Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles

Jur van den Berg   Jamie Snape   Stephen J. Guy   Dinesh Manocha

Abstract— We present an approach for collision avoidance for mobile robots that takes into account acceleration constraints. We discuss both the case of navigating a single robot among moving obstacles, and the case of multiple robots reciprocally avoiding collisions with each other while navigating a common workspace. Inspired by the concept of velocity obstacles [3], we introduce the *acceleration-velocity obstacle* (AVO) to let a robot avoid collisions with moving obstacles while obeying acceleration constraints. AVO characterizes the set of new velocities the robot can safely reach and adopt using proportional control of the acceleration. We extend this concept to *reciprocal* collision avoidance for multi-robot settings, by letting each robot take half of the responsibility of avoiding pairwise collisions. Our formulation guarantees collision-free navigation even as the robots act independently and simultaneously, without coordination. Our approach is designed for holonomic robots, but can also be applied to kinematically constrained non-holonomic robots such as cars. We have implemented our approach, and we show simulation results in challenging environments with large numbers of robots and obstacles.

I. INTRODUCTION

Collision avoidance is a fundamental problem in robotics. The problem can generally be defined in the context of an autonomous mobile robot navigating in an environment with obstacles and/or other moving entities, where the robot employs a continuous cycle of sensing and acting. In each time step, an action for the robot must be computed based on local observations of the environment, such that the robot stays free of collisions with the moving obstacles and the other robots, and progresses towards a goal.

Many works in robotics have addressed the problem of collision avoidance with moving obstacles [5], [9], [14]. Most approaches predict where the moving obstacles might be in the future by extrapolating their observed velocities, and let the robot avoid collisions accordingly. Velocity obstacles [3] formalize this principle by characterizing the set of velocities for the robot that result in a collision at some future time. Continually selecting a velocity outside of this set will then guarantee collision-free navigation for the robot. A major shortcoming, however, is that it requires the robot to change its velocity instantaneously, which is not possible in many cases due to constraints on the acceleration of the robot.

In this paper, we introduce the concept of *acceleration-velocity obstacles* (AVO) for guaranteed collision-avoidance with moving obstacles that accounts for constraints on the acceleration of the robot. AVO lets the robot accelerate towards
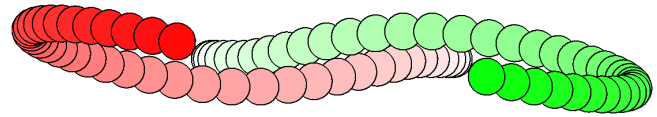
Fig. 1. Two robots with acceleration constraints avoid collisions with each other while exchanging positions, and pass their target location with high speed. Newer frames are on top of older frames and darker.

a new velocity using proportional control, i.e. the applied acceleration is continually proportional to the difference between the new velocity and the current velocity. If the robot would collide at any point in time along its trajectory, the new velocity is considered forbidden. Continually selecting a new velocity outside the forbidden set guarantees collision-free motions that obey the acceleration constraints.

Our work is motivated by robots moving at high speeds, such as unmanned aerial vehicles, airplanes, car-like robots, etc., for which acceleration constraints are particularly significant. We present our approach for a simplified robot model: each robot is a disc moving in a two-dimensional workspace, able to perfectly infer the shape, position and velocity of obstacles and other robots in the environment, and capable of accelerating in any direction up to a maximum. This model applies directly to holonomic robots, and we will show that it applies to a large class of non-holonomic robots with kinodynamic constraints (e.g. cars, airplanes, differential drives), as long as their speed is nonzero.

Approaches that assume that obstacles are non-responding and moving at a constant velocity are insufficient for multi-robot settings, where the robot encounters other robots that also make decisions based on their surroundings: Considering them as moving obstacles overlooks the fact that they react to the robot in the same way as the robot reacts to them, and inherently causes undesirable oscillations in the motion of the robots [10], [18]. We present a *reciprocal collision avoidance* approach based on acceleration-velocity obstacles that specifically accounts for the reactive nature of the other robots. Each robot independently and simultaneously computes its actions, without coordination, by reducing the problem to a 2-D linear program.

We experimented with our approach on several simulation scenarios containing up to a thousand robots. As each robot navigates independently, we can fully parallellize the computation of the actions for each robot and report fast running times. Furthermore, our experiments show that our approach achieves convincing motions that are smooth, collision-free, and obey the acceleration constraints (see Fig. 1).

The remainder of this paper is organized as follows.

In Section II we discuss background and related work. In Section III we introduce our concept of *acceleration-velocity obstacles* for navigation of a single robot among moving obstacles, and in Section IV we extend this concept to reciprocal collision avoidance for multiple robots. We show how our concept is applied to kinematically constrained robots in Section V. We present simulation results in Section VI, and conclude in Section VII.

II. BACKGROUND AND RELATED WORK

Our work builds mainly on the concept of velocity obstacles [3], which has been employed successfully in practice for collision avoidance with moving obstacles [15]. However, it requires the robot to change its velocity instantaneously, which may not be possible for physical robots. Many attempts have been made to incorporate acceleration constraints into the formalism. One approach is to limit the set of valid velocities to those that can be reached within the next time step of the sensing-acting cycle [3]. However, as the time step can be arbitrarily small, only a small set of valid velocities is available, which diminishes look-ahead. A better strategy, therefore, is to clamp down a selected velocity to one that obeys the acceleration constraint [8], [10]. However, collision avoidance may then no longer be guaranteed.

A few variants of velocity obstacles have been proposed that address acceleration constraints specifically. The works of [2], [7] are related to *inevitable collision states* [4], and define velocities as forbidden if no escape maneuver exists given the acceleration constraints to avoid collisions. However, this does not imply that a velocity that is not forbidden is guaranteed to be collision-free, as what may be a safe escape maneuver with respect to one obstacle may be a collision course with respect to another obstacle. In [13], velocity obstacles are defined for the special case of a car-like robot that controls its speed and turning radius, and incorporates second-order constraints on the speed.

Other related concepts include non-linear velocity obstacles [16] and generalized velocity obstacles [20]. The former accounts for higher-order motion of obstacles, while the latter defines "control input obstacles" for kinematically constrained robots. The probabilistic velocity obstacle of [6] addresses uncertainty in the future trajectory of obstacles.

Existing approaches that address *reciprocal* collision avoidance include [1], [17], [10], [18]. The approach of [19] guarantees collision avoidance for multiple robots. However, none of these approaches deal with acceleration constraints. We will combine the approach of [19] with acceleration-velocity obstacles to guarantee collision-free navigation of multiple robots subject to acceleration constraints.

A. Notation

We will use the following notational convention in this paper: Scalars x are set in lower case italics, vectors $\mathbf{x}$ in lower case bold, and sets (of vectors) X in upper case italics. Further, we denote the measure of a set X by $|X|$, the length of a vector $\mathbf{x}$ by $\|\mathbf{x}\|$, and use the following notation for an
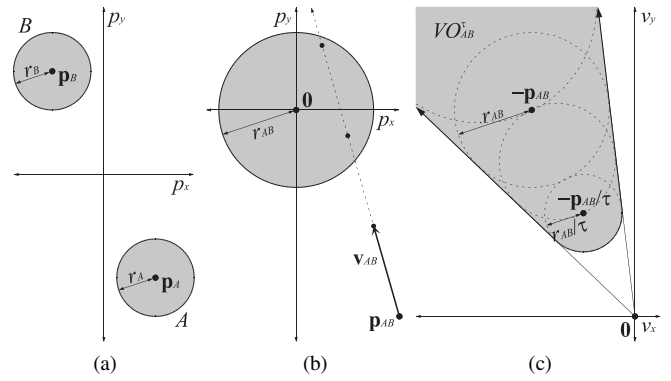

Fig. 2. (a) A configuration of robot A and moving obstacle B. (b) The evolution of their relative position over time when the relative velocity $\mathbf{v}_{AB}$ is applied. A and B collide as soon as they are closer to each other than the sum of their radii (disc). (c) The velocity obstacle VO_{AB}^{τ} with time horizon $\tau = 2$ shown as a union of discs.

open disc of radius r centered at $\mathbf{p}$, scalar-set multiplication, and the Minkowski sum of two sets, respectively:

$$D(\mathbf{p}, r) = \{\mathbf{q} \mid \|\mathbf{q} - \mathbf{p}\| < r\} \tag{1}$$
$$aX = \{a\mathbf{x} \mid \mathbf{x} \in X\} \tag{2}$$
$$X \oplus Y = \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in X, \mathbf{y} \in Y\}. \tag{3}$$

III. COLLISION AVOIDANCE WITH MOVING OBSTACLES

In this section we discuss how a robot A subject to acceleration constraint $\|\mathbf{a}_A\| < a_A^{\max}$ can avoid collisions with moving obstacles. We will first review the existing concept of *velocity obstacles* [3] (Section III-A), and then introduce our new concept of *acceleration-velocity obstacles* (Section III-B). In Section III-C we discuss how acceleration-velocity obstacles are used to navigate a robot among multiple moving obstacles.

A. Velocity Obstacles

Let A be a robot and B a moving obstacle with current positions $\mathbf{p}_A$ and $\mathbf{p}_B$, and radii r_A and r_B, respectively. The velocity obstacle [3] can be used to select a velocity for A such that collisions with B are avoided, assuming that this velocity can be adopted instantaneously.

Definition 1 The *velocity obstacle* VO_{AB}^{τ} (read: the velocity obstacle for A induced by B for 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.

More formally, let $\mathbf{p}_{AB} = \mathbf{p}_A - \mathbf{p}_B$ be the current relative position of A with respect to B and let $r_{AB} = r_A + r_B$ be the sum of their radii, then A and B will collide at time t if (see Fig. 2(b)):

$$\|\mathbf{p}_{AB} + \mathbf{v}_{AB}t\| < r_{AB}. \tag{4}$$

Dividing both sides by t, and rearranging gives:

$$\|\mathbf{v}_{AB} - (-\frac{\mathbf{p}_{AB}}{t})\| < \frac{r_{AB}}{t}, \tag{5}$$

which defines the disc of all relative velocities $\mathbf{v}_{AB}$ that let A and B collide at time t (see Eq. (1)). As such, we can

define the velocity obstacle VO_{AB}^{τ} as a union of discs (see Fig. 2(c)):

$$VO_{AB}^{\tau} = \bigcup_{0 < t \leq \tau} D(-\frac{\mathbf{p}_{AB}}{t}, \frac{r_{AB}}{t}). \quad (6)$$

Geometrically, it can be interpreted as a truncated cone with its apex at the origin (in velocity space) and its sides tangent to the Minkowski sum $B \oplus -A$ of the obstacle and the robot.

The definition of the velocity obstacle implies that if obstacle B is moving at a *constant* velocity $\mathbf{v}_B$, and robot A chooses its velocity $\mathbf{v}_A$ outside $VO_{AB}^{\tau} \oplus \{\mathbf{v}_B\}$ (i.e. the velocity obstacle translated by $\mathbf{v}_B$), robot A is guaranteed not to collide with B before time τ if A and B maintain their velocities for at least τ time. Acceleration constraints may prohibit A from instantaneously adopting velocity $\mathbf{v}_A$, however, in which case collision avoidance is not guaranteed.

To overcome this issue, we may let the robot choose an acceleration rather than a velocity. In a similar way as velocity obstacles, we can define the *acceleration obstacle* AO_{AB}^{τ} as the set of all relative accelerations that will result in a collision before time τ. In contrast to velocity obstacles, however, acceleration obstacles are not suitable for navigation in most practical cases. While it is reasonable to assume that the obstacle and robot maintain their velocities for a while (at least approximately), accelerations tend to change frequently and it is unreasonable to assume that either robot A or obstacle B will maintain an observed acceleration for an extended amount of time τ. The value of τ signifies the look-ahead, and reducing it would lead to unsafe navigation. To address this flaw, we will define *acceleration-velocity obstacles*.

B. Acceleration-Velocity Obstacles

Instead of letting the robot A choose an acceleration, we want to let the robot choose a velocity, as is the case with velocity obstacles. The acceleration constraint $\|\mathbf{a}_A\| < a_A^{\max}$ may prohibit adopting a *new* velocity $\mathbf{v}'_A$ instantaneously, but we can apply valid accelerations to *arrive* at this new velocity at some point in time. To ensure that the robot smoothly arrives at its new velocity, we use proportional control of the acceleration. That is, the acceleration $\mathbf{a}_A(t)$ applied at time t is proportional to the difference between the new velocity $\mathbf{v}'_A$ and the velocity $\mathbf{v}_A(t)$ at time t:

$$\mathbf{a}_A(t) = \dot{\mathbf{v}}_A(t) = \frac{\mathbf{v}'_A - \mathbf{v}_A(t)}{\delta}, \quad (7)$$

where δ is a control parameter whose dimension is time. Solving this differential equation for $\mathbf{v}_A(t)$ gives:

$$\mathbf{v}_A(t) = \mathbf{v}'_A - e^{-t/\delta}(\mathbf{v}'_A - \mathbf{v}_A), \quad (8)$$

where $\mathbf{v}_A = \mathbf{v}_A(0)$ is the current velocity of A. Integrating $\mathbf{v}_A(t)$ gives the trajectory $\mathbf{p}_A(t)$ of the robot:

$$\mathbf{p}_A(t) = \mathbf{p}_A + t\mathbf{v}'_A + \delta(e^{-t/\delta} - 1)(\mathbf{v}'_A - \mathbf{v}_A), \quad (9)$$

where $\mathbf{p}_A = \mathbf{p}_A(0)$ is the current position of A.

If an obstacle B follows a similar trajectory $\mathbf{p}_B(t)$ to arrive at a new velocity $\mathbf{v}'_B$ with the same control parameter
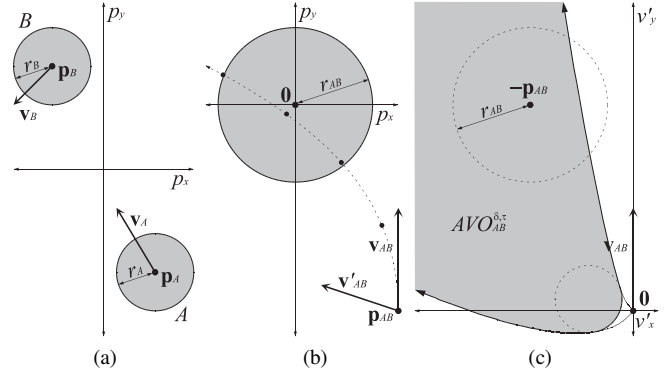

Fig. 3. (a) A configuration of robot A and moving obstacle B. (b) The evolution of their relative position over time when the new relative velocity $\mathbf{v}'_{AB}$ is approached by proportional control of the relative acceleration ($\delta = 2$). A and B collide as soon as they are closer to each other than the sum of their radii (disc). (c) The acceleration-velocity obstacle $AVO_{AB}^{\delta,\tau}$ for control parameter $\delta = 2$ and time horizon $\tau = 4$.

δ, then the *relative* position $\mathbf{p}_{AB}(t) = \mathbf{p}_A(t) - \mathbf{p}_B(t)$, the relative velocity $\mathbf{v}_{AB}(t) = \mathbf{v}_A(t) - \mathbf{v}_B(t)$, and the relative acceleration $\mathbf{a}_{AB}(t) = \mathbf{a}_A(t) - \mathbf{a}_B(t)$ of robot A and obstacle B are also described by Equations (9), (8), and (7), respectively, where $\mathbf{p}_{AB} = \mathbf{p}_A - \mathbf{p}_B$ is the current relative position, $\mathbf{v}_{AB} = \mathbf{v}_A - \mathbf{v}_B$ is the current relative velocity, and $\mathbf{v}'_{AB} = \mathbf{v}'_A - \mathbf{v}'_B$ is the new relative velocity. It is in terms of these that we define the acceleration-velocity obstacle.

Definition 2 The *acceleration-velocity obstacle* $AVO_{AB}^{\delta,\tau}$ (read: the acceleration-velocity obstacle for A induced by B for time horizon τ and control parameter δ) is the set of all new relative velocities $\mathbf{v}'_{AB}$ of A with respect to B that will result in a collision between A and B before time τ, where proportional control of the acceleration (with parameter δ) is used by both A and B to reach the new relative velocity $\mathbf{v}'_{AB}$ from the current relative velocity $\mathbf{v}_{AB}$.

More formally, let $r_{AB} = r_A + r_B$ be the combined radius of A and B, then A and B collide at time t if $\|\mathbf{p}_{AB}(t)\| < r_{AB}$ (see Fig. 3(b)). By Eq. (9), that is:

$$\|\mathbf{p}_{AB} + t\mathbf{v}'_{AB} + \delta(e^{-\frac{t}{\delta}} - 1)(\mathbf{v}'_{AB} - \mathbf{v}_{AB})\| < r_{AB}. \quad (10)$$

Dividing both sides by $t + \delta(e^{-t/\delta} - 1)$ and rearranging gives:

$$\|\mathbf{v}'_{AB} - \frac{\delta(e^{-\frac{t}{\delta}} - 1)\mathbf{v}_{AB} - \mathbf{p}_{AB}}{t + \delta(e^{-\frac{t}{\delta}} - 1)}\| < \frac{r_{AB}}{t + \delta(e^{-\frac{t}{\delta}} - 1)}, \quad (11)$$

which defines the disc of all new relative velocities $\mathbf{v}'_{AB}$ that let A and B collide at time t. Hence, the acceleration-velocity obstacle $AVO_{AB}^{\delta,\tau}$ is a union of discs (see Fig. 3(c)):

$$AVO_{AB}^{\delta,\tau} = \bigcup_{0 < t \leq \tau} D(\frac{\delta(e^{-\frac{t}{\delta}} - 1)\mathbf{v}_{AB} - \mathbf{p}_{AB}}{t + \delta(e^{-\frac{t}{\delta}} - 1)}, \frac{r_{AB}}{t + \delta(e^{-\frac{t}{\delta}} - 1)}). \quad (12)$$

A closed form expression can be obtained for the boundary of the acceleration-velocity obstacle (see Appendix).

The definition of the acceleration-velocity obstacle implies that if obstacle B is moving with a *constant* velocity $\mathbf{v}_B$ (i.e. $\mathbf{v}'_B = \mathbf{v}_B$), and robot A chooses its new velocity $\mathbf{v}'_A$ outside $AVO_{AB}^{\delta,\tau} \oplus \{\mathbf{v}_B\}$, robot A is guaranteed not to collide with
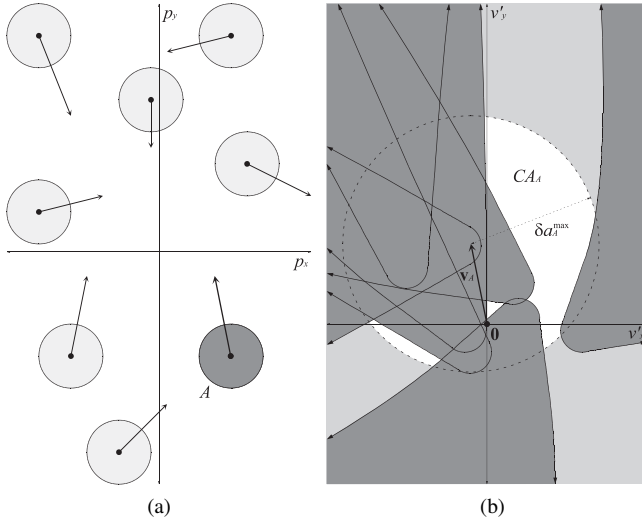
Fig. 4. (a) A configuration of robot A amidst moving obstacles. The arrows indicate their current velocities. (b) The acceleration-velocity obstacles induced by the obstacles for $\delta = 2$ and $\tau = 4$ (dark grey). The white region is the set CA_A of safe new velocities $\mathbf{v}'_A$ for A.

B before time τ if A uses control parameter δ to arrive at its new velocity $\mathbf{v}'_A$. Alternatively, if B applies an acceleration $\mathbf{a}_B$ and also uses control parameter δ to arrive at a new velocity $\mathbf{v}'_B$, it follows from Eq. (7) that $\mathbf{v}'_B = \mathbf{v}_B + \delta\mathbf{a}_B$. Then, robot A should choose its new velocity $\mathbf{v}'_A$ outside $AVO^{\delta,\tau}_{AB} \oplus \{\mathbf{v}'_B\}$ to be guaranteed not to collide with B before time τ.

C. Navigation among Multiple Moving Obstacles

Acceleration-velocity obstacles can be used to navigate a robot among multiple moving obstacles as follows. The robot A performs a continuous cycle of sensing and acting with a (small) time step Δt. In each iteration, the robot determines its *preferred velocity* $\mathbf{v}^{\text{pref}}_A$,[1] and senses the radius r_B, the current position $\mathbf{p}_B$ and the current velocity $\mathbf{v}_B$ of each obstacle B. The robot either assumes that the obstacle moves with a constant velocity, i.e. $\mathbf{v}'_B = \mathbf{v}_B$, or alternatively that $\mathbf{v}'_B = \mathbf{v}_B + \delta\mathbf{a}_B$ if an acceleration $\mathbf{a}_B$ is also observed. Based on this information, the robot computes the acceleration-velocity obstacle $AVO^{\delta,\tau}_{AB} \oplus \{\mathbf{v}'_B\}$ induced by B. The set CA_A of new velocities for A that avoid collisions with all obstacles is the complement of the union of the acceleration-velocity obstacles. In addition, the robot must obey the acceleration constraint $\|\mathbf{a}_A\| < a^{\max}_A$, which by Eq. (7) means that $\|\mathbf{v}'_A - \mathbf{v}_A\| < \delta a^{\max}_A$. So (see Fig. 4):

$$CA_A = D(\mathbf{v}_A, \delta a^{\max}_A) \setminus \bigcup_B (AVO^{\delta,\tau}_{AB} \oplus \{\mathbf{v}'_B\}). \quad (13)$$

Next, the robot selects the velocity in CA_A that is closest to its preferred velocity $\mathbf{v}^{\text{pref}}_A$ as its new velocity $\mathbf{v}'_A$:

$$\mathbf{v}'_A = \underset{\mathbf{v} \in CA_A}{\arg\min} \|\mathbf{v} - \mathbf{v}^{\text{pref}}_A\|. \quad (14)$$

---

[1]The preferred velocity $\mathbf{v}^{\text{pref}}_A$ is the velocity A would have taken if there were no moving obstacles, for instance the vector in the direction of its goal with a magnitude equal to A's preferred speed.

It can be shown that this velocity is either (i) the preferred velocity itself, (ii) an intersection point of the boundaries of two acceleration-velocity obstacles, or (iii) an orthogonal projection of the preferred velocity on the boundary of an acceleration-velocity obstacle [8]. To determine $\mathbf{v}'_A$, all these points are computed and checked whether they are in CA_A. The one closest to the preferred velocity is selected as the new velocity for A.

Finally, the robot applies the acceleration $\mathbf{a}_A = (\mathbf{v}'_A - \mathbf{v}_A)/\delta$, and the sensing-acting cycle repeats after Δt time.

IV. RECIPROCAL COLLISION AVOIDANCE

Above, we have shown how a robot can avoid moving obstacles with the assumption that their future trajectories can be estimated solely by extrapolating current information about the obstacles' motion. However, in many cases the "moving obstacles" are not passively moving entities, but actually other robots that also make decisions based on their surroundings. Simply considering them as moving obstacles will lead to *oscillations* if the other entities too consider all other robots as moving obstacles [10], [18]. Therefore, the reactive nature of the other robots must be specifically taken into account in order to guarantee that collisions are avoided.

This problem is referred to as *reciprocal collision avoidance*, and is addressed in this section. An approach to reciprocal collision avoidance based on velocity obstacles was introduced in [19]. Here, we present a similar approach based on acceleration-velocity obstacles taking into account acceleration constraints on the robots. We assume that the robots act independently using the same collision-avoidance strategy (with the same value of δ), and are able to estimate or observe the relevant physical properties of other robots.

A. Sets of Reciprocal Collision Avoidance

Let us consider the configuration of Fig. 3(a), but in this case assume that both A and B are decision-making robots that both attempt to avoid collisions with control parameter δ and time horizon τ. To obey the acceleration constraints, A will choose its new velocity $\mathbf{v}'_A$ in $D(\mathbf{v}_A, \delta a^{\max}_A)$ and B will choose its new velocity $\mathbf{v}'_B$ in $D(\mathbf{v}_B, \delta a^{\max}_B)$. Hence, the new *relative* velocity $\mathbf{v}'_{AB} = \mathbf{v}'_A - \mathbf{v}'_B$ of A with respect to B will be in $D(\mathbf{v}_{AB}, \delta a^{\max}_{AB})$, where $\mathbf{v}_{AB} = \mathbf{v}_A - \mathbf{v}_B$ is the current relative velocity and $a^{\max}_{AB} = a^{\max}_A + a^{\max}_B$ is the sum of the maximum accelerations of A and B. In addition, to avoid collisions between A and B before time τ, the new relative velocity $\mathbf{v}'_{AB}$ should be outside $AVO^{\delta,\tau}_{AB}$. We denote the set of safe new relative velocities V'_{AB} (see Fig. 5(a)):

$$V'_{AB} = D(\mathbf{v}_{AB}, \delta a^{\max}_{AB}) \setminus AVO^{\delta,\tau}_{AB}. \quad (15)$$

Since A and B select their new velocities independently and simultaneously, A does not know what new velocity B selects, and vice versa. So, neither robot can guarantee that $\mathbf{v}'_{AB} \in V'_{AB}$. However, if we apportion sets of potential new velocities $V'_A \subset D(\mathbf{v}_A, \delta a^{\max}_A)$ and $V'_B \subset D(\mathbf{v}_B, \delta a^{\max}_B)$ to A and B, respectively, such that $V'_A \oplus -V'_B \subset V'_{AB}$, then collisions are guaranteed to be avoided:

Lemma 3 If $V'_A \oplus -V'_B \subset V'_{AB}$, then
$$\mathbf{v}'_A \in V'_A \wedge \mathbf{v}'_B \in V'_B \implies \mathbf{v}'_{AB} \in V'_{AB}.$$

There are infinitely many pairs of sets $V'_A \subset D(\mathbf{v}_A, \delta a_A^{\max})$ and $V'_B \subset D(\mathbf{v}_B, \delta a_B^{\max})$ such that $V'_A \oplus -V'_B \subset V'_{AB}$. The question is which pair is "fair" (i.e. both robots share the responsibility of avoiding collisions equally) and "maximal" (i.e. both sets contain a maximal amount of safe new velocities). Also, both A and B should be able to independently determine its set of safe new velocities without coordination among them.

B. Finding a Convex Subset of V'_{AB}

For a *convex* set X holds that $\alpha X \oplus (1-\alpha)X = X$ for any $0 < \alpha < 1$. We could use this rule to determine V'_A and V'_B if V'_{AB} were convex. However, in general it is not (see Fig. 5(a)). Therefore, we construct a (large) convex subset $\hat{V}'_{AB} \subset V'_{AB}$ by intersecting V'_{AB} with a *halfplane* H:

$$\hat{V}'_{AB} = V'_{AB} \cap H \tag{16}$$

Without making any assumptions on the properties of the shape of $AVO_{AB}^{\tau,\delta}$, we want to find H such that $\hat{V}'_{AB} = V'_{AB} \cap H$ is convex and $|\hat{V}'_{AB}|$ is maximal. We construct H as follows.

Let CH be the *convex hull* of $D(\mathbf{v}_{AB}, \delta a_{AB}^{\max}) \cap AVO_{AB}^{\tau,\delta}$, and let $\mathbf{q}$ be the closest point on the boundary of CH to $\mathbf{v}_{AB}$. Then, we define H as the halfplane that is tangent to CH at $\mathbf{q}$ (see Fig. 5(a)). It follows from the construction of H that $H \cap (D(\mathbf{v}_{AB}, \delta a_{AB}^{\max}) \cap AVO_{AB}^{\tau,\delta}) = \emptyset$. Therefore, $\hat{V}'_{AB} = V'_{AB} \cap H = D(\mathbf{v}_{AB}, \delta a_{AB}^{\max}) \cap H$, which ensures that $\hat{V}'_{AB}$ is convex and a subset of V'_{AB}. Also, H was chosen such that $|\hat{V}'_{AB}|$ is maximal among all possible halfplanes for which $H \cap (D(\mathbf{v}_{AB}, \delta a_{AB}^{\max}) \cap AVO_{AB}^{\tau,\delta}) = \emptyset$.

C. Determining Sets of Reciprocal Collision Avoidance

Given the convex set $\hat{V}'_{AB}$ of safe new relative velocities, we apportion sets V'_A and V'_B to A and B, respectively, such that $V'_A \subset D(\mathbf{v}_A, \delta a_A^{\max})$, $V'_B \subset D(\mathbf{v}_B, \delta a_B^{\max})$ and $V'_A \oplus -V'_B = \hat{V}'_{AB}$. We do this as follows:

$$V'_A = \alpha_{AB}(\hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\}) \oplus \{\mathbf{v}_A\} \tag{17}$$
$$V'_B = -(1-\alpha_{AB})(\hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\}) \oplus \{\mathbf{v}_B\}, \tag{18}$$

for parameter $0 < \alpha_{AB} < 1$ that determines A's share of the responsibility of avoiding collisions between A and B. We can prove that $V'_A \oplus -V'_B = \hat{V}'_{AB}$ as follows:

$$
\begin{aligned}
V'_A \oplus -V'_B &= \alpha_{AB}(\hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\}) \oplus \{\mathbf{v}_A\} \oplus \\
&\quad (1-\alpha_{AB})(\hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\}) \oplus \{-\mathbf{v}_B\} \\
&= (\hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\}) \oplus \{\mathbf{v}_A\} \oplus \{-\mathbf{v}_B\} \\
&= \hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\} \oplus \{\mathbf{v}_{AB}\} = \hat{V}'_{AB}, \quad (19)
\end{aligned}
$$

where we used that $\alpha X \oplus (1-\alpha)X = X$ for convex X.

Second, we need to determine A's share α_{AB} of the responsibility to avoid collisions between A and B. As A and B may have different acceleration constraints, we let

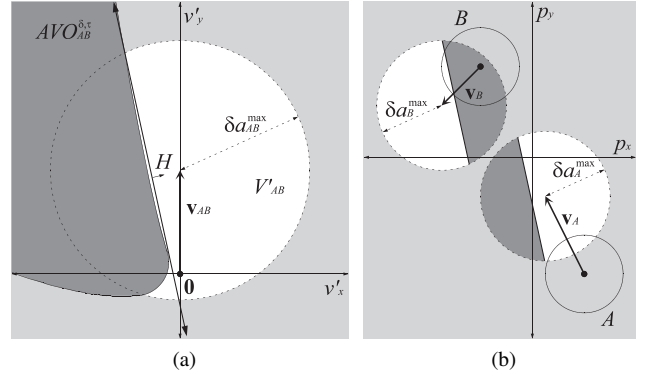$$\alpha_{AB} = \frac{a_A^{\max}}{a_{AB}^{\max}}. \tag{20}$$

Fig. 5. (a) The set V'_{AB} (white) of safe new relative velocities $\mathbf{v}'_{AB}$ for the configuration of Fig. 3(a). The halfplane H maximizes $|D(\mathbf{v}_{AB}, a_{AB}^{\max}) \cap H|$ for $D(\mathbf{v}_{AB}, a_{AB}^{\max}) \cap H \subset V'_{AB}$. (b) The sets (white) of safe new velocities $\mathbf{v}'_A$ and $\mathbf{v}'_B$ for robot A and B, respectively, that guarantee that collisions between them before time τ are avoided.

Note that B's share $\alpha_{BA} = 1 - \alpha_{AB} = \frac{a_B^{\max}}{a_{AB}^{\max}}$, so the share of the responsibility is proportional to each robot's ability to change its velocity and avoid collisions. We can now prove that $V'_A \subset D(\mathbf{v}_A, \delta a_A^{\max})$ and $V'_B \subset D(\mathbf{v}_B, \delta a_B^{\max})$:

$$
\begin{aligned}
V'_A &= \frac{a_A^{\max}}{a_{AB}^{\max}}(\hat{V}'_{AB} \oplus \{-\mathbf{v}_{AB}\}) \oplus \{\mathbf{v}_A\} \\
&\subset \frac{a_A^{\max}}{a_{AB}^{\max}}(D(\mathbf{v}_{AB}, \delta a_{AB}^{\max}) \oplus \{-\mathbf{v}_{AB}\}) \oplus \{\mathbf{v}_A\} \\
&= \frac{a_A^{\max}}{a_{AB}^{\max}}D(\mathbf{0}, \delta a_{AB}^{\max}) \oplus \{\mathbf{v}_A\} = D(\mathbf{0}, \delta a_A^{\max}) \oplus \{\mathbf{v}_A\} \\
&= D(\mathbf{v}_A, \delta a_A^{\max}). \quad (21)
\end{aligned}
$$

The proof for $V'_B \subset D(\mathbf{v}_B, \delta a_B^{\max})$ is constructed similarly. In fact, $|V'_A|/|D(\mathbf{v}_A, \delta a_A^{\max})| = |V'_B|/|D(\mathbf{v}_B, \delta a_B^{\max})|$, which confirms that the potential new velocities are distributed fairly according to each robot's ability to change velocity and avoid collisions (see Fig. 5(b)).

Note that the definitions of V'_A and V'_B are fully symmetric. Reasoning from B's perspective, we have $\mathbf{v}_{BA} = -\mathbf{v}_{AB}$ and $AVO_{BA}^{\tau,\delta} = -AVO_{AB}^{\tau,\delta}$. Therefore, $V'_{BA} = -V'_{AB}$ and $\hat{V}'_{BA} = -\hat{V}'_{AB}$ (see Eqs. (15) and (16)). Further, $\alpha_{BA} = 1 - \alpha_{AB}$. Substituting these identities into Eq. (18) gives $V'_B = \alpha_{BA}(\hat{V}'_{BA} \oplus \{-\mathbf{v}_{BA}\}) \oplus \{\mathbf{v}_B\}$, which is of exactly the same form as Eq. (17) for robot A. As a result, robots A and B can construct their sets V'_A and V'_B, respectively, independently without coordination. We formally call V'_A and V'_B the sets of *optimal reciprocal collision avoidance*.

Definition 4 The set of *optimal reciprocal collision avoidance* $ORCA_{AB}^{\delta,\tau}$ of robot A induced by robot B for control parameter δ and time horizon τ is the set V'_A as defined by Eqs. (17) and (20).

The definition implies that if robot A chooses its new velocity $\mathbf{v}'_A$ in $ORCA_{AB}^{\delta,\tau}$ and robot B chooses its new velocity $\mathbf{v}'_B$ in $ORCA_{BA}^{\delta,\tau}$, robots A and B are guaranteed not to collide with each other before time τ if both A and B use control parameter δ to arrive at their new velocities.
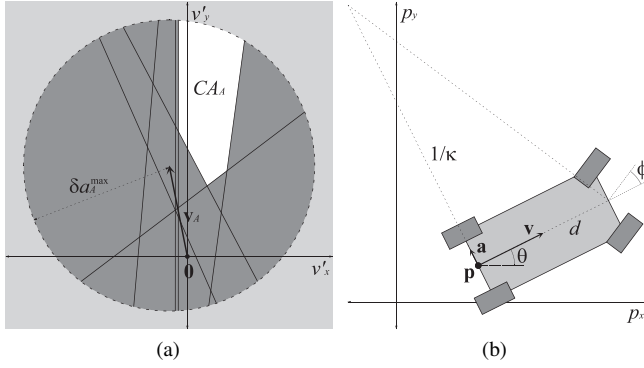
3479

Fig. 6. (a) The halfplanes induced by the other robots for the configuration of Fig. 4(a) for $\delta = 2$ and $\tau = 4$ where all entities are assumed to be decision-making robots with the same acceleration constraint $a_A^{\max}$. The white region is the set CA_A of safe new velocities $\mathbf{v}'_A$ for A. (b) The kinematic model of a car-like robot. Even though it cannot move sidewards, it can accelerate omni-directionally when its speed is nonzero.

D. Multi-Robot Navigation

The *ORCA* formulation can be used for independent navigation of multiple robots sharing a common workspace as follows. Each robot A independently performs a continuous cycle of sensing and acting with time step Δt. In each iteration, the robot determines its preferred velocity $\mathbf{v}_A^{\mathrm{pref}}$, and senses the radius r_B, the current position $\mathbf{p}_B$ and the current velocity $\mathbf{v}_B$ of each other robot B. The robot also needs to know the maximum acceleration $a_B^{\max}$ of each other robot, which may be estimated from its physical properties. Based on this information, the robot A infers the set of safe new velocities $ORCA_{AB}^{\delta,\tau} \oplus \{\mathbf{v}_A\}$ with respect to each other robot B. The set CA_A of new velocities for A that avoid collisions with all robots is the intersection of these sets (see Fig. 6(a)):

$$CA_A = \bigcap_B ORCA_{AB}^{\delta,\tau}. \quad (22)$$

Next, the robot selects the velocity in CA_A that is closest to its preferred velocity $\mathbf{v}_A^{\mathrm{pref}}$ as its new velocity $\mathbf{v}'_A$:

$$\mathbf{v}'_A = \underset{\mathbf{v} \in CA_A}{\arg\min} \|\mathbf{v} - \mathbf{v}_A^{\mathrm{pref}}\|. \quad (23)$$

It can be seen that CA_A is the intersection of the disc $D(\mathbf{v}_A, \delta a_A^{\max})$ of new velocities obeying the acceleration constraint and a set of halfplanes (one halfplane for each other robot). Hence, CA_A is convex, and the new velocity $\mathbf{v}'_A$ as defined in Eq. (23) can be efficiently found using 2-D *linear programming*. It may happen that $CA_A = \emptyset$, i.e. no safe new velocities are available to the robot, particularly in very dense scenarios. In that case, the "least unsafe" velocity can be found using 3-D linear programming (see [19] for details).

Finally, the robot applies the acceleration $\mathbf{a}_A = (\mathbf{v}'_A - \mathbf{v}_A)/\delta$, and the sensing-acting cycle repeats after Δt time. Note that all robots apply this cycle simultaneously.

V. ROBOTS WITH KINEMATIC CONSTRAINTS

Our robot model assumes that the robot is capable of accelerating omni-directionally. This not only applies to holonomic robots, but also to a rich class of kinematically constrained non-holonomic robots, as we show here using the example of a car-like robot. As illustrated in Fig. 6(b), the state of a car-like robot may be given by the center position $\mathbf{p} = (p_x, p_y)$ of its rear axle, its orientation θ, and its linear speed v. Its state-transition equations are then given by:

$$\dot{p}_x = v\cos\theta, \quad \dot{p}_y = v\sin\theta, \quad \dot{\theta} = v\kappa, \quad \dot{v} = a, \quad (24)$$

where the linear acceleration a and the steering wheel angle ϕ are the control inputs, from which the curvature $\kappa = \tan(\phi)/d$ of the car's trajectory directly follows (d is the distance between the car's front and rear axle).

From Eq. (24), it follows that the velocity $\mathbf{v} = (\dot{p}_x, \dot{p}_y)$ and acceleration $\mathbf{a} = (\ddot{p}_x, \ddot{p}_y)$ of the car are given by:

$$\mathbf{v} = \mathbf{R}(\theta)\begin{bmatrix} v \\ 0 \end{bmatrix}, \qquad \mathbf{a} = \mathbf{R}(\theta)\begin{bmatrix} a \\ v^2\kappa \end{bmatrix}, \quad (25)$$

where $\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. Now, let the linear acceleration be bounded by $|a| < a^{\max}$, and the curvature by $|\kappa| < a^{\max}/v^2$, such that extreme steering wheel angles are forbidden at high speeds. It follows that if $v \neq 0$, the vector $(a, v^2\kappa)$ can be controlled to lie anywhere in the axis-aligned square S centered at $\mathbf{0}$ with side $2a^{\max}$. Hence, by Eq. (25), the set of accelerations $\mathbf{a}$ that the robot can attain is given by $\mathbf{R}(\theta)S$. The incircle $D(\mathbf{0}, a^{\max})$ of S therefore contains the accelerations that can be attained regardless of orientation θ.

As a result, by modeling the robot as a disc centered at $\mathbf{p}$ that encompasses the car, we can apply our approach to perform (reciprocal) collision avoidance for a car-like robot, as long as its speed v is nonzero. A similar approach can be taken for other types of kinodynamically constrained robots, such as airplanes and differential-drives.

VI. IMPLEMENTATION AND RESULTS

We have implemented the collision avoidance techniques based on AVO for simulated robots. We assume each robot has a limited sensing radius, within which it can obtain accurate knowledge of the relative position and velocity of neighboring robots. In addition to an acceleration constraint, we let the robots be constrained to a maximum speed $v^{\max}$. All timing results are taken on an Intel Core i7 CPU at 3.2GHz with 4 SMT CPUs.

A. Implementation Details

To compute the AVOs we approximate the boundary as a series of line segments derived from Eq. (29) in the Appendix. We found that 25 samples along the boundary are enough to approximate it well and smoothly avoid collisions. For avoiding moving obstacles we followed the approach of [8], in which the new velocity of the robot is computed as the projection of the preferred velocity onto the boundary of the union of the AVOs. For multi-robot navigation with reciprocal collision avoidance, we used the approximate AVOs to compute the ORCA constraint halfplanes, and used linear programming to find the new velocity for each robot. The implementation was parallelized using OpenMP to exploit the independence of the computations for each
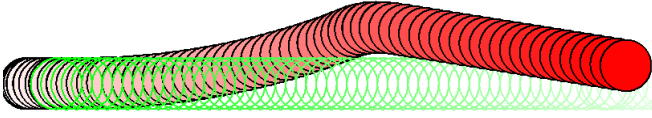
Fig. 7. A robot with acceleration constraints avoid collisions with an oncoming obstacle. Newer frames are on top of older frames and darker.

robot, and take advantage of the multiple cores available on our testbed system.

The control parameter δ and the time horizon τ are tuning parameters of our algorithm. Setting δ too low diminishes the set of valid new velocities, while setting it too high causes the robot to respond too slow to its environment. Its optimal setting depends on the maximum acceleration and speed of the robots. Our experiments suggest that a setting δ such that the set of valid new velocities covers all valid speeds, i.e. $\delta = \frac{2v^{\max}}{a^{\max}}$, is a good heuristic. The value of τ signifies the look-ahead of the robot; setting it too low causes unsafe navigation, while setting it too high reduces the set of new velocities available to the robot. Typically, its value should match the average duration of the validity of extrapolations based on current observations. We have used the values of $\delta = 4$s, $\tau = 10$s, $a^{\max} = 1$m/s^2 and $v^{\max} = 2$m/s in our experiments, and our results suggest that these give convincing behavior in typical scenarios.

B. Benchmarks

We have implemented several benchmarks of simulated robots in a variety of situations to demonstrate the avoidance behavior and analyze runtime.

Moving Obstacle One robot with acceleration constraints avoids collisions with an obstacles moving at a fixed velocity. Fig. 7 shows the trajectories of the robot and the obstacle. The robot reaches its goal without collisions.

Positions Exchange Two robots, both with acceleration constrains, exchange positions. Fig. 1 shows the trajectory of each robot. Each dot, represents the center of an robot separated by fixed time steps. The spacing between the dots increases as the robots accelerate to avoid collision.

Circle-n Robots are initialized in a circle of n robots. Each robots is given a goal at the antipodal position across the circle. Three time-lapsed images from the Circle-100 demo are shown in Fig. 8.

Car-Like Robots We implemented our approach for car-like robots as discussed in Section V. Our approach computes accelerations for the cars, which are translated into proper control inputs using Eq. (25), and integrated according to Eq. (24) to obtain the updated state of the cars. In this scenario (see Fig. 9), 100 cars have random initial positions, orientations, and goals. Their kinematic and dynamic constraints are obeyed while avoiding collisions with each other.

Videos of these and other benchmark scenarios are available at http://gamma.cs.unc.edu/AVO/.

C. Performance Results

To test the performance of our model we implemented the Circle-n benchmark with a varying number of robots. The
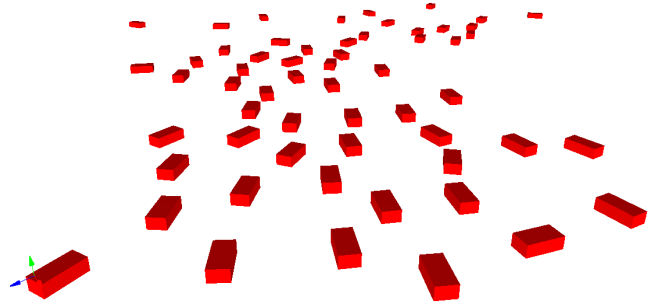

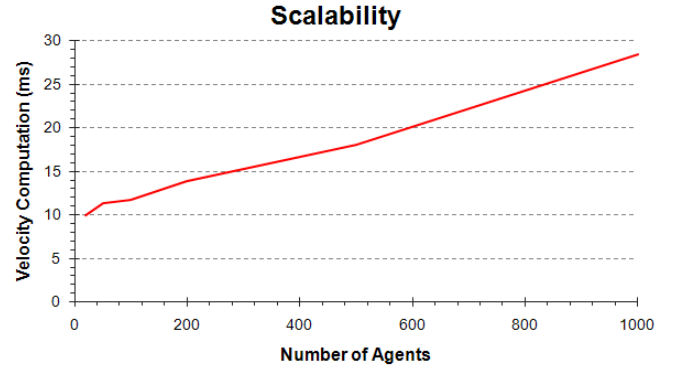Fig. 9. **Car-Like Robots Benchmark** 100 cars move through the scene and avoid collisions.

Fig. 10. **Performance & Scalability** Performance on the circle-n demo up to 1000 robots. For all robots, new velocity computation took under 30 ms and the simulation ran at over 30 FPS.

results are shown in Fig. 10, which graphs the total time taken to compute a new velocity for all the robots as the number of robots increases. We observe an approximately linear increase in the computation time as the number of robots increase. We are able to achieve a simulation rate of over 30FPS for 1,000 robots.

If used in a performance critical application, our implementation could easily be further optimized. For example, our method is well poised to take advantage of data-parallel processing units found in modern CPUs and GPUs.

VII. CONCLUSION AND FUTURE WORK

In this paper we have introduced the *acceleration-velocity obstacle* for collision-avoidance among moving obstacles by robots subject to acceleration constraints. We have used this concept to derive a formulation for *reciprocal collision avoidance* for multiple robots. Our experiments have shown that our approach achieves fast running times and produces convincing collision-free motions obeying the acceleration constraints, even in dense environments.

We have assumed that the robots move in a 2-D workspace, are disc-shaped, and capable of omni-directional acceleration. Even though we have shown that this model can be made applicable to non-holonomic robots such as cars, the question arises whether an approach can be formulated directly for arbitrarily kinematically constrained systems. As shown in [20], it is possible to derive "control input obstacles" for collision avoidance with moving obstacles. For
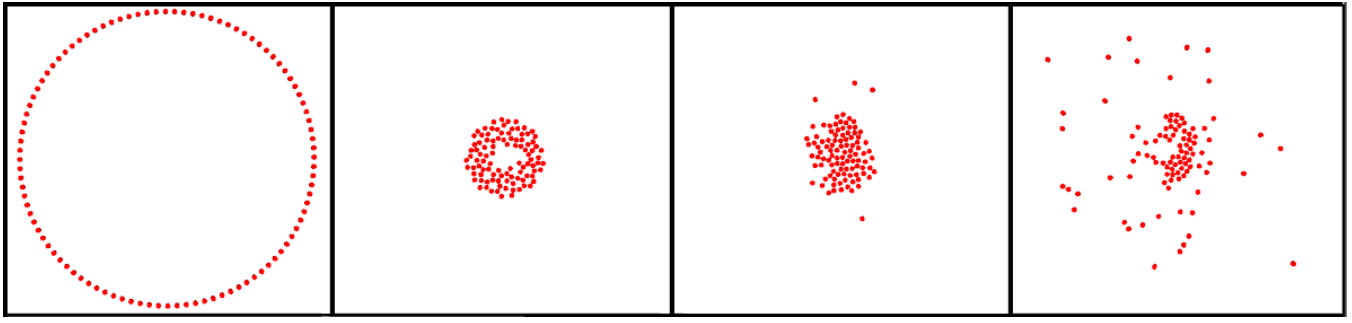
Fig. 8. **Circle-n Benchmark** 100 robots move to their antipodal position on the circle. As they pass through the center, the density increases, but the motions remain smooth and collision-free.

reciprocal collision avoidance, however, the key is that the *relative* motion of two robots can be expressed in terms of the difference between their control inputs. This is not generally the case for kinematically constrained robots.

One of the motivations of this work is to apply it to collision-avoidance for teams of quadrotor helicopters flying at high speeds in tight airspaces. As the attitude and motion control of these vehicles are typically decoupled [12], they can essentially be treated as holonomic robots capable of omni-directional acceleration. Still, it requires extending our approach to dealing with 3-D workspaces and uncertainty in the sensor data. This is subject of ongoing research.

Another potential application of our approach is to use it in motion planning for efficiently checking whether the robot is in an *inevitable collision state* [4], which is considered a challenging problem in the context of multiple moving obstacles [11]. If no valid new velocity is available in the set CA_A of Fig. 4, no trajectory towards a new velocity is safe for at least τ time, and one may (conservatively) conclude that a collision is inevitable.

REFERENCES

[1] Y. Abe, M. Yoshiki. Collision avoidance method for multiple autonomous mobile agents by implicit cooperation. *Proc. IEEE RSJ Int. Conf. on Intelligent Robots and Systems*, 2001.

[2] B. Damas, J. Santos-Victor. Avoiding moving obstacles: the forbidden velocity map. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009.

[3] P. Fiorini, Z. Shiller. Motion planning in dynamic environments using Velocity Obstacles. *Int. J. of Robotics Research* 17(7):760–772, 1998.

[4] T. Fraichard, H. Asama. Inevitable collision states - a step towards safer robots? *Advanced Robotics* 18(10):1001–1024, 2004.

[5] D. Fox, W. Burgard, S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine* 4:23-33, 1997.

[6] C. Fulgenzi, A. Spalanzani, C. Laugier. Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007.

[7] O. Gal, Z. Shiller, E. Rimon. Efficient and safe on-line motion planning in dynamic environments. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2009.

[8] S. Guy, J. Chhugani, C. Kim, N. Satish, P. Dubey, M. Lin, D. Manocha. ClearPath: Highly parallel collision avoidance for multi-agent simulation. Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation, 2009.

[9] D. Hsu, R. Kindel, J. Latombe, S. Rock. Randomized kinodynamic motion planning with moving obstacles. *Int. J. Robotics Research* 21(3):233-255, 2002.

[10] B. Kluge, E. Prassler. Reflective navigation: Individual behaviors and group behaviors. *Proc. IEEE Int. Conf. Robotics and Automation*, 2004.

[11] L. Martinez-Gomez, T. Fraichard. Collision avoidance in dynamic environments: an ICS-based solution and its comparative evaluation. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2009.

[12] N. Michael, D. Mellinger, Q. Lindsey, V. Kumar. The GRASP multiple mirco-UAV test bed: experimental evaluation of multirobot aerial control algorithms. *IEEE Robotics and Automation Magazine* 17(3):56–65, 2010.

[13] E. Owen, L. Montano. Motion planning in dynamic environments using the velocity space. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[14] S. Petti, T. Fraichard. Safe motion planning in dynamic environments. *Proc. IEEE RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[15] E. Prassler, J. Scholz, P. Fiorini. A robotics wheelchair for crowded public environment. *IEEE Robotics and Automation Magazine* 8(1):38-45, 2001.

[16] Z. Shiller, F. Large, and S. Sekhavat. Motion planning in dynamic environments: obstacles moving along arbitrary trajectories. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2001.

[17] J. Snape, J. van den Berg, S. Guy, D. Manocha. Independent navigation of multiple robots with Hybrid Reciprocal Velocity Obstacles. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009.

[18] J. van den Berg, M. Lin, D. Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008.

[19] J. van den Berg, S. Guy, M. Lin, D. Manocha. Reciprocal n-body collision avoidance. *Proc. Int. Symposium of Robotics Research*, 2009.

[20] D. Wilkie, J. van den Berg, D. Manocha. Generalized Velocity Obstacles. *IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, 2009.

APPENDIX

The left and right tangent points $\mathbf{q}^+(\mathbf{p}, r)$ and $\mathbf{q}^-(\mathbf{p}, r)$ on a circle of radius r centered at $\mathbf{p}$ to a line through the origin are given by:

$$\mathbf{q}^{\pm}(\mathbf{p}, r) = \begin{bmatrix} \ell & \mp r \\ \pm r & \ell \end{bmatrix} \mathbf{p} \frac{\ell}{\|\mathbf{p}\|^2}, \quad \ell = \sqrt{\|\mathbf{p}\|^2 - r^2}. \quad (26)$$

Let the centers and the radii of the discs that form the acceleration-velocity obstacle be denoted $\mathbf{c}(t)$ and $r(t)$ respectively (see Eq. (12)):

$$\mathbf{c}(t) = \frac{\delta(e^{-t/\delta} - 1)\mathbf{v}_{AB} - \mathbf{p}_{AB}}{t + \delta(e^{-t/\delta} - 1)}, \quad (27)$$

$$r(t) = \frac{r_{AB}}{t + \delta(e^{-t/\delta} - 1)}. \quad (28)$$

Then, the left and right boundaries $\mathbf{b}^+(t)$ and $\mathbf{b}^-(t)$ of the acceleration-velocity obstacle AVO^{τ}_{AB} are given by:

$$\mathbf{b}^{\pm}(t) = \mathbf{c}(t) - \frac{r(t)}{\dot{r}(t)}\dot{\mathbf{c}}(t) + \mathbf{q}^{\pm}(\frac{r(t)}{\dot{r}(t)}\dot{\mathbf{c}}(t), r(t)), \quad (29)$$

for $0 < t \leq \tau$.