Combining 3D Planning and Control Barrier Functions for Safe Motion of Quadrotor UAVs among Obstacles

A. Cristofaro, M. Ferro, F. Galasso, M. Mizzoni, A. Pacciarelli, M. Vendittelli

Abstract— **This paper proposes the combination of a fast motion planner, which generates potentially unfeasible trajectories for the center of mass of an underactuated UAV, with a tracking controller integrating a Control Barrier Function. In this way avoidance of both stationary and moving obstacles is guaranteed even when the UAV deviates from the collision-free trajectory issued by the planner, due to its unfeasibility. Simulations in a dynamic and cluttered environment validate the method.**

I. Introduction

Multirotor Unmanned Aerial Vehicles (UAVs) represent today a valuable asset in many industrial and service tasks like inspection and maintenance of infrastructures, environment and crop monitoring, security surveillance. These vehicles are also progressively penetrating the leisure and entertainment market, with extensive use in fotography and filming, due to the limited risk that small-size UAVs represent for humans and goods safety, and to the limited cost of commercial platforms. UAVs can feature very different designs, sensory, actuation and computational equipment with resulting different performances in terms of maneuverability, payload carrying capability, autonomy of flight.

Small quadrotor UAVs, in particular, represent one of the most valid option for operation in cluttered environments because they can be small, light, and yet capable to carry small equipments like cameras, and cost effective. They can take off and land vertically, and hover at a desired high. Guaranteeing safety during autonomous flight in environments populated by humans and things presents, however, some challenges related to perception, actuation, control.

This paper considers the problem of guaranteeing the absence of collisions between a quadrotor UAV and the obstacles populating the environment within which it is executing a motion task. The problem can be attacked either by taking a purely reactive collision avoidance approach or by first planning a collision-free trajectory to be tracked. A reactive behaviour is appropriate when the environment is completely unknown. It could, however, result in a poorly efficient task execution. In addition, reactive methods, being local, typically suffer from convergence problems. When a map of the environment is known, partially known, or incrementally reconstructed, a motion planning method can provide some form of completeness property, optimized motion, avoidance of deadlocks typical of reactive approaches.

Collision-free motion planning for quadrotors presents, however, difficulties, due to their underactuated, second order dynamics, as discussed in Sect. II. The main approaches proposed in the literature use motion primitives [1], [2] or solve a QP problem to generate dynamically feasible trajectories [3], [4], in combination with a collision checker. While these, resolution complete, methods directly provide collision-free and dynamically feasible trajectories, they are either not suitable for real-time adaptation to uncertain environments or require highly accurate tracking of the planned trajectories to guarantee safety.

The method proposed in this paper combines a fast motion planner, providing collision-free but possibly dynamically unfeasible trajectories, with a safe tracking controller guaranteeing avoidance of collisions when the UAV deviates from the planned trajectory due to its dynamic unfeasibility. Safe tracking also allows for replanning at a lower rate enabling the application of the method to changing environments. In addition to being computationally light, the overall method is simple to implement and tune.

The design of the control is based on the notion of Control Barrier Functions (CBFs) [5]. CBF-based techniques for obstacle avoidance have been proposed in, among other references, [6], [7], [8], [9], [10]. In this context, the computation of the control inputs typically relies on a quadratic program to be solved online, with the aim of modifying the nominal commands when necessary for safety reasons. Similarly, in [11] we proposed an event-triggered control input, based on the evaluation of a control barrier function, to guarantee safety, without resorting to online optimization. The basic idea is to design an arbitrary trajectory tracking law to be filtered in close proximity of obstacles, so as to retain only the components of the tracking input that move the UAV away from the closest obstacle. The present work continues on the same path, by specifing and exploiting the application of the framework to UAVs flying in cluttered environments.

The contribution consists in a method that combines fast generation of a global collision-free motion plan with formal guarantee of safe execution, which is also suitable for real-time adaptation in changing environments. Safety guarantee, under the conditions provided in [11], is validated through simulations in highly cluttered static environments. Its performance in dynamically changing environments is preliminary analysed through simulations in view of future work aiming to determine the formal conditions for ensuring safety. It is also worth mentioning that the overall method can be used to generate collision-free dynamically feasible trajectories through integration of the closed loop dynamics.

The paper is organized as follows. Section II illustrates the problematics of motion planning for quadrotors UAVs and provides the dynamic model used for control design. Section III provides the mathematical details of the safe tracking controller. Simulation results are reported in Sect. IV.

Authors are with Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti", Sapienza Università di Roma, Italia.

Authors' names are listed in alphabetical order. Corresponding author: M. Vendittelli, marilena.vendittelli@uniroma1.it

Conclusion and future perspectives are given in Sect. V.

II. Problem formulation

In quasi-hovering flight conditions, considering negligible aerodynamic and gyroscopic effects, the dynamic model of the quadrotor sketched in Fig. 1 can be expressed as [12]:

$$\begin{aligned}
\ddot{x} &= -\frac{T}{m}(\cos(\varphi)\sin(\vartheta)\cos(\psi)+\sin(\varphi)\sin(\psi))\\
\ddot{y} &= -\frac{T}{m}(\cos(\varphi)\sin(\vartheta)\sin(\psi)-\sin(\varphi)\cos(\psi))\\
\ddot{z} &= g-\frac{T}{m}\cos(\varphi)\cos(\vartheta)\\
\ddot{\varphi} &= \frac{\tau_\varphi}{I_x}; \qquad \ddot{\vartheta} = \frac{\tau_\vartheta}{I_y}; \qquad \ddot{\psi} = \frac{\tau_\psi}{I_z}
\end{aligned} \tag{1}$$

where $q = (x, y, z, \varphi, \vartheta, \psi)^T \in \mathbb{R}^6$ represents the configuration of the UAV consisting in the Cartesian position (x, y, z) of its center of mass, expressed in the inertial frame SR_I, and the orientation, parameterized through the (RPY) Euler angles $(\varphi, \vartheta, \psi)$, of the body frame SR_B with respect to the inertial frame SR_I. The control input $u = (T, \tau_\varphi, \tau_\vartheta, \tau_\psi)^T$ is given by the collective thrust intensity T and the torques around the body axes frame generated by the four rotors. The UAV mass is m, g is the value of gravity acceleration, $\boldsymbol{I}_B = \mathrm{diag}\{I_x\, I_y\, I_z\} \in \mathbb{R}^{3\times3}$ is the inertia matrix, assumed to be diagonal due to the symmetric shape of the vehicle.

A. *Motion planning for quadrotor UAVs*

The motion planning problem encompassed by the well-known *piano movers' problem* consists in finding a collision-free path from a starting configuration $\boldsymbol{q}_s$ to a goal configuration $\boldsymbol{q}_g$, i.e., a sequence of configurations defined by the mapping $\sigma \mapsto \boldsymbol{q}(\sigma)$, with the path parameter $\sigma \in [0, 1]$ such that $\boldsymbol{q}(0) = \boldsymbol{q}_s$ and $\boldsymbol{q}(1) = \boldsymbol{q}_g$, and each configuration in $\boldsymbol{q}(\sigma)$ does not collides with any obstacle in the environment. The path with an associated timing law provides a reference trajectory to be tracked through feedback control.

The problem is usually solved in the configuration space [13] of the robot, the UAV in our case, using geometric or randomized approaches [14]. The solution for quadrotors presents some difficulties due to their underactuated, second order dynamics. Underactuation, i.e., number of control inputs smaller than the number of the degrees of freedom, has the consequence that not all the paths in the configuration space are feasible for the vehicle, as illustrated in Fig. 2, while the second order dynamics makes it impossible to exactly track nonsmooth trajectories, as illustrated in Fig. 3. The first difficulty can be approached by solving the planning problem for a spherical hull bounding the vehicle's frame. Although this may cause loss of the completeness property, since motion through narrow passages requiring special orientation of the UAV body are not included in the solution, it still represents a reasonable practical approach to the problem solution, especially in moderately cluttered environments and small size UAVs. The second difficulty remains even if the above mentioned conservative approach is adopted in the planning phase. In fact, at corner points of the collision-free planned path, where the path tangent is discontinuous, the quadrotor will deviate from it and eventually hit some obstacle. This risk could be mitigated by an appropriate choice of the controller parameters requiring fine tuning, a priori complete knowledge of the environment and dynamic adaptation of the parameters, with a resulting difficulty in proving any formal property of safety.

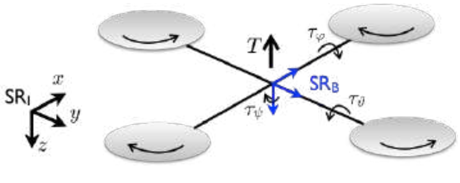


Fig. 1: Reference frames of interest and control inputs.

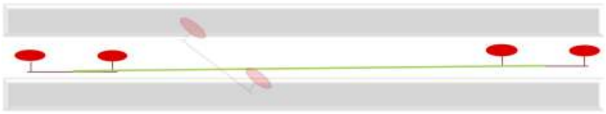


Fig. 2: Due to underactuation, the collision-free path requiring the quadrotor to move with a fixed orientation along the narrow passage is not feasible: frame bending is required to obtain a non-null component of the thrust along the planned path (transparent UAV).

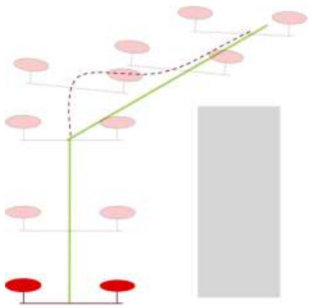


Fig. 3: Exact tracking of paths with discontinous tangent is not possible, due to the quadrotor second order dynamics. During Tracking, the UAV deviates from the planned collision-free path at points where it does not have the required smoothness properties.

As an alternative, collision-free paths with 4th order continuous derivatives should be searched in the space of the flat outputs. Although theoretically providing completeness and dynamic feasibility of the trajectories, it is not a computationally efficient approach. In the literature it often results in the exploration of the space of feasible trajectories using motion primitives [1], [2] or in the solution of a QP problem for the generation of dynamically feasible trajectories [3], [4]. To our best knowledge, a method allowing for real-time (re)planning which is robust to environment and model uncertainties is missing in current literature.

In this paper we take a practical approach by using an off-the-shelf randomized planner, that considers the sphere of minimum radius containing the quadrotor for collision checking, to generate collision-free trajectories for the UAV center of mass in the Cartesian space. These trajectories are potentially unfeasible and the UAV can deviate from them during tracking. To avoid collisions, we use the CBF-based tracking controller proposed in [11] which guarantees invariance of a properly defined safe set during tracking.

B. Nominal tracking controller

In this section we describe the controller adopted to track a Cartesian trajectory, without taking into account the presence of obstacles. This controller, hereafter referred to as *nominal*, will be modified in the next section to enforce collision avoidance during tracking. It exploits the hierarchical dynamics of the quadrotor by devising a Cartesian tracking control that generates reference angles for the attitude controller [15], [16]. Specifically, consider the desired Cartesian trajectory $p_d = (x_d, y_d, z_d)^T$ issued by the motion planner and a PID tracking controller

$$u^\star = \ddot{p}_d + k_d(\dot{p}_d - \dot{p}) + k_p(p_d - p), \tag{2}$$

with $k_d > 0$ and $k_p > 0$ gain matrices. Then, rewrite the translational dynamics of the quadrotor, described by the first three equation in (1), as

$$\begin{aligned} \ddot{x} &= -\frac{T}{m}T_x; \qquad \ddot{y} = -\frac{T}{m}T_y \\ \ddot{z} &= g - \frac{T}{m}\cos(\varphi)\cos(\vartheta), \end{aligned} \tag{3}$$

where T_x and T_y represent the direction components of the thrust vector in the inertial frame. These components are determined by the UAV attitude, but in accordance with the hierachical control approach we consider them as control inputs that can be used to stabilize the translational dynamics. In particular, the choice

$$\begin{aligned} T_x &= -\frac{m}{T}u_x^\star; \qquad T_y = -\frac{m}{T}u_y^\star \\ T &= \frac{m}{\cos(\varphi)\cos(\vartheta)}(g - u_z^\star), \end{aligned} \tag{4}$$

with $u_x^\star$, $u_y^\star$, and $u_z^\star$ the components of the PID control vector (2), asymptotically stabilize the translational dynamics (3). Exploiting then the attitude dynamics, the reference

$$\begin{aligned} \vartheta_d &= \arctan\left(\frac{u_x^\star\cos\psi + u_y^\star\sin\psi}{u_z^\star - g}\right) \\ \varphi_d &= \arctan\left(\cos\vartheta_d\frac{u_x^\star\sin\psi - u_y^\star\cos\psi}{u_z^\star - g}\right) \end{aligned} \tag{5}$$

is given to the low level controller to track the desired control inputs T_x and T_y.

Neglecting the dynamics of the attitude controller, equations (3) and (4) allow to consider the quadrotor dynamics as a double integrator to which we can apply the results in [11]. The controller (2) is therefore enhanced by the combination with a control barrier function to guarantee the avoidance of collisions even when the UAV deviates from the planned trajectory, as illustrated in the next section.

III. Proposed approach

Consider the cartesian unconstrained dynamics of the vehicle described by the double integrators

$$\ddot{p}(t) = u(t), \quad p \in \mathbb{R}^3, \quad u \in \mathbb{R}^3. \tag{6}$$

To track a pre-planned output trajectory $p_d(t)$, which may possibly be not feasible, while guaranteeing to stay clear from obstacles even when moving away from $p_d(t)$, the paper proposes a design rationale for the control law u inspired by the Control Barrier Functions architecture [5], [11]. To this end, let us consider the function

$$h(p(t), \dot{p}(t)) = (p(t) - \bar{p})^T(p(t) - \bar{p}) + \mu(p(t) - \bar{p})^T\dot{p}(t), \tag{7}$$

where $\bar{p}$ denotes the position of a point obstacle and μ is a positive constant. Equation (7) corresponds to the squared distance of body center of mass position p to the obstacle position $\bar{p}$ plus a term with sign depending on the cross product between the relative position of the vehicle with respect to the obstacle and the velocity $\dot{p}$ of the system. Defining $\bar{h}(p(t), \dot{p}(t)) := h(p(t), \dot{p}(t)) - \delta$, with $\delta > 0$ a given constant, the *safety region* is given by

$$\mathcal{S}_{\text{free}} := \{(p, \dot{p}) \in \mathbb{R}^{2n} : \bar{h}(p(t), \dot{p}(t)) \geq 0\}.$$

In particular due to the structure of (7), if the velocity points away from the obstacle, the cross-product temporarily decreases the required clearance. Conversely, when the velocity points towards the obstacle, the cross-product becomes negative increasing the clearance. To guarantee invariance of the safety region, we look for control inputs such that $\dot{h}(t) \geq 0$. Evaluating the derivative $\dot{h}(t)$ yields the inequality

$$2(p - \bar{p})^T\dot{p} + \mu(p - \bar{p})^T u \geq 0, \tag{8}$$

which corresponds to the constraint to be enforced on the control input. Assume that the reference trajectory $p_d(t)$ is a piecewise smooth curve, with a countable number of non-smooth, isolated, points, i.e., there exists an increasing sequence of time steps $0 = t_0 < t_1 < \cdots < t_k < \cdots$ with

$$\begin{bmatrix} p_d(t) \\ \dot{p}_d(t) \\ \ddot{p}_d(t) \end{bmatrix} = \begin{cases} p_d^{(j)}(t) & t \in [t_{j-1}, t_j) \\ \dot{p}_d^{(j)}(t) & t \in [t_{j-1}, t_j) \\ \ddot{p}_d^{(j)}(t) & t \in [t_{j-1}, t_j) \end{cases} \tag{9}$$

where $(p_d^{(j)}, \dot{p}_d^{(j)}, \ddot{p}_d^{(j)})$ are smooth curves such that $\lim_{t \to t_j^-} p_d^{(j)}(t) = p_d^{(j+1)}(t_j)$ holds true for any $j = 1, 2, ...,$ this being a necessary continuity condition on the position. Accordingly, consider the nominal tracking controller (2). Bearing the constraint (8) in mind, we can decompose the state space $\mathbb{R}^6 = \mathcal{D}_{\text{track}}(t) \cup \mathcal{D}_{\perp}(t)$ where $\mathcal{D}_{\text{track}}(t) = \mathcal{D}_{u^\star}(t) \cup \mathcal{D}_{\delta_1}$ is the subset where the control priority is the asymptotic tracking of the reference. In particular, $\mathcal{D}_{u^\star}(t) := \{(p, \dot{p}) : (p - \bar{p})^T(\mu u^\star + 2\dot{p}) > 0\}$ is the region where the forward invariance of $\mathcal{S}_{\text{free}}$ during tracking is preserved because the nominal controller fulfills (8), while $\mathcal{D}_{\delta_1} := \{(p, \dot{p}) : h(p, \dot{p}) > \delta_1 > \delta\}$ is a *conservative* safety region with increased clearance. Conversely, the control priority in the complementary region $\mathcal{D}_{\perp}(t) = \mathbb{R}^{2n} \setminus \mathcal{D}_{\text{track}}(t)$ is to avoid obstacles, which are close enough to constitute an actual threat. Define the projection operators

$$\Pi_{p-\bar{p}} := (p - \bar{p})[(p - \bar{p})^T(p - \bar{p})]^{-1}(p - \bar{p})^T \tag{10}$$

$$\Pi_{p-\bar{p}}^{\perp} := I - \Pi_{p-\bar{p}}. \tag{11}$$

The operator defined by (11) retains the component of w which is orthogonal to $(p - \bar{p})$ and safety during tracking is then enforced by the event-triggered control

$$u = \begin{cases} u^\star & \text{if } (p,\dot{p}) \in \mathcal{D}_{\text{track}} \\ -\frac{2}{\mu}\Pi_{p-\bar{p}}\,\dot{p} + \Pi^{\perp}_{p-\bar{p}}u^\star & \text{if } (p,\dot{p}) \in \mathcal{D}_{\perp}. \end{cases} \tag{12}$$

In fact, whenever $(p,\dot{p}) \in \mathcal{D}_{\perp}$, the evaluation of $\dot{\bar{h}}$ yields

$$\begin{aligned} \dot{\bar{h}} &= 2(p-\bar{p})^T\dot{p} + \mu\dot{p}^T\dot{p} \\ &\quad +\mu(p-\bar{p})^T(-\tfrac{2}{\mu}\Pi_{p-\bar{p}}\,\dot{p} + \Pi^{\perp}_{p-\bar{p}}u^\star) = \mu\dot{p}^T\dot{p} \geq 0, \end{aligned}$$

thus showing that the control action prevents the state of the system from approaching further the obstacle. The control policy (12) essentially overrides, when needed, the PID tracking controller $u^\star$ by pruning the components of the acceleration that might drive the robot towards the obstacle. A stability proof for this control scheme is provided in [11].

A. Multiple obstacles

The previous developments can be extended to scenarios with multiple obstacles. Indeed, consider a finite number of obstacles $\mathcal{O} = \{\bar{p}_i \in \mathbb{R}^n, i = 1, ..., m\}$ and assume that no overlap occurs for the basins of influence of different obstacles:

$$\|\bar{p}_i - \bar{p}_j\|^2 > 2\delta_1 \text{ for any } i, j = 1, ..., m, \tag{13}$$

We define then the family of barrier functions

$$\bar{h}_i(p,\dot{p}) = (p-\bar{p}_i)^T(p-\bar{p}_i) + \mu(p-\bar{p}_i)^T\dot{p} - \delta,\ i = 1, ..., m \tag{14}$$

and we compute the label of the closest obstacle to the current position, i.e., $i^* = \arg\min_{i=1,...,m} \|p - \bar{p}_i\|$, Accordingly, we consider *active* only the barrier function $\bar{h}_{i^*}$.This information can be provided by exteroceptive sensors such as proximity sensors or cameras mounted either on the robot itself or in the surrounding environment. The non-overlapping assumption (13), allows to define the control u as in (12) with sets $\mathcal{D}_{\text{track},i^*}$ and $\mathcal{D}_{\perp,i^*}$ depending on the *active* obstacle $\mathcal{O}_{i^*}$.

IV. Simulation results

This section reports the simulation results obtained by applying the proposed method to UAV navigation in different environments (Fig 4). The trade-offs of the control parameters are also analysed in different scenarios.

The UAV has dimensions $\{0.35, 0.30, 0.98\}$[m], with a mass $m = 0.5$ [Kg] and Inertia $\boldsymbol{I} = \{I_x, I_y, I_z\} = \{0.0019, 0.0019, 0.0033\}$ $[\mathrm{m}^2 \cdot \mathrm{Kg}]$. The controller (12) has been implemented in MATLAB, while the quadrotor dynamics and its interaction with the environment is simulated in CoppeliaSim. The distance of the UAV to the closest obstacle is provided by the simulator. In a real scenario this information can be provided by a proximity sensor.

The adopted motion planner comes from the OMPL (Open Motion Planning Library) software library, providing the implementations of various motion planning algorithms. Collision checking considers the bounding sphere around the UAV center of mass. The average execution time of the planner for the considered scenes is 10 ms. Detailed comments follow, and a video of the simulations is available at: `https://youtu.be/tvaOA31ubxk`.

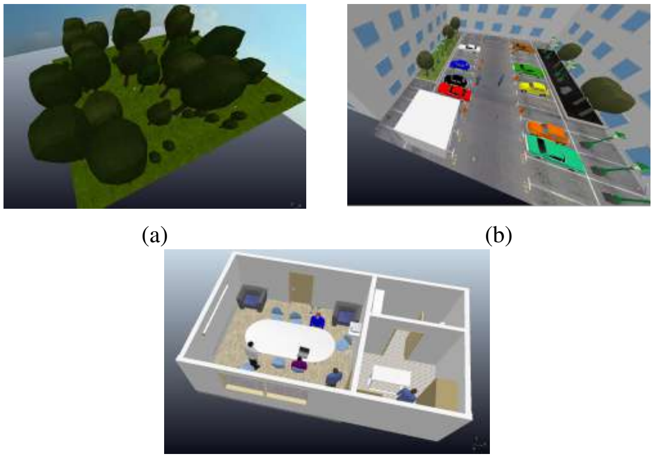


Fig. 4: Environments used in the simulations. An outdoor scenario with static obstacles (a); an outdoor scenario with moving obstacles (b); an indoor highly cluttered scenario with static obstacles (c).

A. Outdoor environment with static obstacles

In this set of simulations, a quadrotor has to safely track a non-feasible collision-free trajectory in a cluttered scene populated with static obstacles (Fig 4a), at a constant speed of $0.3[\mathrm{m/s}]$. The trajectory has been safely tracked adopting the controller (12) with the barrier function (7) using the parameters $\delta_1 = 0.08$, $\delta = 0.05$, $\mu = 0.9$ and $\{k_p, k_d\} = \{0.32, 0.50\}$. During transient, at points with discontinous path tangent, the event-triggered controller filters the nominal inputs to keep its components orthogonal to the obstacle direction and adds component pushing the UAV away from the obstacle (Fig. 5, first row).

Without the CBF, using the same PD controller parameters, the robot collides with the obstacles during transients, i.e., where the trajectory is unfeasible, as shown in Figures 5d-5e-5f. Figure 6 reports an overview of the complete motion (in green the planned path, in blue the one actually followed by the UAV) and the value of the distance to the closest obstacle for the planned and the actual trajectory. The blue curve refers to the distance along the actual trajectory, which is kept always above the safety bound δ.

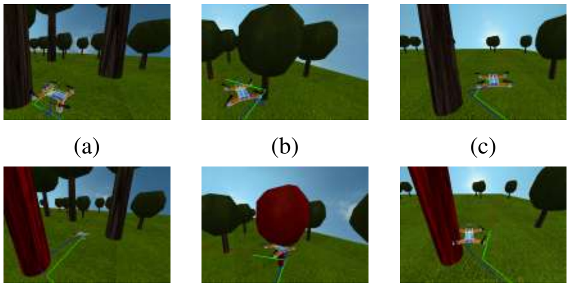
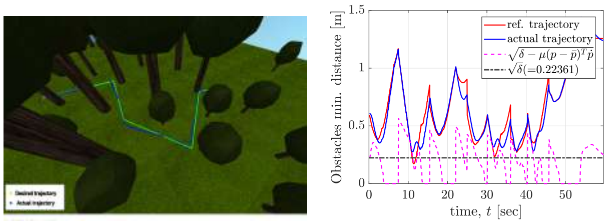


Fig. 5: Planned (green) and actual (blue) path during motion in a cluttered environment. Top: controller with barrier functions; Bottom: unfiltered PD tracking controller controller.



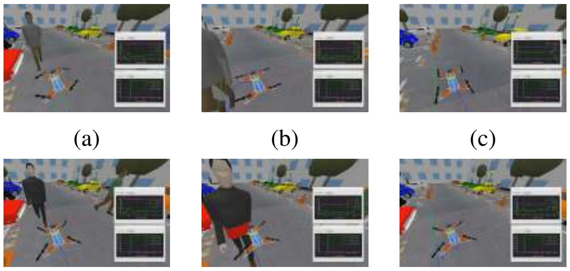
(a) (b)

Fig. 6: Safe motion in a cluttered environment, with the controller (12). (a) shows the final path; (b) shows the evolution in time of distance from the closest obstacle in respect to the actual and the desired trajectory.

B. Outdoor environment with dynamic obstacles

In this set of simulations we consider a wide environment (Fig 4b) with a moving obstacle, through which the quadrotor has to safely track an unfeasible trajectory with the controller (12) using the parameters $\delta_1 = \{0.4, 0.6\}$, $\delta = 0.05$, $\mu = \{0.5, 0.9\}$ and $\{k_p, k_d\} = \{0.32, 0.50\}$. The aim is to analyse the limit of the relative veloctiy between the UAV and the human, once the parameters have been fixed. In the first simulation (Fig 7) we let the obstacle (human) have a velocity lower (in norm) than that of the quadrotor. The proposed controller managed to handle such unexpected event and still generated a safe obstacle avoidance maneuver shown in the first row of Fig. 7. In subsequent simulations the velocity of the obstacle was progressively increased, keeping the parameters and UAV velocity as in the first simulation. As expected, performances gradually degraded, until the UAV collided with the human. The second row of Fig. 7 shows such a behaviour.

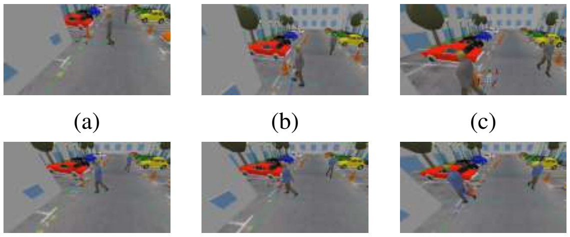
In the second simulation (Fig 8), we put the human on a trajectory orthogonal to the quadrotor path. In that case, we experienced that the orthogonal component of the velocity along the relative position of the two bodies was almost equal to 0. Hence, the maximum the quadrotor could do was to retreat from the obstacle and wait for it to clear the path.



(a) (b) (c)

(d) (e) (f)

Fig. 7: First set of simulations with dynamic obstacles. First row: human moves at a constant speed of 0.2 $[\mathrm{m/s}]$ and the controller has $\mu = 0.5$; Second row: human moves at a constant speed of 0.3 $[\mathrm{m/s}]$ and the controller has $\mu = 0.9$;



(a) (b) (c)

(d) (e) (f)

Fig. 8: Second set of simulations with dynamic obstacles. First row: human moves at a constant speed of 0.2 $[\mathrm{m/s}]$ and the controller has $\mu = 0.9$; Second row: human moves at a constant speed of 0.3 $[\mathrm{m/s}]$ and the controller has $\mu = 0.9$;

C. Critical situation: office

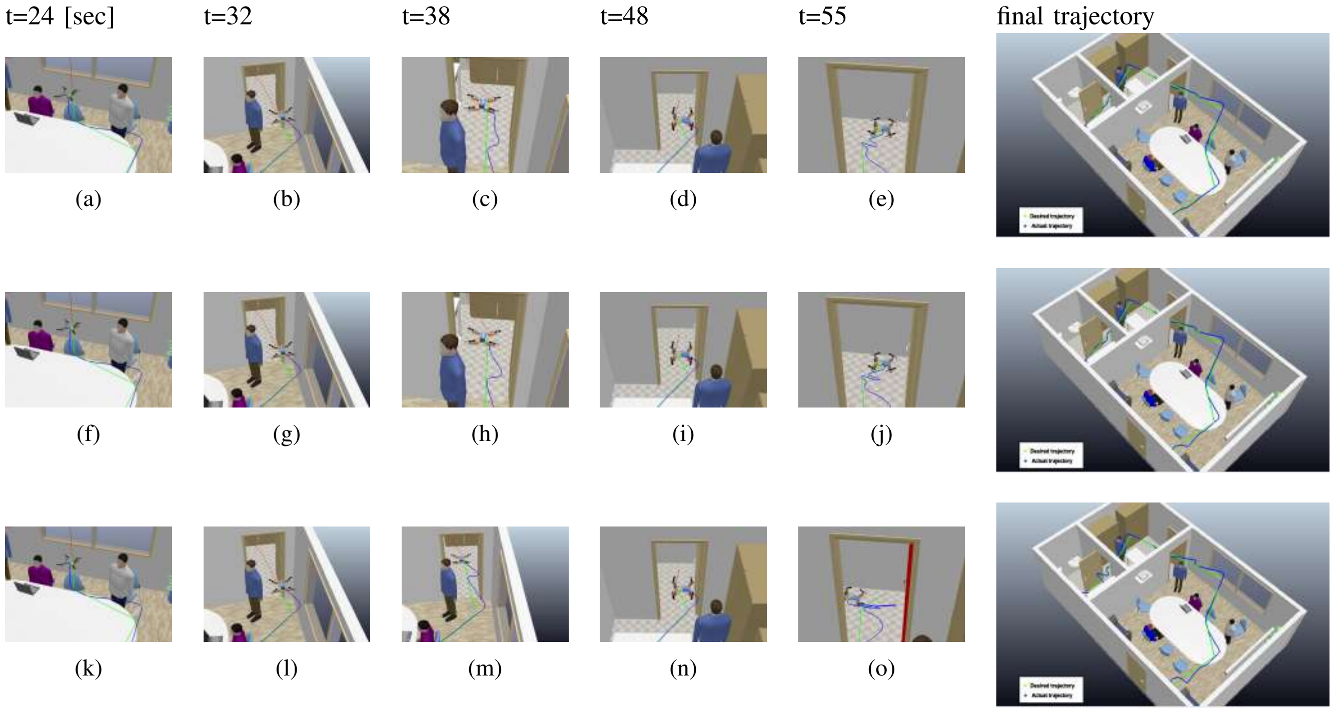
In this set of simulations, we consider an extremely cluttered enviroment with static objects (Fig 4c), through which the quadrotor has to safely track an unfeasible collision-free trajectory, with a desired constant speed of 0.25 $[\mathrm{m/s}]$. The main focus is to move in a crowded scenario in order to empirically analize the effect of variations in the value of μ and δ_1 parameters of the controller (12), to find out the best trade-off. In order to have a fair comparison, for all the runs of the simulations, the parameters considered by the nominal controller are $\{k_p, k_d\} = \{0.5, 0.8\}$ and $\delta = 0.05$.

In the first set of simulations, we considered different values for $\mu \in [0.5, 1.4]$, while keeping $\delta_1 = 0.1$. The trade-off to investigate is based on the fact that while a small μ would be preferable because it produces a more reactive behaviour, it can also induce ascillatory motion when reduced below a given threshold. Initially we have considered $\mu = 1.4$ and the resulting behavior is stable, not leading to collisions during motion (first row of Fig. 9). On the other hand, as expected, the results show that for smaller values of μ, the system has a increasingly less stable behaviour in the proximity of the obstacles. In particular, when the system is at risk of collision, the controller induces a strong overshoot in the opposite direction to the obstacle, caused by the negative term in (12) dominating the control input. Consequently, this leads to an high positional error which in the next timestep induces an overshoot in the opposite direction (second row of Fig. 9).

In the second set of simulations, we kept $\mu = 1.4$ while considering different values for $\delta_1 \in [0.051, 0.3]$. The results show that, during the transients, for values of δ_1 close to the squared radius of the sphere bounding the UAV (i.e. δ), obstacles are detected late and collisions may occur due to the approximation of the UAV dynamics to that of a double integrator, i.e., the attitude dynamics introduces a delay in the reaction that could be critical in higlhy cluttered environments. Conversely, for high values of δ_1 (i.e. 0.3) the controller keeps a large distance from the obstacle, worsening the tracking of the desired trajectory and eventually hit obstacles in narrow passages. The last row of Fig. 9 shows a combination of critical values for these parameters. Clearly, to deal with narrow passages it is necessary to take into account the attitude dynamics by resorting to higher order models of the UAV.

V. Conclusion

We have presented a method allowing collision-free motion of quadrotor UAVs in environments cluttered by ob-



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

Fig. 9: Safe motion in cluttered environment. First row: $\mu = 1.4$ and $\delta_1 = 0.1$; Second row: $\mu = 1.0$ and $\delta_1 = 0.1$; Third row: $\mu = 0.5$ and $\delta_1 = 0.3$. The system presents an increasingly oscillatory behaviour that, due to the approximations in the UAV dynamics, increases the risk of collisions in correspondence of narrow passages, as in (o).

stacles. The methods makes use of a collision-free motion planner in the Cartesian space that generates non-smooth reference trajectories for the UAV center of mass. At points where the trajectories do not have the appropriate smoothness properties, the quadrotor deviates from the reference. Potential collisions with surrounding obstacles are however avoided by virtue of a CBF-based tracking controller enforcing invariance of a properly defined safe set.

With respect to existing solutions for navigation of quadrotors among obstacles, the proposed one combines the advantages of a simple global planning method with the enforcement of safety through control that locally allows to avoid collisions where exact tracking is not possible.

Future work includes: use available information on the velocity of the moving obstacles (measured or estimated); refining the bounds on control parameters to take into account the attitude controller dynamics; devise a control-based efficient method to generate collision-free and dynamically feasible trajectories relying on the results of this work; apply the results in [11] for higher order dynamics to the quadrotor model linearized through dynamic feedback.

References

[1] D. Brescianini and R. D'Andrea, "Computationally efficient trajectory generation for fully actuated multirotor vehicles," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 555–571, 2018.

[2] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2872–2879.

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

[4] B. J. Santiago Sanchez Escalonilla Plaza*, Rodolfo Reyes-Báez, "Online whole-body motion planning for quadrotor using multi-resolution search," in *2023 IEEE Int. Conference on Robotics and Automation*.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[6] L. Doeser, P. Nilsson, A. D. Ames, and R. M. Murray, "Invariant sets for integrators and quadrotor obstacle avoidance," in *2020 American Control Conference (ACC)*, 2020, pp. 3814–3821.

[7] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *2017 IEEE Int. Conference on Robotics and Autom.*, 2017, pp. 3293–3298.

[8] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[9] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 322–328.

[10] A. Suresh and S. Martínez, "Risk-perception-aware control design under dynamic spatial risks," *IEEE Control Systems Letters*, vol. 6, pp. 1802–1807, 2021.

[11] A. Cristofaro, M. Ferro, and M. Vendittelli, "Safe trajectory tracking using closed-form controllers based on control barrier functions," in *2022 IEEE Conf. on Decision and Control*, 2022, pp. 3329–3334.

[12] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an x4-flyer." *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 217–222, 2002.

[13] Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computers*, vol. C-32, no. 2, pp. 108–120, 1983.

[14] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[15] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[16] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated vtolvehicles: A review of basic control design ideas and principles," *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 61–75, 2013.