

# Smooth and Collision-Free Navigation for Multiple Robots Under Differential-Drive Constraints

Jamie Snape, *Student Member, IEEE*, Jur van den Berg, Stephen J. Guy, and Dinesh Manocha

**Abstract**— We present a method for smooth and collision-free navigation for multiple independent robots under differential-drive constraints. Our algorithm is based on the *optimal reciprocal collision avoidance* formulation and guarantees both smoothness in the trajectories of the robots and locally collision-free paths. We provide proofs of these guarantees and demonstrate the effectiveness of our method in experimental scenarios using *iRobot Create* mobile robots navigating amongst each other.

## I. INTRODUCTION

Differential-drive robots are ubiquitous in robotics. These robots use a simple drive mechanism that consists of two drive wheels mounted on a common axis. Moreover, each wheel can be independently driven in both forward and reverse directions. From vacuum cleaners [1] to powered wheelchairs [2], most mobile robots in practical service have differential-drive constraints. Applications have been as diverse as the inspection of walls [3], pharmaceutical warehousing [4], and virtual tour guides [5].

Increasingly, robots are used not only in isolation, but as part of a distributed system of multiple robots. Groups of coordinated mobile robots may be used for surveillance, and environmental monitoring, as well as search and rescue [6]. In such cases it is necessary to develop methods to compute collision-free paths for each of these robots with respect to other robots and obstacles. Moreover, the robot should move smoothly. The smoothness property is important for many mobile or practical service robots, as they must take into account the physical limits of robot actuators and other safety issues.

Most of the prior work in smooth and collision-free navigation has been limited to single robots moving amongst dynamic obstacles. There is extensive work on navigating multiple robots, including global methods based on centralized or decoupled approaches [7] and local and reactive methods [8], [9], [10] for computing collision-free paths. However, most of these algorithms do not take into account kinematic constraints, nor do they provide guarantees of smoothness. The resulting paths may also have discontinuities.

This work was supported in part by ARO contract W911NF-04-1-0088; NSF awards 0636208, 0917040, and 0904990; DARPA/RDECOM contract WR91CRB-08-C-0137; and Intel.

J. Snape, S. J. Guy, and D. Manocha are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA. Email: {snape, sjguy, dm}@cs.unc.edu.

J. van den Berg is with the Department of Industrial Engineering and Operations Research, University of California, Berkeley, Berkeley, CA 94720, USA. Email: berg@berkeley.edu.

Website: <http://gamma.cs.unc.edu/ORCA-DD/>.

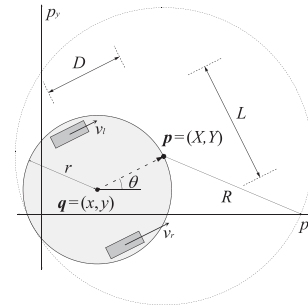


Fig. 1. The kinematic model of a differential-drive robot. The center of the robot is  $q = (x, y)$ , its radius is  $r$ , and the distance between its wheels is  $L$ . Each wheel is attached to a separate motor and may assume a different speed. The effective center  $p = (X, Y)$  is located a distance  $D$  from  $q$  and the effective radius is  $R$ .

In this paper, we present a new method for navigating multiple independent robots with differential-drive constraints in the plane. Our algorithm is local and velocity-based. Each differential-drive robot executes an independent continuous cycle of sensing and acting. Our algorithm computes a new velocity based on the current positions and velocities of the other robots. The robots need not explicitly communicate with each other.

The result of our algorithm is the control inputs, the two wheel speeds, for a robot with differential-drive constraints. Our method works by enlarging the radius used in navigation in a precise way which provides additional maneuverability, allowing us to handle the differential-drive constraints in a smooth and collision-free manner. We combine this enlarged radius with optimal reciprocal collision avoidance [11] to obtain our algorithm.

To our knowledge, this is the first algorithm that mathematically guarantees smooth and collision-free trajectories for multiple independent robots navigating in a shared environment that is local and reactive and takes into account the kinematic constraints of a differential-drive robot.

This rest of this paper is organized as follows. We first summarize related work in Section II. In Section III, we describe the kinematics of a differential-drive robot and introduce the notion of effective center and effective radius that we use to transform a velocity to control inputs for such a robot. In Section IV, we combine this method with optimal reciprocal collision avoidance to create our algorithm for smoothly navigating multiple differential-drive robots amongst each other. We highlight some of the mathematical guarantees on the trajectories computed by our algorithm in Section V. Finally, we describe an implementation of our algorithm using *iRobot Create* mobile robots and describe

our experimental results in Section VI.

## II. PRIOR WORK

In this section, we briefly survey prior work on motion planning under kinematic constraints for single and multiple robots and discuss other approaches for navigating multiple robots.

Some of the earliest work on planning for robots with kinematic constraints dates back to the classical Dubins car [12]. This simplified car model is restricted to forward motions with a fixed speed and bounded turning radius. The Reeds-Shepp car [13] adds a reverse gear to the Dubins car, while the simple car [14], [15] extends the model further with variable speed in any direction.

Many works have examined the issue of a single robot navigating through a cluttered environment containing dynamic obstacles [9], [10]. Some of these approaches are based on the notion of velocity obstacles [8] and its extensions to navigating multiple robots and virtual agents [16], [17]. These navigation methods do not provide any mathematical guarantees on the smoothness of the resulting trajectories.

The work of [18] proposes a reactive navigation scheme based on a number of predefined discrete behaviors, while [19] describes a local planner in which multiple robots explicitly communicate with each other.

Other global navigation algorithms for multiple robots can be classified into decoupled or centralized planners. Decoupled planners [7] consider each robot individually and then modify their velocities as a post-process to avoid collisions with each other. On the other hand, centralized planners generate a single composite system by combining the degrees of freedom of each robot and apply traditional single robot navigation algorithms to that [14]. However, prior single robot algorithms for smooth or continuous curvature paths may not be applicable to such a composite system in terms of computing smooth trajectories for each robot.

Recently, many algorithms have been proposed for navigating multiple aerial [20] or aquatic robots [21] in three dimensions. This follows the successful deployment of unmanned aerial vehicles and autonomous underwater vehicles for both civil and military applications [22].

Other work on the navigation of multiple robots with kinematic constraints has focused on follow-the-leader behavior [23] and time-optimal trajectories [24]. Some of the methods include a modified rapidly-exploring random tree planner [25], and mixed integer nonlinear programming [26].

## III. KINEMATICS OF A DIFFERENTIAL-DRIVE ROBOT

In this section, we give a brief overview of the kinematic constraints of a differential-drive robot, which is applicable to many contemporary mobile robot systems. Next we describe a method for transforming a velocity to the control inputs of the robot, which are simply its two wheel speeds.

### A. Kinematic Model

As illustrated in Fig. 1, the configuration of a differential-drive robot is given by the position of its center  $\mathbf{q} = (x, y)$

and its orientation  $\theta$ . Its configuration transition equations [14] are given as

$$\dot{x} = \frac{v_l + v_r}{2} \cos \theta, \quad \dot{y} = \frac{v_l + v_r}{2} \sin \theta, \quad \dot{\theta} = \frac{v_r - v_l}{L}, \quad (1)$$

where constant  $L > 0$  is the distance between the wheels of the robot, and  $v_l$  and  $v_r$  are the signed speeds and control inputs for its left and right wheels, respectively. In addition, the speeds  $v_l$  and  $v_r$  are bounded such that  $v_l, v_r \in [-v_{\max}, v_{\max}]$ .

The robot is disc-shaped with radius  $r$ , and we assume that it is sufficiently lightweight and its motors powerful enough that it can attain any wheel speed  $v_l$  or  $v_r$  within a bounded interval near instantaneously.

### B. Effective Center and Effective Radius

We now show the precise method used to enlarge the radius for navigation, which results in increased maneuverability for the robot, allowing smooth handling of the kinematic constraints of a differential-drive robot.

Since the center  $\mathbf{q}$  of a differential-drive robot is not fully controllable, we adapt the approach of [27] and define the “effective center” of such a robot to be a point  $\mathbf{p} = (X, Y)$  translated a distance  $D > 0$  from  $\mathbf{q}$  in a direction orthogonal to the axle of the robot. Similarly, we have the “effective radius”  $R > 0$  such that  $R = r + D$ . The effective center and effective radius are also shown on Fig. 1. Unlike the center  $\mathbf{q}$ , the effective center  $\mathbf{p}$  may be translated in a direction orthogonal to the orientation of the wheels of the robots and is fully controllable.

It follows from the definition of the effective center that

$$X = x + D \cos \theta, \quad Y = y + D \sin \theta.$$

Then, substituting equations (1) and applying the chain rule, we have

$$\begin{aligned} \dot{X} &= \left( \frac{\cos \theta}{2} + \frac{D \sin \theta}{L} \right) v_l + \left( \frac{\cos \theta}{2} - \frac{D \sin \theta}{L} \right) v_r, \\ \dot{Y} &= \left( \frac{\sin \theta}{2} - \frac{D \cos \theta}{L} \right) v_l + \left( \frac{\sin \theta}{2} + \frac{D \cos \theta}{L} \right) v_r. \end{aligned} \quad (2)$$

This gives us a linear system of the form

$$\mathbf{v} = M(\theta) \cdot \mathbf{u}, \quad (3)$$

where  $\mathbf{u} = (v_l, v_r)$ ,  $\mathbf{v} = (\dot{X}, \dot{Y})$ , and  $M(\theta)$  is a two-dimensional matrix. Hence, we can obtain wheel speeds  $v_l$  and  $v_r$  from a velocity  $\mathbf{v}$  by solving  $\mathbf{u} = M^{-1}(\theta) \cdot \mathbf{v}$ . Note that  $D \neq 0$  and  $L \neq 0$  by definition, so the matrix  $M(\theta)$  is invertible for all  $\theta$ .

Since the wheel speeds of the differential-drive robot are bounded,  $\mathbf{u} = (v_l, v_r)$  lies within an axis-aligned square  $S$  with lower left vertex  $(-v_{\max}, -v_{\max})$  and upper right vertex  $(v_{\max}, v_{\max})$ . Hence, the set of velocities  $\mathbf{v}$  that the robot can attain is given by the linear transformation  $M(\theta) \cdot S$ . It follows that if  $D = r$ , in which case the effective radius is  $R = 2r$ , then this set of velocities is a square  $S'$  whose center lies at  $\mathbf{v} = (0, 0)$  and whose orientation depends on  $\theta$ . The incircle of  $S'$  therefore contains the velocities that can be attained regardless of orientation  $\theta$ .

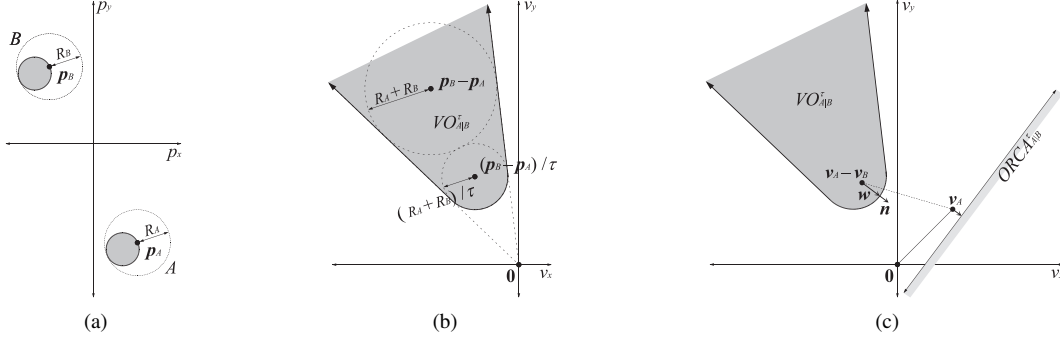


Fig. 2. (a) A configuration of two disc-shaped robots  $A$  and  $B$  in the plane with radii  $r_A$  and  $r_B$ , positions  $\mathbf{q}_A$  and  $\mathbf{q}_B$ , effective radii  $R_A$  and  $R_B$ , and positions  $\mathbf{p}_A$  and  $\mathbf{p}_B$  of their effective center, respectively. (b) The velocity obstacle  $VO_{A|B}^\tau$  for  $A$  induced by  $B$  within the window of time  $\tau = 2$ . The sides of  $VO_{A|B}^\tau$  are tangent to a disc of radius  $R_A + R_B$  with center  $\mathbf{p}_B - \mathbf{p}_A$  and it is truncated by a disc of radius  $(R_A + R_B) / \tau$  with center  $(\mathbf{p}_B - \mathbf{p}_A) / \tau$ . (c) The set of permitted velocities  $ORCA_{A|B}^\tau$  available to  $A$  for optimal reciprocal collision avoidance with  $B$  within the window of time  $\tau$ . The half-plane  $ORCA_{A|B}^\tau$  is bounded by a line perpendicular to  $\mathbf{w}$  through  $\mathbf{v}_A + \frac{1}{2}\mathbf{w}$ , where  $\mathbf{w}$  is from  $\mathbf{v}_A - \mathbf{v}_B$  to the closest point on  $\partial VO_{A|B}^\tau$ .

The result of this transformation is a fully controllable point  $\mathbf{p} = (X, Y)$ . We can apply optimal reciprocal collision avoidance to the disc centered at the effective center  $\mathbf{p}$  with a radius  $R$  that has been enlarged to encompass the robot itself (Fig. 1).

#### IV. MULTI-ROBOT NAVIGATION

In this section, we combine our method for obtaining control inputs from a velocity with optimal reciprocal collision avoidance to design a new algorithm for navigating multiple independent differential-drive robots in the plane that takes into account their kinematic constraints. We begin by reviewing the notion of optimal reciprocal collision avoidance, and then present our algorithm.

##### A. Optimal Reciprocal Collision Avoidance

Optimal reciprocal collision avoidance [11] is a velocity-based local collision avoidance approach based on the notion of velocity obstacles [8].

Consider the configuration of two disc-shaped robots in Fig. 2(a). The velocity obstacle for  $A$  induced by  $B$  within the window of time  $\tau$ , denoted  $VO_{A|B}^\tau$ , is the set of velocities of  $A$  relative to  $B$  that will cause a collision between  $A$  and  $B$  at some moment before time  $\tau$  has elapsed, assuming that both robots maintain a constant trajectory within that time interval:

$$VO_{A|B}^\tau = \{\mathbf{v} \mid \exists t \in [0, \tau] :: t(\mathbf{v} - \mathbf{v}_B) \in D(\mathbf{p}_B - \mathbf{p}_A, R_A + R_B)\},$$

where  $D(\mathbf{p}, R)$  is an open disc of radius  $R$  centered at  $\mathbf{p}$ .

Hence, in the velocity space, the velocity obstacle within a finite window of time takes the geometric form of truncated cone, as shown in Fig. 2(b). In our calculations we use the positions  $\mathbf{p}_A = (X, Y)_A$  and  $\mathbf{p}_B = (X, Y)_B$  of the effective centers of  $A$  and  $B$  and their effective radii  $R_A$  and  $R_B$ , respectively. Moreover, the velocities of each robot are  $\mathbf{v}_A = (\dot{X}, \dot{Y})_A$  and  $\mathbf{v}_B = (\dot{X}, \dot{Y})_B$ . We also assume that each robot has complete knowledge of the environment, that is it is aware of the exact position and velocity of all the other robots at all times.

Now, if  $A$  and  $B$  each choose a velocity outside the velocity obstacle induced by the other, then they will be collision-free for at least  $\tau$  time. Therefore, the set of collision-avoiding velocities for  $A$  given  $B$  within the window of time  $\tau$  is

$$CA_{A|B}^\tau = \{\mathbf{v} \mid \mathbf{v} \notin VO_{A|B}^\tau\}.$$

While choosing  $\mathbf{v}_A$  from  $CA_{A|B}^\tau$  and  $\mathbf{v}_B$  from  $CA_{B|A}^\tau$  ensures that  $A$  and  $B$  will not collide, their respective trajectories may not be smooth due to oscillations in velocity [16]. Optimal reciprocal collision avoidance resolves this situation. The set of velocities  $ORCA_{A|B}^\tau \subset CA_{A|B}^\tau$  available to  $A$  for optimal reciprocal collision avoidance with  $B$  within the window of time  $\tau$  is defined as follows.

Referring to Fig. 2(c), let  $\mathbf{w}$  be the vector from  $\mathbf{v}_A - \mathbf{v}_B$  to the closest point on the boundary of the velocity obstacle,  $\mathbf{w} = (\arg \min_{\mathbf{v} \in \partial VO_{A|B}^\tau} \|\mathbf{v} - (\mathbf{v}_A - \mathbf{v}_B)\|_2) - (\mathbf{v}_A - \mathbf{v}_B)$ , where  $\partial VO_{A|B}^\tau$  is the boundary of  $VO_{A|B}^\tau$  and  $\mathbf{v}_A$  and  $\mathbf{v}_B$  are the current velocities of  $A$  and  $B$ , respectively.

Now, let  $\mathbf{n}$  be the outward normal of  $\partial VO_{A|B}^\tau$  at the point  $\mathbf{v}_A - \mathbf{v}_B + \mathbf{w}$ , and assume that both  $A$  and  $B$  adapt their velocity by  $\frac{1}{2}\mathbf{w}$  to avoid colliding with each other. Then the set of permitted velocities  $ORCA_{A|B}^\tau$  is

$$ORCA_{A|B}^\tau = \{\mathbf{v} \mid (\mathbf{v} - (\mathbf{v}_A + \frac{1}{2}\mathbf{w})) \cdot \mathbf{n} \geq 0\}.$$

This is the half-plane of velocities shown in Fig. 2(c) and is a strict subset of the collision-avoiding velocities  $CA_{A|B}^\tau$ . In common with the velocity obstacle, both robots  $A$  and  $B$  can construct  $ORCA_{A|B}^\tau$  and  $ORCA_{B|A}^\tau$ , respectively, without communication with each other, requiring knowledge of only the radius, position, and velocity of each other.

It follows that if  $A$  and  $B$  select velocities  $\mathbf{v}_A$  from  $ORCA_{A|B}^\tau$  and  $\mathbf{v}_B$  from  $ORCA_{B|A}^\tau$ , respectively, then the trajectories of  $A$  and  $B$  will be smooth and collision-free for at least time  $\tau$ .

##### B. Navigation Algorithm

Incorporating each of the previous components together, we now present our algorithm.

```

Input  $\mathcal{A}$  : List of differential-drive robots
loop
  for all  $A_i \in \mathcal{A}$  do
    Sense  $\mathbf{q}_{A_i} = (x, y)_{A_i}$  and  $\mathbf{v}_{A_i}$ 
    Calculate  $\mathbf{p}_{A_i} = (X, Y)_{A_i}$ 
    for all  $A_j \in \mathcal{A}$  such that  $i \neq j$  do
      Sense  $\mathbf{q}_{A_j} = (x, y)_{A_j}$  and  $\mathbf{v}_{A_j}$ 
      Calculate  $\mathbf{p}_{A_j} = (X, Y)_{A_j}$ 
      Construct  $ORCA_{A_i|A_j}^\tau$ 
    end for
    Construct  $ORCA_{A_i}^\tau$  from all  $ORCA_{A_i|A_j}^\tau$ 
    Compute preferred velocity  $\mathbf{v}_{A_i}^{\text{pref}}$ 
    Compute new velocity  $\mathbf{v}_{A_i}^{\text{new}} \in ORCA_{A_i}^\tau$  closest to  $\mathbf{v}_{A_i}^{\text{pref}}$  using linear programming
    Compute control inputs  $\mathbf{u}_{A_i} = (v_l, v_r)_{A_i}$  from  $\mathbf{v}_{A_i}^{\text{new}}$  by solving  $\mathbf{v}_{A_i} = M(\theta) \cdot \mathbf{u}_{A_i}$ 
    Apply control inputs to actuators of  $A_i$ 
  end for
end loop

```

Fig. 3. Our algorithm for smooth and collision-free multi-robot navigation under differential-drive constraints.

Let  $\mathcal{A}$  be a set of robots sharing an environment. Each robot  $A_i \in \mathcal{A}$  has a current position  $\mathbf{q}_{A_i}$ , a current velocity  $\mathbf{v}_{A_i}$ , and a fixed radius  $r_{A_i}$ . Moreover, each robot has an effective center at position  $\mathbf{p}_{A_i}$  and effective radius  $R_{A_i}$ . Then the set of permitted velocities available to a robot  $A_i$  for optimal reciprocal collision avoidance with all robots  $A_j$  such that  $j \neq i$  within the window of time  $\tau$  is the intersection of half-planes

$$ORCA_{A_i}^\tau = \bigcap_{\substack{A_j \in \mathcal{A} \\ i \neq j}} ORCA_{A_i|A_j}^\tau.$$

Furthermore, we define the preferred velocity  $\mathbf{v}_{A_i}^{\text{pref}}$  of  $A_i$  to be the velocity directed from the effective center  $\mathbf{p}_{A_i}$  towards its goal position  $\mathbf{p}_{A_i}^{\text{goal}}$  with a magnitude equal to some preferred speed  $\|\mathbf{v}_{A_i}^{\text{pref}}\|_2$ . This is the velocity that  $A_i$  would have selected had no other robots been in its way. Then each robot  $A_i$  should choose the new velocity from  $ORCA_{A_i}^\tau$  that is closest to its preferred velocity,  $\mathbf{v}_{A_i}^{\text{new}} = \arg \min_{\mathbf{v} \in ORCA_{A_i}^\tau} \|\mathbf{v} - \mathbf{v}_{A_i}^{\text{pref}}\|_2$ . This may be calculated efficiently using linear programming [11], and the control inputs of the robot may then be calculated from  $\mathbf{v}_{A_i}^{\text{new}}$  using the notion of effective center and effective radius described in Section III-B.

The algorithm used by each robot is summarized by the algorithm in Fig. 3.

## V. MATHEMATICAL GUARANTEES

In this section, we highlight the mathematical guarantees of smooth and collision-free trajectories offered by our algorithm for navigating multiple independent robots with differential-drive constraints. In particular, we provide a proof that optimal reciprocal collision avoidance guarantees

that the motion of each robot is smooth, in the mathematical sense.

### A. Smooth Motion

The key property of our algorithm which distinguishes it from other reactive methods, is that in addition to generating collision-free paths, it guarantees that the motion of each robot is smooth, that is the trajectory generated by the velocities calculated by our algorithm is continuous.

We have the following new result for optimal reciprocal collision avoidance.

*Theorem 1:* Given a small time step  $\delta t$ , the trajectory  $\mathbf{v}_{A_i}(t)$  generated by the sequence of velocities  $\mathbf{v}_{A_i}$  in the velocity space is continuous, that is  $\mathbf{v}_{A_i}(t) \approx \mathbf{v}_{A_i}(t + \delta t)$ , where  $\approx$  denotes ‘‘arbitrarily close to’’ as  $\delta t \rightarrow 0$ .

The proof of Theorem 1 is by induction on the time step  $\delta t$ . In the inductive step, we will prove that  $\mathbf{v}_{A_i}(t + \delta t) \approx \mathbf{v}_{A_i}(t) \Rightarrow \mathbf{v}_{A_i}(t + 2\delta t) \approx \mathbf{v}_{A_i}(t + \delta t)$ , and in the base case we will prove that  $\mathbf{v}_{A_i}(\delta t) \approx \mathbf{v}_{A_i}(0)$  for a proper initialization of the simulation. The proof requires several additional results, which are presented first.

*Lemma 2:* The trajectory  $\mathbf{v}_{A_i}^{\text{pref}}(t)$  generated by the sequence of preferred velocities  $\mathbf{v}_{A_i}^{\text{pref}}$  in the velocity space is continuous, that is  $\mathbf{v}_{A_i}^{\text{pref}}(t) \approx \mathbf{v}_{A_i}^{\text{pref}}(t + \delta t)$ .

*Proof:* The preferred velocity  $\mathbf{v}_{A_i}^{\text{pref}}$  of each robot is the difference of the position  $\mathbf{p}_{A_i}$  of its effective center and its goal position  $\mathbf{p}_{A_i}^{\text{goal}}$ . Since  $\mathbf{p}_{A_i}(t)$  is clearly continuous and  $\mathbf{p}_{A_i}^{\text{goal}}$  is fixed, it follows that  $\mathbf{v}_{A_i}^{\text{pref}}(t)$  is continuous. ■

*Lemma 3:* If the trajectory  $\mathbf{v}_{A_i}(t)$  is continuous, that is  $\mathbf{v}_{A_i}(t) \approx \mathbf{v}_{A_i}(t + \delta t)$ , then  $ORCA_{A_i|A_j}^\tau(t) \approx ORCA_{A_i|A_j}^\tau(t + \delta t)$ .

*Corollary 4:* Let  $ORCA_{A_i}^\tau(t)$  be the intersection of half-planes  $ORCA_{A_i|A_j}^\tau(t)$  at time  $t$  for all  $j \neq i$ , then  $ORCA_{A_i}^\tau(t) \approx ORCA_{A_i}^\tau(t + \delta t)$ .

*Proof:* The half-plane  $ORCA_{A_i|A_j}^\tau$  is a tangent to the velocity obstacle  $VO_{A_i|A_j}^\tau$ . Since  $VO_{A_i|A_j}^\tau(t)$  is continuous, as the position of the effective center  $\mathbf{p}_{A_i}(t)$  and, by assumption, the velocity  $\mathbf{v}_{A_i}(t)$  are continuous, it follows that  $ORCA_{A_i|A_j}^\tau(t)$  is continuous. ■

*Lemma 5:* The optimal reciprocal collision avoidance algorithm selects the new velocity  $\mathbf{v}_{A_i}$  based on the preferred velocity  $\mathbf{v}_{A_i}^{\text{pref}}$  and the intersection of half-planes  $ORCA_{A_i}^\tau$ , that is  $\mathbf{v}_{A_i}(t + \delta t) = f(\mathbf{v}_{A_i}^{\text{pref}}(t), ORCA_{A_i}^\tau(t))$ , for some function  $f$ , the linear programming function. This function is continuous, that is if  $\mathbf{v}_{A_i}^{\text{pref}}(t) \approx \mathbf{v}_{A_i}^{\text{pref}}(t + \delta t)$  and  $ORCA_{A_i}^\tau(t) \approx ORCA_{A_i}^\tau(t + \delta t)$ , then  $f(\mathbf{v}_{A_i}^{\text{pref}}(t), ORCA_{A_i}^\tau(t)) \approx f(\mathbf{v}_{A_i}^{\text{pref}}(t + \delta t), ORCA_{A_i}^\tau(t + \delta t))$ .

*Proof:* The function  $f$  is a projection of  $\mathbf{v}_{A_i}^{\text{pref}}(t)$  onto  $ORCA_{A_i}^\tau(t)$ . Since  $\mathbf{v}_{A_i}^{\text{pref}}(t)$  and  $ORCA_{A_i}^\tau(t)$  are continuous, it follows that  $f$  is continuous. ■

*Proof:* [Proof of Theorem 1] The inductive step proceeds as follows. From the optimal reciprocal collision avoidance algorithm,  $\mathbf{v}_{A_i}(t + 2\delta t) = f(\mathbf{v}_{A_i}^{\text{pref}}(t + \delta t), ORCA_{A_i}^\tau(t + \delta t))$ . Moreover,  $ORCA_{A_i}^\tau(t + \delta t) \approx ORCA_{A_i}^\tau(t)$  from Corollary 4 and  $\mathbf{v}_{A_i}^{\text{pref}}(t + \delta t) \approx \mathbf{v}_{A_i}^{\text{pref}}(t)$  from Lemma 2. Now, the

function  $f$  is continuous, so  $f(\mathbf{v}_{A_i}^{\text{pref}}(t + \delta t), \text{ORCA}_{A_i}^\tau(t + \delta t)) \approx f(\mathbf{v}_{A_i}^{\text{pref}}(t), \text{ORCA}_{A_i}^\tau(t))$ , and by Lemma 5, it follows that  $\mathbf{v}_{A_i}(t + \delta t) = f(\mathbf{v}_{A_i}^{\text{pref}}(t), \text{ORCA}_{A_i}^\tau(t))$ . Hence,  $\mathbf{v}_{A_i}(t + 2\delta t) \approx \mathbf{v}_{A_i}(t + \delta t)$ . By the assumption in Lemma 3 that  $\mathbf{v}_{A_i}(t)$  is continuous, so  $\mathbf{v}_{A_i}(t + \delta t) \approx \mathbf{v}_{A_i}(t)$ , it follows that  $\mathbf{v}_{A_i}(t + 2\delta t) \approx \mathbf{v}_{A_i}(t + \delta t)$ , as required.

The base case of the inductive proof is  $\mathbf{v}_{A_i}(\delta t) \approx \mathbf{v}_{A_i}(0)$ . This occurs if each robot is initialized with  $\mathbf{v}_{A_i}^{\text{pref}}$  and the starting position of its effective center is such that all other robots are sufficiently distant so that  $\mathbf{v}_{A_i}^{\text{pref}}$  is within  $\text{ORCA}_{A_i}^\tau$ . The robot may therefore keep  $\mathbf{v}_{A_i}^{\text{pref}}$  in the first step, and  $\mathbf{v}_{A_i}^{\text{pref}}(t)$  will be continuous by Lemma 2. ■

Finally, we have following result for differential-drive robots.

*Lemma 6:* If the trajectory  $\mathbf{v}_{A_i}(t)$  generated by the sequence of velocities  $\mathbf{v}_{A_i} = (\dot{X}, \dot{Y})_{A_i}$  of the effective center of the robot in the velocity space is continuous, then the trajectory generated by the sequence of velocities  $(\dot{x}, \dot{y})_{A_i}$  of the center of the robot is continuous.

*Proof:* By assumption,  $\mathbf{v}_{A_i} = (\dot{X}, \dot{Y})_{A_i}$  in (3) is continuous. Referring to equations (2), it is easy to show by induction on the time step  $\delta t$  that  $\theta$  is continuous when each robot is initialized as in the base case of the proof of Theorem 1. Therefore,  $\mathbf{u}_{A_i} = (v_l, v_r)_{A_i}$  is continuous. By equations (1), this implies that  $(\dot{x}, \dot{y})_{A_i}$  is continuous, as required. ■

Hence, by Theorem 1 and Lemma 6, our algorithm generates smooth trajectories for each differential-drive robot.

### B. Locally Collision-Free Paths

The guarantee given by [11] that optimal reciprocal collision avoidance generates locally collision-free paths also holds for differential-drive robots navigating using the notion of effective center and effective radius.

*Theorem 7:* Given a small time step  $\delta t$ , the path  $\mathbf{q}_{A_i}(t)$  generated by the sequence of positions  $\mathbf{q}_{A_i}$  in the plane is collision free with all paths  $\mathbf{q}_{A_j}(t)$  for  $j \neq i$ .

*Proof:* Choosing a velocity  $\mathbf{v}_{A_i}$  within the intersection of half-planes  $\text{ORCA}_{A_i}^\tau$  ensures that the disc  $D(\mathbf{p}_{A_i}, R_{A_i})$ , for effective center  $\mathbf{p}_{A_i}$  and effective radius  $R_{A_i}$ , is collision free. The robot  $A_i$  is always completely contained by this disc, so its path  $\mathbf{q}_{A_i}(t)$  is collision free. ■

We can only guarantee a collision-free velocity  $\mathbf{v}_{A_i}$  will be found when  $\text{ORCA}_{A_i}^\tau \neq \emptyset$ , although this has not been an issue in practice.

## VI. EXPERIMENTATION

In this section, we describe the implementation of our algorithm and present the results of our experiments with differential-drive robots in two scenarios.

### A. Implementation

We implemented our algorithm on a set of four *iRobot Create* programmable robots with Bluetooth wireless control and camera-based centralized sensing, as shown in Fig. 4(a).

The *iRobot Create* is a differential-drive robot with two powered wheels and a third passive caster wheel to maintain

balance. Each wheel is actuated individually with a maximum speed of 0.5 m/s in both forward and reverse directions. Weighing less than 2.5 kg, the favorable power-to-weight ratio of the robot allows it to accelerate rapidly to any speed specified by our algorithm.

The *iRobot Create* does not have sufficient sensors to be able to localize itself, so, for ease of implementation, we tracked fiducial markers attached to each robot (Fig. 4(a)) using a ceiling-mounted video camera connected to a standard desktop computer via FireWire interface. The images were captured at a resolution of 1024x768 and a refresh rate of 30 Hz and were processed using the *ARToolKit* augmented reality library [28] to determine the absolute position and orientation of each robot. Velocity was inferred from these measurements using a Kalman filter [29].

For convenience, all calculations were carried out on a single computer. However, to ensure that our approach is also applicable to a robot with its own on-board sensing and computing, only the acquisition of the localization data was performed centrally. Calculations for each robot related to navigation were performed in separate, non-interacting processes. The resultant control inputs were sent to each robot over a Bluetooth virtual serial connection at a speed of 57.6 kb/s and median latency of 0.5 s.

### B. Experimental Results

We tested our approach in two scenarios:

- 1) Four robots are placed at the corners of a rectangular environment. Their goal is to navigate to the corner diagonally opposite. The robots will meet and have to negotiate around each other in the middle.
- 2) The center of the environment is blocked by a dead robot that has malfunctioned and is unable to move. Scenario 1 is repeated, but the remaining robots must avoid each other and the dead robot.

Traces of the robots are shown in Fig. 4(b) for Scenario 1. The paths generated by our algorithm are collision free and do not exhibit any oscillations in velocity. Each of the four robots makes enough room for the other robots, resulting in direct paths from starting position to goal position with a minimal amount of deviation.

In Scenario 2, we demonstrate that our algorithm still generates smooth and collision-free paths, as shown in Fig. 4(c), even when the direct path from the starting position to the goal position for each robot is blocked by the dead robot.

## VII. CONCLUSION

In this paper, we have described a method for obtaining control inputs, the two wheel speeds, from a given velocity for a robot with differential-drive constraints using the notion of effective center and effective radius to overcome the inherent limitation that the center of a differential-drive robot is not fully controllable. We have combined this formulation with optimal reciprocal collision avoidance to derive our algorithm and proved that it guarantees smooth and locally collision-free motion for multiple differential-drive robots navigating in a shared environment. Each robot

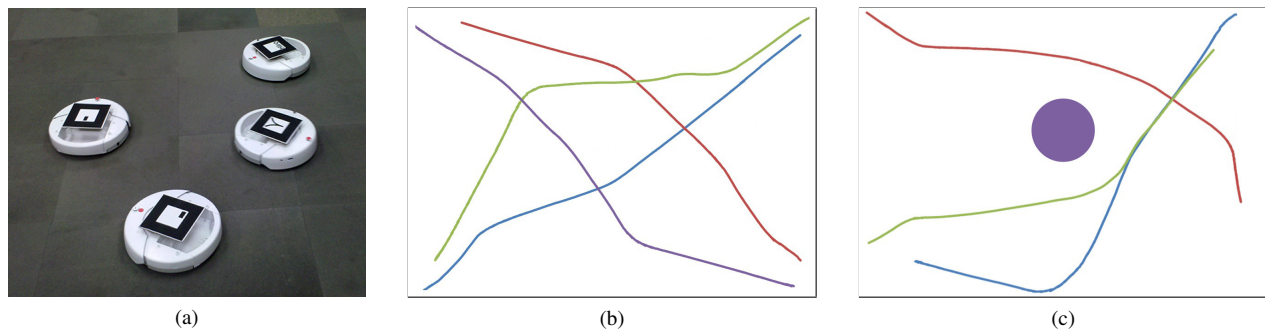


Fig. 4. (a) Four *iRobot Create* mobile robots in our experimentation setting. (b, c) Traces of the robots in Scenarios 1 and 2, respectively. The shaded disc in the center of (c) is the dead robot in Scenario 2.

is independent and is able to react to the other robots without explicit communication by simply observing their current positions and velocities. We have implemented our method using *iRobot Create* mobile robots and shown its effectiveness in two scenarios.

While other approaches [17], [27] exhibit empirically smooth trajectories in limited examples, they provide no mathematical guarantees that the trajectories will be smooth in other circumstances. Furthermore, our algorithm is not constrained to a finite set of behaviors [18], potentially allowing any maneuver permitted by the kinematics of each robot. Unlike other approaches which require explicit communication between every robot [19], robots using our algorithm can be fully independent, making all decisions based only on their own observations. This allows, for instance, fast computation and fault tolerance.

## REFERENCES

- [1] J. L. Jones, N. E. Mack, D. M. Nugent, and P. E. Sandin, "Autonomous floor-cleaning robot," U.S. Patent 6 883 201, 2005.
- [2] E. Prassler, J. Scholz, and P. Fiorini, "A robotic wheelchair for crowded public environments," *IEEE Robot. Autom. Mag.*, vol. 8, no. 1, pp. 38–45, 2001.
- [3] D. Longo and G. Muscato, "The Alicia<sup>3</sup> climbing robot: A three-module robot for automatic wall inspection," *IEEE Robot. Autom. Mag.*, vol. 13, no. 1, pp. 42–50, 2006.
- [4] P. Fiorini and D. Botturi, "Introducing service robotics to the pharmaceutical industry," *Intell. Serv. Robot.*, vol. 1, no. 4, pp. 267–280, 2008.
- [5] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2003, pp. 446–451.
- [6] N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multirobot teams," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 53–61, 2008.
- [7] K. Kant and S. W. Zucker, "Towards efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.
- [8] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.
- [9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, pp. 23–33, 1997.
- [10] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst.*, 2005, pp. 2210–2215.
- [11] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal  $n$ -body collision avoidance," in *Robotics Research*, ser. Tract. Adv. Robot., M. Kaneko and Y. Nakamura, Eds. Springer, 2010, vol. 66.
- [12] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, pp. 497–516, 1957.
- [13] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pac. J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.
- [14] S. M. LaValle, *Planning Algorithms*. Cambridge Univ. Pr., 2006.
- [15] J.-C. Latombe, "A fast path planner for a car-like indoor mobile robot," in *Proc. AAAI Nat. Conf. Artif. Intell.*, 1991, pp. 659–665.
- [16] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1928–1935.
- [17] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," in *Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst.*, 2009, pp. 5917–5922.
- [18] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Trans. Robot. Autom.*, vol. 23, no. 6, pp. 1170–1183, 2007.
- [19] K. E. Bekris, K. I. Tsianos, and L. E. Kavraki, "A decentralized planner that guarantees the safety of communicating vehicles with complex dynamics that replan online," in *Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst.*, 2007, pp. 3784–3790.
- [20] J. Snape and D. Manocha, "Navigating multiple simple-airplanes in 3D workspace," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010.
- [21] D. B. Edwards, T. A. Bean, D. L. Odell, and M. J. Anderson, "A leader-follower algorithm for multiple AUV formations," in *Proc. IEEE OES Auton. Underw. Veh.*, 2004, pp. 40–46.
- [22] P. Cheng, V. Kumar, R. Arkin, M. Steinberg, and K. Hedrick, "Cooperative control of multiple heterogeneous unmanned vehicles for coverage and surveillance," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, p. 12, 2009.
- [23] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, 2001.
- [24] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 199–217, 2002.
- [25] J. Bruce and M. Veloso, "Real-time multi-robot motion planning with safe dynamics," in *Multi-Robot Systems. From Swarms to Intelligent Automata*, L. E. Parker, F. E. Schneider, and A. C. Schultz, Eds. Springer, 2005, vol. 3, pp. 159–170.
- [26] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *Int. J. Robot. Res.*, vol. 24, no. 4, pp. 295–310, 2005.
- [27] B. Kluge, D. Bank, E. Prassler, and M. Strobel, "Coordinating the motion of a human and a robot in a crowded, natural environment," in *Advances in Human-Robot Interaction*, ser. Tract. Adv. Robot., E. Prassler, G. Lawitzky, A. Stopp, G. Grunwald, M. Hägele, R. Dillmann, and I. Iossifidis, Eds. Springer, 2004, vol. 14, pp. 207–219.
- [28] H. Kato and M. Billingham, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. IEEE ACM Int. Work. Augment. Real.*, 1999, pp. 85–94.
- [29] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 35–45, 1960.