Journal of Intelligent & Robotic Systems (2022) 105: 16
https://doi.org/10.1007/s10846-022-01574-8

REGULAR PAPER

Generation of Dynamically Feasible Window Traversing Quadrotor Trajectories Using Logistic Curve

Saurabh Upadhyay[1] · **Thomas Richardson**[1,2] · **Arthur Richards**[1,2]

Received: 20 October 2020 / Accepted: 13 January 2022 / Published online: 7 May 2022

Abstract

This work considers dynamically feasible point-to-point trajectory generation problem for a quadrotor flying through a constrained planner region referred as window (narrow gap). A four parameter logistic (4PL) curve is investigated as a prospective candidate and closed-form conditions are derived on the 4PL design parameters to satisfy the window traversability and vehicle dynamic feasibility constraints. A hierarchical approach first computes a dynamically feasible design parameter set for decoupled trajectory components and then obtains a solution set satisfying 3-D axis-coupled window traversability conditions. Numerical examples with a comparative study are presented to validate the analytical findings that highlight the quick computation of the dynamically feasible window traversing trajectories in complex window scenarios.

Keywords Kinodynamic trajectory planning · Window traversal · Unmanned aerial vehicle · Motion planning

1 Introduction

Quadrotors are widely used in constrained environment operations such as search and rescue, structure inspection, and drone racing competitions due to their agile and fast maneuverability [1–4]. Pertaining to these applications, a quadrotor often requires to fly through arbitrary tilted windows, and hence, the window traversal problem is gaining interest among UAV researchers [5–11]. From a trajectory planning perspective, vehicle dynamic and input feasibility constraints, end-point reachability in a given time, and window traversal are the key challenges that a prospective planner should address. Rapid computation of the multiple trajectories is desirable for online operation.

1.1 Literature Review

A variety of approaches have been proposed for generating point-to-point dynamically feasible quadrotor trajectories. A hierarchical spatial and temporal planning approach generates the spatial paths from the geometric curves (lines [13], polynomials [14], splines [15]) in the first step and then time parameterized them to satisfy the vehicle dynamic and input constraints. A direct optimization based approach formulates the trajectory generation problem as an optimal control problem subjected to vehicle dynamics and input constraints [16–18]. An explicit numerical optimization based approach for a fixed-wing UAV was proposed in [19] where obstacle avoidance is handled by using policy iteration for a Markov decision process. The numerical optimization approaches offers a natural framework for constrained path generation but suffers from computational complexity. A third approach uses differential flatness property to map the trajectory planning problem into a lower

---

A version of the submitted paper appeared in the Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS'20), Athens, Greece.

This work is funded by the Engineering and Physical Sciences Research Council CASCADE Programme (ref EP/R009953/1).

✉ Saurabh Upadhyay
Saurabh.Upadhyay@cranfield.ac.uk

Thomas Richardson
thomas.richardson@bristol.ac.uk

Arthur Richards
arthur.richards@bristol.ac.uk

[1] Department of Aerospace Engineering, University of Bristol, Bristol, UK

[2] Bristol Robotics Laboratory, Bristol, UK

dimension curve-fitting problem where each curve represents the differentially flat quadrotor outputs: three trajectory components and a yaw angle [20–22]. The differential flatness approach based methods typically use time parameterized polynomial curve due to its simplicity in imposing the input feasibility constraints and pose the trajectory generation problem as a constrained optimal problem in the curve design parameter space.

Quadrotor window traversal problem from the perception and control perspective was addressed in [5]. Therein, a window traversing ballistic trajectory with constant control inputs was planned with a polynomial time approaching trajectory [22] which brings the quadrotor from its hovering position to the initial pose (position and control inputs) of the ballistic trajectory. Robust estimation, control and planning for the window traversal problem was considered in [6] wherein a collision-free orientation at a point on the window plane was imposed for generating a feasible polynomial time trajectory. A combination of deep learning and guidance theory was used in [7] for the window traversal. Therein, a robust estimate of the window center was obtained by a convolution neural network based algorithm, and quadrotor heading and roll commands were generated for flying through the window center along the line perpendicular to the window. The work of [8] used a deep network-based perception algorithm to estimate the window position and defined two waypoints such that the line connecting the waypoints passes through the window. Subsequently, the linear interpolation of all window waypoints was given as a reference path to a perception aware model predictive controller [23] for finding the feasible trajectory. An end-to-end reinforcement learning approach was proposed in [9] wherein the geometric tracking controller was replaced by a neural network-based policy controller and the trajectory planning of [5] was used for generating the feasible window traversing trajectories. The works of [5–9] primarily focus on the perception and control approaches, and generate a feasible trajectory passing through a fixed point on the window plane. This specific window point traversal approach restricts the number of solutions and in case of arbitrary tilted windows may require complex window traversing approaches.

Quadrotor trajectory generation in a known window scenario was addressed in [24]. Therein, a sequence of controllers namely, hover control, 3-D path following, and attitude control, was used to generate a sequence of controller parameterized segments, each defined with controller gains and a goal state. The problem of window traversal for a quadrotor with a cable-suspended load was addressed in [10] wherein, orthogonal collocation based optimization method was used for generating the feasible trajectory. In [11], dynamically feasible motion primitives were generated from the time parameterized polynomial curves and window traversability was checked by taking the intersection of the ellipsoid-shaped vehicle orientation with the window. Differential flatness based motion primitives generation is computationally inexpensive [22] and in case of window traversal, can cover the whole window plane rather than a point. Motivated from the motion primitive based approaches, this work explores the possibility of generating multiple feasible trajectories quickly by reducing the design parameters space with simple closed-form dynamic feasibility and window traversal conditions.

1.2 Contributions

In contrast to the existing literature, this work investigates the four parameter logistic (4PL) curve as a prospective candidate for the window traversal problem with given initial and final positions, arbitrary oriented window, and vehicle dynamic and input constraints. The 4PL curve based smooth path planning tool was recently introduced in [25] and applied in obstacle cluttered, passages, airspace restrictions, and parallel parking scenarios [26]. The 4PL curve has shape flexibility with just four design parameters and has significant computational advantages over other smooth curves (e.g. Bézier curves, B-splines, etc.) [25].

As the major contribution, the problem of [27] is revisited for arbitrary oriented static windows and a corridor based approach is proposed to determine the window traversability of 4PL trajectories. Vehicle input feasibility conditions (respecting speed, acceleration, and jerk limits) for the 4PL trajectory are derived in closed-form and a hierarchical trajectory planning method is proposed that generates dynamically feasible trajectories passing through the window. Performance of the proposed approach is tested in simple and specific window scenarios with extreme orientations and a comparative study is carried out to highlight the fast-computation capability.

The remainder of this paper is organized as follows: The window traversal problem is described in Section 2. Basics of the 4PL trajectory are reviewed in Section 3. The proposed trajectory generation approach with input feasibility and window traversal conditions is described in Section 4. Numerical examples including a comparative study with an optimization approach [20] are presented in Section 5. Section 6 provides the concluding remarks.

2 Problem Description

This section discusses the relationship between quadrotor motion and time parameterized curve based dynamic feasibility trajectories and states the window traversal problem. For details on the quadrotor working principles, the interested readers are referred to [28].

2.1 Quadrotor Motion in Terms of Time Parameterized Curves

Consider a quadrotor in an inertial frame with orthogonal axes (X^i, Y^i, Z^i) and a body frame with origin at the quadrotor center of mass, two body axes X^b and Y^b in the direction of vehicle front and left side, and the third axis Z^b normal to the vehicle plane as shown in Fig. 1. The quadrotor is modeled as a rigid body with six degrees of freedom where three degrees of freedom describe the linear translation motion of the quadrotor center of mass $\boldsymbol{x} = [x, y, z]^T$ in the inertial frame and the remaining three describe the vehicle attitude in the body frame. Assume that the quadrotor is sufficiently agile and equipped with an autopilot such that the thrust (mass-normalized force) f and angular rates $\boldsymbol{\omega}(\omega_x, \omega_y, \omega_z)$ can be considered as control inputs. The rigid body quadrotor motion is governed by [22]

$$\ddot{\boldsymbol{x}} = R\boldsymbol{e}_3 f + \mathbf{g} \tag{1}$$

$$\dot{R} = R\boldsymbol{\omega}_\times \tag{2}$$

where R is a proper orthogonal rotation matrix that converts a body frame vector to an inertial frame vector, $\mathbf{g} = [0, 0, -g]^T$ with g representing the gravitational acceleration, $\boldsymbol{\omega}_\times$ is a skew-symmetric matrix of vehicle angular rates $\boldsymbol{\omega}$ given as

$$\boldsymbol{\omega}_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

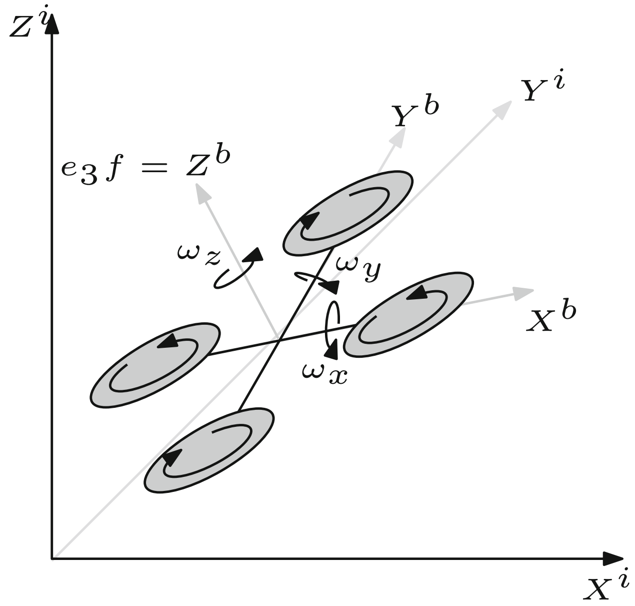


Fig. 1 A quadrotor schematic in an inertial frame (X^i, Y^i, Z^i) with a mass normalized thrust f along the body fixed axis Z^b and angular rates $\omega(\omega_x, \omega_y, \omega_z)$ in the body frame (X^b, Y^b, Z^b)

and $\boldsymbol{e}_3 = [0, 0, 1]^T$. Using Eqs. 1 and 2, the control inputs $(f, \omega_x, \omega_y, \omega_z)$ in terms of second and third derivatives of the trajectory can be deduced as

$$f = ||\ddot{\boldsymbol{x}} - \mathbf{g}|| \tag{3}$$

$$\begin{bmatrix} \omega_y \\ -\omega_x \end{bmatrix} = \frac{1}{f} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R^{-1} \dddot{\boldsymbol{x}} \tag{4}$$

where the absence of ω_z indicates that the yaw motion can be decoupled from the translation motion and hence $\omega_z = 0$ (constant yaw ψ) is considered in this work. Also, as discussed in [20], the vehicle dynamics and inputs of the quadrotor can be written in terms of four independent flat outputs x, y, z, ψ (yaw) and their derivatives. Hence, for dynamically feasible trajectory planning, typically the vehicle motion $\boldsymbol{x}$ can be decoupled in each axis separately by considering thrice differentiable time-parameterized curves in each axis, and the derivatives of curves (velocity $\boldsymbol{v} = \dot{\boldsymbol{x}}$, acceleration $\boldsymbol{a} = \ddot{\boldsymbol{x}}$, and jerk $\boldsymbol{j} = \dddot{\boldsymbol{x}}$) are used to obtain the control inputs from Eqs. 3 and 4. Also, the bounds on the inputs (thrust and angular rates) can be considered in terms of the trajectory acceleration and jerk bounds [11].

2.2 Problem Statement

Consider a window traversal problem with given initial position $\boldsymbol{i} = [x_i, y_i, z_i]^T$, final position $\boldsymbol{f} = [x_f, y_f, z_f]^T$, and a planar static window W as shown in Fig. 2. Without loss of generality, assume that the window has four vertices $W_1(x_1, y_2, z_4)$, $W_2(x_2, y_1, z_2)$, $W_3(x_3, y_4, z_3)$, and $W_4(x_4, y_3, z_1)$, such that $x_1 \le x_2 \le x_3 \le x_4$, $y_1 \le y_2 \le y_3 \le y_4$, and $z_1 \le z_2 \le z_3 \le z_4$. The body of the quadrotor is bounded by a sphere with center at $\boldsymbol{x}$ and radius r (depends on the quadrotor size) which can traverse through the window. With this, the concerned window traversability problem can be formally stated as follows: Find feasible quadrotor trajectories $\Gamma(t) := \{t \rightarrow \boldsymbol{x}(t) \in \mathbb{R}^3\}, t \in [t_i, t_f]$ such that

1. **(End-point reachability)** starts from $\boldsymbol{i}$ at $t = t_i$ and reaches $\boldsymbol{f}$ at $t = t_f$, that is, $\boldsymbol{x}(t_i) = \boldsymbol{i}$ and $\boldsymbol{x}(t_f) = \boldsymbol{f}$,
2. **(Input feasibility)** satisfy the given axis-wise bounds on velocity, acceleration and jerk $(v_{bnd}, a_{bnd}, j_{bnd})$, that is,

$$|\boldsymbol{v}(t)|_\infty \le v_{bnd} \tag{5}$$

$$|\boldsymbol{a}(t)|_\infty \le a_{bnd}, \text{ and} \tag{6}$$

$$|\boldsymbol{j}(t)|_\infty \le j_{bnd} \tag{7}$$

for all $t \in [t_i, t_f]$ which represent the vehicle input constraints.

3. **(Window traversal)** passes through the window W without colliding with the window edges, that is, $\Gamma(t_w) \in \tilde{W}$ for some $t_w \in [t_i, t_f]$, where $\tilde{W} :=$

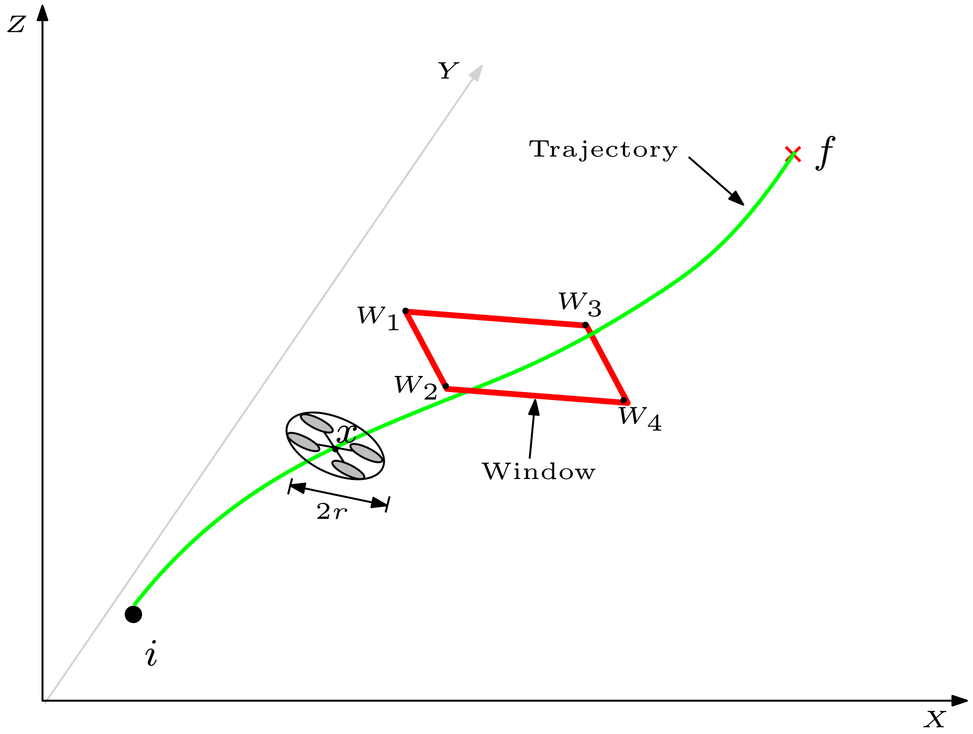


Fig. 2 A quadrotor flying through an arbitrary oriented Window $W(W_1, W_2, W_3, W_4)$. The quadrotor is bounded in a sphere of radius r and the feasible trajectory (green line) connects the end points i and f while satisfying the input feasibility and window traversal constraints

$W \ominus B(r)$ represent the window plane shrunk by the quadrotor sphere radius r.

It is desirable that quadrotor holds position at $\boldsymbol{f}$, that is, $\lim_{t\to\infty} \Gamma(t) = \boldsymbol{f}$ which could be considered simply as a safe terminal (basis) state in a broader receding horizon setting. Discussion on this receding horizon setting is out of the scope of this work and the interested readers are referred to [12] for details.

3 Basics of the 4PL Trajectory

A generic 4PL trajectory in 1-D is defined as

$$p(t; p_i, p_f, B, C) = p_f + \frac{p_i - p_f}{1 + \left(\frac{t-t_i}{C}\right)^B} \tag{8}$$

where t_i, p_i, and p_f represent the initial time, and initial and final values of trajectory, respectively, with $t \in [t_i, t_f]$. Two design parameters, $B > 3$ and $C > t_i$, that control the shift and slope of the 4PL trajectory. From Eq. 8, the 3-D 4PL trajectory components are given as

$$x = p(t; x_i, x_f, B_x, C_x) \tag{9}$$

$$y = p(t; y_i, y_f, B_y, C_y) \tag{10}$$

$$z = p(t; z_i, z_f, B_z, C_z) \tag{11}$$

Useful properties [25, 27] of the 4PL trajectory are as follows:

1. *End-point reachability:* From Eq. 8, the 4PL trajectory starts from the p_i at time $t = t_i$ and reaches p_f asymptotically. For a given t_f, a final position reachability with an acceptable ϵ error can be imposed as

$$\begin{aligned} &|p_f - p(t_f)| \le \epsilon \\ \implies\ & C \le (t_f - t_i)\left(\frac{\epsilon}{|p_i - p_f| - \epsilon}\right)^{\frac{1}{B}} \end{aligned} \tag{12}$$

2. *Continuous velocity, acceleration, and jerk variations:* The first, second, and third derivatives of Eq. 8 with respect to time, give component-wise velocity, acceleration, and jerk expressions for the 4PL trajectory as

$$v_p = \dot{p} = \frac{BC^B(p_f - p_i)(t-t_i)^{B-1}}{\left((t-t_i)^B + C^B\right)^2}, \tag{13}$$

$$a_p = \ddot{p} = \frac{BC^B(p_f - p_i)(t-t_i)^{B-2}}{\left((t-t_i)^B + C^B\right)^2}\left[(B-1) - \frac{2B(t-t_i)^B}{(t-t_i)^B + C^B}\right], \tag{14}$$

$$\begin{aligned} j_p = \dddot{p} = \frac{BC^B(p_f - p_i)\,(t-t_i)^{B-3}}{\left((t-t_i)^B + C^B\right)^2}\,&[(B-1)(B-2) - \\ &\frac{6B(B-1)(t-t_i)^B}{(t-t_i)^B + C^B} + \frac{6B^2(t-t_i)^{2B}}{((t-t_i)^B + C^B)^2}\Bigg], \end{aligned} \tag{15}$$

respectively which have continuous variations for $B > 3$.

3. *Position hold at the final position:* From Eqs. 12–15, it can be deduced that

   $$v_p = a_p = j_p \approx 0, \text{ for } t > t_f, \tag{16}$$

   which corresponds to constant force and zero angular rates and hence, position hold at the final point for $t > t_f$.
4. *Monotonicity* From Eq. 13 it can be seen that

   $$\dot{p} \begin{cases} > 0, \text{ for } p_i < p_f (\text{monotonically increasing}) \\ < 0, \text{ for } p_i > p_f (\text{monotonically decreasing}) \end{cases} \tag{17}$$

   and hence, the 4PL trajectory components vary monotonically with respect to t.
5. *Confinement region for the 4PL trajectories:* As discussed above, the 4PL trajectory components vary monotonically with time and hence, the 4PL trajectories always confined in a rectangular box region defined as

   $$\mathcal{R}_{4PL} := \{(x, y, z) | x \in [x_i, x_f], y \in [y_i, y_f], z \in [z_i, z_f]\} \tag{18}$$

4 Proposed Feasible Trajectory Generation Method

This section provides vehicle input feasibility and window traversal conditions for the 4PL trajectory, and proposes a feasible trajectory generation approach, subsequently.

4.1 Vehicle Input Constraints

This subsection derives input feasibility conditions satisfying Eqs. 5, 6, and 7, respectively for the 4PL trajectories.

4.1.1 Bounded Velocity

Equating Eq. 14 to zero gives

$$t_v^\star = t_i + C \left(\frac{B-1}{B+1} \right)^{\frac{1}{B}}$$

which upon substitution in Eq. 13 gives the component-wise velocity extremum $v_p^\star$ as

$$\max_{t \in [t_i, t_f]} v_p(t) = v_p^\star = \frac{(p_f - p_i)(B+1)^{\frac{1}{B}+1}}{4BC(B-1)^{\frac{1}{B}-1}} \tag{19}$$

Imposing the bounded velocity constraint of Eq. 5 on Eq. 19 leads to

$$|v_p^\star| \leq v_{bnd} \implies C \geq \left| \frac{(p_f - p_i)(B+1)^{\frac{1}{B}+1}}{4B v_{bnd} (B-1)^{\frac{1}{B}-1}} \right| \tag{20}$$

4.1.2 Acceleration Constraint

Now, the two time instants corresponding to the componentwise trajectory acceleration extrema can be computed by equating Eq. 15 to zero as

$$t_{a1,2}^\star = t_i + C \left(\frac{4(B-1)(B+1) \pm 2B\sqrt{3(B-1)(B+1)}}{2(B+1)(B+2)} \right)^{\frac{1}{B}}$$

which from Eq. 6 gives the bounded acceleration conditions

$$\max_{t \in [t_i, t_f]} |a(t)| = \max\{|a_p(t_{a1,2}^\star)|\} \leq a_{bnd}$$

$$\implies C^2 \geq \max_{i=1,2} \left\{ \left| \frac{(p_f - p_i)}{a_{bnd}} \frac{B k_i^{1-\frac{2}{B}}}{(1+k_i)^2} \left[\frac{(B-1) - k_i(B+1)}{1+k_i} \right] \right| \right\} \tag{21}$$

where,

$$k_1 = \frac{4(B-1)(B+1) + 2B\sqrt{3(B-1)(B+1)}}{2(B+1)(B+2)} \tag{22}$$

$$k_2 = \frac{4(B-1)(B+1) - 2B\sqrt{3(B-1)(B+1)}}{2(B+1)(B+2)} \tag{23}$$

4.1.3 Bounded Jerk Constraints

For jerk extrema, differentiating Eq. 15 with respect to t and equating to zero result in

$$(k_1 - k_2 + k_3 - k_4)T^3 - (k_2 - 2k_3 - 3k_4)T^2 + (k_3 - 3k_4)T - k_4 = 0 \tag{24}$$

where,

$$T = \left(\frac{t - t_i}{C} \right)^B$$

$$k_1 = 24B^3$$

$$k_2 = 36B^2(B-1)$$

$$k_3 = 2B(B-1)(7B-11)$$

$$k_4 = (B-1)(B-2)(B-3)$$

Three roots $\{k_{ji}\}_{i=1}^{3}$ of Eq. 24 can be obtained by using Cardano's method [29] on the depressed cubic form of Eq. 24 which is not presented here for the sake of brevity. Substituting $k_{j1,2,3}$ in Eq. 15 gives three jerk extrema

$$j_i^\star = \frac{(p_f - p_i)}{C^3} \frac{B k_{ji}^{1-\frac{3}{B}}}{(1+k_{ji})^2} \left[\frac{6B^2 k_{ji}^2}{(1+k_{ji})^2} - \frac{6B(B-1)k_{ji}}{(1+k_{ji})} + (B-1)(B-2) \right] \tag{25}$$

Using Eqs. 7 and 25, conditions for the bounded jerk can be obtained as

$$C^3 \geq \max_i \left\{ \left| \frac{(p_f - p_i)\hat{k}_{ji}}{j_{bnd}} \right| \right\} \tag{26}$$

where

$$\hat{k}_{ji} = \frac{Bk_{ji}^{1-\frac{3}{B}}}{(1+k_{ji})^2}\left[\frac{6B^2k_{ji}^2}{(1+k_{ji})^2} - \frac{6B(B-1)k_{ji}}{(1+k_{ji})} + (B-1)(B-2)\right] \quad (27)$$

4.2 Corridor-Based Window Traversal

Since the window is not assumed to be aligned with the inertial axes, the window traversibility conditions are coupled across the axes. To avoid the need for an exhaustive search through all combinations of axis-by-axis trajectories, this section proposes an approximation of the problem that admits an axis-by-axis search.

The key idea here is to define three axis-aligned rectangular cuboid corridors passing through the window and ensure trajectory traversal through at least one of them, that is,

$$(\exists T_x \subseteq [t_i, t_f] : \Gamma(t) \in \mathcal{C}_x, \forall t \in T_x) \vee (\exists T_y \subseteq [t_i, t_f] : \Gamma(t) \in \mathcal{C}_y, \forall t \in T_y) \vee (\exists T_z \subseteq [t_i, t_f] : \Gamma(t) \in \mathcal{C}_z, \forall t \in T_z) \quad (28)$$

where, $\mathcal{C}_x$, $\mathcal{C}_y$, and $\mathcal{C}_z$ represent the x, y, and z corridors, respectively, which are shown by the dashed boxes in Fig. 3a. The monotonic property (given by Eq. 17) of the 4PL trajectory is used to obtain the corridor traversal conditions which also guarantees that a corridor traversing 4PL trajectory cannot loop back outside the corridor and hence traverses window without collision. Without loss of generality, traversal conditions for $\mathcal{C}_x$ and hence, the window are derived in Proposition 1.

Proposition 1 *Consider a rectangular window W and an $x-$axis aligned corridor $\mathcal{C}_x$ passing through the window as shown in* Fig. 3 *by the solid red line and dotted black cube, respectively. Without loss of generality assume that window has four vertices (x_1, y_2, z_4), (x_2, y_1, z_2), (x_3, y_4, z_3), and (x_4, y_3, z_1), such that $x_1 \leq x_2 \leq x_3 \leq x_4$, $y_1 \leq y_2 \leq y_3 \leq y_4$, and $z_1 \leq z_2 \leq z_3 \leq z_4$. The corridor $\mathcal{C}_x$ is defined as*

$$\mathcal{C}_x := \{(x, y, z) | x \in [x_1 - r, x_4 + r], y \in [y_2 + r, y_3 - r], z \in [z_2 + r, z_3 - r]\} \quad (29)$$

where r represents the radius of the quadrotor boundary sphere. A 4PL quadrotor trajectory governed by Eq. 8 *is guaranteed to pass through the corridor and through the window for*

$$C_y \in (\underline{C}_{xy}^{wnd}, \bar{C}_{xy}^{wnd}) \quad (30)$$

$$C_z \in (\underline{C}_{xz}^{wnd}, \bar{C}_{xz}^{wnd}) \quad (31)$$

where

$$\underline{C}_{xy}^{wnd} = C_x \left(\frac{x_4 + r - x_i}{x_f - x_4 - r}\right)^{\frac{1}{B_x}} \left(\frac{y_f - y_3 + r}{y_3 - r - y_i}\right)^{\frac{1}{B_y}} \quad (32)$$

$$\bar{C}_{xy}^{wnd} = C_x \left(\frac{x_1 - r - x_i}{x_f - x_1 + r}\right)^{\frac{1}{B_x}} \left(\frac{y_2 + r - y_i}{y_f - y_2 - r}\right)^{\frac{1}{B_y}} \quad (33)$$

$$\underline{C}_{xz}^{wnd} = C_x \left(\frac{x_4 + r - x_i}{x_f - x_4 - r}\right)^{\frac{1}{B_x}} \left(\frac{z_f - z_3 + r}{z_3 - r - z_i}\right)^{\frac{1}{B_z}} \quad (34)$$

$$\bar{C}_{xz}^{wnd} = C_x \left(\frac{x_1 - r - x_i}{x_f - x_1 + r}\right)^{\frac{1}{B_x}} \left(\frac{z_2 + r - z_i}{z_f - z_2 - r}\right)^{\frac{1}{B_z}} \quad (35)$$

$B_x > 3$, $B_y > 3$, $B_z > 3$, $C_x > 0$, $C_y > 0$, $C_z > 0$, $t \in [t_i, t_f]$, $x_i < x_1 - r$, $x_4 + r < x_f$, $y_i < y_1 - r$, $< y_4 + r < y_f$, $z_i < z_1 - r$, and $z_4 + r < z_f$.

Proof The corridor $\mathcal{C}_x$ as defined by Eq. 29, has rectangular projections in the decoupled planes ($X - Y$ and $X - Z$) as shown in Fig. 3 by the dotted boxes. Corridor traversal is achieved by ensuring the traversal of rectangle projections by the respective 2-D 4PL trajectory components. These 2-D 4PL trajectory components are obtained by using Eq. 8 of the 4PL trajectory that leads to an inverted expression

$$t = t_i + C\left(\frac{p - p_i}{p_f - p}\right)^{\frac{1}{B}} \quad (36)$$

Equating the inverted expressions of the two 4PL trajectory components, one trajectory component can be reparameterized with respect to the second one. Accordingly, from Eqs. 9, 10, 11, and 36, the $X - Y$ and $X - Z$ 4PL trajectory components can be written as

$$y(x) = y_f + \frac{y_i - y_f}{1 + \left(\frac{C_x}{C_y}\left(\frac{x - x_i}{x_f - x}\right)^{\frac{1}{B_x}}\right)^{B_y}} \quad (37)$$

$$z(x) = z_f + \frac{z_i - z_f}{1 + \left(\frac{C_x}{C_z}\left(\frac{x - x_i}{x_f - x}\right)^{\frac{1}{B_x}}\right)^{B_z}} \quad (38)$$

The traversal of the $y(x)$ and $z(x)$ in the $X - Y$ and $X - Z$ corridor projections, respectively, guarantees window traversal as shown in Fig. 3 by the solid green lines. Due to the monotonic property of the 4PL curve, the 2-D trajectory components have monotonic variations in the decoupled

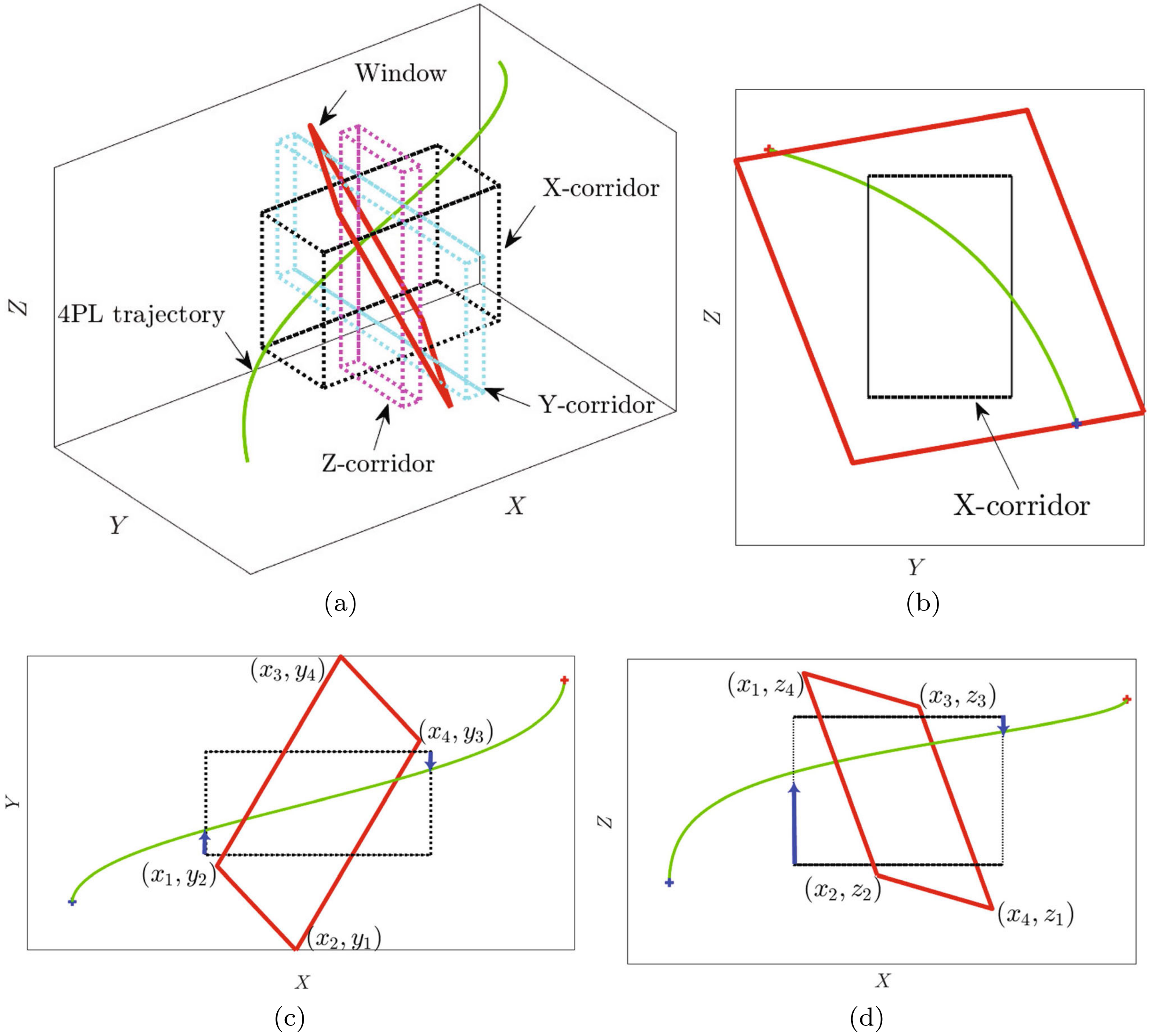


Fig. 3 Proposed window traversal approach defines three axis-aligned window-traversing corridors (dotted cuboids) and ensures that the feasible trajectory (solid green line) passes through at least one corridor (a). The $X-$axis corridor in $Y - Z$ plane (b), corridor traversal conditions (blue arrows) in $X - Y$ (c) and $X - Z$ planes (d) for the $X-$corridor with $\boldsymbol{i} < \boldsymbol{f}$

planes (please see Table 1) and hence, only two conditions (the blue arrows in Fig. 3c and d) need to be imposed for each plane projection to ensure the traversal. Using Eq. 37, the traversal conditions for the corridor projection in $X - Y$ plane are given as

$$y(x_4 + r) < y_3 - r \implies C_y < \underline{C}^{wnd}_{xy} = C_x \left(\frac{x_4 + r - x_i}{x_f - x_4 - r}\right)^{\frac{1}{B_x}} \times \left(\frac{y_f - y_3 + r}{y_3 - r - y_i}\right)^{\frac{1}{B_y}}$$

$$y(x_1 - r) > y_2 + r \implies C_y > \bar{C}^{wnd}_{xy} = C_x \left(\frac{x_1 - r - x_i}{x_f - x_1 + r}\right)^{\frac{1}{B_x}} \times \left(\frac{y_2 + r - y_i}{y_f - y_2 - r}\right)^{\frac{1}{B_y}}$$

Table 1 Monotonicity of the $y(x)$ and $z(x)$ 4PL trajectory components with respect to x

	$y_i < y_f$	$y_i > y_f$	$z_i < z_f$	$z_i > z_f$
$x_i < x_f$	Increasing	Decreasing	Increasing	Decreasing
$x_i > x_f$	Decreasing	Increasing	Decreasing	Increasing

Table 2 Window traversal conditions for the x corridor approach

	$x_i < x_f$	$x_i > x_f$
$y_i < y_f$	$\{y(x_1 - r) > y_2 + r\}$	$\{y(x_1 - r) < y_3 - r\}$
	$\{y(x_4 + r) < y_3 - r\}$	$\{y(x_4 + r) > y_2 + r\}$
$y_i > y_f$	$\{y(x_1 - r) < y_3 - r\}$	$\{y(x_1 - r) > y_2 + r\}$
	$\{y(x_4 + r) > y_2 + r\}$	$\{y(x_4 + r) < y_3 - r\}$
$z_i < z_f$	$\{z(x_1 - r) > z_2 + r\}$	$\{z(x_1 - r) < z_3 - r\}$
	$\{z(x_4 + r) < z_3 - r\}$	$\{z(x_4 + r) > z_2 + r\}$
$z_i > z_f$	$\{z(x_1 - r) < z_3 - r\}$	$\{z(x_1 - r) > z_2 + r\}$
	$\{z(x_4 + r) > z_2 + r\}$	$\{z(x_4 + r) < z_3 - r\}$

Similarly, using Eq. 38, the traversal conditions for the $X - Z$ corridor projection are obtained as

$$z(x_4 + r) < z_3 - r \implies C_z < \underline{C}^{wnd}_{xz} = C_x \left(\frac{x_4 + r - x_i}{x_f - x_4 - r}\right)^{\frac{1}{B_x}} \times \left(\frac{z_f - z_3 + r}{z_3 - r - z_i}\right)^{\frac{1}{B_z}}$$

$$z(x_1 - r) > z_2 + r \implies C_z > \bar{C}^{wnd}_{xz} = C_x \left(\frac{x_1 - r - x_i}{x_f - x_1 + r}\right)^{\frac{1}{B_x}} \times \left(\frac{z_2 + r - z_i}{z_f - z_2 - r}\right)^{\frac{1}{B_z}}$$

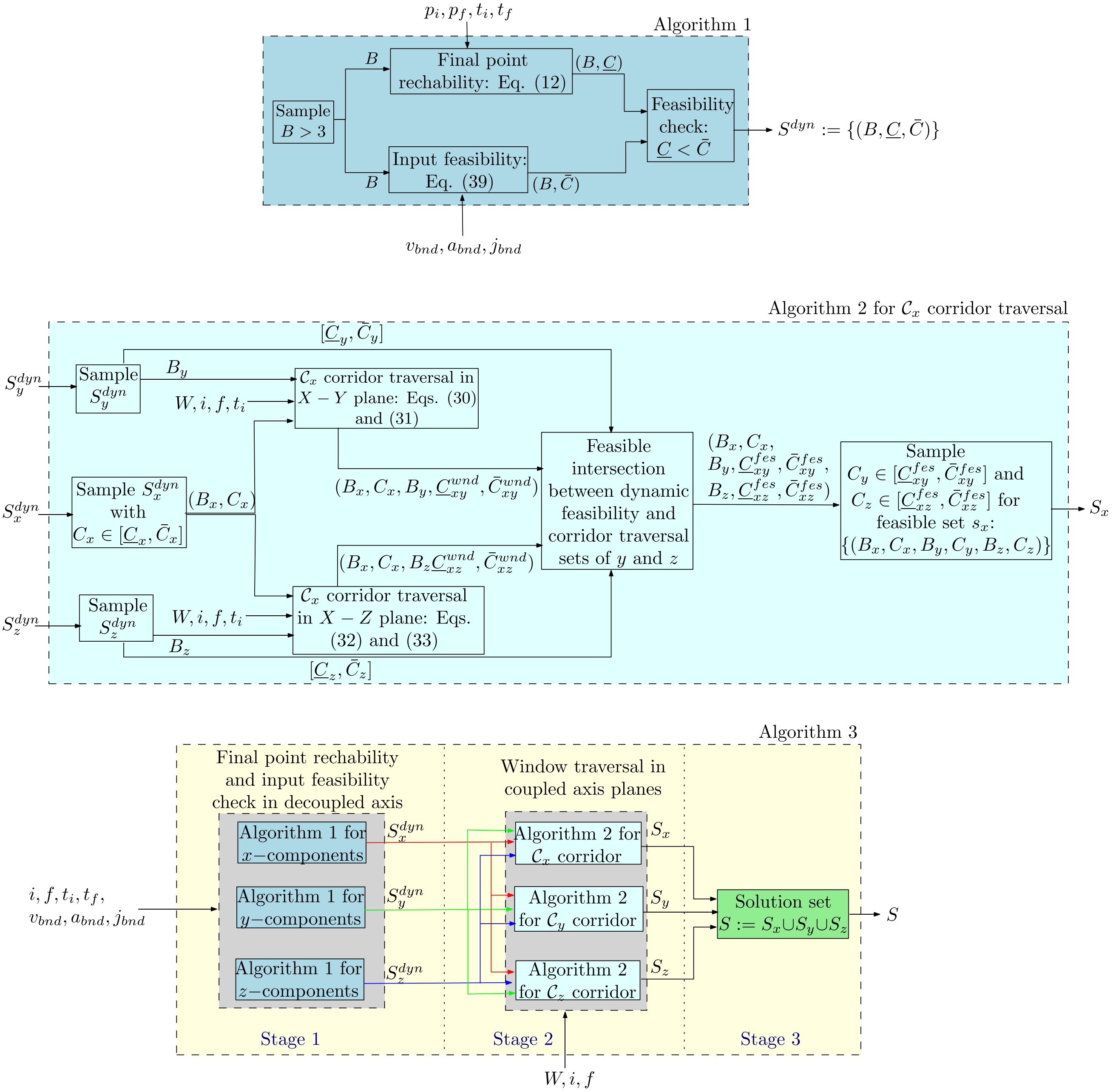

Algorithm 1
p_i, p_f, t_i, t_f
Sample $B > 3$
B
Final point rechability: Eq. (12)
$(B, \underline{C})$
Input feasibility: Eq. (39)
$(B, \bar{C})$
Feasibility check: $\underline{C} < \bar{C}$
$S^{dyn} := \{(B, \underline{C}, \bar{C})\}$
$v_{bnd}, a_{bnd}, j_{bnd}$
Algorithm 2 for $\mathcal{C}_x$ corridor traversal
$[\underline{C}_y, \bar{C}_y]$
S^{dyn}_y
Sample S^{dyn}_y
B_y
W, i, f, t_i
$\mathcal{C}_x$ corridor traversal in $X - Y$ plane: Eqs. (30) and (31)
S^{dyn}_x
Sample S^{dyn}_x with $C_x \in [\underline{C}_x, \bar{C}_x]$
(B_x, C_x)
$(B_x, C_x, B_y, \underline{C}^{wnd}_{xy}, \bar{C}^{wnd}_{xy})$
$(B_x, C_x, B_z \underline{C}^{wnd}_{xz}, \bar{C}^{wnd}_{xz})$
Feasible intersection between dynamic feasibility and corridor traversal sets of y and z
$(B_x, C_x, B_y, \underline{C}^{fes}_{xy}, \bar{C}^{fes}_{xy}, B_z, \underline{C}^{fes}_{xz}, \bar{C}^{fes}_{xz})$
Sample $C_y \in [\underline{C}^{fes}_{xy}, \bar{C}^{fes}_{xy}]$ and $C_z \in [\underline{C}^{fes}_{xz}, \bar{C}^{fes}_{xz}]$ for feasible set s_x: $\{(B_x, C_x, B_y, C_y, B_z, C_z)\}$
S_x
S^{dyn}_z
Sample S^{dyn}_z
W, i, f, t_i
B_z
$\mathcal{C}_x$ corridor traversal in $X - Z$ plane: Eqs. (32) and (33)
$[\underline{C}_z, \bar{C}_z]$
Algorithm 3
Final point rechability and input feasibility check in decoupled axis
Window traversal in coupled axis planes
$i, f, t_i, t_f,$
$v_{bnd}, a_{bnd}, j_{bnd}$
Algorithm 1 for $x-$components
Algorithm 1 for $y-$components
Algorithm 1 for $z-$components
S^{dyn}_x
S^{dyn}_y
S^{dyn}_z
Algorithm 2 for $\mathcal{C}_x$ corridor
Algorithm 2 for $\mathcal{C}_y$ corridor
Algorithm 2 for $\mathcal{C}_z$ corridor
S_x
S_y
S_z
Solution set $S := S_x \cup S_y \cup S_z$
S
Stage 1
Stage 2
Stage 3
W, i, f

Fig. 4 Block diagrams of the proposed algorithms: Algorithm 1 satisfying end point reachability and input feasibility (top), Algorithm 2 for a corridor traversal (middle), and the proposed 3-stage approach (bottom)

A set of design parameter satisfying above equations corresponds to the corridor traversing 4PL trajectories. As shown in Table 1, both 2-D trajectory components have monotonic variations and hence a corridor traversing trajectory cannot loop back outside the corridor which guarantees window traversal. □

Remarks on window traversal using the corridor approach are as follows:

1. Based on $\boldsymbol{i}$ and $\boldsymbol{f}$ values, and respective monotonic variations as discussed in Table 1, the traversal conditions for $\mathcal{C}_x$ are listed in Table 2.
2. Similar conditions can be imposed for $\mathcal{C}_y$ and $\mathcal{C}_z$ and check independently for the window traversal, where $\mathcal{C}_y$ and $\mathcal{C}_z$ are defined as

$$\begin{aligned}\mathcal{C}_y &:= \{(x,y,z)|x \in [x_2+r, x_3-r],\\ &\quad \times y \in [y_1-r, y_4+r], z \in [z_2+r, z_3-r]\}\\ \mathcal{C}_z &:= \{(x,y,z)|x \in [x_2+r, x_3-r],\\ &\quad \times y \in [y_2+r, y_3-r], z \in [z_1-r, z_4+r]\}\end{aligned}$$

3. Any window outside the 4PL confined region ($W \cap \mathcal{R}_{4PL} = \emptyset$), where $\mathcal{R}_{4PL}$ is given by Eq. 18, is not traversable by the 4PL trajectories. Also, a window only partially overlapping with the 4PL confined region($W \not\subset \mathcal{R}_{4PL}$) needs to be clipped to apply the window traversal conditions.

4.3 Computation of Solution Design Parameter Set

A hierarchical method for computing feasible solution set is described in Algorithm 3 and its block diagram is shown in Fig. 4. Here, the first stage considers final point reachability and input feasibility (Algorithm 1) followed by the window traversal (Algorithm 2). Details of the algorithms are given as follows:

4.3.1 Input Feasibility and Final Point Reachability

Algorithm 1 provides a design parameter set satisfying input feasibility and end-point reachability constraints for a 4PL trajectory component. Step 2 and 3 in Algorithm 1, sample a $B \in (3, \infty)$ and compute a lower bound on C such that a 4PL trajectory component generated with the sample B and a C greater than the lower bound satisfy all input feasibility conditions altogether. This lower bound is given as

$$\underline{C} = \max\{f_v, \sqrt{f_a}, \sqrt[3]{f_j}\} \tag{39}$$

where, f_v, f_a, and f_j are the right hand side terms of Eqs. 20, 21, and 26, respectively. From Eq. 12, an upper bound ($\bar{C}$) on C is computed in step 4 for the sample B that ensures the end point reachability of the 4PL trajectory component for the sample B and any $C \le \bar{C}$. A 4PL trajectory component with the sample B and any $C \in [\underline{C}, \bar{C}]$ satisfies the input feasibility and final point reachability constraints (Step 5). Hence, the feasible tuple $(B, \underline{C}, \bar{C})$ is stored in a feasible set S^{dyn} (Step 6). This procedure is shown in Fig. 4 by the light blue box which is repeated for generating the desired number of samples. Typically sample B is in a range of a few points increasing from 3. However, to minimize computation, it is possible to pick just a single value of B and still find many solutions. This trade will be explored further in the comparative study in Section 5.4. Algorithm 1 is used independently on the x, y, and z axis components to obtain the feasible sets $S_x^{dyn} := \{(B_x, \underline{C}_x, \bar{C}_x)\}$, $S_y^{dyn} := \{(B_y, \underline{C}_y, \bar{C}_y)\}$, and $S_z^{dyn} := \{(B_z, \underline{C}_z, \bar{C}_z)\}$, respectively.

Algorithm 1 TrajFeasibility: Feasible design parameter set satisfying final point reachability and input feasibility.

```
INPUT: Initial position $p_i$, final position $p_f$, input
    bounds $(f_{bnd}, f_{min}, \omega_{bnd})$, intial and final times $(t_i, t_f)$,
    reachability tolerance $\epsilon$, sample set $\mathcal{B} : \mathcal{B} \subset (3, \infty)$
OUTPUT: Feasible set $S^{dyn} := \{(B, \underline{C}, \bar{C})\}$
1: Initialize $S^{dyn} \leftarrow \emptyset$
2: for $B \in \mathcal{B}$ do
3:     $\underline{C} = \max\{f_v, \sqrt{f_a}, \sqrt[3]{f_j}\}$   ▷ $f_v$, $f_a$, and $f_j$ are the
    right hand side terms of Eqs. 20, 21, and 26,
    respectively
4:     $\bar{C} = (t_f - t_i)\left(\frac{\epsilon}{|x_i - x_f| - \epsilon}\right)^{\frac{1}{B_x}}$
5:     if $\underline{C} < \bar{C}$ then
6:         $S^{dyn} \leftarrow S^{dyn} \cup (B, \underline{C}, \bar{C})$
7:     end if
8: end for
```

4.3.2 Feasible Window Traversing Set

For sake of simplicity, Algorithm 2 only considers the window traversal through $\mathcal{C}_x$ corridor for $x_i < x_f$, $y_i < y_f$, and $z_i < z_f$ which is shown in Fig. 4 by the light cyan box. Window traversal through $\mathcal{C}_y$ and $\mathcal{C}_z$ corridors can be checked by changing the order of the input set in Algorithm 2. Also, any initial and final position conditions can be addressed by imposing respective corridor traversability conditions from Tables 1 and 2.

In step 2 to 5, Algorithm 2 selects three samples $(B_x, \underline{C}_x, \bar{C}_x) \in S_x^{dyn}$, $(B_y, \underline{C}_y, \bar{C}_y) \in S_y^{dyn}$, and $(B_z, \underline{C}_z, \bar{C}_z) \in S_z^{dyn}$ which correspond to multiple 4PL trajectories satisfying input feasibility and end-point reachability constraints for all $C_x \in [\underline{C}_x, \bar{C}_x]$, $C_y \in [\underline{C}_y, \bar{C}_y]$, and $C_z \in [\underline{C}_z, \bar{C}_z]$. Steps 6 and 7 of Algorithm 2 check if the $X - Y$ 4PL trajectory passes through the projection of the corridor into the X-Y plane and provides window traversing limits for C_y. Step 8 and 9 do the same for the $X - Z$ plane. In summary, all 4PL trajectories with $(B_x, C_x, B_y, C_y, B_z, C_z)$ where $C_y \in (\underline{C}_{xy}^{wnd}, \bar{C}_{xy}^{wnd})$ and $C_z \in (\underline{C}_{xz}^{wnd}, \bar{C}_{xz}^{wnd})$ pass the window through the corridor $\mathcal{C}_x$. However, these window traversing sets do not guarantee the input feasibility and end-point reachability in y and z axis.

In steps 10 to 12, the window traversal limits are combined, by intersection, with the respective dynamic feasibility limits $[\underline{C}_y, \bar{C}_y]$ and $[\underline{C}_z, \bar{C}_z]$. If any feasible trajectories remain (Step 13), meaning a set of parameters satisfying input feasibility, end-point reachability, and window constraints. The corresponding parameter set is added to the set of feasible solutions S^{fes} in Step 14. At the very end Steps 21 to 25 convert the set of intervals to set of 6−tuple parameter options by grid-based sampling.

4.3.3 Proposed Hierarchical Approach

In the first stage, Algorithm 3 initiates Algorithm 1 to compute the feasible design parameter sets S_x^{dyn}, S_y^{dyn}, and S_z^{dyn} in the decoupled axis (steps 1 to 3) that correspond to 4PL trajectory components satisfying component-wise input feasibility and final point reachability constraints. In steps 4 to 6, these sets are given to Algorithm 2 for corridors $\mathcal{C}_x$, $\mathcal{C}_y$, and $\mathcal{C}_z$ with rearranged order that provides the feasible window traversing sets S_x, S_y, and S_z, respectively. The last stage gives the solution set S by taking the union of all window-traversing sets (step 7). All design parameter pairs in S satisfy the end-point reachability, input feasibility, and window traversal constraints and hence multiple dynamically feasible window traversing trajectories can be generated simultaneously using Eqs. 9, 10, and 11 by choosing samples from S.

5 Numerical Examples

This section tests the performance of the proposed approach in three cases and provides a comparative study. The initial and final positions are at $\boldsymbol{i}(0, 0, 0)$ and $\boldsymbol{f}(5, 3, 3)$ which are shown in Fig. 5 by the blue and red "+" markers, respectively, and $t \in [0, 10]$. The dynamic feasibility constraints of $|v_{bnd}| = 5$ m/s, $|a_{bnd}| = 10$ m/s^2, and $|j_{bnd}| = 20$ m/s^3 are considered. Window and computation data for each case is provided in Table 3. For all simulations, $B \in [4, 10]$ with step size of one, C with step size of 0.1, $r = 0.045$ m (Crazyflie 2.0 quadrotor) and $\epsilon = 0.01$ are considered. All computations are performed on a 3.20 GHz Intel® Core™ i7-8770 CPU with 16 GB RAM and the proposed approach is implemented in MATLAB R2019a.

Algorithm 2 WindowCheck: Feasible deign parameter set satisfying $\mathcal{C}_x$ corridor traversal for $x_i < x_f$, $y_i < y_f$, and $z_i < z_f$.

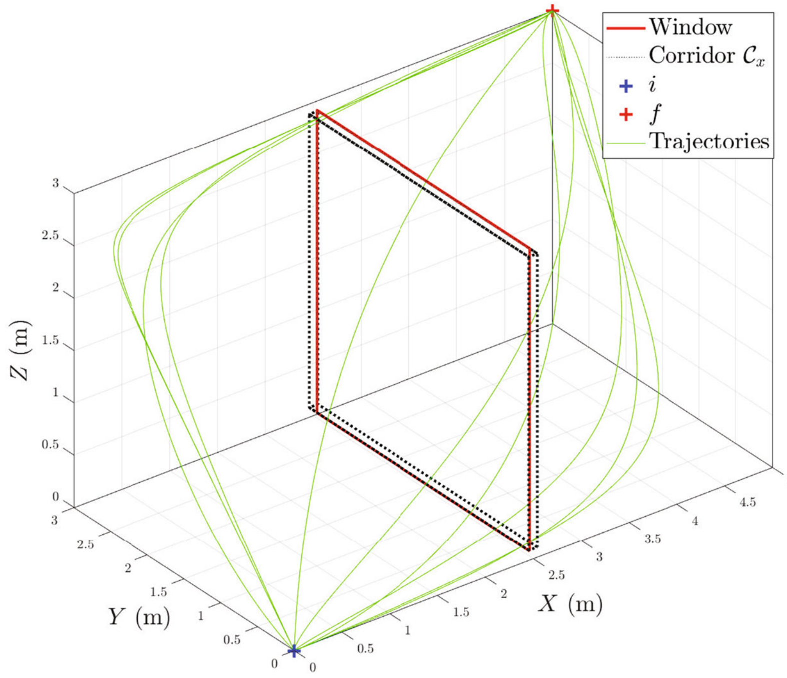
```
INPUT: $t_i, \boldsymbol{i}, \boldsymbol{f}, W, S_x^{dyn}, S_y^{dyn}, S_z^{dyn}$
OUTPUT: Design parameter set $S_x := \{(B_x, C_x, B_y, C_y, B_z, C_z)\}$
 1: Initialize $S_x^{fes} \leftarrow \emptyset$
 2: for $(B_x, \underline{C}_x, \bar{C}_x) \in S_x^{dyn}$ do
 3:     for $C_x \in (\underline{C}_x, \bar{C}_x)$ do
 4:         for $(B_y, \underline{C}_y, \bar{C}_y) \in S_y^{dyn}$ do
 5:             for $(B_z, \underline{C}_z, \bar{C}_z) \in S_z^{dyn}$ do
 6:                 $\underline{C}_{xy}^{wnd} = C_x \left(\frac{x_4+r-x_i}{x_f-x_4-r}\right)^{\frac{1}{B_x}} \left(\frac{y_f-y_3+r}{y_3-r-y_i}\right)^{\frac{1}{B_y}}$
 7:                 $\bar{C}_{xy}^{wnd} = C_x \left(\frac{x_1-r-x_i}{x_f-x_1+r}\right)^{\frac{1}{B_x}} \left(\frac{y_2+r-y_i}{y_f-y_2-r}\right)^{\frac{1}{B_y}}$
 8:                 $\underline{C}_{xz}^{wnd} = C_x \left(\frac{x_4+r-x_i}{x_f-x_4-r}\right)^{\frac{1}{B_x}} \left(\frac{z_f-z_3+r}{z_3-r-z_i}\right)^{\frac{1}{B_z}}$
 9:                 $\bar{C}_{xz}^{wnd} = C_x \left(\frac{x_1-r-x_i}{x_f-x_1+r}\right)^{\frac{1}{B_x}} \left(\frac{z_2+r-z_i}{z_f-z_2-r}\right)^{\frac{1}{B_z}}$
10:                 if $(\underline{C}_{xy}^{wnd} < \bar{C}_{xy}^{wnd})$ and $(\underline{C}_{xz}^{wnd} < \bar{C}_{xz}^{wnd})$ then
11:                     $\underline{C}_{xy}^{fes} = \max(\underline{C}_{xy}^{wnd}, \underline{C}_y)$, $\bar{C}_{xy}^{fes} = \min(\bar{C}_{xy}^{wnd}, \bar{C}_y)$
12:                     $\underline{C}_{xz}^{fes} = \max(\underline{C}_{xz}^{wnd}, \underline{C}_z)$, $\bar{C}_{xz}^{fes} = \min(\bar{C}_{xz}^{wnd}, \bar{C}_z)$
13:                     if $\underline{C}_{xy}^{fes} < \bar{C}_{xy}^{fes}$ and $\underline{C}_{xz}^{fes} < \bar{C}_{xz}^{fes}$ then
14:                         $S_x^{fes} \leftarrow S_x^{fes} \cup (B_x, C_x, B_y, \underline{C}_{xy}^{fes}, \bar{C}_{xy}^{fes}, B_z, \underline{C}_{xz}^{fes}, \bar{C}_{xz}^{fes})$
15:                     end if
16:                 end if
17:             end for
18:         end for
19:     end for
20: end for
21: Initialize $S_x \leftarrow \emptyset$
22: for all $(B_x, C_x, B_y, \underline{C}_{xy}^{fes}, \bar{C}_{xy}^{fes}, B_z, \underline{C}_{xz}^{fes}, \bar{C}_{xz}^{fes}) \in S_x^{wnd}$ do
23:     Sample $C_y \in (\underline{C}_{xy}^{fes}, \bar{C}_{xy}^{fes})$ and $C_z \in (\underline{C}_{xz}^{fes}, \bar{C}_{xz}^{fes})$
24:     $S_x \leftarrow S_x \cup (B_x, C_x, B_y, C_y, B_z, C_z)$
25: end for
```

Algorithm 3 SolutionSet: Computation of solution set.

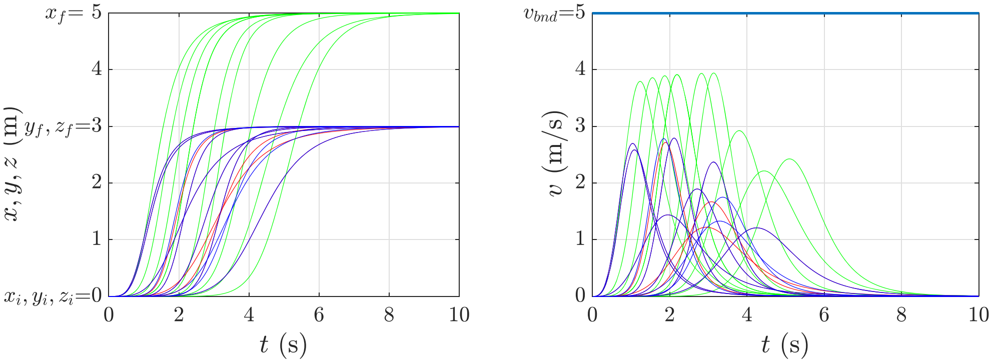
```
INPUT: Initial position $\boldsymbol{i}$, final position $\boldsymbol{f}$, input bounds
    $(f_{bnd}, f_{min}, \omega_{bnd})$, window $(W)$, intial and final times $(t_i, t_f)$,
    $\epsilon, n, d$
OUTPUT: Solution set $S := \{(B_x, C_x, B_y, C_y, B_z, C_z)\}$
 1: $S_x^{dyn} \leftarrow TrajFeasibility(x_i, x_f, f_{bnd}, f_{min}, \omega_{bnd}, t_i, t_f, \epsilon)$
 2: $S_y^{dyn} \leftarrow TrajFeasibility(y_i, y_f, f_{bnd}, f_{min}, \omega_{bnd}, t_i, t_f, \epsilon)$
 3: $S_z^{dyn} \leftarrow TrajFeasibility(z_i, z_f, f_{bnd}, f_{min}, \omega_{bnd}, t_i, t_f, \epsilon)$
 4: $S_x \leftarrow WindowCheck(t_i, \boldsymbol{i}, \boldsymbol{f}, W, S_x^{dyn}, S_y^{dyn}, S_z^{dyn})$
 5: $S_y \leftarrow WindowCheck(t_i, \boldsymbol{i}, \boldsymbol{f}, W, S_y^{dyn}, S_x^{dyn}, S_z^{dyn})$
 6: $S_z \leftarrow WindowCheck(t_i, \boldsymbol{i}, \boldsymbol{f}, W, S_z^{dyn}, S_x^{dyn}, S_y^{dyn})$
 7: $S \leftarrow \{S_x \cup S_y \cup S_z\}$   ▷ Reorder $S_y$ and $S_z$ before the operation
 8: if $S = \emptyset$ then
 9:     Report failure
10: end if
```

5.1 Case 1: Trivial Window Passing Scenario

A simple scenario with a large window is considered such that the all trajectory satisfying the final position

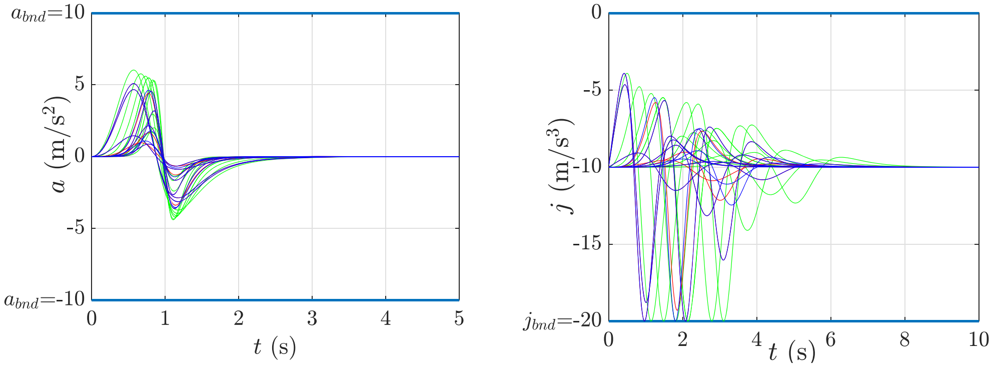


(a) Solution trajectories (green lines) traversing corridor $\mathcal{C}_x$ and window



(b) x (red), y (green), and z (blue) trajectory compoments satisfying end point reachability constraints

(c) Continuous and bounded velocity profiles



(d) Smooth bounded acceleration variations

(e) Bounded smooth jerk profiles

Fig. 5 Case 1: The proposed approach in a trivial window passing case where all solutions of Algorithm 3 Stage 1 (Fig. 4) satisfy $\mathcal{C}_x$ traversal conditions. Ten sample solutions are randomly selected from 11540 solutions and plotted

Table 3 Computation data for simulation results

Scenario	t_f	Window vertices	Solutions	Planning time
Case 1	10 s.	(2.5, 0.05, 0.05),(2.5, 2.95, 0.05),	11540	104 ms.
Trivial window traversal		(2.5, 2.95, 2.95), (2.5, 0.05, 2.95)		
No dynamic feasibility	5 s.		0	15 ms.
Case 2	10 s.	(2.51 1.43 1.43), (2.51 1.56 1.43),	6	81 ms.
Limited solutions		(2.48 1.56 1.56), (2.48, 1.43, 1.56)		
Case 3a	10 s.	(4.20,2.25,2.41),(4.28,2.68,2.58)	518	91 ms
		(3.79,2.77,2.58), (3.71,2.31,2.41)		
Case 3b	10 s.	(1.25, 1.87,0.78),(1.25,2.12,1.21)	1060	93 ms
		(0.75,2.12,1.21), (0.75,1.87,0.78)		
Case 3c	10 s.	(4.60.0.90,1),(4.60,1.10,1)	2858	104 ms
		(4.40,1.10,1), (4.40,0.90,1)		

reachability and input constraints satisfy the window traversal conditions. The window data is provided in Table 3 and plotted by the solid red rectangle in Fig. 5a. As the window is parallel to the $Y-Z$ plane, only $\mathcal{C}_x$ corridor exists for the given case which is plotted by the dashed black box in Fig. 5a. The proposed approach takes average 25 ms to provide the dynamically feasible design parameter set and provides 11540 solutions in total 104 ms. Ten such window traversing solutions are plotted by the solid green lines in Fig. 5a. Figure 5b shows that all trajectory components reach their final points within the $\epsilon = 0.01$ error band at the desired time $t_f = 10$ and hence satisfy the final point reachability constraint. The continuous bounded velocity, acceleration, and jerk profiles of the trajectory components are plotted in Fig. 5c, d, and e, respectively that satisfy the vehicle input constraints. The same setup is tested for $t_f = 5$ s where the proposed approach reports no solutions in 15 ms due to the vehicle input infeasibility.

5.2 Case 2: Limited Solution Scenario

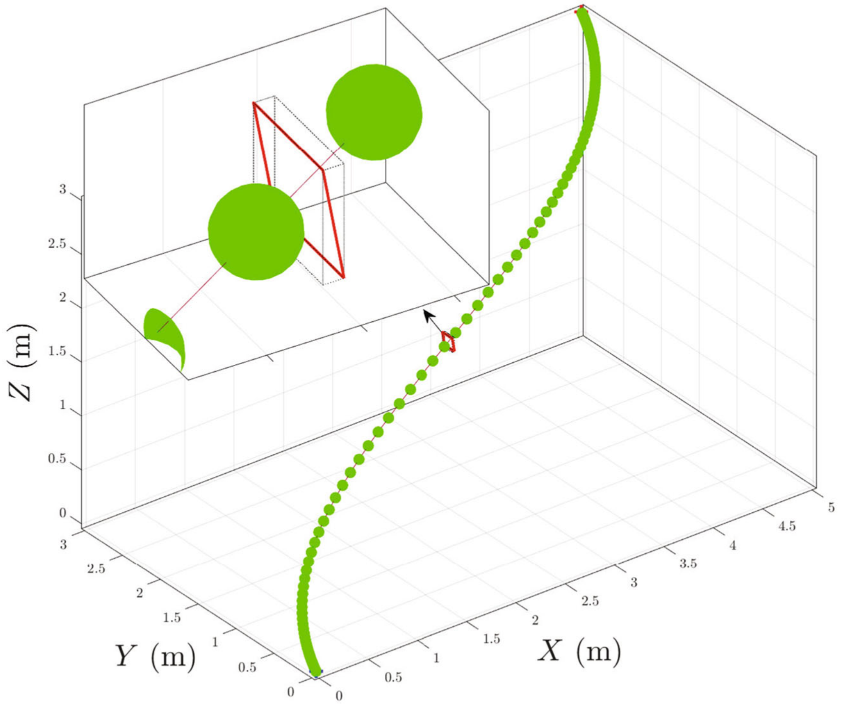
For exhaustive window traversal search, a window slightly bigger than the quadrotor is considered. Due to the narrow window size, the number of solutions reduced to six solutions and the proposed approach took 81 ms to find them. One such window traversing trajectory is plotted in Fig. 6a by the solid green line. The trajectory points are expanded by a sphere of $r = 0.045$ m radius and plotted by the green spheres in Fig. 6a which demonstrates that a finite dimension quadrotor can traverse the window collision-free. As shown in Fig. 6b, all trajectory components reach the end points in the given final time. Also, the generated trajectory has smooth and bounded velocity, acceleration and jerk variations as plotted in Figs. 6c, d, e, respectively.

5.3 Case 3: General Window Scenarios

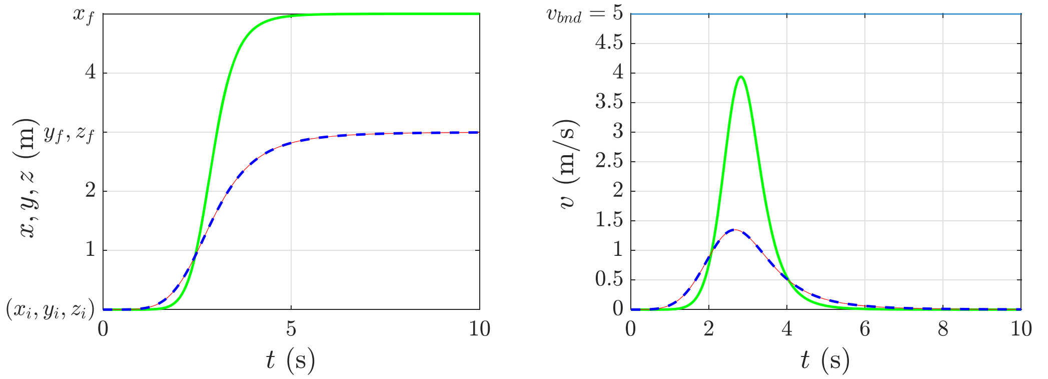
Three scenarios (Case 3a, 3b, and 3C) with three arbitrary oriented windows are considered to test the viability of the proposed approach in a general window scenario. All three windows are shown by the solid red rectangles in Fig. 7a and their data is provided in Table 3. The proposed approach results 518, 1060, and 2858 solutions in 91, 93, and 104 ms, for Case 3a, 3b, and 3C, respectively. Corresponding dynamically feasible window-traversing trajectories are plotted by the green, cyan, and gray lines, respectively in the Fig. 7. Figure 7a, b, and c show that the generated trajectories satisfy the end-point reachability constraint. Also, the velocity (Fig. 7e, f, and g), acceleration (Fig. 7h, i, and j) and jerk (Fig. 7k, l, and m) profiles of the generated trajectories have continuous bounded variation that satisfy the vehicle input feasibility. As shown in Fig. 8, all subcase trajectories maintenance a minimum distance grater than $r = 0.045$ m from window edges that validate window traversal without collision.

5.4 Comparative Study

The proposed approach is compared with a polynomial trajectory based approach [20] in randomly generated 1000 scenarios. The initial and final positions for each scenario are sampled such that $x_i \in [0, 1]$, $y_i \in [0, 1]$, $z_i \in [0, 1]$, $x_f \in [4, 5]$, $y_f \in [2, 3]$, and $z_f \in [2, 3]$. Final position reachability with $\epsilon = 0.2$ and zero velocity and acceleration at both ends are imposed. For windows traversal, random $X-$corridors with centroids at $(\boldsymbol{i} + \boldsymbol{f})/2$ and dimensions between 0 to 1 m are generated. Integral of square trajectory snap is considered as a cost and feasible solutions are generated using both approaches.

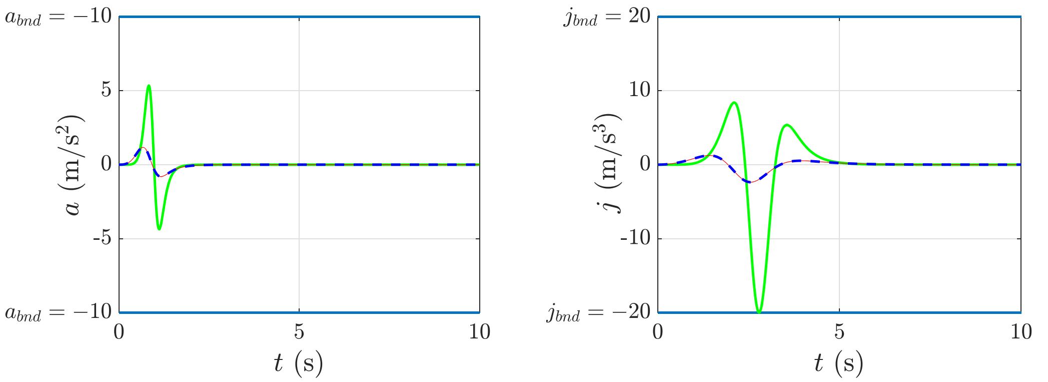


(a) A finite dimmenssion quadrotor passing though a narrow window



(b) x (red), y (green), and z (blue) trajectory compoments

(c) Continuous and bounded velocity profiles



(d) Smooth acceleration variation

(e) Bounded smooth jerk profile

Fig. 6 Case 2: Feasible trajectory generation for a quadrotor sphere passing through a small window

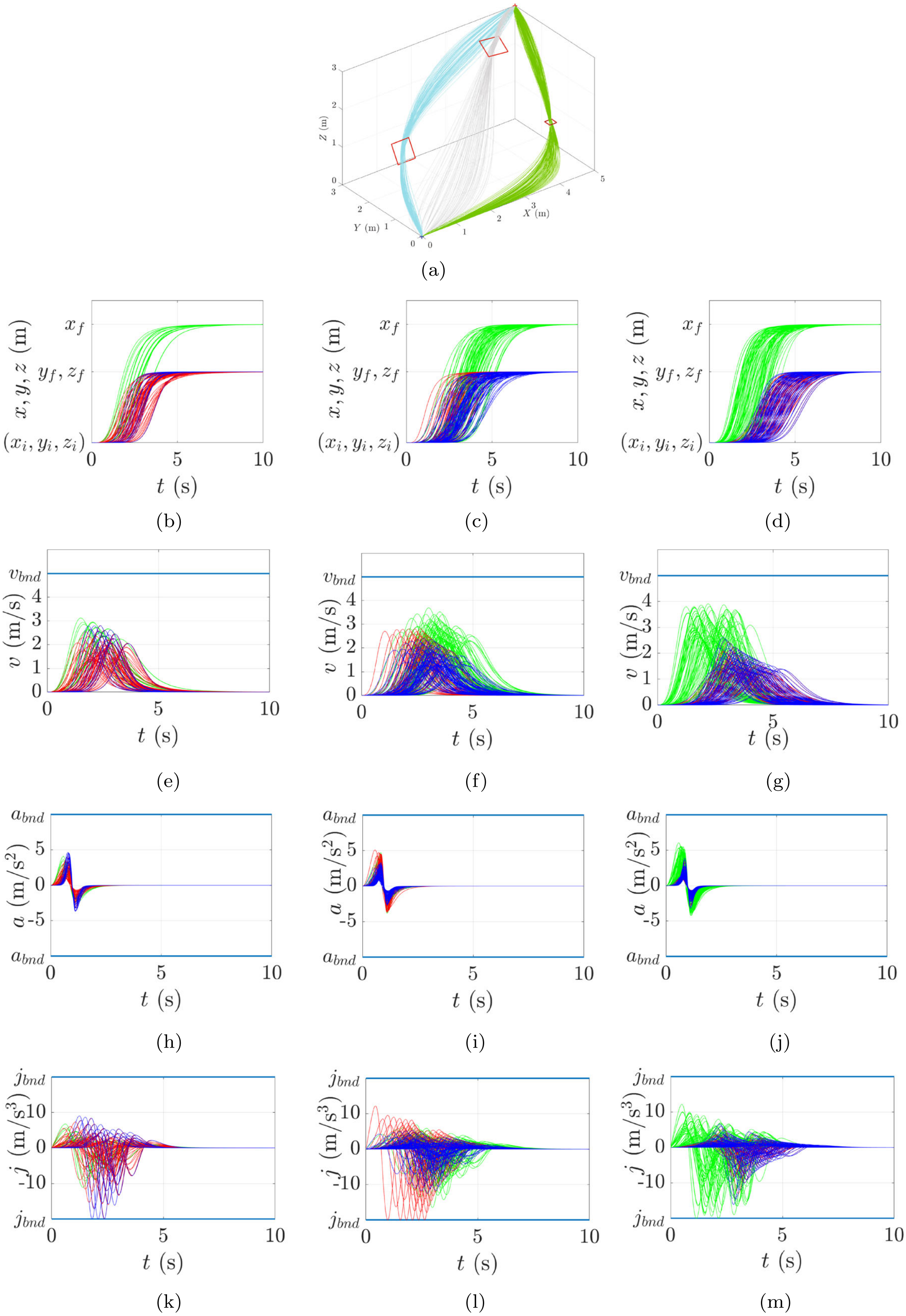


Fig. 7 Case 3: Feasible trajectories traversing three different window scenarios (**a**), their trajectory components (**b,c,d**), and component-wise smooth bounded velocity (**e,f,g**), acceleration (**h,i,j**), and jerk (**k,l,m**) profiles in Case 3a to Case 3c (left to right)

For the polynomial trajectory approach, one corridor constraint is imposed at both corridor ends (for more details, please see [20]) for corridor traversal. Time to compute a solution with respect to minimum cost is plotted in Fig. 9 by the red "+" markers for the polynomial approach in all scenarios which shows that the polynomial approach finds solutions for 997 scenarios in at least 5 ms per scenario. For the proposed 4PL approach, the sample set $\mathcal{B} = \{4, 5, 6, 7\}$ is considered first and using the empirical knowledge on the relation between 4PL jerk peak and snap value, a feasible trajectory with minimum peak jerk value is chosen from the obtained 4PL solutions. The cost of the chosen trajectory is computed numerically (MATLAB *trapz* function) for comparison. As shown in Fig. 9 by the green "+" markers, the proposed 4PL approach finds cost solutions in the approximate half time of polynomial approach with 172 fails and only a modest loss of minimum cost.

As discussed in Section 4.3.1, to exploit the minimum computation capability of the proposed approach, a sample set with only one B ($\mathcal{B} = \{7\}$) value is considered. In this case, the computation time reduces significantly with comparable minimum cost and 352 fails, as shown in Fig. 9 by the blue "+" markers. The trade-off between solution existence and computational efficiency can be observed from Fig. 9 where the polynomial approach provides solutions in the maximum number of scenarios and the proposed approach generates solutions quickly with a modest loss of path quality.

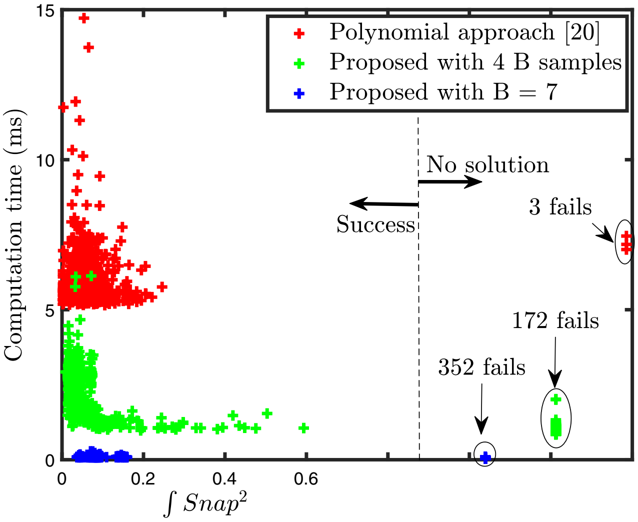


Fig. 9 Comparative study: Trade-off between solution existence and fast computation. The proposed is tested in 1000 randomly generated scenarios where it finds solutions in less time as compared to the polynomial approach [20], with a modest loss of path quality and existence

6 Conclusions

This work derived the closed-form input feasibility and window traversability conditions for the 4PL trajectory and proposed a hierarchical quadrotor trajectory planning method for the window traversal problem. The input feasibility conditions are decoupled in each axis and the

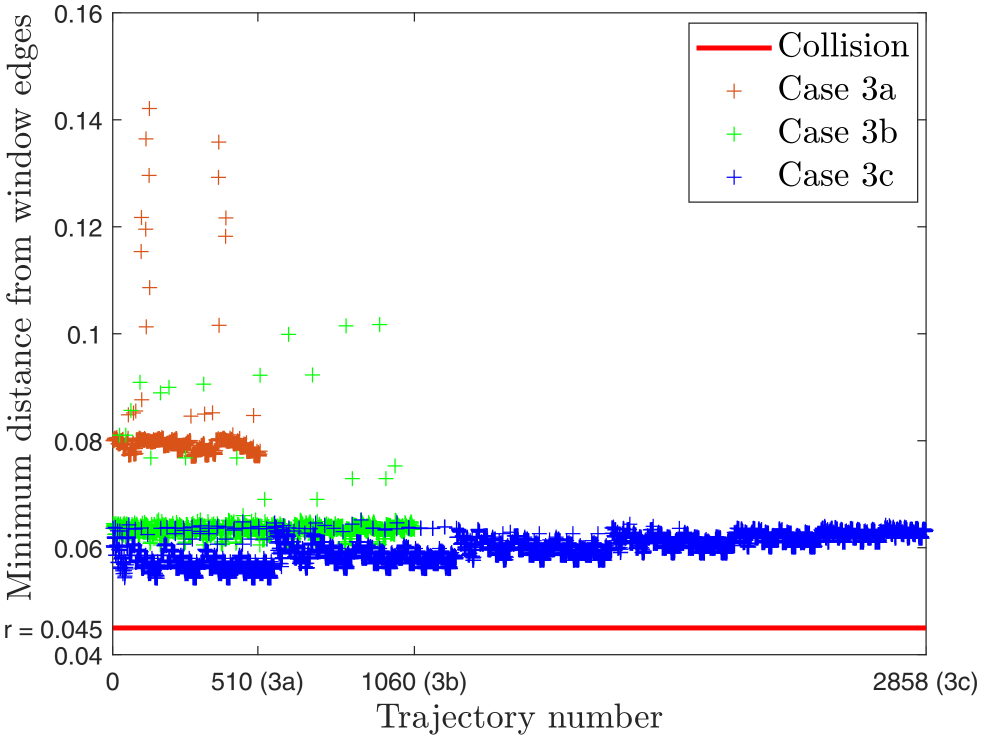


Fig. 8 Window traversal without collision: Minimum distances between all Case 3 trajectories and window edges are greater than the collision distance $r = 0.045$ m (quadrotor sphere radius)

window traversal conditions are coupled in 2-D planes which targets a window region rather than a specific point. Using these simple conditions, the proposed hierarchical approach initializes with only three design parameters and reduces the search space after each step that decreases the computation cost. Numerical examples demonstrate that the proposed 4PL approach is capable of providing thousands of solutions in a fraction of second for an arbitrary oriented window. The detailed comparative study provides the trade-off between the solution existence and fast computations, and highlights computational efficiency of the proposed approach in terms of fast computations for only modest loss of path quality.

References

1. Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., Gambardella, L.M., Siegwart, R., Scaramuzza, D.: The current state and future outlook of rescue robotics. J. Field Robot. **36**, 1171–1191 (2019). https://doi.org/10.1002/rob.21887
2. Moon, H., Martinez-Carranza, J., Cieslewski, T., Faessler, M., Falanga, D., Simovic, A., Scaramuzza, D., Li, S., Ozo, M., Wagter, C.D., Croon, G.D., Hwang, S., Jung, S., Shim, H., Kim, H., Park, M., Au, T.C., Kim, S.J.: Challenges and implemented technologies used in autonomous drone racing. Intell. Serv. Robot. **12**, 137–148 (2019). https://doi.org/10.1007/s11370-018-00271-6
3. Birk, A., Wiggerich, B., Bülow, H., Pfingsthorn, M., Schwertfeger, S.: Safety, security, and rescue missions with an unmanned aerial vehicle. J. Intell. Robot. Syst. **64**, 57–76 (2011). https://doi.org/10.1007/s10846-011-9546-8
4. Sampedro, C., Rodriguez-Ramos, A., Bavle, H., Carrio, A., Puente, P.D.L., Campoy, P.: A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. J. Intell. Robot. Syst. **95**, 601–627 (2019). https://doi.org/10.1007/s10846-018-0898-1
5. Falanga, D., Mueggler, E., Faessler, M., Scaramuzza, D.: Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In: IEEE International Conference on Robotics and Automation, pp. 5774–5781. https://doi.org/10.1109/ICRA.2017.7989679 (2017)
6. Loianno, G., Brunner, C., McGrath, G., Kumar, V.: Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. IEEE Robot. Autom. Lett. **2**, 404–411 (2017). https://doi.org/10.1109/LRA.2016.2633290
7. Jung, S., Hwang, S., Shin, H., Shim, D.H.: Perception, guidance, and navigation for indoor autonomous drone racing using deep learning. IEEE Robot. Autom. Lett. **3**, 2539–2544 (2018)
8. Kaufmann, E., Gehrig, M., Foehn, P., Ranftl, R., Dosovitskiy, A., Koltun, V., Scaramuzza, D.: Beauty and the beast: Optimal methods meet learning for drone racing. In: International Conference on Robotics and Automation, pp. 690–696 (2019)
9. Lin, J., Wang, L., Gao, F., Shen, S., Zhang, F.: Flying through a narrow gap using neural network: An end-to-end planning and control approach. In: IEEE/RSJ International Conference on Intell, Robots and Systems, pp. 3526–3533 (2019)
10. Guo, M., Gu, D., Zha, W., Zhu, X., Su, Y.: Controlling a quadrotor carrying a cable-suspended load to pass through a window. J. Intell. Robot. Syst. **98**, 387–401 (2020)
11. Liu, S., Mohta, K., Atanasov, N., Kumar, V.: Search-based motion planning for aggressive flight in SE(3). IEEE Robot. Autom. Lett. **3**, 2439–2446 (2018). https://doi.org/10.1109/LRA.2018.2795654
12. Schouwenaars, T., How, J., Feron, E.: Receding horizon path planning with implicit safety guarantees. In: Proc. of the 2004 American Control Conference, pp. 5576–5581. https://doi.org/10.23919/ACC.2004.1384742 (2004)
13. Hoffmann, G.M., Waslander, S.L., Tomlin, C.J.: Quadrotor helicopter trajectory tracking control. In: AIAA Guidance, Navigation and Control Conference and Exhib. AIAA, pp. 2008–7410. https://doi.org/10.2514/6.2008-7410 (2008)
14. Cowling, I.D., Yakimenko, O.A., Whidborne, J.F., Cooke, A.K.: A prototype of an autonomous controller for a quadrotor UAV. In: European Control Conference, pp. 4001–4008. https://doi.org/10.23919/ecc.2007.7068316 (2007)
15. Bouktir, Y., Haddad, M., Chettibi, T.: Trajectory planning for a quadrotor helicopter. In: Mediterranean Conference on Control and Automation, pp. 1258–1263. https://doi.org/10.1109/MED.2008.4602025 (2008)
16. Hehn, M., Ritz, R., D'Andrea, R.: Performance benchmarking of quadrotor systems using time-optimal control. Auton. Robot. **33**, 69–88 (2012). https://doi.org/10.1007/s10514-012-9282-3
17. Geisert, M., Mansard, N.: Trajectory generation for quadrotor based systems using numerical optimal control. In: IEEE International Conference on Robotics and Automation, pp. 2958–2964. https://doi.org/10.1109/ICRA.2016.7487460 (2016)
18. Neunert, M., Crousaz, C.D., Furrer, F., Kamel, M., Farshidian, F., Siegwart, R., Buchli, J.: Fast nonlinear model predictive control for unified trajectory optimization and tracking. In: IEEE International Conference on Robotics and Automation, pp. 1398–1404. https://doi.org/10.1109/ICRA.2016.7487274 (2016)
19. Yu, X., Zhou, X., Zhang, Y.: Collision-free trajectory generation and tracking for uavs using Markov decision process in a cluttered environment. J. Intell. Robotics Systems **93**, 17–32 (2019). https://doi.org/10.1007/s10846-018-0802-z
20. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: IEEE International Conference on Robotics and Automation, pp. 2520–2525. https://doi.org/10.1109/ICRA.2011.5980409 (2011)
21. Hehn, M., D'Andrea, R.: Quadrocopter trajectory generation and control. IFAC Proc. **44**, 1485–1491 (2011). https://doi.org/10.3182/20110828-6-IT-1002.03178
22. Mueller, M.W., Hehn, M., D'Andrea, R.: A computationally-efficient motion primitive for quadrocopter trajectory generation. IEEE Trans. Robot. **31**, 1294–1310 (2015). https://doi.org/10.1109/TRO.2015.2479878
23. Falanga, D., Foehn, P., Lu, P., Scaramuzza, D.: PAMPC: Perception-aware model predictive control for quadrotors. In: IEEE/RSJ International Conference on Intell, Robots and Systems, pp. 5200–5207 (2018)

24. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. Int. J. Robot. Res. **31**, 664–674 (2012). https://doi.org/10.1177/0278364911434236
25. Upadhyay, S., Ratnoo, A.: Continuous-curvature path planning with obstacle avoidance using four parameter logistic curves. IEEE Robot. Autom. Lett. **1**, 609–616 (2016). https://doi.org/10.1109/LRA.2016.2521165
26. Upadhyay, S.: Continuous-curvature path planning using four parameter logistic curves. Ph.D. Thesis, Indian Institute of Science (2018)
27. Upadhyay, S., Richards, A., Richardson, T.: Generation of window-traversing flyable trajectories using logistic curve. In: International Conference on Unmanned Aircr. Systems, pp. 59–65. https://doi.org/10.1109/ICUAS48674.2020.9214060 (2020)
28. Quan, Q.: Introduction to Multicopter Design and Control. Springer, Singapore (2017)
29. Cardano's Method. Brillient.org. https://brilliant.org/wiki/cardano-method/. Accessed: 03 October 2020 (2020)

Saurabh Upadhyay received the B.E. degree from Shri Sant Gajanan Maharaj College of Engineering, Shegaon in 2009, the M.Tech. degree from the Indian Institute of Technology Guwahati in 2012, and the Ph.D. degree from the Indian Institute of Science Bengaluru in 2018. He was a Postdoctoral Researcher at University of Porto (FEUP) in 2018-2019 and research associate at University of Bristol in 2019-2022. He is currently working as a Lecture in Space Engineering at Centre for Autonomous and Cyber-Physical Systems, Cranfield University, Cranfield, UK. His research interests include motion planning and control, robot autonomy, and mobile robots.

Thomas Richardson is a professor of Aerial Robotics at the University of Bristol. With a PhD in nonlinear control system design, he specializes in the application of modern control theory and novel sensors to Uncrewed Air Systems (UAS). Tom is also a founding partner of Perceptual Robotics which has recently been awarded 'Robotics & AI in Extreme Environments' funding by Innovate UK.

Arthur G. Richards received the M.Eng. degree from Cambridge University in 2000 and the SM and PhD degrees from MIT in 2002 and 2004, respectively. Since 2004, he has been with the Department of Aerospace Engineering at the University of Bristol, Bristol, U.K. He is now Professor of Robotics and Control and also with the Bristol Robotics Laboratory. His research interests include trajectory optimization, model predictive control, robotics, and their combination to develop high-performance guidance for autonomous vehicles.