



Collision Avoidance through Path Replanning using Bézier Curves*

Syed Bilal Mehdi[†], Ronald Choe[‡], Venanzio Cichella[§], Naira Hovakimyan[¶]

University of Illinois at Urbana-Champaign, Urbana, IL 61801

This paper introduces a collision prediction and avoidance algorithm for multi-vehicle time-critical cooperative missions. The algorithm extends our previous results on collision avoidance based on speed adjustment and allows for safe operation in a wider range of collision scenarios. Unlike the speed-adjustment method, the new algorithm changes the shape of the vehicle's planned trajectory by adding an appropriate detour. We show that the bounds on the changes in position, velocity and acceleration, as a result of the path replanning, are computable and depend on the collision scenarios under consideration. Hence, these bounds can be used during the mission-planning phase, to ensure that the replanned path satisfies the dynamic constraints of the vehicles.

I. Introduction

By the year 2025, highly integrated operations are envisioned under the NextGen Air Transportation System (NextGen) [1] where multiple capability manned and unmanned vehicles will be sharing the National Airspace System (NAS). To make this vision possible, operation of Unmanned Aerial Vehicles (UAVs) needs to be simplified enough for a human to control multiple UAVs at the same time. Furthermore, with human life on the line, ensuring safety for this scenario is of paramount importance.

Remotely piloted vehicles have several algorithms onboard that alleviate the workload of the operator. For example, guidance algorithms along with an autopilot allow autonomous path following in the presence of disturbances. Furthermore, algorithms that allow coordination for multiple-vehicle missions exist in the literature [2-10]. However, one challenging issue is the generation of trajectories for vehicles to track. This is not a trivial problem as these trajectories need to satisfy different dynamic constraints of the aircraft under consideration. For example, fixed-wing UAVs have a minimum turning radius which puts a constraint on the shape of these trajectories. Namely, these trajectories need to have a bounded and continuous curvature. Furthermore, all aircraft have a maximum speed and acceleration which constrain the temporal properties of the trajectories, i.e. these trajectories need to have a bounded speed and acceleration.

During the trajectory-generation process, an issue of consequence in checking such constraints is the computational load. To alleviate this, Bézier curves have been found useful as a choice for trajectory representation [11-17]. Here, efficient algorithms specific to Bézier curves, such as the de Casteljau and minimum distance calculation algorithm, allow for fast computation of bounds on, for example, vehicles' velocities and accelerations.

We described our *offline* trajectory-generation algorithm in [16, 17] where we used Bézier curves to represent trajectories. However, for safe operation of vehicles, it is important for vehicles to modify trajectories *online* in order to avoid obstacles. To this end, we presented a collision-avoidance algorithm based on speed adjustment in [18]. Although this algorithm has the advantage of being simple, its application is limited to a small set of collision scenarios. In fact, the algorithm surely fails if an unforeseen stationary obstacle blocks a vehicle's path. Therefore, in this paper we present a collision-avoidance algorithm based on trajectory replanning that integrates well into our cooperative path-following framework. Furthermore, with

*This research is supported by NASA.

[†]Doctoral Student, Dept. of Mechanical Science and Engineering, Student Member AIAA; mehdi1@illinois.edu.

[‡]Doctoral Student, Dept. of Aerospace Engineering, Student Member AIAA; choe19@illinois.edu.

[§]Doctoral Student, Dept. of Mechanical Science and Engineering, Student Member AIAA; cichell2@illinois.edu.

[¶]Professor, Dept. of Mechanical Science and Engineering, Associate Fellow AIAA; nhovakim@illinois.edu.

this approach, both moving and stationary obstacles can be avoided by taking a detour away from them. In our overall collision-avoidance framework, this algorithm is used whenever collision avoidance through speed adjustment is not feasible [18].

For our algorithm, we assume that an obstacle's trajectory can be predicted and represented as a Bézier curve upon its detection. This allows a simple subtraction of the obstacle's trajectory from the vehicle's own trajectory to provide with the separation Bézier curve - a curve representing the separation vector between the obstacle and the vehicle as a function of time. With such a representation, we can exploit properties of Bézier curves to efficiently check for an imminent collision. In the case of such a collision, we find a detour Bézier curve through a computationally inexpensive method, allowing for onboard implementation.

In addition, the detour Bézier curve can also be written in a closed analytical form and depends on: (i) the difference between obstacle detection and collision time; (ii) time interval where the minimum distance between the vehicle and the obstacle is violated; (iii) the difference between collision and mission end time; (iv) the safety distance required between the vehicle and the obstacle; and (v) the design parameters of the collision-avoidance algorithm. Using this analytical form of solution, and given some practical constraints on possible collision scenarios (such as the obstacle's maximum speed, obstacle detection radius for the vehicle etc.), we can calculate bounds on the changes in position, velocity and acceleration caused by the detour. This is useful in practice as such a detour can be made to stay within vehicle's dynamic constraints by providing sufficient margins for position, velocity and acceleration changes during the trajectory-generation phase.

In this paper, we also provide an illustrative example where required margins for position, velocity and acceleration changes are calculated using assumptions on vehicle's speed along with obstacle's size and speed. Using these margins, we generate trajectories with methods already presented in [16,17]. Finally, we illustrate the efficacy of our online collision-avoidance algorithm to add a dynamically feasible detour to the vehicle's path in order to avoid obstacles.

The paper is organized as follows: in Section II, we provide the background for the algorithm and discuss our trajectory-generation methodology. Section III describes the problem under consideration followed by Section IV where we introduce our algorithm. Analysis and performance guarantees for the proposed methodology are provided in Section V. Lastly, we present simulation examples in Section VI and end the paper with conclusions and future work in Section VII.

II. Background: Offline Trajectory Generation

We presented our trajectory-generation algorithm in [16,17] where we used Pythagorean Hodograph Bézier curves for trajectory parameterization. In this section, we will briefly overview our trajectory-generation algorithm in terms of its functionality. The discussion that follows will also serve as motivation for the proposed collision-avoidance scheme. Since our trajectory-generation algorithm uses Bézier curves for representation, we start this section by introducing these curves in the following.

II.A. Bézier Curves

Bézier curves are polynomial curves defined over the finite interval $[0, 1]$ by their control points $\bar{\mathbf{r}}_k$ as

$$\mathbf{r}(\zeta) = \sum_{k=0}^n \bar{\mathbf{r}}_k b_k^n(\zeta), \quad \zeta \in [0, 1], \quad (1)$$

where n is the degree of the polynomial and $b_k^n(\zeta)$ are the Bernstein polynomials defined as

$$b_k^n(\zeta) = \binom{n}{k} (1 - \zeta)^{n-k} \zeta^k, \quad \zeta \in [0, 1]. \quad (2)$$

Bézier curves exhibit several useful properties. Using a general Bézier curve $\mathbf{r}(\zeta)$ given in Equation (1), we go over a few relevant properties:

- Bézier curves are always completely contained within the convex hull of their control points.
- The initial and final points of a Bézier curve are the first and the last control points, respectively ($\mathbf{r}(0) = \bar{\mathbf{r}}_0, \mathbf{r}(1) = \bar{\mathbf{r}}_n$).

- The derivative of an n^{th} order Bézier curve $\left(\mathbf{q}(\zeta) = \frac{d\mathbf{r}(\zeta)}{d\zeta}\right)$ is an $(n - 1)^{th}$ order Bézier curve with control points given as

$$\bar{\mathbf{q}}_k = n(\bar{\mathbf{r}}_{k+1} - \bar{\mathbf{r}}_k), \quad k = 0, 1, \dots, n - 1. \quad (3)$$

- Addition (or subtraction) of n^{th} order Bézier curves results in an n^{th} order Bézier curve with the control points given as the sum (or difference) of the control points of the original curves.

An example Bézier curve is shown in Figure 1a where a few of these properties are illustrated.

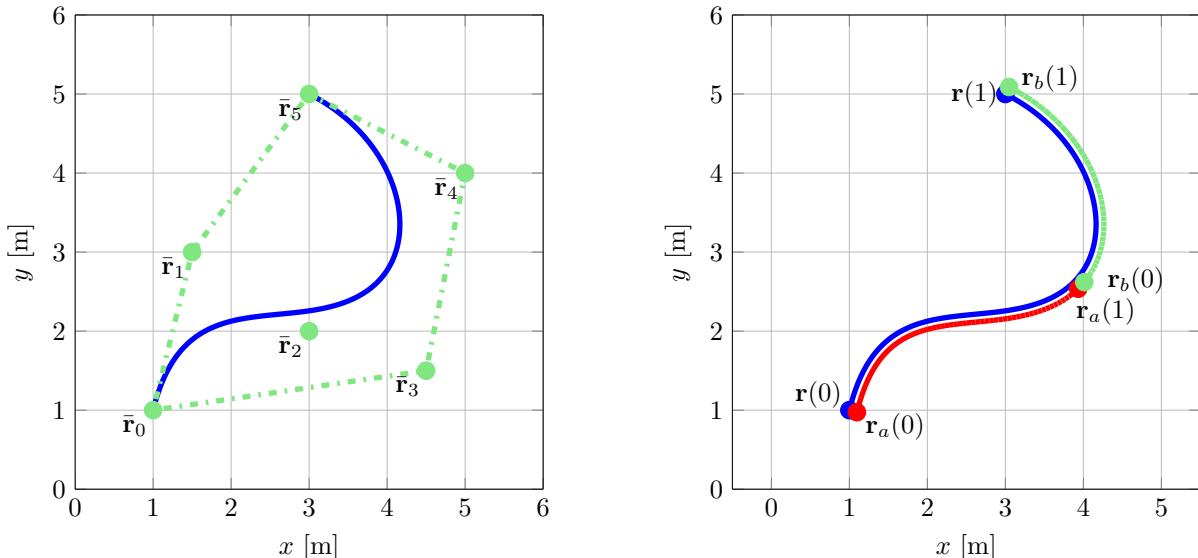
Using properties of Bézier curves, several algorithms have been presented in the literature that efficiently analyze and operate on these curves. A couple of such algorithms relevant to our collision-avoidance procedure are discussed in the following.

The de Casteljau Algorithm

The de Casteljau algorithm can be used to subdivide a Bézier curve into two independent ones of the same order. More precisely, given an n^{th} order Bézier curve $\mathbf{r}(\zeta) = \sum_{k=0}^n \bar{\mathbf{r}}_k b_k^n(\zeta)$ and a scalar $\zeta_{\text{div}} \in (0, 1)$, the de Casteljau algorithm outputs two n^{th} order Bézier curves $\mathbf{r}_a(\zeta)$ and $\mathbf{r}_b(\zeta)$ with $\zeta \in [0, 1]$ as

$$\mathbf{r}_a(\zeta) = \sum_{k=0}^n \bar{\mathbf{r}}_{a,k} b_k^n(\zeta) = \mathbf{r}(\zeta_{\text{div}} \zeta), \quad \mathbf{r}_b(\zeta) = \sum_{k=0}^n \bar{\mathbf{r}}_{b,k} b_k^n(\zeta) = \mathbf{r}(\zeta_{\text{div}} + \zeta(1 - \zeta_{\text{div}})), \quad \zeta \in [0, 1].$$

Figure 1b illustrates application of this algorithm on a two-dimensional curve.



(a) An example Bézier curve (shown in blue) defined by its six control points $\bar{\mathbf{r}}_i, i = 0, 1, \dots, 5$ (shown in green). The curve is completely contained within the convex hull (shown in dashed-dot green) of its control points.

(b) An example Bézier curve (shown in blue) is subdivided using the de Casteljau algorithm to obtain two new curves (shown in red and green). In this figure, the red and green curves have been slightly shifted for illustrative purposes, but in fact they lie on top of the blue curve.

Figure 1: An illustration of planar Bézier curves along with application of the de Casteljau algorithm.

Minimum Distance Calculation

The minimum distance between two Bézier curves can be efficiently calculated using the algorithm described in [19]. Given two Bézier curves,

$$\mathbf{r}_1(\zeta) = \sum_{k=0}^n \bar{\mathbf{r}}_{1,k} b_k^n(\zeta), \quad \mathbf{r}_2(\zeta) = \sum_{k=0}^n \bar{\mathbf{r}}_{2,k} b_k^n(\zeta), \quad \zeta \in [0, 1],$$

the algorithm calculates

$$\min_{\zeta_1, \zeta_2 \in [0, 1]} \|\mathbf{r}_1(\zeta_1) - \mathbf{r}_2(\zeta_2)\| \quad \text{and} \quad \arg \min_{\zeta_1, \zeta_2 \in [0, 1]} \|\mathbf{r}_1(\zeta_1) - \mathbf{r}_2(\zeta_2)\|.$$

II.B. Trajectory Generation

In planar setting, our cooperative trajectory-generation algorithm [16, 17] finds N trajectories along with a planned total mission time $t_i^f \in \mathbb{R}^+$ associated with each vehicle

$$\mathbf{p}_i : [0, t_i^f] \rightarrow \mathbb{R}^2, \quad i = 1, 2, \dots, N.$$

We define the total mission time as $t^f = \max\{t_1^f, t_2^f, \dots, t_N^f\}$ and $t \in [0, t^f]$ as the current instant in the mission. Please note that the mission for the i^{th} vehicle terminates when $t = t_i^f$, whereas, the overall mission only terminates at $t = t^f$.

We represent these trajectories using Bézier curves as

$$\mathbf{p}_i(t) = \sum_{k=0}^n \bar{\mathbf{p}}_{i,k} b_k^n \left(\frac{t}{t_i^f} \right), \quad i = 1, 2, \dots, N,$$

where each trajectory is completely defined by the $n + 1$ control points $\bar{\mathbf{p}}_{i,k} \in \mathbb{R}^2$ and the final mission time variable t_i^f .

For sake of simplicity, we will assume the same total mission time for all vehicles in this paper ($t_i^f = t^f, \forall i = 1, 2, \dots, N$). Then, defining a normalized mission time variable $\hat{t} = \frac{t}{t^f}$, we can represent the vehicles' trajectories obtained through our algorithm as

$$\mathbf{p}_i : [0, 1] \rightarrow \mathbb{R}^2, \quad i = 1, 2, \dots, N,$$

where

$$\mathbf{p}_i(\hat{t}) = \sum_{k=0}^n \bar{\mathbf{p}}_{i,k} b_k^n (\hat{t}), \quad i = 1, 2, \dots, N. \quad (4)$$

Apart from other constraints, the trajectories obtained from our algorithm satisfy dynamic constraints of the vehicles. Depending on the type of the vehicles (such as ground robots, quadrotors and fixed-wing aircraft), these constraints can take various forms. In this paper, however, we focus on quadrotors and therefore consider such constraints in terms of maximum speed and acceleration of the vehicles as

$$\max_{\hat{t} \in [0, 1]} \|\mathbf{v}_i(\hat{t})\| \leq v_{d,\max}, \quad \max_{\hat{t} \in [0, 1]} \|\mathbf{a}_i(\hat{t})\| \leq a_{d,\max}, \quad i = 1, 2, \dots, N.$$

where $\mathbf{v}_i(\hat{t}) = \frac{d\mathbf{p}_i(\hat{t})}{d\hat{t}} = \frac{d\mathbf{p}_i(\hat{t})}{d\hat{t}} \frac{1}{t^f}$, $\mathbf{a}_i(\hat{t}) = \frac{d^2\mathbf{p}_i(\hat{t})}{d\hat{t}^2} = \frac{d^2\mathbf{p}_i(\hat{t})}{d\hat{t}^2} \frac{1}{(t^f)^2}$, and $v_{d,\max}$ and $a_{d,\max}$ represent the maximum allowed speed and acceleration for the vehicles during the planning phase. Given the dynamics of a vehicle, $v_{d,\max}$ and $a_{d,\max}$ can be set as the maximum speed and acceleration achievable by the vehicle. However, as we will later see in the paper, lower values are necessary in order to allow for trajectory replanning.

Our trajectory-generation algorithm also takes into account a set of mission-specific constraints. In this paper, we consider such constraints in terms of the initial and final positions and velocities, i.e. the trajectories obtained from our algorithm satisfy:

$$\mathbf{p}_i(0) = \mathbf{p}_i^i, \quad \mathbf{p}_i(1) = \mathbf{p}_i^f \quad \mathbf{v}_i(0) = \mathbf{v}_i^i, \quad \mathbf{v}_i(1) = \mathbf{v}_i^f,$$

where \mathbf{p}_i^i and \mathbf{p}_i^f are the initial and final positions, while \mathbf{v}_i^i and \mathbf{v}_i^f are the initial and final velocities of the i^{th} vehicle.

Lastly, the trajectories ensure minimum separation between the vehicles. Depending on the mission requirements, this condition can be in terms of *spatial* or *temporal* separation. Spatial separation ensures a minimum distance between paths of the vehicles as

$$\min_{i,j=1, \dots, N, i \neq j} \|\mathbf{p}_i(\hat{t}_i) - \mathbf{p}_j(\hat{t}_j)\| \geq E, \quad \forall \hat{t}_i, \hat{t}_j \in [0, 1],$$

whereas temporal separation ensures a minimum distance between vehicles at every mission time instant $\hat{t} \in [0, 1]$ i.e.

$$\min_{i,j=1, \dots, N, i \neq j} \|\mathbf{p}_i(\hat{t}) - \mathbf{p}_j(\hat{t})\| \geq E, \quad \forall \hat{t} \in [0, 1].$$

III. Problem Formulation

We consider N vehicles perfectly following their respective planned trajectories given by Equation (4), with $n > 6$. Our proposed algorithm runs independently on all N vehicles, therefore from now on, we will focus on one vehicle and omit the subscript i to denote its trajectory as

$$\mathbf{p} : [0, 1] \rightarrow \mathbb{R}^2, \quad \mathbf{p}(\hat{t}) = \sum_{k=0}^n \bar{\mathbf{p}}_k b_k^n(\hat{t}), \quad \hat{t} \in [0, 1],$$

where $\bar{\mathbf{p}}_k$ are control points for the trajectory. Furthermore, the dynamic constraints imposed during trajectory generation can now be written as

$$\max_{\hat{t} \in [0, 1]} \|\mathbf{v}(\hat{t})\| \leq v_{d,\max}, \quad \max_{\hat{t} \in [0, 1]} \|\mathbf{a}(\hat{t})\| \leq a_{d,\max},$$

where $\mathbf{v}(\hat{t}) = \frac{d\mathbf{p}(\hat{t})}{dt} = \frac{d\mathbf{p}(\hat{t})}{d\hat{t}} \frac{1}{t^f}$, $\mathbf{a}(\hat{t}) = \frac{d^2\mathbf{p}(\hat{t})}{dt^2} = \frac{d\mathbf{p}(\hat{t})}{d\hat{t}^2} \frac{1}{(t^f)^2}$, and $v_{d,\max}$ and $a_{d,\max}$ represent the maximum allowed speed and acceleration for the vehicle during mission planning. As mentioned earlier, these may be set lower than what is physically possible.

Finally, mission-specific constraints can be written as

$$\mathbf{p}(0) = \mathbf{p}^i, \quad \mathbf{p}(1) = \mathbf{p}^f, \quad \mathbf{v}(0) = \mathbf{v}^i, \quad \mathbf{v}(1) = \mathbf{v}^f,$$

where \mathbf{p}^i , \mathbf{p}^f , \mathbf{v}^i and \mathbf{v}^f represent the initial and final positions and velocities of the vehicle.

We want our collision avoidance algorithm to satisfy the following criteria:

- 1. Collision Prediction:** The first step in avoiding a collision is to predict if one is to occur in the future. To this end, given the vehicle's and obstacle's trajectories we want our algorithm to determine if there is a possible collision in the future. Furthermore, if a collision is to occur, we want the algorithm to find a collision reference time \hat{t}_* at which we have a collision.

In precise terms, we say that the vehicle is in collision with the obstacle at any mission time $\hat{t} \in [0, 1]$, if

$$\|\mathbf{p}(\hat{t}) - \mathbf{p}_o(\hat{t})\| \leq d_{\text{safe}}, \quad (5)$$

for some positive scalar d_{safe} , where $\mathbf{p}_o(\hat{t})$ describes the position of the obstacle at mission time \hat{t} .

Therefore, given $\mathbf{p}(\hat{t})$ and $\mathbf{p}_o(\hat{t})$ as n^{th} order Bézier (or any other polynomial) curves, we want our algorithm to find a collision reference time \hat{t}_* at which Equation (5) is satisfied, and return a failure if no such instant exists.

- 2. Bounded Replanned Trajectory:** Once the vehicle under consideration predicts a collision, we want our algorithm to replan the trajectory such that the collision is averted and the mission-specific constraints (such as initial and final positions and velocities) are satisfied. Furthermore, the new trajectory needs to be twice differentiable (i.e. needs to have continuous position, velocity and acceleration) to be practically feasible for any vehicle. However, this is only possible if the collision occurs strictly after the obstacle detection and ends before the completion of the mission. For example, if the vehicle is in collision at the end of the mission, the final position constraint cannot be satisfied. In specific terms, we describe our goals for the algorithm discussed here as follows:

Consider that at mission time $\hat{t} = \hat{t}_c$, the vehicle predicts a collision with reference time $\hat{t} = \hat{t}_* > \hat{t}_c$. Furthermore, let the vehicle be in collision during the interval $[\hat{t}_*^\ell, \hat{t}_*^u]$, i.e.

$$\|\mathbf{p}(\hat{t}) - \mathbf{p}_o(\hat{t})\| \leq d_{\text{safe}}, \quad \hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u], \quad (6)$$

$$\|\mathbf{p}(\hat{t}) - \mathbf{p}_o(\hat{t})\| > d_{\text{safe}}, \quad \hat{t} \in [0, \hat{t}_*^\ell) \cup (\hat{t}_*^u, 1], \quad (7)$$

where \hat{t}_*^ℓ and \hat{t}_*^u represent the beginning and end of the collision window. Note that from the definition, the collision reference time \hat{t}_* must satisfy $\hat{t}_* \in [\hat{t}_*^\ell, \hat{t}_*^u]$.

Given only \hat{t}_* (and not \hat{t}_*^ℓ and \hat{t}_*^u) we want our algorithm to find a twice differentiable trajectory $\mathbf{p}_{\text{new}}(\hat{t}) : [0, 1] \rightarrow \mathbb{R}^2$ that keeps the current position, velocity and acceleration the same

$$\mathbf{p}_{\text{new}}(\hat{t}_c) = \mathbf{p}(\hat{t}_c), \quad \mathbf{v}_{\text{new}}(\hat{t}_c) = \mathbf{v}(\hat{t}_c), \quad \mathbf{a}_{\text{new}}(\hat{t}_c) = \mathbf{a}(\hat{t}_c), \quad (8)$$

and avoids the collision

$$\|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}_o(\hat{t})\| > d_{\text{safe}}, \quad \hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u],$$

while satisfying the mission-specific constraints

$$\mathbf{p}_{\text{new}}(1) = \mathbf{p}^f, \quad \mathbf{v}_{\text{new}}(1) = \mathbf{v}^f,$$

$$\text{where } \mathbf{v}_{\text{new}}(\hat{t}) = \frac{d\mathbf{p}_{\text{new}}(\hat{t})}{d\hat{t}} = \frac{d\mathbf{p}_{\text{new}}(\hat{t})}{d\hat{t}} \frac{1}{t^f} \text{ and } \mathbf{a}_{\text{new}}(\hat{t}) = \frac{d^2\mathbf{p}_{\text{new}}(\hat{t})}{d\hat{t}^2} = \frac{d^2\mathbf{p}_{\text{new}}(\hat{t})}{d\hat{t}^2} \frac{1}{(t^f)^2}.$$

- 3. Computable Bounds for the Replanned Trajectory:** For the multi-vehicle missions under consideration, a replanned trajectory which is very different from the original is undesirable as it may bring the vehicles on a collision course with one another. Furthermore, a large change in the velocity or acceleration profile may violate the dynamic constraints of the vehicle. Therefore, it is reasonable to expect that the algorithm provides a trajectory similar to the original one. Furthermore, if the bounds on the difference between the original and the new trajectories are computable, the offline trajectory-generation algorithm can use them to provide enough margins. Our specific requirements on the algorithm in this regard are described below.

Given some known positive constants T_1 , T_2 and T_{col} , if the collision is described by Equations (6), (7) and satisfies

$$(\hat{t}_*^\ell - \hat{t}_c)t^f > T_1, \quad (1 - \hat{t}_*^u)t^f > T_2, \quad (\hat{t}_*^u - \hat{t}_*^\ell)t^f < T_{\text{col}}, \quad (9)$$

(i.e. the collision window begins later than T_1 seconds after detection, ends earlier than T_2 seconds before the mission final time t^f , and lasts for less than T_{col} seconds), then, under the mild assumption $\min\{T_1, T_2\} > T_{\text{col}}$, we want the replanned trajectory to satisfy

$$\begin{aligned} \max_{\hat{t} \in [0, 1]} \|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}(\hat{t})\| &< \Delta_p, & \max_{\hat{t} \in [0, 1]} \|\mathbf{v}_{\text{new}}(\hat{t}) - \mathbf{v}(\hat{t})\| &< \Delta_v, \\ \max_{\hat{t} \in [0, 1]} \|\mathbf{a}_{\text{new}}(\hat{t}) - \mathbf{a}(\hat{t})\| &< \Delta_a, \end{aligned}$$

such that Δ_p , Δ_v and Δ_a can be computed offline using T_1 , T_2 , T_{col} , d_{safe} and design parameters of the algorithm that will be discussed in the next section.

IV. Proposed Algorithm

In this section, we will provide our algorithm that satisfies the criteria described above.

IV.A. Collision Prediction

Collision prediction is the first part of our algorithm and is used upon detection of an obstacle to determine if a collision is about to occur in the future. We assume that the trajectory of the obstacle can be predicted and represented as an n^{th} order Bézier curve upon its detection. The algorithm is explained step-by-step as follows:

Step 1: Express Obstacle's Trajectory as a Bézier Curve

To use algorithms designed for Bézier curves, the trajectories of both the vehicle and the obstacle need to be represented as such curves. To this end, we approximate the obstacle's trajectory with an n^{th} order Bézier curve in this step. From this point on, we will consider this approximation as representative of the actual obstacle's trajectory.

Since Bézier curves are a linear combination of the control points, we can simply use the least squares method to find control points $\bar{\mathbf{p}}_{o,k}$ for $k = 0, 1, \dots, n$, such that

$$\mathbf{p}_o(\hat{t}) = \sum_{k=0}^n \bar{\mathbf{p}}_{o,k} b_k^n(\hat{t}),$$

where $\mathbf{p}_o(\hat{t})$ is the predicted trajectory of the obstacle as a function of mission time \hat{t} .

Step 2: Find the Separation Bézier Curve

In this step we find a Bézier curve that describes the separation vector between the vehicle and the obstacle as a function of time. This will allow us to write the collision condition (5) in a way such that the minimum distance algorithm discussed in Section II.A can be used.

Given trajectories of the vehicle and the obstacle as n^{th} order Bézier curves, we can write the separation between the obstacle and the vehicle as

$$\mathbf{d}(\hat{t}) = \mathbf{p}(\hat{t}) - \mathbf{p}_o(\hat{t}) = \sum_{k=0}^n (\bar{\mathbf{p}}_k - \bar{\mathbf{p}}_{o,k}) b_k^n(\hat{t}). \quad (10)$$

From the above equation, it follows that $\mathbf{d}(\hat{t})$ is a Bézier curve with control points $(\bar{\mathbf{p}}_k - \bar{\mathbf{p}}_{o,k})$ for $k = 0, 1, \dots, n$. This step is also illustrated in Figures 2 and 3.

Step 3: Use Minimum Distance Algorithm for Bézier Curves

Given the separation Bézier curve, we can write the collision condition as

$$\min_{\hat{t} \in [0,1]} \|\mathbf{p}(\hat{t}) - \mathbf{p}_o(\hat{t})\| = \min_{\hat{t} \in [0,1]} \|\mathbf{d}(\hat{t})\| \leq d_{\text{safe}},$$

and use the minimum distance algorithm to find both

$$d_{\min} = \min_{\hat{t} \in [0,1]} \|\mathbf{d}(\hat{t})\| \quad \text{and} \quad \arg \min_{\hat{t} \in [0,1]} \|\mathbf{d}(\hat{t})\|.$$

In case $d_{\min} \leq d_{\text{safe}}$, we predict a collision with reference time \hat{t}_* given as

$$\hat{t}_* = \arg \min_{\hat{t} \in [0,1]} \|\mathbf{d}(\hat{t})\|.$$

IV.B. Path Replanning

If the vehicle under consideration predicts a collision, it replans its trajectory in order to avert it. This section describes our proposed step-by-step procedure for replanning. The idea behind our algorithm is to add a detour to the original trajectory in such a way that $\|\mathbf{d}(\hat{t})\|$ becomes larger than d_{safe} . Graphically, if $\mathbf{d}(\hat{t})$ enters the circle with a radius d_{safe} centered around origin (such as the case shown in Figure 3b), then our algorithm adds a Gaussian-like detour to it in order to make it avoid the circle.

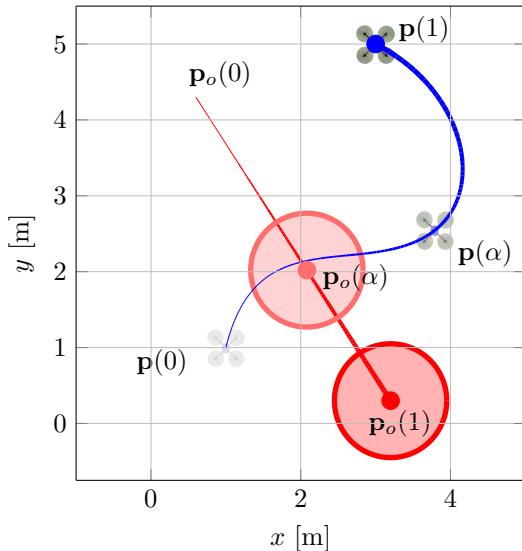
We consider the scenario as follows: the vehicle under consideration, with its trajectory described as $\mathbf{p}(\hat{t})$, detects an obstacle at mission time $\hat{t} = \hat{t}_c$ and predicts a collision with reference time \hat{t}_* , where the separation curve is described as $\mathbf{d}(\hat{t})$. The vehicle then replans its trajectory according to the steps described below:

Step 1: Subdivide the Separation Bézier Curve

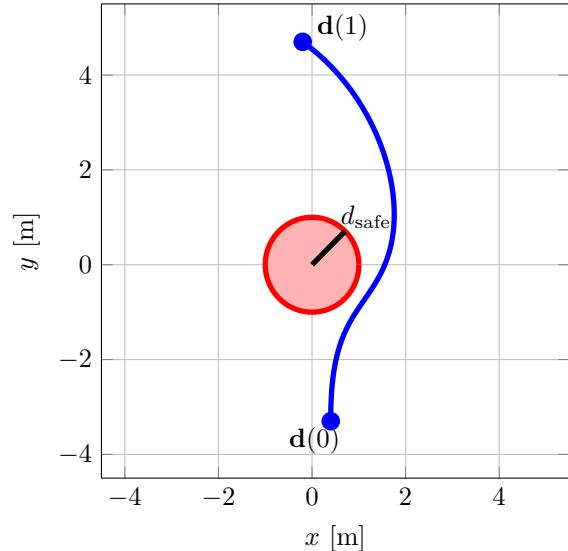
Notice that during the mission, it is impossible to change the whole trajectory of the vehicle (for example $\mathbf{d}(\hat{t})$ for $\hat{t} \leq \hat{t}_c$ cannot be changed). Furthermore, it may not be desirable to change parts of trajectory that are far off from the collision in time. Therefore, in this step we separate parts of the trajectory that are either irrelevant to the collision-avoidance maneuver or cannot be changed anymore.

We determine the subsegment of $\mathbf{d}(\hat{t})$ that is defined on the interval $\hat{t} \in [\hat{t}_{\text{div}}^\ell, \hat{t}_{\text{div}}^u]$. To this end, we use the de Casteljau algorithm and subdivide $\mathbf{d}(\hat{t})$ at $\hat{t} = \hat{t}_{\text{div}}^\ell$ and $\hat{t} = \hat{t}_{\text{div}}^u$ and obtain

$$\mathbf{d}_{\text{sub}}(\hat{\tau}) = \sum_{k=0}^n \bar{\mathbf{d}}_{\text{sub},k} b_k^n(\hat{\tau}) = \mathbf{d}(\hat{t}_{\text{div}}^\ell + \hat{\tau}(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell)), \quad \hat{\tau} \in [0, 1],$$

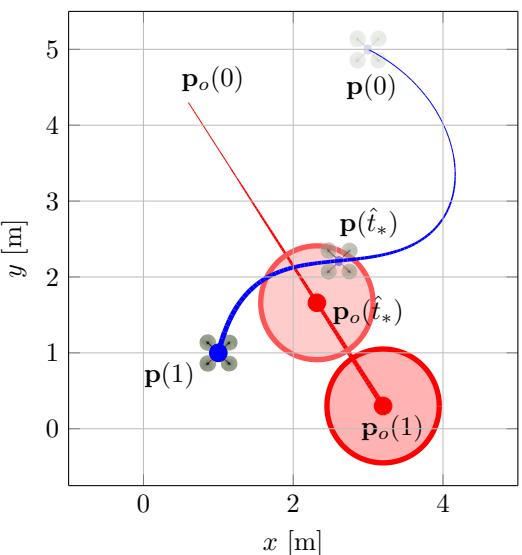


(a) Obstacle's trajectory $\mathbf{p}_o(\hat{t})$ (shown in red) against the vehicle's trajectory $\mathbf{p}(\hat{t})$ (shown in blue).

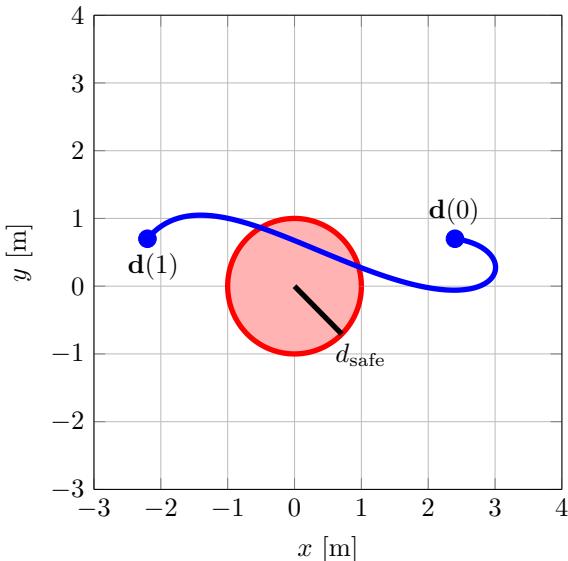


(b) Separation Bézier curve $\mathbf{d}(\hat{t})$ shown against a circle around origin. Collision is not predicted since the separation curve stays outside the circle with a radius d_{safe} .

Figure 2: Illustration of the collision-prediction algorithm: although the curves representing trajectories of the vehicle and the obstacle intersect, no collision is predicted since the separation curve stays outside the circle with a radius d_{safe} centered at the origin.



(a) Obstacle's trajectory $\mathbf{p}_o(\hat{t})$ (shown in red) against the vehicle's trajectory $\mathbf{p}(\hat{t})$ (shown in blue).



(b) Separation Bézier curve $\mathbf{d}(\hat{t})$ shown against a circle around origin. Collision is predicted since the separation curve enters the circle.

Figure 3: Illustration of the collision-prediction algorithm: a collision is predicted since the separation curve enters the circle with a radius d_{safe} centered at the origin.

where $\bar{\mathbf{d}}_{\text{sub},k}$ are the control points of the subdivided curve and the parameters $\hat{t}_{\text{div}}^\ell$ and \hat{t}_{div}^u are determined as

$$\begin{bmatrix} \hat{t}_{\text{div}}^\ell \\ \hat{t}_{\text{div}}^u \end{bmatrix} = \begin{cases} \left[\hat{t}_c, \frac{\hat{t}_* - \hat{t}_c + \hat{\tau}_{\text{ds}}^\ell \hat{t}_c}{\hat{\tau}_{\text{ds}}^\ell} \right]^\top, & \text{if } \frac{\hat{t}_* - \hat{t}_c}{1 - \hat{t}_c} < \hat{\tau}_{\text{ds}}^\ell, \\ \left[\frac{\hat{t}_* - \hat{\tau}_{\text{ds}}^u}{1 - \hat{\tau}_{\text{ds}}^u}, 1 \right]^\top, & \text{if } \frac{\hat{t}_* - \hat{t}_c}{1 - \hat{t}_c} > \hat{\tau}_{\text{ds}}^u, \\ \left[\hat{t}_c, 1 \right]^\top, & \text{otherwise,} \end{cases} \quad (11)$$

where $0 < \hat{\tau}_{\text{ds}}^\ell \leq \hat{\tau}_{\text{ds}}^u < 1$ are design parameters that will be discussed later in this paper. We also define the function $g : [\hat{t}_{\text{div}}^\ell, \hat{t}_{\text{div}}^u] \rightarrow [0, 1]$ that maps the variable \hat{t} to $\hat{\tau}$ and is defined as:

$$g(\hat{t}) = \frac{\hat{t} - \hat{t}_{\text{div}}^\ell}{\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell}. \quad (12)$$

Next, we define the following instants related to the collision as:

$$\hat{\tau}_* \triangleq g(\hat{t}_*), \quad \hat{\tau}_*^\ell \triangleq g(\hat{t}_*^\ell), \quad \text{and} \quad \hat{\tau}_*^u \triangleq g(\hat{t}_*^u). \quad (13)$$

In Figure 4, we provide an illustrative timeline containing these time instants. Note that we have set $\hat{t}_{\text{div}}^\ell \neq \hat{t}_c$ and $\hat{t}_{\text{div}}^u \neq 1$ in the figure. This, however, is only for illustrative purposes and not an accurate representation of our algorithm which always sets $\hat{t}_{\text{div}}^\ell = \hat{t}_c$ and/or $\hat{t}_{\text{div}}^u = 1$ (See Equation (11)). A second illustration for this step is provided in Figure 5 where the times of subdivision $\hat{t}_{\text{div}}^\ell, \hat{t}_{\text{div}}^u$ are marked out on the trajectory $\mathbf{p}(\hat{t})$, and the separation curve $\mathbf{d}(\hat{t})$ is shown as subdivided at these times.

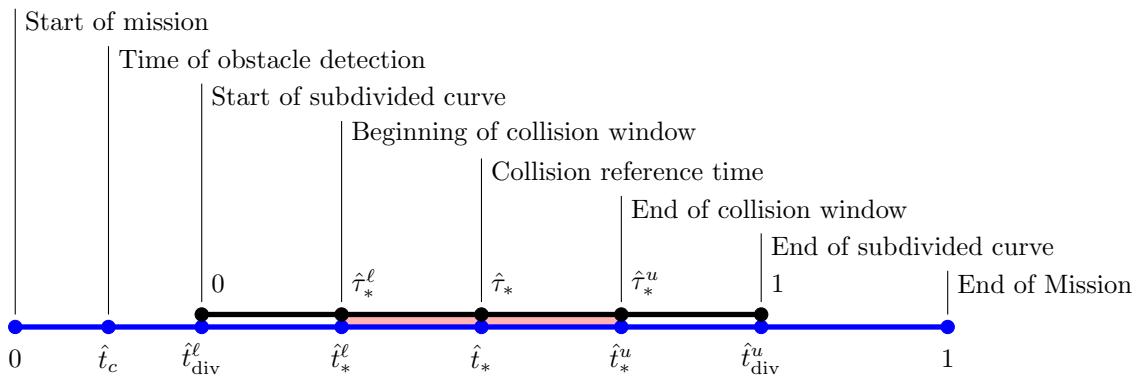


Figure 4: A timeline showing relevant instants in terms of \hat{t} and $\hat{\tau}$.

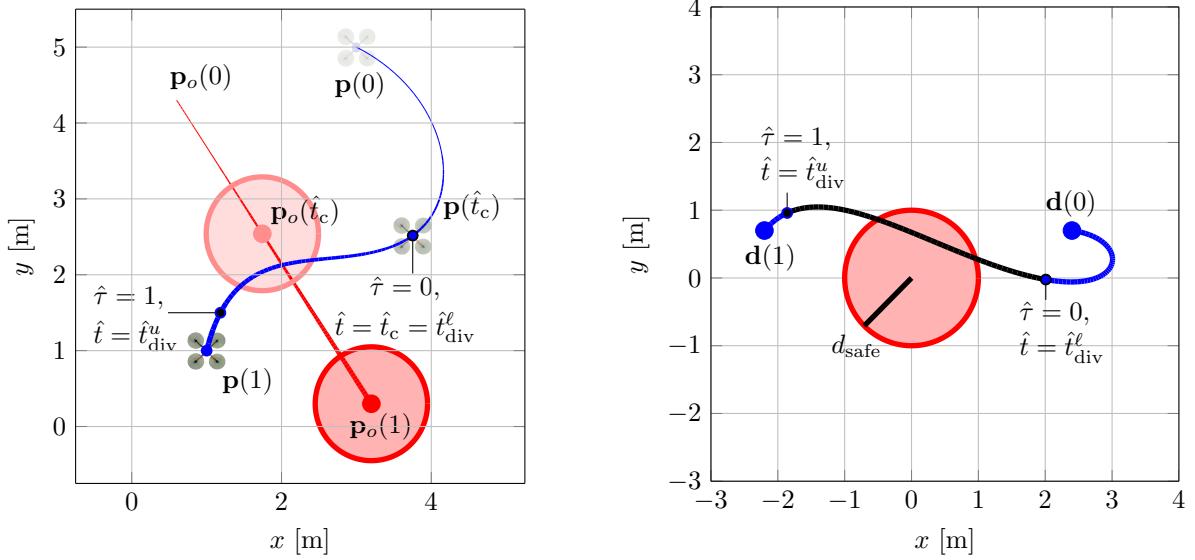
Before we move to our next step in the algorithm, we will partially describe the choice of times of subdivision $\hat{t}_{\text{div}}^\ell$ and \hat{t}_{div}^u and give a physical interpretation of the design variables $\hat{\tau}_{\text{ds}}^\ell$ and $\hat{\tau}_{\text{ds}}^u$ with the following lemma.

Lemma 1. *Let $\hat{t}_c < \hat{t}_* < 1$ and the design parameters $\hat{\tau}_{\text{ds}}^\ell$ and $\hat{\tau}_{\text{ds}}^u$ satisfy $[\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \subset (0, 1)$. If $\hat{t}_{\text{div}}^\ell$ and \hat{t}_{div}^u are determined through Equation (11), the collision reference time in terms of t and $\hat{\tau}$ (denoted as \hat{t}_* and $\hat{\tau}_*$, respectively) satisfy*

$$\hat{t}_* \in [\hat{t}_{\text{div}}^\ell, \hat{t}_{\text{div}}^u] \quad \text{and} \quad \hat{\tau}_* \in [\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \subset (0, 1).$$

The proof of the above lemma follows directly from Equation (11) and is omitted here for brevity.

In the next section, we show the significance of this result in ensuring reasonable solutions for our algorithm.



(a) The trajectory $\mathbf{p}(\hat{t})$ with times of subdivision $\hat{t}_{\text{div}}^{\ell}$ and \hat{t}_{div}^u marked out.
(b) The separation curve $\mathbf{d}(\hat{t})$ shown as subdivided at $\hat{t} = \hat{t}_{\text{div}}^{\ell}$ and $\hat{t} = \hat{t}_{\text{div}}^u$ where the new curve $\mathbf{d}_{\text{sub}}(\hat{t})$ is shown in black.

Figure 5: Illustration of first step of the collision-avoidance algorithm, where the separation Bézier curve is subdivided at $\hat{t} = \hat{t}_{\text{div}}^{\ell}$ and $\hat{t} = \hat{t}_{\text{div}}^u$. The corresponding trajectory $\mathbf{p}(\hat{t})$ is also shown as subdivided for reference.

Step 2: Calculate the Time Profile of the Detour Magnitude

As we will later see, the detour curve will exist on the interval $\hat{\tau} \in [0, 1]$. This detour will always point in one direction, whereas, its magnitude will vary with $\hat{\tau}$. In this step, we derive the magnitude profile of the detour curve in terms of $\hat{\tau}$.

Specifically, the magnitude profile is found as an n^{th} order Bézier curve as

$$s(\hat{\tau}_*, \hat{\tau}) = \sum_{k=0}^n \bar{s}_k(\hat{\tau}_*) b_k^n(\hat{\tau}), \quad \hat{\tau} \in [0, 1], \quad (14)$$

with the control points given as

$$\bar{s}_k(\hat{\tau}_*) = \begin{cases} \frac{b_k^n(\hat{\tau}_*)}{\sum_{j=3}^{n-3} (b_j^n(\hat{\tau}_*))^2}, & 3 \leq k \leq n-3, \\ 0, & \text{otherwise,} \end{cases} \quad k = 0, 1, \dots, n. \quad (15)$$

We now explain our choice for this magnitude profile.

First, we define it as a Bézier curve: this, as we will later see, ensures that the full detour curve remains a Bézier curve that can be added to the original trajectory.

Second, we keep the first three and the last three control points as zero: this ensures that the initial and final positions, velocities and accelerations of the detour are zero. As we will later see, this enables twice differentiability of the replanned trajectory. Since \hat{t} and $\hat{\tau}$ are linearly related, this translates to

$$s(\hat{\tau}_*, 0) = 0, \quad \frac{ds(\hat{\tau}_*, 0)}{d\hat{\tau}} = 0, \quad \frac{d^2 s(\hat{\tau}_*, 0)}{d\hat{\tau}^2} = 0, \quad (16)$$

$$s(\hat{\tau}_*, 1) = 0, \quad \frac{ds(\hat{\tau}_*, 1)}{d\hat{\tau}} = 0, \quad \frac{d^2 s(\hat{\tau}_*, 1)}{d\hat{\tau}^2} = 0. \quad (17)$$

Last, the expression the control points for $3 \leq k \leq n - 3$ in Equation (15) is found as the least squares solution to the equation

$$s(\hat{\tau}_*, \hat{\tau} = \hat{\tau}_*) = \sum_{k=0}^n \bar{s}_k(\hat{\tau}_*) b_k^n(\hat{\tau}_*) = [b_0^n(\hat{\tau}_*) \ b_1^n(\hat{\tau}_*) \ \dots \ b_n^n(\hat{\tau}_*)] \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \bar{s}_n \\ \vdots \\ \bar{s}_{n-3} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} = 1,$$

where $\mathbf{0}_{3 \times 1}$ is a column vector of zero entries. Thus, our choice of control points ensures a detour magnitude of 1 at $\hat{\tau} = \hat{\tau}_*$, while minimizing the 2-norm of the vector containing the control points $\bar{s}_k(\hat{\tau}_*)$. Although other norms (such as the infinity norm) could have been used here, we use the 2-norm as it lends itself to a closed-form solution of Equation (15) and thus allows for onboard implementation of the algorithm.

A few different detour magnitude profiles are shown in Figure 6, where we can see the Gaussian-like shape of the profile. Notice that the magnitude profiles assume unnecessarily large values for very small or very large values of $\hat{\tau}_*$. Recalling our result from Lemma 1, that $\hat{\tau}_* \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u]$, we can conclude that such scenarios are avoided through proper selection of the design parameters $\hat{\tau}_{ds}^\ell$ and $\hat{\tau}_{ds}^u$.

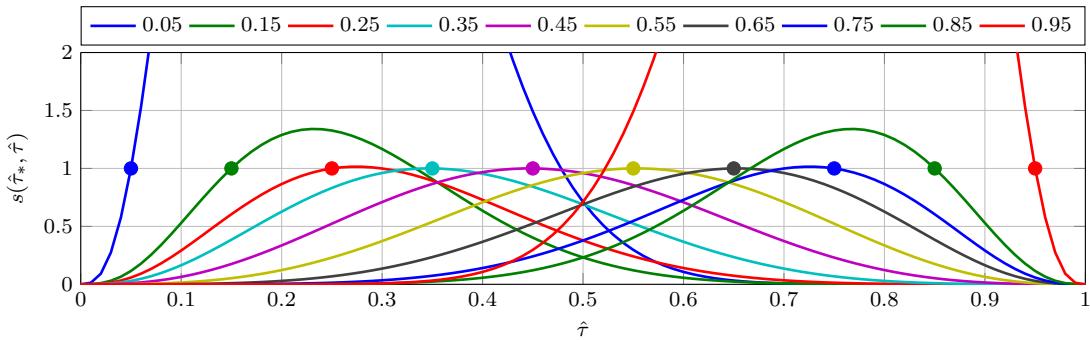


Figure 6: Magnitude profile $s(\hat{\tau}_*, \hat{\tau})$ for different values of $\hat{\tau}_*$ and $n = 15$. Similar profiles are obtained for other values of n . The value of $\hat{\tau}_*$ is also shown as a solid circle for every curve.

Step 3: Scale Detour with an Appropriate Vector

In this step, we find the complete detour as

$$\Delta_K(\hat{\tau}_*, \hat{\tau}) = K s(\hat{\tau}_*, \hat{\tau}) \mathbf{u}, \quad (18)$$

where $\mathbf{u} = \frac{\mathbf{d}_{\text{sub}}(\hat{\tau}_*)}{\|\mathbf{d}_{\text{sub}}(\hat{\tau}_*)\|}$ and $K \in \mathbb{R}^+$ is a scaling factor that satisfies

$$\min_{\hat{\tau} \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u]} \|\Delta_K(\hat{\tau}_*, \hat{\tau}) + \mathbf{d}_{\text{sub}}(\hat{\tau})\| > d_{\text{safe}}. \quad (19)$$

Ideally, we want K to be the minimum value that satisfies Equation (19). However, finding the exact minimum may not be feasible for an onboard and real-time implementation. Therefore, we select K as the lowest value that can be guaranteed to satisfy Equation (19) by the algorithm in real-time. Our iterative procedure for this is briefly described in the following.

First, the iterative procedure is initialized by

$$K_k = K^u, \quad \text{for } k = 0,$$

where k denotes the iteration step and K^u is defined in Lemma 3. In fact, K^u results in a very conservative value for K that satisfies Equation (19), however, any $K > K^u$ would be an even more conservative choice. Hence, the algorithm aims to find a lower value than K^u . Also note, that the condition in (19) is not satisfied for any $K < K^\ell \triangleq d_{\text{safe}} - \|\mathbf{d}_{\text{sub}}(\hat{\tau}_*)\|$ at $\hat{\tau} = \hat{\tau}_*$. Therefore, we can limit the domain of search to the interval (K^ℓ, K^u) . Next, we define the constraint function

$$H(K) = \min_{\hat{\tau} \in [0, 1]} \|\Delta_K(\hat{\tau}_*, \hat{\tau}) + \mathbf{d}_{\text{sub}}(\hat{\tau})\| - d_{\text{safe}}.$$

Here, the constraint function $H(K)$ is defined for $\hat{\tau} \in [0, 1]$, since $\hat{\tau}_*^\ell$ and $\hat{\tau}_*^u$ are not known in practice, making Equation (19) infeasible to evaluate. The algorithm proceeds by sampling the search domain and evaluating the constraint function $H(K_k)$ at every iteration step k . The value K , that determines the detour shape $\Delta_K(\hat{\tau}_*, \hat{\tau})$, is then found using

$$\begin{aligned} K &= \min_{k=0, \dots, m} K_k, \\ \text{subject to } H(K_k) &> 0, \end{aligned}$$

where m is the number of samples. At this point, we remark that the procedure outlined above, allows for practical implementation and online evaluation onboard the vehicles. However, this comes at the cost of a more conservative value of K than possible.

Step 4: Add Detour to the Original Trajectory

With the detour finalized, the last step is to add this detour to the vehicle's trajectory as

$$\mathbf{p}_{\text{new}}(\hat{t}) = \mathbf{p}(\hat{t}) + \begin{cases} \Delta_K(\hat{\tau}_*, g(\hat{t})) , & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0, & \text{otherwise,} \end{cases} \quad \hat{t} \in [0, 1]. \quad (20)$$

For this new trajectory, the separation curve $\mathbf{d}_{\text{new}}(\hat{t}) = \mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}_o(\hat{t})$ can now be written as

$$\mathbf{d}_{\text{new}}(\hat{t}) = \mathbf{d}(\hat{t}) + \begin{cases} \Delta_K(\hat{\tau}_*, g(\hat{t})) , & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0, & \text{otherwise,} \end{cases} \quad \hat{t} \in [0, 1],$$

which satisfies $\|\mathbf{d}_{\text{new}}(\hat{t})\| > d_{\text{safe}}$ for $\hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u]$. Finally, the difference between the new and the original trajectory in terms of position, velocity and acceleration can be written as

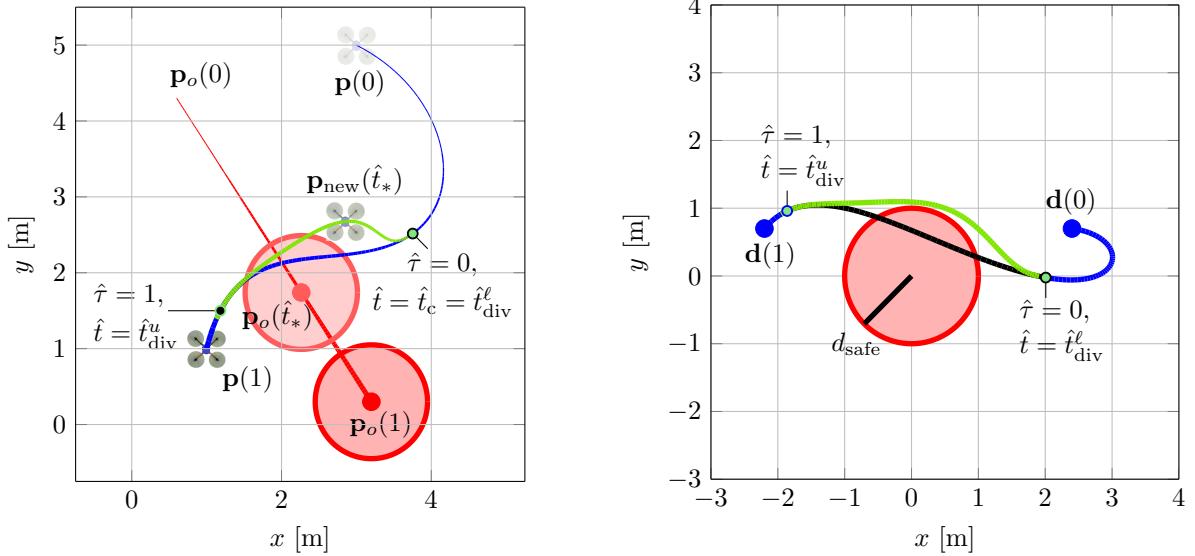
$$\begin{aligned} \mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}(\hat{t}) &= \begin{cases} K s(\hat{\tau}_*, g(\hat{t})) \mathbf{u}, & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0, & \text{otherwise,} \end{cases} \quad \hat{t} \in [0, 1], \\ \mathbf{v}_{\text{new}}(\hat{t}) - \mathbf{v}(\hat{t}) &= \begin{cases} \frac{K \mathbf{u}}{(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell) t^f} \frac{ds(\hat{\tau}_*, g(\hat{t}))}{dg(\hat{t})}, & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0, & \text{otherwise,} \end{cases} \quad \hat{t} \in [0, 1], \\ \mathbf{a}_{\text{new}}(\hat{t}) - \mathbf{a}(\hat{t}) &= \begin{cases} \frac{K \mathbf{u}}{(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell)^2 (t^f)^2} \frac{d^2 s(\hat{\tau}_*, g(\hat{t}))}{(dg(\hat{t}))^2}, & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0, & \text{otherwise,} \end{cases} \quad \hat{t} \in [0, 1], \end{aligned}$$

where the derivatives exist and are continuous at $\hat{t}_{\text{div}}^\ell$ and \hat{t}_{div}^u (or equivalently $\hat{\tau} = 0$ and $\hat{\tau} = 1$) because of Equations (16) and (17). An illustration of this final step is provided in Figure 7, where the new trajectory $\mathbf{p}_{\text{new}}(\hat{t})$ and the new separation curve $\mathbf{d}_{\text{new}}(\hat{t})$ are shown.

V. Analysis

In this section, we provide analysis of our proposed algorithm along with performance guarantees. Through the analysis, we also describe criteria for choosing the design parameters for the algorithm. To satisfy our criteria, as discussed in the problem statement, we need to ensure that the difference between the new and the old trajectory is small. To this end, this section provides theorems that bound $\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}(\hat{t})$ in terms of position, velocity and acceleration. While making a few mild assumptions, we proceed in two steps: we first show that the detour magnitude profile $s(\hat{\tau}_*, \hat{\tau})$ is bounded, followed by the proof that the scaling factor $K = K^u$ satisfies Equation (19). The lemmas and theorems in this section also show that these bounds are computable.

Our first lemma in this section shows that we can calculate bounds on the detour magnitude profile under the condition that the collision time $\hat{\tau}_*$ is in the interval $[\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \subset (0, 1)$. Furthermore, the lemma shows that the magnitude profile is non-zero for all values of $\hat{\tau} \in (0, 1)$; this latter part of the lemma provides a means to calculate K^u .



(a) The original trajectory $\mathbf{p}(\hat{t})$ (shown in blue) against the replanned trajectory $\mathbf{p}_{\text{new}}(\hat{t})$ (shown in blue and green).

(b) The original separation curve $\mathbf{d}(\hat{t})$ (shown in blue and black) against the new separation curve $\mathbf{d}_{\text{new}}(\hat{t})$ (shown in blue and green).

Figure 7: Illustration of the last two steps of the collision-avoidance algorithm in which the exact detour curve is calculated and added to the trajectory.

Lemma 2. Given $\hat{\tau}_* \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \subset (0, 1)$, the detour magnitude profile $s(\hat{\tau}_*, \hat{\tau})$, calculated using Equations (14) and (15), and its derivatives have finite bounds

$$s_{\max} = \max_{\substack{\hat{\tau}_* \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \\ \hat{\tau} \in [0, 1]}} s(\hat{\tau}_*, \hat{\tau}), \quad s'_{\max} = \max_{\substack{\hat{\tau}_* \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \\ \hat{\tau} \in [0, 1]}} \frac{ds(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}}, \quad s''_{\max} = \max_{\substack{\hat{\tau}_* \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \\ \hat{\tau} \in [0, 1]}} \frac{d^2s(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}^2}. \quad (21)$$

Furthermore, $s(\hat{\tau}_*, \hat{\tau}) > 0$ for $\hat{\tau}_*, \hat{\tau} \in (0, 1)$.

Proof. Given

$$\bar{s}_k(\hat{\tau}_*) = \frac{b_k^n(\hat{\tau}_*)}{\sum_{j=3}^{n-3} (b_j^n(\hat{\tau}_*))^2} \quad k = 3, 4, \dots, n-4, n-3,$$

and the fact that $b_j^n(\hat{\tau}_*) > 0$ for $\hat{\tau}_* \in (0, 1)$, we conclude that the control points $\bar{s}_k(\hat{\tau}_*)$ are all bounded. Since the curve $s(\hat{\tau}_*, \hat{\tau})$ is a linear combination of these control points, it is bounded and has a maximum s_{\max} in the closed set $(\hat{\tau}_*, \hat{\tau}) \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \times [0, 1]$.

Furthermore, control points of the derivative curves $\frac{ds(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}}$ and $\frac{d^2s(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}^2}$ are bounded (they are a linear combination of $\bar{s}_k(\hat{\tau}_*)$, see Equation (3)). This leads to the conclusion that $\frac{ds(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}}$ and $\frac{d^2s(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}^2}$ have a maximum on the closed set $(\hat{\tau}_*, \hat{\tau}) \in [\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \times [0, 1]$ completing the proof for Equation (21) of the lemma.

To prove the second result of the lemma, we first point out that $\bar{s}_k(\hat{\tau}_*) = 0$ for $0 \leq k \leq 2$ and $n-2 \leq k \leq n$. Therefore, we have

$$s(\hat{\tau}_*, \hat{\tau}) = \sum_{k=0}^n \bar{s}_k(\hat{\tau}_*) b_k^n(\hat{\tau}) = \sum_{k=3}^{n-3} \bar{s}_k(\hat{\tau}_*) b_k^n(\hat{\tau}).$$

Secondly, we know that $\bar{s}_k(\hat{\tau}_*) > 0$ for any $\hat{\tau}_* \in (0, 1)$ and $k = 3, 4, \dots, n-4, n-3$. Hence, using the fact that $b_k^n(\hat{\tau}) > 0$ for $\hat{\tau} \in (0, 1)$ and for all $0 \leq k \leq n$, we can conclude that

$$s(\hat{\tau}_*, \hat{\tau}) = \sum_{k=3}^{n-3} \bar{s}_k(\hat{\tau}_*) b_k^n(\hat{\tau}) > 0, \quad \hat{\tau}_*, \hat{\tau} \in (0, 1).$$

This completes the proof. \square

Using properties of the detour magnitude profile proved in Lemma 2, we now show that the K^u is guaranteed to satisfy Equation (19) if the beginning $\hat{\tau}_*^\ell$ and end $\hat{\tau}_*^u$ of the collision window satisfy $[\hat{\tau}_*^\ell, \hat{\tau}_*^u] \subset (0, 1)$.

Lemma 3. If $\hat{\tau}_* \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u] \subseteq [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u] \subset (0, 1)$ for some $\hat{\tau}_{\text{bnd}}^\ell$ and $\hat{\tau}_{\text{bnd}}^u$, then $K = K^u$ satisfies Equation (19) where $K^u = \frac{2d_{\text{safe}}}{s_\epsilon}$ and

$$s_\epsilon = \min_{\hat{\tau}_*, \hat{\tau} \in [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u]} s(\hat{\tau}_*, \hat{\tau}) - \epsilon,$$

where $\epsilon > 0$ is a some small positive constant.

Proof. From Lemma 2, we know that $s(\hat{\tau}_*, \hat{\tau}) > 0$ for $\hat{\tau}_*, \hat{\tau} \in (0, 1)$. Therefore, for the closed and bounded (compact) set $[\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u] \subset (0, 1)$, the following equation is satisfied

$$\min_{\hat{\tau}_*, \hat{\tau} \in [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u]} s(\hat{\tau}_*, \hat{\tau}) > 0,$$

and hence, there exists an $\epsilon > 0$, such that

$$s_\epsilon = \min_{\hat{\tau}_*, \hat{\tau} \in [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u]} s(\hat{\tau}_*, \hat{\tau}) - \epsilon > 0.$$

Since $[\hat{\tau}_*^\ell, \hat{\tau}_*^u] \subseteq [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u]$, we have

$$\|\Delta_K(\hat{\tau}_*, \hat{\tau})\| = \|Ks(\hat{\tau}_*, \hat{\tau})\| = K\|s(\hat{\tau}_*, \hat{\tau})\| > Ks_\epsilon = 2d_{\text{safe}}, \quad \hat{\tau}_*, \hat{\tau} \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u].$$

With this, we can use the reverse triangular inequality to conclude that

$$\min_{\hat{\tau} \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u]} \|\Delta_K(\hat{\tau}_*, \hat{\tau}) + \mathbf{d}_{\text{sub}}(\hat{\tau})\| \geq \min_{\hat{\tau} \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u]} (\|\Delta_K(\hat{\tau}_*, \hat{\tau})\| - \|\mathbf{d}_{\text{sub}}(\hat{\tau})\|).$$

Finally, we recall that $\|\mathbf{d}_{\text{sub}}(g(\hat{t}))\| = \|\mathbf{d}(\hat{t})\| \leq d_{\text{safe}}$ for the collision window $\hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u]$. Equivalently, we can say that $\|\mathbf{d}_{\text{sub}}(\hat{\tau})\| \leq d_{\text{safe}}$ for the collision window $\hat{\tau} \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u]$ and conclude that

$$\min_{\hat{\tau} \in [\hat{\tau}_*^\ell, \hat{\tau}_*^u]} \|\Delta_K(\hat{\tau}_*, \hat{\tau}) + \mathbf{d}_{\text{sub}}(\hat{\tau})\| > 2d_{\text{safe}} - d_{\text{safe}} = d_{\text{safe}}.$$

This completes the proof. \square

To conclude our analysis of the algorithm, we show that as long as the design parameters are chosen appropriately, our algorithm generates a new trajectory that is boundedly different from the original one in terms of position, velocity and acceleration, where the bounds are computable.

Theorem 1. Consider that at mission time $\hat{t} = \hat{t}_c$, a vehicle predicts a collision with reference time \hat{t}_* such that $\|\mathbf{p}(\hat{t}) - \mathbf{p}_o(\hat{t})\| \leq d_{\text{safe}}$ for $\hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u]$ and

$$(\hat{t}_*^\ell - \hat{t}_c)t^f > T_1, \quad (1 - \hat{t}_*^u)t^f > T_2, \quad (\hat{t}_*^u - \hat{t}_*^\ell)t^f < T_{\text{col}},$$

then the trajectory found through Equations (11)–(14), (15), (18), (19) and (20) with design parameters $\hat{\tau}_{ds}^\ell$ and $\hat{\tau}_{ds}^u$, that satisfy $[\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \subset [\frac{T_{\text{col}}}{\delta_{\hat{\tau}}}, 1 - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}] \subset (0, 1)$ with $\delta_{\hat{\tau}} = \min \left\{ \frac{T_2}{1 - \hat{\tau}_{ds}^u}, \frac{T_1}{\hat{\tau}_{ds}^\ell}, T_1 + T_2 \right\}$, will avert the collision, as

$$\|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}_o(\hat{t})\| > d_{\text{safe}}, \quad \forall \hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u],$$

and satisfy the mission-specific constraints

$$\mathbf{p}_{\text{new}}(1) = \mathbf{p}^f, \quad \mathbf{v}_{\text{new}}(1) = \mathbf{v}^f,$$

without changing the position, velocity and acceleration at current mission time \hat{t}_c

$$\mathbf{p}_{\text{new}}(\hat{t}_c) = \mathbf{p}(\hat{t}_c), \quad \mathbf{v}_{\text{new}}(\hat{t}_c) = \mathbf{v}(\hat{t}_c), \quad \mathbf{a}_{\text{new}}(\hat{t}_c) = \mathbf{a}(\hat{t}_c),$$

and ensuring a bounded difference in the position, velocity and acceleration

$$\begin{aligned} \max_{\hat{t} \in [0, 1]} \|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}(\hat{t})\| &< \Delta_p, & \max_{\hat{t} \in [0, 1]} \|\mathbf{v}_{\text{new}}(\hat{t}) - \mathbf{v}(\hat{t})\| &< \Delta_v \\ \max_{\hat{t} \in [0, 1]} \|\mathbf{a}_{\text{new}}(\hat{t}) - \mathbf{a}(\hat{t})\| &< \Delta_a. \end{aligned}$$

Furthermore, the bounds Δ_p, Δ_v and Δ_a are computable.

The proof of this theorem is provided in Appendix A.

Remark 1. If $\min\{T_1, T_2\} > T_{\text{col}}$, there exist design parameters $\hat{\tau}_{ds}^\ell$ and $\hat{\tau}_{ds}^u$ that satisfy

$$[\hat{\tau}_{ds}^\ell, \hat{\tau}_{ds}^u] \subset \left[\frac{T_{\text{col}}}{\delta_{\hat{\tau}}}, 1 - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}} \right] \subset (0, 1)$$

with $\delta_{\hat{\tau}} = \min \left\{ \frac{T_2}{1 - \hat{\tau}_{ds}^u}, \frac{T_1}{\hat{\tau}_{ds}^\ell}, T_1 + T_2 \right\}$. One such design choice is, for example, $\hat{\tau}_{ds}^\ell = \hat{\tau}_{ds}^u = 0.5$.

VI. Simulations

This section provides a demonstration of the efficacy of our algorithm through a typical mission scenario. We consider a specific multi-vehicle mission with assumptions on possible collisions. With this, we numerically derive position, velocity and acceleration bounds for a possible collision-avoidance maneuver and ensure its feasibility by providing sufficient margins in the planning phase. Lastly, we present simulation results for the collision-avoidance algorithm for different collision scenarios. One of these cases does not satisfy our assumptions on the collision scenario.

Mission Scenario and Assumptions: We consider a formation-flying mission with 2 quadrotors that have a maximum speed and acceleration of $v_{\max} = 5 \text{ m/s}$ and $a_{\max} = 12 \text{ m/s}^2$, respectively. We assume the quadrotors to be equipped with obstacle-detection sensors with ranges of 25 m. The obstacles are assumed to be moving with a speed of 10 m/s, whereas, a center-to-center distance of $d_{\text{safe}} = 1 \text{ m}$ is required to avoid a collision with them. Furthermore, the quadrotors are required to satisfy temporal separation of $E = 1 \text{ m}$ between each other at all times during the mission.

In this scenario, if the obstacle and the quadrotor were to travel towards each other at their respective maximum speeds, the time difference between obstacle detection and collision will be only 1.67 s. Also, if the obstacle and the quadrotor were to travel in parallel, there could be collision lasting up to 0.4 s. So we set $T_1 = 1.67 \text{ s}$, $T_{\text{col}} = 0.4 \text{ s}$ and assume $T_2 = T_1$.

Calculation of Bounds and Initial Trajectory Generation: For this example, we set our design parameters as $\hat{\tau}_{\text{ds}}^l = 0.48$ and $\hat{\tau}_{\text{ds}}^u = 0.52$, and get the following bounds for a possible collision-avoidance maneuver:

$$\Delta_p = 2.95 \text{ m}, \quad \Delta_v = 3.24 \text{ m/s}, \quad \Delta_a = 7.72 \text{ m/s}^2.$$

Utilizing these bounds, we use the trajectory-generation algorithm to generate trajectories

$$\mathbf{p}_1 : [0, 1] \rightarrow \mathbb{R}^2 \quad \text{and} \quad \mathbf{p}_2 : [0, 1] \rightarrow \mathbb{R}^2,$$

that satisfy the temporal separation constraint as

$$\|\mathbf{p}_1(\hat{t}) - \mathbf{p}_2(\hat{t})\| \geq E + \Delta_p = 3.95 \text{ m}, \quad \forall \hat{t} \in [0, 1].$$

Furthermore, we impose maximum velocity and acceleration constraints as

$$\begin{aligned} \max_{\hat{t} \in [0, 1]} \|\mathbf{v}_i(\hat{t})\| &\leq v_{\max} - \Delta_v = 1.76 \text{ m/s}, & i &= 1, 2, \\ \max_{\hat{t} \in [0, 1]} \|\mathbf{a}_i(\hat{t})\| &\leq a_{\max} - \Delta_a = 4.28 \text{ m/s}^2, & i &= 1, 2. \end{aligned}$$

Please note that these constraints are restrictive to allow for aggressive collision-avoidance maneuvers in order to avoid fast obstacles.

The trajectory-generation algorithm generates trajectories that satisfy all constraints of the mission and $t^f = 10 \text{ s}$. The planned trajectories are shown in Figure 8. Notice that if one of the vehicles, for example vehicle 1, encounters an obstacle during mission execution, such that our assumptions in terms of T_1 , T_2 and T_{col} are satisfied, then the new trajectory $\mathbf{p}_{1,\text{new}}$ is guaranteed to satisfy $\|\mathbf{p}_{1,\text{new}}(\hat{t}) - \mathbf{p}_1(\hat{t})\| < \Delta_p$, $\|\mathbf{v}_{1,\text{new}}(\hat{t}) - \mathbf{v}_1(\hat{t})\| < \Delta_v$ and $\|\mathbf{a}_{1,\text{new}}(\hat{t}) - \mathbf{a}_1(\hat{t})\| < \Delta_a$ for $\hat{t} \in [0, 1]$. Then through the set of constraints in the planned trajectories, temporal separation is ensured, since

$$\|\mathbf{p}_{1,\text{new}}(\hat{t}) - \mathbf{p}_2(\hat{t})\| > E, \quad \forall \hat{t} \in [0, 1].$$

Furthermore, bounds on the velocity and acceleration profiles of the new trajectory are guaranteed to satisfy

$$\begin{aligned} \max_{\hat{t} \in [0, 1]} \|\mathbf{v}_{1,\text{new}}(\hat{t})\| &< v_{\max}, \\ \max_{\hat{t} \in [0, 1]} \|\mathbf{a}_{1,\text{new}}(\hat{t})\| &< a_{\max}. \end{aligned}$$

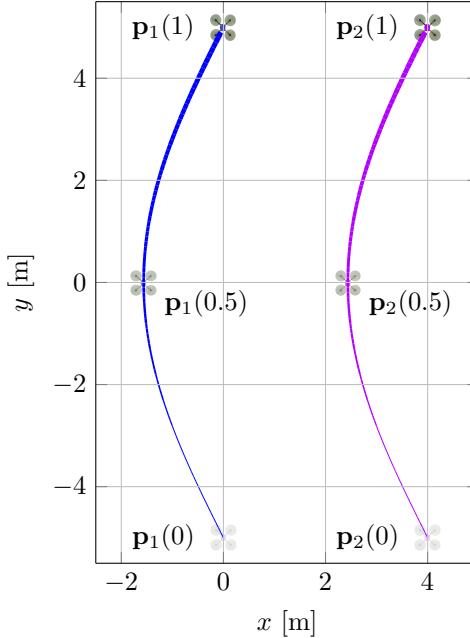


Figure 8: Desired trajectories obtained from the trajectory-generation algorithm for the formation-flying mission with 2 quadrotors

In other words, we know that the new trajectory will also satisfy the temporal separation constraint along with dynamic constraints of the vehicle. Note that Theorem 1 ensures that other required constraints will also be satisfied. For example, we are guaranteed that the new and the original trajectory will have the same position, velocity and acceleration at the time instant \hat{t}_c , when the obstacle is detected. Furthermore, mission-specific constraints will also be satisfied.

Simulation Results: We consider two different cases^a. In the first case, we simulate an obstacle traveling at 10 m/s on a collision course with vehicle 1 such that $\hat{t}_*^\ell = 0.492$ and $\hat{t}_*^u = 0.511$. Please note that our algorithm does not need \hat{t}_*^ℓ and \hat{t}_*^u . However, for analysis we can confirm that our assumptions $(1 - \hat{t}_*)^f > T_2$ and $(\hat{t}_*^u - \hat{t}_*^\ell)^f < T_{\text{col}}$ are now satisfied. Furthermore, the onboard obstacle detection sensor with a range of 25 m ensures that our last remaining assumption $(\hat{t}_*^u - \hat{t}_c)^f > T_1$ is also satisfied. With all three assumptions met, we are guaranteed that all required constraints will be satisfied by the new trajectory.

In the simulation, vehicle 1 detects the obstacle at mission time $\hat{t}_c = 0.27$ and predicts a collision with reference time $\hat{t}_* = 0.5$. It generates the new trajectory $p_{1,\text{new}}$ for which a time lapse is shown in Figure 9. The distances of vehicle 1 to the obstacle and vehicle 2 are plotted in Figures 10 and 11, respectively, whereas the velocity and acceleration profiles are shown in Figure 12. It is clear that all our constraints are satisfied, while the collision is avoided.

For the second case, we consider a static obstacle blocking the path for vehicle 1. In this case, we have $\hat{t}_*^\ell = 0.4$ and $\hat{t}_*^u = 0.6$. It follows that our assumption $(\hat{t}_*^u - \hat{t}_*^\ell)^f < T_{\text{col}}$ is not satisfied. However, we show that our algorithm successfully avoids the collision despite violating the assumption.

During the simulation, vehicle 1 detects the obstacle immediately at the start of the mission $\hat{t}_c = 0$ and predicts a collision with reference time $\hat{t}_* = 0.5$. It then generates the new trajectory $p_{1,\text{new}}$ for which a time lapse is shown in Figure 13. The distances of vehicle 1 to the obstacle and vehicle 2 are plotted in Figures 14 and 15, respectively, whereas the velocity and acceleration profiles are shown in Figure 16. We see that all constraints are still satisfied by the algorithm. This, in fact, is not surprising since the derived constraints represent worst-case scenarios.

^aA video for both the cases can be found at <https://www.youtube.com/watch?v=oOkSVTVsDHU> [Online; accessed 19 January 2015]

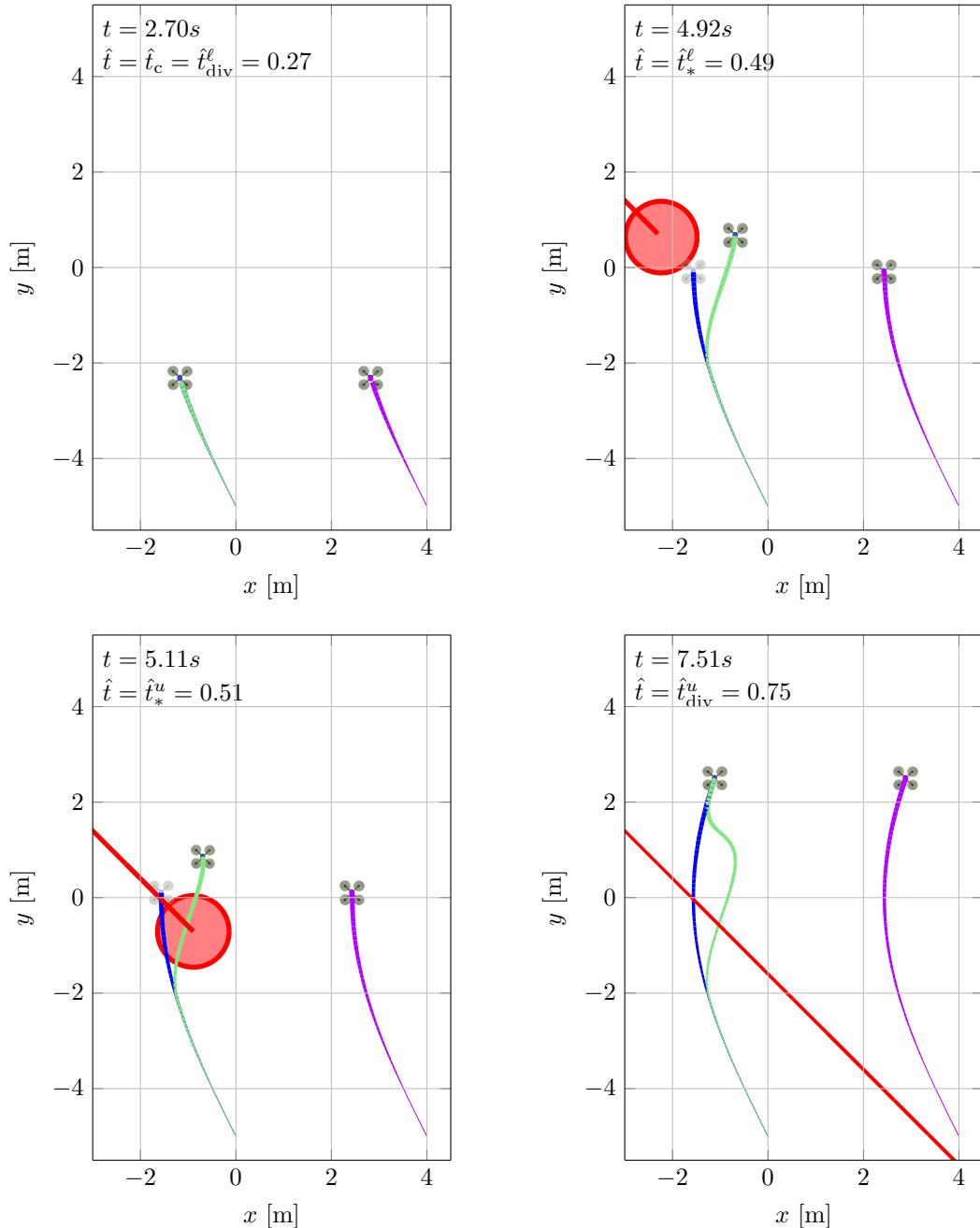


Figure 9: Simulation of case 1 shown at different times with green and purple trails representing the trajectories followed by the vehicles. The translucent quadrotor with blue trail shows the mission progress without the collision-avoidance maneuver.

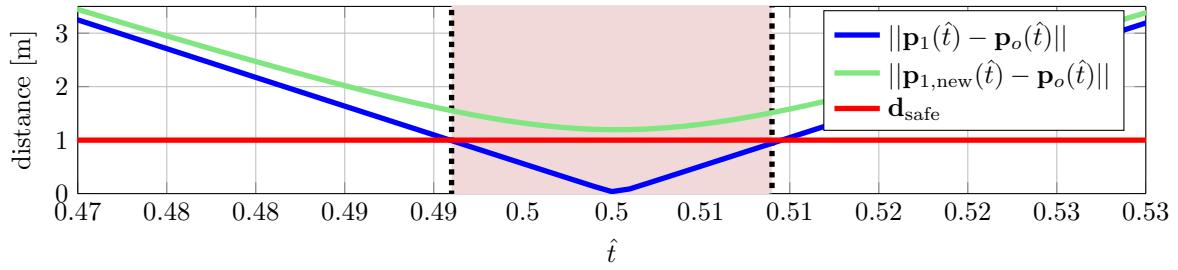


Figure 10: Distance between vehicle 1 and the obstacle for case 1. The red region represents the collision window $[\hat{t}_{\text{div}}^{\ell}, \hat{t}_{\text{div}}^u]$.

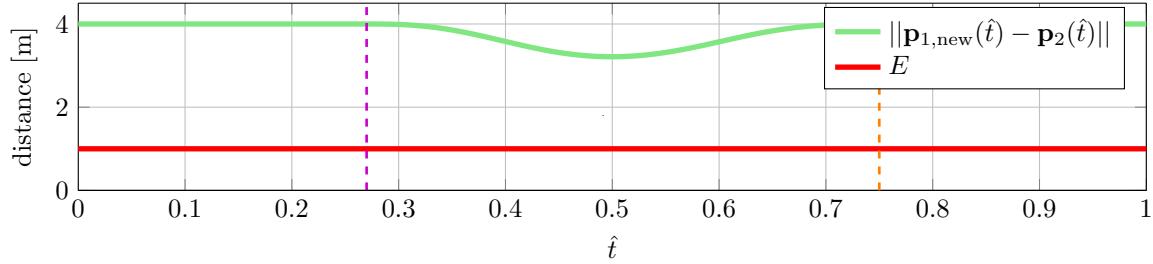


Figure 11: Distance between the two vehicles for case 1. The dashed purple and orange lines represent $\hat{t}_{\text{div}}^{\ell}$ and \hat{t}_{div}^u , respectively.

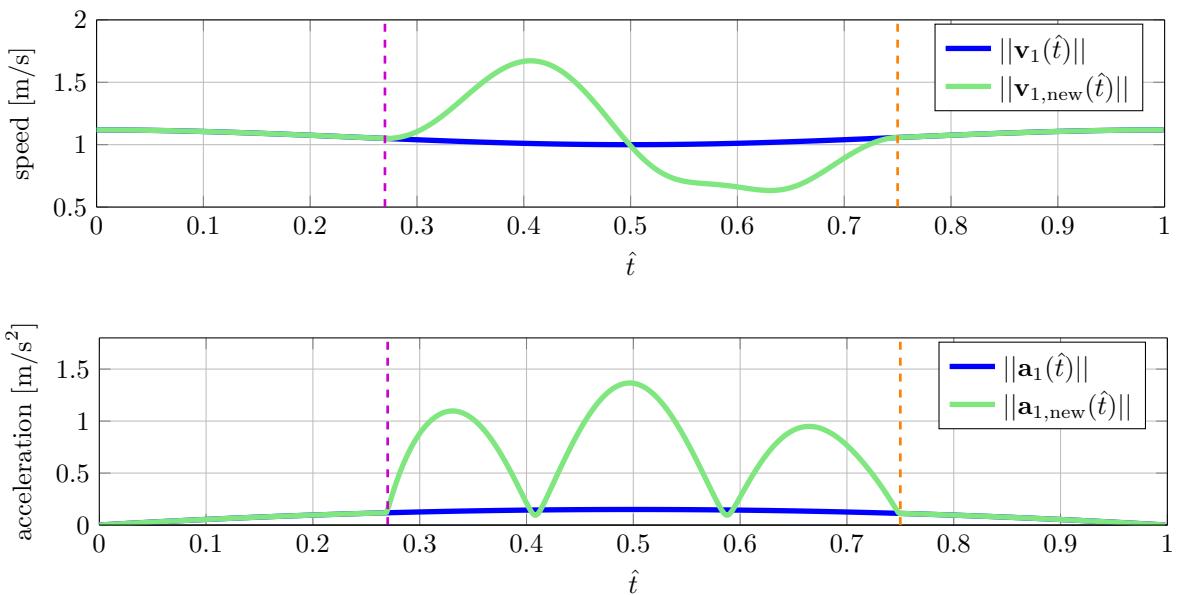


Figure 12: Speed and acceleration of vehicle 1 for case 1. The dashed purple and orange lines represent $\hat{t}_{\text{div}}^{\ell}$ and \hat{t}_{div}^u , respectively.

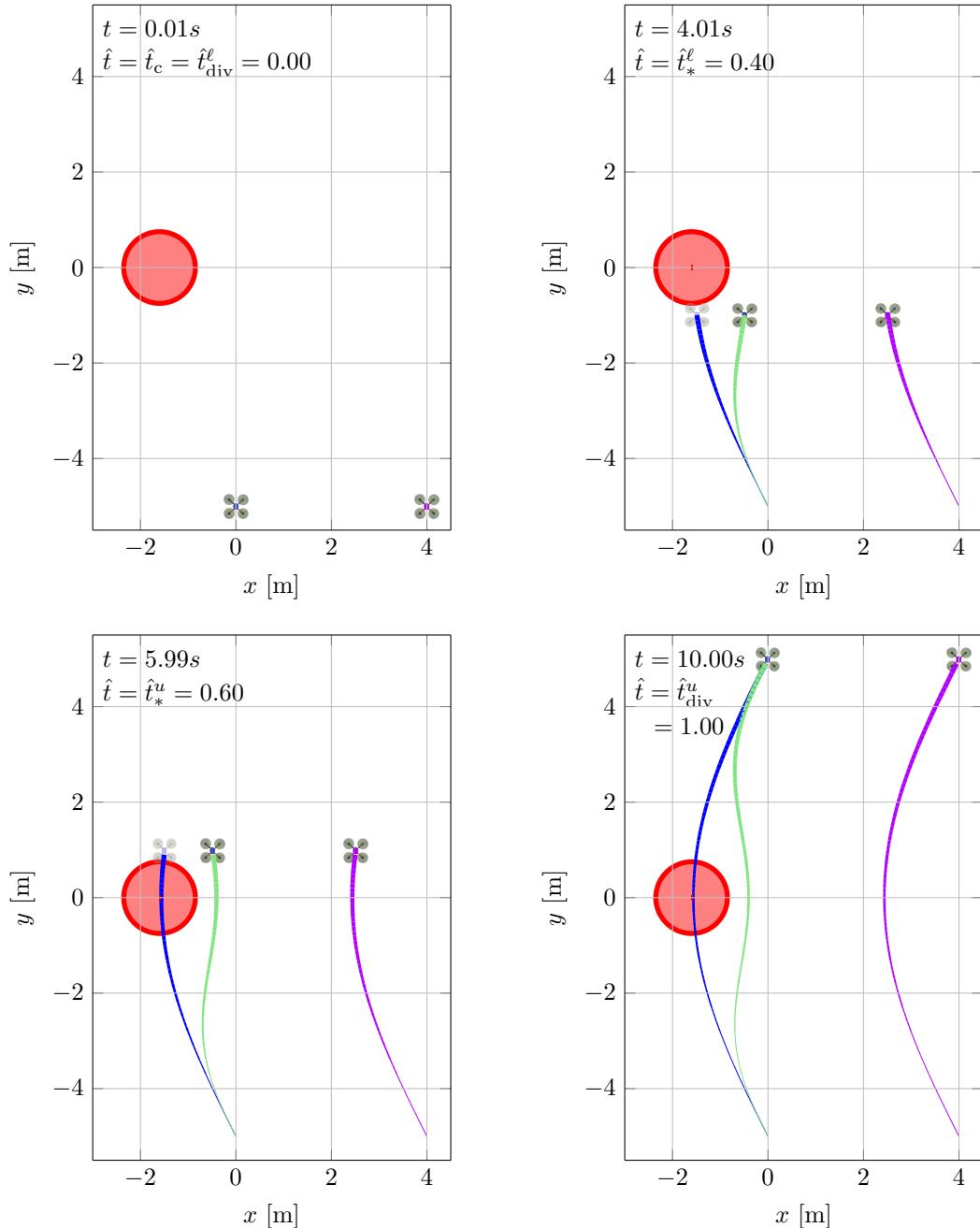


Figure 13: Simulation of case 2 shown at different times with green and purple trails representing the trajectories followed by the vehicles. The translucent quadrotor with blue trail shows the mission progress without the collision avoidance maneuver.

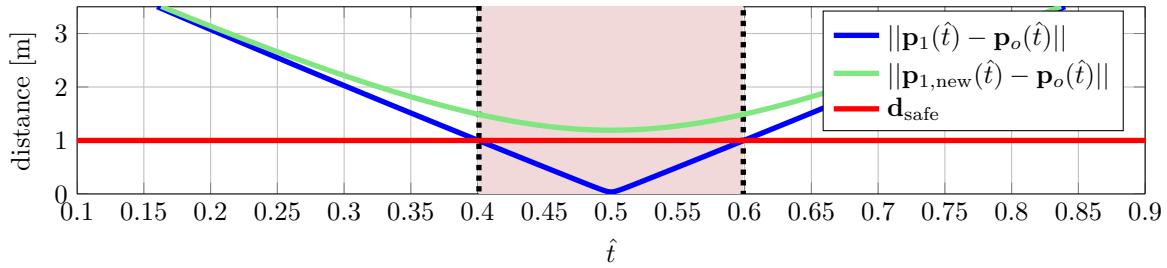


Figure 14: Distance between vehicle 1 and the obstacle for case 2. The shaded region represents the collision window $[\hat{t}_*^l, \hat{t}_*^u]$.

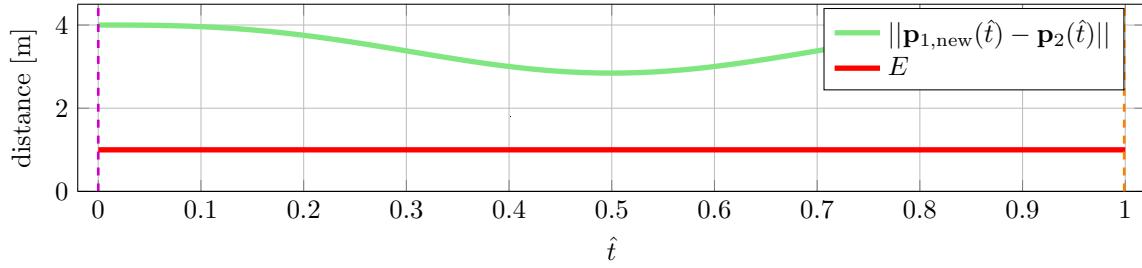


Figure 15: Distance between the two vehicles for case 2. The dashed purple and orange lines represent $\hat{t}_{\text{div}}^l = \hat{t}_c$ and $\hat{t}_{\text{div}}^u = 1$, respectively.

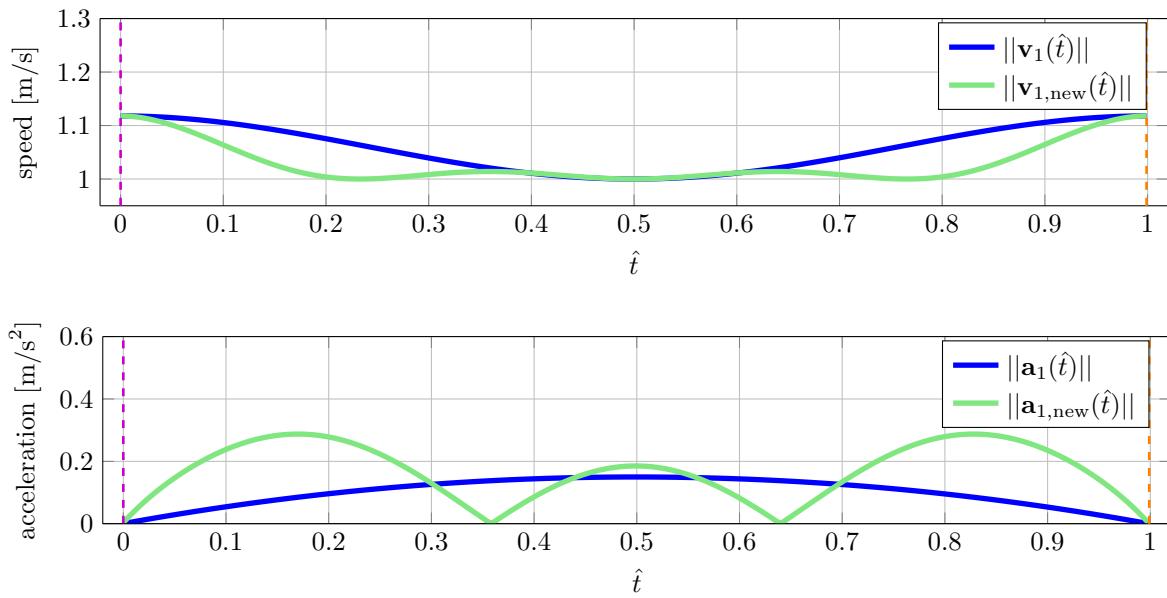


Figure 16: Speed and acceleration of vehicle 1 for case 2. The dashed purple and orange lines represent $\hat{t}_{\text{div}}^l = \hat{t}_c$ and $\hat{t}_{\text{div}}^u = 1$, respectively.

VII. Conclusion

In this paper, an algorithm for collision prediction and avoidance using Bézier curves has been presented. The proposed algorithm replans the original trajectory such that a possible collision in the future is avoided, whereas continuity requirements for the trajectory along with mission-specific constraints are satisfied. Under some mild assumptions on the collision scenarios, calculation of bounds on position, velocity and acceleration changes in the trajectory caused by the algorithm are computable offline. These bounds can be used during the trajectory-generation phase, to ensure that the dynamic constraints are satisfied in case the vehicle replans the trajectory during mission execution. A demonstration of our algorithm for a formation-flying mission for two quadrotors is also provided.

Future works include extending this algorithm to more general settings. One such extension will be for collisions that last for extended periods of time. The detour magnitude profile used in this paper is not suitable for such cases, however, slight modifications will allow the algorithm to avoid these collisions with computable bounds. Another extension will be towards a three-dimensional setting where the choice for detour direction is considerably more complex. Lastly, the algorithm will be analyzed for more types of vehicles such as fixed-wing aircraft that are subject to more stringent dynamic constraints.

References

- ¹ “Concept of operations for the next generation air transportation system,” Joint Planning and Development Office, Version 3.2, September 2010, available at <http://www.jpdo.gov/library.asp>.
- ²D. Mellinger, M. Shomin, N. Michael, and V. Kumar, “Cooperative grasping and transport using multiple quadrotors,” in *Distributed autonomous robotic systems*, 2013, pp. 545–558.
- ³N. Michael, J. Fink, and V. Kumar, “Cooperative manipulation and transportation with aerial robots,” *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.
- ⁴U. Gurcuoglu, G. A. Puerto-Souza, F. Morbidi, and G. L. Mariottini, “Hierarchical control of a team of quadrotors for cooperative active target tracking,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 5730–5735.
- ⁵R. Ritz, M. W. Müller, M. Hehn, and R. D’Andrea, “Cooperative quadrocopter ball throwing and catching,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 4972–4978.
- ⁶R. Ritz and R. D’Andrea, “Carrying a flexible payload with multiple flying vehicles,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3465–3471.
- ⁷S. Gupte, P. I. T. Mohandas, and J. M. Conrad, “A survey of quadrotor unmanned aerial vehicles,” in *Proceedings of IEEE Southeastcon 2012 conference*. IEEE, 2012, pp. 1–6.
- ⁸A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, “Towards a swarm of agile micro quadrotors,” *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.
- ⁹V. Cichella, I. Kaminer, E. Xargay, V. Dobrokhotov, N. Hovakimyan, A. P. Aguiar, and A. M. Pascoal, “A Lyapunov-based approach for time-coordinated 3D path-following of multiple quadrotors,” in *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, 2012, pp. 1776–1781.
- ¹⁰V. Cichella, I. Kaminer, V. Dobrokhotov, E. Xargay, R. Choe, N. Hovakimyan, A. P. Aguiar, and A. M. Pascoal, “Cooperative path-following of multiple multirotors over time-varying networks,” *IEEE Transaction on Automation Science and Engineering*, 2014, Accepted.
- ¹¹R. T. Farouki, *Pythagorean-Hodograph Curves*. Berlin Heidelberg: Springer-Verlag, 2008.
- ¹²L. Fowler and J. Rogers, “Bézier curve path planning for parafoil terminal guidance,” *Journal of Aerospace Information Systems*, vol. 11, no. 5, pp. 300–315, 2014.
- ¹³J. Choi, R. Curry, and G. Elkaim, “Path planning based on Bézier curve for autonomous ground vehicles,” in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, WCECS ’08*, Oct 2008, pp. 158–166.
- ¹⁴J. Connors and G. Elkaim, “Manipulating B-spline based paths for obstacle avoidance in autonomous ground vehicles,” in *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation*, 2007, pp. 1081–1088.
- ¹⁵K. Yang and S. Sukkarieh, “Planning continuous curvature paths for UAVs amongst obstacles,” in *Australas. Conf. Robot. Autom.*, Dec 2008, pp. 1081–1088.
- ¹⁶R. Choe, V. Cichella, E. Xargay, N. Hovakimyan, A. C. Trujillo, and I. Kaminer, “A trajectory-generation framework for time-critical cooperative missions,” in *AIAA Infotech@Aerospace*, Boston, MA, August 2013, AIAA 2013-4582.
- ¹⁷R. Choe, J. Puig-Navarro, V. Cichella, E. Xargay, and N. Hovakimyan, “Trajectory generation using spatial Pythagorean Hodograph Bézier curves,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2015.
- ¹⁸V. Cichella, R. Choe, S. B. Mehdi, E. Xargay, N. Hovakimyan, V. Dobrokhotov, and I. Kaminer, “Trajectory generation and collision avoidance for safe operation of cooperating UAVs,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2014.
- ¹⁹J.-W. Chang, Y.-K. Choi, M.-S. Kim, and W. Wang, “Computation of the minimum distance between two Bézier curves/surfaces,” *Computers & Graphics*, vol. 35, no. 3, pp. 677–684, June 2011.

Appendix

A. Proof of Theorem 1

Proof. We start off by showing that the mission-specific constraints, along with continuity constraints at time of obstacle detection \hat{t}_c are satisfied. From Equation (11), it follows that $[\hat{t}_{\text{div}}^\ell, \hat{t}_{\text{div}}^u] \subseteq [\hat{t}_c, 1]$. Recalling our results for $s(\hat{\tau}_*, \hat{\tau})$ from Equation (16) and (17), it follows that the new trajectory defined in Equation (20) satisfies

$$\mathbf{p}_{\text{new}}(1) = \mathbf{p}^f, \quad \mathbf{v}_{\text{new}}(1) = \mathbf{v}^f, \quad \mathbf{p}_{\text{new}}(\hat{t}_c) = \mathbf{p}(\hat{t}_c), \quad \mathbf{v}_{\text{new}}(\hat{t}_c) = \mathbf{v}(\hat{t}_c), \quad \mathbf{a}_{\text{new}}(\hat{t}_c) = \mathbf{a}(\hat{t}_c).$$

Now, we start deriving bounds for the detour. Our first step will be to ensure bounds for the magnitude profile $s(\hat{\tau}_*, \hat{\tau})$ and its derivatives for $\hat{\tau} \in [0, 1]$. From Lemma 1, we know that $\hat{\tau}_* \in [\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u]$. Then Lemma 2 shows the existence of the bounds s_{\max}, s'_{\max} and s''_{\max} , where

$$s_{\max} = \max_{\substack{\hat{\tau}_* \in [\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \\ \hat{\tau} \in [0, 1]}} s(\hat{\tau}_*, \hat{\tau}), \quad s'_{\max} = \max_{\substack{\hat{\tau}_* \in [\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \\ \hat{\tau} \in [0, 1]}} \frac{ds(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}}, \quad s''_{\max} = \max_{\substack{\hat{\tau}_* \in [\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \\ \hat{\tau} \in [0, 1]}} \frac{d^2 s(\hat{\tau}_*, \hat{\tau})}{d\hat{\tau}^2}.$$

Here s_{\max}, s'_{\max} and s''_{\max} are computable.

Next, we show that $[\hat{\tau}_*^\ell, \hat{\tau}_*^u] \subseteq [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u]$ for $\hat{\tau}_{\text{bnd}}^\ell = \hat{\tau}_{\text{ds}}^\ell - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}, \hat{\tau}_{\text{bnd}}^u = \hat{\tau}_{\text{ds}}^u + \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}$. In this direction, we first show that $\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell$ is always bounded from below as

$$\begin{aligned} \hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell &\geq \min \left\{ 1 - \frac{\hat{t}_* - \hat{\tau}_{\text{ds}}^u}{1 - \hat{\tau}_{\text{ds}}^u}, \frac{\hat{t}_* - \hat{t}_c + \hat{\tau}_{\text{ds}}^\ell \hat{t}_c}{\hat{\tau}_{\text{ds}}^\ell} - \hat{t}_c, 1 - \hat{t}_c \right\} \\ &= \min \left\{ \frac{1 - \hat{t}_*}{1 - \hat{\tau}_{\text{ds}}^u}, \frac{\hat{t}_* - \hat{t}_c}{\hat{\tau}_{\text{ds}}^\ell}, 1 - \hat{t}_c \right\} \\ &\geq \min \left\{ \frac{T_2}{t^f(1 - \hat{\tau}_{\text{ds}}^u)}, \frac{T_1}{t^f \hat{\tau}_{\text{ds}}^\ell}, \frac{T_1 + T_2}{t^f} \right\} \\ &= \frac{1}{t^f} \min \left\{ \frac{T_2}{1 - \hat{\tau}_{\text{ds}}^u}, \frac{T_1}{\hat{\tau}_{\text{ds}}^\ell}, T_1 + T_2 \right\} = \frac{1}{t^f} \delta_{\hat{\tau}}, \end{aligned}$$

where we define $\delta_{\hat{\tau}} = \min \left\{ \frac{T_2}{1 - \hat{\tau}_{\text{ds}}^u}, \frac{T_1}{\hat{\tau}_{\text{ds}}^\ell}, T_1 + T_2 \right\}$.

Since, $[\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u] \subset [\frac{T_{\text{col}}}{\delta_{\hat{\tau}}}, 1 - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}]$, it follows that $[\hat{\tau}_{\text{ds}}^\ell - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}, \hat{\tau}_{\text{ds}}^u + \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}] \subset (0, 1)$. Now using the fact $\hat{\tau}_* \in [\hat{\tau}_{\text{ds}}^\ell, \hat{\tau}_{\text{ds}}^u]$, it follows that

$$\begin{aligned} \hat{\tau}_*^\ell = g(\hat{t}_*^\ell) &= \frac{\hat{t}_*^\ell - \hat{t}_{\text{div}}^\ell}{\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell} > \frac{\hat{t}_* - \frac{T_{\text{col}}}{t^f} - \hat{t}_{\text{div}}^\ell}{\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell} = \hat{\tau}_* - \frac{T_{\text{col}}}{t^f(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell)} \geq \hat{\tau}_{\text{ds}}^\ell - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}} > 0, \\ \hat{\tau}_*^u = g(\hat{t}_*^u) &= \frac{\hat{t}_*^u - \hat{t}_{\text{div}}^u}{\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell} < \frac{\hat{t}_* + \frac{T_{\text{col}}}{t^f} - \hat{t}_{\text{div}}^u}{\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell} = \hat{\tau}_* + \frac{T_{\text{col}}}{t^f(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell)} \leq \hat{\tau}_{\text{ds}}^u + \frac{T_{\text{col}}}{\delta_{\hat{\tau}}} < 1. \end{aligned}$$

In other words, we have $[\hat{\tau}_*^\ell, \hat{\tau}_*^u] \subseteq [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u] \subset (0, 1)$ where $\hat{\tau}_{\text{bnd}}^\ell = \hat{\tau}_{\text{ds}}^\ell - \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}$ and $\hat{\tau}_{\text{bnd}}^u = \hat{\tau}_{\text{ds}}^u + \frac{T_{\text{col}}}{\delta_{\hat{\tau}}}$. Using Lemma 3 we conclude that $K^u = \frac{2d_{\text{safe}}}{s_\epsilon}$ satisfies Equation (19) where

$$s_\epsilon = \min_{\hat{\tau}_*, \hat{\tau} \in [\hat{\tau}_{\text{bnd}}^\ell, \hat{\tau}_{\text{bnd}}^u]} s(\hat{\tau}_*, \hat{\tau}) - \epsilon.$$

for some small $\epsilon > 0$.

Since K is chosen from the set $K \in (K^\ell, K^u]$, we have $K \leq \frac{2d_{\text{safe}}}{s_\epsilon}$ such that K satisfies Equation (19). This also implies that

$$\|\mathbf{d}_{\text{new}}(\hat{t})\| = \|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}_o(\hat{t})\| > d_{\text{safe}}, \quad \forall \hat{t} \in [\hat{t}_*^\ell, \hat{t}_*^u].$$

Since,

$$\begin{aligned} \max_{\hat{t} \in [0,1]} \|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}(\hat{t})\| &= \max_{\hat{t} \in [0,1]} \begin{cases} \|Ks(\hat{\tau}_*, \hat{\tau})\mathbf{u}\| & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0 & \text{otherwise,} \end{cases} \\ \max_{\hat{t} \in [0,1]} \|\mathbf{v}_{\text{new}}(\hat{t}) - \mathbf{v}(\hat{t})\| &= \max_{\hat{t} \in [0,1]} \begin{cases} \left\| \frac{K\mathbf{u}}{(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell)t^f} \frac{ds(\hat{\tau}_*, g(\hat{t}))}{dg(\hat{t})} \right\| & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0 & \text{otherwise,} \end{cases} \\ \max_{\hat{t} \in [0,1]} \|\mathbf{a}_{\text{new}}(\hat{t}) - \mathbf{a}(\hat{t})\| &= \max_{\hat{t} \in [0,1]} \begin{cases} \left\| \frac{K\mathbf{u}}{(\hat{t}_{\text{div}}^u - \hat{t}_{\text{div}}^\ell)^2(t^f)^2} \frac{d^2 s(\hat{\tau}_*, g(\hat{t}))}{(dg(\hat{t}))^2} \right\| & \text{if } \hat{t}_{\text{div}}^\ell < \hat{t} \leq \hat{t}_{\text{div}}^u, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

it follows that

$$\begin{aligned} \max_{\hat{t} \in [0,1]} \|\mathbf{p}_{\text{new}}(\hat{t}) - \mathbf{p}(\hat{t})\| &\leq \frac{2d_{\text{safe}}s_{\max}}{s_\epsilon} = \Delta_p, \\ \max_{\hat{t} \in [0,1]} \|\mathbf{v}_{\text{new}}(\hat{t}) - \mathbf{v}(\hat{t})\| &\leq \frac{2d_{\text{safe}}s'_{\max}}{\delta_{\hat{\tau}}s_\epsilon} = \Delta_v, \\ \max_{\hat{t} \in [0,1]} \|\mathbf{a}_{\text{new}}(\hat{t}) - \mathbf{a}(\hat{t})\| &\leq \frac{2d_{\text{safe}}s''_{\max}}{\delta_{\hat{\tau}}^2s_\epsilon} = \Delta_a, \end{aligned}$$

This completes the proof. \square