IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 21, 2020, accepted July 13, 2020, date of publication July 28, 2020, date of current version August 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012513

Collision Avoidance of High-Speed Obstacles for Mobile Robots via Maximum-Speed Aware Velocity Obstacle Method

TIANYE XU[1], SHUIQING ZHANG[1], ZEYU JIANG[1], ZHONGCHANG LIU[2], (Member, IEEE), AND HUI CHENG[1], (Member, IEEE)

[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China
[2]College of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China

Corresponding author: Hui Cheng (chengh9@mail.sysu.edu.cn)

This work was supported by the Major Program of Science and Technology Planning Project of Guangdong Province under Grant 2017B010116003, the National Natural Science Foundation of China under Grant 61703445, the NSFC-Shenzhen Robotics Projects under Grant U1613211, the Natural Science Foundation of Liaoning Province under Grant 20180540064, and the Natural Science Foundation of Guangdong Province under Grant 2017A030310050.

ABSTRACT It is challenging for a mobile robot to avoid moving obstacles in dynamic environments. Traditional velocity obstacle methods do not fully consider the obstacles moving with the speeds larger than the maximum speed of the robot. In this article, a new obstacle avoidance method, named the maximum-speed aware velocity obstacle (MVO) algorithm, is proposed for a mobile robot to avoid one or multiple high-speed obstacles. The proposed algorithm expands the velocity obstacle region into two parts, where one of the parts foresees collisions beyond the time horizon to ensure the feasible solutions of the current and the next control step. In practical applications, the perception capability of the robot is generally limited, and a non-holonomic robot can't move into any direction due to its kinematic constraints. In this article, the limited sensing field of view and non-holonomic kinematic constraints of the mobile robot are incorporated into the proposed MVO method. Moreover, continuity, safety, and computational complexity of the MVO approach are analyzed and presented. Extensive simulations and physical experiments are conducted to verify the efficacy of the MVO method, where a quadrotor and a differential-drive robot are used to perform dynamic obstacle avoidance.

INDEX TERMS Motion planning, collision avoidance, high-speed obstacles, limited field of view, kinematic constraints.

I. INTRODUCTION

Mobile robots are increasingly popular in tasks like transportation, inspection, and surveillance. In these applications, the robot should autonomously avoid obstacles when navigating to the destination, and path planning algorithms have been developed to perform obstacle avoidance.

Generally, path planning algorithms are divided into global path planning and local path planning according to environmental perception information. Global path planning requires comprehensive perception and high computational capacities. In the presence of dynamic obstacles, replanning may cause stops and reduce the maneuverability of the robot. Moreover, as the field of view of the robot is limited, local planning algorithms will observe and avoid obstacles only when the obstacles are within the perception range of the robot. Therefore, for mobile robots with limited perception, it is necessary to take account of high-speed obstacles avoidance into the path planning.

Considering the speed of the robot to be the perceived range as well as its limited field of view, [1] proposes a velocity obstacle (VO) based method to guarantee a smooth and collision-free action. On the contrary, a relaxed constraint Model Predictive Control (MPC) framework is constructed in [2] that guarantees path safety by allowing motions outside the limited field of view. The core of the method is the ability to choose motion primitive at the past time step. In [3], a deep neural network is trained to map uncertain sensor

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

measurements to the velocity increment of the robot. A probabilistic variant of the Reciprocal Velocity Obstacle (RVO) method [4] is proposed, which takes account of uncertain state estimation and actuation of the robots.

In realistic scenarios, there are many applications where the speed of the robot is much lower than the speed of the obstacles, for example mobile service robots and sweeping robots usually move much slower than human beings or some pets (e.g. dogs, cats), while these robots are expected to avoid collision with a careless human.

When there exists high-speed moving obstacle, the velocity feasible region may be almost in the velocity obstacle region. Since the maximum speed of a robot is generally limited, it may have no ability to flee away from the velocity obstacle region. In this article, considering the limited field of view of the robot, we aim to study the robotic navigation problem in the presence of multiple dynamic obstacles including high-speed ones.

The mobile robots can be classified as holonomic and non-holonomic ones according to the kinematic constraints. Holonomic robots can move in any direction from any state while non-holonomic robots can't. Typical holonomic robots include quadrotors and omnidirectional unmanned ground vehicles. On the other hand, typical non-holonomic robots include differential-drive and car-like models. A differential-drive model can be treated as one type of unicycle kinematic model, where the body orientation with angular velocity constraints is considered. The car-like model is often known as a bicycle kinematic model, with a steerable front wheel and a fixed rear wheel for simplification [5]. Usually, the commercial car in daily life can be regarded as a bicycle model, whose two steerable wheels can be simplified as a whole. Ignoring the angular speed of the steerable wheel, the turning rate of the car-like model is related to the robot speed and the steering curvature. The car-like model thus needs high-order planning and complex control schemes in certain environments.

The VO methods have been applied to non-holonomic robots. Van's work [6] and Javier's work [7], [8] employ different methods to apply the optimal reciprocal collision avoidance (ORCA) [9], [10] to the non-holonomic robots and show the feasibility of the VO methods to handle the non-holonomic kinematic constraints. In [11] and [12], VO is redefined to apply to a Dubins-like mobile robot, which is a non-holonomic system and can only move forward. In this article, differential-driving and car-like kinematic constraints are incorporated into the proposed MVO algorithm.

A. RELATED WORK

In this section, a brief introduction of prior works of collision avoidance algorithms is presented, where the robots are navigated in dynamic scenarios. The obstacle avoidance algorithms include global and local path planning algorithms according to environmental perception information

When the knowledge of the environment has been known in prior, off-line global path planning can be employed to navigate robots through cluttered environments. Global path planning is a well-studied research area [13]. Rapidly exploring random trees (RRT) [14] algorithm is efficient to solve the global path planning problem. The RRT algorithm creates a tree constructed by the possible actions from the initial point to the target point. In [15], the RRT-Connect method is proposed to search the feasible paths, with smaller calculations. The presence of unknown obstacles in a dynamic environment requires path replanning. In [16], a sampling-based path optimization method and the positive perception criterion are combined to minimize the reprogramming risk by predicting the future uncertainty of the map. This method ensures a faster and safer path. The graph searching method [17] is a heuristic searching method aiming to minimize the cumulative cost from the first step to the last step. A coordination mechanism is proposed in [18] to avoid the collision among the robots, which solves the problem of robot path overlap, and the robot's journey is much less than the heuristic searching method.

However, in most cases, the whole map information could not be fully obtained in prior except for the target direction. Robots can obtain local map information using onboard sensing. Real-time local path planning to generate a collision-free path using local map information is desirable for mobile robots with limited computational capacities. However, in practical applications, the sensing field of view (FOV) is usually limited. In [19], the maximum dispersion algorithm uses a local map in a limited field of view and backtracking maneuvers. In [20] and [21], motion primitives are generated to guide a quadrotor through a forest and evaluated using a local map. Local path planning is also called on-line path planning. An artificial field (APF) method [22] and its variants [23] are usually employed to generate collision-free trajectories. In the APF-based methods, repulsive forces are computed to push the robot far away from the obstacles, while attractive forces are computed to draw the robot to the target positions. In [24], a UAV group mobility model based on the APF principle is proposed, where the UAVs cooperate by sharing obstacle data within the observation range. In this way, the UAV can not only avoid obstacles within the perceived range but also avoid obstacles outside the perceived range.

Model predictive control (MPC) [25] is efficient to solve control problems on dynamic systems with input constraints or other constraints. For example, the boundary of obstacles can be considered as position constraints. With the kinematic and dynamic constraints of the robot, the MPC method can obtain the control values of the next several steps meanwhile minimizing the objective function. It can also be combined with other collision avoidance methods, such as optimal reciprocal collision avoidance (ORCA) method [26]. Yet, the MPC-based method may meet unsolvable situations because of too many constraints. Considering the uncertainty of the quadrotor model and the external disturbance, a new nonlinear model predictive control (NMPC) optimization technique is proposed in [27], which incorporates the model parameter and initial state uncertainties into path planning.

When calculating the quadrotor's path, the specified safety margin should be reached to avoid obstacles. This method is combined with online model identification to ensure robust obstacle avoidance behavior.

Another flexible method is based on the velocity region. To solve the multi-agent navigation in cluttered scenarios, the robot needs to replan local paths and actions in real-time. Velocity obstacle-based methods are suitable to generate a series of collision-free actions and also popular in solving multi-agent navigation.

The VO method and its variants, especially ORCA, are widely used in real-world tasks. In the work of [28], a method modified from ORCA has been adopted in the video games to lead the virtual humans performing collision avoidance in clustered scenarios. The work by [29] presents a decentralized method for group-based coherent and reciprocal multi-agent navigation applying ORCA. ORCA is also used in the field of aerial robots. In [30], ORCA sets a scenario with the AR-Drones which estimates the other UAVs' positions and velocities by the on-board cameras. Another work [31] designs a system with the ability to let multiple UAVs cooperate and track the preset paths under unreliable velocity estimates. In [32], an exhibit of multiple holonomic robots system is presented in the National Museum of Mathematics, which uses a hybrid ORCA method and improves the ability of noise tolerance. In [33], a new idea is proposed using discrete motion planning and optimal Finite-time Velocity Obstacle. In [34], a pedestrian is regarded as an agent, and the ORCA is applied for the motion planning process.

The first idea of the VO method is proposed in [35]. After that, many variants of the VO method are presented, such as the reciprocal velocity obstacle (RVO) [36], ORCA and the extended velocity obstacle (EVO) [32], etc. The RVO and ORCA simplify the velocity obstacle in the VO and avoid the problem of oscillations. These two methods come up with a concept of half responsibility in the navigation of multiple cooperative agents. The VO algorithm and its variants generate collision-free paths by setting a time window. If the time window is shorter, it leads to a higher probability of collision; while the time window is longer, the prediction accuracy of the obstacle position will be lower. Therefore, setting the time window of the path planning method is not desirable. A framework for dynamic risk assessment is proposed in [37], in which the forward random reachable set is used to predict the distribution of obstacles in the time window and balance the risks caused by the near and far obstacles. In [38], a VO-based algorithm is presented to handle the high-speed obstacle in dynamic scenarios. In this article, the work [38] is systematically and thoroughly investigated considering the limited sensing field of view and kinematic constraints.

B. CONTRIBUTION

In this article, a novel method derived from the traditional velocity-based method is proposed to avoid collision with high-speed obstacles. The motion planning of a mobile robot is investigated considering its limited field of view in the presence of multiple dynamic obstacles including high-speed ones. The main contributions of the paper are summarized as follows.

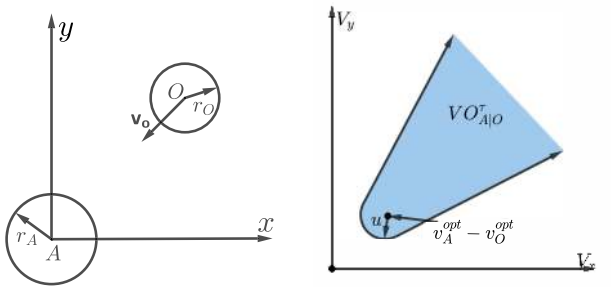
- A novel algorithm called *maximum speed aware VO* (MVO) is presented for a mobile robot to avoid high-speed obstacles, where the velocity of one or multiple dynamic obstacles is larger than the maximum velocity of the robot. Compared with the ORCA method, the proposed MVO algorithm expands the velocity obstacle region into two parts, where the newly introduced region foresees collisions beyond the time horizon to ensure feasible solutions at the next control step. Hence, the MVO algorithm achieves better performance to avoid collision with high-speed obstacles.
- In practical applications, the field of view of the robot is usually limited, and the local planner can only avoid obstacles within its perception range. In addition, non-holonomic robots unable to move into any direction are much more common due to their lower cost. In this article, the limited field of view and non-holonomic kinematic constraints are incorporated into the MVO framework. The mathematical expressions of kinematic constraints of car-like and differential-drive robots are presented, which are employed as kinematic constraints of the MVO algorithm. Hence, the MVO method can handle different non-holonomic kinematic models.
- The smoothness of the generated path is examined via analyzing the continuity of the MVO approach. In addition, safety related to the limited sensing distance as well as the computational complexity of the MVO method are analyzed and presented. Extensive simulations and physical experiments are conducted to verify the efficiency of the MVO method. In physical experiments, robotic platforms including a holonomic quadrotor and a differential-drive robot are used to perform dynamic obstacle avoidance.

C. PAPER STRUCTURE

The paper is organized as follows. Sect. II states the problem concerned in this article. The main method is presented in Sect. III, in which the usage of our method is fully discussed. In Sect. IV, we discuss the efficacy of the algorithm on the practical usage scenarios in the multi-obstacle scenarios and combine this method with the holonomic and non-holonomic kinematic models. In Sect. V, we present the continuity analysis, safety analysis and computational complexity analysis on the MVO method. Sect. VI presents the results of simulations and experiments, in which the UAV and Jackal differential drive platforms are used to perform collision avoidance in the clustered scenarios. In Sect. VII we conclude the work in this article and present the future expectation.

II. PROBLEM FORMULATION

In this section, the navigation problem concerned in this article is defined. In order to make the problem definition



(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.

FIGURE 1. Configurations of a robot *A* and a dynamic obstacle *O*.

more understandable, we describe the entire scene in a global framework. The robot often has a perception distance and a limited field of view. For velocity-based method, the velocity estimation should be within the perception range. Reference [1] proposes a method which focus on the limited sensing velocity obstacle method. They provide sufficient proof for guaranteeing collision avoidance motion. Here is the problem formulation of a single robot and multi obstacles situation for example. There is a 2D scenario where a robot A moves to its target point and may run into the obstacles O_i $(i = 1, 2, \ldots)$ with high velocity moving in any probable direction.

For simplicity, the shapes of robot and obstacles are regarded as discs, which center at $\mathbf{p}_A$ and $\mathbf{p}_{O_i}$ with radii $\mathbf{r}_A$ and $\mathbf{r}_{O_i}$ (Fig.2(a)). Suppose that the speed remains fixed for a short time, and can be observed by external or on-board sensors. The robot A also has a heading angle θ_A, and a heading direction $\mathbf{d}_A = [cos(\theta_A), sin(\theta_A)]^T$. The allowed velocity set is defined as the:

$$U_A = D(0, v_A^{\max}) \cap S_A, \tag{1}$$

where $v_A^{\max}$ is the maximum speed of robot A. S_A is the sensing constraint for the robot, which is shown later. The limited field of view of the robot is defined as:

$$F_A = \{\mathbf{p} \mid \|\mathbf{p} - \mathbf{p}_A\| < r_A^{\mathrm{s}}, |\angle(\mathbf{d}_A, \mathbf{p} - \mathbf{p}_A)| \leq \alpha_A^{\mathrm{s}}\}, \tag{2}$$

where r_A^{s} is the sensor distance range and α_A^{s} is the sensor angle range within $(\frac{\pi}{2}, \pi)$. $\mathbf{p}$ is the position of robot or obstacle. Therefore, to keep the obstacle within the robot's sensing range and to control the robot A move away from the obstacle O, we define the sensing constraint set for the robot's velocity as follows when the obstacle is detected by the robot:

$$S_A = \{\mathbf{v} \mid |\angle(\mathbf{v}, \mathbf{d}_A)| \leq \alpha_A^s - \frac{\pi}{2}\}. \tag{3}$$

As shown in Fig. 2, here define a line $\mathrm{L}\perp$ $(\mathbf{p}_O - \mathbf{p}_A)$ with the distance of $\frac{|\mathbf{p}_O - \mathbf{p}_A|}{2}$ between robot A and obstacle O, respectively. With the velocity in the set S_A, a robot tends to choose a velocity near the detection direction to make the angle of velocity change smoothly.

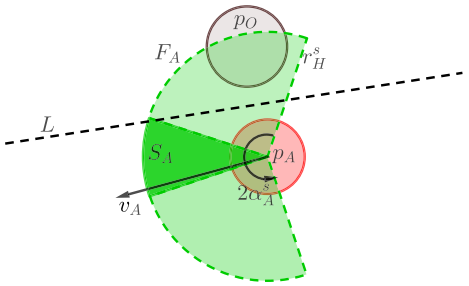


FIGURE 2. The limited field of view of the robot.

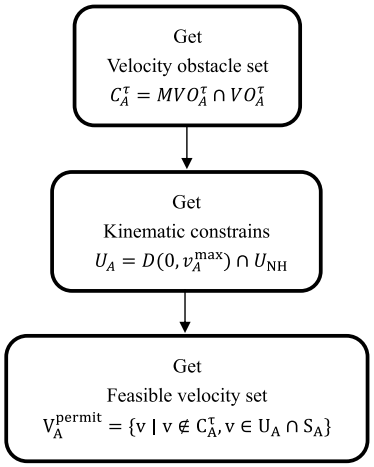


FIGURE 3. Flowchart of proposed algorithm.

The preferred speed $\mathbf{v}_A^{\mathrm{pref}}$ is used to guide the robot to move in the direction of the target point. The robot generates the new optimal velocity as input in every control period, by considering these following factors: the position and velocity information of the robot and obstacles, the preferred velocity $\mathbf{v}_A^{\mathrm{pref}}$, the radii, the time horizon τ, and sensor constraints S_A.

$$\mathbf{v}_A^{\mathrm{new}} = F(\mathbf{v}_A, \mathbf{p}_A, \{\mathbf{v}_{O_i}\}, \{\mathbf{p}_{O_i}\}, r_A, \{r_{O_i}\}, \mathbf{v}_A^{\mathrm{pref}}, v_A^{\max}, \tau, S_A). \tag{4}$$

Here Maximum-speed Aware Velocity Obstacle Method framework which is presented in Sect. III generates a velocity obstacle and chooses a new robot velocity which is nearest to the preferred velocity $\mathbf{v}_A^{\mathrm{pref}}$.

The whole process can be briefly summarized in three parts(Fig. 3):

- In the section III, we discuss how to build the velocity obstacle set $C_A^\tau = MVO_A^\tau \cap VO_A^\tau$, which contains our new method MVO. The velocity obstacle set means the set of velocities where a robot cannot choose a new desired velocity therein.
- In the section IV, we discuss how to use the MVO method on different kinematic robot models by adding a kinematic constrains U_{NH}. Here we take the differential-drive robot and car-like robot for example.

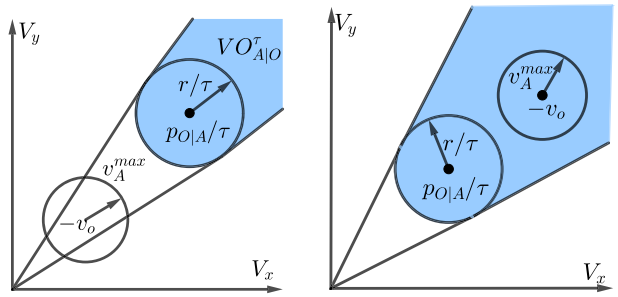


FIGURE 4. (a) When $\mathbf{v}_O$ is higher than the robot's maximum velocity, the relatively feasible cycle does not contain the original point. (b) When $\mathbf{v}_O$ gets higher, the relatively feasible velocity cycle may be fully contained by the velocity obstacle.

- At last, a new velocity $\mathbf{v}_A^{\text{new}} \in V_A^{\text{permit}}$ which should be the closest to the preferred velocity $\mathbf{v}_A^{\text{pref}}$ is chosen for robot A.

III. MAXIMUM-SPEED AWARE VELOCITY OBSTACLE METHOD

Due to the limited field of view of a robot, the obstacles can only be observed and avoided when they are close enough to the robot. This also means that the robot only has a short reaction time when avoiding obstacles, especially high-speed obstacles, and the collision probability increases. Hence, for robots with a limited perception, it is necessary to take account of the ability to avoid obstacles at high speed into path planning. In this section, the maximum-speed aware velocity obstacle (MVO) method is proposed. MVO complements the velocity obstacle region in *optimal reciprocal collision avoidance* (ORCA). Therefore, a brief review of the ORCA and its velocity obstacle region is necessary.

A. OPTIMAL RECIPROCAL COLLISION AVOIDANCE METHOD

For robot A and obstacle O, $VO^{\tau}_{A|O}$ is defined as the velocity obstacle of A for obstacle O. It means that if the relative velocity is chosen in the $VO^{\tau}_{A|O}$, the collision will occur within the time interval τ. The set of relative velocities which will cause collision within τ can be represented as follow:

$$VO^{\tau}_{A|O}\left(p_{O|A}\right) = \{\mathbf{v} \,|\exists t \in [0, \tau], t\mathbf{v} \in D\left(p_{O|A}, r_{AO}\right)\}, \quad (5)$$

where $D(p, r)$ is a disc of radius r centered at p, r_{AO} is the sum of r_A and r_O. If a robot keeps moving with a velocity out of the velocity obstacle, the collision will occur (Fig. 4(b)). In the ORCA method, the collision-free velocity set is formulated as

$$ORCA^{\tau}_{A|O} = \left\{\mathbf{v}|\left(\mathbf{v} - \left(\mathbf{v}_A^{\text{new}} + \lambda\mathbf{u}\right)\right) \cdot \mathbf{n} \geq 0\right\}, \quad (6)$$

where the $\mathbf{v}_A^{\text{new}}$ is the current velocity of the robot. $\mathbf{u}$ denotes the minimum velocity change which leads the relative velocity out of $VO^{\tau}_{A|O}$ and $\mathbf{n}$ is the outward normal (Fig. 1(b)). Coefficient λ is the responsibility coefficient, which determines how much responsibility the robot should undertake.

For robot A, the control input $\mathbf{v}_A^{\text{new}}$ of next step is a velocity nearest to the preferred velocity $\mathbf{v}_A^{\text{pref}}$,

$$\mathbf{v}_A^{\text{new}} = \underset{\mathbf{v} \in ORCA^{\tau}}{\arg\min} \left\|\mathbf{v} - \mathbf{v}_A^{\text{pref}}\right\|. \quad (7)$$

The ORCA method has been popularly used in many multi-agent scenarios. It solves the oscillation between the multiple agents and enlarges the feasible velocity set. In most situations, the feasible velocity set of robot $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ must be considered in the navigation task. When $\mathbf{v}_O$ gets higher, the $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ becomes more distant from the ordinate origins, like Fig. 4(a). For even higher velocity $\mathbf{v}_O$, the feasible velocity set of robot A would have a higher probability of being fully contained in the velocity obstacle region (in Fig. 4(b)), which makes the robot cannot escape the velocity obstacle region. It should keep awareness of unsolvable situations. The following sections bring a new conception to solve this kind of problem.

B. AVOIDING HIGH-SPEED OBSTACLE METHOD

The ORCA method proposes an idea to address the problem of avoiding multi-agents collision and oscillations. But when τ is set at the beginning, the ORCA method has not taken the high-speed obstacles and the limited maximum robot's velocity into account. In this section, we give a concise description of our method to handle this kind of problem, by increasing the additional velocity obstacle.

Since the limited field of view of the robot is independent of the analysis of avoiding high-speed obstacles problems, the analysis of MVO algorithm in this section assumes that the robot only has the maximum speed limit, i.e.:

$$U_A = D(0, v_A^{\max}). \quad (8)$$

1) BASIC INTRODUCTION

To avoid the situation unsolvable, here we consider the choice of optimal velocity at the current step. Suppose that there is a feasible velocity set when we choose a new velocity, meanwhile, we want to have a solution at the next control step. Here we denote the new velocity obstacle region as $MVO^{\tau}_{A|O}\left(p_{O|A}\right)$, which is a set of relative velocity bringing collision in the next time horizon.

$$MVO^{\tau}_{A|O}(p_{O|A}) = \{\mathbf{v}|D(-\mathbf{v}_O, v_A^{\max}) \subset VO_{A|O}(p_{O|A} - \mathbf{v} \cdot \tau)\}. \quad (9)$$

For robot A and obstacle O, the relative position is denoted as $p_{O|A} = p_O - p_A$. Thus, the new relative position at the next control step would be $p^{\text{new}}_{O|A} = p^{\text{new}}_O - p^{\text{new}}_A$ and because the velocity of the obstacle is supposed as a constant in a single period, $p^{\text{new}}_O = p_O + \mathbf{v}_O \cdot \tau$. Thus, the new relative position $p^{\text{new}}_{O|A} = p_{O|A} - \mathbf{v}^{\text{new}}_{A|O} \cdot \tau$. There, we can see the relationship between the chosen velocity $\mathbf{v}^{\text{new}}_{A|O}$ and the new relative position $p^{\text{new}}_{O|A}$ is:

$$\mathbf{v}^{\text{new}}_{A|O} = (p_{O|A} - p^{\text{new}}_{O|A})/\tau, \quad (10)$$

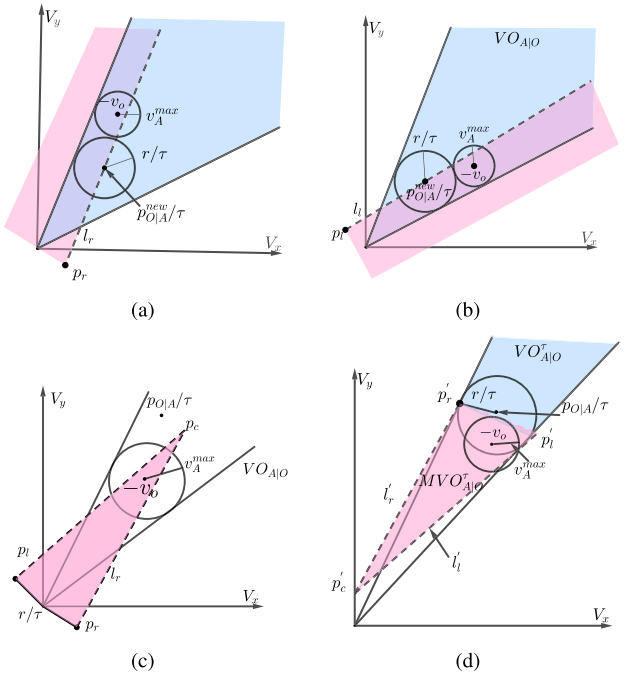


FIGURE 5. The derivation of a new velocity obstacle.

where $\mathbf{v}_{A|O}^{\text{new}}$ is nearest to its preferred velocity $\mathbf{v}_{A|O}^{\text{pref}}$ amongst all velocities inside the feasible velocity region. According to the concept of the VO method, the position of the velocity obstacle depends on the relative position $p_{O|A}^{\text{new}}$. To prevent the whole feasible velocity set $D\left(\text{-}\mathbf{v}_O, v_A^{\max}\right)$ from being all contained in the $VO_{A|O}\left(p_{O|A}^{\text{new}}\right)$, the new relative position $p_{O|A}^{\text{new}}$ should be specially selected.

As is shown in Fig. 5(a) and Fig. 5(b), the $D\left(\text{-}\mathbf{v}_O, v_A^{\max}\right)$ is fixed and we can adjust the velocity obstacle by selecting the $p_{O|A}^{\text{new}}$. l_l and l_r are the upper and lower bound of the $p_{O|A}^{\text{new}}$, which are also the boundary tangent to the disc $D\left(\text{-}\mathbf{v}_O, v_A^{\max}\right)$. If $p_{O|A}^{\text{new}}$ falls between l_l and l_r, the robot will have no chance to escape the new velocity obstacle region. The two half-lines l_l and l_r together with the two segments $0-\mathbf{p}_l$ and $0-\mathbf{p}_r$ encompass the largest set of points (Fig. 5(c)).

According to Eq.(10), we can get $MVO_{A|O}^{\tau}\left(p_{O|A}\right)$ in Fig. 5(d). l_l' and l_r' are the boundary of $MVO_{A|O}^{\tau}\left(p_{O|A}\right)$ which are mapped from l_r and l_l. $\mathbf{p}_c'$ is the intersection of l_l' and l_r'. And the boundary of $MVO_{A|O}^{\tau}\left(p_{O|A}\right)$ is tangent to the cycle $D\left(p_{O|A}/\tau, r/\tau\right)$ with $\mathbf{p}_\mathbf{l}'$ and $\mathbf{p}_\mathbf{r}'$.

It is more likely to occur when $v_A^{\max}$ is far smaller than $\mathbf{v}_O$, as is shown in Fig. 4. From another perspective, if each step the robot chooses a velocity that is outside $MVO_{A|O}^{\tau}\left(p_{O|A}\right)$ and $VO_{A|O}^{\tau}\left(p_{O|A}\right)$, the robot will always have solutions. So the set of velocity obstacle regions is the intersection of MVO_A^{τ} and VO_A^{τ}.

$$V_{A|O}^{\text{permit}} = \{\mathbf{v}|\mathbf{v} \in D\left(-v_O, v_A^{\max}\right) \cap S_A, \quad \mathbf{v} \notin \left(MVO_{A|O}^{\tau} \cap VO_{A|O}^{\tau}\right)\}. \tag{11}$$

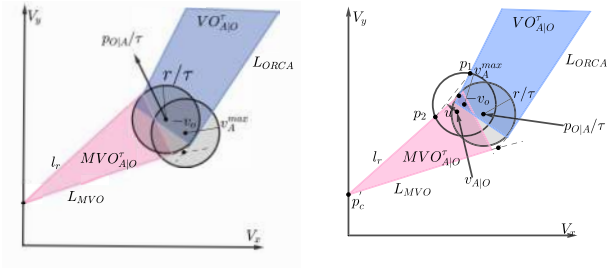


FIGURE 6. Two kinds of simplification.

The new velocity of robot A is given by the calculation in following and the sensor constraints should also be considered.

$$\begin{aligned} \mathbf{v}_{A|O}^{\text{new}} &= \arg\min_{\mathbf{v} \in V_{A|O}^{\text{permit}}} \left\| \mathbf{v} - \mathbf{v}_{A|O}^{\text{pref}} \right\|, \\ \mathbf{v}_A^{\text{new}} &= \mathbf{v}_{A|O}^{\text{new}} + \mathbf{v}_O. \end{aligned} \tag{12}$$

When enlarging τ, the velocity obstacle area will become larger and the robot tends to be farsighted. But it will bring the risk of getting no feasible velocity area when the number of obstacles become larger. On the other side, when the parameter τ gets smaller, the robot will react quickly to other robots and obstacles in its path [9], [10].

However, a small τ may result in less smooth paths as well, since the robot may have to frequently rotate to achieve its target orientation. In practice, choosing τ as three times the planning period can yield good results.

2) SIMPLIFICATION

To simplify the calculation and avoid the concave set, here we do some simplification. See Fig. 6(a) and Fig. 6(b), these two kinds of situations often happen when choosing an optimal velocity. In Fig. 6(a), when the feasible velocity set makes intersection with $D\left(-\mathbf{v}_O, v_A^{\max}\right)$, it appears a concave region constructed by ORCA boundary l_{ORCA}, MVO boundary l_{MVO}, and the arcs. Here, we choose to prolong the l_{ORCA} or l_{MVO} to make a new velocity region, which is a convex set. And in Fig. 6(a), the VO set intersects the circle $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ at two points $\mathbf{p}_1$ and $\mathbf{p}_2$. For convenience, we connect $l_{p_1p_2}$ as a new boundary of the VO region.

The MVO algorithm also has limitations, that is, the set of feasible speeds of the robot is reduced due to the additional velocity obstacle area in the second prediction period, which may cause the absence of a feasible solution. This limitation would become more apparent as the number of obstacles increases.

IV. PRACTICAL USAGE OF MVO METHOD

To show the efficacy of the algorithm on the practical usage scenarios, here we discuss the practical usage of this method in the multi-obstacle scenarios and combine this method with the holonomic and non-holonomic kinematic models,

including differential-drive and car-like models. After considering the limited field of view of the robot, the whole method shows efficacy in the practical usage scenario.

A. MULTIPLE OBSTACLES AVOIDANCE SCENARIO

The abilities of collision-free navigation and smooth action planning in multiple obstacle scenarios are among the basic requirements for every working robot. Therefore, our MVO method should also consider the increase of robot density and cooperative interaction, and avoid collision and oscillation. A method similar to the case of a single robot can be applied to the collision avoidance scenarios of multiple robots. As the same to the single-obstacle scenario, the robot here is supposed to have a continuous cycle of sensing and the time horizon τ. In every control step, it can detect the information of obstacles' positions, velocities and radius. When the robot encounters a collision, each obstacle makes the robot generating these two velocity obstacle regions, $MVO^{\tau}_{A|O_i}\left(p_{O_i|A}\right)$ and $VO^{\tau}_{A|O_i}\left(p_{O_i|A}\right)$.

Then calculate the velocity obstacle regions based on $MVO^{\tau}_{A}=\bigcap(MVO^{\tau}_{A|O_i}\left(p_{O_i|A}\right)\oplus V_{O_i})\ (i=1,2,\ldots)$ and $VO^{\tau}_{A}=\bigcap(VO^{\tau}_{A|O_i}\left(p_{O_i|A}\right)\oplus V_{O_i})\ (i=1,2,\ldots)$, where suppose $X\oplus Y$ is the Minkowski sum of X and Y.

$$X\oplus Y=\{x+y|x\in X,y\in Y\}. \tag{13}$$

Let $V_{O_i}=\left\{\mathbf{v}_{O_i}\right\}$. Every velocity region produces a feasible velocity set. So the set of velocity obstacle regions is the intersection of MVO^{τ}_{A} and VO^{τ}_{A}.

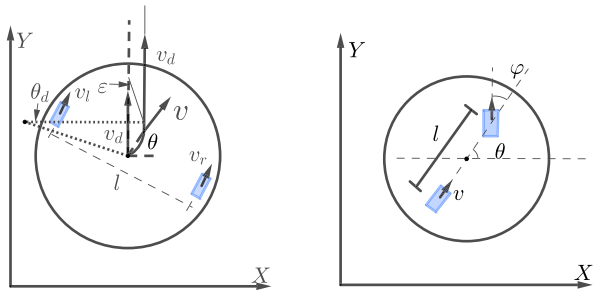
$$V^{\text{permit}}_{A}=\{\mathbf{v}|\mathbf{v}\in D\left(0,v^{\max}_{A}\right)\cap S_A,\mathbf{v}\notin\left(MVO^{\tau}_{A}\cap VO^{\tau}_{A}\right)\}. \tag{14}$$

B. SCENARIO WITH KINEMATIC CONSTRAINTS

Non-holonomic robots are much more common in practical applications due to their lower cost. Typical non-holonomic robots include differentially-driven and car-like vehicles. As non-holonomic robots cannot move to any direction from any state, differential-drive and car-like kinematic constraints are incorporated into the MVO method such that the proposed method can be applied to different kinematic models. In [7], an ORCA-based method is proposed considering differential-drive kinematic model. Motivated by the work [7], the non-holonomic constraint formulas of differential-drive and car-like vehicles are derived, which can be easily used as the kinematic constraints of the MVO algorithm. Hence, the MVO approach can be applied to different kinematic models.

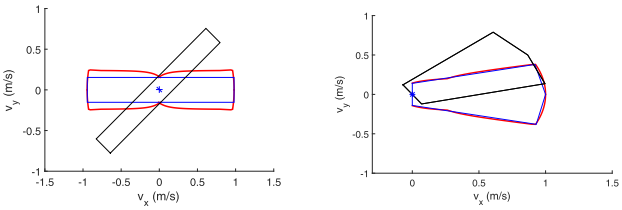
1) DIFFERENTIAL-DRIVE ROBOT

Since the non-holonomic constraint formula of the differential-drive vehicle has been derived in [7], it is briefly described here. For more details, it can be referred to [7]. As shown in Fig. 7(a), the differential-drive robot's position in the two-dimension space can be described by three state parameters, including the position of center (x, y) and the orientation θ.



(a) Differential-drive robot (b) Car-like robot

FIGURE 7. Two kinds of non-holonomic robots.



(a) The case of differential-drive model. (b) The case of the car-like model.

FIGURE 8. Two kinds of kinematic model constraints.

The final motion constraint U_{NH} which is derived by the different kinematic constraint model is shown in Fig. 8(a), the red line boundary reveals that it is not a linear calculation in the velocity obstacle. To simplify the calculation, we choose the inscribed polygon to replace the non-linear space. Because of rotational invariance, the constraint-rectangle can be computed at zero orientation and rotated to match the robot's current orientation θ. After that, the black box is the final motion constraint U_{NH}. And then, we combine this velocity constraint area into the feasible velocity region calculated by MVO, which is shown in Fig. 9(a).

2) CAR-LIKE ROBOT

The car-like robot, as the name implies, is similar to a car. We often simplify the car model with a fixed rear wheel and a steerable front wheel like a bicycle, as shown in Fig. 7(b). Consider the car-like robot rotates at the midpoint between steering and rear wheel.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}\cos(\theta)-\tan(\varphi)\sin(\theta)\\ \sin(\theta)+\tan(\varphi)\cos(\theta)\\ 2\tan(\varphi)/l\end{bmatrix}v, \tag{15}$$

where l is the distance between the steering wheel and rear wheel. The car-like robot has three state variables $[x, y, \theta]$, and the control inputs are $[v, \varphi]$. $\varphi\in[-\varphi_{\max},\varphi_{\max}]$ is the steering wheel angle and $v\in[-v_{\max},v_{\max}]$ is the rear wheel's velocity. Because of the kinematic model, angular velocity is depended on φ and v. Let $\omega=\dot{\theta}\in[-2v\tan(\varphi_{\max})/l, 2v\tan(\varphi_{\max})/l]$. When φ reaches $\varphi_{\max}$, the maximum of angular velocity still changes along with

v. Here we suppose that in the permitted range, the steering wheel can reach any angle immediately. Like the differential-drive robot, it needs a curved path to reach the needed angle, which will bring errors. The curved path brings the tracking error and here are several constraints in these variables as below. And we can find that the equation is quadratic for the v or ω,

$$\varepsilon^2 = v_d^2 t^2 - \frac{2v_d t \sin(\theta_d)}{\omega} v + \frac{2(1-\cos(\theta_d))}{\omega^2} v^2. \quad (16)$$

We control the error within a given ε and add the velocity constraints into two-period velocity obstacle regions. Here we suppose that the desired speed has a higher priority.

$$\begin{aligned} v &= \min(v_m, v_{\max}), \\ \omega &= \min\left(\frac{\theta_d}{T}, \frac{2v\tan(\varphi_{\max})}{l}\right). \end{aligned} \quad (17)$$

T is the maximum time in which the robot needs to reach the desired velocity direction. Thus we can get the maximum path tracking error for the following conditions $[a, b, c, d]$.

$$v_d^{\max} = \begin{cases} \min\left(\frac{\varepsilon_{\max}}{T}\sqrt{\frac{2(1-\cos(\theta_d))}{2(1-\cos(\theta_d))-\sin^2(\theta_d)}}, v_{\max}\right), (a) \\ \min\left(\frac{-\beta_1+\sqrt{\beta_1{}^2-4\alpha_1\gamma_1}}{2\gamma_1}, v_{\max}\right), (b) \\ \min\left(\frac{-\beta_2+\sqrt{\beta_2{}^2-4\alpha_2\gamma_2}}{2\gamma_2}, v_{\max}\right), (c) \\ \min\left(\frac{-\beta_3+\sqrt{\beta_3{}^2-4\alpha_3\gamma_3}}{2\gamma_3}, v_{\max}\right), (d) \end{cases} \quad (18)$$

where $[\alpha_i, \beta_i, \gamma_i]$ $(i = 1, 2, 3)$ are the quadratic term, linear term and coefficient term of the Eq.(18). Similarly, the black box is the final motion constraint U_{NH}, which is shown in Fig. 8(b). The robot feasible velocity set U_{NH} should be combined with V_A^{permit} concerned above. As shown in Fig. 9(b), the new permitted velocity set is:

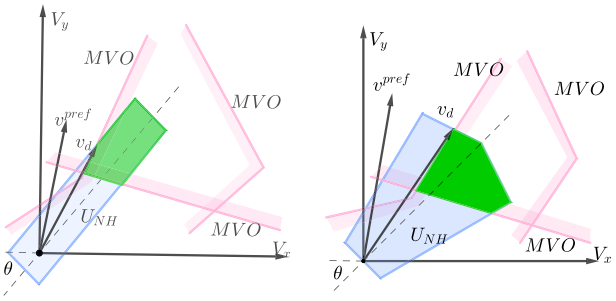
$$V_A^{\mathrm{permit}} = \{\mathbf{v} | \mathbf{v} \in D(0, \mathbf{v}_A^{\max}) \cap S_A \cap U_{\mathrm{NH}}, \mathbf{v} \notin (MVO_A^\tau \cap VO_A^\tau)\}. \quad (19)$$

After getting the permitted velocity of the robot, it should choose the new desired velocity which is closed to the preferred velocity. The preferred velocity is used to guide the robot to move in the direction of the target point to the greatest extent.

$$\mathbf{v}_A^{\mathrm{new}} = \arg\min_{\mathbf{v} \in V_A^{\mathrm{permit}}} \left\| \mathbf{v} - \mathbf{v}_A^{\mathrm{pref}} \right\|. \quad (20)$$

V. ANALYSIS

In this section, the continuity analysis of the MVO method is presented, which ensures the smoothness of the generated actions. Safety analysis of the MVO approach is also presented in the scenario with the limited sensing distance range. Furthermore, by analyzing and comparing the computational complexity of ORCA and MVO algorithms, it is shown that the MVO algorithm will not increase the computational complexity.



(a) The case of differential-drive model. (b) The case of car-like model.

FIGURE 9. Combine a feasible velocity region with MVO. The green region in the figures shows the intersection of kinematic constraints U_{NH} and velocity feasible region by generated by MVO. We can observe the shapes of the differential-drive and the car-like model constraints.

A. CONTINUITY ANALYSIS OF MVO METHOD

Like the ORCA method, our MVO could also generate the collision-free and smooth actions in the position region, in other words, the selected velocities are continuous.

First, we denote Δt as a small-time step and $\mathbf{v}_A(t)$ as the current velocity in the sequence of $\mathbf{v}_A$. And $\mathbf{v}_A(t+\Delta t)$ is the velocity in the next time step, but also the selected velocity at the current time step. To analyze the continuity of $\mathbf{v}_A$, it needs to deduce the $\mathbf{v}_A(t+\Delta t) \approx \mathbf{v}_A(t)$, when $\Delta t \to 0$. By the method of induction, from $\mathbf{v}_A(t+\Delta t) \approx \mathbf{v}_A(t)$, we will get $\mathbf{v}_A(t+2\Delta t) \approx \mathbf{v}_A(t+\Delta t)$, when the Δt is small. So, to prove $\mathbf{v}_A(t+\Delta t) \approx \mathbf{v}_A(t)$, it needs to prove that $\mathbf{v}_A(\Delta t) \approx \mathbf{v}_A(0)$. As yet, it is clear that we should focus on the method to get $\mathbf{v}_A(t+\Delta t)$ and the initialization velocity $\mathbf{v}_A(0)$.

The selected optimal velocity of the robot $\mathbf{v}_A^{\mathrm{new}} = \mathbf{v}_A(t+\Delta t)$ is based on the MVO method can be generated by the function $\mathbf{v}_A(t+\Delta t) = f(h(t), g(t), q(t))$, where the $h(t) = \mathbf{v}_A^{\mathrm{pref}}(t)$ and $g(t) = MVO_A^\tau(t)$, $q(t) = VO_A^\tau(t)$.

The preferred velocity $\mathbf{v}_A^{\mathrm{pref}}(t)$ is calculated by the current position $\mathbf{p}_A(t)$ and the target position $\mathbf{p}_A^{\mathrm{goal}}$. It is clear that the $\mathbf{p}_A(t)$ is continuous and the $\mathbf{p}_A^{\mathrm{goal}}$ is fixed. So the preferred velocity $h(t) = \mathbf{v}_A^{\mathrm{pref}}(t)$ is also continuous, where $\mathbf{v}_A^{\mathrm{pref}}(t) \approx \mathbf{v}_A^{\mathrm{pref}}(t+\Delta t)$.

The $MVO_A^\tau(t)$ is the intersection of the velocity region $MVO_{A|O_i}^\tau(t)$. So it needs to prove $MVO_{A|O_i}^\tau(t) \approx MVO_{A|O_i}^\tau(t+\Delta t)$. As the introduction proposed above, $MVO_{A|O_i}^\tau(t)$ is inferred from the velocity obstacle region based on the continuous position $\mathbf{p}_A(t)$ and the velocity $\mathbf{v}_A(t)$. So, if $\mathbf{v}_A(t)$ is continuous, the $MVO_{A|O_i}^\tau(t)$ is also continuous. Similarly, $VO_{A|O_i}^\tau(t)$ is also continuous. Thus, $f(h(t), g(t), q(t)) \approx f(h(t+\Delta t), g(t+\Delta t), q(t+\Delta t))$.

On the other aspect, to make $\mathbf{v}_A(\Delta t) \approx \mathbf{v}_A(0)$, it will occur when the robot's initial condition is beginning with $\mathbf{v}_A^{\mathrm{pref}}(0)$.

When $\mathbf{v}_A(0) = \mathbf{v}_A^{\text{pref}}(0)$, the $\mathbf{v}_A$ will begin with continuity and make the next sequence of $MVO^{\tau}_{A|O_i}(t)$ and $VO^{\tau}_{A|O_i}(t)$ continuous.

As is concerned above, if a robot can keep $\mathbf{v}_A^{\text{pref}}(0)$ as the initial velocity, the sequence of $\mathbf{v}_A(t)$ is continuous and the MVO method will generate a smooth collision-free action series.

B. SAFETY ANALYSIS CONSIDERING LIMITED SENSING DISTANCE RANGE

Generally, the robot has limited perception capability including limited sensing distance and sensing angle. It is dangerous for the robot if the obstacle runs towards its blind spot when they are passing to each other. On the other hand, according to continuity analysis of the MVO approach discussed in section V-A, if the feasible velocity circle has been fully contained at the beginning, the robot has no chance to escape from the velocity obstacle in the next steps. In this section, the effect of the limited sensing distance range on the safety of the MVO approach is analyzed. Suppose that the robot has a limited perception distance r_s (Fig.10(a)). The following theorem 1 shows the maximum velocity of a dynamic obstacle with respect to the maximum velocity of the robot, beyond which the MVO approach cannot avoid the dynamic obstacle under the condition of limited sensing distance.

Theorem 1: The extreme situation is discussed when the obstacle rushes to the robot in the velocity $\mathbf{v}_O$*. When the obstacle enters into the boundary of detection, there exists a maximum velocity of an obstacle for the robot to avoid collision.*

$$|v_O|_{upper} = v_A^{\max} \cdot r_s / r, \tag{21}$$

where $v_A^{\max}$ *is the maximum velocity of robot,* r *is the sum of radii of robot and obstacle,* r_s *is the limited perception distance.*

Proof: In the ORCA method, the extreme situation is that the feasible velocity circle $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ of a robot is contained by $D\left(-p_{O|A}/\tau, r/\tau\right)$ (Fig.10(b)). When $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ is tangent with the bottom arc of the velocity obstacle, it is the upper bound for the obstacle velocity:

$$|v_O|_{upper} = (|\mathbf{p}_{O|A}| - r)/\tau + v_A^{\max}. \tag{22}$$

In the MVO method, because of the additional part, the feasible velocity set should not be contained by the original and additional velocity obstacle. When the obstacle rushes to the robot straightly, the feasible velocity circle $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ may first meet the additional velocity obstacle $MVO^{\tau}_{A|O}(t)$, as shown in Fig.10(c). When the circle is tangent with the boundary of MVO, it can be calculated that $T_0 = -\frac{r/\tau}{v_A^{\max}} \cdot \mathbf{v}_O + p_{O|A}/\tau$, $T_1 = \mathbf{p}_{O|A}/\tau + \mathbf{v}_O$, $T_2 = -v_O$, $r_1 = r/\tau - v_A^{\max}$, and $r_2 = v_A^{\max}$. Based on homothetic triangle theory, it is found that the upper bound of obstacle's velocity in this situation,

$$|v_O|_{upper} = v_A^{\max} \cdot |\mathbf{p}_{O|A}|/r, \tag{23}$$

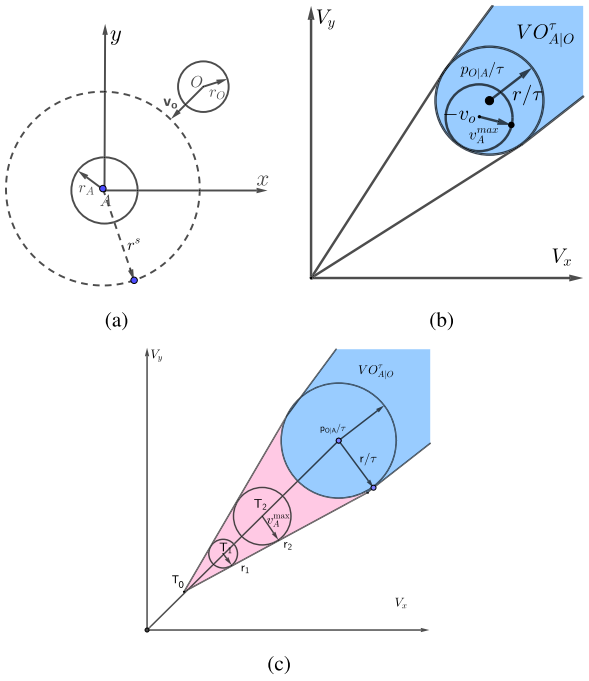


FIGURE 10. The extreme situation with a limited perception distance.

where r is the sum of radii of robot and obstacle. When there exist a detection limits r_s, the relative distance $|\mathbf{p}_{O|A}|$ between robot and obstacle can be replaced by r_s. ■

Considering the kinematic constraints of the robot, models of the robot can be divided into the holonomic and non-holonomic ones. Therefore, the following corollaries can be derived from Theorem 1.

Corollary 2: If the robot is holonomic and omnidirectional, with radius r_H*. The upper bound of the* $\mathbf{v}_O$ *can be applied to the holonomic robot.*

$$|v_O|_{upper} = v_A^{\max} \cdot r_s / r_H. \tag{24}$$

Corollary 3: If the robot is non-holonomic as concerned in section IV-B, the trajectory error is controlled within the maximum error $\varepsilon_{\max}$*. It is equal to the increasing radius of the holonomic robot so that* $r_{NH} = r_H + \varepsilon_{\max}$*. The upper bound of the* $\mathbf{v}_O$ *can be also applied to the non-holonomic robot.*

$$|v_O|_{upper} = v_A^{\max} \cdot r_s / r_{NH}. \tag{25}$$

C. COMPUTATION COMPLEXITY ANALYSIS

Like the ORCA algorithm, the MVO algorithm still maintains a low computational complexity and is suitable for quadrotor platforms with limited computational capacities. In the ORCA algorithm, the velocity obstacle area is firstly constructed referred to Eq.(6). For limited operations of the equation, the computational complexity is $O(n)$, where n is the number of obstacles. Referring to Eq.(7), it is a QP problem to calculate the optimal speed, and the computational complexity is $O(n^2)$. In the MVO algorithm, the velocity obstacle area

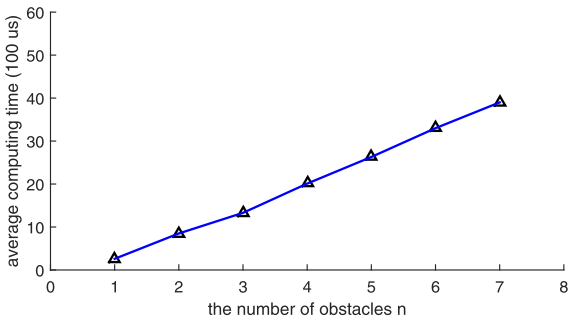


FIGURE 11. Average computing time of constuction of MVO region per execution vs the number of obstacles.

is expanded into two parts referred to Eq.(11), where the computational complexity of the construction of the VO region is $O(n)$, the computational complexity of the construction of the MVO region is $O(n)$. The computational complexity after synthesis is still $O(n)$ (Fig.11). Similarly, after the obstacle area is obtained, it is a QP problem to calculate the optimal speed referred to Eq.(12), and the computational complexity is $O(n^2)$. Besides, the limited sensing constraint and kinematics constraint are designed as linear constraints of the QP problem, which will not increase the computational complexity. Therefore, the MVO algorithm does not increase the computational complexity and applies the complex environment with high-frequency and high-speed obstacles.

VI. SIMULATIONS AND EXPERIMENTS

In this section, simulation and experimental results are presented to illustrate the performance of the MVO approach in the presence of one and multiple dynamic obstacles. Various numerical simulations are conduceted to evaluate the proposed method. Moreover, physical experiments are performed using a holonomic unmanned aerial vehicle platform Parrot Bebop 2 and a differential-drive vehicle Jackal. Experimental results comparing between the MVO and the ORCA are also presented in the presence of dynamic obstacles.

A. SIMULATION

In Fig. 12, the simulations are conducted for the scenario where a robot starts at the original point and moves to the target point with a maximum speed of 1 m/s. Meanwhile, the obstacle rushes to the robot with an average speed of 15 m/s. For comparison, the MVO and ORCA method are both implied in the simulations. There are four snapshots to show these two methods' different appearances. Using the MVO method, the robot generates effective actions before the feasible velocity cycle $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ is fully contained by the velocity obstacle. After that, the robot moves to the target point with maximum speed. But when using the ORCA method, the robot tends to move in the same direction of the obstacle. And because the $\mathbf{v}_O$ is much larger than the robot's, $D\left(-\mathbf{v}_O, v_A^{\max}\right)$ can be easily contained in the velocity obstacle. In Fig. 12(b), the region surrounded by blue lines is the MVO velocity obstacle, which makes it move aside.

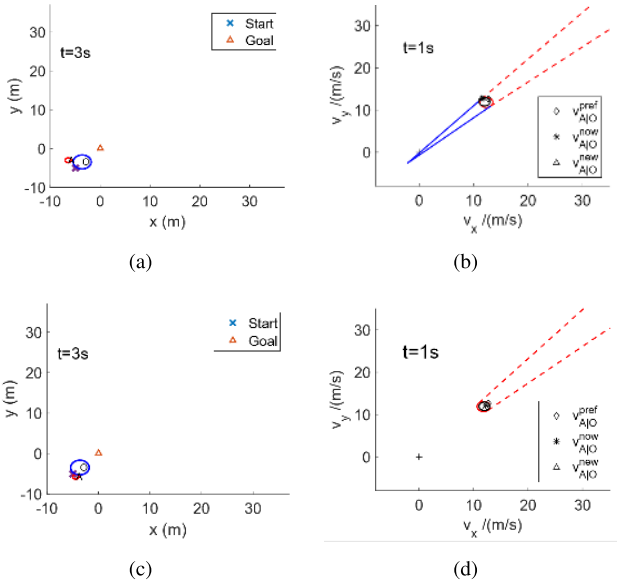


FIGURE 12. The red and blue cycle represents the robot and the obstacle, respectively. The speed of the obstacle is almost 15 m/s and the maximum speed of robot is 1 m/s. (a) The robot is navigated with the MVO method, and at 3s, the robot avoids collision successfully. (b) This figure is the velocity obstacle, the region constructed by the blue line is the MVO and the red region is the velocity obstacle generated by ORCA. We can observe that the velocity feasible cycle is in the MVO so that the robot could generate the actions ahead of time. (c) The robot is navigated by ORCA, and collision occurs at 3s. (d) We can observe that the ORCA method have not enough awareness before the velocity feasible cycle is fully contained by the velocity obstacle region.

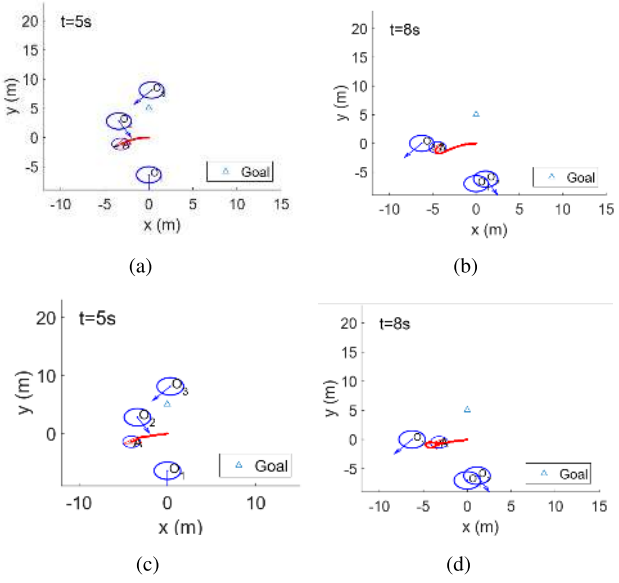


FIGURE 13. (a) and (b) Is the scenario where the robot is navigated by the method without considering kinematic constraints. Before the collision occurs, the MVO method generates the optimal velocity which can't be arrived for the robot. (c) and (d) is the scenario where the robot is navigated by the MVO method with kinematic constraints, and we can see the robot's action is smooth and collision-free.

In Fig. 13, when using MVO without considering kinematic constraints, the robot pays not enough attention to choosing optimal velocity. It will lead the robot cannot reach

TABLE 1. The variances of the robot's actions under different collision avoidance algorithms.

	Variance
ORCA	1.0417
MVO	0.9879
MVO-NH	0.6447

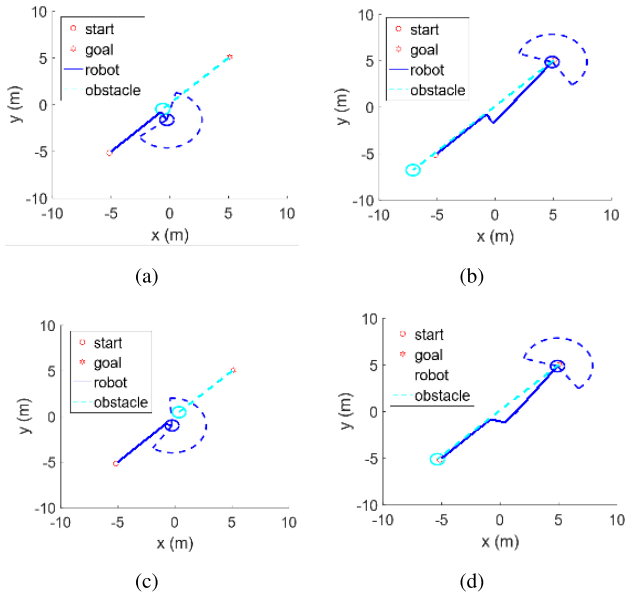


FIGURE 14. Blue and cyan circles are the robot and obstacle, respectively. The blue solid line and cyan dotted line are robot's path and obstacle's path, respectively. The blue circular sector is the sensing range. (a) Obstacle gets away from the robot's sensing range when the robot is back on the obstacle at 13s. (b) The path generated by the MVO method is not smooth enough and oscillation occurs. (c) The robot keeps the obstacle within its sensing range before passing it. (d) The path generated by the MVO method with limited sensing constraints is smooth.

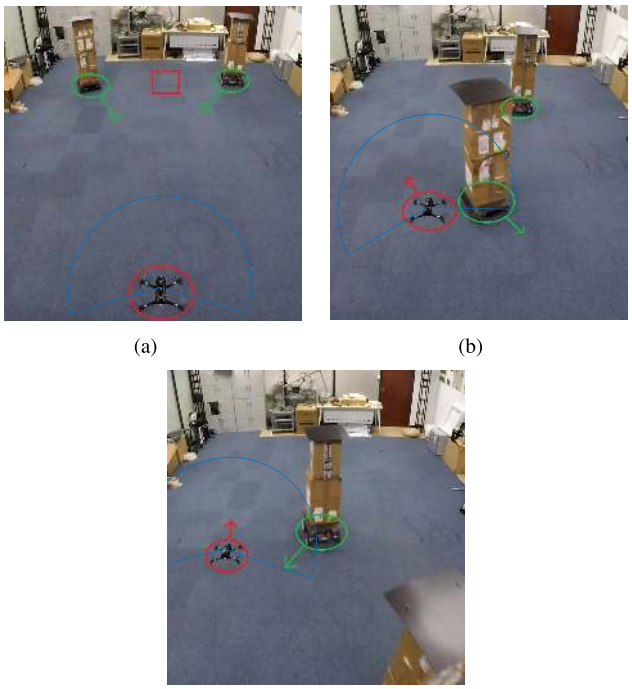


FIGURE 15. The three snapshots of the UAV experiment show the process of collision avoidance with multiple obstacles. In these snapshots, the red rectangular box is the target point of the robot. The red circle and green circle are denoted as robot and obstacle, respectively. The blue circular sector is the sensing range. Two obstacles rush to the robot in different directions at twice the speed of the robot.

the desired velocity when the distance between them becomes small, and the result is the robot cannot get a feasible solution at last. After considering the kinematic constraints when using MVO (Fig. 9), the path becomes smooth and the robot avoids collision successfully. Moreover, the limited field of view of the robot is not shown since the simulation experiment is mainly to demonstrate the necessity of considering kinematic constraints.

We conducted another experiment to show that the MVO algorithm with kinematic constraints(MVO-NH) can produce smooth actions. The experimental scenario is the same as that in Fig. 13, and an obstacle rushes to the non-holonomic robot from different random directions with a random speed in the interval [3, 6]m/s. In this scenario, We conducted 1000 experiments and the results are averaged. The variances of the robot's speed under the MVO algorithm and the MVO-NH algorithm are computed with respect to the preferred velocity $\mathbf{v}_A^{\text{pref}}$. The data in TABLE 1 show that the MVO-NH algorithm can generate smoother velocities for the robot.

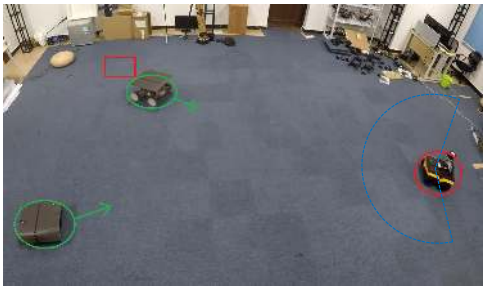
In Fig. 14, the trajectories are obtained by the contrast experiment of MVO and MVO method with limited sensing constraints. The common parameters of both robot and obstacle are radius $r = 0.6\,m$, maximum velocity $v_A^{\max} = 0.6\,m/s$. The parameters of robot are sensor range $r^s = 3\,m$, sensor's half-angle $\alpha^s = 1.919\,rad/s$, initial point $= [-5, -5]^T$, goal point $= [5, 5]^T$. The parameters of the obstacle are initial point $= [5, 5]^T$, goal point $= [-5.3, -5.3]^T$. We can observe that the robot navigated with only MVO loses the sight obstacle briefly when the robot is back on the obstacle. And the MVO method with limited sensing constraints avoids this situation happened. So it is necessary to consider the limited field of view of a robot especially.

B. EXPERIMENT SETUP

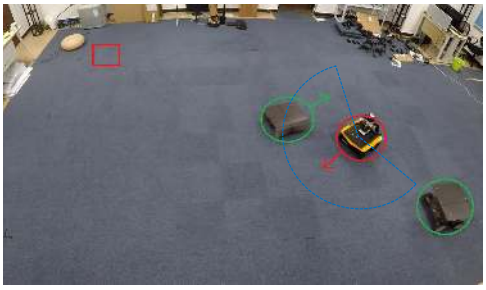
1) IMPLEMENT

Our method is evaluated on the holonomic and non-holonomic platforms with encountering a different number of obstacles in the serials scenario. Note that in every scenario, the obstacles rush to the robot while the robot moving to the target point. The robot calculates collision-free actions every period using the MVO method.

The holonomic robot platform is *Parrot Bebop 2* (Fig. 15), which is equipped with Raspberry Pi 3 with the Robot Operating System (ROS). Bebop is a kind of quad-rotor helicopter with opened interfaces to control three-axis linear speed. The maximum speed is 0.5 m/s in both forward and backward directions and its mass is about 0.5 kg. The lighter weight



(a)



(b)

FIGURE 16. The two snapshots of the non-holonomic experiment show the process of collision avoidance with multiple obstacles. In these snapshots, the red rectangular box is the target point of the robot. The red and green circles are denoted as the robot and obstacles. The blue circular sector is the sensing range.

brings more flexibility, so it has a faster speed response. In the experiments, the shape of Bebop is assumed to be a circle with radius 0.3 m. But the local sensing power is not strong enough to get accurate location information.

Jackal Unmanned Ground Vehicle is a differential-drive robot platform with on-board mini PC preloaded ROS packages (Fig. 16). On-board CPU is Intel's Core i7 6700. The maximum velocity is limited at 0.6 m/s in our experiments and the maximum rotate speed is 1 rad/s. We can use ROS interfaces to deliver the orders of velocity and rotate speed to the robot. The obstacles are the human and the remote-controlled holonomic robot refitted by *DJI ROBOMASTER 2016 INFANTRY*.

In the indoor experiments, for convenience, the robot and obstacles are tracked by the OptiTrack motion capture system. The position and orientation information is sent every 5 ms from the central computer through Wi-Fi. To simulate the limited sensing scenario, the robot is artificially set to avoid obstacles only within the sensor range, where the sensor distance rang r_A^s is 3m and the half detection angular range α_A^s is $\frac{3\pi}{5}$. And the velocity information of every robot and obstacle is inferred from the position and orientation data using Kalman filter.

In the outdoor experiments, Velodyne VLP16 LIDAR is applied as a detection module to gather the environment laser-point cloud (Fig. 20). Therefore, in the scenario which limits the robot's field of view, the maximum distance of LIDAR we restricted under 3m and the half detection angular range α_A^s is $\frac{3\pi}{5}$. After that, the obstacles' current positions are predicted by the Kalman equation through the Hungary algorithm.

2) PERFORMANCE METRICS

Here we propose the performance metrics of our experiments, to verify the feasibility of our MVO method.

- *Method feasibility*: The MVO method tends to reduce the cases of no solution when facing high-speed obstacles. If the unsolvable cases are too much, the robot will collide with an obstacle or overreact.
- *Collision-free movement*: The robot travels from the original point to the target point, while the actions are collision-free ones with encountering obstacles. And the robot would stop the movement when the relative distance is smaller than the safe distance.
- *Smooth action*: The action orders should be smooth and easy for a robot to follow. The non-holonomic robot has limited linear velocity and angular velocity, which are obvious observation features.

C. EXPERIMENT RESULTS

The experiments are conducted in following three basic scenarios which limit the robot's field of view:

- *Holonomic robot scenario*: We set two basic scenarios in this part. One is the robot encountering a single obstacle and another is multi-obstacle existing(Fig. 16). The robot's initial point and goal point are the same one or different ones, and Bebop needs to reach the goal point without collision. The speed of obstacles is almost twice as large as the Bebop's. And the maximum velocity of the robot is limited at 0.6 m/s in our experiments
- *Non-holonomic robot scenario*: The scenario is the same as the holonomic robot scenario. The obstacles rush to Jackal with a constant velocity. Jackal has to calculate its actions to navigate to the goals.

Fig. 17 shows the traces of the agents, which are recorded in different scenarios. Note that when the number of obstacles increases, the robot never overreacts and oscillates. The MVO with kinematic constraints combined with the feasible velocity region of the non-holonomic robot gives the desired velocity and rotate speed of the robot. It brings more smooth and safety to the non-holonomic robot. We simply consider that our formulations are capable of handling the distinction should it be made and satisfy the metrics. The snapshots in Fig. 16 shows the performance of a non-holonomic robot in the multi-obstacle scenario.

Fig. 18 shows the holonomic robot avoided collision with one obstacle. The velocity of the obstacle is almost 1.1 m/s and it moved straight to the robot. And the average speed of the robot is 0.5 m/s. Notice that here we added a little margin to the robot's radius. In the figure, we can observe that the robot generated evasive action when the obstacle is close to the robot at 5 s. And the snapshots in Fig. 15 present the collision-free navigation in the multi-obstacle scenario for the holonomic robot.

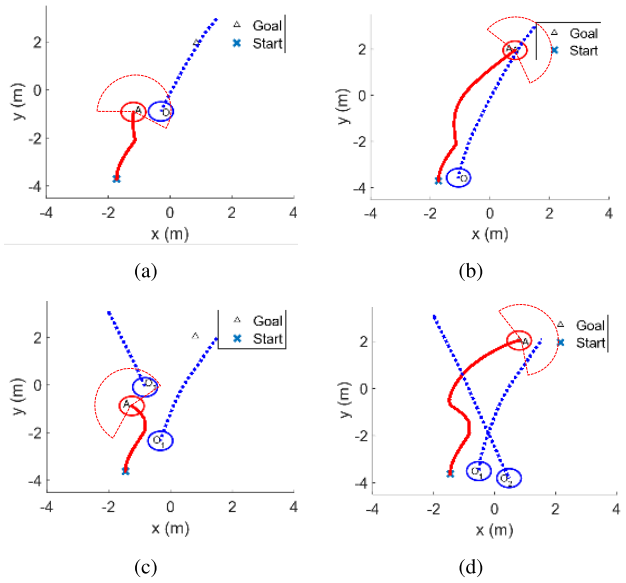


FIGURE 17. The path point data of non-holonomic robot experiments are plotted by using Matlab, which shows smooth navigation of the robot. The red one is the robot and the blue ones are obstacles. The red circular sector is the sensing range. (a) and (b), single obstacle. (c) and (d), multi-obstacle.

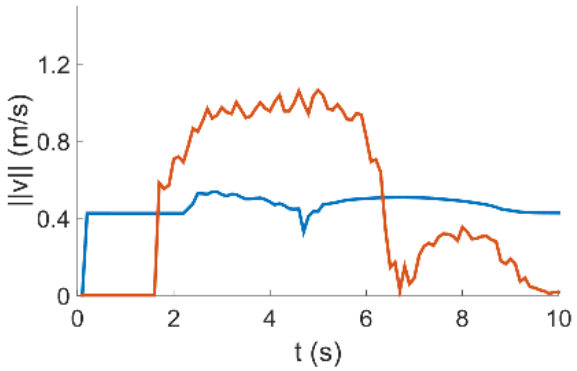


FIGURE 18. Bebop avoided collision with a single obstacle using MVO. The blue line is the velocity of the robot and the orange one is the velocity of the obstacle.

We also conduct a comparing experiment between the ORCA method and the MVO method. The scenario in Fig. 19 is the same as the scenario with the non-holonomic robot. The only difference is the method used for navigation. We can see that the robot has not enough awareness about the coming obstacle on account of the limited field of view, and it generated effective action when the obstacle was so closed to the robot. When the obstacle is close enough to the robot, ORCA generates the velocity order in the same direction with the obstacle. Because of the kinematic constraints, the robot cannot flee from the velocity obstacle. And put the experiment result with Fig. 17, we can get the conclusion that our MVO method is useful when there exist high-speed obstacles in the workspace.

In Fig. 20, Jackal's detection module is LIDAR and the field of view is limited in a specific range. In the experiment, we artificially limited the detection range of the sensor, to verify our algorithm. The robot moves to the target point and avoid collision with the human and moving obstacle. The human speed is around 2 m/s and the robot's maximum speed is 0.6 m/s. We use the MVO method concerned above with limited field of view.

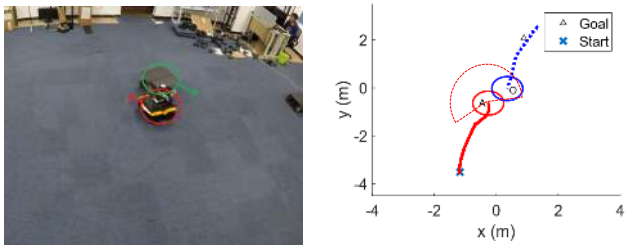


FIGURE 19. The non-holonomic robot experiment navigated by the ORCA method. Collision occurs in the process of navigation. The red circular sector is the sensing range. (a) The snapshot of the experiment shows the collision occurred. (b) The path of the experiment is plotted in Matlab.

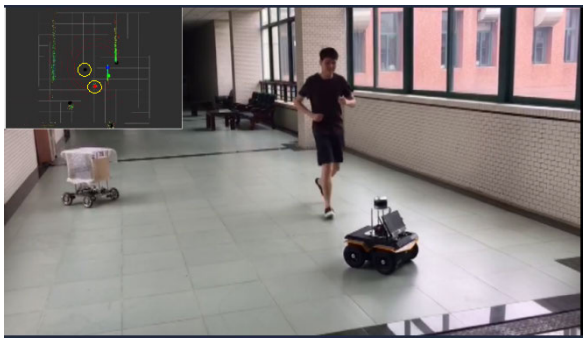


FIGURE 20. Non-holonomic robot avoided collision with multiple obstacles using MVO, which has a certain range of angles and a limited field of view. In the image on the upper left corner, the red point is the running human and the black one in the yellow cycle is another moving obstacle.

VII. CONCLUSION

In this article, a novel two-period velocity obstacle method called MVO is presented for a mobile robot with a limited field of view to avoid collision with high-speed obstacles in dynamic scenarios. An extended velocity obstacle is constructed to avoid collision with the obstacles whose velocities are larger than the maximum speed of the robot. The limited sensing field of view and kinematic constraints are incorporated into the MVO approach. Extensive experiments are conducted on the holonomic and non-holonomic robotic platforms to verify the efficacy of the proposed method. In future work, the dynamic constraints of the mobile robot will be considered. In addition, the trajectory prediction of dynamic obstacles will be studied to improve the performance of the MVO method. We will apply the MVO algorithm to more scenarios, such as different robot models and different sensor perception capabilities.

REFERENCES

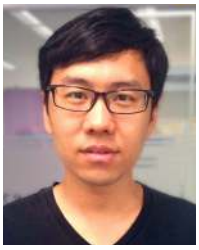
[1] S. Roelofsen, D. Gillet, and A. Martinoli, "Collision avoidance with limited field of view sensing: A velocity obstacle approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1922–1927.

[2] B. T. Lopez and J. P. How, "Aggressive collision avoidance with limited field-of-view sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1358–1365.
[3] P. Long, W. Liu, and J. Pan, "Deep-learned collision avoidance policy for distributed multiagent navigation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 656–663, Apr. 2017.
[4] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, and D. Manocha, "PRVO: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1089–1096.
[5] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica*, vol. 33, no. 3, pp. 463–497, Mar. 2015.
[6] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Smooth and collision-free navigation for multiple robots under differential-drive constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4584–4589.
[7] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*. Berlin, Germany: Springer, 2013, pp. 203–216.
[8] J. Alonso-Mora, P. Beardsley, and R. Siegwart, "Cooperative collision avoidance for nonholonomic robots," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 404–420, Apr. 2018.
[9] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Berlin, Germany: Springer, 2011, pp. 3–19.
[10] J. Snape, J. V. D. Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 696–706, Aug. 2011.
[11] E. G. Szadeczky-Kardoss and B. Kiss, "Velocity obstacles for Dubins-like mobile robots," in *Proc. 25th Medit. Conf. Control Autom. (MED)*, Jul. 2017, pp. 346–351.
[12] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Proc. ICRA. IEEE Int. Conf. Robot. Autom.*, vol. 4, May 2001, pp. 3716–3721.
[13] A. S. López, R. Zapata, and M. A. O. Lama, "Sampling-based motion planning: A survey," *Computación y Sistemas*, vol. 12, no. 1, pp. 5–24, 2008.
[14] R. Diankov and J. Kuffner, "Randomized statistical path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 1–6.
[15] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. ICRA. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, vol. 2, Apr. 2000, pp. 995–1001.
[16] E. Heiden, K. Hausman, G. S. Sukhatme, and A.-A. Agha-Mohammadi, "Planning high-speed safe trajectories in confidence-rich maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2880–2886.
[17] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
[18] A. Dutta and P. Dasgupta, "Bipartite graph matching-based coordination mechanism for multi-robot path planning under communication constraints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 857–862.
[19] C. J. Green and A. Kelly, "Toward optimal sampling in the space of paths," in *Proc. Int. Symp. Robot. Res.*, 2007, p. 1.
[20] S. Daftry, S. Zeng, A. Khan, D. Dey, N. Melik-Barkhudarov, J. A. Bagnell, and M. Hebert, "Robust monocular flight in cluttered outdoor environments," 2016, *arXiv:1604.04779*. [Online]. Available: http://arxiv.org/abs/1604.04779
[21] D. Dey, K. S. Shankar, S. Zeng, R. Mehta, M. T. Agcayazi, C. Eriksen, S. Daftry, M. Hebert, and J. A. Bagnell, "Vision and learning for deliberative monocular cluttered flight," in *Field and Service Robotics*. Cham, Switzerland: Springer, 2016, pp. 391–409.
[22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.
[23] H. Wang, L. Y. Zhao, and W. Chen, "A mobile robot obstacle avoidance method based on improved potential field method," in *Applied Mechanics and Materials*, vol. 467. Stäfa, Switzerland, Trans Tech, 2014, pp. 496–501.
[24] E. Falomir, S. Chaumette, and G. Guerrini, "A mobility model based on improved artificial potential fields for swarms of UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 8499–8504.
[25] W. B. Dunbar, M. B. Milam, R. Franz, and R. M. Murray, "Model predictive control of a thrust-vectored flight control experiment," *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 355–360, 2002.
[26] H. Cheng, Q. Zhu, Z. Liu, T. Xu, and L. Lin, "Decentralized navigation of multiple agents based on ORCA and model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3446–3451.
[27] G. Garimella, M. Sheckells, and M. Kobilarov, "Robust obstacle avoidance for aerial platforms using adaptive model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5876–5882.
[28] S. J. Guy, M. C. Lin, and D. Manocha, "Modeling collision avoidance behavior for virtual humans," in *Proc. 9th Int. Conf. Auto. Agents Multiagent Syst.*, vol. 2, 2010, pp. 575–582.
[29] L. He, J. Pan, W. Wang, and D. Manocha, "Proxemic group behaviors using reciprocal multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 292–297.
[30] P. Conroy, D. Bareiss, M. Beall, and J. van den Berg, "3-D reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," 2014, *arXiv:1411.3794*. [Online]. Available: http://arxiv.org/abs/1411.3794
[31] S. Roelofsen, D. Gillet, and A. Martinoli, "Reciprocal collision avoidance for quadrotors using on-board visual detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 4810–4817.
[32] A. Levy, C. Keitel, S. Engel, and J. McLurkin, "The extended velocity obstacle and applying ORCA in the real world," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 16–22.
[33] S. Samavati, M. Zarei, and M. T. Masouleh, "An optimal motion planning and obstacle avoidance algorithm based on the finite time velocity obstacle approach," in *Proc. Artif. Intell. Signal Process. Conf. (AISP)*, Oct. 2017, pp. 250–255.
[34] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "PORCA: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3418–3425, Oct. 2018.
[35] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, Jul. 1998.
[36] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1928–1935.
[37] H.-T.-L. Chiang, B. HomChaudhuri, A. P. Vinod, M. Oishi, and L. Tapia, "Dynamic risk tolerance: Motion planning by balancing short-term and long-term stochastic dynamic predictions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3762–3769.
[38] Z. Liu, Z. Jiang, T. Xu, H. Cheng, Z. Xie, and L. Lin, "Avoidance of high-speed obstacles based on velocity obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7624–7630.

TIANYE XU received the B.Eng. degree from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, in 2016. He is currently pursuing the master's degree with the School of Data and Computer Science, Sun Yat-sen University. His research interests include robotic motion planning and robotic swarm intelligence.

SHUIQING ZHANG received the B.Eng. degree in automation from Sun Yat-sen University, Guangzhou, China, in 2018. She is currently a Graduate Student with the School of Data and Computer Science, Sun Yat-sen University. Her research interests include robotic motion planning and robotic swarm intelligence.

ZEYU JIANG received the master's degree from the School of Data and Computer Science, Sun Yat-sen University, Guangzhou. His research interests include robotic motion planning and multi-robot systems.



ZHONGCHANG LIU (Member, IEEE) received the B.S. and M.S. degrees in control theory and control engineering from Dalian Maritime University, Dalian, China, in 2009 and 2011, respectively, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2015. He was a Visiting Fellow with Boston University, USA, in Summer 2013. From 2015 to 2016, he was a Postdoctoral Fellow with CUHK. From 2016 to 2017, he was a Research Associate with Sun Yat-sen University. He is currently an Associate Professor with the College of Marine Electrical Engineering, Dalian Maritime University. His research interests include cooperative control of distributed systems, networked control, and synchronization problems in multiagent systems.

HUI CHENG (Member, IEEE) received the B.Eng. degree in electrical engineering from Yanshan University, Qinhuangdao, China, in 1998, the M.Phil. degree in electrical and electronic engineering from The Hong Kong University of Science and Technology, in 2001, and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, in 2006. She worked as a Postdoctoral Fellow at the Department of Information Engineering, The Chinese University of Hong Kong, from August 2006 to December 2007. She is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou. Her research interests include intelligent robots and networked control.