Robust Collision Avoidance for Mobile Robots in the Presence of Moving Obstacles

Jorge A. Ricardo Jr. and Davi A. Santos

***Abstract*—This letter is concerned with the collision avoidance for mobile robots with uncertain dynamics in the presence of obstacles that can considerably change their velocities over time. To address this problem, we propose a robust collision-avoidance method based on the continuous-control-obstacles one. The proposed method uses an arbitrary-order overdamped low-pass filter to generate sufficiently smooth position commands for the robot and a high-order sliding mode differentiator to robustly estimate the obstacles' maximum accelerations. Based on these estimates, we define a set of possible future positions for the obstacles according to how each one is changing its velocity to calculate a robust position command for the robot. The method has been numerically evaluated using a conventional quadcopter flying among moving obstacles and has been shown to be effective in providing collision avoidance and velocity constraints satisfaction.**

***Index Terms*—Sliding mode, collision avoidance, velocity obstacles, mobile robots.**

I. Introduction

COLLISION avoidance is a fundamental problem of autonomous mobile robotic systems. Its task is to select appropriate control commands for the robot to complete a given mission without colliding. However, this turns out to be a difficult problem since these vehicles are generally subject to model uncertainties and disturbances, and may execute these missions in scenarios involving obstacles with unknown trajectories. Different methods have been used to address the collision avoidance for mobile robots, such as model predictive control [1], sampling-based search [2], reinforcement learning [3], and velocity obstacles (VO) [4]. Nevertheless, the last one, in particular, has a relatively low computational burden and has shown to be effective in avoiding collisions in very cluttered environments.

The VO's basic idea is to define a set of velocity vectors for the robot that could result in a collision if both the mobile robot and the obstacles kept constant velocity. Then, by continuously selecting a velocity outside this set, it is possible to guarantee collision avoidance under the assumption that this velocity can be instantaneously achieved, which only makes the VO method applicable to mobile robots with the linear equation of motion $\dot{\mathbf{p}} = \mathbf{v}$, where $\mathbf{p}$ is the position and $\mathbf{v}$ is the velocity. To overcome this drawback, different approaches were developed to address more complicated dynamics. The acceleration velocity obstacles (AVO) [5] were defined for mobile robots with dynamics described by $\dot{\mathbf{p}} = \mathbf{v}$ and $\dot{\mathbf{v}} = k(\mathbf{v}^* - \mathbf{v})$, where k is a gain and $\mathbf{v}^*$ is a target velocity. The AVO method was generalized to the so-called continuous control obstacles (CCO) [6] in order to be suitable for mobile robots described by an arbitrary-degree integrator. Bareiss and Van Den Berg [7] have proposed a new control obstacles method that, on the one hand, unifies the VO, the AVO, and the CCO methods, and, on the other hand, is applicable to mobile robots with nonlinear dynamics. However, these methods do not ensure collision avoidance for mobile robots with uncertain dynamics and generally predict the obstacles' future positions using a first-order model approximation, *i.e.*, assuming that they keep a constant velocity. Such an assumption is normally not satisfied in practice, but is addressed by using a quick replanning loop without the assurance that collision will be avoided when the obstacles accelerate.

Trying to fill the aforementioned gaps in the state of the art, this letter addresses the problem of collision avoidance for mobile robots with uncertain dynamics in the presence of obstacles that can considerably change their velocities. We propose a robust collision-avoidance method based on the CCO. The proposed method uses an arbitrary-order overdamped low-pass filter (LPF) to generate sufficiently smooth position commands for the robot controller. Then, by properly restricting the admissible sets according to the bounds of the robot's position and velocity tracking errors relative to the commanded position trajectory, the proposed method, differently from Van Den Berg et al. [5], Rufli et al. [6], and Bareiss and Van Den Berg [7], is able to guarantee collision avoidance and linear velocity constraints satisfaction for mobile robots with uncertain dynamics. To address the problem of collision avoidance in the presence of obstacles that can considerably

Manuscript received 2 March 2023; revised 22 April 2023; accepted 8 May 2023. Date of publication 11 May 2023; date of current version 22 May 2023. This work was supported in part by the São Paulo Research Foundation (FAPESP) under Grant 2019/053340; in part by the Coordination of Superior Level Staff Improvement (CAPES), EMBRAER S.A., and the Aeronautics Institute of Technology (ITA) under the Doctorate Scholarship through the Academic–Industrial Graduate Program (DAI); in part by the National Council for Scientific and Technological Development (CNPq) under Grant 304300/2021-7; and in part by the Funding Authority for Studies and Projects (FINEP) under Grant 01.22.0069.00. Recommended by Senior Editor C. Manzie. *(Corresponding author: Jorge A. Ricardo Jr.)*

The authors are with the Division of Mechanical Engineering, Department of Mechatronics, Instituto Tecnológico de Aeronáutica, São Paulo 12228900, Brazil (e-mail: jrjunior@poli.ufrj.br; davists@ita.br).

Digital Object Identifier 10.1109/LCSYS.2023.3275498

change their velocities, we employ a high-order sliding mode differentiator (SMD) [8], [9] to robustly estimate the maximum acceleration of each obstacle. Then, based on these estimates, instead of approximating the obstacles' future positions using first-order models [6], we define a set of possible future positions taking into account the obstacles' maximum accelerations. In summary, the main contribution of this letter is the proposal of a novel robust collision-avoidance method, based on the CCO, for mobile robots with uncertain dynamics in the presence of obstacles that can considerably change their velocities over time.

The remaining text is organized as follows. Section I-A presents the notation. Section II defines the problem. Section III formulates the collision-avoidance method. Section IV evaluates the proposed method using computer simulations. Finally, Section V concludes this letter.

A. Notation

The sets of real numbers, positive real numbers, and non-negative real numbers are denoted, respectively, by $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$. Similarly, the sets of integer numbers, positive integer numbers, and non-negative integer numbers are denoted, respectively, by $\mathbb{Z}$, $\mathbb{Z}_{>0}$, and $\mathbb{Z}_{\geq 0}$. Matrices and algebraic vectors are denoted, respectively, by uppercase and lowercase boldface letters, while geometric (Euclidian) vectors are denoted as in $\vec{a}$. The $n \times n$ identity matrix and the $n \times m$ zero matrix are denoted, respectively, by $\mathbf{I}_n$ and $\mathbf{0}_{n\times m}$. Moreover, the n-dimensional vectors of ones and zeros are denoted, respectively, by $\mathbf{1}_n$ and $\mathbf{0}_n$. Consider two arbitrary algebraic vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$. The vector inequality $\mathbf{x} < \mathbf{y}$ means that $x_i < y_i$, $\forall i \in \{1, \ldots, n\}$. The kth-order time derivative of $\mathbf{x}$ is denoted by $\mathbf{x}^{(k)}$. Furthermore, we define the vector signum function as $\operatorname{sign}(\mathbf{x}) \triangleq (\operatorname{sign}(x_1), \ldots, \operatorname{sign}(x_n))$, where

$$\operatorname{sign}(x_i) \triangleq \begin{cases} 1 & x_i > 0, \\ 0 & x_i = 0, \\ -1 & x_i < 0, \end{cases}$$

with $i \in \{1, \ldots, n\}$. The Euclidean norm and component-wise absolute value of $\mathbf{x}$ are denoted, respectively, by $\|\mathbf{x}\| \triangleq \sqrt{x_1^2 + \cdots + x_n^2}$ and $|\mathbf{x}| \triangleq (|x_1|, \ldots, |x_n|)$. A closed m-dimensional ball of radius $\rho \in \mathbb{R}_{>0}$ centered at $\mathbf{p} \in \mathbb{R}^m$, the Minkowski sum of two sets, and the set subtraction are denoted, respectively, by

$$\mathcal{B}(\mathbf{p}, \rho) = \left\{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x} - \mathbf{p}\| \leq \rho\right\},$$
$$\mathcal{X} \oplus \mathcal{Y} = \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in \mathcal{X},\ \mathbf{y} \in \mathcal{Y}\},$$
$$\mathcal{X} \setminus \mathcal{Y} = \{\mathbf{x} \mid \mathbf{x} \in \mathcal{X},\ \mathbf{x} \notin \mathcal{Y}\}.$$

II. Problem Statement

Consider an inertial Cartesian coordinate system (CCS) $\mathcal{S}_r$, fixed on the ground at a known point R, and a point A located at the robot's center of mass. Denote the position and velocity of A relative to $\mathcal{S}_r$, respectively, by $\mathbf{r}_\mathrm{A} \in \mathbb{R}^n$ and $\mathbf{v}_\mathrm{A} \in \mathbb{R}^n$, with $n \in \{2, 3\}$ being the dimension of the position space.

Let us denote the command of a given variable using an overbar symbol; *e.g.*, $\bar{\mathbf{r}}_\mathrm{A}$ is the position command. We assume that the robot has an internal controller that regulates the position and velocity tracking errors defined, respectively, by

$$\tilde{\mathbf{r}}_\mathrm{A} \triangleq \mathbf{r}_\mathrm{A} - \bar{\mathbf{r}}_\mathrm{A}, \tag{1}$$
$$\tilde{\mathbf{v}}_\mathrm{A} \triangleq \mathbf{v}_\mathrm{A} - \bar{\mathbf{v}}_\mathrm{A}. \tag{2}$$

By means of this internal controller [10], the error dynamics of the mobile robot can be represented in general by

$$\dot{\tilde{\mathbf{r}}}_\mathrm{A} = \tilde{\mathbf{v}}_\mathrm{A}, \tag{3}$$
$$\dot{\tilde{\mathbf{v}}}_\mathrm{A} = \mathbf{f}\big(\tilde{\mathbf{r}}_\mathrm{A}, \tilde{\mathbf{v}}_\mathrm{A}\big) + \mathbf{d}, \tag{4}$$

where $\mathbf{d} \in \mathbb{R}^n$ is a disturbance input and $\mathbf{f} \in \mathbb{R}^n$ is a known function. It is worth emphasizing that the dynamics of different mobile robots such as fully actuated and underactuated multirotor aerial vehicles, car-like robots, and underwater vehicles can be written in the same form as (3)–(4).

Consider, respectively, the following assumptions regarding the disturbance $\mathbf{d}$ and the convergence of the system (3)–(4).

Assumption 1: The disturbance is bounded according to $|\mathbf{d}| \leq \mathbf{d}^{\max}$, where $\mathbf{d}^{\max} \in \mathbb{R}^n$ is a known vector with positive components.

Assumption 2: The function $\mathbf{f}$ is such that (3)–(4) are ultimately bounded.

The boundedness in Assumption 1 is reasonable in practice. Moreover, Assumption 2 is not restrictive since it states that the system (3)–(4), which is composed of the robot and its internal controller, is robustly stable. In fact, given Assumption 1, Assumption 2 can be fulfilled by designing any stabilizing internal control [11], [12], [13].

Consider that the robot is subject to linear velocity constraints and shares the environment with moving obstacles, in such a way to impose the following constraints:

$$\mathbf{r}_\mathrm{A} \in \mathcal{P}(t), \tag{5}$$
$$\mathbf{v}_\mathrm{A} \in \mathcal{V}, \tag{6}$$

where $\mathcal{P}(t) \subseteq \mathbb{R}^n$ is the collision-free space, which varies as the obstacles move, and $\mathcal{V} \subseteq \mathbb{R}^n$ is an admissible convex set.

We now define this letter's main problem.

Problem 1: Design a collision-avoidance algorithm to generate the command $\bar{\mathbf{r}}_\mathrm{A}$ to make the robot described by (3)–(4) reach a desired position $\check{\mathbf{r}}_\mathrm{A} \in \mathbb{R}^n$ while respecting constraints (5)–(6).

III. Problem Solution

Consider that the robot moves in the presence of $n_o \in \mathbb{Z}_{>0}$ obstacles and define a set $\mathcal{I}$ containing the obstacles' identification indexes. Consider a point C_i fixed to the center of mass of obstacle $i \in \mathcal{I}$. Therefore, let us represent the position and velocity of obstacle i relative to $\mathcal{S}_r$ by $\mathbf{r}_i \in \mathbb{R}^n$ and $\mathbf{v}_i \in \mathbb{R}^n$, respectively.

To design our robust version of the CCO method, we use a position reference filter consisting of an arbitrary-order overdamped LPF, to make $\bar{\mathbf{v}}_\mathrm{A}$ smoothly converge to a target velocity $\mathbf{v}_\mathrm{A}^* \in \mathbb{R}^n$ provided by the CCO, and an integrator for calculating $\bar{\mathbf{r}}_\mathrm{A}$, as depicted in the block diagram of Figure 1. We assume that the robot is aware of the obstacles' position and velocity $\mathcal{O} \triangleq \{(\mathbf{r}_i, \mathbf{v}_i), \forall i \in \mathcal{I}\}$. Then, to address

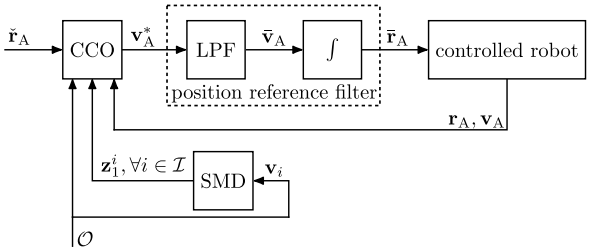

Fig. 1. Block diagram of the proposed collision-avoidance method.

the problem of collision avoidance in the presence of obstacles that can considerably change their velocities, we employ a high-order SMD [8] to provide an estimate of the acceleration of obstacle i, denoted by $\mathbf{z}_1^i \in \mathbb{R}^n$, from observations of $\mathbf{v}_i$. The CCO receives the desired position $\check{\mathbf{r}}_\mathrm{A}$ as input and $\mathbf{r}_\mathrm{A}$, $\mathbf{v}_\mathrm{A}$, $\mathbf{z}_1^i, \forall i \in \mathcal{I}$, and $\mathcal{O}$, as feedback. On the other hand, it chooses $\mathbf{v}_\mathrm{A}^*$ attempting to avoid collisions and respect the linear velocity constraint (6).

The position reference filter is designed using an overdamped LPF of an arbitrary order $p \in \mathbb{Z}_{>0}$. By considering an arbitrary p, the proposed collision-avoidance method can be combined with control laws that require commands of different smoothness degrees. The position reference filter can be represented by the state-space model

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{v}_\mathrm{A}^*, \tag{7}$$

where $\mathbf{y} \triangleq (\bar{\mathbf{r}}_\mathrm{A}, \dot{\bar{\mathbf{r}}}_\mathrm{A}, \ldots, \bar{\mathbf{r}}_\mathrm{A}^{(p)}) \in \mathbb{R}^{n(p+1)}$, $\mathbf{B} \triangleq [\mathbf{0}_{n\times np} \;\; \mathbf{I}_n\tau_y^{-p}]^\mathrm{T} \in \mathbb{R}^{n(p+1)\times n}$, $\tau_y \in \mathbb{R}_{>0}$ is a time constant,

$$\mathbf{A} \triangleq \begin{bmatrix} \mathbf{0}_{np\times n} & \mathbf{I}_{np} \\ \mathbf{0}_{n\times n} & \mathbf{F} \end{bmatrix} \in \mathbb{R}^{n(p+1)\times n(p+1)},$$
$$\mathbf{F} \triangleq \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \ldots & \mathbf{F}_p \end{bmatrix} \in \mathbb{R}^{n\times np},$$
$$\mathbf{F}_{k+1} \triangleq -\frac{p!}{k!(p-k)!}\mathbf{I}_n\tau_y^{-(p-k)}, k \in \{0, 1, \ldots, p-1\}.$$

The target velocity $\mathbf{v}_\mathrm{A}^*(t)$ provided by the CCO may be a discontinuous function [11]. In this context, note that the choice of p guarantees that the position command $\bar{\mathbf{r}}_\mathrm{A}$ is at least p-times differentiable with respect to time.

The initial position and velocity commands are set equal to the robot's respective states, while the remaining states of the filter are set equal to zero, *i.e.*,

$$\mathbf{y}(0) \triangleq (\mathbf{r}_\mathrm{A}(0), \mathbf{v}_\mathrm{A}(0), \mathbf{0}_n, \ldots, \mathbf{0}_n). \tag{8}$$

From the choice of $\mathbf{y}(0)$, it holds that $\tilde{\mathbf{r}}_\mathrm{A}(0) = \mathbf{0}_n$ and $\tilde{\mathbf{v}}_\mathrm{A}(0) = \mathbf{0}_n$. Then, under Assumptions 1–2, one can limit $\tilde{\mathbf{r}}_\mathrm{A}$ and $\tilde{\mathbf{v}}_\mathrm{A}$, respectively, as

$$\|\tilde{\mathbf{r}}_\mathrm{A}(t)\| \leq \epsilon^r(t), \tag{9}$$
$$\|\tilde{\mathbf{v}}_\mathrm{A}(t)\| \leq \epsilon^v(t), \tag{10}$$

where $\epsilon^r \in \mathbb{R}_{>0}$ and $\epsilon^v \in \mathbb{R}_{>0}$ are in general time-dependent functions. It is worth mentioning that ϵ^r and ϵ^v cannot be analytically calculated, but they can be approximated from numerical simulations of the position closed-loop system (3)–(4) considering a great amount of the possible values of the disturbance inside the bounds given by Assumption 1.

Consider that obstacle i is contained in a closed ball $\mathcal{B}(C_i, \rho_i)$, where $\rho_i \in \mathbb{R}_{>0}$ is a given radius. Also, consider that the robot is contained in $\mathcal{B}(A, \rho_\mathrm{A})$, where $\rho_\mathrm{A} \in \mathbb{R}_{>0}$ is a given radius. Therefore, a collision is assumed to occur at instant t between the robot and obstacle i if and only if

$$\|\mathbf{r}_\mathrm{A}(t) - \mathbf{r}_i(t)\| \leq \rho_{\mathrm{A}i}, \tag{11}$$

where $\rho_{\mathrm{A}i} \triangleq \rho_\mathrm{A} + \rho_i$.

Using the error bound in (9) as well as the error definition (1), condition (11) can be tightened, yielding

$$\|\bar{\mathbf{r}}_\mathrm{A}(t) - \mathbf{r}_i(t)\| \leq \rho_{\mathrm{A}i} + \epsilon^r, \tag{12}$$

whose satisfaction implies that condition (11) is true.

To prevent collisions, the adopted strategy is to avoid the satisfaction of (12) in a future time horizon of a given length $\tau \in \mathbb{R}_{>0}$. To this end, we have to predict $\bar{\mathbf{r}}_\mathrm{A}(t)$ and $\mathbf{r}_i(t)$ inside the time interval $[t_0, t_0+\tau]$, where $t_0 \in \mathbb{R}_{\geq 0}$ is the current time. By adopting the length τ, we only consider the obstacles that can cause imminent collisions, *i.e.*, collisions that might occur inside $[t_0, t_0 + \tau]$. This can be particularly useful to reduce the computational burden when many obstacles are present. However, it is worth mentioning that τ must be chosen based on the physical bounds of the obstacles' trajectories and the robot dynamics in such a way that, when a collision is detected as imminent, the robot can avoid it.

From (7), one can see that the future values of $\bar{\mathbf{r}}_\mathrm{A}$ are unknown since they depend on the future values of $\mathbf{v}_\mathrm{A}^*$, which is provided by the CCO and dependent on the future behavior of the obstacles (which is unknown). For this reason, we predict $\bar{\mathbf{r}}_\mathrm{A}$ on the time interval $[t_0, t_0 + \tau]$ by considering $\mathbf{v}_\mathrm{A}^*$ as a constant input. To support this prediction, the following lemma provides a unique solution to (7) given an initial condition $\mathbf{y}(t_0)$ and considering $\mathbf{v}_\mathrm{A}^*$ as a constant input.

Lemma 1: Considering $\mathbf{v}_\mathrm{A}^*$ as a constant input, the solution of (7) with initial condition $\mathbf{y}(t_0)$ is given by

$$\mathbf{y}(t) = e^{\mathbf{A}\delta t}\mathbf{y}(t_0) + \mathbf{G}(\delta t)\mathbf{v}_\mathrm{A}^*, \tag{13}$$

where $\delta t \triangleq t - t_0$,

$$\mathbf{G}(\delta t) \triangleq \begin{bmatrix} \mathbf{I}_{n(p+1)} & \mathbf{0}_{n(p+1)\times n} \end{bmatrix} e^{\bar{\mathbf{A}}\delta t} \begin{bmatrix} \mathbf{0}_{n\times n(p+1)} & \mathbf{I}_n \end{bmatrix}^\mathrm{T},$$
$$\bar{\mathbf{A}} \triangleq \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{n\times n(p+1)} & \mathbf{0}_{n\times n} \end{bmatrix} \in \mathbb{R}^{n(p+2)\times n(p+2)}.$$

Proof: See the Appendix. ∎

The future values of the position command $\bar{\mathbf{r}}_\mathrm{A}(t)$, considering $\mathbf{v}_\mathrm{A}^*$ as a constant input, can be calculated from (13) as

$$\bar{\mathbf{r}}_\mathrm{A}(t) = \mathbf{C}_1 e^{\mathbf{A}\delta t}\mathbf{y}(t_0) + \mathbf{G}_1(\delta t)\mathbf{v}_\mathrm{A}^*, \tag{14}$$

where $\mathbf{G}_1(\delta t) \triangleq \mathbf{C}_1\mathbf{G}(\delta t)$ and $\mathbf{C}_1 \triangleq [\mathbf{I}_n \;\; \mathbf{0}_{n\times np}] \in \mathbb{R}^{n\times n(p+1)}$.

On the other hand, to predict the obstacles position, the previous VO-based methods, in general, assume that the obstacles keep a constant velocity. However, this assumption is normally not satisfied in real applications, being addressed by using a quick replanning loop without the guarantee that collision will be avoided when the obstacles accelerate. Here, we propose a different strategy that, instead of using a first-order approximation of the obstacles' trajectories, considers a set of possible future trajectories for the obstacles based on estimates

of their acceleration bounds calculated from $\mathbf{z}_1^i$ provided by the SMD. Therefore, the predicted position $\mathbf{r}_i(t)$ of obstacle i in the time interval $[t_0, t_0+\tau]$ can be said to belong to

$$\mathcal{P}_i \triangleq \Big\{ \mathbf{r} = \mathbf{r}_i(t_0) + \delta t \mathbf{v}_i(t_0) + \hat{a}_i^{\max} \boldsymbol{\vartheta} \frac{\delta t^2}{2}, \quad \delta t \in [0,\tau],\, \boldsymbol{\vartheta} \in \mathcal{B}(\mathbf{0}_m, 1) \Big\}, \tag{15}$$

where $\hat{a}_i^{\max} \in \mathbb{R}_{\geq 0}$ is the estimate of the acceleration bound of obstacle i.

Let $\mathbf{v}_i$ be m-times differentiable such that $\mathbf{v}_i^{(m)}$ has a known Lipschitz constant vector $\boldsymbol{\gamma}_i > \mathbf{0}_n$. Then, define a vector $\boldsymbol{\gamma} \in \mathbb{R}^n$ such that $\boldsymbol{\gamma} > \boldsymbol{\gamma}_i$, $\forall i \in \mathcal{I}$. To estimate the acceleration of obstacle i, we adopt the following high-order SMD [8], [12]:

$$\begin{cases} \dot{\mathbf{z}}_0^i &= \mathbf{w}_0^i, \\ \mathbf{w}_0^i &= -\boldsymbol{\Lambda}_0 \boldsymbol{\Psi}(\boldsymbol{\gamma})^{1/(m+1)} \boldsymbol{\Psi}(|\mathbf{z}_0^i - \mathbf{v}_i|)^{m/(m+1)} \mathrm{sign}(\mathbf{z}_0^i - \mathbf{v}_i) \\ &\quad + \mathbf{z}_1^i, \\ \dot{\mathbf{z}}_1^i &= \mathbf{w}_1^i, \\ \mathbf{w}_1^i &= -\boldsymbol{\Lambda}_1 \boldsymbol{\Psi}(\boldsymbol{\gamma})^{1/m} \boldsymbol{\Psi}(|\boldsymbol{\eta}_1^i|)^{(m-1)/m} \mathrm{sign}(\boldsymbol{\eta}_1^i) + \mathbf{z}_2^i, \\ &\vdots \\ \dot{\mathbf{z}}_{m-1}^i &= \mathbf{w}_{m-1}^i, \\ \mathbf{w}_{m-1}^i &= -\boldsymbol{\Lambda}_{m-1} \boldsymbol{\Psi}(\boldsymbol{\gamma})^{1/2} \boldsymbol{\Psi}(|\boldsymbol{\eta}_{m-1}^i|)^{1/2} \mathrm{sign}(\boldsymbol{\eta}_{m-1}^i) + \mathbf{z}_m^i, \\ \dot{\mathbf{z}}_m^i &= -\boldsymbol{\Lambda}_m \boldsymbol{\Psi}(\boldsymbol{\gamma}) \mathrm{sign}(\boldsymbol{\eta}_m^i), \end{cases} \tag{16}$$

where $\boldsymbol{\eta}_k^i \triangleq \mathbf{z}_k^i - \mathbf{w}_{k-1}^i$, with $k \in \{1,\dots,m\}$, $\boldsymbol{\Lambda}_j \in \mathbb{R}^{n\times n}$ is a positive-definite diagonal matrix, with $j \in \{0,\dots,m\}$, and $\boldsymbol{\Psi}(|\cdot|) \triangleq \mathrm{diag}(|\cdot|)$ is a function that maps a vector into a diagonal matrix. The initial conditions of the SMD (16) are set to $\mathbf{z}_0^i(0) = \mathbf{v}_i(0)$ and $\mathbf{z}_j^i(0) = \mathbf{0}_n$, $\forall j \in \{1,\dots,m\}$.

Lemma 2 [8], [14]: Consider the SMD (16). For any given $\boldsymbol{\Lambda}_m$ satisfying $\boldsymbol{\Lambda}_m \mathbf{1}_n > \mathbf{1}_n$, there exists a sequence of positive-definite diagonal matrices $\{\boldsymbol{\Lambda}_0,\dots,\boldsymbol{\Lambda}_{m-1}\}$ that provides the finite-time convergence of $\mathbf{z}_k^i$ to $\mathbf{v}_i^{(k)}$, $\forall k \in \{0,\dots,m\}$ and $\forall i \in \mathcal{I}$.

It turns out that the SMD (16) is not very sensitive to parameter variation, which makes it easy to be tuned. The tuning trade-off is as follows: the larger the gains $\boldsymbol{\Lambda}_j$, where $j \in \{0,\dots,m\}$, the faster the convergence and the higher the sensitivity to measurement noises, discretization error, and unmodeled dynamics [8].

Before the SMD convergence time, denoted by $t_c^i \in \mathbb{R}_{>0}$, we have no reliable information about the bound of the obstacle i acceleration. In this context, we calculate $\hat{a}_i^{\max}$ as

$$\hat{a}_i^{\max} \triangleq \begin{cases} \max \|\mathbf{z}_1^i(t)\| & \forall t \geq t_c^i, \\ \|\mathbf{z}_1^i(t)\| & \forall t < t_c^i. \end{cases} \tag{17}$$

According to (17), before t_c^i, $\hat{a}_i^{\max}$ is converging according to the SMD (16), and after t_c^i, by maximizing $\|\mathbf{z}_1^i\|$, $\hat{a}_i^{\max}$ becomes a very accurate estimate of the obstacle i acceleration bound. However, t_c^i is hard to be calculated without conservativeness. In this sense, a very intuitive choice is to approximate t_c^i by checking when the SMD states enter in a small neighborhood of the sliding manifold, *i.e.*, when

$$\|\mathbf{z}_0^i - \mathbf{v}_i\| + \|\boldsymbol{\eta}_1^i\| + \cdots + \|\boldsymbol{\eta}_m^i\| < \alpha, \tag{18}$$

where $\alpha \in \mathbb{R}_{>0}$, is satisfied.

Using the predictions of $\bar{\mathbf{r}}_\mathrm{A}$ and $\mathbf{r}_i$ given, respectively, by (14) and (15), the tightened collision condition (12) becomes

$$\left\| \mathbf{G}_1(\delta t)\mathbf{v}_\mathrm{A}^* - \mathbf{c}_i(\delta t) - \hat{a}_i^{\max} \boldsymbol{\vartheta} \frac{\delta t^2}{2} \right\| \leq \rho_{\mathrm{A}i} + \epsilon^r, \tag{19}$$

where $\mathbf{c}_i(\delta t) \triangleq -\mathbf{C}_1 e^{\mathbf{A}\delta t}\mathbf{y}(t_0) + \mathbf{r}_i(t_0) + \delta t \mathbf{v}_i(t_0)$. One can immediately see that the collision condition (19) is satisfied if

$$\left\| \mathbf{G}_1(\delta t)\mathbf{v}_\mathrm{A}^* - \mathbf{c}_i(\delta t) \right\| \leq \rho_{\mathrm{A}i}^+(\delta t), \tag{20}$$

where $\rho_{\mathrm{A}i}^+(\delta t) \triangleq \rho_{\mathrm{A}i} + \epsilon^r + \hat{a}_i^{\max}\frac{\delta t^2}{2}$.

From the definition of $\mathbf{G}_1(\delta t)$, one can easily check that it is nonsingular $\forall \delta t > 0$. In this sense, by multiplying both sides of (20) by $\|\mathbf{G}_1^{-1}(\delta t)\|$, it can be shown that (20) is satisfied if

$$\left\| \mathbf{v}_\mathrm{A}^* - \mathbf{G}_1^{-1}(\delta t)\mathbf{c}_i(\delta t) \right\| \leq \|\mathbf{G}_1^{-1}(\delta t)\| \rho_{\mathrm{A}i}^+(\delta t). \tag{21}$$

Now, using (21), define a set of robot target velocities $\mathbf{v}_\mathrm{A}^*$ that may result in a collision with obstacle i within the time interval $(t_0, t_0+\tau]$ as

$$\mathcal{CCO}_{\mathrm{A}i}^\tau \triangleq \bigcup_{0<\delta t\leq\tau} \mathcal{B}\Big(\mathbf{G}_1^{-1}(\delta t)\mathbf{c}_i(\delta t), \|\mathbf{G}_1^{-1}(\delta t)\|\rho_{\mathrm{A}i}^+(\delta t)\Big).$$

Therefore, the set of target velocities $\mathbf{v}_\mathrm{A}^*$ that may result in a collision with any obstacle within $(t_0, t_0+\tau]$, is given by

$$\mathcal{CCO}_\mathrm{A}^\tau \triangleq \bigcup_{\forall i\in\mathcal{I}} \mathcal{CCO}_{\mathrm{A}i}^\tau.$$

By continuously choosing $\mathbf{v}_\mathrm{A}^* \notin \mathcal{CCO}_\mathrm{A}^\tau$, we can guarantee collision avoidance, thus satisfying the position constraint (5). Under this strategy, the robot can avoid collisions with obstacles that can considerably change their velocities.

To account for the velocity constraint (6), it has to be explicitly rewritten in terms of the target velocity $\mathbf{v}_\mathrm{A}^*$. To this end, note that, given the choice of $\mathbf{y}(0)$ in (8) and the design of the position reference filter (7) using the adopted LPF, it holds that the velocity command $\bar{\mathbf{v}}_\mathrm{A}$ presents no overshoot relative to the input $\mathbf{v}_\mathrm{A}^*$. Then, by choosing $\mathbf{v}_\mathrm{A}^* \in \mathcal{V}$, we guarantee that $\bar{\mathbf{v}}_\mathrm{A} \in \mathcal{V}$ since $\mathcal{V}$ is a convex set. Consequently, from (10), one can show that by choosing

$$\mathbf{v}_\mathrm{A}^* \in \mathcal{V} \ominus \mathcal{B}\big(\mathbf{0}_n, \epsilon^v\big), \tag{22}$$

guarantees that $\mathbf{v}_\mathrm{A} \in \mathcal{V}$, thus satisfying the linear velocity constraint (6). As as result, the position and velocity constraints (5)–(6) can be satisfied by continuously choosing

$$\mathbf{v}_\mathrm{A}^* \in \mathcal{V}_\mathrm{R} \triangleq \big(\mathcal{V} \ominus \mathcal{B}\big(\mathbf{0}_n, \epsilon^v\big)\big) \backslash \mathcal{CCO}_\mathrm{A}^\tau. \tag{23}$$

Note that the set $\mathcal{V}_\mathrm{R}$ is generally non-convex.

Finally, to complete the solution of Problem 1, we compute $\mathbf{v}_\mathrm{A}^*$ according to the following non-convex minimization problem that aims to satisfy the linear velocity constraint (6) and guide the robot to its desired position $\check{\mathbf{r}}_\mathrm{A}$ without collision:

$$\mathbf{v}_\mathrm{A}^* = \underset{\boldsymbol{v}\in\mathcal{V}_\mathrm{R}}{\arg\min} \|\mathbf{v}^{\mathrm{pref}} - \boldsymbol{v}\|, \tag{24}$$

where $\mathbf{v}^{\mathrm{pref}} \in \mathbb{R}^n$ is a preferred velocity. Here, $\mathbf{v}^{\mathrm{pref}}$ is a vector that points in the direction of the desired position and has the magnitude equal to the maximum admissible velocity in

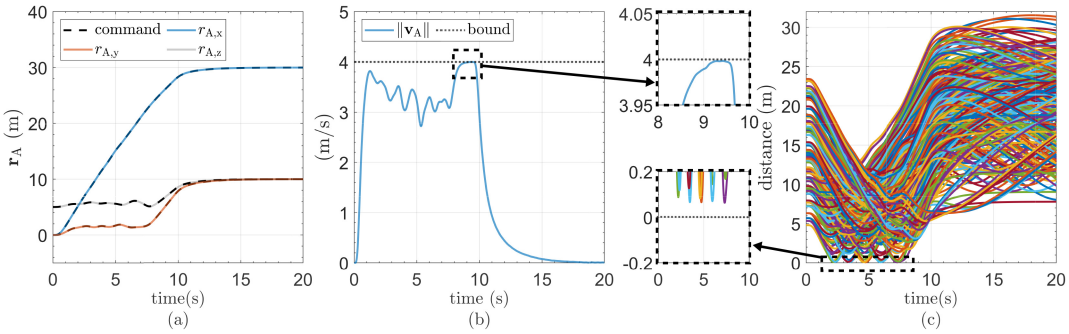

Fig. 2. Results for the proposed method. (a) Position components and their commands. (b) Euclidian norm of the quadcopter velocity and its bound. (c) Distance from the quadcopter to the obstacles.

this direction. To provide a smooth deceleration phase, in this letter, the magnitude of $\mathbf{v}^{\text{pref}}$ is gradually decreased according to the remaining distance once the vehicle is sufficiently close to the desired position. However, we highlight that the design of $\mathbf{v}^{\text{pref}}$ can be done using different strategies [4], [15].

When many obstacles are present, the set $\mathcal{V}_{\text{R}}$ may become empty, and, consequently, the optimization problem (24) may have no solution. To avoid such infeasibility, we choose a target velocity that maximizes the time to collide and respects the velocity constraint (6). The shortest time $t_{\text{col}}(\mathbf{v}_{\text{A}}^*) \in \mathbb{R}_{>0}$ for a collision corresponding to a given $\mathbf{v}_{\text{A}}^*$ is calculated by

$$t_{\text{col}}\left(\mathbf{v}_{\text{A}}^*\right) = \min_{i \in \mathcal{I}} \ \min_{0<\delta\leq\tau} \ \delta \quad \text{s.t.} \quad \mathbf{v}_{\text{A}}^* \in \mathcal{CCO}_{\text{A}i}^{\delta}.$$

Therefore, to deal with the afore-described infeasibility, we can choose the target velocity $\mathbf{v}_{\text{A}}^*$ that maximizes $t_{\text{col}}(\mathbf{v}_{\text{A}}^*)$ and satisfies the velocity constraint (6), *i.e.*,

$$\mathbf{v}_{\text{A}}^* = \underset{\mathbf{v}\in\mathcal{V}\ominus\mathcal{B}(\mathbf{0}_n,\epsilon^v)}{\arg\max} \ t_{\text{col}}(\mathbf{v}). \tag{25}$$

It is worth mentioning that if the robot is entirely surrounded by obstacles with no clear path to its desired position, it may enter a deadlock situation until the scenario changes and a collision-free trajectory becomes available.

IV. Numerical Simulation

The proposed collision-avoidance method is evaluated using a numerical simulation of an x-shaped quadcopter with a mass of 1kg, arm length of 0.5 m, and subject to the disturbance input

$$\mathbf{d}(t) = 0.5[1, \ -1, \ -1]^{\text{T}} \sin(0.1t) \ \text{N/kg}.$$

The numerical simulation is coded in MATLAB using the first-order explicit Euler method with a sampling period of 0.01 s and considering the quadcopter nonlinear equations of motion. To stabilize the vehicle dynamics, we implement the control strategy proposed in Ricardo Jr. and Santos [10], which uses a sliding mode attitude control law and a proportional-derivative position one. Under this strategy, the quadcopter closed-loop position dynamics can be written in the same form of (3)–(4). In this simulation, the inner-loop control and the outer-loop collision avoidance run at the same sampling period of 0.01 s. However, in practice, those rates must be fast enough to guarantee satisfactory tracking performance and collision avoidance without exceeding the available computing capacity.

The quadcopter has a zero initial velocity, initial position $(0, 0, 5)$ m, and is required to move to the desired position $(30, 10, 10)$ m, while avoiding collision with 270 obstacles also represented by quadcopters. The obstacles have a maximum acceleration magnitude of 1.15 m/s^2 and the vehicle has the linear velocity admissible set

$$\mathcal{V} \triangleq \left\{\mathbf{v} \in \mathbb{R}^3 \mid \|\mathbf{v}\| \leq 4\right\}.$$

The robot is considered to be contained in a sphere of radius 0.5 m, while all the obstacles are considered to be contained in spheres of radius 0.7 m.

We compare the proposed method with the original CCO [6], which is designed by considering $\epsilon^r = 0$, $\epsilon^v = 0$, and $\hat{a}_i^{\max} = 0$, $\forall i \in \mathcal{I}$, in the proposed method.

We have set $p = 4$, $\tau = 3$ s, $\tau_y = 0.15$ s, $\epsilon^r = 0.05$ m, $\epsilon^v = 0.04$ m/s, and $m = 2$. For the SMD (16), we have set $\mathbf{\Lambda}_0 = 4\mathbf{I}_3$, $\mathbf{\Lambda}_1 = 3\mathbf{I}_3$, $\mathbf{\Lambda}_2 = 2\mathbf{I}_3$, and $\boldsymbol{\gamma} = (1.5, 1.5, 1.5)$. Moreover, we approximate the solution of (24) by sampling 700 velocity vectors from a uniform distribution over $\mathcal{V} \ominus \mathcal{B}(\mathbf{0}_n, \epsilon^v)$ and looking for the sample that has the lowest cost and does not belong to $\mathcal{CCO}_{\text{A}}^{\tau}$. When no solution is found, the optimization problem (25) is then (approximately) solved by returning the sample that has the biggest time to collide.

A video containing the simulation of both methods is available at https://youtu.be/BhPcu_XXFVY.

Figures 2 and 3 present, respectively, the simulation results for the proposed and the original CCO methods. Figures 2(c) and 3(c) present the time evolution of the distance from the robot to the obstacles. They show that, despite the presence of disturbances, the proposed method guides the robot without collision, while using the original CCO, the robot collides with nine different obstacles. Figures 2(b) and 3(b) present the Euclidian norm of the robot velocity and its bound. They show that the original CCO, differently from the proposed method, violates linear velocity constraints. Lastly, Figures 2(a) and 3(a) present the robot position components and their respective commands; in both cases the tracking

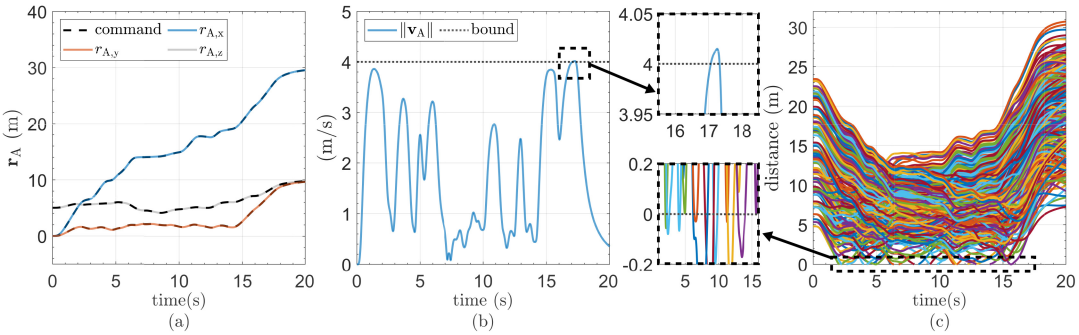

Fig. 3. Results for the original CCO method. (a) Position components and their commands. (b) Euclidian norm of quadcopter velocity and its bound. (c) Distance from the quadcopter to the obstacles.

performance is similar since they are based on the same inner-loop (robust) controller. These figures show that, under the proposed method, the quadcopter reaches its desired position faster, which is due to its better predictability of the obstacles' motion.

V. Conclusion

This letter proposed a robust collision-avoidance method for mobile robots with uncertain dynamics and subject to velocity constraints in the presence of moving obstacles that can considerably change their velocities. The proposed method is based on the CCO one, and designed using an arbitrary-order overdamped LPF to generate sufficiently smooth position commands for the robot and a high-order SMD to robustly estimate the obstacles' maximum accelerations. A detailed numerical evaluation comparing the proposed method with the original CCO was performed using a conventional quadcopter. Differently from the original CCO, the proposed method has shown to be effective in respecting linear velocity constraints and guiding the mobile robot to its desired position without colliding with the obstacles. In future works, the proposed method will be experimentally demonstrated.

Appendix

Proof of Lemma 1

The solution of the position reference filter differential equation (7) is given by

$$\mathbf{y}(t) = e^{\mathbf{A}\delta t}\mathbf{y}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{v}_A^* d\tau, \tag{26}$$

where $\delta t \triangleq t - t_0$.

The integral term in (26) cannot be directly calculated since matrix $\mathbf{A}$ is singular due to the presence of an integrator. To analytically calculate (26), we consider $\mathbf{v}_A^*$ as a constant input and define an augmented vector $\mathbf{w} \triangleq (\mathbf{y}, \mathbf{v}_A^*) \in \mathbb{R}^{n(p+2)}$. Therefore, using (7) we can write the dynamic model

$$\dot{\mathbf{w}} = \bar{\mathbf{A}}\mathbf{w}, \tag{27}$$

where

$$\bar{\mathbf{A}} \triangleq \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{n\times n(p+1)} & \mathbf{0}_{n\times n} \end{bmatrix} \in \mathbb{R}^{n(p+2)\times n(p+2)}.$$

The solution of (27) is given by $\mathbf{w}(t) = e^{\bar{\mathbf{A}}\delta t}\mathbf{w}(0)$, where

$$e^{\bar{\mathbf{A}}\delta t} = \begin{bmatrix} e^{\mathbf{A}\delta t} & \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}d\tau \\ \mathbf{0}_{n\times n(p+1)} & \mathbf{I}_n \end{bmatrix}. \tag{28}$$

Using (28), equation (26) can be rewritten as $\mathbf{y}(t) = e^{\mathbf{A}\delta t}\mathbf{y}(t_0) + \mathbf{G}(\delta t)\ \mathbf{v}_A^*$, where

$$\mathbf{G}(\delta t) \triangleq \left[\mathbf{I}_{n(p+1)}, \mathbf{0}_{n(p+1)\times n}\right] e^{\bar{\mathbf{A}}\delta t} \begin{bmatrix} \mathbf{0}_{n(p+1)\times n} \\ \mathbf{I}_n \end{bmatrix},$$

thus completing the proof.

References

[1] D. Saccani, L. Cecchin, and L. Fagiano, "Multitrajectory model predictive control for safe UAV navigation in an unknown environment," *IEEE Trans. Control Syst. Technol.*, early access, Nov. 3, 2022, doi: 10.1109/TCST.2022.3216989.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[3] M. J. Er and C. Deng, "Obstacle avoidance of a mobile robot using hybrid learning approach," *IEEE Trans. Ind. Electron.*, vol. 52, no. 3, pp. 898–905, Jun. 2005.

[4] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.

[5] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3475–3482.

[6] M. Rufli, J. Alonso-Mora, and R. Siegwart, "Reciprocal collision avoidance with motion continuity constraints," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 899–912, Aug. 2013.

[7] D. Bareiss and J. Van Den Berg, "Generalized reciprocal collision avoidance," *Int. J. Robot. Res.*, vol. 34, no. 12, pp. 1501–1514, 2015.

[8] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *Int. J. Control*, vol. 76, nos. 9–10, pp. 924–941, 2003.

[9] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993.

[10] J. A. Ricardo Jr. and D. A. Santos, "On the assurance of time-scale separation in hierarchical sliding mode control of underactuated multicopters," *Aerosp. Sci. Technol.*, to be published.

[11] J. A. Ricardo Jr. and D. A. Santos, "Robot guidance and control using global sliding modes and acceleration velocity obstacles," in *Proc. Int. Workshop Variable Struct. Syst. Sliding Mode Control*, 2022, pp. 41–46.

[12] J. A. Ricardo Jr. and D. A. Santos, "Smooth second-order sliding mode control for fully actuated multirotor aerial vehicles," *ISA Trans.*, vol. 129, pp. 169–178, Oct. 2022.

[13] S. He, "Feedback control design of differential-drive wheeled mobile robots," in *Proc. IEEE Int. Conf. Adv. Robot.*, 2005, pp. 135–140.

[14] Y. B. Shtessel, I. A. Shkolnikov, and A. Levant, "Smooth second-order sliding modes: Missile guidance application," *Automatica*, vol. 43, no. 8, pp. 614–619, 2007.

[15] J. Van Den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2008, pp. 1928–1935, doi: 10.1109/ROBOT.2008.4543489.