2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Avoidance of High-speed Obstacles Based on Velocity Obstacles

Zhongchang Liu, Zeyu Jiang, Tianye Xu, Hui Cheng*, Zhipeng Xie, Liang Lin

Abstract— **For obstacles moving with high speeds, existing motion planning methods can rarely guarantee collision avoidance. This paper proposes a viable two-period velocity obstacle algorithm where one period predicts potential collisions within a limited time horizon, and the second period foresees collisions beyond that horizon. The second period is activated only when the obstacle's moving speed is larger than the maximum speed of the robot. The applicability of the new algorithm and the related computation issues are discussed. Both computer simulations and laboratory experiments illustrated the effectiveness of the proposed obstacle avoidance algorithm.**

I. Introduction

As the development of robotics, flexible mobile robots are more and more frequently engaged in tasks in complex environments, such as searching in a hazardous and hostile environments, filming movies, broadcasting sport games, and transportation in a plant accompanying labours and other vehicles. Motion planning techniques ensure a robot safe navigation from a position to its goal without colliding with any obstacles or any other robots in a workspace. To avoid static obstacles in a known environment, off-line path planning algorithms could be useful. However, for dynamic obstacles, real-time collision avoidance algorithms are desired.

This paper studies the problem of avoiding obstacles whose speeds are higher than the maximum speed of the robot. For examples, a ground robot searching in a post-disaster area should avoid lethal attacks from uncontrollable moving objects such as falling rocks; an intelligent football game shooting robot needs to avoid a ball flying towards it so as to avoid potential damages; a transportation robot should avoid reckless human beings who are unaware of the robot, and so on. This paper accomplishes this task by setting the robot's velocity out of the set where potential collisions with the obstacle may occur in a limited time horizon and where collisions can occur beyond that time horizon.

A. Related work

There are two major classes of algorithm for real-time avoidance of moving obstacles. The first class of approaches are based on the relative distances between the robot and the obstacles. The representative one is the artificial potential field method [1], [2] which can generate increasing repulsive forces that push the robot moving away from the obstacle as the obstacle approaches. However, the repulsive forces would usually render the robot being "chased" after by the obstacle until collision occurs if the obstacle's speed is higher than the robot's. The model predictive control (MPC) also has a long history of being used for collision avoidance [3]. However, it suffers the same problem as the artificial potential field method since it usually uses the relative distance based potential field functions as penalty functions.

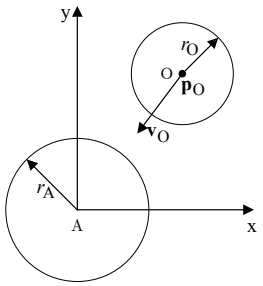
The second class of approaches aim to plan the robot's moving velocities in each step. The well-known algorithm is the velocity obstacle (VO) algorithm [4] which first finds the set of relative velocities leading to collision, and then guides the robot's velocity leaving that set. There are subsequent modifications for the basic VO method, resulting in new algorithms such as reciprocal velocity obstacle (RVO) [5], optimal reciprocal collision avoidance (ORCA) [6], extended velocity obstacle (EVO) [7], etc. The RVO and ORCA algorithms narrowed down the set of infeasible relative velocities in VO by introducing a fixed time horizon when predicting potential collisions, and thus they can provide more space for a robot to choose collision avoiding velocities. In addition, these two algorithms let each robot take half the responsibility of avoiding collisions with each other, and thus are suitable for cooperative robots. The ORCA algorithm [6] also solves the problem of motion oscillations in RVO as the number of robots grows. The EVO algorithm [7] is built on the ORCA algorithm, and it lets the robots communicate with each other their preferred velocities so that potential collisions along their desired paths can be avoided. Some other variants of the VO based algorithms also take the robots' physical models into computation, for example, the second-order motion equations [8], general linear system models [9], [10], and non-holonomic models [11], [12]. These extensions released the requirement of instantaneous velocity changes for the robots in the early VO algorithm. Due to the distributed nature and the efficiency for real-time computation, these velocity based collision avoidance algorithms have been used in physical systems such as quadrotors via on-board relative velocity sensing [13], [14].

B. Contribution and paper organization

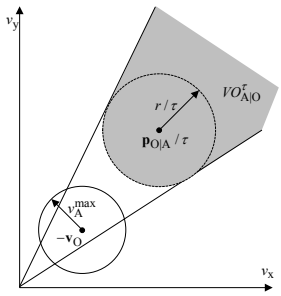
Previous velocity obstacle based methods focused on avoiding collisions among multiple cooperative robots, while few has studied how to avoid noncooperative obstacles which are moving with relatively high speeds. The variants of VO based method, such as RVO, ORCA and EVO, could fail the task because they are shortsighted by predicting potential collisions only within a limited time horizon. On the other

This work is supported by National Natural Science Foundation of China (61703445), NSFC-Shenzhen Robotics Projects (U1613211), Guangdong Natural Science Foundation (1614050001452, 2017A030310050), Major Program of Science and Technology Planning Project of Guangdong Province (2017B010116003), and the Fundamental Research Funds for the Central Universities.

The authors are with the Sun Yat-sen University, Guangzhou, P. R. China.
*Corresponding author: chengh9@mail.sysu.edu.cn



(a) Motion configurations of robot A with radius r_A and obstacle O with radius r_O.



(b) Velocity obstacle $VO^{\tau}_{A|O}$ (the grey region) for robot A with respect to obstacle O within time τ.

Fig. 1. Configurations of a robot A and a dynamic obstacle O.

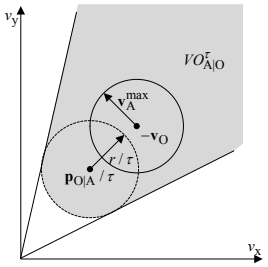


Fig. 2. There is no obstacle avoiding velocity for robot A if $D(-\mathbf{v}_O, v_A^{max}) \subset VO^{\tau}_{A|O}$.

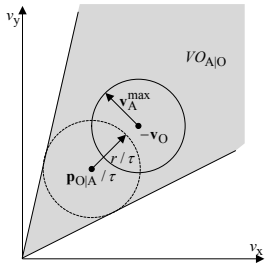


Fig. 3. Velocity obstacle $VO_{A|O}$ (the grey region) for robot A with respect to obstacle O.

hand, prolonging the prediction time horizon would reduce the set of feasible solutions which makes the algorithm too conservative. The extreme case is the original VO algorithm [4] where an infinite prediction time horizon is presumed for the robots so that all future paths of an obstacle are treated as infeasible paths for the robot. Apparently, the original VO algorithm is insensitive to the obstacles' velocities.

In this paper, the proposed new algorithm has two new features:

1) The robot predicts potential collisions not only within a fixed time horizon, but also collisions beyond that time horizon. So, the robot can react to the obstacle much earlier.
2) The set of infeasible relative velocities between the robot and the obstacle is extended from that in the RVO or ORCA algorithm when the obstacle's velocity is large enough, and will reduce to that in RVO or ORCA otherwise. Hence, the new algorithm is activated only when encountering obstacles moving with a relatively high speed.

The remainder of this paper is organized as follows. Section II states the problem concerned in this paper. The new algorithm is presented in section III. In section IV, both numerical simulations and experiments are carried out. Section V concludes the work of this paper and provides future investigations.

II. Problem Statement

Consider a scenario where a robot A encounters an obstacle O moving with velocity $\mathbf{v}_O$ as shown in Fig.1(a) where the origin of the coordinate is the center of the robot. The robot and the obstacle are represented as discs centered at $\mathbf{p}_A$ and $\mathbf{p}_O$ with radii r_A and r_O, respectively. Assuming the obstacle's velocity keeps constant in a future time horizon $\tau > 0$, then the set of relative velocities between the obstacle O and robot A which will lead to collision within τ is named the velocity obstacle [4], [6]. The mathematical representation is

$$VO^{\tau}_{A|O}(\mathbf{p}_{O|A}) := \{\mathbf{v} : \mathbf{v} \in D(\mathbf{p}_{O|A}/t, r/t), \forall t \in (0, \tau]\} \quad (1)$$

where $\mathbf{p}_{O|A} = \mathbf{p}_O - \mathbf{p}_A$, $r = r_A + r_O$, and $D(\mathbf{p}_{O|A}/t, r/t) := \{\mathbf{v} : \|\mathbf{v} - \mathbf{p}_{O|A}/t\| \leq r/t\}$ is a disc centered at $\mathbf{p}_{O|A}/t$ with radius r/t. Fig.1(b) shows the velocity obstacle of the configuration in Fig.1(a) by the grey region.

Robot A would control its velocity to leave the velocity obstacle if its current velocity with respect to the obstacle's is already in it (i.e., if $\mathbf{v}_A - \mathbf{v}_O \in VO^{\tau}_{A|O}$). However, if the obstacle's velocity $\mathbf{v}_O$ is such that the set $D(-\mathbf{v}_O, v_A^{max})$ is entirely contained in the velocity obstacle $VO^{\tau}_{A|O}$, it will be impossible for the robot to avoid the moving obstacle within time τ. Note that this situation can happen only if $\|\mathbf{v}_O\| > v_A^{max}$ as illustrated in Fig.2. This implies that, for a dynamic obstacle moving with speed $\|\mathbf{v}_O\| > v_A^{max}$, there exist motion configurations where the robot A cannot avoid collision with it within any time horizon $\tau > 0$. Fig.3 illustrates such a configuration where the set

$$VO_{A|O}(\mathbf{p}_{O|A}) := \{\mathbf{v} : \mathbf{v} \in D(\mathbf{p}_{O|A}/t, r/t), \forall t > 0\} \quad (2)$$

is the velocity obstacle in which the robot will collide with the obstacle in some time beyond the horizon $\tau > 0$.

This infeasible configuration didn't rise in reciprocal collision avoidance scenarios among multiple cooperative robots since all robots were assumed to have the same maximum velocity [5], [6]. This paper is dedicated to fixing this problem by proposing an obstacle avoidance algorithm that can handle high-speed obstacles (in the sense that $\|\mathbf{v}_O\| > v_A^{max}$). In the sequel section, we will first analyse how the infeasible configurations was caused, and then we present the algorithm based on the idea of building velocity obstacles.

III. Avoiding High-Speed Obstacles

For simplicity of analysis, the obstacle's velocity is assumed to be constant, and can be measured precisely by the robot. The method proposed in this paper can be adapted so as to avoid dynamic obstacles with predictable velocities.

A. Methodology

Suppose the robot and the obstacle are in a motion configuration such that $D(-\mathbf{v}_O, v_A^{max}) \nsubseteq VO^{\tau}_{A|O}(\mathbf{p}_{O|A})$. Denote by $\mathbf{v}_{A|O} := \mathbf{v}_A - \mathbf{v}_O$ their current relative velocity. Let $\mathbf{v}_A^{\text{new}}$ be the robot's new velocity which can ensure the robot be collision free with the obstacle O within a

future time horizon τ. Denote $\mathbf{v}_{A|O}^{\text{new}} = \mathbf{v}_A^{\text{new}} - \mathbf{v}_O$, and let $\mathbf{p}_{O|A}^{\text{new}} = \mathbf{p}_O + \mathbf{v}_O \cdot \tau - \mathbf{p}_A^{\text{new}}$ be the new relative position after time τ. It is clear that

$$\mathbf{p}_{O|A}^{\text{new}} = \mathbf{p}_{O|A} - \mathbf{v}_{A|O}^{\text{new}} \cdot \tau. \tag{3}$$

If the new relative velocity $\mathbf{v}_{A|O}^{\text{new}}$ and the resulted new relative position $\mathbf{p}_{O|A}^{\text{new}}$ satisfy $\mathbf{v}_{A|O}^{\text{new}} \in VO_{A|O}(\mathbf{p}_{O|A}^{\text{new}})$ and $D(-\mathbf{v}_O, v_A^{max}) \subset VO_{A|O}(\mathbf{p}_{O|A}^{\text{new}})$, then the robot cannot avoid the obstacle any longer no matter how the robot changes its velocity. All such new relative velocities are captured by the following set

$$\begin{aligned} &MVO_{A|O}^{\tau}(\mathbf{p}_{O|A}) \\ &= \{\mathbf{v} : D(-\mathbf{v}_O, v_A^{max}) \subset VO_{A|O}(\mathbf{p}_{O|A} - \mathbf{v} \cdot \tau)\}. \end{aligned} \tag{4}$$

This is the set of relative velocities which can definitely lead to collision beyond time horizon τ. The letter M here represents "maximum" since the set exists only when $v_A^{max} < \|\mathbf{v}_O\|$ as discussed in the previous section. Recall that $VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ contains the set of relative velocities leading to collision within time τ. So, the robot can avoid collision with a high-speed obstacle if at each execution time, the robot computes $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ and $VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ and then selects a new velocity such that $\mathbf{v}_{A|O}^{\text{new}}$ does not lie in either of the two sets.

Generally, the robot would have a predefined desired velocity $\mathbf{v}_A^{\text{pref}}$. So, $\mathbf{v}_{A|O}^{\text{new}}$ is chosen as close to the preferred relative velocity $\mathbf{v}_{A|O}^{\text{pref}} := \mathbf{v}_A^{\text{pref}} - \mathbf{v}_O$ as possible, i.e.

$$\mathbf{v}_{A|O}^{\text{new}} = \underset{\mathbf{v} \in D(-\mathbf{v}_O, v_A^{max}), \mathbf{v} \notin VO_{A|O}^{\tau} \cup MVO_{A|O}^{\tau}}{\arg\min} \left\| \mathbf{v} - \mathbf{v}_{A|O}^{\text{pref}} \right\|. \tag{5}$$

As a last step, the new velocity of robot A is set as

$$\mathbf{v}_A^{\text{new}} = \mathbf{v}_{A|O}^{\text{new}} + \mathbf{v}_O. \tag{6}$$

Since this method builds velocity obstacles both in time horizon τ and foresees velocity obstacles at any time beyond τ, we call it a two-period velocity obstacle algorithm.

This new algorithm can be easily extended to scenarios of avoiding multiple high-speed obstacles by constructing the infeasible set $VO_{A|O_i}^{\tau}(\mathbf{p}_{O_i|A}) \cup MVO_{A|O_i}^{\tau}(\mathbf{p}_{O_i|A})$ for each obstacle O_i, $i = 1, 2, \ldots$, and then setting the robot's new velocity outside their union. In the following, we will discuss the feasibility issues of the new algorithm and the computation of these infeasible sets. Discussions will be made for avoiding one obstacle only for clarity of presentation.

B. Feasibility of the algorithm

It is clear that the new approach is applicable only when $D(-\mathbf{v}_O, v_A^{max}) \nsubseteq VO_{A|O}^{\tau}(\mathbf{p}_{O|A}) \cup MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ at the initial time.

- The part $D(-\mathbf{v}_O, v_A^{max}) \nsubseteq VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ is a prerequisite for obstacle avoidance within time horizon τ. If this part fails, the robot will collide with the obstacle within τ and so there is no need to compute the set $MVO_{A|O}^{\tau}$ any more.
- If the above is satisfied, the feasibility condition $D(-\mathbf{v}_O, v_A^{max}) \nsubseteq VO_{A|O}^{\tau}(\mathbf{p}_{O|A}) \cup MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ is guaranteed when the radius of the disc $D(-\mathbf{v}_O, v_A^{max})$ is larger than that of the disc $D(\mathbf{p}_{O|A}/\tau, r/\tau)$, i.e., $v_A^{max} > r/\tau$. The implication is that the robot can at least brush pass the obstacle within time τ by moving with speed v_A^{max} in the direction vertical to $\mathbf{v}_O$.

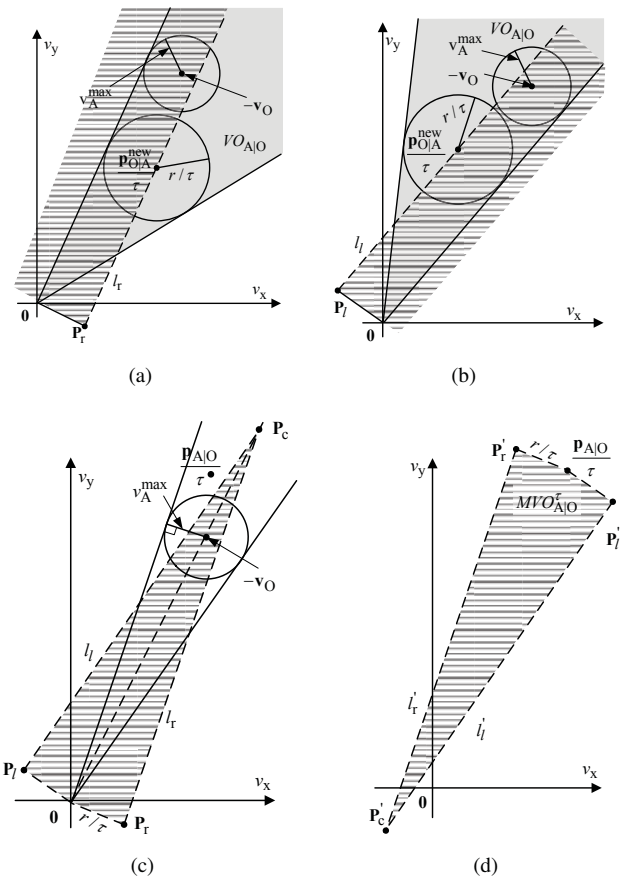


Fig. 4. The derivation of the set $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$.

C. Computation issues

In order to use the above method, one needs to get the set $VO_{A|O}^{\tau}$ defined in (1), $MVO_{A|O}^{\tau}$ defined in (4), and their union $VO_{A|O}^{\tau} \cup MVO_{A|O}^{\tau}$. While the computation of $VO_{A|O}^{\tau}$ is well studied in existing literature [5], [6], we only present the computation issues of the latter two sets.

1) Computation of $MVO_{A|O}^{\tau}$: Based on the definition in (4), the set $MVO_{A|O}^{\tau}$ can be constructed as follows.

a) Find the set of points, $\mathbf{p}_{O|A}^{\text{new}}/\tau$, each of whose associated velocity obstacle $VO_{A|O}(\mathbf{p}_{O|A}^{\text{new}})$ contains the disc $D(-\mathbf{v}_O, v_A^{max})$ and has its left boundary tangent to the disc. This set is the dotted half line l_r in Fig.4(a) where the segment $\mathbf{0}-\mathbf{P}_r$ has length r/τ and is perpendicular to l_r.

b) Similarly, get another set of points, $\mathbf{p}_{O|A}^{\text{new}}/\tau$ (represented by the dotted half line l_l in Fig.4(b)), each of whose associated velocity obstacle contains the same disc $D(-\mathbf{v}_O, v_A^{max})$ and has its right boundary tangent to the disc.

c) The two half lines l_r and l_l together with the two segments $\mathbf{0}-\mathbf{P}_r$ and $\mathbf{0}-\mathbf{P}_l$ encompass the largest set of points, $\mathbf{p}_{O|A}^{\text{new}}/\tau$, which can generate a velocity obstacle $VO_{A|O}(\mathbf{p}_{O|A}^{\text{new}})$ that contains $D(-\mathbf{v}_O, v_A^{max})$. This region is denoted by $MPO_{A|O}^{\tau}(\mathbf{v}_O)$ and is represented by the shadowed area in Fig.4(c).

d) Get $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ using the following expression

$$\mathbf{v}_{A|O}^{\text{new}} = \frac{1}{\tau}(\mathbf{p}_{O|A} - \mathbf{p}_{O|A}^{\text{new}}), \forall \mathbf{p}_{O|A}^{\text{new}}/\tau \in MPO_{A|O}^{\tau}(\mathbf{v}_O). \tag{7}$$

The set $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ is shown in Fig. 4(d).

As seen from Fig. 4(d), the set $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ in a two dimensional Euclidean space is a quadrilateral with the line of symmetry passing $\mathbf{P}_c'$ and $\mathbf{p}_{O|A}/\tau$. Its vertices $\mathbf{P}_c'$, $\mathbf{P}_r'$, and $\mathbf{P}_l'$ can be derived by first finding the coordinates of the vertices $\mathbf{P}_c$, $\mathbf{P}_r$, and $\mathbf{P}_l$ in Fig. 4(c) and then using the transformation in (7). $\mathbf{P}_c$ is proportional to the vector $-\mathbf{v}_O$, while $\mathbf{P}_r$, and $\mathbf{P}_l$ can be derived by rotating and scaling $-\mathbf{v}_O$. They are given by

$$\mathbf{P}_c = -\frac{r}{v_A^{max}\tau}\mathbf{v}_O, \tag{8}$$

$$\mathbf{P}_r = \begin{bmatrix} v_A^{max} & m \\ -m & v_A^{max} \end{bmatrix} \frac{-\mathbf{v}_O}{\|\mathbf{v}_O\|^2} \cdot \frac{r}{\tau}, \tag{9}$$

$$\mathbf{P}_l = \begin{bmatrix} v_A^{max} & -m \\ m & v_A^{max} \end{bmatrix} \frac{-\mathbf{v}_O}{\|\mathbf{v}_O\|^2} \cdot \frac{r}{\tau}, \tag{10}$$

where $m = \sqrt{\|\mathbf{v}_O\|^2 - (v_A^{max})^2}$.

2) Computation of $VO_{A|O}^{\tau} \cup MVO_{A|O}^{\tau}$: As seen from Fig. 4(c) - 4(d), the vertex $\mathbf{0}$ of $MPO_{A|O}^{\tau}$ is transformed to the point $\mathbf{p}_{O|A}/\tau$ in $MVO_{A|O}^{\tau}$ according to (7). Hence, the two sets, $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ and $VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$, always have a nontrivial intersection as shown in Fig. 5. In fact, each set has both sides tangent to the boundary of $D(\mathbf{p}_{O|A}/\tau, r/\tau)$. It follows that the union $VO_{A|O}^{\tau} \cup MVO_{A|O}^{\tau}$

i) is equal to $VO_{A|O}^{\tau}$ if the point $\mathbf{P}_c'$ locates in $VO_{A|O}^{\tau}$. (Rigorously speaking, $\mathbf{P}_c'$ will never locate in the disc $D(\mathbf{p}_{O|A}/\tau, r/\tau)$ which is a subset of $VO_{A|O}^{\tau}$ if $\|\mathbf{v}_O\| > v_A^{max}$ since $\|\mathbf{P}_c' - \mathbf{p}_{O|A}/\tau\| = \|\frac{r}{v_A^{max}\tau}\mathbf{v}_O\| > r/\tau$.)

ii) is surrounded by the linear boundaries of $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ and $VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ if $\mathbf{P}_c'$ locates in $VO_{A|O}(-\mathbf{p}_{O|A})$.

iii) is surrounded by the linear boundaries of $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ and $VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ as well as arcs from the disc $D(\mathbf{p}_{O|A}/\tau, r/\tau)$, otherwise.

The first two cases are easy to handle when solving the optimization problem in (5). However, the infeasible set in case iii) are complicated due to the existence of the nonlinear arcs. For ease and efficiency of optimization, one can replace these arcs with the extensions of the linear boundaries of $MVO_{A|O}^{\tau}(\mathbf{p}_{O|A})$ and $VO_{A|O}^{\tau}(\mathbf{p}_{O|A})$. Details are omitted in this paper.

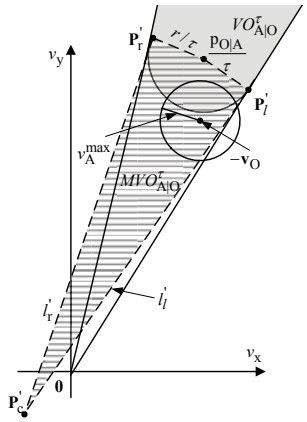


Fig. 5. Infeasible set $VO_{A|O}^{\tau} \cup MVO_{A|O}^{\tau}$ for choosing $\mathbf{v}_{A|O}^{\text{new}}$.

IV. Simulations and Experiments

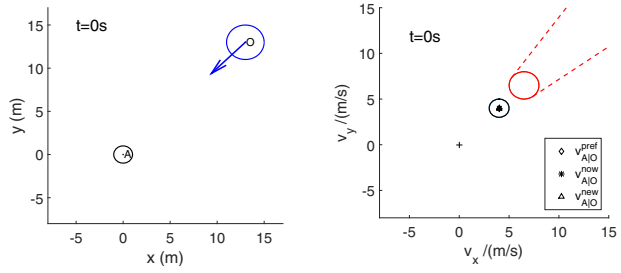
In this section, we conduct both computer simulations and laboratory experiments to show the effectiveness of the proposed algorithm.

A. Simulation results

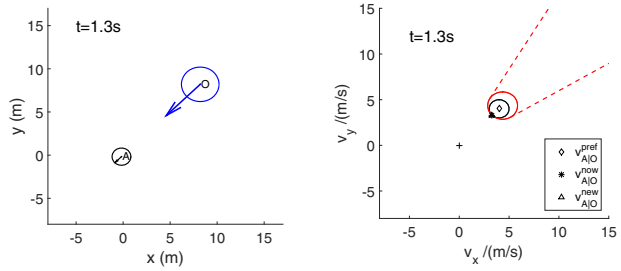
1) Avoiding one fast obstacle: In this subsection, simulations are conducted for a scenario where the robot A stays statically at the origin and tries to avoid an obstacle O which moves towards the robot with a constant direction and speed ($\mathbf{v}_O = [-4; -4]$ m/s) larger than the maximum speed of the robot ($v_A^{max} = 1$m/s). The initial position of the obstacle is $(13, 13)$. The radii of the robot and the obstacle are $r_A = 1$m and $r_O = 2$m, respectively. The preferred velocity of the robot is given by $\mathbf{v}_A^{pref} = (\mathbf{p}_A^d - \mathbf{p}_A^{now})/\tau$, where $\mathbf{p}_A^d$ is the target position and $\mathbf{p}_A^{now}$ is the real-time position of the robot. The prediction time horizon $\tau = 2s$ and the execution time interval is $0.1s$.

For the purpose of comparison, the ORCA algorithm in [6] is applied first. Then the new two-period VO algorithm is applied. Simulation results applying these two algorithms are shown in Fig. 6 and Fig. 7, respectively where snapshots for the motion configurations in both the position space and the velocity space are presented. Using the ORCA algorithm, the robot tried to avoid the obstacle by moving in the same direction with the obstacle as shown in Fig. 6. In Fig. 6(d), we see that the robot's velocity reached its maximum by time $1.3s$, and all of its possible velocities with respect to the obstacle's were contained in the velocity obstacle, i.e., $D(-\mathbf{v}_O, v_A^{max}) \subset VO_{A|O}^{\tau}$. Afterward, the robot kept moving with the maximum speed in the same direction of the obstacle, and it eventually failed avoiding the obstacle as shown in Fig. 6(e).

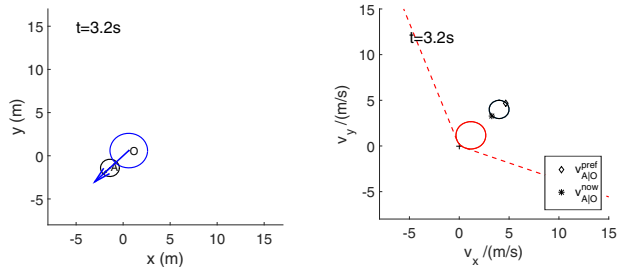
Using the two-period VO algorithm instead, the robot succeeded in avoiding the obstacle as shown in snapshots in Fig. 7(a), 7(c) & 7(e). In Fig. 7(c), we see that the robot tried to avoid collision by giving way to the obstacle rather than escaping in the obstacle's direction. This was achieved due to the introduction of the set $MVO_{A|O}^{\tau}$ which made moving in the same direction of the obstacle infeasible so that the robot



(a) Position space (b) Velocity space

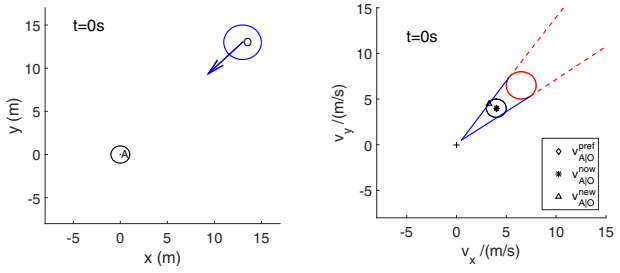


(c) Position space (d) Velocity space

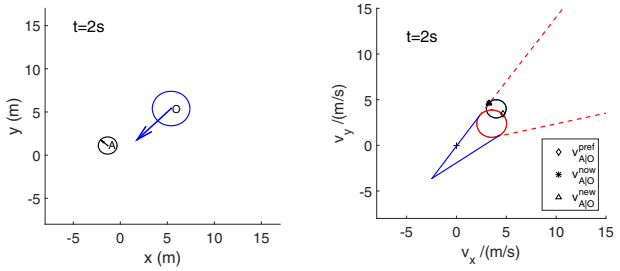


(e) Position space (f) Velocity space

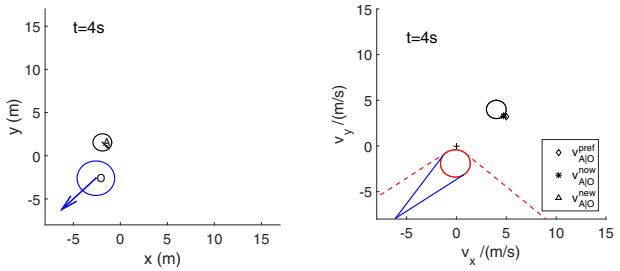
Fig. 6. Snapshots of robot A trying to avoid the obstacle O by implementing the ORCA algorithm. In each position space, the large circle represents the obstacle, the small circle is the robot, and the arrows indicate the directions and sizes of the obstacles' velocities. In each velocity space, the black circle encloses $D(-\mathbf{v}_O, v_A^{max})$, the red circle encloses $D(\mathbf{p}_{O|A}/\tau, r/\tau)$, and the dotted lines are the linear boundaries of $VO^{\tau}_{A|O}$.



(a) Position space (b) Velocity space



(c) Position space (d) Velocity space



(e) Position space (f) Velocity space

Fig. 7. Snapshots of robot A avoiding the obstacle O by implementing the two-period VO algorithm. The solid straight lines in each velocity space are the linear boundaries of $MVO^{\tau}_{A|O}$, and the other graphics represent the same items with those in Fig. 6.

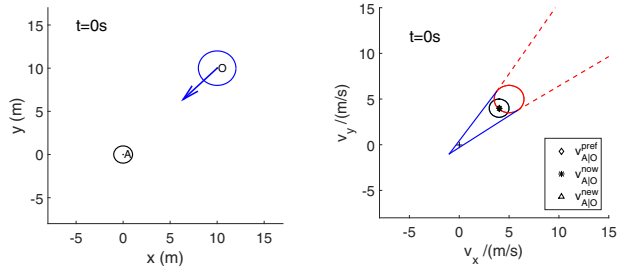
had to move sideward. This simulation example illustrates that the new algorithm can avoid a high-speed obstacle that cannot be avoided by the existing VO method.

2) Feasibility of the two-period VO algorithm: In the above example, one sees that $v_A^{max} = 1$ m/s $< r/\tau = (1 + 2)/2$ m/s. According to the feasibility discussions in Section III-B, there are initial position configurations for the robot and the obstacle such that the new algorithm is incapable of avoiding the obstacle. The next simulation example shows such a scenario where the values of the parameters are the same with those in the first example except that the initial position of the obstacle is changed to $[10; 10]$ as shown in Fig. 8(a). This configuration makes the set $D(-\mathbf{v}_O, v_A^{max})$ contained in $VO^{\tau}_{A|O}(\mathbf{p}_{O|A}) \cup MVO^{\tau}_{A|O}(\mathbf{p}_{O|A})$ at the initial time as shown in Fig. 8(b). As expected, the robot cannot avoid the obstacle as presented in Fig. 8(c).

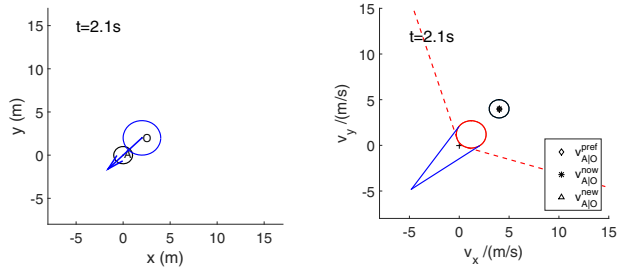
3) Avoiding multiple fast obstacles: This subsection presents a scenario where the robot A tried to transfer from the initial position $(0, 0)$ to the goal position $(0, 7)$ while avoiding three dynamic obstacles, all of which were moving with speeds larger than the maximum speed of the robot ($v_A^{max} = 2$ m/s). The initial configurations are drawn in Fig. 9(a), in which the velocity vectors of the three obstacles are $[0; -8]$, $[2.6; -4]$, and $[-4; -4]$, respectively. Fig. 9(b) shows that the ORCA algorithm cannot generate collision-free guidance for the robot. On the other hand, using the proposed two-period algorithm, the robot succeeded in avoiding the three obstacles and arrived at its goal position as presented in Fig. 10.

B. Experiment

The experiment was accomplished in the laboratory for a robot to avoid two moving obstacles by using the two-period VO algorithm. The obstacles are dummy toy cars manipulated remotely by lab members. The robot is an omni-directional four-wheeled vehicle with maximum speed 0.7 m/s. The robot's velocity is estimated by measurements from the encoders connected to the wheels. A 2-dimentional

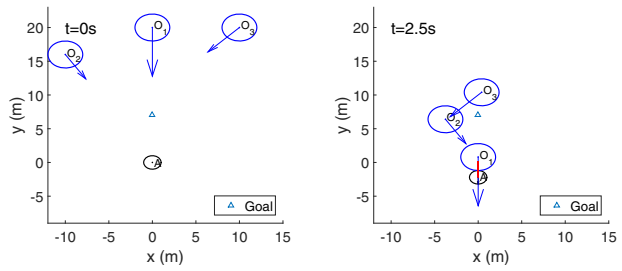


(a) Position space (b) Velocity space



(c) Position space (d) Velocity space

Fig. 8. Robot A failed avoiding the obstacle O when applying the two-period VO algorithm due to the violation of the feasibility condition at the initial time.

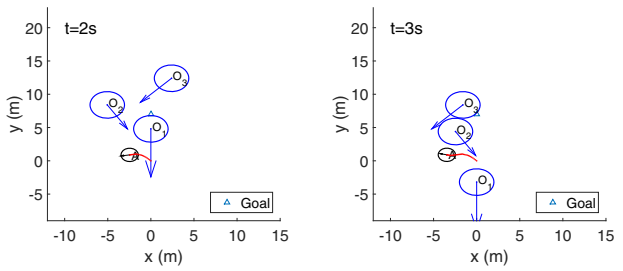


(a) Initial positions of three obstacles and one robot. (b) Robot A collided with obstacle O_1 after 2.5s.

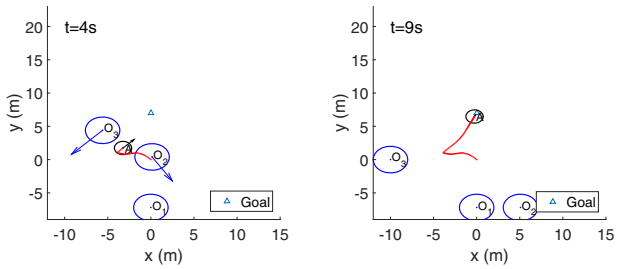
Fig. 9. A scenario where the robot failed avoiding the obstacles by using the ORCA algorithm. (The red line in (b) is the trajectory of robot A.)

Hokuyo's UTM-30LX scanning laser rangefinder is mounted on the top front of the vehicle to detect the dynamic obstacles at a rate of 20Hz. The computation unit of the robot is a laptop mounted on the vehicle with Ubuntu operation system.

The results of the experiment are shown in Fig. 11 where the process of successful avoidance of the two obstacles by the robot were demonstrated in the four snapshots. The velocities of the robot and the obstacles estimated from measurements of on-board sensors are drawn in Fig. 12 which shows that the obstacles' moving speeds are much larger than the robot's maximum speed. Note that the clock time is shifted so as to start from zero. Obstacle O_1 was detected first, and O_2 appeared in the view range of the radar after time 1.5s. The robot's speed fluctuations between time 1.0s and 4.0s indicate the action of obstacle avoidance, while the speed changes between 6.0s to 7.0s indicate that the robot was moving back to its original position. The abrupt of



(a) Robot A avoided O_1 after 2s. (b) Robot A avoided O_2 after 3s.



(c) Robot A avoided O_3 after 4s. (d) Robot A arrived at its goal position after 9s.

Fig. 10. Starting from the initial configuration in Fig. 9(a), robot A avoided three obstacles and reached its goal by using the two-period VO algorithm.

obstacle O_2's velocity at time interval 4.5s to 5.3s occurred due to view interruption of the radar by obstacle O_1 since O_1 was staying in front of O_2 (see Fig. 11(c)).

V. Conclusions

This paper presents a motion planning algorithm for a robot to avoid high-speed obstacles. A new velocity obstacle is constructed in each step, which can predict collisions beyond the time horizon of the existing VO based method. Since this new velocity obstacle is activated only when the obstacle's speed is larger than the maximum speed of the robot, the new algorithm is computationally acceptable, more sensitive to the obstacles' velocities compared to the original version in [4], and more powerful in handling dynamic obstacles than the refined versions of the VO methods such as ORCA in [6]. In the future, one may extend this method to avoid obstacles whose velocities are dynamically varying but predictable using estimators. One another future work is to take the dynamical model of the robot and those of the obstacles into account so as to provide more accurate motion planning and control for the robot. The idea of combining the two-period VO method with model predictive control as in [15] may provide a promising solution.

Acknowledgment

The authors would like to thank Youlian Long and Lixiang Huang for their assistance with the experiments.

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

(a) Initial positions of the robot and two obstacles.

(b) The robot started to avoid the obstacles.

(c) The robot avoided one obstacle.

(d) The robot avoided the two obstacles and started to move back to its initial position.

Fig. 11. Snapshots of a robot avoiding two obstacles by implementing the two-period VO algorithm.

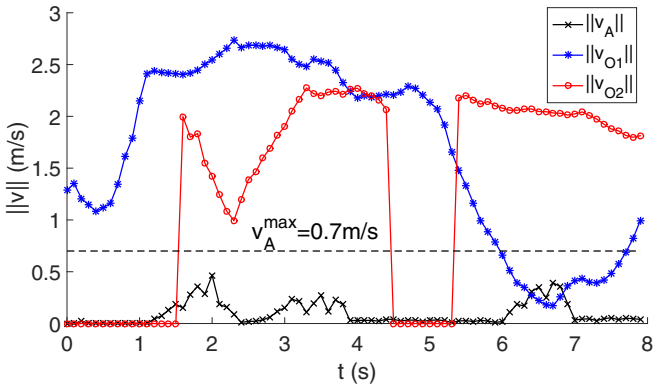


Fig. 12. Speeds of the robot and the obstacles.

[2] S. S. Ge, X. Liu,C. H. Goh, and L. Xu, "Formation tracking control of multiagents in constrained space," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 992–1003, 2016.

[3] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proceedings of IEEE Conference on Decision and Control*, 2003, pp. 3621–3626.

[4] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[5] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 1928–1935.

[6] J. Van den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," *Robotics research*, pp. 3–19, 2011.

[7] A. Levy, C. Keitel, S. Engel, and J. McLurkin, "The extended velocity obstacle and applying ORCA in the real world," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 16–22.

[8] J. Van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *IEEE International Conf. Robotics and Automation*, 2011, pp. 3475–3482.

[9] M. Rufli, J. Alonso-Mora, and R. Siegwart, "Reciprocal collision avoidance with motion continuity constraints," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 899–912, 2013.

[10] D. Bareiss and J. Van den Berg, "Reciprocal collision avoidance for robots with linear dynamics using LQR-obstacles," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 3847–3853.

[11] J. Alonso-Mora, A. Breitenmoser, M. Rufli, et al, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 203–216.

[12] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *IEEE International Conf. Robotics and Automation*, 2012, pp. 360–366.

[13] P. Conroy, D. Bareiss, M. Beall, et al, "3-D reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," *arXiv:1411.3794*, 2014.

[14] S. Roelofsen,D. Gillet, A. Martinoli, "Reciprocal collision avoidance for quadrotors using on-board visual detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4810–4817.

[15] H. Cheng, Q. Zhu, Z. Liu, T. Xu, and L. Lin, "Decentralized navigation of multiple agents based on ORCA and model predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, September, 2017, pp. 3446-3451.