Receding Horizon Motion Planning for Multi-Agent Systems: A Velocity Obstacle Based Probabilistic Method

Xiaoxue Zhang, Jun Ma, Zilong Cheng, Sunan Huang, and Tong Heng Lee

Abstract—In this paper, a novel and innovative methodology for feasible motion planning in the multi-agent system is developed. On the basis of velocity obstacles characteristics, the chance constraints are formulated in the receding horizon control (RHC) problem, and geometric information of collision cones is used to generate the feasible regions of velocities for the host agent. By this approach, the motion planning is conducted at the velocity level instead of the position level. Thus, it guarantees a safer collision-free trajectory for the multi-agent system, especially for the systems with high-speed moving agents. Moreover, a probability threshold of potential collisions can be satisfied during the motion planning process. In order to validate the effectiveness of the methodology, different scenarios for multiple agents are investigated, and the simulation results clearly show that the proposed approach can effectively avoid potential collisions with a collision probability less than a specific threshold.

Index Terms—Receding horizon control, motion planning, multi-agent systems, chance constraints, velocity obstacle.

I. INTRODUCTION

Motion planning is one of the essential components of intelligent robots, and an efficient trajectory can effectively improve robots’ intelligence and autonomy [1], [2]. The velocity obstacles approach is a geometry-based method that defines conic regions as constraints on the feasible velocities for agents. The concepts of collision cones and velocity obstacles approach are introduced in [3]. This kind of method can be used to find the range of feasible velocity quickly with minimal obstacle information without any prior knowledge or prediction, after given the collision scenario geometrically [4]. There are several variants of the velocity obstacles approach, i.e., reciprocal velocity obstacles [5], accelerated velocity obstacles [6], generalized velocity obstacles [7], hybrid reciprocal velocity obstacles [8], etc. However, most of these variants consider the feasible velocity direction with neglecting the optimality of the planned trajectory.

In order to find a feasible velocity based on the collision region provided by velocity obstacles, receding horizon control (RHC), which is also known as model predictive control (MPC), can be utilized to find the optimal velocity and position [9]. It is one typical effective approach that has been widely applied in the decision making, planning, and control of robots [10]–[12]. MPC has the capability of addressing various constraints as part of the control synthesis problem [13]–[16]. In terms of the uncertainties of path planning, one can characterize these uncertainties in a probabilistic manner and find the optimal sequence of control inputs subject to the chance constraints, which means that the probability of collision avoidance should not be lower than a user-defined threshold [17]–[19]. Such a probabilistic approach has multiple advantages compared to the traditional box constraints approach, as most of the uncertainties can be represented by a stochastic model instead of a bounded set [20]. Besides, the probability threshold will influence the conservatism of the planned trajectory. A proper probability threshold can balance conservatism and performance. There are a number of previous works regarding the chance constrained path planning with the existence of obstacles [21]–[24]. However, most of the existing probabilistic approaches still stay on the position-based level to achieve motion planning, which means the position information is mainly utilized to make a decision. Thus, these approaches at the position-based level could not be suitable even valid in some scenes, especially in fast-moving scenarios.

This paper presents a novel approach to plan the optimal trajectories for the multi-agent system through the confinement of collision probability, based on the information of velocity obstacles. This approach generates a feasible region of velocity for the host agent when computing an optimal trajectory. Then the feasible region is further formulated as probabilistic collision constraints. This method is capable of planning the trajectories at the velocity level, which is much more meaningful than the traditional counterpart at the position level, especially effective for the scenarios with high-speed obstacles. Also, a collision probability can be confined within a specific range in the proposed approach. The structure of this paper is as follows. In Section II, the basic concept and definitions of the velocity obstacles method are introduced. Section III formulates the chance constrained receding horizon optimization problem based on the velocity obstacles. In Section IV, we transform the probabilistic constraints in the model predictive control (MPC) problem into deterministic constraints and then solve this problem by multiple shooting method. Section V shows the simulation results in different
scenarios, and conclusion is presented in Section VII.

II. VELOCITY OBSTACLES

In this section, the basic concept of the velocity obstacles approach is introduced, and the geometric illustration is shown in Fig. [I]. The related parameters and variables in this figure are explained below.

![Velocity obstacles](image)

Fig. 1. Velocity obstacles

Let all circular moving objects \( O_i \) and \( O_j \) be centered at \( p_i \in \mathbb{R}^n \) and \( p_j \in \mathbb{R}^n \) with radius \( r_i \) and \( r_j \) and velocities \( v_i \in \mathbb{R}^n \) and \( v_j \in \mathbb{R}^n \), respectively, where

\[
O_i = \{ p_i + \mu r_i \mid \| \mu \|_2 \leq 1 \} \\
O_j = \{ p_j + \mu r_j \mid \| \mu \|_2 \leq 1 \}.
\]

(1)

Let \( \oplus \) denotes the Minkowski sum operation, and we have

\[
O_i \oplus O_j = \{ m_i + m_j \mid m_i \in O_i, m_j \in O_j \}.
\]

(2)

The relative velocity between \( i \)th object and \( j \)th object is \( v_{ij} := v_i - v_j \). Let \( \lambda(v_{ij}) \) denote the ray with direction \( v_{ij} \) from the position \( p_i \), where

\[
\lambda(v_{ij}) = \{ p_i + \lambda v_{ij} \mid \lambda \geq 0 \}.
\]

(3)

Then the object \( i \) and the object \( j \) will collide if and only if

\[
\lambda(v_{ij}) \cap (O_i \oplus O_j) \neq \emptyset.
\]

(4)

Therefore, a collision cone \( C_{ij} \) can be represented by

\[
C_{ij} = \{ v_{ij} \mid (O_i \oplus O_j) \cap \lambda(v_{ij}) \neq \emptyset \}.
\]

(5)

In order to determine whether the velocity \( v_i \) of the \( i \)th object has a risk of collision with the \( j \)th object, the velocity obstacle \( \mathbb{V}_{ij} \) is defined as

\[
\mathbb{V}_{ij} = \{ v_i \mid (v_i - v_j) \in C_{ij} \},
\]

(6)

which is equivalent to

\[
\mathbb{V}_{ij} = v_j + C_{ij}.
\]

(7)

for the \( i \)th object. Any velocity \( v_i \in \mathbb{V}_{ij} \) will result in a collision, as shown in Fig. [I].

The set of all moving surrounding objects can be considered as obstacles for the host object \( i \). Assume the set of all moving objects is \( \mathbb{N}_n = \{1, 2, \cdots, n\} \). The composite velocity obstacles and collision cones are

\[
\mathbb{V}_i = \bigcup_{j \neq i} \mathbb{V}_{ij} \\
C_i = \bigcup_{j \neq i} C_{ij}.
\]

(8)

III. PROBLEM STATEMENT

A. Objects Model and Collision Chance Constraints

1) Objects Model: The dynamics of each planar object \( i \in \mathbb{N}_n \) can be represented by any stochastic nonlinear or linear, continuous-time or discrete-time model, where

\[
\dot{x}_i = f_i(x_i, u_i) + \omega_i \quad \text{or} \quad x_i^{k+1} = g_i(x_i^k, u_i^k) + \omega_i^k,
\]

(9)

where \( x_i = [p_i^T, v_i^T]^T \in \mathcal{X}_i \subset \mathbb{R}^{n_i} \) denotes the state vector consisting of positions and velocities, \( u_i \in \mathcal{U}_i \subset \mathbb{R}^{n_u} \) is the input control vector, \( n_c \) and \( n_u \) represent the dimension of the state vector and the control input vector, respectively, and \( \mathcal{X}_i \) and \( \mathcal{U}_i \) are the state and control space of the \( i \)th object, respectively. In the discrete-time model, \( k \) means the \( k \)th time step for the objects. \( f_i \) and \( g_i \) are the nonlinear continuous-time and discrete-time dynamics models of the \( i \)th object.

We consider the Gaussian process noise of velocity in the objects model, i.e., \( \omega_i \sim \mathcal{N}(0, \Sigma_i) \) with a diagonal covariance matrix \( \Sigma_i \). In the following, we use the discrete-time dynamics model to illustrate our approach.

2) Collision Chance Constraints: According to the previous introduction of velocity obstacles, we can obtain the collision condition of the object \( i \) with respect to the object \( j \), which is defined as

\[
v_i^k \notin \mathbb{V}_i \quad \text{or} \quad v_{ij}^k \notin C_i.
\]

(10)

Since there is additional noise on the velocity of objects, the velocity of each object can be described as random variables \( \dot{x}_i^k := \dot{x}_i^k + \omega_i^k \sim \mathcal{N}(\dot{x}_i^k, \Sigma_{i,k}) \), where \( \dot{x}_i^k \) is the mean of the velocity and \( \omega_i^k \) is the additional noise at time \( k \). Hence, the collision avoidance constraints can be described in a probabilistic manner, where for each object \( i \), the chance constraints can be expressed as

\[
\Pr \left( v_i^k \notin \mathbb{V}_i \right) \geq 1 - \delta_i, \quad \forall i \in \mathbb{N}_n
\]

(11a)

or

\[
\Pr \left( v_{ij}^k \notin C_i \right) \geq 1 - \delta_i, \quad \forall i \in \mathbb{N}_n,
\]

(11b)

where \( \delta_i \) is the probability threshold of the collision risk.

B. Distributed Collision Avoidance Problem

Here, we formulate a distributed collision avoidance problem. For each object \( i \in \mathbb{N}_n \), we formulate a discrete-time chance constrained optimization problem on \( N \) prediction steps with a sampling time \( \Delta t \).

Problem 1: (Optimization with Probabilistic Chance Constraints) For each host object \( i \), the position of the other objects \( p_j, \forall j \in \mathbb{N}_n, j \neq i \), the initial state \( x_i^0 \) with uncertain noise, the reference state vector \( \text{vec}_{ref,i} \), and the collision probability threshold \( \delta_i \) have been provided. The objective is to compute
the optimal trajectories and control inputs for all of the objects. Thus, these objects can move from their initial states to the target states while maintaining the collision probability below the given threshold. The problem is defined as

\[
\min_{x_{i}^{k},u_{i}^{(k+1)}} \sum_{k=0}^{N-1} \| x_{i}^{k} - x_{ref,i}^{k} \|_{Q_{i}} + \| u_{i}^{k} \|_{R_{i}}
\]

subject to

\[
x_{i}^{k+1} = g_{i}(x_{i}^{k}, u_{i}^{k}) + \omega_{i}^{k},
\]

\[
u_{i}^{k+1} \sim \mathcal{N}(\mu_{i}^{k+1}, \Sigma_{i}^{k+1}),
\]

\[
Pr\left(\nu_{i}^{k+1} \not\in \mathcal{V}_{i}^{k+1}\right) \geq 1 - \delta, \quad \forall i \in \mathbb{N}_{n}
\]

\[
x_{i}^{k+1} = \sum_{j=1}^{N} \mathcal{N}^{\leq} + \mathcal{N}^{\geq},
\]

\[
u_{i}^{k+1} \in \mathcal{V}_{i}, \quad u_{i}^{k+1} \in \mathcal{U}_{i},
\]

where \(\| x_{i}^{k} - x_{ref,i}^{k} \|_{Q_{i}} = \langle (x_{i}^{k} - x_{ref,i}^{k}) , Q_{i}(x_{i}^{k} - x_{ref,i}^{k}) \rangle , Q_{i} \) and \(R_{i} \) are weighting matrices to penalize the deviation from the reference states and the unnecessary large control inputs, respectively. \(\mathcal{V}_{i}, \mathcal{U}_{i}, \mathcal{N}_{i}, \mathcal{R}_{i} \) are the lower bound and upper bound of state variables and control inputs.

Remark 1: In the problem formulation (12), the collision avoidance constraints are based on the \((k+1)\)th time step. But in the next mathematical transformation, our discussion is based on the \(k\)th time step for simplicity.

IV. RECEDING HORIZON OPTIMIZATION PROBLEM WITH CHANGE CONSTRAINTS

The chance constraints of velocity obstacles \(Pr(\nu_{i}^{k} \not\in \mathcal{V}_{i}^{k}) \geq 1 - \delta, \forall i \in \mathbb{N}_{n} \) mean that the risk of collision should be less than \(\delta\); however, it is hard to determine it. Therefore, we need to do some mathematical manipulations to make it much easier to solve. The composite collision cone \(\mathcal{C}_{i}^{k}\) and velocity obstacles \(\mathcal{V}_{i}^{k}\) are the union of \(\mathcal{C}_{i}^{k}\) and \(\mathcal{V}_{i}^{k}\) with respect to object \(j\) with \(j \neq i\). Thus, we start to analyze it in terms of one collision cone \(\mathcal{C}_{i}^{k}\) or one velocity obstacle \(\mathcal{V}_{i}^{k}\). All of the pertinent variables and their relationships are demonstrated in Fig. 2.

Define the vector \(p_{ij}^{k} := p_{j}^{k} - p_{i}^{k} \in \mathbb{R}^{n_{p}}\), where \(n_{p}\) is the dimension of the position vector, and the radius of the circle is given by \(r = r_{i} + r_{j}\). Then, the angle between \(p_{ij}^{k}\) and the boundary line of the collision cone is the angle \(\alpha_{ij}^{k}\), as shown in Fig. 2. Obviously, \(\sin \alpha_{ij}^{k} = \frac{r}{|p_{ij}^{k}|}\), and \(\cos \alpha_{ij}^{k} = \sqrt{1 - \sin^{2} \alpha_{ij}^{k}} = \frac{\sqrt{|p_{ij}^{k}|^{2} - r^{2}}}{|p_{ij}^{k}|}\). Based on the rotation relationship of the vector \(p_{ij}^{k}\), we can obtain the two tangent directional vectors \(T_{ij,1}^{k}\) and \(T_{ij,2}^{k}\), where

\[
T_{ij,1}^{k} = \begin{bmatrix} \cos \alpha_{ij}^{k} & \sin \alpha_{ij}^{k} \\ -\sin \alpha_{ij}^{k} & \cos \alpha_{ij}^{k} \end{bmatrix} p_{ij}^{k}
\]

\[
T_{ij,2}^{k} = \begin{bmatrix} \cos \alpha_{ij}^{k} & -\sin \alpha_{ij}^{k} \\ \sin \alpha_{ij}^{k} & \cos \alpha_{ij}^{k} \end{bmatrix} p_{ij}^{k}.
\]

A. Analysis on Velocity Obstacles

Based on the velocity obstacle \(\mathcal{V}_{ij}^{k}\), if the collision avoidance condition \(\mathcal{V}_{ij}^{k} \cap \mathcal{V}_{ij}^{k} = \emptyset\) is satisfied, the feasible region of \(\nu_{i}^{k}\) should be the orange region shown in Fig. 2. According to the direction of \(\nu_{i}^{k}\) shown as the blue arrow, obviously, different directions of \(\nu_{i}^{k}\) can result in different feasible regions of \(\nu_{i}^{k}\). Therefore, we need to discuss it in terms of different cases, as shown in Fig. 3.

- Case 1: \(a > 0, b > 0\)

Here, \(\nu_{i}^{k}\) is inside the collision cone \(\mathcal{C}_{i}^{k}\), which is shown as the red arrow in Fig. 3. The feasible region of \(\nu_{i}^{k}\) is shown as the red region, which means

\[
\nu_{i}^{k} = cT_{ij,1}^{k} + dT_{ij,2}^{k}, \quad -(c > 0, d > 0),
\]

where \(-\) represents the logical negation operation.

- Case 2: \(a \leq 0, b \leq 0\)

In this case, \(\nu_{i}^{k}\) is inside the inverted cone of \(\mathcal{C}_{i}^{k}\), which is shown as the yellow arrow in Fig. 3. The feasible region of \(\nu_{i}^{k}\) is in the yellow region, which means

\[
\nu_{i}^{k} = cT_{ij,1}^{k} + dT_{ij,2}^{k}, \quad (c > 0, d > 0).
\]
• Case 3: \(a < 0, b > 0\)
In this case, \( v_i^k \) is shown as the green arrow in Fig. 3 so we have
\[
v_i^k = cT_{ij,1}^k + ev_j^k, \quad \theta(e > 0, e > 0).
\]
\[(16)\]
• Case 4: \(a > 0, b < 0\)
Here, \( v_i^k \) is shown as the blue arrow in Fig. 3
\[
v_i^k = dT_{ij,2}^k + ev_j^k, \quad \theta(d > 0, e > 0).
\]
\[(17)\]

To summarize all the scenarios as discussed above, let \( v_i^k = aT_{ij,1}^k + bT_{ij,2}^k \), where \(a, b \in \mathbb{R} \), and assume \( v_i^k = cT_{ij,1}^k + dT_{ij,2}^k + ev_j^k \), then we have the following condition:
\[
\begin{cases}
e = 0, \theta(c > 0, d > 0), & \text{if } a > 0, b > 0 \\
e = 0, \theta(c > 0, d > 0), & \text{if } a < 0, b < 0 \\
d = 0, \theta(c > 0, e > 0), & \text{if } a < 0, b > 0 \\
c = 0, \theta(d > 0, e > 0), & \text{if } a > 0, b < 0.
\end{cases}
\]
\[(18)\]
This condition is a case-by-case discussion about \( v_i^k \) according to the direction of \( v_i^k \) and there are logical negation operations in the condition, whereby it is sectionally discontinuous and disjunctive, which is hard to compute.

B. Analysis on Collision Cones
In this part, we focus on the relationship between the relative velocity \( v_{ij}^k \) and the collision cone \( C_{ij}^k \) as shown in Fig. 4. In order to meet the collision-free requirement, i.e., there is no intersection between the relative velocity \( v_{ij}^k \) and the collision cone \( C_{ij}^k \), \( v_{ij}^k \) should point outside the collision cone, which means that the angle between the two outer normal vectors \( N_{ij,1}^k, N_{ij,2}^k \), and \( v_{ij}^k \) should be within the range \((-\frac{\pi}{2}, \frac{\pi}{2})\), as shown in Fig. 4.

Fig. 4. Collision avoidance conditions analysis for \( v_{ij}^k \)

Two outer normal vectors \( N_{ij,1}^k, N_{ij,2}^k \) are given by
\[
N_{ij,1}^k = \begin{bmatrix} \cos \frac{k}{2} & \sin \frac{k}{2} \\ -\sin \frac{k}{2} & \cos \frac{k}{2} \end{bmatrix} T_{ij,1}^k
\]
\[
N_{ij,2}^k = \begin{bmatrix} \cos \frac{k}{2} & -\sin \frac{k}{2} \\ \sin \frac{k}{2} & \cos \frac{k}{2} \end{bmatrix} T_{ij,2}^k
\]
\[(19)\]
The probabilistic condition of collision avoidance (11b) can be transformed as
\[
\left( v_{ij}^k \cdot N_{ij,1}^k > 0 \right) \cup \left( v_{ij}^k \cdot N_{ij,2}^k > 0 \right), \quad \forall j \in N_i, j \neq i.
\]
\[(20)\]
In addition, chance constraints can be rewritten as
\[
\Pr \left( \left( N_{ij,1}^k \cdot v_{ij}^k > 0 \right) \cup \left( N_{ij,2}^k \cdot v_{ij}^k > 0 \right) \right) \geq 1 - \delta_{ij}
\]
\[
\iff \Pr \left( \left( N_{ij,1}^k \cdot v_{ij}^k \leq 0 \right) \cup \left( N_{ij,2}^k \cdot v_{ij}^k \leq 0 \right) \right) \leq \delta_{ij}
\]
\[
\iff \Pr \left( N_{ij,1}^k \cdot v_{ij}^k \leq 0 \right) \leq \delta_{ij,1}
\]
\[
\Pr \left( N_{ij,2}^k \cdot v_{ij}^k \leq 0 \right) \leq \delta_{ij,2}
\]
\[(21)\]

Due to the existence of noise, \( v_{ij}^k \) is subject to a normal distribution, then we have
\[
\delta_{ij,1}\delta_{ij,2} = \delta_{ij}.
\]
\[(22)\]

In order to compute probabilistic chance constraints in a deterministic manner, we introduce the following lemma.

Lemma 1: Given any matrix \( A \) and scalar \( b \), for a multivariate random variable \( X(t) \) corresponding to the mean \( \mu(t) \) and covariance \( \Sigma(t) \), the chance constraint
\[
\Pr (A^T X(t) < b) \leq \varphi,
\]
\[(23)\]
is equivalent to a deterministic linear constraint
\[
A^T \mu(t) - b \geq \eta,
\]
\[(24)\]
where \( \eta = \sqrt{2A^T \Sigma(t)} \eta \) er\( f^{-1}(1 - 2\varphi) \), er\( f \) is the error function
\( \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) \, dt \), and \( \varphi \) is the predefined allowable probability threshold of collision.

Notice that the variable \( v_{ij}^k \) is a multivariate Gaussian random variable \( v_{ij}^k \sim N(\nu_{ij}^k, W_{ij}) \), we can obtain the equivalent constraint:
\[
\Pr \left( (N_{ij,1}^k \cdot v_{ij}^k \leq N_{ij,1}^k \cdot \nu_{ij}^k) \right) \leq \delta_{ij,1}
\]
\[
\iff \quad N_{ij,1}^k \cdot v_{ij}^k - N_{ij,1}^k \cdot \nu_{ij}^k \geq k_{ij,1}
\]
\[
\Pr \left( (N_{ij,2}^k \cdot v_{ij}^k \leq N_{ij,2}^k \cdot \nu_{ij}^k) \right) \leq \delta_{ij,2}
\]
\[
\iff \quad N_{ij,2}^k \cdot v_{ij}^k - N_{ij,2}^k \cdot \nu_{ij}^k \geq k_{ij,2},
\]
\[(25)\]
where \( k_{ij,1} = \sqrt{2N_{ij,1}^k W_{ij} N_{ij,1}^k} \cdot \text{erf}^{-1}(1 - 2\delta_{ij,1}) \) and \( k_{ij,2} = \sqrt{2N_{ij,2}^k W_{ij} N_{ij,2}^k} \cdot \text{erf}^{-1}(1 - 2\delta_{ij,2}) \).

The calculations of \( k_{ij,1} \) and \( k_{ij,2} \) require the knowledge on \( W_i \), i.e., the covariance of the \( i \)-th robot at the current time. Moreover, the Kalman filter can be used to estimate the other objects velocity in the prediction horizon.

C. Deterministic MPC Formulation and Solving

Problem 2: (MPC Problem with Deterministic Chance Constraints) The probabilistic chance constraints in (12) can be
Fig. 5. Simulation results on multiple objects in different scenarios
transformed into a deterministic manner, which is tractable. This problem can be formulated as

$$\min_{u_1^{N-1}, x_1^{N-1}} \sum_{k=0}^{N-1} \left( \left\| x_1^k - x^{\text{ref}, i}_1 \right\|_{Q_i} + \left\| u_1^k \right\|_{R_i} \right)$$

subject to

$$x_1^{k+1} = f_1 \left( x_1^k, u_1^k \right), \quad x_1^0 = p$$

$$N_{i,j,1}^{(k+1)T} v_j^{k+1} - N_{i,j,1}^{(k+1)T} v_j^{k+1} \geq \kappa_{i,j,1}, \forall j \in \mathbb{N}_n, j \neq i$$

$$N_{i,j,2}^{(k+1)T} v_j^{k+1} - N_{i,j,2}^{(k+1)T} v_j^{k+1} \geq \kappa_{i,j,2}, \forall j \in \mathbb{N}_n, j \neq i$$

$$1 \leq k^{i,j} + 1 \leq x_i^{k+1}$$

$$\sum_{k=0}^{N-1} u_1^k \leq u_1^0 \leq \bar{u}_1$$

$$x_1^{k+1} \sim \mathcal{N} \left( u_1^{k+1}, \Sigma_i \right)$$

$$x_1^{k+1} \in \mathcal{X}_i, \quad u_1^k \in \mathcal{U}_i.$$  \hfill (26)

The collision probability between the host object $i$ and all the other objects $j \in \mathbb{N}_n, j \neq i$ is less than a probability threshold $\delta_i$, where

$$\delta_i = 1 - \prod_{j \in \mathbb{N}_n, j \neq i} \left(1 - \delta_{i,j,1} \delta_{i,j,2} \right).$$ \hfill (27)

At time $t$, the cost function is optimized under the constraints in (26) to obtain the optimal control sequence.

$$u_1^* = \left[ \left( u_1^0 \right)^T \left( u_1^1 \right)^T \ldots \left( u_1^{N-1} \right)^T \right]^T,$$ \hfill (28)

and only the first control input $u_1^0$ will be executed. The pseudocode of how to address the velocity obstacle based receding horizon motion planning with chance constraints is shown in Algorithm 1.

Algorithm 1 Velocity Obstacle Based Receding Horizon Motion Planning with Chance Constraints.

**Initialization:** Dynamic model for all agents; the agents number $n_{num}$; the radius of agents $r_i$; weighting matrices $Q_i$ and $R_i$ in the cost function; initial position and target position of all agents; reference positions of all agents; upper and lower bound of the state $x$ and control input $u$, i.e., $\mathcal{X}$ and $\mathcal{U}$, respectively; runtime of MPC $r_{run}$; sampling time $\tau$; covariance matrix $W_i$; threshold $\delta_1$ and $\delta_2$.

Set the time step $t = 0$.

**while** $t <= r_{run}$ **do**

for $i \in \mathbb{N}_n$ **do**

for $k = 0, 1, \ldots, N-1$ **do**

Compute $x_1^{k+1}$ for the $i$th agent based on $x_1^k$ by using (9).

Generate constraints of physical limitations for the $i$th agent.

Compute the two transformed deterministic constraints for the $i$th agent by (25).

**end for**

Compute the probabilistic cost function for the $i$th agent.

Solve the deterministic optimization problem (26).

Obtain the optimal control sequence $u_1^i$.

Execute the first control input $u_1^0$ for the $i$th agent.

Pass the updated states of neighbors $j \in \mathbb{N}_n, j \neq i$ to the $i$th agent.

**end for**

$t = t + \tau$.

**end while**

V. RESULTS

This section describes the implementation of the proposed method, and the effectiveness of the method is evaluated by simulations. All the relevant parameters of the simulation are shown in Table I in the Appendix. Here, we add the Gaussian noise to the velocity of objects model, and the added measurement noise is zero mean with the covariance $W_i$. Taking the noisy measurements as inputs, a Kalman filter is designed for validation purposes, and the simulation results are implemented in Python 3.7 environment on a PC with Intel i5 CPU@3.30 GHz. The video demonstrating the results can be found at https://www.youtube.com/watch?v=Mw/A6eUblAiw4 and the source code is available at https://github.com/Lisno1/Velocity-Obstacle-Motion-Planning.

Here, four different scenarios for $6$ and $12$ objects are designed for validation purposes, and the simulation results are shown in Fig. 5. Different colors represent different objects with radius $r_i$. In the simulation, the radius of all objects is set as $r_i = 0.2$ m. The circles in (b), (d), (f), and (h) of Fig. 5 demonstrate the approximate shape of objects. The solid lines and dotted lines in (a), (c), (e), and (g) of Fig. 5 represent the planned trajectories and reference trajectories for each object, respectively. The scenario 1 in Fig. 5 means that there are $6$ objects whose target position is the symmetry point of the initial position along the $x$ or $y$ axis. For example, for the green object, its initial position is $(-2, 2)$ m and then its target position is $(-2, -2)$ m. Fig. 5(a) shows the trajectories of the $6$ objects and Fig. 5(b) presents the position of all objects in each time step. According to Fig. 5(a) and (b), there is no collision happening in this scenario, as no two objects occur in the same position at the same time. The scenario 2 in Fig. 5 indicates that the $6$ objects starting from their initial positions need to pass through the origin point $(0, 0)$ m without any collision and reach their target positions that are the symmetry points of the initial positions along with the origin, i.e., along the both $x$ and $y$ axis. To be specific, the green object in scenario 2 starts from $(-2, 2)$ m and needs to reach its target position $(2, -2)$ m. The results of the scenario 2 are shown in Fig. 5(c) and (d). Also, all objects are able to arrive at their destinations with no collision, based on the two subfigures. Similarly, the task in scenario 3 is to reach the target positions that are symmetry points along the $x$ or $y$ axis of the initial positions for all $12$ objects. Fig. 5(e) and (f) illustrate the resulted trajectories of these $12$ objects, and it is straightforward to observe that all of the $12$ objects can reach their target positions without collisions. In scenario 4, there are $12$ objects whose target positions are the symmetry points of the initial positions along with the origin, i.e., along the both $x$ and $y$ axis. The trajectories of these $12$ objects are shown in Fig. 5(g) and (h), and we can observe that all of the trajectories are collision-free, which satisfies our requirement. Overall, all the subfigures from Fig. 5(a) to (h) indicate that this approach can effectively avoid collisions with the other...
moving objects (including agents and obstacles, which can be represented by different dynamics and shapes).

The distances between each pair of objects in all scenarios are shown in Fig. 6. The subfigures from the first one to the fourth one are the results from scenario 1 to scenario 4, respectively. According to all these subfigures in Fig. 6, it is apparent that each object can maintain a certain safety distance from other objects, as the minimum safety distance is greater than 0.1 m, which is the radius of each object. Therefore, there is no collision happening in all of the four scenarios.

![Image](https://via.placeholder.com/150)

Fig. 6. Distance between each pair of objects in all scenarios

For scenario 4, we compare the results of our method under different three levels of measurement noise ($\frac{1}{2}W_i$, $W_i$, and $4W_i$) with the deterministic MPC method, which is proposed in [25]. The minimum distance between each pair of agents and the success rate are treated as the safety metrics. The comparison results are shown in Table I. Besides, due to a tighter bound of collision probability approximation, our method can keep a larger minimum distance during running under the same noise level, compared with the deterministic MPC method. Also, with a larger noise level, our proposed method maintains the success rate of 100%, but the success rate of deterministic MPC decreases from 71% to 41%. According to Table I we can observe that our proposed method achieves higher safety performance compared with the deterministic MPC method.

![Image](https://via.placeholder.com/150)

Fig. 7. Distance between each pair of objects in all scenarios

**VI. Conclusion**

In this paper, a velocity obstacle based receding horizon motion planning method, is proposed for potential collision avoidance. Based on the feasible region of $v_i$ provided by the velocity obstacles method, a chance constrained RHC problem is formulated and solved. A feasible region of velocity for the host agent is derived and formulated as probabilistic collision constraints. Hence the proposed method can generate the trajectories at the velocity level. Besides, this method can also provide a probability threshold of potential collision during the motion planning process. Several simulation scenarios for multiple agents are employed to validate the effectiveness and efficiency of our proposed methodology. In terms of the future work, one prospective research is to realize the motion planning in 3-dimensional space. Moreover, another future work is to consider the uncertainty in shape and velocity of the agents at the same time.

**APPENDIX**

Table II shows the values of parameters in Section IV.
### TABLE II

| Meaning                  | Notation | Value | Unit |
|--------------------------|----------|-------|------|
| Radius                   | $r_i$    | 0.1   | m    |
| Mass                     | $m$      | 1     | kg   |
| Covariance of noise      | $W_i$    |       |      |
|                         | $\text{diag}(0.01, 0.01, 0.05, 0.05)$ |      |      |
| Threshold                | $\delta_{i1}, \delta_{i2}$ | 0.1, 0.1 |      |
| Sampling time            | $\Delta t$ | 0.05  | s    |
| Prediction horizon       | $N$      | 25    | -    |

#### State weighting matrix

| $Q_i$ | $\text{diag}(10, 10, 1, 1)$ | - |

#### Control input weighting matrix

| $R_i$ | $\text{diag}(1, 1)$ | - |

#### Maximum/minimum state limitations

| $\pi_i/\gamma_i$ | $[\infty \infty 10 10]^T / [-\infty -\infty -10 -10]^T$ | - |

#### Maximum/minimum input limitations

| $\pi_i/\gamma_i$ | $[\infty \infty]^T / [-\infty -\infty]^T$ | - |

### REFERENCES

[1] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, “Tunable and stable real-time trajectory planning for urban autonomous driving,” in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 250–256.

[2] P. Hang, S. Huang, X. Chen, and K. K. Tan, “Path planning of collision avoidance for unmanned ground vehicles: A nonlinear model predictive control approach,” Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 235, no. 2, pp. 222–236, 2021.

[3] 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.

[4] J. A. Douthwaite, S. Zhao, and L. S. Mihaylova, “Velocity obstacle approaches for multi-agent collision avoidance,” Unmanned Systems, vol. 7, no. 01, pp. 55–64, 2019.

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

[6] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, “Reciprocal collision avoidance with acceleration-velocity obstacles,” in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 3475–3482.

[7] D. Wilkie, J. Van Den Berg, and D. Manocha, “Generalized velocity obstacles,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009, pp. 5573–5578.

[8] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, “The hybrid reciprocal velocity obstacle,” IEEE Transactions on Robotics, vol. 27, no. 4, pp. 696–706, 2011.

[9] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, “Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,” IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 952–964, 2016.

[10] X. Zhang, J. Ma, Z. Cheng, F. L. Lewis, and T. H. Lee, “Sequential convex programming for collaboration of connected and automated vehicles,” arXiv preprint arXiv:2101.00202, 2021.

[11] Z. Cheng, J. Ma, X. Zhang, C. W. de Silva, and T. H. Lee, “ADMM-based parallel optimization for multi-agent collision-free model predictive control,” arXiv preprint arXiv:2101.09894, 2021.

[12] X. Zhang, J. Ma, Z. Cheng, S. Huang, C. W. de Silva, and T. H. Lee, “Accelerated hierarchical ADMM for nonconvex optimization in multi-agent decision making,” arXiv preprint arXiv:2011.00463, 2020.

[13] M. Defoort, A. Doniec, and N. Bouraoui, “Decentralized robust collision avoidance based on receding horizon planning and potential field for multi-robots systems,” in Informatics in Control Automation and Robotics. Springer, 2011, pp. 201–215.

[14] H. Cheng, Q. Zhu, Z. Liu, T. Xu, and L. Lin, “Decentralized navigation of multiple agents based on ORCA and model predictive control,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 3446–3451.

[15] X. Zhang, J. Ma, S. Huang, Z. Cheng, and T. H. Lee, “Integrated planning and control for collision-free trajectory generation in 3d environment with obstacles,” in 2019 19th International Conference on Control, Automation and Systems (ICCAS). IEEE, 2019, pp. 974–979.

[16] C. E. Luís and A. P. Schoellig, “Trajectory generation for multi-agent point-to-point transitions via distributed model predictive control,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 375–382, 2019.

[17] H. Zhu and J. Alonso-Mora, “Chance-constrained collision avoidance for mavs in dynamic environments,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 776–783, 2019.

[18] M. da Silva Arantes, C. F. M. Toledo, B. C. Williams, and M. Ono, “Collision-free encoding for chance-constrained nonconvex path planning,” IEEE Transactions on Robotics, vol. 35, no. 2, pp. 433–449, 2019.

[19] J. Ma, Z. Cheng, X. Zhang, A. A. Mamun, C. W. de Silva, and T. H. Lee, “Data-driven predictive control for multi-agent decision making with chance constraints,” arXiv preprint arXiv:2011.02113, 2020.

[20] J. Ma, Z. Cheng, X. Zhang, M. Tomizuka, and T. H. Lee, “Optimal decentralized control for uncertain systems by symmetric Gauss-Seidel semi-proximal ALM,” IEEE Transactions on Automatic Control, 2021.

[21] L. Blackmore, H. Li, and B. Williams, “A probabilistic approach to optimal robust path planning with obstacles,” in 2006 American Control Conference. IEEE, 2006, pp. 2831–2837.

[22] X. Zhang, J. Ma, Z. Cheng, S. Huang, S. S. Ge, and T. H. Lee, “Trajectory generation by chance-constrained nonlinear mpc with probabilistic prediction,” IEEE Transactions on Cybernetics, 2020.

[23] L. Blackmore, M. Ono, and B. C. Williams, “Chance-constrained optimal path planning with obstacles,” IEEE Transactions on Robotics, 2020.

[24] D. Lenz, T. Kessler, and A. Knoll, “Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles,” in 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015, pp. 292–297.

[25] T. Nageli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, “Real-time planning for automated multi-view drone cinematography,” ACM Transactions on Graphics (TOG), vol. 36, no. 4, pp. 1–10, 2017.