An imminent collision monitoring system with safe stopping interventions for autonomous aerial flights

Jasmine Cheng, Xuning Yang and Nathan Michael

***Abstract*— Collision avoidance requires tradeoffs in planning time horizons. Depending on the planner, safety cannot always be guaranteed in uncertain environments given map updates. To mitigate situations where the planner leads the vehicle into a state of collision or the vehicle reaches a point where no trajectories are feasible, we propose a continuous collision checking algorithm. The imminent collision checking system continuously monitors vehicle safety, and plans a safe trajectory that leads the vehicle to a stop within the observed map. We test our proposed pipeline alongside a teleoperated navigation in a random-forest environment and a realistic warehouse method and show that with our method, we are able to mitigate collisions with a success rate of at least 90%.**

I. Introduction

In tasks such as navigation and exploration in dense environments, static and dynamic collision avoidance require trade-offs in planning time horizons. Planning at a high frequency results in increased demand for computation at unrealistic rates, while planning at low frequencies may result in inaccurate plans if map updates between planning iterations. This work seeks to improve aerial vehicle resilience and robustness in a variety of environments by introducing a safety monitoring system that can supplement any autonomous or semi-autonomous aerial system. Prior work has explored collision avoidance and replanning with respect to goal-seeking, where the vehicle is autonomously controlled and the goal is known a priori [1]. The proposed method mitigates possible collisions at map update rate in a lightweight way using the vehicle state, preventing the vehicle from reaching an unrecoverable state in situations where the planner fails to mitigate the collision in time.

The proposed system prevents collisions by continuously monitoring collision safety and generating a stopping trajectory if the vehicle is at risk of collision. This system is first included in [2]. In this work, we demonstrate the imminent collision monitoring system in a variety of scenarios and show its applicability and robustness with respect to a wide variety of vehicle and environment conditions. We show that, with the inclusion of the proposed system across a variety of autonomous/semi-autonomous flights, we are able to detect unsafe states and safely avoid collisions.

II. Method

The proposed method proceeds as follows: the system continuously monitors collision possibility by evaluating the closest obstacles to the vehicle's current location at a fixed rate. Then, we generate a dynamically feasible and safe trajectory and bring the vehicle to a stop by first selecting an escape point along the direction of motion, then generating a trajectory to the escape point. The method is detailed as follows.

The Authors are with the Robotics Institute at Carnegie Mellon University, Pittsburgh PA, 15213, USA {xuning, nmichael}@cmu.edu

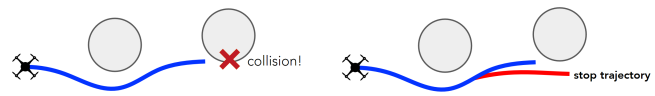

Fig. 1: Trajectory-based teleoperation, current trajectory is collision free, but imminent collision after the trajectory.

A. Imminent collision checking

We first find the set of possible collision points nearby by evaluating the normalized vector projection between a set of closest obstacle locations $\{\mathbf{p}\}$ given by a known map and the vehicle velocity:

$$\text{proj}(\mathbf{x},\mathbf{p}) = \left\langle \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}, \frac{\mathbf{p}-\mathbf{x}}{\|\mathbf{p}-\mathbf{x}\|} \right\rangle = \left\langle \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}, \frac{\mathbf{r}}{\|\mathbf{r}\|} \right\rangle$$

Where $\mathbf{x}$, $\dot{\mathbf{x}}$ is the vehicle position and velocity respectively, and the vector to an obstacle point $\mathbf{p}$ is $\mathbf{r} = \mathbf{p} - \mathbf{x}$. The points away from the direction of motion, e.g., $\text{proj}(\mathbf{x},\mathbf{p})<0$ are discarded.

Then, for all points $\{\mathbf{p}\}$ in remaining set, we compute a combined stop criterion based on distance to the obstacle, speed, and the angle offset:

$$C_{\text{stop}}(\mathbf{x},\mathbf{p}) = w_1 \|\mathbf{r}\| - w_2 \|\dot{\mathbf{x}}\| + w_3 \arccos(\text{proj}(\mathbf{x},\mathbf{p})) \quad \text{if } \text{proj}(\mathbf{x},\mathbf{p}) >= 0$$

The third term represents the relative angle difference between the heading of the vehicle and the direction of the obstacle point, as shown in Figure 3. A imminent stop trajectory is issued if for any obstacle point $\mathbf{p}$, $C_{\text{stop}}(\mathbf{x},\mathbf{p}) < \beta$.

B. Escape points sampling

Escape points are defined as possible goal points where the vehicle could stop. We first generate escape points and then plan trajectories to them. The first feasible trajectory incurring the lowest cost is sent to the controller. Thus, to minimize search time, we aim to check escape points that are more likely to produce feasible trajectories.

An initial group of escape points $\{\mathbf{e}\}$ is generated using a uniform 3D grid, scaled proportionally to the vehicle's velocity. Points are sampled along the direction of motion using stratified sampling for points near the vehicle's original heading via the following cost:

$$C_{\text{escape}}(\mathbf{x},\mathbf{e}) = w_1 q + w_2 d$$

Where $q{=}||(\mathbf{x}{-}\mathbf{e}{-}((\mathbf{x}{-}\mathbf{e})\cdot\hat{\dot{\mathbf{x}}})\hat{\dot{\mathbf{x}}}||$ is the shortest distance from the escape point $\mathbf{e}$ to the line $l(s){=}\hat{\dot{\mathbf{x}}}s{+}\mathbf{x}$ along the velocity vector, $\hat{\dot{\mathbf{x}}}$ is the normalized velocity vector, and d is the distance from the the escape point to its nearest obstacle.

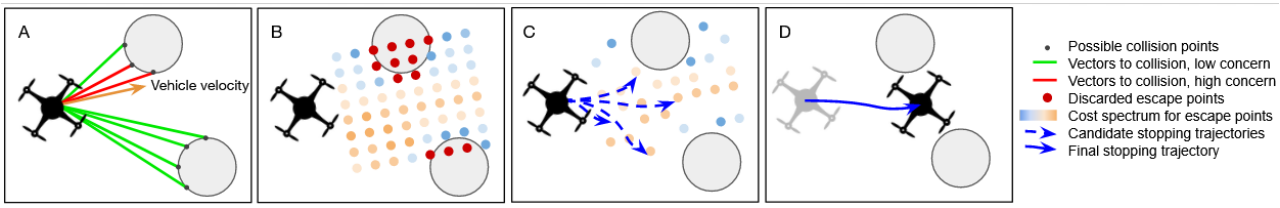

Fig. 2: Illustration of imminent collision monitoring and safe stop. (A) Initial assessment: possible collision points are sampled from nearby obstacles and vectors to collision are assessed. If high concern vectors exist, stopping trajectory generation begins. (B) Initial grid of escape points with costs computed and colliding points discarded (C) A stratified sampling method is used to down sample the possible escape points, with candidate trajectories generated. Dynamically infeasible candidate trajectories are discarded. (D) Lowest cost, dynamically feasible trajectory is selected and sent to the controller.

First, points farther from obstacles are lower cost to encourage the vehicle towards an area with more free space. Second, points closer to the vehicles velocity vector are given lower costs. Points farther from the vehicles path require the robot to turn severely, which results in undesirable dynamics.

Next, points are downsampled before trajectory generation via stratified sampling, which samples a fixed number of points from each strata based on cost. To ensure enough high quality points are chosen, more points are sampled from higher strata (low cost). In this implementation, 10 points are sampled from top 1%, 40 points from next 9%, 30 from next 40% and 20 from bottom 50%. The stratified method guarantees a set of points with both high and low costs such that in the event low cost trajectories are not dynamically feasible, we are able to sacrifice cost for dynamic feasibility.

C. Stopping trajectory generation

Given the set of escape points, we generate candidate stop trajectories starting with the lowest C_{escape}.

A trajectory is a time-parameterized function $\xi(t)$, defined over a time interval $t \in [0, T]$ that maps a given time t to a position $\mathbf{x}_t$. A stop trajectory is a trajectory that brings a vehicle from in-motion to at-rest:

$$\begin{aligned} \text{s.t.} \quad & \xi : [0, T] \to X \\ & \xi^{(j)}(T) = 0, j > 0 \\ & \xi \in \Xi_{\text{safe}} \end{aligned}$$

A single polynomial trajectory is computed by solving an 8th order polynomial on the current vehicle state and desired vehicle state. The current vehicle state is defined by the current position, velocity, acceleration, and jerk in each dimension (x, y, z, yaw). The desired state is the position of the escape point, with all higher-order derivatives of 0. Each dimension's polynomial is computed independently.

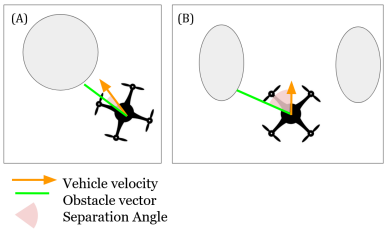

Fig. 3: (A) Vehicle is heading in the direction of an obstacle. Vehicle is in higher risk of collision, C_{stop} is lower. (B) Vehicle is heading between obstacles/ Vehicle is in lower risk of collision, C_{stop} higher.

We solve for the coefficients, c_i, of the polynomial

$$f(x) = \sum_{i=0}^{8} c_i x_i$$

with constraints

$$\begin{aligned} & f(0) = x_0, \dot{f}(0) = v_0, \ddot{f}(0) = a_0, f^{(3)}(0) = j_0 \\ & f(t) = x_f, \dot{f}(t), \ddot{f}(t), f^{(3)}(t) = 0 \end{aligned}$$

where x_0, v_0, a_0, j_0 are the initial state and x_f is the final position given by the escape point.

Next, the trajectory is interpolated and waypoints are checked against a known map to ensure it is collision-free. The trajectory is also checked for dynamic feasibility via an acceleration bound along the trajectory, i.e., $\ddot{\mathbf{x}} < \bar{\alpha}$. For our experiments, we select $\bar{\alpha} = 10\text{m/s}^2$.

The first trajectory that is safe and feasible is executed, and the search stops. If all potential escape points have been exhausted without finding a valid trajectory, then a stopping trajectory with no goal position is executed as a last effort.

III. Experiment and Results

We evaluate our method in a simulated random forest environment and in a dense realistic warehouse environment in a teleoperated navigation task minimizing time. We conduct 20 trials in each environment, 10 with the proposed pipeline enabled, and 10 without. In our experiments, the stop criterion parameters w_1, w_2, w_3, β are chosen to be 0.6, 0.4, 1.2, 0.3 respectively.

We discuss task completion with and without enabling collision monitoring in Table I. With collision avoidance enabled, the operator completes the task with a success rate of 90% for the forest scenario and 100% for the warehouse scenario. When collision monitoring is disabled, the vehicle collides under aggressive flight, leading to 20% task completion rate. The average velocity of enabled collision monitoring is comparable to disabled collision monitoring,

Algorithm 1: Stopping Trajectory Generation

Input: Vehicle position $\mathbf{x}$ and sample set of points $\mathcal{S}$
for $s \in \mathcal{S}$ **do**
  Compute C_{escape};
S_{escape}:=StratifiedSample($\mathcal{S}$);
sort(escapePoints, C_{escape});
for $\mathbf{e}_i$ *in* S_{escape} **do**
  ξ:=SolvePolynomial($\mathbf{x}$, $\mathbf{e}_i$);
  if *feasible(ξ) and collisionFree(ξ)* **then**
    executeTrajectory(ξ);

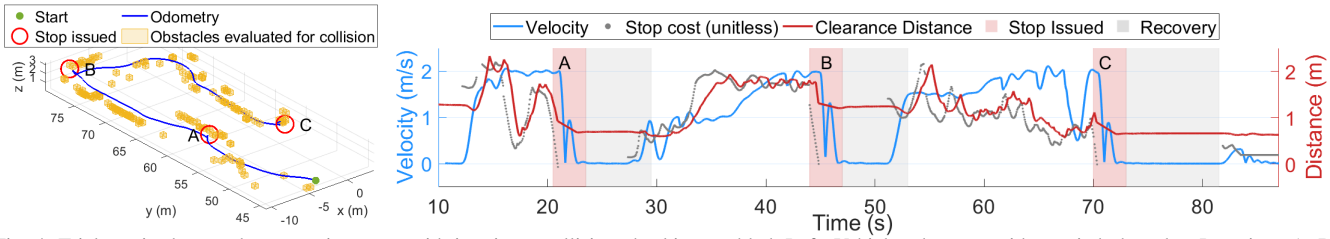

Fig. 4: Trial run in the warehouse environment with imminent collision checking enabled. Left: Vehicle odometry with queried obstacles. Locations A, B, C indicate where stopping trajectories were issued corresponding with the left plot. Right: Stopping trajectories are issued when stop cost drops below a threshold, ensuring that the vehicle is safe. The operator recovers the vehicle with an in-place yaw before normal flight is resumed.

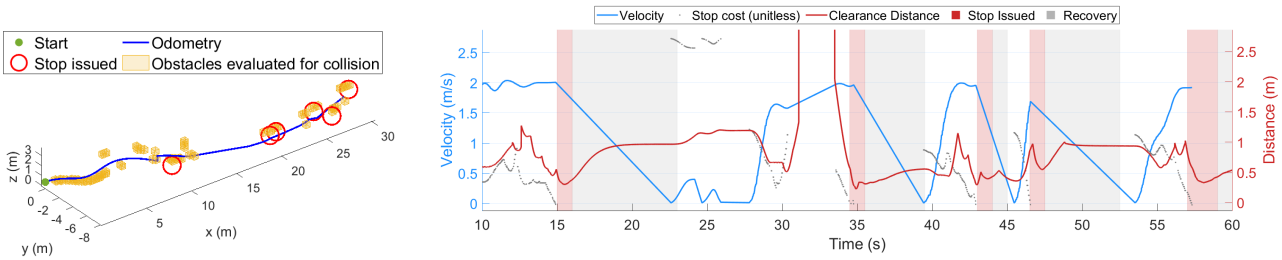

Fig. 5: Trial run in the random forest environment with imminent collision checking enabled. Left: Vehicle odometry with queried obstacles, locations in red indicate where stopped trajectories were issued. Left: Similarly to Figure 4, stopping trajectories are issued when the stop cost reaches a threshold, corresponding with high velocity and low clearance.

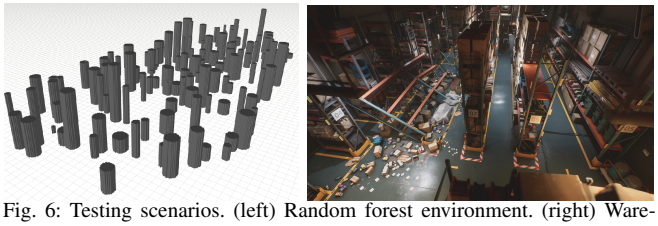

Fig. 6: Testing scenarios. (left) Random forest environment. (right) Warehouse environment.

TABLE I: Task completion with and without collision monitoring

Scenario	Success rate	Velocity (m/s)	Distance (m)	Min Distance (m)
Forest				
Enabled	90%	1.37 ± 0.18	1.61 ± 0.11	0.54 ± 0.13
Disabled	20%	1.69 ± 0.18	1.44 ± 0.22	0.39 ± 0.16
Warehouse				
Enabled	100%	1.73 ± 0.07	1.89 ± 0.11	0.47 ± 0.08
Disabled	20%	1.80 ± 0.15	1.82 ± 0.20	0.32 ± 0.13

TABLE II: Results for trials with collision detecting enabled

Environment	# Stops Issued	Velocity (m/s)	Obstacle Distance (m)
Forest	6 ± 1	1.84 ± 0.12	1.39 ± 0.09
Warehouse	4 ± 1	1.97 ± 0.04	1.37 ± 0.08

but the vehicle safety is maintained. "Distance" refers to the average distance from obstacles throughout the entire trial, while "Min Distance" is the minimum distance from an obstacle on a given trial.

Table II shows the average number of stops issued over the course of completing task with collision monitoring enabled. In the forest scenario, we prevent 6 potential collisions on average. In the warehouse scenario, we prevent 4 potential collisions. The table shows the vehicle velocity and obstacle distance conditions that trigger these stops.

For the final workshop presentation, we intend to have hardware results, where we evaluate similar tasks on a physical vehicle. The platform will use visual odometry for localization and a seperate depth camera to build a local-map.

Figure 7 shows the correlation between the obstacle offset angle and the distance to the obstacle when determining whether to stop. A linear boundary is formed between stop

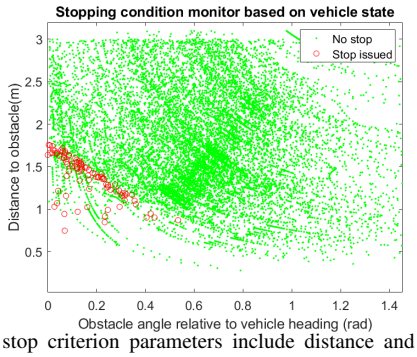

Fig. 7: The stop criterion parameters include distance and relative angle to obstacle. An approximately linear boundary is formed between stop and no-stop conditions. Red circles represent conditions for which stops were triggered.

and no-stop scenarios. Some no-stop points lie below the boundary due to the third velocity factor, because the vehicle does not stop when it near obstacles at low velocity.

IV. Conclusion and Future Work

In this work, we presented an imminent collision avoidance system for safe teleoperation in dense environments. This system revolves around three main ideas. First, a continuous monitor on the vehicle state and nearby obstacles, which will trigger a stop if conditions are met. Second, escape point sampling to select potential goal locations in free space for the vehicle to stop. Third, a polynomial trajectoy generator which iterates over escape points to select a feasibile and collision-free stopping trajectory. The system can be integrated in an existing teleoperation framework.

Future work can improve upon the stopping condition heuristic. This work used a KD-tree map representation, but could be expanded to use other map representations.

REFERENCES

[1] Andr Ferreira and Vassallo Raquel Frizera Bastos Filho Teodiano Freire Sarcinelli Filho Mrio Pereira, Flvio Garcia. An approach to avoid obstacles in mobile robot navigation: the tangential escape. *Sba: Controle Automao Sociedade Brasileira de Automatica"*. ISSN 0103-1759.

[2] X. Yang, J. Cheng, and N. Michael. An intention guided hierarchical framework for trajectory-based teleoperation of mobile robots. *2021 IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2021.