Reciprocal Collision Avoidance for Multiple Car-like Robots

Javier Alonso-Mora, Andreas Breitenmoser, Paul Beardsley and Roland Siegwart

***Abstract*— In this paper a method for distributed reciprocal collision avoidance among multiple non-holonomic robots with bike kinematics is presented. The proposed algorithm, *bicycle reciprocal collision avoidance (B-ORCA)*, builds on the concept of optimal reciprocal collision avoidance (ORCA) for holonomic robots but furthermore guarantees collision-free motions under the kinematic constraints of car-like vehicles. The underlying principle of the B-ORCA algorithm applies more generally to other kinematic models, as it combines velocity obstacles with generic tracking control. The theoretical results on collision avoidance are validated by several simulation experiments between multiple car-like robots.**

I. Introduction and related work

In this paper, a novel collision avoidance strategy for a group of car-like robots is presented. Various application areas throughout research and industry have seen an ever-growing interest in mobile robots. Industrial and service robots are mostly non-holonomic, and often designed as car-like vehicles. A particular example of car-like vehicles deployed in an industrial setting are the MagneBikes [1], compact robots with bicycle kinematics designed for the collaborative inspection in power plants. This and all other applications, where multiple car-like robots interact in their workspaces, require reciprocal collision avoidance methods.

Moving a vehicle on a collision-free path is a well-studied problem in robot navigation. The work in [2], [3] and [4] presents representative examples of collision avoidance methods for single mobile robots. Basically, similar approaches as in the single robot cases can be applied in the context of collision avoidance for multiple robots. However, the increase in robot density and collaborative interaction needs methods that scale well with the number of robots and avoid collisions as well as oscillations. The collision avoidance approaches are extended in [5] among others for multiple robots by decoupling path planning and coordination. In this line, [6] presented a method based on velocity profiles and scheduling to navigate several cars in a common environment. Collisions are then avoided but some of the cars need to pause and stop completely to let others move ahead freely. Other work investigated potential fields [7] and cooperative control laws [8] to direct a group of robots to their objectives while avoiding collisions. Decentralized control helps lowering computational cost and introduces additional robustness and flexibility to the multi-robot system. The problem of navigating car-like robots in dynamic scenarios has also been studied, with a great interest in navigation among humans [9]. A successful approach for this kind of scenarios based on a dynamic window was proposed in [10].

Our approach builds on Optimal Reciprocal Collision Avoidance (ORCA) [11] for holonomic robots and extends it to robots with car-like kinematics by using a trajectory tracking control [12], which is specific for this type of kinematics. However, the concepts here proposed apply to other kinematic models in general since the trajectory tracking controller is seen as a module that can be replaced to adapt the collision avoidance method to the particular kinematics of other systems. ORCA is a collaborative collision avoidance method based on velocity obstacles, where each holonomic robot makes a similar collision avoidance reasoning and collision-free motion is guaranteed without oscillations. Furthermore, in our approach, ORCA could be substituted by other sampling-based collision avoidance methods, such as Reciprocal Velocity Obstacles [13] or Hybrid Reciprocal Velocity Obstacles [14].

A formal extension of ORCA to differentially-driven robots was presented by the authors in [15]. That work shares with this paper the idea of extending ORCA to robots with non-holonomic kinematics by tracking a holonomic trajectory. ORCA was also extended to navigating simple airplanes with car-like kinematics in 3D space [16], where a set of trajectories is precomputed. Nevertheless, safety is not fully guaranteed as collisions may arise in the transient before reaching the desired velocity. In this paper we introduce a formal approach where this is taken into account by enlarging the radius of the robots. As an alternative, [17] presented the acceleration velocity obstacles for agents with holonomic acceleration capabilities, which explicitly takes into account acceleration limits and results in trajectories with continuous velocity (this was not the case for RVO and ORCA). Nevertheless, it does not generalize to general kinematics and cannot be directly applied to car-like vehicles. In contrast, in our approach the continuity in velocity and actuators is achieved thanks to the trajectory tracking strategy.

In contrast to purely deterministic methods, in [18] a method for recursive probabilistic velocity obstacles is studied, and in [19] collision-free trajectories are found by using Gaussian processes.

The remainder of the paper is structured as follows. Section II gives an overview of our collision avoidance algorithm. Section III describes the kinematics of the robot, whereas Section IV presents the trajectory tracking controller and Section V gives an overview of optimal reciprocal

This work was partially supported by ALSTOM.

J. Alonso-Mora, A. Breitenmoser and R. Siegwart are with the Autonomous Systems Lab, ETH Zurich, 8092 Zurich, Switzerland {jalonso,andrbrei,rsiegwart}@ethz.ch

J. Alonso-Mora, P. Beardsley are with Disney Research Zurich, 8092 Zurich, Switzerland {jalonso,pab}@disneyresearch.com

collision avoidance for holonomic robots. In Section VI the B-ORCA algorithm is described in detail. In Section VII the simulation experiments are discussed. Finally, Section VIII concludes and gives an outlook on our future work.

II. Overview of the B-ORCA algorithm

Bicycle reciprocal collision avoidance (B-ORCA) presents an efficient method for avoiding collisions in a scenario with multiple car-like robots. The method is fully distributed and the information required by each robot in order to avoid collisions includes the position, velocity and radius of its neighbors. The B-ORCA algorithm does not only offer oscillation-free reciprocal collision avoidance among multiple possibly heterogeneous robot units (i.e. the robot kinematics may not be of the same type), but also avoids collisions with dynamic and static obstacles.

Likewise to [15], the main idea is that a robot with given kinematic constraints is able to track a holonomic trajectory within a certain maximum error bound. Therefore, by enlarging the radius of the robot by this bound, it can be treated as holonomic. In this case, a collision-free trajectory is efficiently computed following [11]. By using a standard trajectory tracking controller [12] and precomputing the maximum tracking errors, a set of holonomic trajectories is obtained that can be tracked within the given maximum error bound. This set is introduced as a further constraint in the selection of collision-free inputs for the robot. Furthermore, the controller of [12] guarantees continuity in the driving velocity and acceleration of the robot, as well as in the steering angle, and respects the kinematic limits (maximum driving velocity, driving acceleration, steering angle and steering velocity) of the vehicle. Nevertheless, likewise to ORCA, a circular robot-shape is required.

III. Robot kinematic model

In this work the robots are considered to be non-holonomic car-like vehicles. A simplified car model with a fixed rear wheel and a steerable front wheel, as shown in Figure 1, is used. The generalized coordinates are $q = (x, y, \theta, \phi)$, where x, y represent the position of the rear wheel, θ the orientation of the car and ϕ the steering angle. If the car of length L has rear-wheel driving, the kinematic model is given (in accordance with [12]) by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi/L \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2, \tag{1}$$

where v_1 and v_2 are the driving and steering velocity inputs, respectively. The model singularity at $\phi = \pm\pi/2$ is avoided by restricting the range of the steering angle to $|\phi| < \phi^{max} < \pi/2$. Furthermore, both inputs are limited to $|v_1| \leq v_1^{max}$ and $|v_2| \leq v_2^{max}$, as well as the driving acceleration $|\dot{v}_1| \leq a_1^{max}$.

The parameters of the bicycle robots (see Section IV-D) used in the simulation experiments of this work are those of the inspection robot MagneBike as described in [20] and those of a faster car-like vehicle.

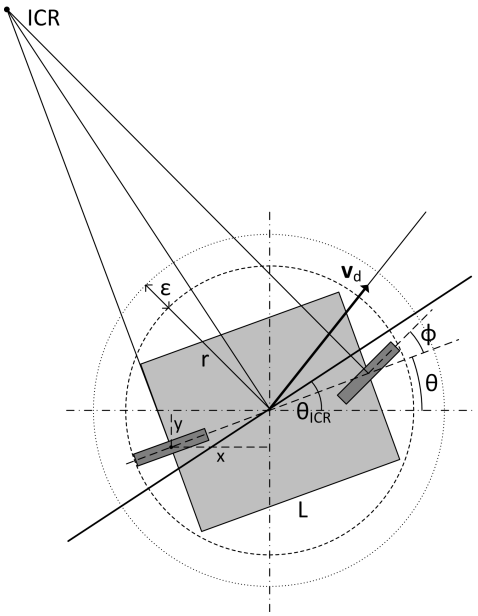

Fig. 1. Schema of a car-like robot, with extended radius ϵ and desired velocity $\mathbf{v}_d$. Its middle point is denoted by $\mathbf{p}$.

IV. Trajectory tracking

One of the underlying concepts of the B-ORCA algorithm is that a car-like robot tracks a constant-speed straight-line trajectory while staying within a known tracking error.

A. Trajectory tracking controller

The trajectory tracking controller [12] is obtained by applying full-state linearization via dynamic feedback to the non-linear system of Equation (1).

The two system outputs and their derivatives are given by

$$\mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \dot{\mathbf{z}} = \begin{bmatrix} \xi_1 \cos\theta \\ \xi_1 \sin\theta \end{bmatrix},$$
$$\ddot{\mathbf{z}} = \begin{bmatrix} -\xi_1^2 \tan\phi \sin\theta/L + \xi_2 \cos(\theta) \\ \xi_1^2 \tan\phi \cos\theta/L + \xi_2 \sin(\theta) \end{bmatrix}, \tag{2}$$

with ξ_1 and ξ_2 two integrators added to the system. It can be seen that the dynamic controller takes the form

$$\begin{aligned} v_1 &= \xi_1 \\ v_2 &= -3\xi_2 \cos\phi^2 \tan\phi/\xi_1 - Lr_1 \cos\phi^2 sin\theta/\xi_1^2 \\ &\quad + Lr_2 \cos\phi^2 \cos\theta/\xi_1^2 \\ \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_1^3 \tan\phi^2/L^2 + r_1 \cos\theta + r_2 \sin\theta, \end{aligned} \tag{3}$$

where the feedback terms r_i ($i = 1, 2$) are given by

$$r_i = \dddot{z}_{d,i} + k_{a,i}(\ddot{z}_{d,i} - \ddot{z}_i) + k_{v,i}(\dot{z}_{d,i} - \dot{z}_i) + k_{p,i}(z_{d,i} - z_i), \tag{4}$$

where z_d, $\dot{z}_d$, $\ddot{z}_d$ and $\dddot{z}_d$ are computed for the desired trajectory to track (see Section IV-B). The feedback gains are such that the polynomials

$$\lambda^3 + k_{a,i}\lambda^2 + k_{v,i}\lambda + k_{p,i}, \quad i = 1, 2, \tag{5}$$

are Hurwitz (all roots of the polynomial are real negative).

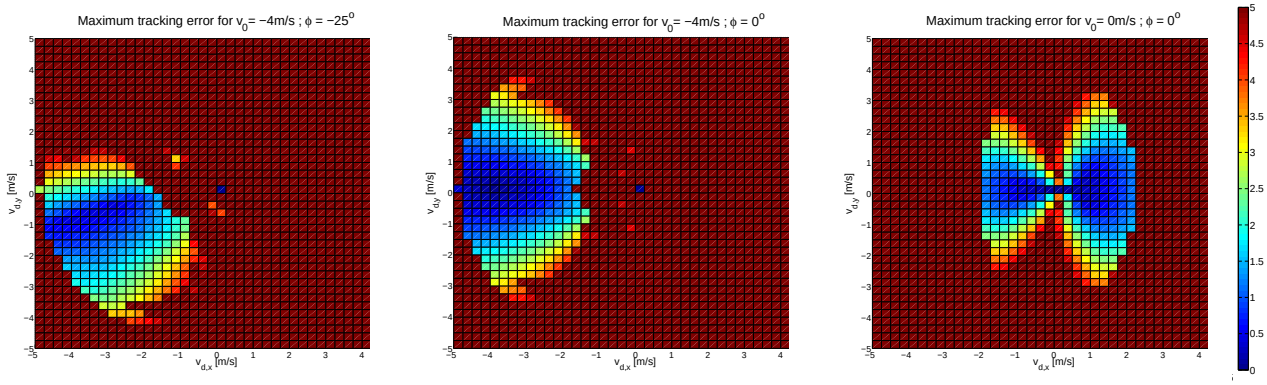

Fig. 2. Maximum tracking errors in $[m]$ for a desired trajectory given by $\mathbf{v}_d \in \mathcal{V}$ and saturated at 5m. From left to right, the initial driving velocity v_0 and the steering angle ϕ vary from $(v_0,\, \phi) = (-4\text{m/s},\, -25\,^\circ)\,, (-4\text{m/s},\, 0\,^\circ)$ to $(0\text{m/s},\, 0\,^\circ)$. Images best viewed in color.

Furthermore, recall [12] that this controller allows for parking maneuvers. Constraints in maximum steering angle, driving and steering velocity inputs and driving acceleration are directly added by saturating the respective variables (ϕ, ξ_1, v_2 and ξ_2).

B. Tracking of constant-speed straight-line trajectory

Due to position and rotation invariance, consider a car initially centered at the origin ($\mathbf{p}(0) = \mathbf{0}$) and with orientation $\theta(0) = 0$. Consider a desired straight-line trajectory given by a constant velocity $\mathbf{v}_d$ and passing through $\mathbf{p}(0)$. Denote $v_d = \|\mathbf{v}_d\|$ and $\theta_d = \text{atan2}\,(\mathbf{v}_d)$. The feedback terms of Equation (4) are then given by

$$\begin{aligned} r_1(t) = \quad & -k_{a,1}\ddot{z}_1(t) + k_{v,1}(v_d\cos\theta_d - \dot{z}_1(t)) \\ & +k_{p,1}((v_d t - s_1 L/2)\cos\theta_d - z_1(t)) \\ r_2(t) = \quad & -k_{a,2}\ddot{z}_2(t) + k_{v,2}(v_d\sin\theta_d - \dot{z}_2(t)) \\ & +k_{p,2}((v_d t - s_1 L/2)\sin\theta_d - z_2(t)), \end{aligned} \tag{6}$$

where $s_1 = 1$ if the car to be tracked is considered to move forward and $s_1 = -1$ otherwise. This ambiguity appears because the trajectory to be tracked is given with respect to the center of the robot, whilst the controller is designed for rear-wheel tracking.

The initial conditions of the variables are given by

$$\mathbf{z}(0) = \begin{bmatrix} \frac{L}{2}\cos\theta(0) \\ \frac{L}{2}\sin\theta(0) \end{bmatrix}; \quad \xi(0) = \begin{bmatrix} v_0 \\ a_0 \end{bmatrix} \tag{7}$$

where $v_0 = v_1(0)$ and $a_0 = \dot{v}_1(0)$ are the driving velocity and acceleration respectively. In our implementation, we choose $s_1 = \text{sign}\,(\cos\theta_d\cos\theta_{ICR} + \sin\theta_d\sin\theta_{ICR})$ where $\theta_{ICR} = \text{sign}\,(\phi)(\pi/2 - |\text{atan}\,(2/|\tan\phi|)|) + \theta(0)$ is the angle between the abscissa and the perpendicular to the line formed by the instantaneous center of rotation (ICR) and the middle point $\mathbf{p}(0)$ of the vehicle at initial time. As a further simplification, in our experiments, we consider $a_0 = 0$, thus guaranteeing continuity in velocity but not in acceleration.

Despite the initialization, it may occur that the tracking robot and the tracked virtual car move with opposite orientations, i.e. one forward and one backward. This would lead to perfect tracking of the rear wheel but large error in the tracking of the reference robot center point. In order to compensate, if this situation is detected ($\cos\theta_d\cos\theta(t) + \sin\theta_d\sin\theta(t) < 1$), the velocity of the tracked point $\mathbf{z}_d$ is temporally increased, or decreased respectively, until the orientation of the reference car is reversed. Note that the center point of the reference car always moves at speed v_d.

C. Achievable velocities

Given the initial conditions of the robot (initial driving velocity v_0 and steering angle ϕ) and the desired velocity $\mathbf{v}_d \in \mathcal{V} \subset \mathbb{R}^2$, its trajectory subject to the controller presented in this section is simulated and the maximum tracking error in the robot center point is computed. For given ϕ and v_0, the set of precomputed tracking errors for $\mathbf{v}_d \in \mathcal{V}$ is denoted by $\mathcal{E}_{\phi,v_0}$. Consider

$$\mathcal{V}_{\phi,v_0,\varepsilon} = \{\mathbf{v}_d \in \mathcal{V} \,|\quad \mathcal{E}_{\phi,v_0}(\mathbf{v}_d) \leq \varepsilon\}, \tag{8}$$

the subset of $\mathcal{V}$ of velocities that can be tracked with an error lower than ε (computed with respect to the robot center point).

We consider the discretizations $\mathcal{V} = [-v_1^{max} : \Delta v_1 : v_1^{max}]^2$, $\phi \in \Phi = [-\phi^{max} : \Delta\phi : \phi^{max}]$ and $v_0 \in \mathcal{V}_0 = [-v_1^{max} : \Delta v_1 : v_1^{max}]^2$. For $\phi \in \Phi$, $v_0 \in \mathcal{V}_0$ and $\mathbf{v}_d \in \mathcal{V}$, the trajectories of the car-like robot are simulated, and the maximum tracking errors precomputed and stored in a look-up table. Note that this computation is expensive, but is done off-line and only once for the kinematics of a given robot. In our simulations, the feedback gains of Equation (5) are computed such that all roots equal to -1 (MagneBike) and -2.5 (fast car).

In Figure 2, the maximum tracking errors obtained for the kinematics of the fast car are visualized for $(v_0,\, \phi) = (-4\text{m/s},\, -25\,^\circ)\,, (-4\text{m/s},\, 0\,^\circ)$ and $(0\text{m/s},\, 0\,^\circ)$. Note that due to symmetry, the tracking errors only need to be computed for one half of the full range of steering angles ϕ, e.g. $\phi \in [-\phi^{max},\, 0\,^\circ]$. However, the same does not hold true for the driving velocities.

Figure 3 shows the tracking errors for the MagneBike robot. Here continuity in speed is not imposed, which results

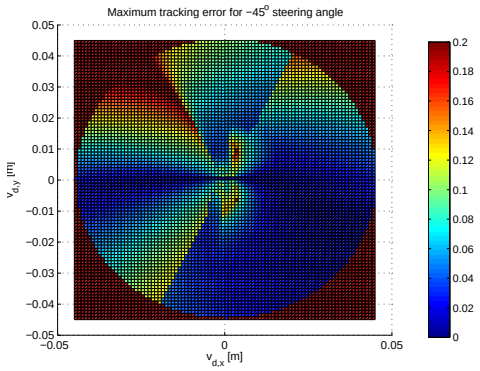

Fig. 3. Maximum tracking errors in $[m]$ for a desired trajectory given by $\mathbf{v}_d \in \mathcal{V}$, here shown for $\phi = -45\,^\circ$. In the case of MagneBike, v_0 is set to $s_1 v_d$, and thus precomputation includes varying steering angles only.

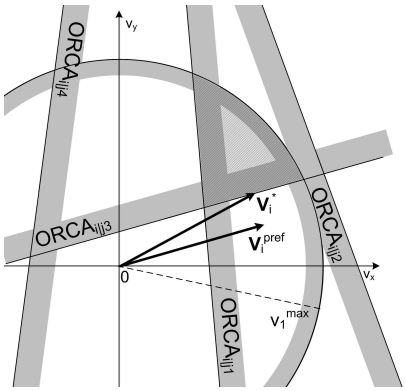

Fig. 4. Constraints in velocity space generated by $ORCA^\tau_{i|j}$ from multiple robots. The region of collision-free velocities $ORCA^\tau_i$ is highlighted and $\mathbf{v}^*_i$ is displayed.

in sets that are clearly different from the sets of the car illustrated in Figure 2 (see also Section IV-D below).

It is observed that the areas of best tracking are strongly related to the steering angle of the front wheel, which has an impact in the maneuverability of the robot.

D. Parameters for the simulated vehicles

The parameters for the simulated vehicles are as follows.

1) Car: $\phi^{max} = 30^o$, $v_1^{max} = 5$m/s, $v_2^{max} = 30^o$/s, $a_1^{max} = 2\text{m/s}^2$, $L = 2$m, $\Delta\phi = 1^o$, $\Delta v_1 = 0.25$m/s.

2) MagneBike: $\phi^{max} = 85^o$, $v_1^{max} = 0.045$m/s, $v_2^{max} = 33^o$/s, $L = 0.25$m, $\Delta\phi = 5^o$, $\Delta v_1 = 0.0025$m/s.

For the MagneBike, unconstrained acceleration is considered. To allow for discontinuities in driving velocity, in Equation (7), the initial conditions may be rewritten as $\xi(0) = [s_1 v_d, 0]$.

V. Reciprocal collision avoidance for holonomic robots

B-ORCA relies on the concept of Optimal Reciprocal Collision Avoidance (ORCA) for holonomic robots presented by [11]. In this section an overview of ORCA is given.

Consider a group of N disk-shaped robots with radius r_i at position $\mathbf{p}_i$ and with current velocity $\mathbf{v}_i$. Further consider each robot has a preferred velocity $\mathbf{v}_i^{pref}$ towards its goal position.

The velocity obstacle for robot $i \in [1, N]$ induced by any other robot $j \neq i$ is defined as the set of relative velocities $\bar{\mathbf{v}} = \mathbf{v}_i - \mathbf{v}_j$ leading to a collision within a time-horizon τ

$$VO^\tau_{i|j} = \left\{ \bar{\mathbf{v}} \,|\, \exists t \in [0,\tau]\,,\; t \cdot \bar{\mathbf{v}} \in D(\mathbf{p}_j - \mathbf{p}_i,\, r_i + r_j) \right\}\,, \tag{9}$$

where $D(\mathbf{p}, r) = \{\mathbf{q} \,|\, \|\mathbf{q} - \mathbf{p}\| < r\}$ is the open ball of radius r centered at $\mathbf{p}$. The set of collision-free velocities $ORCA^\tau_{i|j}$ for robot i with respect to robot j can be geometrically constructed from $VO^\tau_{i|j}$. First, the minimum change in velocity

$$\mathbf{u} = (\underset{\bar{\mathbf{v}} \in \partial VO^\tau_{i|j}}{argmin} \|\bar{\mathbf{v}} - (\mathbf{v}_i^{opt} - \mathbf{v}_j^{opt})\|) - (\mathbf{v}_i^{opt} - \mathbf{v}_j^{opt})\,, \tag{10}$$

which needs to be added to $\bar{\mathbf{v}}$ to avoid a collision, is computed. $\mathbf{v}_i^{opt}$ is the optimization velocity, set to the current velocity $\mathbf{v}_i$ of the robot. From our experience with ORCA, this choice gives good results. It can further be seen that

$$ORCA^\tau_{i|j} = \{\mathbf{v}_i | (\mathbf{v}_i - (\mathbf{v}_i^{opt} + \lambda_{i,j}\mathbf{u})) \cdot \mathbf{n} \geq 0\} \tag{11}$$

where $\mathbf{n}$ denotes the outward normal of the boundary of $VO^\tau_{i|j}$ at $(\mathbf{v}_i^{opt} - \mathbf{v}_j^{opt}) + \mathbf{u}$, and $\lambda_{i,j}$ defines how much each robot gets involved in avoiding a collision (where $\lambda_{i,j} + \lambda_{j,i} = 1$). $\lambda_{i,j} = \lambda_{j,i} = 0.5$ means both robots help to equal amounts to avoid colliding with each other; $\lambda_{i,j} = 1$ means robot i fully avoids collisions with a dynamic obstacle j. Likewise, the velocity obstacle can be computed for static obstacles [11].

$ORCA^\tau_i$, the set of collision-free velocities for robot i is then given by

$$ORCA^\tau_i = D(0, V_i^{max}) \cap \bigcap_{j \neq i} ORCA^\tau_{i|j}\,, \tag{12}$$

Figure 4 shows the set $ORCA^\tau_i$ for a configuration with multiple robots.

The optimal collision-free velocity for robot i is given by

$$\mathbf{v}^*_i = \underset{\mathbf{v} \in ORCA^\tau_i}{argmin} \|\mathbf{v} - \mathbf{v}_i^{pref}\|. \tag{13}$$

This optimization with linear constraints can be efficiently solved, returning a convex and compact set $ORCA^\tau_i$ and a collision-free velocity $\mathbf{v}^*_i$. In order to avoid reciprocal dances, one of the sides of $VO^\tau_{i|j}$ may slightly be enlarged to avoid the symmetry. In our case, $VO^\tau_{i|j}$ is enlarged by 0.001m/s to one side.

VI. The B-ORCA algorithm

The B-ORCA method first of all precomputes the tracking errors $\mathcal{E}_{\phi,v_0}$ with respect to the straight-line trajectories defined by the velocity vectors $\mathbf{v}_d \in \mathcal{V}$ for all possible initial steering angles $\phi \in \Phi$ and initial velocities $v_0 \in \mathcal{V}_0$ following Section IV-B. In this step the kinematics of the robot are taken into account. As the velocities to be tracked are considered relative to a robot's orientation, the

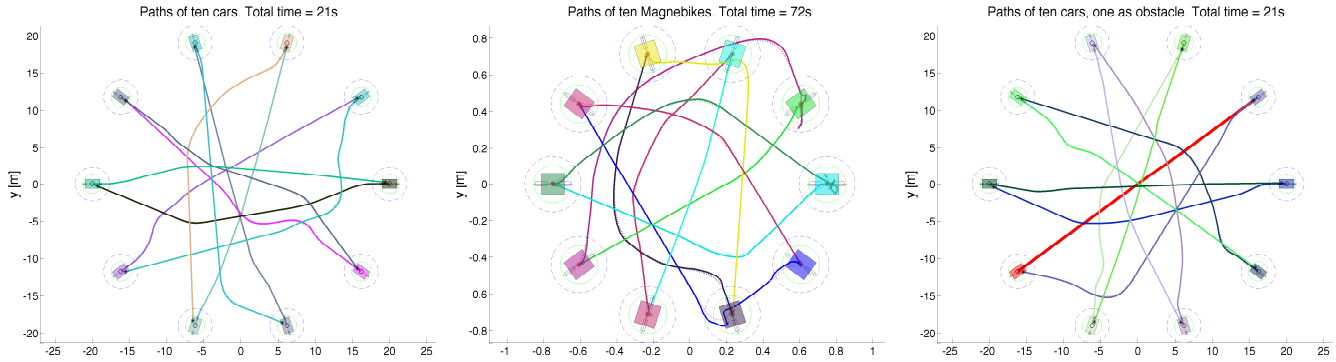

Fig. 5. Trajectories of ten car-like robots exchanging antipodal positions on a circle. **Left:** Experiment 1 with car-like vehicles. **Middle:** Experiment 1 with MagneBikes. **Right:** Experiment 2, where one car is non-reactive (straight-line trajectory in red), thus ignoring the other robots.

previously obtained tracking errors are not only invariant to the position of the robot but also to its current orientation. In the following, $\mathcal{E}_{\phi_i,v_0}$ is expressed in a relative frame oriented with θ_i, whilst $ORCA_i^\tau$ is computed in the global reference frame.

In every iteration of the collision avoidance stage, each robot reads out its sensors and gains knowledge about its internal state, given by its position $\mathbf{p}_i$, orientation θ_i, steering angle ϕ_i, current velocity $\mathbf{v}_i$, preferred velocity $\mathbf{v}_i^{pref}$, current driving velocity $v_0 = v_1$, radius r_i and desired radius extension $\hat{\varepsilon}_i$. Furthermore, each robot obtains from its neighbors via communication or sensing their position $\mathbf{p}_j$, current velocity (or velocity estimate) $\mathbf{v}_j$ and extended radius $r_j + \varepsilon_j$.

Given a group of N robots, with known aggressiveness $\lambda_{i,j}$, fixed maximum time to collision τ_{max} and sensing range d_{max}, assume a known fixed update rate of the controller of dt_c and of the sensing of dt_s, with $dt_c << dt_s$.

The following steps are computed independently by each robot in every iteration:

1) A preferred velocity $\mathbf{v}_i^{pref}$ towards the goal is obtained.
2) The extended radius $r_i + \varepsilon_i$ is set to $r_i + \min_j\left(\hat{\varepsilon}_i, (d(i,j) - r_i - r_j)/2\right)$, where $d(i,j)$ denotes the distance (middle points) from robot i to robot j.
3) All robots (including robot i) within d_{max} are considered as holonomic robots of radius $r_j + \varepsilon_j$. Following Section V the set $ORCA_i^\tau$ is computed.
4) A new collision-free velocity $\mathbf{v}_i$ is computed, such that it is closest to $\mathbf{v}_i^{pref}$ and such that it verifies $\mathbf{v}_i \in ORCA_i^\tau \cap \mathcal{V}_{\phi_i,v_0,\varepsilon_i}$. Thus,

$$\mathbf{v}_i = \underset{\mathbf{v}\in ORCA_i^\tau \cap \mathcal{V}_{\phi_i,v_0,\varepsilon_i}}{argmin} \|\mathbf{v} - \mathbf{v}_i^{pref}\|. \qquad (14)$$

5) The trajectory given by $\mathbf{v}_i$ is tracked with control update rate dt_c, as described in Section IV.

If $ORCA_i^\tau \cap \mathcal{V}_{\phi_i,v_0,\varepsilon_i} = \emptyset$, the time to collision τ_{max} is reduced ($\tau_{max} = \tau_{max}/2$), and steps 3) and 4) are repeated. If τ_{max} reaches a minimum admissible value $\tau_{max}^{min} \geq v_1^{max}/a_1^{max}$, the problem is considered unfeasible and robot i decelerates at maximum acceleration. If this is the case for robot i, all other robots must fully avoid collisions with it in the coming time steps while its optimization remains unfeasible; this is achieved by temporally setting $\lambda_{j,i} = 1$ for every other robot j.

A. Implementation details on step 4) of B-ORCA

Depending on the complexity of $\mathcal{V}_{\phi_i,v_0,\varepsilon_i}$, two options are discussed below.

1) Polygonal approximation of $\mathcal{V}_{\phi_i,v_0,\varepsilon_i}$: Likewise to [15], the set $\mathcal{V}_{\phi_i,v_0,\varepsilon_i}$ may be approximated by a convex polygon $\mathcal{P}_{\phi_i,v_0,\varepsilon_i} \subset \mathcal{V}_{\phi_i,v_0,\varepsilon_i}$ (or by two convex polygons respectively). If the approximation is accurate, step 4) of B-ORCA can be efficiently computed as an optimization with linear constraints given by $\mathcal{P}_{\phi_i,v_0,\varepsilon_i}$ and $ORCA_i^\tau$. This is the case for the sets depicted in Figure 2.

2) Sampling of $\mathcal{V}_{\phi_i,v_0,\varepsilon_i}$: For complex sets $\mathcal{V}_{\phi_i,v_0,\varepsilon_i}$ where a convex polygonal approximation is over-restrictive, the optimization can be solved by sampling. This is the case for the sets depicted in Figure 3.

As a naive approach, starting from the velocity $\mathbf{v}_i^{pref}$ and searching the discrete space $ORCA_i^\tau \cap \mathcal{V}_{\phi_i,v_0,\varepsilon_i}$ for the closest velocity $\mathbf{v}_i$ could computationally be expensive. Nevertheless, $\mathbf{v}_i$ can be efficiently computed. First, $\mathbf{v}_i^*$ is obtained solving the optimization with linear constraints given by Equation (14). Then, the procedure in step 4) of the algorithm continues with a constrained wave expansion from $\mathbf{v}_i^*$ as follows: An ordered list is initialized with $\mathbf{v} \in \mathcal{V}$ as the closest velocity to $\mathbf{v}_i^*$ according to a given distance metric, and all its neighbors are added keeping ascending order in distance. While the list is non-empty the first velocity $\mathbf{v}$ of the list (with minimum distance to $\mathbf{v}_i^*$) is checked. If $\mathbf{v}$ verifies a set of linear constraints, i.e. $\mathbf{v} \in ORCA_i^\tau$, the list is expanded with the neighbor velocities of $\mathbf{v}$. If $\mathbf{v}$ further verifies the precomputed $\mathcal{E}_{\phi_i,v_0}(\mathbf{v}) \leq \varepsilon_i$, i.e. $\mathbf{v} \in \mathcal{V}_{\phi_i,v_0,\varepsilon_i}$, then $\mathbf{v}$ is directly returned as the collision-free velocity $\mathbf{v}_i$.

This search method is bounded to the convex polygon given by $ORCA_i^\tau$, and thus the optimal velocity is found in a few steps unless $ORCA_i^\tau \cap \mathcal{V}_{\phi_i,v_0\varepsilon_i} = \emptyset$, where no solution exists.

B. Remarks on the B-ORCA algorithm

***Remark** 1 (Collision-free):* B-ORCA guarantees collision -free trajectories. In each time-step, the planned straight-

line trajectories given by $\mathbf{v}_i$ are collision-free for holonomic robots of radius $r_i + \varepsilon_i$. Further, the trajectory of each car-like robot stays within ε_i of the planned straight line. This guarantees that the distance between two robots is greater than the sum of their radii, thus requiring step 2) of B-ORCA. After each time-step a new collision-free trajectory is computed, leading to more complex global paths.

***Remark** 2 (Kinematic continuity):* B-ORCA guarantees trajectories with continuity in (at least) velocity and steering angle, and fully respects the kinematic constraints and limits in actuators, velocities and accelerations. This properties follow from the controller presented in Section IV.

***Remark** 3 (Convergence):* Convergence to goal destinations is not fully guaranteed in a reasonable time. Deadlock situations may result when the robot's collision-free velocity closest to its preferred velocity tends to zero or $\mathcal{V}_{\phi_i, v_0 \varepsilon_i}$ is over-restricted. This can be resolved by choosing a new preferred velocity given by a global path planner.

VII. Simulation Results

A set of simulated experiments has been conducted to show the performance of the proposed B-ORCA algorithm. The simulated bicycles and car-like vehicles are governed by the kinematics and parameters of Section III and Section IV. Furthermore, the following parameters are chosen for the simulations:

1) Car: $\tau_{max} = 10$s, $\tau_{max}^{min} = 2$s, $d_{max} = 35$m, $dt_c = 0.025$s, $dt_s = 0.2$s and $\hat{\varepsilon}_i = 1$m.

2) MagneBike: $\tau_{max} = 30$s, $\tau_{max}^{min} = 4$s, $d_{max} = 2$m, $dt_c = 0.1$s, $dt_s = 1$s and $\hat{\varepsilon}_i = 0.05$m.

The desired extension $\hat{\epsilon}$ of the robots' radii is selected as a value that presents a good trade-off between radius enlargement and maneuverability for the considered robots. Although the aggressiveness $\lambda_{i,j}$ can be variable, it is chosen as $\lambda_{i,j} = 0.5$ for every pair of robots in the presented simulations, and thus all robots take the same responsibility in avoiding collisions.

Three experiments are presented in this work, all of them performed with ten simulated vehicles of both types (cars and MagneBikes), as follows:

- Experiment 1: Exchange of antipodal positions on a circle.
- Experiment 2: Exchange of antipodal positions on a circle; one robot acts as dynamic obstacle and does not perform any collision avoidance. The remaining nine robots take full responsibility ($\lambda_{i,j} = 1$) in avoiding it.
- Experiment 3: All robots start from random positions, orientations and steering angles and move to random goal positions.

In all of the experiments, uniform noise in position of amplitude 0.1m for the cars and 0.01m for the MagneBikes is added.

In the left of Figure 5 the trajectories of all ten simulated cars, and in the middle of Figure 5 the trajectories of all ten simulated MagneBikes are displayed for the first experiment.

Finally, on the right of Figure 5 the trajectories of the cars are shown for the second experiment, where one of the cars is non-reactive and follows a straight-line trajectory towards its goal.

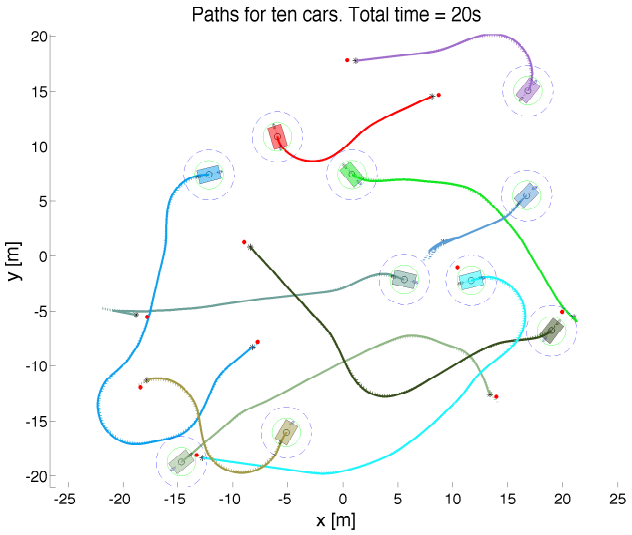

Fig. 6. Trajectories of ten car-like robots starting from a random configuration and moving to random goal positions. The straight line and dashed line trajectories represent the middle and rear-wheel points of the cars, respectively. The robots are displayed in their initial configurations and goal positions are represented by red circles.

These experiments all present extreme symmetry and are thus challenging. B-ORCA performs best in more natural scenarios, where robots are in any position with any orientation and steering angle, and the velocity-based local collision avoidance provides a simple solution. In Figure 6 the trajectories of the third experiment are shown. In this case, the ten cars start from a random configuration and evolve towards a set of random goals. The paths are again smooth. The robots are stopped in the proximity of their goals because the controller of Section IV is designed for trajectory tracking. In order to have perfect convergence, a position controller must be applied when reaching the neighborhood of the goals.

In the accompanying video, all three experiments are presented in full length for both vehicle types, where for each robot three arrows are plotted, representing v_i^{pref} (red), v_i^* (blue) and v_i (black).

We have further implemented the B-ORCA algorithm under ROS[1], and are currently experimenting on collision avoidance with several real MagneBike robots [20], [1].

VIII. Conclusion and Future Work

In this work, a distributed method for reciprocal local collision avoidance among bicycle or car-like robots, so-called B-ORCA, is presented, where each individual robot does not need information about the kinematics of other robots. The method guarantees collision-free motions and achieves smooth trajectories as shown in simulated experiments with ten MagneBike and ten car robots. The method

[1]www.ros.org

relies on the ORCA algorithm that computes a collision-free velocity as if the robots were holonomic. The method further relies on a trajectory tracking controller for car-like vehicles, which could essentially be substituted by any other tracking controller for kinematic constraints different than those presented in this paper.

Furthermore, reciprocal collision-free motions are guaranteed in heterogeneous groups of robots with car-like robots running B-ORCA, navigating in an environment with differentially-driven robots running NH-ORCA [15] and holonomic robots running ORCA [11]. Moreover, collisions with both dynamic and static obstacles are avoided, except in the cases of unfeasibility when due to the kinematic constraints of the robot, no solution exists. Nevertheless, in order to avoid deadlocks in a scenario with static obstacles, a global path planner is required.

Further research is needed in solving deadlock situations in extremely crowded situations. For less controlled environments, or a full integration of sensing and actuation, the method must also be extended to compensate for uncertainties and communication delays.

REFERENCES

[1] A. Breitenmoser, F. Tâche, G. Caprari, R. Siegwart and R. Moser, "MagneBike - Toward Multi Climbing Robots for Power Plant Inspection", in *Proc. of The 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, 2010.

[2] J. Borenstein, Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots", in *IEEE Trans. Robot. Autom.*, **(**7), 278–288, 1991.

[3] P. Fiorini, Z. Shiller, "Motion planning in dynamic environments using velocity obstacles", in *Int. J. Robot. Res.* **(**17)(7), 760–772, 1998.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", in *Int. J. Robot. Res.* **(**5), 90–98, 1986.

[5] T. Siméon, S. Leroy, J.-P. Laumond, "Path coordination for multiple mobile robots: a resolution complete algorithm", in *IEEE Trans. Robot. Autom.* **(**18)(1), 2002.

[6] J. Peng, S. Akella, "Coordinating Multiple Robots with Kinodynamic Constraints Along Specified Paths", in *Int. J. Robot. Res.*, vol. 24 no. 4 295-310, 2005.

[7] D.E. Chang, S. Shadden, J.E. Marsden, R. Olfati Saber, "Collision Avoidance for Multiple Agent Systems", in *Proc. IEEE Conf. Dec. Contr.*, 2003.

[8] D.M. Stipanović, P.F. Hokayem, M.W. Spong, D.D. Šiljak, "Cooperative Avoidance Control for Multiagent Systems", in *ASME J. Dyn. Sys. Meas. Control*, **(**129)(5), 699–707, 2007.

[9] C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. Bessire, C. Laugier, "The CyCab: a car-like robot navigating autonomously and safely among pedestrians", in *Robotics and Autonomous Systems* vol. 50, no. 1, pp. 51-67, 2005.

[10] O. Brock, O. Khatib, "High-speed navigation using the global dynamic window approach", in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999.

[11] J. van den Berg, S. J. Guy, M. Lin and D. Manocha, "Reciprocal n-body Collision Avoidance", in *Int. Symp. on Robotics Research*, 2009.

[12] A. De Luca, G. Oriolo and C. Samson, "Feedback control of a non-holonomic car-like robot", in *"Robot Motion Planning and Control"*, chapter 4, Springer, 1998.

[13] J. van den Berg, M.C. Lin, D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation", in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008.

[14] J. Snape, J. van den Berg, S.J. Guy, D. Manocha, "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles", in *Proc. IEEE Int. Conf. Intell. Rob. Syst.*, 5917–5922, 2009.

[15] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, R. Siegwart, "Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots", in *Proc. Int. Symp. on Distributed Autonomous Robotics Systems*, 2010.

[16] J. Snape, S.J. Guy, D. Manocha, "Navigating Multiple Simple-Airplanes in 3D Workspace", in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010.

[17] J. van den Berg, J. Snape, S. J. Guy, D. Manocha, "Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles," in *Proc. IEEE Int. Conf. on Robots and Automation*, 2011.

[18] B. Kluge and E. Prassler, "Recursive Agent Modeling with Probabilistic Velocity Obstacles for Mobile Robot Navigation Among Humans", in *Springer Tracts in Adv. Robotics*, **(**35), 121-134, 2007.

[19] P. Trautman and A. Krause. Unfreezing the Robot: Navigation in Dense, Interacting Crowds, in *Proc. IEEE Int. Conf. Intell. Rob. Syst.*, 2010.

[20] F. Tâche, W. Fischer, G. Caprari, R. Moser, F. Mondada and R. Siegwart, "Magnebike: A Magnetic Wheeled Robot with High Mobility for Inspecting Complex Shaped Structures", in *Journal of Field Robotics*, **(**26), 453–476, 2009.