Reciprocal Collision Avoidance With Motion Continuity Constraints

Martin Rufli, *Member, IEEE*, Javier Alonso-Mora, *Student Member, IEEE*, and Roland Siegwart, *Fellow, IEEE*

***Abstract*—This paper addresses decentralized motion planning among a homogeneous set of feedback-controlled, *decision-making* agents. It introduces the *continuous control obstacle* (C^n-CO), which describes the set of $\mathcal{C}^n$-continuous control sequences (and thus trajectories) that lead to a collision between interacting agents. By selecting a feasible trajectory from C^n-CO's complement, a collision-free motion is obtained. The approach represents an extension to the *reciprocal* velocity obstacle (RVO, ORCA) collision-avoidance methods so that trajectory segments verify $\mathcal{C}^n$ continuity rather than piecewise linearity. This allows the large class of robots capable of tracking $\mathcal{C}^n$-continuous trajectories to employ it for partial motion planning directly—rather than as a mere tool for collision checking. This paper further establishes that both the original velocity obstacle method and several of its recently developed reciprocal extensions (which treat specific robot physiologies only) correspond to particular instances of C^n-CO. In addition to the described extension in trajectory continuity, C^n-CO thus represents a *unification* of existing RVO theory. Finally, the presented method is validated in simulation—and a parameter study reveals under which environmental and control conditions C^n-CO with $n > 0$ admits significantly improved navigation performance compared with inflated approaches based on ORCA.**

***Index Terms*—Autonomous agents, path planning for multiple mobile robot systems, reciprocal collision avoidance.**

I. Introduction

SUCCESSFUL robot operation builds on at least three interconnected competences, namely, localization, mapping, and motion planning/control. The latter is concerned with computing a (lowest cost) path or trajectory between two configurations embedded in a cost field, while taking into account motion constraints, static obstacles, and dynamic objects. The important case where dynamic objects are *decision-making agents* forms the topic of this paper.

Manuscript received August 3, 2012; revised January 23, 2013; accepted April 11, 2013. Date of publication May 15, 2013; date of current version August 2, 2013. This paper was recommended for publication by Associate Editor C. Torras and Editor J.-P. Laumond upon evaluation of the reviewers' comments. This work was supported by the European Project *V-Charge* under Contract FP7-ICT 269916.

M. Rufli and R. Siegwart are with the Institute of Robotics and Intelligent Systems, ETH Zurich, 8092 Zurich, Switzerland (e-mail: martin.rufli@mavt.ethz.ch; rsiegwart@ethz.ch).

J. Alonso-Mora is with the Institute of Robotics and Intelligent Systems, ETH Zurich, 8092 Zurich, Switzerland, and also with Disney Research Zurich, 8092 Zurich, Switzerland (e-mail: jalonso@disneyresearch.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org including a supplementary color MPEG-4 file that visualizes typical sample runs.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2013.2258733

A. Related Work

Specific problems of the motion planning problem have been studied since the late 17th century [1], but progress since has proven slow and difficult. To this day, only a few local (convex) problems concerned with stability and convergence—spanning the modern field of control theory [2]—are understood analytically. It is thus not surprising that the large remaining space of optimization problems has been attacked by approaches admitting various forms of approximations. In this regard, advances in deterministic graph search [3], [4], graph representation [5], [6], and randomized sampling-based methods [7] have enabled (approximately optimal) solution strategies on a global scale including obstacles. The resulting fundamental separation between (global) planning and motion control has had a tremendous influence on mobile robotic navigation and led to systems fitted with multitiered navigation frameworks that often lacked coherence between the individual modules [8]. The recent development of asymptotically optimal sampling-based planning algorithms [9] and optimal tree-based approaches [10] has considerably softened this separation and, for the first time, enabled truly unified *system-compliant* online motion planning with optimality guarantees. Consequently, robotic agents increasingly operate in challenging dynamic environments with applications as diverse as warehouse delivery [11], service robotics [12], and entertainment [13]. Nonetheless, if this operation takes place among other *decision-making* agents, system-compliant planning alone does not seem adequate to ensure performant navigation. Rather, it becomes important that the individual planning strategies are aware of (and take into account) that other agents are also engaging in a similar activity. Yet, many prevalent navigation methods remain centralized and, hence, do not scale well [14], [15] or are limited to open-loop motion predictions instead of considering (collaborative) interaction directly [16], [17].

The reciprocal velocity obstacle (RVO) method [18], by contrast, models robot interaction both in a decentralized manner and pairwise optimally. It restricts robots' action capabilities to a set of constant velocities. Under the assumption that other agents also continue their present motion along a straight line trajectory, future collisions may be estimated as a function of *relative velocity* alone. Although RVO was originally developed as a collision checking method for holonomic (i.e., omnidirectionally actuatable) robots and results in $\mathcal{C}^0$-continuous trajectories, its success in multirobot applications paved the road toward a multitude of extensions to and revisions of the basic framework: The optimal reciprocal collision-avoidance (ORCA) method [19] prevents reciprocal dances and casts the problem into a linear programming framework, which can be solved efficiently.

1552-3098/$31.00 © 2013 IEEE

ORCA-DD [20] and NH-ORCA [21] are extensions to robots of type unicycle, which artificially inflate the agents' diameter in compensation for the more constrained kinematics' inability to maneuver $\mathcal{C}^0$-continuous paths. However, these methods still subject robots to sudden changes in speed. The acceleration velocity obstacle (AVO) method [22] extends the continuity of the RVO solution by one order (to a $\mathcal{C}^1$-continuous trajectory) and, thus, obtains a smoother response that can be followed by some dynamically constrained vehicles. Nevertheless, for several practically relevant robot physiologies, continuity in actuators can still not be enforced. The B-ORCA method [23], on the other hand, allows for arbitrary motion continuity, but requires an inflation of the robot shape similar to [21].

Thus, despite the large body of related contributions, we note that most of the extensions are limited toward a specific vehicle model or a specific order of solution continuity. In this paper, by contrast, we derive a unifying theory for reciprocal collision avoidance based on the concept of *control obstacles*.

B. Contributions

This paper develops three main contributions. First, we introduce the *order n continuous control obstacle* (C^n-CO), which describes the set of feasible feedback control trajectories that lead to a collision between interacting robots. The complement of this set forms the basis for safe motion planning. The method represents an extension to the RVO collision-avoidance method, such that trajectory segments verify $\mathcal{C}^n$ continuity rather than piecewise linearity. This property renders the presented method suitable as the local motion planning component for the large class of practically relevant robot physiologies that admit a linear or feedback-linearizable model representation.

Second, we establish that both the original velocity obstacle (VO) method and several of its recently developed reciprocal extensions (which treat specific robot physiologies only) correspond to particular instances of C^n-CO. In addition to the described extension in trajectory continuity, C^n-CO, thus, represents a *unification* of existing RVO theory.

Third, we conduct an experimental evaluation of C^n-CO on a set of standard test scenarios. Additionally, we provide the results of a thorough parameter study that reveals under which environmental and control conditions the presented method with $n>0$ is preferable over previous approaches that take the vehicle's kinematics into account via a radius enlargement of the original (zero-order) ORCA framework (see [20], [21], and [23]).

C. Limitations

While the C^n-CO method significantly extends the applicability of the relative velocity paradigm (as outlined previously), it also inherits some of VO's limitations. In particular, C^n-CO requires agents to conform to a circular shape. For noncircular objects, collision avoidance is only guaranteed if the out-circle of their shape is selected instead. Furthermore, all agents are expected to employ the same collision-avoidance strategy, including *identical control parameters*. For heterogeneous groups of robots, a combination between C^n-CO and the method, which is presented in [23], may be employed instead.

Finally, this paper describes a method for *local* navigation. As such, it requires a global motion planner to guarantee convergence among large static obstacles. One possible implementation is presented in Section VI, which results in an interactive partial motion planning framework.

D. Organization

The remainder of this paper is structured as follows: In Section II, we review the relative velocity paradigm that the control obstacle builds upon, extends, and unifies. We then develop the C^n-CO method for *linear* robot models in *normal form* in Section III. In Section IV, we describe the adaptations that are necessary to enable the treatment of nonlinear systems via the well-known method of feedback linearization. Section V describes how C^n-CO may be seamlessly integrated into global graph search. The presented methods are then experimentally validated on a set of standardized test scenarios (see Section VI). Finally, in Section VII, we discuss the obtained results and conclude the paper in Section VIII.

II. Background

The development of the C^n-CO method builds directly on VO theory, and thus the relative velocity paradigm. In this section, we outline the fundamentals of VO [24] and its reciprocal extension ORCA [19]. We employ the formulation of [22], and set scalars in lower case italics (x), vectors in lower case bold ($\mathbf{x}$), and sets in uppercase italics (X).

A. Velocity Obstacle

The original VO method [24] considers a single decision-making agent A (of radius r_A), which is located at position $\mathbf{p}_A$ with velocity $\mathbf{v}_A$, that moves among known objects B. Agent A's velocity obstacle, which is induced by object B, can then be introduced as a useful extension of the work-space obstacle concept.

Definition 1 (Velocity Obstacle): The velocity obstacle VO^{τ}_{AB}, which is induced for agent A by agent (or object) B, for a time horizon τ is the set of all *relative* velocities $\mathbf{v}_{AB}=\mathbf{v}_A-\mathbf{v}_B$ of A with respect to B that will result in a collision between A and B before time τ.

The VO method thus assumes that direct command of velocities is feasible, and further that other objects and agents continue to move along a linear extrapolation of their current motion. Hence, it constitutes a predictive first-order method. Under these assumptions, the colliding configurations at time t are given by

$$\|\mathbf{p}_{AB,0}+\mathbf{v}_{AB}t\|_2<r_{AB}$$

where $\mathbf{p}_{AB,0}=\mathbf{p}_A(0)-\mathbf{p}_B(0)$, and $r_{AB}=r_A+r_B$. The composite velocity obstacle VO^{τ}_{AB} is obtained as the union of velocities leading to collision

$$VO^{\tau}_{AB}=\bigcup_{0<t\leq\tau} D\left(-\frac{\mathbf{p}_{AB,0}}{t},\frac{r_{AB}}{t}\right)$$

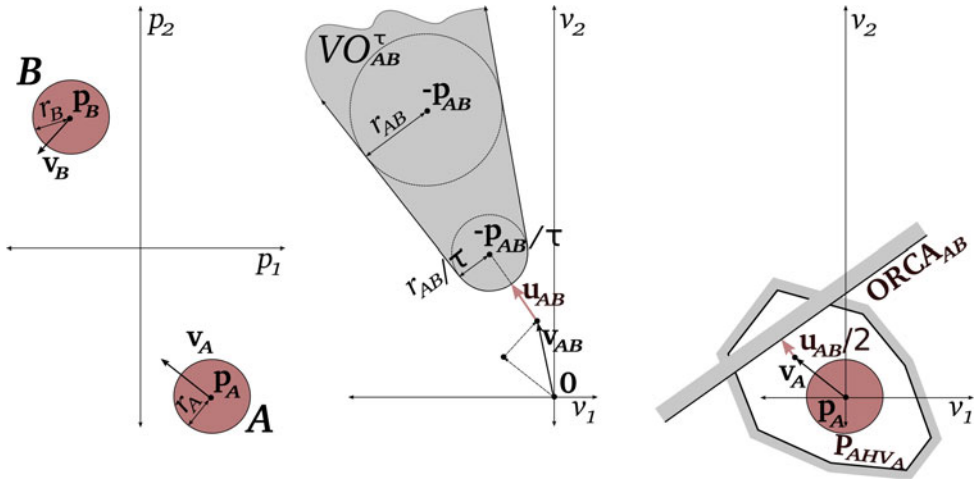


Fig. 1. VO and ORCA methods. (Left) Two agents A (located at $\mathbf{p}_A$) and B (at $\mathbf{p}_B$) interact using a VO-based method. In the VO formulation, A assumes that B continues moving along its current velocity; hence, $\mathbf{v}_{B,0} = \mathbf{v}_{B,\infty}$. In the ORCA approach, A assumes that B also takes part in collision avoidance. (Center) Velocity obstacle VO^{τ}_{AB} induced by B is composed of all relative new velocity choices $\mathbf{v}_{AB,\infty} = \mathbf{v}_{A,\infty} - \mathbf{v}_{B,\infty}$ that lead to a collision before time τ. In the VO formulation, A selects a new velocity $\mathbf{v}_{A,\infty}$ outside of $VO^{\tau}_{AB} \oplus \mathbf{v}_B$ to remain safe. In the ORCA approach, a vector $\mathbf{u}_{AB}$ is constructed that originates at $\mathbf{v}_{AB}$ and ends at the point on the VO boundary closest to $\mathbf{v}_{AB}$. (Right) ORCA approach only. Each agent is subjected to one $ORCA$ half-plane constraint for each interacting agent. Under the assumption that interaction proceeds *fairly*, $ORCA_{AB}$ is characterized by the point–direction pair $(\mathbf{v}_A + \frac{\mathbf{u}_{AB}}{2}, R(\frac{\pi}{2})\mathbf{u}_{AB})$, where $R(\cdot)$ denotes the 2-D rotation matrix. In the case where robots are kinematically constrained, additional constraints that limit trajectory tracking error (such as P_{AHV_A} in the NH-ORCA approach [21]) may be specified. A then selects a velocity $\mathbf{v}_{A,\infty}$ that fulfills all constraints and minimizes the Euclidean distance to a preferred velocity $\mathbf{v}_A^{\text{pref}}$, which is linked to the agent's motion target.

where $D(\mathbf{p}, r)$ is the disk of radius r centered at $\mathbf{p}$ [22]. The VO of A, which is induced by B, is illustrated in Fig. 1.

Collision avoidance is then accomplished as follows: Agent A selects a new velocity $\mathbf{v}_{A,\infty} \notin VO^{\tau}_{AB} \oplus \mathbf{v}_B$ (with $\oplus$ the Minkowski sum operator) to remain safe for at least τ seconds under the stated assumptions. For local navigation, $\mathbf{v}_{A,\infty}$ is usually selected so that it minimizes the Euclidean distance to $\mathbf{v}_A^{\text{pref}}$, which in turn points in direction of the next (sub)goal extracted from a global waypoint-based plan. $\mathbf{v}_{A,\infty}$ is then applied for a short time period until the control loop begins anew. In practice, the constant velocity assumption rarely holds but is addressed by having a fast replanning loop.

B. Reciprocal Velocity Obstacle

In the presence of decision-making agents, rather than passive dynamic objects, the assumptions underlying the VO approach become inaccurate and may lead to strong oscillations in the resulting trajectories. The recently introduced RVO [18] and ORCA [19] methods eliminate these oscillations. They allow us to compute a decentralized steady-state solution directly by *implicitly* requiring that the collision avoidance effort is shared between interacting agents. We refer to such methods as predictive-interactive first-order approaches. The RVO method proposes a shifted formulation of that presented for the VO approach: By shifting the collision cone by $VO^{\tau}_{AB} \oplus \frac{\mathbf{v}_A + \mathbf{v}_B}{2}$ instead of $VO^{\tau}_{AB} \oplus \mathbf{v}_B$, interacting agents partake at least half in any resulting avoidance maneuver. Unfortunately, the agents may choose to avoid each other on opposing sides, which may result in so-called reciprocal dances (not unlike to the ones observed in human crowds) [19]. ORCA, on the other hand, constructs a reciprocally collision-free velocity by solving an optimization instance that involves linear constraints in velocity space, where each constraint is computed from the VO induced by a particular neighboring agent. Additional constraints can be enforced to account for actuator limits [21], [23]. Fig. 1 illustrates and describes the methods. Due to the use of linear constraints, ORCA prevents oscillations altogether, and ensures that contradictory decisions cannot be taken in absence of noise. See the literature for a more thorough treatment of this topic.

III. C^n-CO: Higher Order Continuous Velocity Obstacles for Linear Systems in Normal Form

A. Overview

We now proceed by developing the main contribution of this paper. We construct the control obstacle of arbitrary continuity C^n-CO, which represents the set of all colliding *relative* control sequences between two interacting agents that perform feedback control onto straight line reference trajectories [see (1)]. In contrast with other methods that have been developed for input or control obstacles [25], [26], our approach admits a formulation in relative velocity space that is suitable for *reciprocal* collision avoidance. In particular, $\mathbf{v}_{AB,\infty}$ appears linearly and, thus, separably in the resulting formulation, analogous to RVO. Our restriction to straight line reference segments is a consequence of this. Nonetheless, the transients locally ensure an expressive path set suitable for local motion planning. This property is exploited in Section V to integrate C^n-CO with global graph search.

In this section, we develop the method for linear robot representations in *normal form* (i.e., a chain of integrators; see System 1), such as holonomic robots with acceleration control. In Section III-B, we derive a state feedback controller that lets the agent converge onto straight line reference trajectories. By using this formulation, the trajectories of two agents can

be combined in relative velocity space from which a control obstacle may be computed. This obstacle specifies the set of control sequences leading to a collision (see Section III-C). Finally, in Section III-D, we describe how to transform the relative control obstacle into absolute coordinates, while achieving the desired reciprocity in collision avoidance in absence of explicit communication.

B. Definitions

System 1 introduces the proposed agent description.

System 1 (Chain of Integrators):

$$\mathbf{p}^{(n+1)} = \mathbf{u}.$$

Let $\mathbf{u} \in \mathbb{R}^m$ be piecewise continuous, and $(.)^{(n+1)}$ represent the derivative of order $n+1$ with respect to time. Then, the resulting solution $\mathbf{p}$(t) is $\mathcal{C}^n$-differentiable. For $n=0$, a system representation that is directly compatible with the original ORCA approach is obtained. In the case where the input signal $\mathbf{u}$(t) is such that System 1 is controlled onto (1), we furthermore derive below that $\mathbf{p}$(t) remains compatible with the relative velocity paradigm. This motivates the definition of a more general obstacle.

Definition 2 (C^n-Control Obstacle): The *continuous control obstacle of order n* that is induced for agent A by agent B for a time horizon τ (C^n-CO^{τ}_{AB}) is the set of all *relative* desired velocities at $t \to \infty$: $\mathbf{v}_{AB,\infty} = \mathbf{v}_{A,\infty} - \mathbf{v}_{B,\infty}$ of A with respect to B that result in a collision before time τ if both agents are modeled according to System 1 and controlled via state feedback onto the piecewise linear trajectory given by (1).

C. Control Trajectory for C^n-CO

The linear trajectory segments resulting from the original ORCA approach represent primitives for which reciprocal collision avoidance for $t \to \infty$ can be specified. In the C^n-CO approach, we retain these segments as reference trajectories

$$\mathbf{p}_{\mathrm{ref}}(t) = \mathbf{p}_{\mathrm{ref},0} + \mathbf{v}_{\infty} t, t \in [0, \infty) \tag{1}$$

where $\mathbf{v}_{\infty}$ represents the ideal (or desired) velocity vector at $t \to \infty$. The kinematic model of System 1 with $n=0$ is able to precisely follow such reference motions $\mathbf{p}_{\mathrm{ref}}(t)$. Conversely, in the case of $n>0$, it cannot precisely track (1) for almost all initial conditions. However, for $n>0$, we may formulate a state feedback control law

$$\begin{aligned} \mathbf{p}^{(n+1)} &= \mathbf{u}(t) \\ \mathbf{u}(t) &= c_n(\mathbf{p}^{(n)}_{\mathrm{ref}} - \mathbf{p}^{(n)}) \\ &\quad + \cdots + c_0(\mathbf{p}_{\mathrm{ref}} - \mathbf{p}) \end{aligned}$$

where c_i are control parameters, and $\mathbf{p}_{\mathrm{ref}}$ is specified according to (1) with $\mathbf{p}^{(k)}_{\mathrm{ref}} \equiv 0, \forall k > 1$. The control input can then be rewritten as

$$\begin{aligned} \mathbf{u}(t) = &-c_n \mathbf{p}^{(n)} - \cdots - c_2 \ddot{\mathbf{p}} \\ &+ c_1(\mathbf{v}_{\infty} - \dot{\mathbf{p}}) + c_0(\mathbf{p}_{\mathrm{ref},0} + \mathbf{v}_{\infty} t - \mathbf{p}). \end{aligned} \tag{2}$$

We will now show that the resulting transients onto (1) also admit a reciprocal formulation: System 1, which is controlled via the feedback law (2) represents a nonhomogeneous linear ordinary differential equation (ODE) whose solutions have been studied extensively [27]. Such solutions consist of a complementary $\mathbf{p}^{\mathrm{c}}(t)$ and a particular $\mathbf{p}^{\mathrm{p}}(t)$ part with $\mathbf{p}(t) = \mathbf{p}^{\mathrm{c}}(t) + \mathbf{p}^{\mathrm{p}}(t)$. The particular solution corresponds to the reference trajectory of (1) and is independent of the order of System 1

$$\mathbf{p}^{\mathrm{p}}(t) = \mathbf{p}_{\mathrm{ref},0} + \mathbf{v}_{\infty} t, t \in [0, \infty).$$

The complementary solution is obtained by solving the associated characteristic polynomial

$$\mathbf{q}^{n+1} + c_n \mathbf{q}^n + \cdots + c_1 \mathbf{q} + c_0 \mathbf{1} = \mathbf{0}$$

and accounts for the added continuity requirements of the C^n-CO approach. For C^0-CO (i.e., ORCA), the complementary solution vanishes. In the general case, for the control law to have a vanishing steady-state error, the boundary conditions $\mathbf{p}^{\mathrm{c}}(0) = \lim_{t \to \infty} \mathbf{p}^{\mathrm{c}}(t) = 0$ are imposed.

Example 1: (Construction of $\mathbf{p}^{\mathcal{C}^1}(t)$) We illustrate the solution process for System 1 with continuity in velocity ($n=1$). Intuitively, System 1 with $n=1$ can be thought of as a point mass m attached to a moving reference position $\mathbf{p}_{\mathrm{ref}}$ via a linear spring with spring constant k and a damper with viscous damping coefficient d. The associated characteristic polynomial is

$$\begin{aligned} \mathbf{0} &= \mathbf{q}^2 + c_1 \mathbf{q} + c_0 \mathbf{1} \\ &= \mathbf{q}^2 + d/m \mathbf{q} + k/m \mathbf{1}. \end{aligned}$$

In this case, the system's complementary solution admits three *stable* regimes, namely, underdamped, overdamped, and critically damped control. The resulting regime is determined through the control parameters c_i or, equivalently, k and d. In the absence of other constraints, the critically damped solution ($d = 2\sqrt{mk}$) represents a natural choice, as it provides the fastest nonoscillating decay to zero. After insertion of the boundary conditions and with $\lambda_0 = \sqrt{\frac{k}{m}}$, the complementary solution takes the form

$$\begin{aligned} \mathbf{p}^{\mathrm{c}}(t) = e^{-t\lambda_0} &[\mathbf{p}_0 - \mathbf{p}_{\mathrm{ref},0} \\ &+ t\,(\mathbf{v}_0 - \mathbf{v}_{\infty} + \lambda_0(\mathbf{p}_0 - \mathbf{p}_{\mathrm{ref},0}))]. \end{aligned}$$

The combined solution for $n=1$ is then given by

$$\mathbf{p}^{\mathcal{C}^1}(t) = \mathbf{p}^{p}(t) + \mathbf{p}^{c}(t). \tag{3}$$

We note that in (3), the desired velocity $\mathbf{v}_{\infty}$ appears in the linear and additive form that is required by the relative velocity paradigm if the parameter λ_0 is selected identically for all agents. Example 1 thus demonstrates that there exist more complex than linear motions that are compatible with the RVO family of approaches. Similar results can be computed for the under- and over-damped cases. However, then two distinct eigenvalues are obtained.

The combined solution in (3) also unbosoms the similarity to the AVO approach [22] that generates an input force proportional to the difference between an agent's instantaneous velocity $\mathbf{v}(t)$

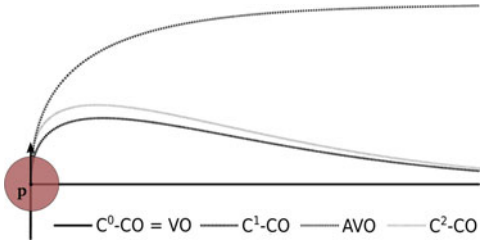


Fig. 2. Trajectory continuity across replanning steps. Assume that robot A is located at $\mathbf{p}_{A,0} = \mathbf{p}$ and faces upward with an initial velocity of $\mathbf{v}_{A,0} = 1\,\mathrm{m/s}$, while a replanning event due to a new goal located to the right is triggered. Using the ORCA $= C^0$-CO approach, $\mathbf{v}_\infty$ is attained immediately. C^n-CO-controlled agents with $n > 0$ approach the new reference trajectory more slowly and, thereby, ensure continuity of appropriate order. AVO [22] guarantees $\mathcal{C}^1$ continuity but retains a steady-state tracking error to $\mathbf{p}_{\mathrm{ref}}$.

and the reference velocity $\mathbf{v}_\infty$

$$\mathbf{p}^{\mathrm{AVO}}(t) = \mathbf{p}_0 + t\mathbf{v}_\infty + 1/\lambda_0(e^{-t\lambda_0} - 1)(\mathbf{v}_\infty - \mathbf{v}_0). \quad (4)$$

A comparison between (3) and (4) affirms that AVO is indeed a special case of C^1-CO with spring constant $c_0 = 0$. This choice is responsible for AVO's observed steady-state tracking error at $t \to \infty$, as depicted in Fig. 2.

It can be furthermore shown that all solutions that are obtained by controlling System 1 of arbitrary order onto the reference trajectory (1) via state-feedback are in accordance with the relative velocity paradigm. In particular, the important case of $n = 2$ may be derived as

$$\mathbf{p}^{\mathcal{C}^2}(t) = \mathbf{p}^{\mathcal{C}^1}(t) + Fe^{-t\lambda_1} + ((\lambda_1 - \lambda_0)Ft - F)\,e^{-t\lambda_0} \quad (5)$$

with

$$F = \frac{\mathbf{a}_0 + 2\lambda_0(\mathbf{v}_0 - \mathbf{v}_\infty) + \lambda_0^2(\mathbf{p}_0 - \mathbf{p}_{\mathrm{ref},0})}{(\lambda_0 - \lambda_1)^2}$$

and λ_1 a design parameter that is related to the decay of the initial acceleration $\mathbf{a}_0$. Notice that $\mathbf{p}^{\mathcal{C}^1}(t)$ forms an additive part of the solution. The derivation is analogous to Example 1.

Fig. 2 depicts $\mathbf{p}^{\mathcal{C}^n}(t)$ for $n \in \{0, 1, 2\}$ and identical $\lambda_{(\cdot)}$. In our present implementation, we select $\mathbf{p}_{\mathrm{ref},0} = \mathbf{p}_0$ (i.e., the reference trajectory passes through the robot's current position), which substantially simplifies both (3) and (5) and allows for a simple way to infer other agents' reference position in absence of communication. In principle, $\mathbf{p}_{\mathrm{ref},0}$ could be used as an additional degree of freedom for the placement of reference trajectories, however. This would also allow us to regenerate identical solution curves across replanning steps.

D. Derivation of C^n-CO

Given the general solution of System 1, we now construct C^n-CO$^\tau_{AB}$ for two interacting agents A and B that each employ state feedback control onto their reference trajectory $\mathbf{p}_{\mathrm{ref}}(t)$, following the procedure outlined in Section III-C. C^n-CO$^\tau_{AB}$ represents the set of relative reference velocities $\mathbf{v}_{AB,\infty}$ that result in a collision at time $t \le \tau$. Thus, C^n-CO$^\tau_{AB} = \{\mathbf{v}_{AB,\infty} | \|\mathbf{p}_{AB}(t)\|_2 \le r_{AB}, t \le \tau\}$ (see Fig. 3). For

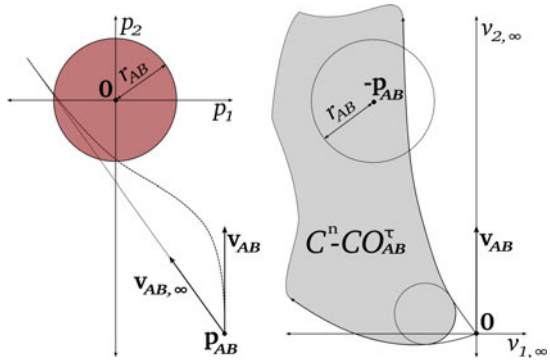


Fig. 3. Control obstacle with continuity constraints (C^n-CO). (Left) Two agents A (located at $\mathbf{p}_A$) and B (at $\mathbf{p}_B$) move along continuous trajectories using the C^n-CO approach. Their relative position $\mathbf{p}_{AB}(t)$ can be expressed as a function of $\mathbf{v}_{AB,\infty}$. (Right) Continuous control obstacle C^n-CO$^\tau_{AB}$, which is induced for A by B, is then composed of all relative velocities $\mathbf{v}_{AB,\infty} = \mathbf{v}_{A,\infty} - \mathbf{v}_{B,\infty}$ that lead to a collision before time τ.

notational simplicity, we explain the solution process for the example of $n = 1$.

Example 2 (Construction of C^1-CO$^\tau_{AB}$): A collision between agents A and B occurs if their relative position becomes smaller than their combined radius, i.e., $\|\mathbf{p}_{AB}(t)\| \le r_{AB}$. The agents' relative position follows from (3) if λ_0 is selected identically for both of them

$$\mathbf{p}_{AB}(t) = \mathbf{p}_{AB,0} + t\mathbf{v}_{AB,\infty} + e^{-t\lambda_0}t(\mathbf{v}_{AB,0} - \mathbf{v}_{AB,\infty}).$$

The relative velocity $\mathbf{v}_{AB,\infty}$ may be separated by dividing both sides by $t(1 + e^{-t\lambda_0})$. We obtain

$$\left\| \mathbf{v}_{AB,\infty} + \frac{\mathbf{p}_{AB,0} + te^{-t\lambda_0}\mathbf{v}_{AB,0}}{t(1 + e^{-t\lambda_0})} \right\| \le \frac{r_{AB}}{t(1 + e^{-t\lambda_0})}.$$

Finally, the continuous VO C^1-CO$^\tau_{AB}$ is given by the union of disks D centered at $\mathbf{p}_\odot(t)$ with radius $r_\odot(t)$

$$C^1\text{-CO}^\tau_{AB} = \bigcup_{0<t\le\tau} D(\mathbf{p}_\odot(t), r_\odot(t))$$

$$D(\mathbf{p}_\odot(t), r_\odot(t)) = D\left(-\frac{\mathbf{p}_{AB,0} + te^{-t\lambda_0}\mathbf{v}_{AB,0}}{t(1 + e^{-t\lambda_0})}, \frac{r_{AB}}{t(1 + e^{-t\lambda_0})}\right).$$

Similar results hold for C^n-CO$^\tau_{AB}$ with higher order continuity, as $\mathbf{v}_{AB,\infty}$ always appears linearly and, thus, separably in the solution of $\mathbf{p}_{AB}(t)$. The left and right boundary segments $\mathbf{b}^\pm(t)$ of C^n-CO$^\tau_{AB}$ can be developed in closed form. They are given by

$$\mathbf{b}^\pm(t) = \mathbf{p}_\odot(t) - \frac{r_\odot(t)}{\dot{r}_\odot(t)}\dot{\mathbf{p}}_\odot(t) + \mathbf{q}^\pm\left(\frac{r_\odot(t)}{\dot{r}_\odot(t)}\dot{\mathbf{p}}_\odot(t), r_\odot(t)\right)$$

with

$$\mathbf{q}^\pm(\mathbf{p}, r) = \begin{bmatrix} s & \mp r \\ \pm r & s \end{bmatrix} \mathbf{p}\frac{l}{\|\mathbf{p}\|^2}$$

$$s = \sqrt{\|\mathbf{p}\|^2 - r^2}.$$

These equations were first derived in [22] for AVO.

E. C^n-CO in Reciprocal Collision Avoidance

Similar to the ORCA algorithm, we may require that each agent partially engages in collision avoidance based on a linear partitioning scheme to obtain reciprocity in navigation. However, contrary to VO, the boundary segments of C^n-CO with $n>0$ are curved, which would necessitate consideration of its convex hull to properly designate the vector $\mathbf{u}_{AB}$ (introduced in Fig. 1). If the convex hull was applied on the whole control obstacle, large parts of the feasible relative velocity space may be wasted. Fortunately, interacting agents are typically subject to input and state constraints. This allows us to restrict our attention to the reachable subset (see Example 3) of the relative velocity space, and apply the convex hull on that only. Let us thus develop the shape of the reachable set V_{AB} in relative velocity space before proceeding with the computation of fair partitions. V_{AB} amounts to the Minkowski sum of the reachable sets in absolute velocity space, namely, $V_{AB}=V_A\oplus -V_B$. Their shape, in turn, depends on the particular constraints and agent model considered. Constraints arising from generic (non)linear models are discussed in Section IV. For decoupled linear systems, however, state constraints act along each dimension of the physical space independently, resulting in simple reachable shapes for V_A and V_B. Let us illustrate this with an example.

Example 3 (Set of Reachable Velocities $V_{A,\infty}$ Using C^1-CO): From (3) and its derivatives, the time instances t^{ext} at which extremal velocities $\mathbf{v}^{\text{ext}}$ and accelerations $\mathbf{a}^{\text{ext}}$ occur along control trajectories are given by

$$\mathbf{v}^{\text{ext}}=\mathbf{v}(2/\lambda_0)=\mathbf{v}_\infty-(\mathbf{v}_0-\mathbf{v}_\infty)e^{-2}$$
$$\mathbf{a}^{\text{ext}}=\mathbf{a}(3/\lambda_0)=(\mathbf{v}_0-\mathbf{v}_\infty)e^{-3}\lambda_0.$$

By observing the values both at these points and at the boundaries, the set of terminal velocities $\mathbf{v}_\infty$, which is compliant with velocity and acceleration constraints, is obtained. For decoupled linear system models, the resulting set is square. In order to guarantee equal responsiveness in all directions, we restrict $V_{AB,\infty}$ to its in-circle, however.

The procedure for $n>1$ is identical to the one in Example 3, but the number of extremal points that need to be verified grows with n. In practice, it is thus often more appropriate to sample a discrete set of trajectories $\mathbf{p}(t)$ and selectively *discard* control sequences that lead to a violation of a system constraint. By doing so, arbitrary constraints may be considered.

After $V_{AB,\infty}$ (i.e., the set of reachable relative velocities) has been computed, a collision-free velocity for each agent can be obtained analogously to [22], as depicted in Fig. 4: A half-plane H_{AB} is placed tangentially onto the convex hull of $V_{AB,\infty}\cap C^n\text{-CO}^\tau_{AB}$ so that the minimal *signed* distance to $\mathbf{v}_{AB}$ is maximized. The vector connecting the convex Hull with $\mathbf{v}_{AB}$ is again denoted by $\mathbf{u}_{AB}$. In each interacting agent A's absolute velocity space, linear half-plane constraints are then applied at $\mathbf{v}_A+\frac{\mathbf{u}_{AB}}{2}$, in direction of $R(\frac{\pi}{2})\mathbf{u}_{AB}$, where $R(\cdot)$ denotes the 2-D rotation matrix. For the duration of a control cycle, each agent then selects the feasible velocity $\mathbf{v}_{(\cdot),\infty}$ that minimizes the Euclidean distance to its preferred velocity.

Let us conclude this section with the following remarks: The C^n-CO approach allows us to apply the RVO concept *in closed form* to the class of robot models equivalent to System 1. And while such models rarely depend on solution continuity beyond $\mathcal{C}^1$, more general (non)linear systems often do. In Section IV, we extend C^n-CO's applicability to a large class of (non)linear robotic systems that can be brought into an equivalent (higher order) representation in normal form via feedback linearization. This ensures the practical relevance of C^n-CO with $n>1$.

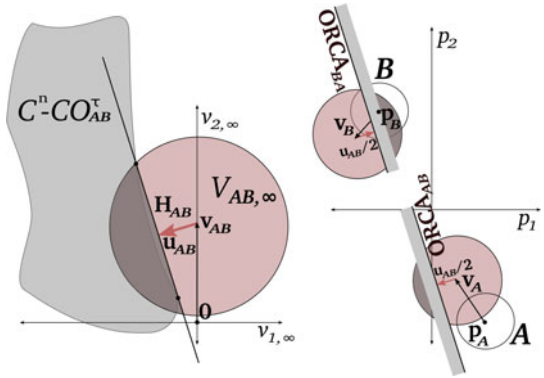


Fig. 4. Reciprocal collision avoidance using C^n-CO (described for agent A). (Left) Set of reachable relative velocities $V_{AB,\infty}$ is colored in red, C^n-CO^τ_{AB} in light gray, and the convex hull of their intersection $V_{AB,\infty}\cap C^n$-CO^τ_{AB} in dark gray. $\mathbf{u}_{AB}$ is computed by intersecting $V_{AB,\infty}$ with a half-plane H_{AB}. The half-plane H_{AB} is constructed via tangential placement onto the convex hull of $V_{AB,\infty}\cap C^n\text{-CO}^\tau_{AB}$ through the point on the convex hull that maximizes the *signed* minimal distance $d_{\min}$ between $\mathbf{v}_{AB}$ and H_{AB}. $\mathbf{u}_{AB}$ is a vector perpendicular to H_{AB} with $\|\mathbf{u}_{AB}\|_2=d_{\min}$. (Right) $\mathbf{u}_{AB}$ is used to construct linear half-plane constraints in each agent's absolute velocity space analogical to the ORCA approach: Under the assumption that interaction proceeds *fairly*, $ORCA_{AB}$ is characterized by the point–direction pair $(\mathbf{v}_A+\frac{\mathbf{u}_{AB}}{2}, R(\frac{\pi}{2})\mathbf{u}_{AB})$, where $R(\cdot)$ denotes the 2-D rotation matrix. *A* then selects the velocity $\mathbf{v}_{A,\infty}$ in accordance with all constraints that minimize the Euclidean distance to its preferred velocity $\mathbf{v}_A^{\text{pref}}$. This procedure is repeated at control frequency.

IV. C^n-CO Applied to Feedback Linearizable Systems

In the previous section, we described a reciprocal collision avoidance framework for linear integrators of arbitrary order and thus continuity. In this section, the framework is extended via feedback linearization to a broader range of both linear and nonlinear agent models. The procedure is as follows: If found to be conformant, the system under consideration is first feedback linearized (see Section IV-A). Second, a collision-free reference trajectory of sufficient continuity (see Sections IV-C and D) is computed following the method of Section III for the equivalent system in linear normal form. Third, input and state constraints that stem from the nonlinear system may be added to the optimization problem, as described in Section IV-E. The ensuing optimization results in the selection of $\mathbf{v}_\infty$. Finally, the nonlinear system is controlled onto the selected high-order reference trajectory given by (1), using the input transformation obtained from the feedback linearization step.

A. Feedback Linearization

Feedback linearization is a well-known control technique that allows us to modify a controlled system's inputs so that its input–output (and in some cases input-state) behavior assumes

the linear normal form of System 1. Such systems can thus be controlled via well-understood linear control theory. Conversely, the C^n-CO framework—which we developed for linear systems above—can be naturally extended to feedback linearizable systems. The technique is applicable to systems of the following form.

System 2 (General Feedback Linearizable System):

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{v}$$
$$\boldsymbol{y} = h(\boldsymbol{x})$$

with $\mathbf{x}$ being the state vector, and $\mathbf{y}$ being the output of the system. The aim is then to derive an appropriate control input transformation

$$\mathbf{v} = a(\mathbf{x}) + b(\mathbf{x})\mathbf{u}$$

that creates a linear mapping between $\mathbf{u}$ and $\mathbf{y}$ [28]. We distinguish between static and dynamic feedback linearization. Static feedback linearization applies to systems that are fully controllable about their reference point already. Examples include generic linear systems. In that case, the resulting equivalent system in normal form is of the same order as the original system. If, on the other hand, the reference point is not fully controllable (such as $\mathbf{p}$ in Systems 4 and 5), the state space needs to be augmented with dynamic compensators, resulting in a higher order equivalent system in normal form [29].

It turns out that many robotic system models of practical relevance are feedback linearizable but we will, in this paper, restrict our attention to general linear systems, kinematic unicycles, and bicycle models for two main reasons. On one hand, they are interesting in view of their practical relevance—omnidrives, diff-drive robots, and cars may all be modeled this way. On the other hand, this choice allows us to show that C^n-CO subsumes both ORCA-DD [20], which was developed for unicycles, and LQR-obstacles [30], [31], which has (independently from our study) been developed for general linear systems.

B. Selected Agent Models

The most general class of time-invariant linear systems may be expressed as follows.

System 3 (General Linear System):

$$\dot{\mathbf{x}} = F\,\boldsymbol{x} + G\,\boldsymbol{v}$$

where F is an $(n+1)\times(n+1)$ matrix, and G is of size $(n+1)\times m$, with m the number of inputs.

Kinematic models of unicycles and bicycles, on the other hand, belong to the class of nonholonomic, and, thus, nonlinear systems. Such systems have in common that they are subject to at least one nonholonomic constraint. The kinematic model of a unicycle is given by System 4.

System 4 (Unicycle):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos\delta \\ \sin\delta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2$$

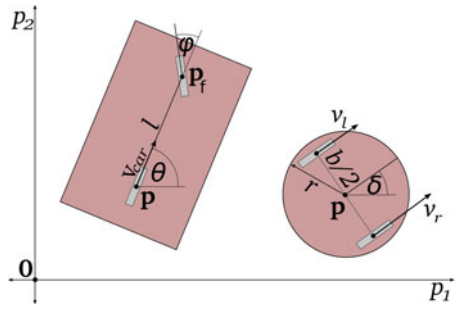


Fig. 5. Nonholonomic system models. (Left) Bicycle model of a car. It is steered via throttle v_{car} and steering wheel velocity $\dot{\varphi}$. The reference point is usually selected at the center of the rear-wheel axle, but another common choice includes the center of the front-wheel axle ($\mathbf{p}_{\mathrm{f}}$). In the latter case, v_{car} is applied at the front wheels as well. (Right) Unicycle model of a differential-drive robot that is steered via left and right wheel motors. Forward and rotational velocity result in $v_{\mathrm{dd}} = (v_{\mathrm{r}} + v_{\mathrm{l}})/2, \omega_{\mathrm{dd}} = (v_{\mathrm{r}} - v_{\mathrm{l}})/b$.

with $\mathbf{v} = (v_1, v_2)^T$, the forward and rotational speed of the platform, respectively. The relevant variables are introduced in Fig. 5.

In contrast with the unicycle, bicycles are commonly commanded via the throttle $v_1 = \| \dot{\mathbf{p}} \|$ and the front wheel's steering velocity $v_2 = \dot{\varphi}$. The kinematic model of a bicycle with rear-wheel drive is represented as follows.

System 5 (Rear-Wheel Driven Bicycle):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\varphi/l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2.$$

A structural similarity between Systems 4 and 5 becomes apparent by moving the bicycle's reference point to the front axle center [29]. The resulting system, a bicycle with front-wheel drive, may be thought of as a unicycle with a trailer attached [29]; hence, $\delta = \theta + \varphi$. The derivation of the aforementioned robot models are described in [6] and [29].

C. Linearization via Static State Feedback

We noted that static feedback linearization is applicable to systems whose reference point is fully controllable. To this end, let us first consider general linear systems (characterized by System 3). Such models are said to be fully controllable iff $[F, FG, \ldots, F^{n+1}G]$ has full rank. Under the assumption of full controllability, System 3 then admits feedback linearization into System 1, thanks to its conformance with System 2. Interestingly, the LQR obstacle method [30] steers precisely such general linear systems onto straight line reference trajectories using state feedback control. Hence, C^n-CO completely subsumes the LQR obstacle method on the basis of System 2 and feedback linearization.

We now proceed with the nonlinear systems, which are introduced in Section IV-B. For Systems 4 and 5, the reference point $\mathbf{p}$ is not fully controllable. Following [29], any reference point $\hat{\mathbf{p}}$ to the front of the vehicle axles is fully controllable, however, and linearization via static feedback becomes feasible. For symmetry reasons, the new reference point $\hat{\mathbf{p}}$ is usually chosen

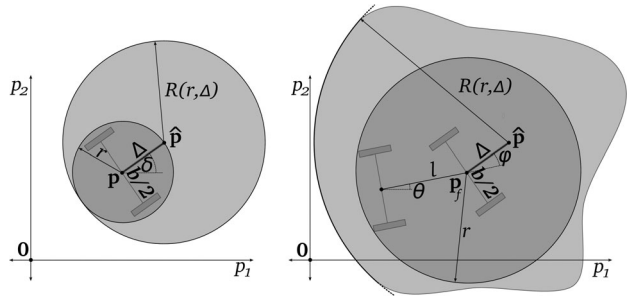


Fig. 6. Consequence of feedback linearization on the effective robot radius. Dynamic feedback linearization allows us to retain the reference point $\mathbf{p}$ (and $\mathbf{p}_\mathrm{f}$, respectively) allowing us to formulate C^n-CO with the true robot radius r (i.e., the out-circle of the robot, which is colored in dark gray). Static feedback linearization necessitates a redefined reference point $\hat{\mathbf{p}}$, which increases the effective radius to $R(r,\Delta)$, which is shown in light gray. (Left) Unicycle. (Right) Bicycle.

as follows (depicted in Fig. 6):

$$\hat{\mathbf{p}} = \mathbf{p} + [\cos\delta, \sin\delta]^T \Delta$$

where $\Delta > 0$ defines the maneuverability of the vehicle. For the robot of System 4, the following relations hold:

$$\begin{aligned}\dot{\hat{\mathbf{p}}} &= \mathbf{u}\\ v_1 &= \mathbf{u}\cdot[\cos\delta, \sin\delta]^T\\ v_2 &= \mathbf{u}\cdot[-\sin\delta, \cos\delta]^T/\Delta.\end{aligned}$$

System 4 is then controllable about $\hat{\mathbf{p}}$ with holonomic velocity inputs given by $\mathbf{u}$ and C^n-CO may be applied at this point. Due to the internal dynamics of the new system and the circular shape requirement of VO-based methods, an increased radius, which is given by $R = r + \Delta$, is required. This leads to the situation depicted on the left side of Fig. 6 and where C^0-CO is equivalent to the ORCA-DD approach presented in [20]. Note that the parameter Δ is variable so that it may be decreased in dense situations—at the expense of lateral maneuverability.

As established in Section IV-B, a front-wheel driven bicycle is mathematically equivalent to a unicycle (System 4) pulling a trailer with relative orientation φ. Analogous to the aforementioned case, such a system is then controllable about $\hat{\mathbf{p}}$, and C^n-CO can be applied at this point. Nevertheless, in this case, the increased radius R, which is centered at $\hat{\mathbf{p}}$, may become prohibitively large, as depicted on the right side of Fig. 6. Note that this method, likewise, extends to vehicles with trailers—at the cost of the corresponding enlargement. By contrast, dynamic feedback linearization does not require artificial inflation of the vehicle radius, as even partially controllable reference points $\mathbf{p}$ can be employed in that case.

D. Linearization via Dynamic State Feedback

Dynamic feedback linearization overcomes the noncontrollability of $\mathbf{p}$ via so-called dynamic extension. Here, we apply it again to Systems 4 and 5. The degree of the extension is dependent on the number of states of the original system.

For the kinematic unicycle, its three states need to be appended by a single dynamic compensator only. The resulting chain of integrators is of second order, as derived in [29], and hence, C^1-CO applies

$$\begin{aligned}\ddot{\mathbf{p}} &= \mathbf{u}\\ v_1 &= \|\dot{\mathbf{p}}\|\\ v_2 &= v_1 \frac{\dot{p_1}\ddot{p_2} - \dot{p_2}\ddot{p_1}}{(\dot{p_1}\dot{p_1} + \dot{p_2}\dot{p_2})^{3/2}}.\end{aligned}$$

After optimization, the inputs to the original system $(v_1, v_2)^T$ can be deduced from the holonomic solution trajectory (and its derivatives). Recalling System 4, this implies $\delta \in \mathcal{C}^0$, and $\dot{\mathbf{p}} \in \mathcal{C}^0$. If continuity in the actuators and in the curvature of the unicycle's trajectory is desired (kinodynamic case), C^2-CO becomes necessary.

Dynamic feedback linearization thus allows us to retain the vehicle reference point $\mathbf{p}$, and avoids artificial inflation of the robot radius r that the shifting of $\mathbf{p}$ would entail. This benefit comes at the price of a higher order model.

For the rear-wheel driven bicycle (System 5), the same procedure results in the following chain of integrators and input transformations [29]:

$$\begin{aligned}\dddot{\mathbf{p}} &= \mathbf{u}\\ v_1 &= \|\dot{\mathbf{p}}\|\\ v_2 &= l v_1 \frac{(\dddot{p}_2\dot{p}_1 - \dddot{p}_1\dot{p}_2)v_1 - 3(\ddot{p}_2\dot{p}_1 - \ddot{p}_1\dot{p}_2)(\dot{p}_1\ddot{p}_1 + \dot{p}_2\ddot{p}_2)}{v_1^6 + l^2(\ddot{p}_2\dot{p}_1 - \ddot{p}_1\dot{p}_2)^2}.\end{aligned}$$

This result can be motivated as follows: For the kinematic version of the bicycle, the linear and steering velocities must at least verify $v_1, v_2 \in \mathcal{C}^{-1}$. Recalling System 5, this implies $\varphi \in \mathcal{C}^0$, $\theta \in \mathcal{C}^1$, and $\dot{\mathbf{p}} \in \mathcal{C}^1$, which necessitates C^2-CO. If the continuity requirements are increased to continuity in steering rate $\varphi \in \mathcal{C}^1$ (kinodynamic model), C^3-CO is needed.

We note that the aforementioned formulations do not inherently respect system constraints. Handling of actuator and state constraints is described next.

E. Nonlinear Constraint Satisfaction

In Section III-E, we established that for decoupled linear systems, the set of attainable velocities $V_{AB,\infty}(\mathbf{v}_{AB,0})$ is a square (or, more conveniently, its in-circle) centered at $\mathbf{p}_0$. For feedback-linearized systems, the situation is more complex, as the actuator and state constraints of the underlying system may impose nonlinear requirements on the attainable shapes of the solution trajectory $\mathbf{p}(t)$. While, in principle, the individual sets of constraint-compliant velocities $V_{A,\infty}$ can be computed using agent A's system model, in general, the transformed constraints do not admit a closed form solution. Instead, we may (exhaustively) sample from $V_{A,\infty}$, and for each resulting trajectory numerically verify conformance with nonlinear system constraints. This operation may be performed offline and stored in a *constraint mask* that represents the set of reachable velocities (for discretized initial conditions), as proposed in [23].

F. Reciprocal Collision Avoidance

Reciprocal collision avoidance for feedback linearized systems proceeds similarly to Section III-E, except that the set $V_{AB,\infty}$ of reachable relative velocities at $t \to \infty$ is obtained by taking the Minkowski sum of the individual agents' constraint masks computed previously. Consequently, the set $V_{AB,\infty}$ will in general no longer be convex. By selecting its *out-circle* $\tilde{V}_{AB,\infty}$ for the purpose of constructing the half-plane H_{AB} only, we ensure that feasible actions are never inhibited. On the other hand, we need to ensure that unreachable velocities will later be discarded again. The half-plane constraint H_{AB} is then computed tangent to the convex hull of $\tilde{V}_{AB,\infty} \cap C^n\text{-CO}^{\tau}_{AB}$ so that its signed minimal distance $d_{\min}$ to $\mathbf{v}_{AB}$ is maximized. The vector $\mathbf{u}_{AB}$ is formed perpendicular to H_{AB} with $\|\mathbf{u}_{AB}\|_2 = d_{\min}$ and is used to construct ORCA half-plane constraints in each agent's absolute velocity space. Finally, each agent *A* selects the collision-free velocity within $V_{A,\infty}$ that is closest to its preferred velocity $\mathbf{v}_A^{\text{pref}}$. Due to the potential nonconvexity of the set V_A, this optimization may be efficiently solved by a wavefront propagation algorithm that proceeds from $\mathbf{v}_A^{\text{pref}}$ outward [23], [32].

V. C^n-CO Employed in Partial Motion Planning

The practice of employing the Euclidean distance to $\mathbf{v}^{\text{pref}}$ as optimization criterion has rendered relative velocity-based methods inherently local—and thus prone to local minima in the vicinity of concave static obstacles. C^n-CO's direct incorporation of an agent model, on the other hand, facilitates the adoption of a global optimization criterion. The key insight lies in the existence of a bijection between elements of $V_{(\cdot),\infty}$ and trajectory segments $\mathbf{p}_{(\cdot)}(t)$: A sampling in $V_{(\cdot),\infty}$ forms a highly expressive trajectory set locally that approaches the various straight line reference velocity vectors at $t \to \infty$ (see Fig. 2). By considering these system-compliant segments up to a time horizon τ only (linked to the control parameter $\lambda_{(\cdot)}$), a tree of depth one is obtained. This tree may be searched and extended online using a graph search method compatible with the system models underlying C^n-CO, such as ARAE* [10] or RRT* [9]. During search, sampled edges at depth one that are found to be in violation of fair interaction principles (i.e., segments that do not conform with the imposed linear constraints) are discarded. The same applies to segments (at any search depth) in collision with static obstacles. By running these components in a loop at control frequency, an interactive model-predictive partial motion planning framework is obtained. This is demonstrated in Section VI-C.

VI. Experiments

The experimental section is divided into four parts. In Section VI-A, we provide experimental evidence supporting the adequacy of key design choices made during the development of Section III. In particular, we quantitatively specify the advantage of modeling a curved higher order collision cone (in accordance with robot dynamics) over the much simpler instance of the original VO approach—even if a linear half-plane constraint is used to partition sets of fair velocities for performance reasons. Tied to this, we provide evidence that in absence of (inefficient) centralized optimization, it seems difficult to obtain significantly better results by using a partition scheme different from said fast linear approach. In Section VI-B, we perform simulated multiagent navigation experiments. Specifically, we assess C^1-CO against inflated C^0-CO (representing a generic controller tracking ORCA) for a large set of environmental and agent conditions. In Section VI-C, we demonstrate that C^n-CO can be seamlessly integrated into a global navigation framework. For Sections VI-B and C, we have included a supplementary color MPEG-4 file that visualizes typical sample runs. This material is available at http://ieeexplore.ieee.org. Finally, in Section VI-D, the runtime performance and scalability of the presented methods are analyzed.

A. Assessment of Design Choices

The exposition of this paper is largely based on the assumption that C^n-CO's collision cone with $n > 0$ is at most as large as the corresponding omnidirectionally inflated C^0-CO. This is quantitatively verified via a comparison between the C^1-CO method and an ε-inflated C^0-CO method tracked by a generic controller, where ε accounts for tracking errors. The latter method essentially corresponds to NH-ORCA [21]. However, for better comparability, we substitute its controller with the one employed in C^1-CO.

We furthermore observed that C^n-CO's curved shape complicates the construction of a fair partition scheme. In Section III, we opted for a well-known linear partition for computational performance reasons. The potential loss in solution quality stemming from this decision, with respect to a nonlinear partition scheme, thus, also requires analysis. The size of the obtained partition $|V|$ or the minimal distance between $\mathbf{v}_{AB,0}$ and the boundary of the partition (i.e., $d_{\min}$) serve as performance criteria.

1) Linear Versus Nonlinear Constraints: Let us recall from Fig. 4 that the C^n-CO method selects a half-plane H_{AB} such that the *signed* shortest distance $d_{\min}$ between the convex hull of $V_{AB,\infty} \cap C^n\text{-CO}^{\tau}_{AB}$ and $\mathbf{v}_{AB,0}$ is maximized. This is equivalent to the maximization of the radius $r_{\min}$ of a disk centered at $\mathbf{v}_{AB,0}$. Hence, we observe that our method favors allotment of fair velocities close to the current absolute velocities over maximizing the overall size of the set. For higher order models and the associated—characteristically nonconvex—collision cones, partitions more efficient than those arising from linear constraints could be constructed. However, many such (e.g., deconvolution-based) methods are not well posed inside a distributed framework and, thus, require centralized coordination. By retaining the same optimization criteria as for the ORCA algorithm, for comparison purposes, a (computationally expensive) decentralized nonlinear iterative wavefront expansion method can be designed to partition curved collision cones instead. This method incrementally expands a wavefront from $\mathbf{v}_A$ and $\mathbf{v}_B$ simultaneously. The covered velocities are then appropriated to the sets of fair velocities $\hat{V}_{A,\infty}$ and $\hat{V}_{B,\infty}$, respectively. Expansion in a particular direction is interrupted if the Minkowski sum of the two growing

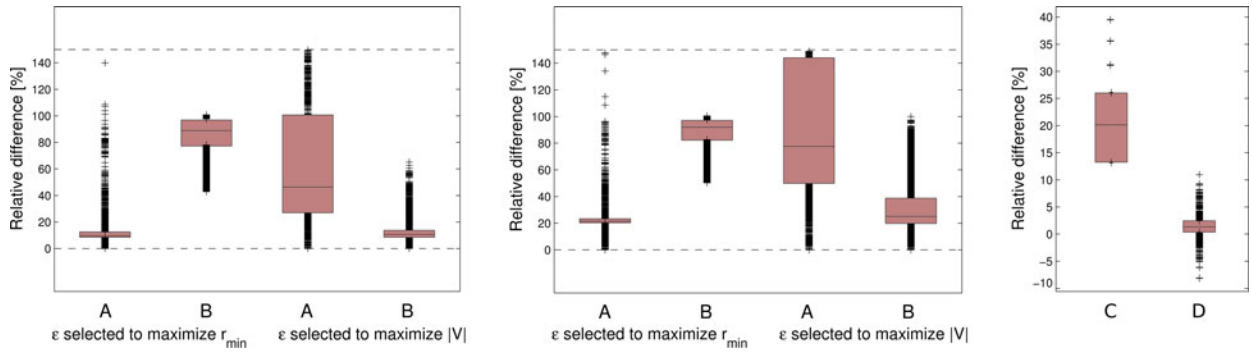


Fig. 7. Assessment of design choices. (Left) Comparison of $r_{\min}$ and $|\hat{V}_{AB,\infty}|$ between C^1-CO and εC^0-CO for control parameters $\lambda_0 = 0.50\,\mathrm{s}^{-1}$. Displayed are the relative values *A*: $\frac{r_{\min}^{C^1\text{-CO}} - r_{\min}^{\varepsilon C^0\text{-CO}}}{r_{\min}^{C^1\text{-CO}}}$ and *B*: $\frac{|V|_{\min}^{C^1\text{-CO}} - |V|_{\min}^{\varepsilon C^0\text{-CO}}}{|V|_{\min}^{C^1\text{-CO}}}$, respectively. (Center) Same scenario for $\lambda_0 = 0.25\,\mathrm{s}^{-1}$. (Right) Comparison between the C^1-CO method performing a wavefront-based nonlinear partition of $\hat{V}_{AB,\infty}$ and the C^1-CO method based on a linear partition. Control parameters are selected as $\lambda_0 = 0.25\,\mathrm{s}^{-1}$. Displayed are the relative values *C*: $\frac{r_{\min}^{\mathrm{wave}} - r_{\min}^{\mathrm{ORCA}}}{r_{\min}^{\mathrm{wave}}}$ and *D*: $\frac{|V|_{\min}^{\mathrm{wave}} - |V|_{\min}^{\mathrm{ORCA}}}{|V|_{\min}^{\mathrm{wave}}}$. We observe that the nonlinear method's improvement in minimal radius $r_{\min}$ is insignificant (i.e., below the 0.05 m discretization) in almost all cases (27 490 out of 28 802). Displayed are the significant cases only.

sets $\hat{V}_{AB,\infty} = \hat{V}_{A,\infty} \oplus -\hat{V}_{B,\infty}$ intersects the nonlinear collision cone C^n-CO$_{AB}$. The method terminates once the boundary cannot be expanded any further. In practice, we implemented this approach via a discrete scheme using polar coordinates centered around the agents' current velocities with 0.05 m/s radial and $\pi/50$ rad/s tangential discretization.

The experimental setup consists of two identical circular interacting agents A and B of radius 1.0 m. Agent *A* is placed at the origin. Successively, a discrete set of speed levels $s_A = \{-0.5 : 0.25 : 0.5\}$ m/s is applied in direction $\theta_A = \{\pi/4 : \pi/4 : \pi\}$. Agent *B* is successively placed on a rectangular grid with coordinates $\mathbf{p}_B = \{(2.5 : 0.5 : 5, 2.5 : 0.5 : 5)\}$ m, and subjected to velocities from the same set as that of agent *A*. For both agents, the eigenvalues of the closed-loop system are either placed at $\lambda_{0_1} = 0.5\,\mathrm{s}^{-1}$ or at $\lambda_{0_2} = 0.25\,\mathrm{s}^{-1}$. These choices correspond to maximal acceleration constraints $\mathbf{a}_{\max_1} = 1.0\,\mathrm{m/s}^2$ and $\mathbf{a}_{\max_2} = 0.5\,\mathrm{m/s}^2$, respectively, occurring for 90° turns at $\mathbf{v}_{\max} = 1$ m/s. The $\mathbf{a}_{\max}$-values roughly represent upper limits for road-borne vehicles on dry and wet roads, respectively, after which tires start to slip. We draw statistics over the whole set of configurations.

As expected, the more involved wavefront expansion generally produces fair sets of larger $r_{\min}$—the largest differences are observed in cases where the collision cone is strongly curved. However, in the overall context, the gains are mostly insignificant. In fact, only 1312 out of 28 802 samples resulted in an increase in $r_{\min}$ larger than the discretization employed (0.05 m/s). The right side of Fig. 7 visualizes the distribution of the improvement of $r_{\min}$ over the 1312 significant experiments. In the light of these findings, the choice favoring the much faster linear constraints is justified.

2) Partition size of C^1-CO Versus εC^0-C0: Prior to this study, most RVO-based methods required an inflation of the effective agent radii to compensate for the systems' inability to follow C^0-continuous trajectories. Hence, we assess the advantage of employing C^n-CO (subject to linear constraints, as justified previously) for higher order robot models directly over the simpler specification of an *ε-inflated* linear collision cone.

Let us consider the same experimental setup as in Section VI-A1 composed of two agents A and B. We examine two *maximization* criteria. The first one pertains to the minimal radius of a disk centered around $\mathbf{v}_{AB,0}$ (as in the ORCA method). The second one is complementary and concerns the size of the fair velocity set $|\hat{V}_{AB,\infty}|$.

For the inflated C^0-CO approach, the optimal result represents a tradeoff between increasing ε (which increases the size of the set of trackable velocities) and decreasing inflation (which represses the half-plane boundary constraint). While in theory the inflation value ε may be adaptively selected, practical aspects further constrain this choice. In fact, pairwise interactions between multiple agents may produce differing optimal values (according to one of the mentioned criteria). Nonetheless, in our comparison limited to a single interaction, we densely sample over ε values and select the one that maximizes the appropriate metric. The results should thus be interpreted as a best possible outcome for the εC^0-CO approach, which may not fully materialize in multiagent navigation. As the left side and center of Fig. 7 show for $\lambda_0 \in \{0.25,\ 0.5\}\,\mathrm{s}^{-1}$, however, the higher order solution is even under these conditions significantly preferable, producing both larger $r_{\min}$ and $|\hat{V}_{AB,\infty}|$ *at the same time*. On the other hand, the results also suggest that the performance of both approaches converges for agents admitting even faster dynamics, i.e., $\lambda_0 > 0.5\,\mathrm{s}^{-1}$. This corresponds to agents for which a first-order model is approximately applicable.

Together, the aforementioned two results justify our proposed approach in general and the choice for linear constraints fitted onto the nonlinear collision cone in particular.

B. Assessment of Local Multiagent Navigation Performance

While the aforementioned test cases indicate that an agent employing C^1-CO is generally allowed more avoidance options than one using inflated C^0-CO, these theoretical benefits do not

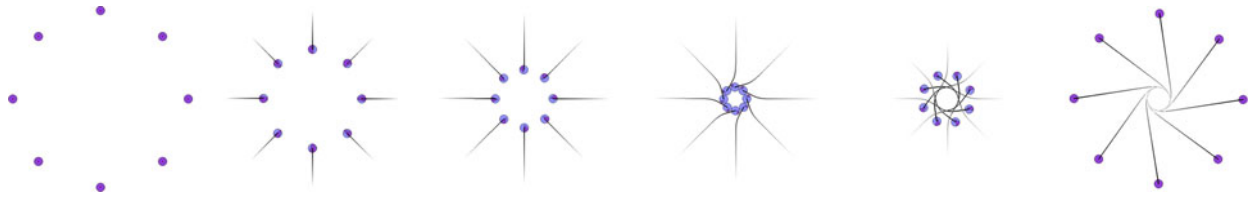


Fig. 8. Local multiagent navigation. Circle swap with eight agents employing C^1-CO. Eigenvalues are set at $\lambda_0 = 0.5\,\mathrm{s}^{-1}$. Once agents have agreed on an avoidance *topology*, the physical constraints inherent to the agents preserve stability. By employing a navigation scheme with a slight avoidance preference to a common side, efficient avoidance patterns reminiscent of roundabouts emerge.

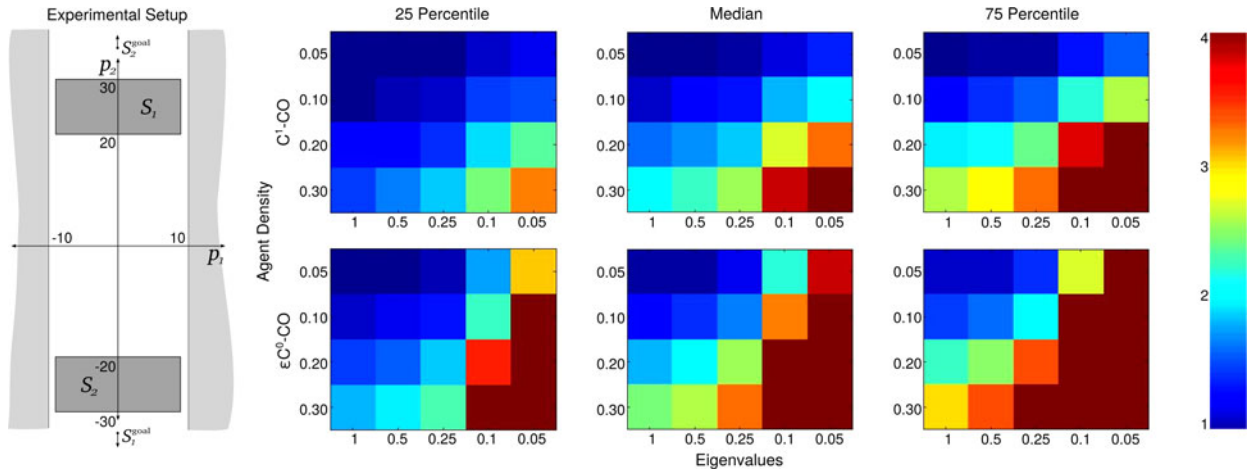


Fig. 9. Local multiagent navigation. (Left) Experimental setup consisting of two sets of agents $\mathcal{S}_1$ and $\mathcal{S}_2$ that operate within a bounded scene, as described in Section VI-B2. (Right) Displayed are the 25, 50, and 75 percentile convergence times for said randomized experiments. For each parameter combination, 20 experiments have been performed. (Right—Top row) C^1-CO. (Right—Bottom row) $\varepsilon\, C^0$-CO. The results demonstrate that the advantage of employing a higher order C^n-CO over inflated C^0-CO materializes stronger for slow agent dynamics. Conversely, for fast agent dynamics, i.e., $\lambda_0 \in [0.5, \infty)\,\mathrm{s}^{-1}$, the performance of both approaches converges. The same conclusions apply for densely versus sparsely populated scenarios. A color video of this experiment is available at http://ieeexplore.ieee.org.

necessarily need to materialize in typical planning scenarios. To this end, we compare the C^1-CO approach versus an inflated C^0-CO approach on a series of test cases.

1) Local Navigation in a Circle Swap: Let us consider a circular setup, composed out of eight agents employing C^1-CO with $\lambda_0 = 0.5\,\mathrm{s}^{-1}$. Agents need to exchange their position diametrically. The circle swap represents a hard scenario for decentralized local navigation approaches. In contrast with ordinary ORCA, local minima (occurring in the cluttered center) are much more difficult to resolve (via, e.g., random walks) due to the constrained kinematics of the platforms considered. Conversely, once agents agree on an avoidance *topology*, the physical constraints preserve stability. By employing a scheme that prefers avoidance to one side (similar to humans), efficient patterns reminiscent of roundabouts emerge (see Fig. 8).

2) Local Navigation in Counterflow: Next, let us consider a scenario composed of two sets of agents. The two sets are bounded by rectangles of $20\,\mathrm{m} \times 10\,\mathrm{m}$ size, and are separated by 40 m (displayed on the left side of Fig. 9). Agents are circular with $r = 1\,\mathrm{m}$ and obey $\mathbf{v}_{\max} = 1.0\,\mathrm{m/s}$. Sensor limits are modeled as uniformly distributed noise $\sigma \in [-0.05, 0.05]\,\mathrm{m}$ on position estimates. Goal locations are placed in direction of the opposing set of agents at $\pm\infty$ m. For each agent, the time is elapsed between its start position and 50 m toward the goal. Consequently, the first encounters between agents of different sets, on average, occur after 25 m. Simulations are aborted after $4\,t_{\min}$, where $t_{\min}$ is the minimal time required to traverse the scene in absence of other agents. Experiments are performed over increasingly more densely populated scenes (between 5% and 30% of the rectangular areas covered by agents), and decreasing agent dynamic capabilities $\lambda_0 \in \{0.05, 0.1, 0.25, 0.5, 1.0\}\,\mathrm{s}^{-1}$. For each parameter combination, statistics are drawn over 20 simulations. Fig. 9 displays the 25, 50, and 75 percentile convergence times for the described randomized experiments. Note that deadlocks usually resolve spontaneously due to perception noise. The results confirm our earlier finding that the advantage of employing a higher order C^n-CO over inflated C^0-CO appears strongest for slow agent dynamics. Conversely, for fast agent dynamics, $\lambda_0 \in (0.5, \infty]\,\mathrm{s}^{-1}$, the performance of both approaches converges. The same conclusions apply for densely versus sparsely populated scenarios.

C. Demonstration of C^n-CO in Partial Motion Planning

Natural environments usually consist of static obstacles in addition to dynamic agents. These obstacles may attain arbitrary shapes and are often concave, which renders VO-based methods in their basic form prone to local minima. In Section V,

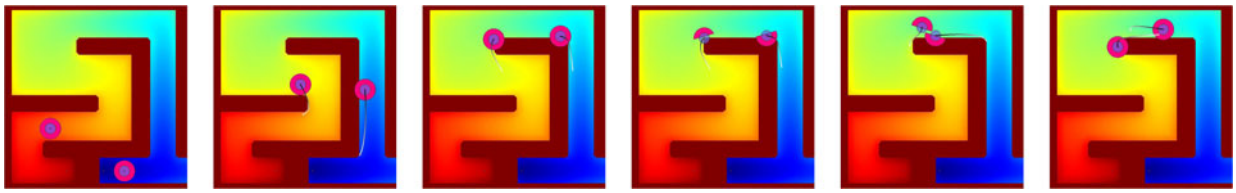


Fig. 10. Global multiagent navigation. Both agents employ the heuristic graph search strategy, which is described in Section VI-C: First, fair partitions in absolute velocity space are computed between interacting agents using the C^1-CO method with a time horizon of $\tau = 4$ s (in pink). Then, the associated trajectory segments are collision checked using the static map. At the same time, a Field Dijkstra approach computes the cost-to-go in absence of dynamic objects (displayed as a color gradient). Finally, the sampled $\mathcal{C}^1$-continuous trajectory admitting the lowest combined cost is selected for execution during the next control cycle. A color video of this experiment is available at http://ieeexplore.ieee.org.

we have proposed a framework, which allows for the seamless integration of C^n-CO into global graph search. Fig. 10 demonstrates this framework in simulation. It visualizes snapshots at several time instances along a scenario, where two agents need to exchange their position within a maze. Global convergence to their respective goals is obtained despite the presence of concave static obstacles. Both agents are modeled according to C^1-CO with $\lambda_0 = 0.5\,\mathrm{s}^{-1}$. The time horizon after which interaction effects are neglected has been selected at $\tau = 4$ s (corresponding to t^{ext}; see Example 3). $\mathbf{v}_\infty$ is sampled exhaustively using a discretization of 0.1 m/s and $v_{\max} = 1.0$ m/s. This results in a tree composed of at most 441 segments at depth one. Tree segments in violation with linear constraints stemming from interaction or in collision with the static map are discarded immediately. The remaining segments are penalized according to the cells they traverse. Graph search terminates at a tree depth of one, at which point it connects to a field Dijkstra heuristic function (displayed via the color gradient). The edge segment with the lowest combined cost is selected as solution and executed for the duration of one control cycle. As a result, the solution approaches global optimality in purely static environments. In dynamic environments, on the other hand, no *formal* optimality guarantees can be provided due to the decentralized nature of C^n-CO.

D. Runtime Analysis

Robotic collision avoidance approaches by definition need to be executable online. It is thus of interest to analyze the performance and scalability of the methods presented in Sections III–IV. C^n-CO is composed of five main steps, including the setting of a preferred velocity $\mathbf{v}^{\mathrm{pref}}$, the determination of neighboring agents within interaction distance, the computation of a collision cone (CC) for each neighboring agent, the solution of the associated linear program (LP), the updating, and, thus, forward simulation of agents' states. While the first two and the fourth step are identical for the considered methods (but may depend on the overall number of agents in the scenario), steps three and five induce additional computational complexity for $n > 0$: The computation of a curved collision cone and the extraction of its associated linear constraint takes up to 10^3 times longer than in the linear case. This is owed to the required sampling of the cone's boundary, followed by a convex hull computation. Similarly, for $n > 0$, the update of agents' states needs to be performed via forward simulation of an agent model, which also produces an (albeit minor) computational overhead. Table I provides maximal runtime results for each of these steps for C^0-CO (i.e., ORCA [19]), ε C^0-CO (i.e., NH-ORCA [21]), as well as C^1-CO. The provided timings are in seconds per agent (considering $n \in \{1, 10, 100\}$ neighbors, if applicable) and have been compiled on a single core of an Intel Core2 Duo running at 2.53 GHz. Overall, C^1-CO is two orders of magnitude slower than C^0-CO but remains online capable.

TABLE I
RUNTIME COMPARISON BETWEEN C^0-CO, ε C^0-CO, AND C^1-CO IN LOCAL MULTIAGENT SCENARIOS

Step Descr.	**# Neigh.**	**C^0-CO**	**ε C^0-CO**	**C^1-CO**
Set $\mathbf{v}^{\mathrm{pref}}$	N/A	$3{\cdot}10^{-7}$ s	$3{\cdot}10^{-7}$ s	$3{\cdot}10^{-7}$ s
Neigh. search	1	$2{\cdot}10^{-8}$ s	$2{\cdot}10^{-8}$ s	$2{\cdot}10^{-8}$ s
	10	$3{\cdot}10^{-7}$ s	$3{\cdot}10^{-7}$ s	$3{\cdot}10^{-7}$ s
	100	$5{\cdot}10^{-6}$ s	$5{\cdot}10^{-6}$ s	$5{\cdot}10^{-6}$ s
Compute CC	N/A	$4{\cdot}10^{-7}$ s	$4{\cdot}10^{-7}$ s	$6{\cdot}10^{-4}$ s
Solve LP	1	$2{\cdot}10^{-8}$ s	$2{\cdot}10^{-8}$ s	$2{\cdot}10^{-8}$ s
	10	$6{\cdot}10^{-8}$ s	$6{\cdot}10^{-8}$ s	$6{\cdot}10^{-8}$ s
	100	$4{\cdot}10^{-7}$ s	$4{\cdot}10^{-7}$ s	$4{\cdot}10^{-7}$ s
Update state	N/A	$1{\cdot}10^{-6}$ s	$2{\cdot}10^{-6}$ s	$2{\cdot}10^{-6}$ s

VII. DISCUSSION

The presented control obstacle of order n (C^n-CO) generalizes the VO concept and allows *holonomic* robots to move along $\mathcal{C}^n$-continuous trajectories, while performing reciprocal collision avoidance without the necessity of artificially inflating their shape. Furthermore, through feedback linearization, we demonstrated that a large class of general linear as well as nonlinear robot models can be brought into a form suitable for C^n-CO. The two methods covered—static and dynamic feedback linearization—both pose benefits and shortcomings.

As shown in Section IV-C, static feedback linearization is applicable to fully controllable points only. This implies that the method is not suitable for nonholonomic systems directly, but requires a redefinition of the reference point away from the wheel axles. Since the relative velocity paradigm can only handle circular robots, this usually results in an increased robot radius. On the other hand, the order of the statically feedback linearized system is retained, and thus, the same controller (with identical parameters) can be chosen for all interacting robots, independent of their kinematic characteristics.

Conversely, as shown in Section IV-D, dynamic feedback linearization can often be directly applied to nonholonomic reference points. This prevents an artificial inflation in robot radius. However, in heterogeneous environments, it may result in robots following control laws of different order of continuity, which do not directly admit a common treatment under the presented framework. Equation (5) shows, however, that the trajectories of higher order models are composed of their lower order counterparts and ever less relevant additional terms. These additional terms represent rapidly decaying elements (as depicted in Fig. 2). In the presence of several robot types, using C^n-COs of different degrees of continuity, an additional inflation on the robot shape (equal to the decaying parts) may hence be considered to compensate for the difference between the expected and the actual trajectory of other vehicles. This suggests the combination of C^n-CO with the method presented in [23] for certain multiplatform scenarios.

In a series of experiments, we demonstrated the gain in navigation performance C^n-CO provides over inflated C^0-CO. Benefits have been shown to be largest for slow agent dynamics (i.e., $\lambda_0 \leq 0.25\,\mathrm{s}^{-1}$) or moderately to densely populated scenes (i.e., obstacle and agent density $\rho \geq 0.2$)—and it is under these circumstances where the employment of a higher order C^n-CO formulation pays off most.

VIII. Conclusion and Future Work

In this paper, we introduced the order n continuous control obstacle (C^n-CO), which allows holonomic agents to move along collision-free $\mathcal{C}^n$-continuous trajectories, while performing local reciprocal collision avoidance among a homogeneous set of feedback-controlled agents. While holonomic robots rarely require trajectories of continuity higher than $\mathcal{C}^1$, we showed that a large class of robot models can be brought into the normal form suitable for C^n-CO via feedback linearization, thus extending the method's applicability to general linear and many nonlinear agents including differentially driven robots, and Ackerman vehicles. We illustrated that any degree of continuity can be achieved at these models' *actuators* via appropriate selection of C^n-CO's order of continuity.

We further established that C^n-CO generalizes RVO and several recent VO extensions: First, the approach of [20] has been shown to be equivalent to applying C^0-CO after static feedback linearization in an off-axis point—as discussed in Section IV-C. We further extended this method to cars and cars pulling trailers. Second, the approach of [22] represents a specific instance of C^1-CO. Third, the LQR-obstacle method [30] corresponds to C^n-CO for general *linear* systems where eigenvalues are determined according to the solution of an algebraic Riccati equation.

In summary, C^n-CO represents an extension of the RVO concept toward solution trajectories of higher continuity, including actuator continuity for nonholonomic agents, and a unification of several recent extensions to the basic RVO framework. Given the promising properties of C^n-CO, we are currently investigating approaches to solve the main remaining limitations of the relative velocity concept, as laid out in Section I-C.

References

[1] H. J. Sussmann and J. C. Willems, "300 years of optimal control: From the brachystochrone to the maximum principle," *IEEE Control Syst*., vol. 17, no. 3, pp. 32–44, Jun. 1997.

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 2001.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," in *Autonomous Mobile Robots: Perception, Mapping, and Navigation*. Washington, DC, USA: IEEE Comput. Soc., 1991.

[4] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," in *Proc. IEEE Int. Conf. Robot. Autom*., 2002, pp. 968–975.

[5] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robot*., vol. 26, no. 1, pp. 308–333, 2009.

[6] M. Rufli and R. Siegwart, "On the design of deformable input and state-lattice graphs," in *Proc. IEEE Int. Conf. Robot. Autom*., 2010, pp. 3071–3077.

[7] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res*., vol. 20, no. 5, pp. 378–400, 2001.

[8] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: Bradford, 2010.

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

[10] J. P. Gonzalez and M. Likhachev, "Search-based planning with provable suboptimality bounds for continuous state spaces," in *Proc. 4th Annu. Symp. Combinat. Search*, 2011, pp. 60–67.

[11] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," in *Proc. 19th Nat. Conf. Innovat. Appl. Artif. Intell*., 2007, pp. 1752–1759.

[12] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE-RSJ Int. Conf. Intell. Robots Syst*., 2010, pp. 797–803.

[13] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Image and animation display with multiple robots," *Int. J. Robot. Res*., vol. 31, pp. 753–773, 2012.

[14] S. Hoogendoorn and P. H. L. Bovy, "Simulation of pedestrian flows by optimal control and differential games," *Opt. Control Appl. Methods*, vol. 24, no. 3, pp. 153–172, 2003.

[15] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *Proc. IEEE Int. Conf. Robot. Autom*., 2012, pp. 477–483.

[16] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol*., vol. 17, no. 5, pp. 1105–1118, Sep. 2009.

[17] N. E. Du Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," in *Proc. IEEE Int. Conf. Robot. Autom*., 2010, pp. 966–973.

[18] J. van den Berg, M. C. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom*., 2008, pp. 1928–1935.

[19] J. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. Int. Symp. Robot. Res*., 2009, pp. 3–19.

[20] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Smooth and collision-free navigation for multiple robots under differential-drive constraints," in *Proc. IEEE-RSJ Int. Conf. Intell. Robots Syst*., 2010, pp. 4584–4589.

[21] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Proc. Int. Symp. Distrib. Auton. Robot. Syst*., 2010, pp. 203–216.

[22] 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.

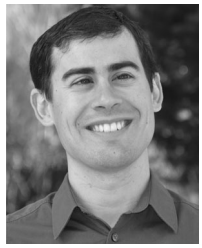
[23] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *Proc. IEEE Int. Conf. Robot. Autom*., 2012, pp. 360–366.

[24] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res*., vol. 17, pp. 760–772, 1998.

[25] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *Proc. IEEE-RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 5573–5578.
[26] J. Minguez and L. Montano, "Extending reactive collision avoidance methods to consider any vehicle shape and the kinematics and dynamic constraints," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 367–381, Apr. 2009.
[27] E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, 2nd ed. Boca Raton, FL, USA: CRC, 2002.
[28] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice–Hall, 2002.
[29] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control* (Lecture Notes in Control and Information Sciences), vol. 229, J.-P. Laumond, Ed. Berlin, Germany: Springer-Verlag, 1998, pp. 171–253.
[30] J. van den Berg, D. Wilkie, S. J. Guy, M. Niethammer, and D. Manocha, "LQG-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 346–353.
[31] D. F. Bareiss and J. van den Berg, "Reciprocal collision avoidance for quadrotor helicopters using LQR-obstacles," in *Proc. AAAI Workshop Multi-Agent Pathfind.*, 2012, pp. 6–7.
[32] R. Philippsen, "A light formulation of the E* interpolated path replanner," Autonomous Systems Lab, Ecole Polytech. Féd. Lausanne, Lausanne, Switzerland, Tech. Rep. LSA-REPORT-2006-001, 2006.

Martin Rufli (M'11) received the B.Sc. and M.Sc. degrees in mechanical engineering and the Ph.D. degree in robotics, all from ETH Zurich, Zurich, Switzerland, in 2006, 2008, and 2012, respectively.

Since 2012, he has been a Postdoctoral Research Fellow and Lecturer with the Autonomous Systems Lab, ETH Zurich. His current research interests include the design of novel algorithms for robotic navigation with the aim to bring robots out of factory lines and into our everyday lives. The main focus in this regard concerns system-based online methods and algorithms that model robot–robot interaction.



Javier Alonso-Mora (S'11) received the Diploma degree in mathematics in 2008 and in engineering in 2010, both from the Universitat Politècnica de Catalunya (UPC) Barcelona, Barcelona, Spain, and the M.Sc. degree in robotics, systems, and control from ETH Zurich, Zurich, Switzerland in 2010. He is currently working toward the Ph.D. degree with ETH Zurich and Disney Research Zurich.

His current research focuses on multirobot navigation, with an emphasis on real-time algorithms that model interagent decision making. He is also interested in developing intuitive methods for human–robot swarm interaction.

Mr. Alonso-Mora received the Willi-Studer Award for the best Master Diploma in robotics. He has received excellence scholarships from UPC Barcelona, the Spanish Research Council, and the Swiss Government.

Roland Siegwart (F'08) was born in 1959. He received the M.Sc. and Ph.D. degrees in mechanical engineering from ETH Zurich, Zurich, Switzerland.

He has been a Full Professor for Autonomous Systems with ETH Zurich, Zurich, Switzerland, since 2006 and the Vice President Research and Corporate Relations since 2010. From 1996 to 2006, he was an Associate and later a Full Professor for Autonomous Microsystems and Robotics with the Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He leads a research group of around 30 people working on several aspects of robotics.

Mr. Siegwart is a member of the Swiss Academy of Engineering Sciences and the Officer of the International Federation of Robotics Research. He has served as the Vice President for Technical Activities from 2004 to 2005, a Distinguished Lecturer from 2006 to 2007, and an AdCom member from 2007 to 2009 of the IEEE Robotics and Automation Society.