$ 'postures' respectively.

B. Yaw Parametrization

A well-known singularity exists in the approach taken by [5] for constructing the quadrotor attitude when $\mathbf{e_3} \cdot \mathbf{b_3} = 0$, effectively limiting pitch and roll to 90° from hover *i.e.* $c \geq 0$. Instead, we follow the approach of [13] for two overlapping almost-global representations of quadrotor's orientation in $SO(3)$ [17]. By exploiting the Hopf fibration, an almost-global invertible mapping from $\mathbb{S}^3$ to $\mathbb{S}^2 \times \mathbb{S}^1$ (we refer to [18] for a more complete explanation), two sets of quaternions can be constructed in view of (6) and a desired yaw ψ, with respective singularity points at $c = -1$ and $c = 1$. We achieve global coverage on $SO(3)$ by switching as needed between these two mappings, which we refer to as the North (N) and South (S) charts, given by

$$\begin{aligned} q_N &: \mathbb{S}^2 \setminus \{-e_3\} \times \mathbb{S}^1 \to \mathbb{S}^3 \\ (\mathbf{b}_3, \psi) &\mapsto q_{abc,N}(\mathbf{b}_3) \otimes q_\psi(\psi) \end{aligned} \tag{8}$$

$$\begin{aligned} q_S &: \mathbb{S}^2 \setminus \{e_3\} \times \mathbb{S}^1 \to \mathbb{S}^3 \\ (\mathbf{b}_3, \psi) &\mapsto q_{abc,S}(\mathbf{b}_3) \otimes q_\psi(\psi) \end{aligned} \tag{9}$$

These charts are in turn determined using the maps

$$q_{abc,N} : (a,b,c) \mapsto \frac{1}{\sqrt{2(1+c)}} \begin{bmatrix} 1+c \\ a \\ -b \\ 0 \end{bmatrix} \tag{10}$$

$$q_{abc,S} : (a,b,c) \mapsto \frac{1}{\sqrt{2(1-c)}} \begin{bmatrix} -b \\ 1-c \\ 0 \\ a \end{bmatrix} \tag{11}$$

which give the quaternion representing the orientation of the intermediate frame $\mathcal{A}$ in terms of the $\mathbf{b}_3$ axis. Finally, both the (N) and (S) charts employ the same map describing the difference in orientation between the $\mathcal{A}$ and $\mathcal{B}$ frames in terms of the yaw angle, i.e.

$$q_\psi : \psi \mapsto \begin{bmatrix} cos(\psi/2) \\ 0 \\ 0 \\ sin(\psi/2) \end{bmatrix} \tag{12}$$

The use of more than one yaw parametrization in the flat outputs is unavoidable due to topological obstructions [19].

C. Hybrid Modes

As the $+/-$ postures and N/S charts are independent of one another, there are a total of four modes (Fig. 2), each corresponding to a differentially flat system. Together, these modes, $\{+N, -N, +S, -S\}$, describe the bidirectional quadrotor's dynamics.

D. Guards and Resets

Next, we examine the switching dynamics between these four modes. The transition between $(+)$ and $(-)$ (and vice versa) naturally occurs when $||\mathbf{F}|| = 0$. Thus, we express

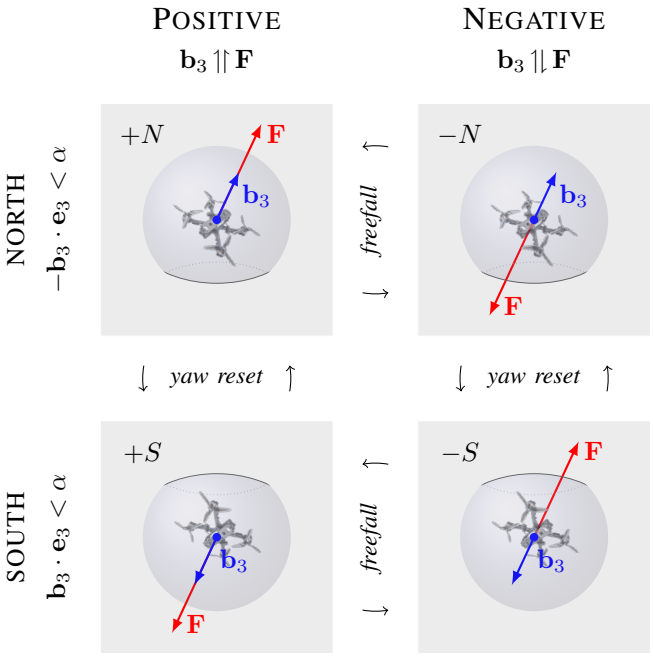

Fig. 2. A bidirectional quadrotor can be modeled as a differentially flat hybrid system where the four distinct modes $\{+N, -N, +S, -S\}$ arise due to the possibility of forward or reverse thrust and the need for two yaw parametrizations.

the guard, using (5), as $\ddot{\mathbf{r}} = -g\mathbf{e}_3$. The corresponding reset is an identity map, as the values of the flat outputs of the quadrotor do not change. It is clear from (6) that $\ddot{\mathbf{r}}$, $\mathbf{F}$, and $\mathbf{b}_3$ are coupled, and without meeting the guard, any transition between $(+)$ and $(-)$ would result in a discontinuity in the quadrotor's attitude.

Looking at Fig. 2, we see that (8) and (9) overlap everywhere but $c = \pm 1$. Selecting a critical angle beyond which mode transition should occur, we express the guard for the mode switch from $(N) \to (S)$ as

$$e_3 \cdot \frac{\ddot{\mathbf{r}} + g\mathbf{e}_3}{||\ddot{\mathbf{r}} + g\mathbf{e}_3||} = -\eta\,\alpha, \tag{13}$$

and the guard for the mode switch from $(S) \to (N)$

$$e_3 \cdot \frac{\ddot{\mathbf{r}} + g\mathbf{e}_3}{||\ddot{\mathbf{r}} + g\mathbf{e}_3||} = \eta\,\alpha. \tag{14}$$

Note that because these guard surfaces are not located at the same angle, but instead includes overlap between the charts over a large region. This provides a degree of hysteresis to ensure only a finite number of switches between (N) and (S) occur in finite time.

We determine the reset mapping from the quaternion formulation demonstrated in [13]. Application of $q_{abc,N}$ and $q_{abc,S}$ rotate $\mathcal{W}$ to differing intermediary $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ frames, aligned along a_3. Differing yaw angles (ψ_N, ψ_S) are applied to $q_\psi(\psi)$ from (8, 9) in (N) and (S) to reach $\mathcal{B}$. We find the angle between these two definitions of $\mathbf{a}_1, \mathbf{a}_2$ at any given $\mathbf{a}_3$ by equating the two quaternions,

$$q_{abc,N} \otimes q_\psi(\psi_N) = q_{abc,S} \otimes q_\psi(\psi_S). \tag{15}$$

and simplifying to obtain a relationship between ψ_N and ψ_S, namely

$$\psi_S = 2\arctan 2(a,b) + \psi_N \tag{16}$$

to determine the yaw reset map, while the reset for all other flat outputs is an identity map.

V. Trajectory Planning

We construct system trajectories as a set of piece-wise polynomials, each confined to a single mode of the differentially flat hybrid system. We plan such trajectories in two steps. First, we solve for the translational trajectory according to preassigned $+/-$ postures. Next, as (13), (14) depend only on the translational acceleration, we can assign N/S charts that are consistent with the already planned $\ddot{\mathbf{r}}$ and the assigned $+/-$ posture, fully determining the mode of each segment of the trajectory. We then insert the necessary transition constraints from the reset maps to solve for the yaw trajectory in a second pass.

To plan the translational portion, we follow the work of [5], using a minimum-snap planner to generate a piece-wise polynomial trajectory ($\sigma(t) \in \mathbb{R}^3$) given k waypoints, where the $+/-$ posture of each segment (σ_j, $j = [1, 2, \ldots, k]$) is assigned a priori.[1] Each $\sigma_j(t)$ is given by

$$\sigma_j(t) = \sum_{i=0}^{7} c_{j_i} t^i \tag{17}$$

We then formulate the planning problem as the optimization problem

$$\min_{\mathbf{c}} \int_{t_0}^{t_k} \left|\left| \frac{d^4}{dt^4}\sigma(t) \right|\right|^2 dt. \tag{18}$$

subject to:

$$\textit{Waypoint Constraints:} \quad A_{eq}\mathbf{c} = \mathbf{b}_{eq} \tag{19}$$

$$\textit{Collision Constraints:} \quad A_{ineq}\mathbf{c} \leq \mathbf{b}_{ineq} \tag{20}$$

$$\textit{Continuity Constraints:} \quad \begin{cases} \dot{\sigma}_{j-1}(t_{j-1}) = \dot{\sigma}_j(0) \\ \ddot{\sigma}_{j-1}(t_{j-1}) = \ddot{\sigma}_j(0) \\ \dddot{\sigma}_{j-1}(t_{j-1}) = \dddot{\sigma}_j(0) \end{cases} \tag{21}$$

$$\textit{Boundary Constraints:} \quad \begin{cases} \dot{\sigma}_1 = \ddot{\sigma}_1 = \dddot{\sigma}_1 = 0 \\ \dot{\sigma}_k = \ddot{\sigma}_k = \dddot{\sigma}_k = 0 \end{cases} \tag{22}$$

$$\textit{Reset Constraints:} \quad \ddot{\sigma}_r(t_r) = -g\mathbf{e}_3, \; r \in R \tag{23}$$

where R is the set of waypoints at which mode switches occur and $\mathbf{c}$ the 1-dimensional array of polynomial coefficients (c_{j_i}). This minimization can be cast as a Quadratic Program with the trajectory coefficients as decision variables. The costs and constraints take the form

$$\begin{aligned} &\min_x \mathbf{c}^T H \mathbf{c} + f\mathbf{c} \\ &\text{s.t. } A_{\text{eq}}\mathbf{c} = b_{\text{eq}}, \\ &\quad A_{\text{ineq}}\mathbf{c} \leq b_{\text{ineq}} \end{aligned} \tag{24}$$

[1]Note that a search overall possible mode sequences can be formulated as a mixed integer quadratic program [15], if we wish to explore all possible trajectories. Because a small number of mode switches are usually sufficient to perform a given flight task, such optimization problems are tractable.

where H is a positive definite matrix such that $\mathbf{c}^T H\mathbf{c} + f\mathbf{c}$ is the re-formatted cost function. We operate under the assumption that mode switches occur only at waypoints. We determine the interval of time between waypoints through a simple heuristic which divides the Euclidean distance between waypoints by a nominal speed v. We can repeat an analogous process to plan the yaw trajectory.

We remark that unlike previous motion planners for bidirectional quadrotors, we do not explicitly specify a flipping maneuver transitioning from upright to inverted flight, nor where a flip should occur if one does. As we demonstrate in our experiments, our planner generates trajectories that may include flight orientation transitions, net thrust transitions without inversion, or both, according to the costs and constraints imposed by the environment.

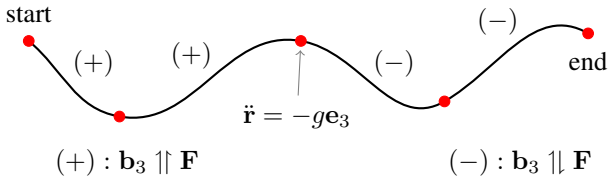

Fig. 3. A sample translation trajectory generated by the minimum snap planner, with the first two segments in (+) and the next two segments in (−). The freefall acceleration constraint of $\ddot{\mathbf{r}} = -g\mathbf{e}_3$ is added at the second waypoint to ensure a smooth transition between (+) and (−).

VI. Simulation Results

For simulation experiments, a quadrotor model is constructed through the equations of motion given in (1) and (2). The `scipy.integrate`[2] package is used for numerical integration to model the quadrotor responses at a rate of 500Hz. We use quadrotor parameters from the Crazyflie 2.0 from bitcraze[3]. No motor dynamics were incorporated. For each trajectory, we specify positional and continuity constraints at each waypoint, plus start and stop hover conditions. In our approach, the planning problem is ultimately cast as a quadratic program of similar size and sparsity to existing methods which achieve real time planning on resource-constrained platforms [20].

A. Experiment 1: Mode Transitions

We first present a series of trajectories where a bidirectional quadrotor passes through a sequence of modes, transitioning to sustained inverted flight where necessitated by the constraints. In each trajectory, a sequence of four modes is specified to traverse a planar five waypoint ($k = 5$) path. The quadrotor travels a total of $10m$ along the y-axis and $5m$ along the z-axis, with $v = 2m/s$. Different mode sequences are specified to demonstrate the adaptability of the planner.

In the the first trajectory, Fig. 4a, the mode switch from (+) to (−) occurs at the middle waypoint. We see in Fig. 4c that $\mathbf{b}_3$ is horizontal at the point of the the mode switch

[2]docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp
[3]www.bitcraze.io/products/old-products/crazyflie-2-0/

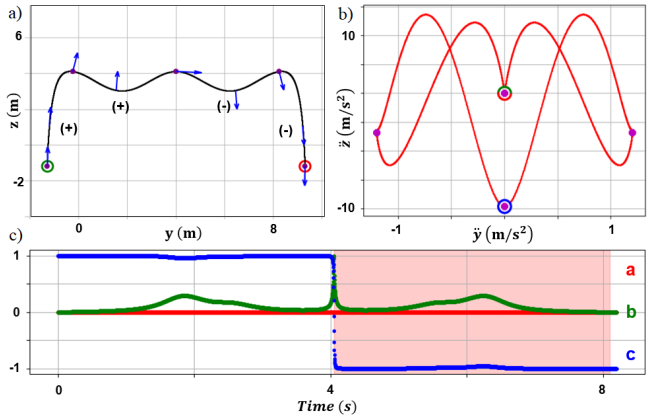

Fig. 4. Top left: A trajectory (black) with five waypoints (purple) and one mode switch. The start and goal waypoints are marked by the green and red makers respectively. Top right: The trajectory $\ddot{y}$ vs $\ddot{z}$, passing through the free-fall singularity ($\ddot{x}, \ddot{y} = 0, \ddot{z} = -g$) (blue) at the point of mode transition ($\ddot{x}, \ddot{y} = 0, \ddot{z} = -g$). Bottom: The components of $\mathbf{b}_3$ over the duration of the trajectory. $(+)$ and $(-)$ are marked by white and red zones respectively.

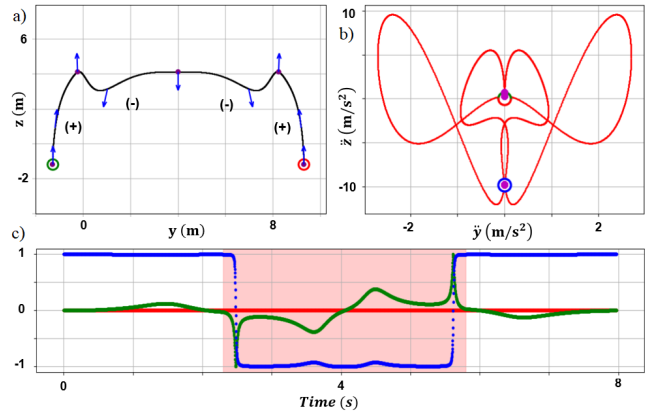

Fig. 5. Top left: A trajectory with five waypoints and two mode switches Top right: The trajectory $\ddot{y}$ vs $\ddot{z}$. Bottom: The components of $\mathbf{b}_3$ over the duration of the trajectory.

during its transition in flight orientation. This observation is reinforced in Fig. 4a,b. From (5) and (6), we know the direction of the quadrotor's thrust vector and acceleration vector are aligned. As this is a yz planar path, a vector drawn from the blue circle indicating the $F = 0$ singularity to any point on the red path is aligned with the quadrotor's net thrust at that point in time. As the quadrotor's acceleration approaches the singularity point, we observe the orientation vector approaches the horizontal.

In the second trajectory, Fig. 5a, mode switches are specified at the second and fourth waypoints. Again, we see that the transition in flight orientation occurs around the mode switch (Fig. 5c). Fig. 5b shows that $\mathbf{b}_3$ is vertical at the point of the mode switch. The full transition occurs shortly before the first mode switch and shortly after the second.

B. Experiment 2: Agile Flight

Next, we demonstrate the use of our planner to plan a trajectory where a quadrotor switches from positive to

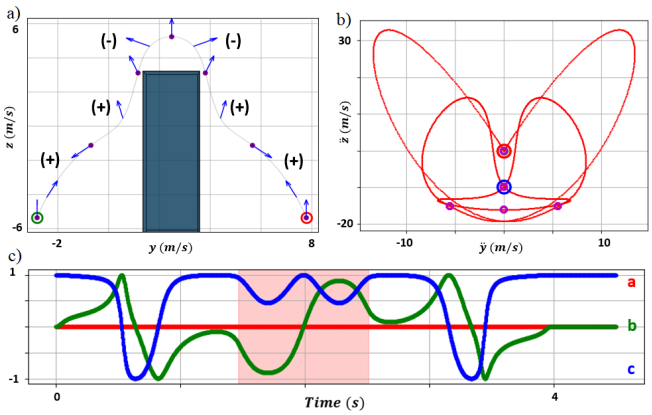

Fig. 6. Top Right: A trajectory with two mode switches without flight orientation changes. Top Left: The trajectory $\ddot{y}$ vs $\ddot{z}$. c) The components of $\mathbf{b}_3$ over the duration of the trajectory.

negative thrust without transitioning flight orientations. In this trajectory, Fig. 6a, we traverse ($k = 5$) waypoints set approximately along a sine curve, avoiding the $2m \times 9m$ obstacle in green. We use $v = 4.5m/s$.

From Fig. 6b and Fig. 6c, we can see that through the course of the mode switch from $(+)$ to $(-)$ and back, $\mathbf{b}_3$ remains close to vertical, showing that while that quadrotor switches thrust directions, the quadrotor never transitions to sustained inverted flight. Near the beginning and end of the trajectory, the quadrotor does temporarily transition to inverted flight before returning upright.

C. Experiment 3: Narrow Window

In the classic narrow window problem, a quadrotor must perform a knife-edge maneuver, rotating itself to 90° to pass through an obstacle without collision. With unidirectional quadrotors, once the knife-edge has been achieved, the quadrotor must then fight momentum to roll itself back to an upright position. Here, we present a formulation of the narrow window path where a bidirectional quadrotor follows its momentum after the knife-edge to roll itself into an inverted hover.

With existing methods such as [21], an acceleration constraint is used to to specify a quadrotor's desired orientation ($\mathbf{b}_3^W$).

$$\ddot{\mathbf{r}} = ||\ddot{\mathbf{r}} + g\mathbf{e}_3||\mathbf{b}_3^W + g\mathbf{e}_3 \tag{25}$$

where $\mathbf{b}_3^W$ is non-zero. It becomes easy to see a conflict of constraints when the location of an orientation constraint coincides with the guard between $(+)$ and $(-)$.

Instead, we examine the leading and exiting trajectories to this point. We assume there exists some θ angle tolerance from the horizontal where the quadrotor can traverse the obstacle collision-free (Fig. 7).

For clarity, we choose to align our coordinate system with the narrow window, such that the plane of the aperture is normal to the y-axis with the narrow dimension running along the x-axis. To ensure the vehicle passes through the aperture, we require that at the time the quadrotor traverses

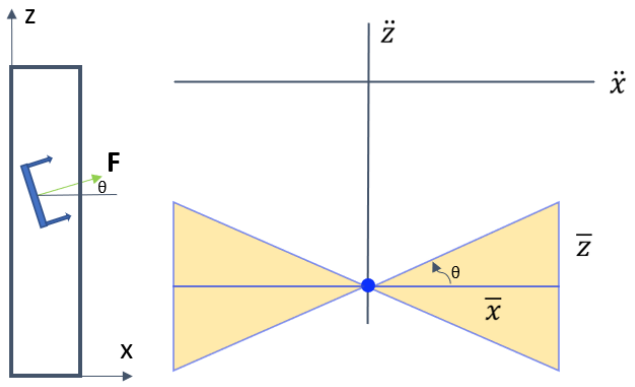

Fig. 7. A quadrotor with its $\mathbf{b}_3$ vector angled θ from the horizontal, traversing through a narrow window. The blue dot marks the free-fall singularity ($\ddot{x}, \ddot{y} = 0, \ddot{z} - -g$. $\bar{x}$ and $\bar{z}$ represent values of $\ddot{x}, \ddot{z}$ referenced from the singularity. The yellow regions correspond to the acceleration constraint (27).

the window, its velocity is orthogonal to the narrow direction of the window, i.e.

$$\dot{\mathbf{x}} = \mathbf{0} \tag{26}$$

Additionally, by exploiting the connection between quadrotor acceleration and orientation, we can write a series of inequality constraints in the form of (20), ensuring that during a short interval of time leading to and exiting from the narrow window, the quadrotor maintains an angle from the horizontal of less than θ, *i.e.*

$$\ddot{\mathbf{x}} \geq \left| \frac{\ddot{\mathbf{z}} + g\mathbf{e}_3}{\tan(\theta)} \right| \tag{27}$$

This method allows us to simultaneously constrain the orientation of a quadrotor's transition from upright to inverted flight while ensuring a mode switch in the same maneuver without conflicting constraints. In addition, unlike the method used in [21], this method does not require imposing an arbitrary magnitude for the quadrotor's acceleration.

With these added constraints, we demonstrate a knife-edge maneuver in a non-planar trajectory (Fig. 8a), with $v = 4m/s$. The quadrotor passes through a window of width $0.2m$, given $\theta = 10°$. Where the quadrotor previously flipped about the axis perpendicular to its trajectory, here we demonstrate a trajectory where the quadrotor flips about its axis of travel (Fig. 8b), successfully traversing through a narrow window obstacle. The green and red triangles in Fig. 8c mark the entrance and exit to the region of acceleration inequality constraints outlined in (27).

D. Discussion of Simulation Results

With these simulation experiments, we have demonstrated smooth, dynamically feasible trajectories where the transition from sustained upright to inverted flight (and vice versa) emerges from the imposed constraints and cost function, rather than being manually programmed. This is particularly clear in view of the bottom plots in Figs. 4, 5, 6, and 8, since the quadrotor's orientation changes dramatically before, at, after, or independently of a change in the sign of the thrust, instead of being commanded to instantaneously flip

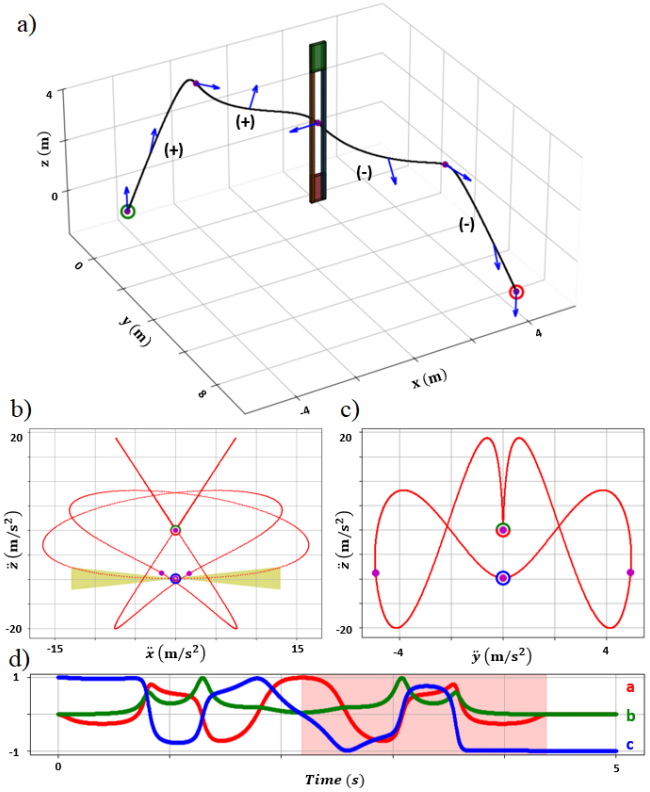

Fig. 8. Top: A trajectory with one mode switch passing through a narrow window. Middle Left: The trajectory $\ddot{x}$ vs $\ddot{z}$. The yellow zones highlight the constraints imposed by Eq.27 Middle Right: The trajectory $\ddot{y}$ vs $\ddot{z}$. Bottom The components of $\mathbf{b}_3$ over the duration of the trajectory.

at some predetermined point in time or simultaneously to a hybrid mode switch. A time-optimization algorithm such as [5] could be utilized to further refine the trajectory.

VII. Conclusion

The main contribution of this paper is a dynamically feasible trajectory planner for bidirectional quadrotors that utilizes the so-called free-fall singularity to smoothly transition the net thrust. The method relies on the property of hybrid differential flatness with four distinct modes corresponding to forward and reverse thrust and the choice of yaw parameterizations given by the Hopf fibration. We show how this method can be used to automatically synthesize trajectories that can transition the net thrust both with and without changing flight orientations by moving through the free-fall singularity. Further, we show how the specification of appropriate constraints can lead to aggressive trajectories through narrow window-like apertures. Our current work addresses the implementation of aggressive trajectories on platforms with bidirectional propellers. Because our planning strategies make full use of the portion of the flight envelope surrounding the free-fall singularity, we anticipate that the use of sensored speed controllers which eliminate the thrust deadband [8] around zero speed [22] will be an important requirement to achieve high performance flight.

REFERENCES

[1] P. Yu and K. Wong, "An implementation framework for vision-based bat-like inverted perching with bi-directionalthrust quadrotor," *International Journal of Micro Air Vehicles*, vol. 14, p. 17568293211073672, 2022. [Online]. Available: https://doi.org/10.1177/17568293211073672

[2] J. L. Paneque, J. R. M.-d. Dios, A. Ollero, D. Hanover, S. Sun, A. Romero, and D. Scaramuzza, "Perception-aware perching on powerlines with multirotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3077–3084, 2022.

[3] J. Bass, I. Tunney, and A. L. Desbiens, "Adaptative friction shock absorbers and reverse thrust for fast multirotor landing on inclined surfaces," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6701–6708, 2022.

[4] J. Bass and A. L. Desbiens, "Improving multirotor landing performance on inclined surfaces using reverse thrust," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5850–5857, 2020.

[5] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

[6] 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, 2018.

[7] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[8] W. Jothiraj, I. Sharf, and M. Nahon, "Control allocation of bidirectional thrust quadrotor subject to actuator constraints," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 932–938.

[9] W. Jothiraj, C. Miles, E. Bulka, I. Sharf, and M. Nahon, "Enabling bidirectional thrust for aggressive and inverted quadrotor flight," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 534–541.

[10] J. Wehbeh and I. Sharf, "An mpc formulation on $so(3)$ for a quadrotor with bidirectional thrust and nonlinear thrust constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4945–4952, 2022.

[11] M. Watterson, A. Zahra, and V. Kumar, "Geometric control and trajectory optimization for bidirectional thrust quadrotors," in *Proceedings of the 2018 International Symposium on Experimental Robotics*, J. Xiao, T. Kröger, and O. Khatib, Eds. Cham: Springer International Publishing, 2020, pp. 165–176.

[12] M. Maier, "Bidirectional thrust for multirotor mavs with fixed-pitch propellers," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[13] M. Watterson and V. R. Kumar, "Control of quadrotors using the hopf fibration on so(3)," in *ISRR*, 2017.

[14] J. Solà, "Quaternion kinematics for the error-state kf," 03 2015.

[15] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2216–2222.

[16] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor uav with a cable-suspended load," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 2269–2274.

[17] B. Morrell, M. Rigter, G. Merewether, R. Reid, R. Thakker, T. Tzanetos, V. Rajur, and G. Chamitoff, "Differential flatness transformations for aggressive quadrotor flight," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5204–5210.

[18] D. W. Lyons, "An elementary introduction to the hopf fibration," *Mathematics magazine*, vol. 76, no. 2, pp. 87–98, 2003.

[19] J. Welde, M. D. Kvalheim, and V. Kumar, "The role of symmetry in constructing geometric flat outputs for free-flying robotic systems," in *2023 IEEE International Conference on Robotics and Automation*, 2023.

[20] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[21] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2017.

[22] *23-06 220Kv Module Datasheet*, IQinetics Technologies Inc. (Vertiq), 2022, v1.2. [Online]. Available: https://www.vertiq.co/23-06-module