Smooth Coordination and Navigation for Multiple Differential-Drive Robots

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

Abstract Multiple independent robots sharing the workspace need to be able to navigate to their goals while avoiding collisions with each other. In this paper, we describe and evaluate two algorithms for smooth and collision-free navigation for multiple independent differential-drive robots. We extend reciprocal collision avoidance algorithms based on velocity obstacles and on acceleration-velocity obstacles. We implement both methods on multiple iRobot Create differential-drive robots, and report on the quality and ability of the robots using the two algorithms to navigate to their goals in a smooth and collision-free manner.

1 Introduction and Motivation

We address the problem of computing smooth and collision-free motion for multiple differential-drive robots in the two-dimensional workspace. Differential-drive robots are widely used in real-world scenarios for tasks such as vacuum cleaning, e.g. iRobot Roomba [11], warehousing [8], virtual tour guides [21], powered wheelchairs [22], and inspecting pipelines [23].

The recent trend is to deploy multiple robots to tasks as a distributed system. Groups of coordinated differential-drive robots may be used for surveillance, environmental monitoring, and search and rescue applications [18]. Hence, it is important to develop methods that can be used for automatic navigation and coordination amongst multiple such robots.

Some of the recent work in robotics has focused on developing algorithms for collision-free motion using the notion of velocity obstacles [9] and its variants [5, 24]. We extend these methods to present two algorithms that also ensure smooth motion for each robot. The smoothness property is important for many robots, since

Jamie Snape, Stephen J. Guy, Jur van den Berg, and Dinesh Manocha
University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA
e-mail: {snape,_sjguy,_berg,_dm}@cs.unc.edu

they must take into account the physical limits of their actuators and other safety factors. We assume that the motion of each robot is computed in an independent manner, and that the robots need not communicate with each other. Furthermore, our algorithms do not make any assumptions about the intentions of other robots in the workspace. The main characteristics of the two algorithms are as follows:

1. We combine velocity obstacles, optimal reciprocal collision avoidance [4], and an enlarged planning radius [25] to allow navigation for differential-drive robots at both high and low speeds.
2. We combine acceleration-velocity obstacles [6], optimal reciprocal collision avoidance, and artificial acceleration constraints to allow navigation at high speeds without enlarging the planning radius of the robot.

We evaluate and demonstrate the effectiveness of the two algorithms by applying them to multiple iRobot Create differential-drive robots using wireless control and camera-based centralized sensing.

This rest of this paper is organized as follows. We begin by summarizing related prior work in Sect. 2. In Sect. 3, we describe velocity obstacles and acceleration-velocity obstacles for local collision avoidance and show how we apply them to reciprocal collision avoidance. In Sect. 4, we briefly review the kinematics of a differential-drive robot and outline the two methods that we use to transform velocities calculated using optimal reciprocal collision avoidance to wheel speeds for such robots. We describe our implementation using multiple iRobot Create differential-drive robots and discuss our experimental results in Sect. 5.

2 Prior Work

Prior work in robot motion planning has often focused on a single robot navigating through an environment shared with dynamic obstacles [10, 20]. A widely studied approach has been the velocity obstacle [9], which has been used for motion planning for robots with dynamic obstacles. This has been extended to reciprocal velocity obstacles [5] and other variants [1, 4, 15, 24] for reciprocal collision avoidance for multiple robots. Other work on navigating multiple robots has focused on follow-the-leader behavior [7], time-optimal trajectories [2], explicit communication between robots [3], and predefined discrete behaviors [19].

Most work on generating smooth trajectories has been limited to a single robot [16, 17]. Many other methods for global collision-free path computations are based on centralized [16, 17] or decoupled [13] algorithms.

3 Local Collision Avoidance

In this section, we describe the velocity obstacle and the acceleration-velocity obstacle for local collision avoidance, as well as the notion of optimal reciprocal collision avoidance for a velocity obstacle and its variants.

3.1 Notation

We adopt the following notation. Scalar values v are typeset in lowercase italics, vectors $\mathbf{v}$ are typeset in lowercase boldface, sets V of vectors are typeset in uppercase italics, and matrices $\mathbf{M}$ are typeset in uppercase boldface. In addition, the Euclidean length of a vector $\mathbf{p}$ is denoted by $\|\mathbf{p}\|_2$, and the open disc of radius r centered at $\mathbf{p}$ is denoted by

$$\mathrm{D}(\mathbf{p}, r) = \{\mathbf{q} \mid \|\mathbf{q} - \mathbf{p}\|_2 \in [0, r)\}.$$

For a disc-shaped robot A in the two-dimensional workspace, we denote by r_A the radius of the robot, by $\mathbf{p}_A$ the position of its center, by $\mathbf{v}_A$ the velocity of its center, and by $\mathbf{a}_A$ the acceleration of its center. Assuming a second robot B, similarly defined, we denote by $r_{AB} = r_A + r_B$ the combined radius of the robots, by $\mathbf{p}_{AB} = \mathbf{p}_A - \mathbf{p}_B$ the relative position of their centers, by $\mathbf{v}_{AB} = \mathbf{v}_A - \mathbf{v}_B$ the relative velocity of their centers, and by $\mathbf{a}_{AB} = \mathbf{a}_A - \mathbf{a}_B$ the relative acceleration of their centers.

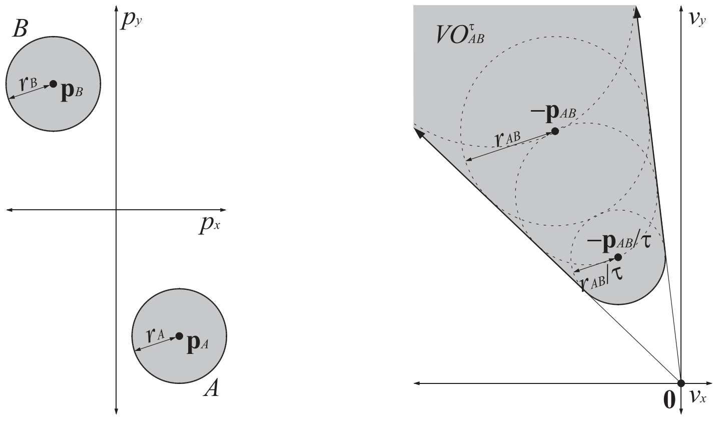
The velocities and accelerations of a robot are bounded such that $\mathbf{v} \in \mathrm{D}(\mathbf{0}, v^{\max})$ and $\mathbf{a} \in \mathrm{D}(\mathbf{0}, a^{\max})$, respectively. When we wish to emphasize a dependence on time, we write $\mathbf{p} = \mathbf{p}(t)$, $\mathbf{v} = \mathbf{v}(t)$, and $\mathbf{a} = \mathbf{a}(t)$, for the position, velocity, and acceleration, respectively, of a robot at time t.

3.2 Velocity Obstacles

Consider two disc-shaped robots A and B with radii, positions, velocities, and accelerations at any time t described using the notation above. The velocity obstacle [9] for A relative to B within the window of time τ is the set of velocities $\mathbf{v}_{AB}$ of A relative to B described by

$$VO^{\tau}_{AB} = \bigcup_{t \in [0,\tau]} \mathrm{D}\left(\frac{-\mathbf{p}_{AB}}{t}, \frac{r_{AB}}{t}\right).$$

This is the set of velocities of A relative to B that will cause a collision at some moment before time τ has elapsed, assuming that both robots maintain a constant velocity during that window of time. Geometrically, this set is a union of discs in the velocity space with decreasing radii as time elapses, such that the boundary of the



(a) (b)

Fig. 1 (a) A configuration of disc-shaped robots A and B in the two-dimensional workspace. (b) The velocity obstacle VO^{τ}_{AB} for A relative to B within the window of time $\tau = 2$ in the corresponding velocity space.

set is a truncated cone with apex at the origin and sides tangent to the Minkowski difference $B \ominus A$. Figure 1 shows a configuration of two robots A and B and the corresponding velocity obstacle of A relative to B.

It follows that if A chooses a velocity relative to B that is outside the velocity obstacle VO^{τ}_{AB}, then the robot will be collision free for at least τ time. Given that A has a finite maximum speed, the set of attainable local collision avoiding velocities $\mathbf{v}_{AB}$ for A relative to B within the window of time τ is

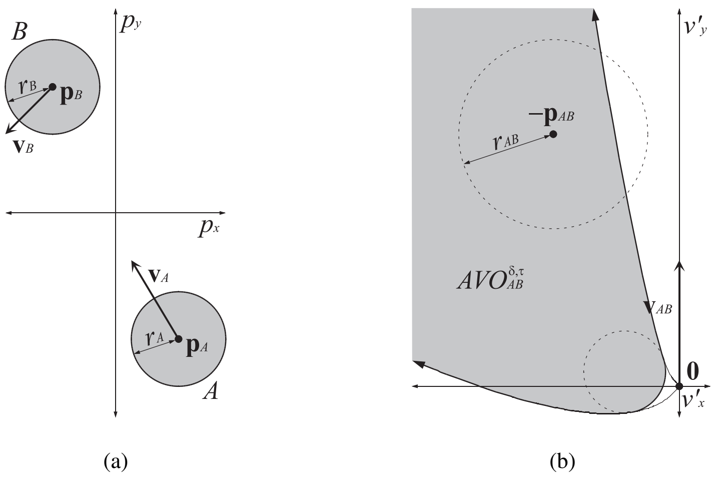
$$CA^{\tau}_{AB} = \mathrm{D}(\mathbf{v}_B, v_A^{\max}) \cap \{\mathbf{v} \mid \mathbf{v} \notin VO^{\tau}_{AB}\}.$$

3.3 Acceleration-Velocity Obstacles

It is clear from the definition above that the velocity obstacle does not allow for any bounds on the acceleration of a robot. Indeed, the robots should adopt their relative velocity instantaneously and, for the velocity obstacle to remain valid, maintain a constant velocity with the chosen window of time. The acceleration-velocity obstacle [6] addresses this limitation.

The acceleration-velocity obstacle for A relative to B with acceleration parameter δ within the window of time τ is the set of velocities $\mathbf{v}_{AB}$ of A relative to B described by

$$AVO^{\delta,\tau}_{AB} = \bigcup_{t \in [0,\tau]} \mathrm{D}\left(\frac{\delta(\mathrm{e}^{-t/\delta} - 1)\mathbf{v}_{AB} - \mathbf{p}_{AB}}{t + \delta(\mathrm{e}^{-t/\delta} - 1)}, \frac{r_{AB}}{t + \delta(\mathrm{e}^{-t/\delta} - 1)}\right),$$



(a) (b)

Fig. 2 (a) A configuration of disc-shaped robots A and B in the two-dimensional workspace. (b) The acceleration-velocity obstacle $AVO^{\delta,\tau}_{AB}$ for A relative to B with acceleration parameter $\delta = 2$ within the window of time $\tau = 4$ in the corresponding velocity space.

where δ is defined such that $\mathbf{a}_{AB}(t) = (\mathbf{v} - \mathbf{v}_{AB}(t))/\delta$ for some relative velocity $\mathbf{v}$ within the acceleration-velocity obstacle. This is the set of velocities of A relative to B that will cause a collision at some moment before time τ has elapsed where proportional control of the acceleration with parameter δ is used by both A and B to reach the relative velocity $\mathbf{v}$ from the current velocity $\mathbf{v}_{AB}$ of A relative to B. Similarly to the velocity obstacle, the acceleration-velocity obstacle is also a union of discs in the velocity space, albeit with the centers and radii varying with time such that the boundary of the set is a truncated cone with curved rather than straight sides due to the additional $\delta(\mathrm{e}^{-t/\delta} - 1)$ terms. Figure 2 shows a second configuration of two robots A and B and the corresponding acceleration-velocity obstacle of A relative to B.

In an analogous manner to the velocity obstacle, if A and B both have the same acceleration parameter δ and if A chooses a velocity relative to B that is outside the acceleration-velocity obstacle $AVO^{\delta,\tau}_{AB}$, then the robot will be collision free for at least τ time. Given that A has a finite maximum speed and A and B have finite maximum accelerations, the set of attainable local collision avoiding velocities $\mathbf{v}_{AB}$ for A relative to B with acceleration parameter δ within the window of time τ is

$$CA^{\delta,\tau}_{AB} = \mathrm{D}(\mathbf{v}_B, v^{\max}_A) \cap \mathrm{D}(\mathbf{v}_{AB}, \delta a^{\max}_{AB}) \cap \{\mathbf{v} \mid \mathbf{v} \notin AVO^{\delta,\tau}_{AB}\}.$$

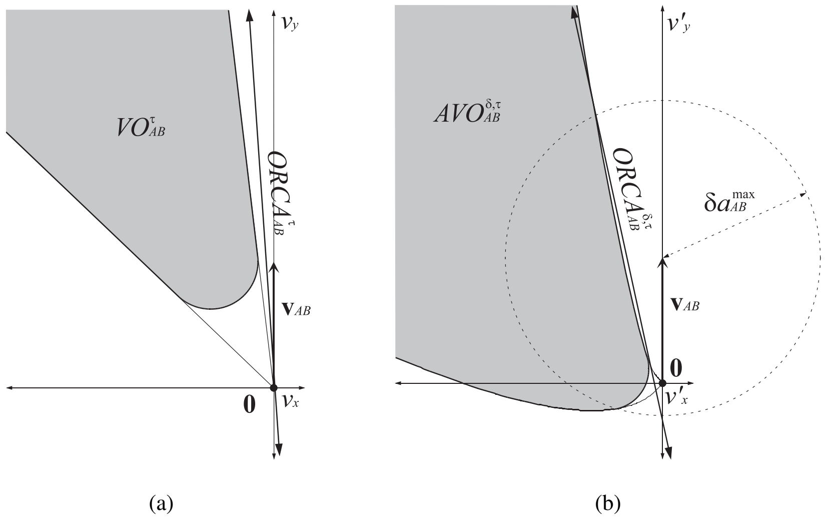


Fig. 3 (a) The optimal reciprocal collision avoidance half-plane $ORCA^{\tau}_{AB}$ for A relative to B within the window of time $\tau = 2$ in the velocity space. (b) The optimal reciprocal collision avoidance half-plane $ORCA^{\delta,\tau}_{AB}$ for A relative to B with acceleration parameter $\delta = 2$ within the window of time $\tau = 4$ in the velocity space.

3.4 Optimal Reciprocal Collision Avoidance

Optimal reciprocal collision avoidance [6] augments the construction of a velocity obstacle or its variants with an additional linear constraint that provides a half-plane of local collision avoiding velocities rather than those which lie outside a set which is bounded geometrically by a truncated cone. Reciprocity is important to ensure that oscillations in the motion of the robots are avoided.

For a velocity obstacle VO^{τ}_{AB}, the optimal reciprocal collision avoidance half-plane for A relative to B within the window of time τ is the set of velocities

$$ORCA^{\tau}_{AB} = \{\mathbf{v} \mid (\mathbf{v} - (\mathbf{v}_{AB} + \tfrac{1}{2}\mathbf{w})) \cdot \mathbf{n} \in [0, \infty)\},$$

where $\mathbf{w} = (\arg\min_{\mathbf{v} \in \partial VO^{\tau}_{AB}} \|\mathbf{v} - \mathbf{v}_{AB}\|_2) - \mathbf{v}_{AB}$ and $\mathbf{n}$ is the outward normal of the velocity obstacle boundary ∂VO^{τ}_{AB} at $\mathbf{v}_{AB} + \mathbf{w}$. The term $\frac{1}{2}\mathbf{w}$ represents the reciprocity between A and B. Each adapt their velocity by at least this quantity to avoid colliding with each other. The optimal reciprocal collision avoidance half-plane is a strict subset of the collision avoiding velocities CA^{τ}_{AB} of A relative to B. Figure 3(a) shows this half-plane $ORCA^{\tau}_{AB}$ for a velocity obstacle.

For the acceleration-velocity obstacle $AVO^{\delta,\tau}_{AB}$, we first take the convex hull of $\mathrm{D}(\mathbf{v}_{AB}, \delta a^{\max}_{AB}) \cap AVO^{\delta,\tau}_{AB}$ and then proceed as with the velocity obstacle. Figure 3(b) shows the optimal reciprocal collision avoidance half-plane $ORCA^{\delta,\tau}_{AB}$ for an acceleration-velocity obstacle.

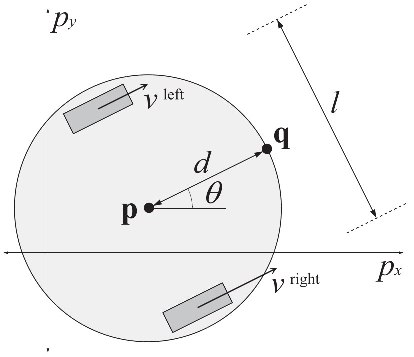


Fig. 4 The configuration of a disc-shaped differential-drive robot in the workspace.

If there are more than two robots in the workspace, we simply take the intersection of $\{\mathbf{v}_B\} \oplus ORCA^{\tau}_{AB}$ relative to every other robot B, that is

$$ORCA^{\tau}_{A} = \bigcap_{B \neq A} (\{\mathbf{v}_B\} \oplus ORCA^{\tau}_{AB}),$$

and similarly for $ORCA^{\delta,\tau}_{A}$.

4 Differential-Drive Constraints

In this section, we give a brief overview of the kinematic constraints of a differential-drive robot and describe two methods for transforming a given velocity to the two wheel speeds of the robot.

4.1 Overview

Figure 4 shows a differential-drive robot. The configuration of such a robot is given by its center $\mathbf{p} = (p_x, p_y)$ and its orientation θ. Its configuration transition equations are

$$\dot{p}_x = \frac{v^{\text{left}} + v^{\text{right}}}{2} \cos\theta, \qquad \dot{p}_y = \frac{v^{\text{left}} + v^{\text{right}}}{2} \sin\theta, \qquad \dot{\theta} = \frac{v^{\text{right}} - v^{\text{left}}}{l},$$

where $v^{\text{left}}, v^{\text{right}} \in [-v^{\max}, v^{\max}]$ are the left and right signed wheel speeds, respectively, of the robot and l is the distance between its wheels. Each wheel may take a different speed, hence the robot may spin in place and follow any continuous path in the workspace [17].

4.2 Enlarged Planning Radius

The center $\mathbf{p}$ of a differential-drive robot cannot immediately translate in a direction parallel to the axle of the robot. To counter this, when constructing velocity obstacles for the robots, we use a disc with an artificially enlarged planning radius [25] which encompasses the disc-shaped robot and is centered at a point forward a distance $d \in (0, \infty)$ from $\mathbf{p}$ in a direction perpendicular to the axle of the robot.

Denoting by $\mathbf{q} = (q_x, q_y)$ the center of the enlarged disc, we have

$$q_x = p_x + d\cos\theta, \qquad q_y = p_y + d\sin\theta.$$

Combining this with the configuration transition equations, it follows that

$$\mathbf{u} = \mathbf{M}(\theta)\begin{pmatrix} v^{\text{left}} \\ v^{\text{right}} \end{pmatrix},$$

where $\mathbf{u}$ is the velocity of the center of the enlarged disc and

$$\mathbf{M}(\theta) = \tfrac{1}{2}\begin{pmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \end{pmatrix} + \frac{d}{l}\begin{pmatrix} \sin\theta & -\sin\theta \\ -\cos\theta & \cos\theta \end{pmatrix}.$$

Since the wheel speeds of the differential-drive robot are bounded, $(v^{\text{left}}, v^{\text{right}})$ lies within an axis-aligned square S with sides of length $2v^{\max}$ and center $\mathbf{0}$ in the velocity space. Therefore, the set of velocities $\mathbf{u}$ that the robot can attain is given by $R = \mathbf{M}(\theta)\, S$. It follows that if $d = \frac{1}{2}l$, then this set of velocities is a square whose center lies at $(v^{\text{left}}, v^{\text{right}}) = \mathbf{0}$ and whose orientation depends on θ. The incircle $\mathrm{D}(\mathbf{0}, v^{\max})$ of R therefore contains the velocities that can be attained regardless of orientation θ.

Since we require $d \in (0, \infty)$, this approach is less suited for navigation in small or highly congested workspaces due to the enlarged planning radius of the robot, but will allow smooth and collision-free navigation at both low and high speeds.

4.3 Adding Acceleration Constraints

When using acceleration-velocity obstacles and optimal reciprocal collision avoidance, we can take a different approach. Denoting the average wheel speed of the robot by $v = (v^{\text{left}} + v^{\text{right}})\,/\,2$ and the difference in wheel speeds by $\omega = (v^{\text{right}} - v^{\text{left}})\,/\,l$, from the configuration transition equations, we have

$$\dot{p}_x = v\cos\theta, \qquad \dot{p}_y = v\sin\theta, \qquad \dot{\theta} = \omega,$$

Now, instead of artificially expanding the radius of the robot to allow us to couple differential-drive constraints with the acceleration-velocity obstacle, we add artificial constraints on acceleration. Specifically, these are $\dot{v} = a$, where $|a| \in [0, a^{\max}]$

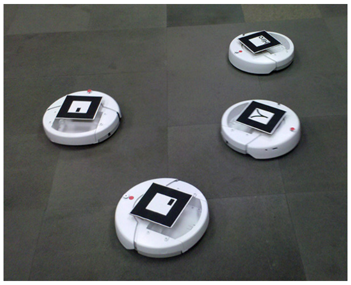


Fig. 5 iRobot Create differential-drive robots in our experimentation setting.

and $|v\omega| \in [0, a^{\max}]$. We do not constrain ω. Hence,

$$\mathbf{a} = \mathbf{M}(\theta) \begin{pmatrix} a \\ v\omega \end{pmatrix},$$

where

$$\mathbf{M}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$

By limiting the linear acceleration a and curvature $v\omega$ of the motion of the robot, it follows that if $v \neq 0$, then $(a, v\omega)$ lies within a square S with sides of length $2a^{\max}$ and center $\mathbf{0}$ in the acceleration space. Therefore, the set of accelerations $\mathbf{a}$ that the robot can attain is given by $R = \mathbf{M}(\theta)\,S$. The incircle $\mathrm{D}(\mathbf{0}, a^{\max})$ of R therefore contains the accelerations that can be attained regardless of orientation θ.

Since we require $v \neq 0$, this approach is less suited for navigation at low wheel speeds due to this singularity, but will allow smooth and collision-free navigation in both small and congested workspaces.

5 Experimentation and Results

In this section, we describe our implementation and the results of our experiments with multiple iRobot Create differential-drive robots.

5.1 Implementation Details

Our implementation was for iRobot Create programmable robots. These are shown in Fig. 5. The iRobot Create is a differential-drive robot with two individually actuated wheels and a third passive caster wheel for balance. The maximum speed of the robot is $0.5\,\mathrm{m\,s^{-1}}$ in both forward and reverse directions, its shape is circular with radius 0.17 m, and its mass of less than 2.5 kg allows the robot to accelerate to any speed computed by our algorithms within less than 2 s.

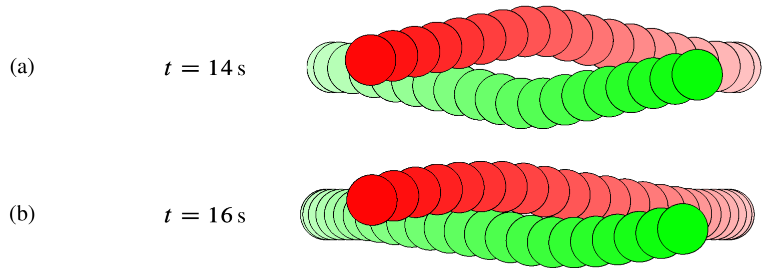


Fig. 6 Plots of the motion of the robots in the "chicken" scenario using (a) the velocity-obstacle-based algorithm and (b) the acceleration-velocity-obstacle-based algorithm.

The differential-drive robots were tracked using fiducial markers within a $6\,\mathrm{m}^2$ indoor space using a ceiling-mounted digital video camera connected to a standard desktop computer. The images were captured at a resolution of 1024x768 and refresh rate of $30\,\mathrm{s}^{-1}$, and processed using the ARToolKit augmented reality library [14] to determine position, with an absolute error of 0.01 m, and orientation. Velocity was computed from these measurements using a Kalman filter [12, 26].

All calculations were performed on a single computer. In order to ensure that our approach is applicable to a robot with its own on-board sensing and computing, only the acquisition of the localization data was performed centrally. All other calculations for each differential-drive robot were performed using separate and independent processes. The results of the algorithm, wheel speeds encoded in 4 b serial data packets, were sent to the robots over a Bluetooth virtual serial connection at a speed of $57.6\,\mathrm{kb\,s}^{-1}$ and average latency of 0.5 s.

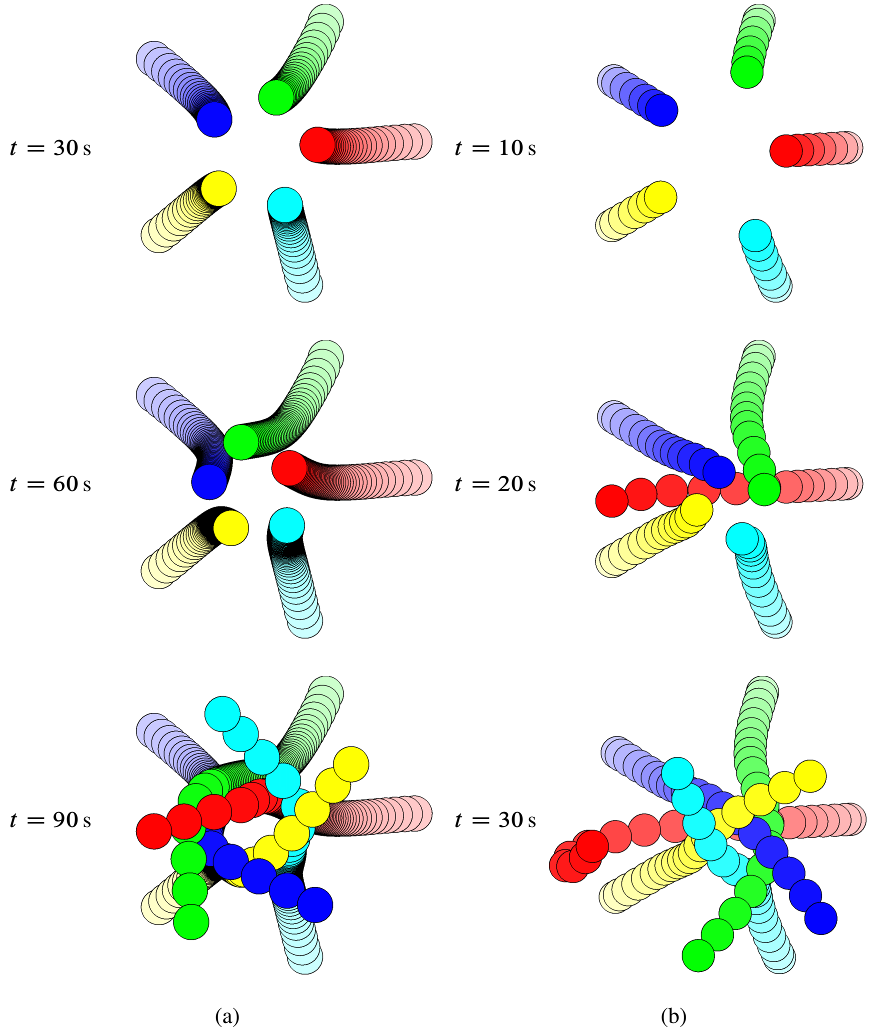
5.2 Experimental Results

We tested our two algorithms in the following scenarios:

Chicken Two robots are initialized at opposite ends of the workspace. They must pass each other to exchange positions.

Circle Five robots are initialized in a circle around the edge of the workspace. They must pass through the center of the circle to reach the diametrically opposite position on the edge of the circle.

Plots of the motion of the robots are shown in Fig. 6 for the "chicken" scenario and Fig. 7 for the "circle" scenario. The positions of the robots every two seconds are shown with a disc. Later positions are drawn on top of earlier positions, and in a darker shade. Figure 7 shows the evolution of the "circle" scenario through three equal time intervals.



(a) (b)

Fig. 7 Plots of the motion of the robots in the "circle" scenario using (a) the velocity-obstacle-based algorithm and (b) the acceleration-velocity-obstacle-based algorithm.

5.3 Discussion

Figure 6 shows that both algorithms generate smooth and collision-free motion for two differential-drive robots in the "chicken" scenario. Figure 6(a) shows the effect of the enlarged planning radius, since the two robots pass each other at approximately the halfway point with some distance between them. In Fig. 6(b), there is no enlarged radius, so the robots pass closely. It can also be seen that due to using acceleration-velocity obstacles, the paths of the robots are less symmetric, accelerating more slowly at the start of their paths and reaching their farthest distance laterally from a direct path from start to goal later. One issue captured less well in Fig. 6 is that the motion generated using acceleration-velocity obstacles is less smooth as the speeds of the robots are close to zero at the start of their paths due to the singularity at $v = 0$, as noted above. Both robots reach their goals within 14 s for the velocity-obstacle-based algorithm and 16 s for the acceleration-velocity-obstacle-based algorithm.

The differences between the two algorithms are more pronounced in Fig. 7 for the "circle" scenario. Again the robots in Fig. 7(b) pass more closely together than Fig. 7(a), however more noticeable is the disparity in the speed between the robots in the two algorithms. In Fig. 7(a), the robots generally all have a similar speed, although they slow in the center of the workspace to negotiate around each other. In Fig. 7(b), one robot accelerates more rapidly than the others and reaches its goal as the other robots approach the center of the workspace. Here the robots maintain a more even speed rather than slowing in the congestion. In this scenario, the robots again performed less well at slow speeds when using acceleration-velocity obstacles. All robots reach their goals within 90 s for the velocity-obstacle-based algorithm and 30 s for the acceleration-velocity-obstacle-based algorithm.

6 Insights and Conclusion

We have presented two algorithms based on velocity obstacles for generating smooth and collision-free motion for multiple differential-drive robots navigating in the two-dimensional workspace. Our first algorithm combines velocity obstacles, optimal reciprocal collision avoidance, and an enlarged planning radius to allow navigation at both high and low speeds. Our second algorithm combines acceleration-velocity obstacles, optimal reciprocal collision avoidance, and artificial acceleration constraints to allow navigation at high speeds without enlarging the planning radius of the robot.

Acknowledgements This work was supported in part by the Army Research Office under Contract W911NF-04-1-0088, by the National Science Foundation under Award 0636208, Award 0917040, and Award 0904990, by the Defense Advanced Research Projects Agency and U.S. Army Research, Development, and Engineering Command under Contract WR91CRB-08-C-0137, and by Intel Corporation.

References

1. Abe, Y., Yoshiki, M.: Collision avoidance method for multiple autonomous mobile agents by implicit cooperation. In: Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst., pp. 1207–1212 (2001)

2. Balkcom, D.J., Mason, M.T.: Time optimal trajectories for bounded velocity differential drive vehicles. Int. J. Robot. Res. **21**(3), 199–217 (2002)
3. Bekris, K.E., Tsianos, K.I., Kavraki, L.E.: 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., pp. 3784–3790 (2007)
4. v.d. Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Proc. Int. Symp. Robot. Res. (2009)
5. v.d. Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 1928–1935 (2008)
6. v.d. Berg, J., Snape, J., Guy, S.J., Manocha, D.: Reciprocal collision avoidance with acceleration-velocity obstacles. In: Proc. IEEE Int. Conf. Robot. Autom. (2011)
7. Desai, J.P., Ostrowski, J.P., Kumar, V.: Modeling and control of formations of nonholonomic mobile robots. IEEE Trans. Robot. Autom. **17**(6), 905–908 (2001)
8. Fiorini, P., Botturi, D.: Introducing service robotics to the pharmaceutical industry. Intell. Serv. Robot. **1**(4), 267–280 (2008)
9. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. Int. J. Robot. Res. **17**(7), 760–772 (1998)
10. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robot. Autom. Mag. **4**, 23–33 (1997)
11. Jones, J.L., Mack, N.E., Nugent, D.M., Sandin, P.E.: Autonomous floor-cleaning robot. U.S. Pat. 6,883,201 (2005)
12. Kalman, R.E.: A new approach to linear filtering and prediction problems. Trans. ASME J. Basic Eng. **82**, 35–45 (1960)
13. Kant, K., Zucker, S.W.: Towards efficient trajectory planning: The path-velocity decomposition. Int. J. Robot. Res. **5**(3), 72–89 (1986)
14. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proc. IEEE ACM Int. Work. Augment. Real., pp. 85–94 (1999)
15. Kluge, B., Prassler, E.: Reflective navigation: Individual behaviors and group behaviors. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 4172–4177 (2004)
16. Latombe, J.C.: Robot Motion Planning, *Springer Int. Ser. Eng. Comput. Sci.*, vol. 124. Springer (1991)
17. LaValle, S.M.: Planning Algorithms. Cambridge Univ. Pr. (2006)
18. Michael, N., Fink, J., Kumar, V.: Experimental testbed for large multirobot teams. IEEE Robot. Autom. Mag. **15**(1), 53–61 (2008)
19. Pallottino, L., Scordio, V.G., Bicchi, A., Frazzoli, E.: Decentralized cooperative policy for conflict resolution in multivehicle systems. IEEE Trans. Robot. Autom. **23**(6), 1170–1183 (2007)
20. Petti, S., Fraichard, T.: Safe motion planning in dynamic environments. In: Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst., pp. 2210–2215 (2005)
21. Philippsen, R., Siegwart, R.: Smooth and efficient obstacle avoidance for a tour guide robot. In: Proc. IEEE Int. Conf. Robot. Autom., vol. 1, pp. 446–451 (2003)
22. Prassler, E., Scholz, J., Fiorini, P.: A robotic wheelchair for crowded public environments. IEEE Robot. Autom. Mag. **8**(1), 38–45 (2001)
23. Roh, S., Choi, H.: Strategy for navigation inside pipelines with differential-drive inpipe robot. In: Proc. IEEE Int. Conf. Robot. Autom., vol. 3, pp. 2575–2580 (2002)
24. Snape, J., v.d. Berg, J., Guy, S.J., Manocha, D.: Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles. In: Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst., pp. 5917–5922 (2009)
25. Snape, J., v.d. Berg, J., Guy, S.J., Manocha, D.: Smooth and collision-free navigation for multiple robots under differential-drive constraints. In: Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst., pp. 4584–4589 (2010)
26. Welch, G., Bishop, G.: An introduction to the Kalman filter. Tech. Rep. 95-041, Univ. N. Carolina Chapel Hill (1995)