2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Multi-Robot Collision Avoidance with Probabilistic Mahalanobis Distance Constraints

Zhaodong Chen[1], Dingfu Liu[1], Chuqing Feng[1] and Yunxiao Shan[1]

***Abstract*—In multi-robot systems operating under uncertainty, maintaining safe inter-robot distances while avoiding collisions with obstacles is crucial. Although chance-constrained methods have been widely adopted to handle such uncertainties, existing approaches often exhibit conservatism due to their reliance on linearized integration regions. To address this limitation, this paper introduces a novel probabilistic Mahalanobis distance constraint that enables tighter reformulations of collision avoidance constraints both between robots and between robots and obstacles. These constraints are integrated into a Model Predictive Path Integral (MPPI) control framework for efficient trajectory optimization. The effectiveness of the proposed method is validated through comprehensive simulations comparing it against state-of-the-art approaches, as well as through real-world experiments conducted across various scenarios. The code and video of our experiment could be available at the https://github.com/CreedonChan/MRCA-PMDC.**

***Index Terms*—multi-robot systems, obstacle avoidance under uncertainty, Mahalanobis distance constraint**

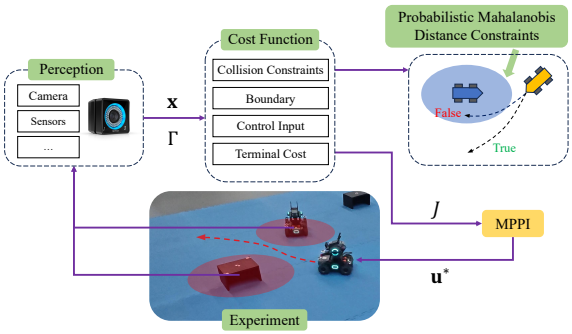


Fig. 1. Multi-robot collision avoidance framework with probabilistic Mahalanobis distance constraint and MPPI algorithm. It get the state of other objects with Gaussian noise ($\mathbf{x}$ and Γ) through perception, and compute the probabilistic Mahalanobis distance constraints and others for cost function. Then MPPI generates the optimal control $\mathbf{u}^*$ to robot given the cost J.

I. Introduction

The most critical requirement in multi-robot systems [1] is ensuring that each robot maintains a safe distance from others while avoiding collisions with potential obstacles. Although collision avoidance has been extensively studied [2]–[4], most existing methods rely on idealized assumptions, such as perfect localization and perception measurement [5]. In real-world scenarios, however, robots often face uncertainties due to measurement noise. For instance, localization measurement may become unreliable in urban canyons or dense foliage. Consequently, improving safety and ensuring reliable collision avoidance in multi-robot systems under uncertainty remains a crucial research challenge.

A classical approach to multi-robot collision avoidance is the Velocity Obstacle (VO) method [6]. However, this method relies on the assumption of perfect measurement, which is often unrealistic in practical scenarios. To address this limitation, Zhu et al. proposed the CCNMPC (Chance-Constrained Nonlinear Model Predictive Control) method for collision avoidance among multiple UAVs [7]. Their approach integrates 3D chance constraints into NMPC to provide probabilistic safety guarantees for maintaining safe distances between UAVs and obstacles. However, this method relies on linearized chance constraints, which inherently introduce conservatism and may limit performance. Similarly, decentralized collision avoidance methods based on Voronoi diagrams [8] also employ linear chance constraints, but face challenges in effectively balancing multiple constraints during operation.

The development of robust collision avoidance under uncertainty faces two fundamental challenges. First, while chance constraints have been widely adopted to handle uncertainty, most existing methods focus on applying them rather than improving their formulation. Current approaches for converting chance constraints into deterministic constraints are inherently conservative due to their reliance on linear integration regions. Therefore, developing tighter obstacle avoidance constraints is a critical priority. Second, while MPC method has emerged as the preferred optimization framework owing to its simplicity and proven effectiveness [7], [9], its conventional implementation faces significant limitations. By enforcing chance constraints as rigid boundaries, standard MPC approaches suffer from progressively shrinking feasible regions and deteriorating numerical stability as planning horizons extend. These computational challenges frequently lead to either infeasible solutions or excessively cautious trajectories, particularly in constrained environments or when dealing with uncooperative dynamic obstacles.

Being aware of the above problems, we proposed efficient collision avoidance constraints by integrating Mahalanobois distance constraint with the MPPI (Model Predictive Path Integral) [10] planning framework. Our contributions are listed

This work was supported partly by National Natural Science Foundation of China (No. 62232008), partly supported by the Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515010252. *(Corresponding author: Yunxiao Shan.)*

[1]Zhaodong Chen and Dingfu Liu are with the School of Mathematics (Zhuhai), Chuqing Feng is in the School of Software Engineering, and Yunxiao Shan is with the School of Artificial Intelligence, Sun Yat-sen University, & the Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai, Guangdong, P.R.China (Email: shanyx@mail.sysu.edu.cn).

as followings:

- A novel framework combing the Mahalanobis Distance Constraint and MPPI, which effectively address complex uncertain challenges in multi-robot obstacle avoidance.
- A more exact and tighter probabilistic Mahalanobis distance boundary is established, providing a safe constraint for multi-robot system with Gaussian noise uncertainty.
- Simulations and experiments in three cases, including symmetrical, static obstacle, and dynamic obstacle scenarios, demonstrate the exceptional collision avoidance capability and computational efficiency of our method.

II. Related Work

The VO method [2] represents a classical approach to collaborative collision avoidance. Its improved variant, the Reciprocal Velocity Obstacle (RVO) [5], introduced reciprocal reasoning between agents. Recent advances have integrated RVO with Deep Reinforcement Learning (DRL) to further enhance reciprocal collision avoidance performance [11]. There are also other learning-based framework. For instances, Bai et al. [12] employs a dueling DQN with CNN and LSTM modules to process sensor data and neighbor states, enabling decentralized collision avoidance within a formation.

Although these methods demonstrate effectiveness in ideal scenarios with perfect state information, they fail to account for the risks posed by real-world uncertainties. To address this limitation, researchers have developed probabilistic approaches that transform chance constraints into deterministic formulations within optimization frameworks [7], [13]. Notably, Zhu et al. [7] proposed a CCNMPC method that encodes probabilistic collision constraints through deterministic expressions involving the mean and covariance of the robot's state. A common challenge of algorithms with NMPC is low computational efficiency [14]. Without incorporating heuristic strategies, NMPC is prone to being trapped in local optima in non-convex problems and may fail to find feasible solutions under numerous hard constraints [15], [16]. For global planning, the CC-K-CBS framework [17] combines Kinodynamic Conflict-Based Search (K-CBS) with Gaussian belief trees to generate safe trajectories from start to goal while accounting for uncertainties.

In addition to probabilistic optimization methods, several approaches have been developed that utilize distance or regional constraints to ensure safe navigation. The Probabilistic Safety Barrier Certificate (PrSBC) method [18] employs Control Barrier Functions (CBF) to construct safe control action spaces. Another approach involves the construction of Buffered Uncertainty-Aware Voronoi Cells (B-UAVC) [8] to compute chance-constrained safe regions. Nevertheless, it is difficult for this method to deal with the non-cooperation agents.

In summary, converting probabilistic chance constraints into deterministic formulations has become a standard approach for handling navigation uncertainties in optimization frameworks. However, current methods predominantly rely on linear chance constraints, which inherently limit their ability to overcome conservatism. This fundamental limitation underscores the need for novel constraint transformation methodologies.

III. Preliminaries

Throughout this paper, we adopt the following notation conventions. Bold lowercase letters such as $\mathbf{x}$ or $\boldsymbol{v}$ denote vectors, while uppercase letters (e.g., A, M) represent matrices. In particular, Γ and Σ denotes covariance matrices of state and position respectively. Script letters (e.g., $\mathcal{I}, \mathcal{O}$) are used to denote sets. The operator $\Pr\{\cdot\}$ indicates the probability of an event. $\hat{\cdot}$ denotes the expectation of a random variable. $\cdot^\top$ denotes the transpose of vector or matrix. The symbol $\boldsymbol{\tau}_i = \{\mathbf{x}_i^t\}_{t=0}^T$ is used to represent the trajectory of robot i from time $t = 0$ to T, in which $\mathbf{x}$ denotes a state vector. $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$. Unless otherwise specified, scalars are represented by lowercase letters, and constants or parameters by Greek letters.

Firstly, consider a multi-robot system with n robots in a set $\mathcal{I} = \{1, 2, \ldots, n\}$, where each robot $i \in \mathcal{I}$ is modeled as a rigid body with radius r_i, a state space $\mathcal{X}_i$ and a control space $\mathcal{U}_i$. The workspace $\mathcal{C} \subseteq \mathbb{R}^d$ is a bounded d-dimensional Euclidean space, where d is typically 2 or 3. The state of robot i at time step t is represented by $\mathbf{x}_i^t = [\mathbf{p}_i^t, \mathbf{v}_i^t, \boldsymbol{\theta}_i^t]^\top \in \mathcal{X}_i$, where $\mathbf{p}_i^t \in \mathcal{C}$ is position, $\mathbf{v}_i^t$ velocity and $\boldsymbol{\theta}_i^t$ orientation. Robot's control input is denoted by $\mathbf{u}_i^t \in \mathcal{U}_i$. The dynamics of robot i are described by the discrete-time equation:

$$\mathbf{x}_i^{t+1} = \mathbf{f}(\mathbf{x}_i^t, \mathbf{u}_i^t; \Delta t) + \boldsymbol{\omega}_i^t, \tag{1}$$

where $\mathbf{f}$ is the system dynamics function and Δt is the time step duration. Each robot has a starting state $\mathbf{x}_i^0$ and a goal position $\mathbf{p}_i^g$, where we assume that $\mathbf{x}_i^0 \sim \mathcal{N}(\hat{\mathbf{x}}_i^0, \Gamma_i^0)$. $\boldsymbol{\omega}_i^t \sim \mathcal{N}(\mathbf{0}, Q_i^t)$ denotes the process noise in dynamics with the diagonal covariance Q_i^t. In our paper, we almost adopt single-integrator dynamics in simulations and experiments, i.e. $\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + \mathbf{v}_i^t \Delta t + \boldsymbol{\omega}_i^t$, $\mathbf{v}_i^t = \mathbf{u}_i^t$.

Next, each obstacle is modeled as static or dynamic one, non-penetrating rectangle with half of its diagonal line r_o and position $\mathbf{p}_o \in \mathbb{R}^d$, where $\mathbf{p}_o \sim \mathcal{N}(\hat{\mathbf{p}}_o, \Sigma_o)$. $\mathcal{O}$ is the set of obstacles, and o be the index of an obstacle. Furthermore, a constant velocity model is adopted to characterize the motion of each dynamic obstacle from starting point $\mathbf{p}_o^t$ to destination $\mathbf{p}_o^g$ with constant velocity $\mathbf{v}_o$.

Finally, we aim to establish an optimization model to quantitatively evaluate the effectiveness of the current control scheme. Before elaboration, we introduce some notations. Let refer to both robots and obstacles collectively as objects. $\mathbf{p}_i^t$ and $\mathbf{p}_j^t$ are the positions of objects i and j at time t. Then, the relative position between them is defined as $\mathbf{p}_{ij}^t = \mathbf{p}_i^t - \mathbf{p}_j^t$. The minimum safe distance between two different objects is defined as $l_{ij} = r_i + r_j$. And ϵ is a small positive number that represents the threshold for collision avoidance.

Our goal is to find a set of control inputs by minimizing a certain cost function in the optimization model. Since the uncertainties are considered, the traditional determination optimization control model becomes stochastic optimal control.

Therefore, instead of minimizing a deterministic cost, we minimize the expected value of the cost:

$$\min_{u_i} \quad \mathbb{E}_{\mathbb{Q}}\left[\sum_{t=0}^{T-1} C(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t) + \phi(\hat{\mathbf{x}}_i^T, \mathbf{p}_i^g)\right] \tag{2a}$$

$$\text{s.t.} \quad \hat{\mathbf{x}}_i^{t+1} = \mathbf{f}(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t; \Delta t), \tag{2b}$$

$$\Pr\{\|\hat{\mathbf{p}}_{ij}^t\| \leq l_{ij}\} \leq \epsilon, \forall j \in \mathcal{I}, j \neq i, \tag{2c}$$

$$\Pr\{\|\hat{\mathbf{p}}_{io}^t\| \leq l_{io}\} \leq \epsilon, \forall o \in \mathcal{O}, \tag{2d}$$

$$\hat{\mathbf{x}}_i^t \in \mathcal{X}_i, \mathbf{u}_i^t \in \mathcal{U}_i, \forall t = 0, \ldots, T, \tag{2e}$$

in which $\mathbb{Q}$ is the probability distribution of the trajectory $\boldsymbol{\tau}_i$ induced by the control input $\mathbf{u}_i$ and $C(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t)$ is the cost associated with the state and control input of robot i at time t, and $\phi(\hat{\mathbf{x}}_i^t, \mathbf{p}_i^g)$ is the terminal cost that penalizes the distance between the final position of robot i and its goal position. Eqs. (2c) and (2d) are two collision avoidance constraints to guarantee the both probability of collision between robot i and robot j, and between robot i and obstacle o are less than ϵ.

IV. Collision Avoidance Constraints

The key to solve Eq. (2a) is to reformulate the probabilistic collision constraints into deterministic ones. Although the chance constraints method [7], [17] could identify sufficient conditions for probabilistic collision constraints, it only accounts for the probabilistic integration with linear region approximation, resulting in relative conservation. To address this limitation, we propose a tighter deterministic reformulation method based on the Mahalanobis distance.

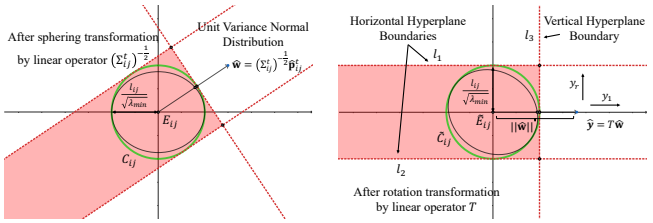
A. Probabilistic Mahalanobis Distance Constraint

Considering the uncertainty in the measurement, we model the positions as random vectors satisfying Gaussian distributions: $\mathbf{p}_i^t \sim \mathcal{N}(\hat{\mathbf{p}}_i^t, \Sigma_i^t)$ and $\mathbf{p}_j^t \sim \mathcal{N}(\hat{\mathbf{p}}_j^t, \Sigma_j^t)$, where $\hat{\mathbf{p}}_i^t$ and $\hat{\mathbf{p}}_j^t$ are the mean positions of objects i and j at time t, and Σ_i^t and Σ_j^t are the positive definite covariance matrices of the positions of objects i and j at time t. Since $\hat{\mathbf{p}}_i^t$ and $\hat{\mathbf{p}}_j^t$ are independent, we have $\mathbf{p}_{ij}^t \sim \mathcal{N}(\hat{\mathbf{p}}_{ij}^t, \Sigma_{ij}^t)$ due to the reproductive property of Gaussian distribution, where $\hat{\mathbf{p}}_{ij}^t = \hat{\mathbf{p}}_i^t - \hat{\mathbf{p}}_j^t$ and $\Sigma_{ij}^t = \Sigma_i^t + \Sigma_j^t$. Before presenting the Theorem 2 (Probabilistic Mahalanobis Distance Constraint), which provides a sufficient condition for the probabilistic collision constraint in Eqs. (2c) and (2d), we introduce a lemma that will be used in the proof of that.

Lemma 1: Given a positive-definite matrix $\Sigma \in \mathbb{R}^{d\times d}$ and a vector $\mathbf{z} \in \mathbb{R}^d$, then $\lambda_{min}\mathbf{z}^\top\mathbf{z} \leq \mathbf{z}^\top\Sigma\mathbf{z} \leq \lambda_{max}\mathbf{z}^\top\mathbf{z}$, where λ_{min} and λ_{max} are the smallest and largest eigenvalues of Σ.

Theorem 2: If the expectation of relative position $\hat{\mathbf{p}}_{ij}^t$ between object i and object j satisfies the probabilistic Mahalanobis distance constraint:

$$\sqrt{(\hat{\mathbf{p}}_{ij}^t)^\top(\Sigma_{ij}^t)^{-1}\hat{\mathbf{p}}_{ij}^t} > \frac{l_{ij}}{\sqrt{\lambda_{min}}} - \Phi^{-1}(\epsilon \cdot (2\Phi(\frac{l_{ij}}{\sqrt{\lambda_{min}}}) - 1)^{1-d}), \tag{3}$$



(a) (b)

Fig. 2. Proof sketch. (a) The space after covariance normalization. The black ellipse E_{ij} is the transformed integration region where collisions may occur. (b) The space after rotation transformation. The green circle C_{ij} is extended to a red semi-infinite rectangular region, which is our final integration region.

where λ_{min} is the minimum eigenvalue of Σ_{ij}^t, Φ is the cumulative distribution function of the standard normal distribution and l_{ij} is the minimum safe distance between object i and object j. Then we can guarantee that:

$$\Pr\{\|\mathbf{p}_{ij}^t\| < l_{ij}\} < \epsilon. \tag{4}$$

Proof: Firstly, due to the dependence among components in a multivariate Gaussian distribution, probability calculations become computationally challenging. So, we use a linear operator $(\Sigma_{ij}^t)^{-\frac{1}{2}}$ to transform the Euclidean space, which aims to make the covariance of relative position Σ_{ij}^t into unit variance. The transformed space is equivalent to the original Euclidean space, and the transformed relative position exhibits independence in all directions, facilitating subsequent region expansion operations. Let $\mathbf{w} = (\Sigma_{ij}^t)^{-\frac{1}{2}}\mathbf{p}_{ij}^t$ called transformed relative postion and $\mathbf{w} \sim \mathcal{N}(\hat{\mathbf{w}}, I)$, where $\hat{\mathbf{w}} = (\Sigma_{ij}^t)^{-\frac{1}{2}}\hat{\mathbf{p}}_{ij}^t$. It notes that

$$\|\hat{\mathbf{w}}\| = \sqrt{(\hat{\mathbf{w}})^\top\hat{\mathbf{w}}} = \sqrt{(\hat{\mathbf{p}}_{ij}^t)^\top(\Sigma_{ij}^t)^{-1}\hat{\mathbf{p}}_{ij}^t}. \tag{5}$$

In Fig. 2(a), the ellipse E_{ij} is the integration region where collisions may occur, i.e.

$$\|\mathbf{p}_{ij}^t\| = \|(\Sigma_{ij}^t)^{\frac{1}{2}}\mathbf{w}\| = \sqrt{\mathbf{w}^\top\Sigma_{ij}^t\mathbf{w}} < l_{ij}. \tag{6}$$

Generally, integration over elliptical domains is generally difficult due to the lack of symmetry. Considering that circle possesses symmetry in all directions and facilitates our subsequent effort to identify a compact region bounded by multiple hyperplanes. Then we can extend the integral over an elliptical region to an integral over a circular region, i.e the green circle C_{ij} in Fig. 2(a). So, according to Eq. (6) and Lemma 1, we have

$$\|\mathbf{p}_{ij}^t\| = \sqrt{\mathbf{w}^\top\Sigma_{ij}^t\mathbf{w}} \geq \sqrt{\lambda_{min}}\|\mathbf{w}\|. \tag{7}$$

Thus, we successfully increased the probability integral over the elliptical region as follow.

$$\Pr(\|\mathbf{p}_{ij}^t\| < l_{ij}) \leq \Pr(\|\mathbf{w}\| < \frac{l_{ij}}{\sqrt{\lambda_{min}}}). \tag{8}$$

The left side of the Ineq. (8) represents the integral over an elliptical region and the right side is the integral over a circular region with radius of $\frac{l_{ij}}{\sqrt{\lambda_{min}}}$.

Subsequently, to facilitate next region scaling operation, we need to rotate the expectation of the transformed relative position $\hat{\mathbf{w}}$ onto the coordinate axes via a linear rotation transformation. Notice that $\hat{\mathbf{w}} \in \mathbb{R}^d$, there exists an orthogonal rotation matrix $T \in \mathbb{R}^{d\times d}$ such that

$$T\hat{\mathbf{w}} = (\|\hat{\mathbf{w}}\|, 0, \cdots, 0)^\top.$$

Then fix $\mathbf{y} = T\mathbf{w}$ and $\mathbf{y} \sim \mathcal{N}(\hat{\mathbf{y}}, I)$, where $\hat{\mathbf{y}} = T\hat{\mathbf{w}} = (\|\hat{\mathbf{w}}\|, 0, \cdots, 0)^\top$ i.e.

$$\begin{aligned} y_1 &\sim \mathcal{N}(\|\hat{\mathbf{w}}\|, 1); \\ y_r &\sim \mathcal{N}(0, 1), r = 2, \cdots, d. \end{aligned} \tag{9}$$

Fig. 2(b) is the space after rotation transformation. It clearly shows that the transformed relative position random vector $\mathbf{y}$ comprises independent components, as given by the Eq. (9).

Furthermore, we extend the previously scaled circular region to an integration region partitioned by several hyperplanes. For instances in Fig. 2(b), the green circle $\tilde{C}_{ij}$ is extended to a red semi-infinite rectangular region. There are two horizontal (l_1, l_2) and one vertical (l_3) hyperplane boundaries, all tangent to the green circle. According to $\|\mathbf{y}\| = \|T\mathbf{w}\| = \|\mathbf{w}\|$, Ineq. (7) and Eq. (9), the probability integral over the region during its expansion is equivalent to the following inequality.

$$\Pr(\|\mathbf{p}_{ij}^t\| < l_{ij}) \tag{10a}$$
$$\leq \Pr(\sqrt{\lambda_{min}}\|\mathbf{w}\| < l_{ij}) \tag{10b}$$
$$= \Pr(\sqrt{\lambda_{min}}\|\mathbf{y}\| < l_{ij}) \tag{10c}$$
$$= \Pr\{\sqrt{\sum_{r=1}^{d} y_r^2} < \frac{l_{ij}}{\sqrt{\lambda_{\min}}}\} \tag{10d}$$
$$\leq \prod_{r=1}^{d} \Pr\{-\frac{l_{ij}}{\sqrt{\lambda_{\min}}} < y_r < \frac{l_{ij}}{\sqrt{\lambda_{\min}}}\} \tag{10e}$$
$$\leq \Pr\{y_1 < \frac{l_{ij}}{\sqrt{\lambda_{\min}}}\} \prod_{r=2}^{d} \Pr\{-\frac{l_{ij}}{\sqrt{\lambda_{\min}}} < y_r < \frac{l_{ij}}{\sqrt{\lambda_{\min}}}\} \tag{10f}$$
$$= (\Phi(\frac{l_{ij}}{\sqrt{\lambda_{\min}}} - \|\hat{\mathbf{w}}\|))(2\Phi(\frac{l_{ij}}{\sqrt{\lambda_{\min}}}) - 1)^{d-1}, \tag{10g}$$

where Ineq. (10d) represents the probability integral over the green circular region, and Ineq. (10f) represents the probability integral over the red semi-infinite rectangular region in Fig. 2(b).

Eventually, derived from Ineq. (3) and Eq. (5),

$$(3) \iff \Phi(\frac{l_{ij}}{\sqrt{\lambda_{\min}}} - \|\hat{\mathbf{w}}\|)(2\Phi(\frac{l_{ij}}{\sqrt{\lambda_{\min}}}) - 1)^{d-1} < \epsilon, \tag{11}$$

which combined with the above Ineq. (10g), can confirm that

$$\Pr(\|\mathbf{p}_{ij}^t\| < l_{ij}) < \epsilon.$$

Thus, the conclusion holds. ■

Eq. (3) is a sufficient condition for the collision avoidance constraints between object i and object j, according to the definition of probabilistic collision conditions (2c) and (2d). It quantifies the separation between objects relative to their uncertainty, making it a more reliable indicator for collision risk than simple Euclidean distance.

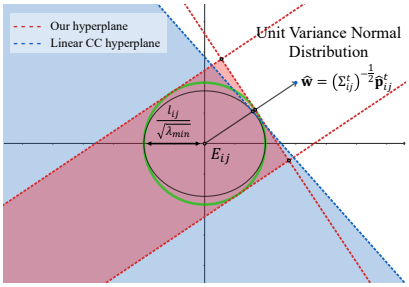


Fig. 3. Comparison of region of integration between our method and linear chance constraint method. The entire space is transformed by a linear operator $(\Sigma_{ij}^t)^{-\frac{1}{2}}$. The ellipsoid E_{ij} (black line) is integration region where collisions may occur. The red one is our integration region and the blue is the linear chance constraint region.

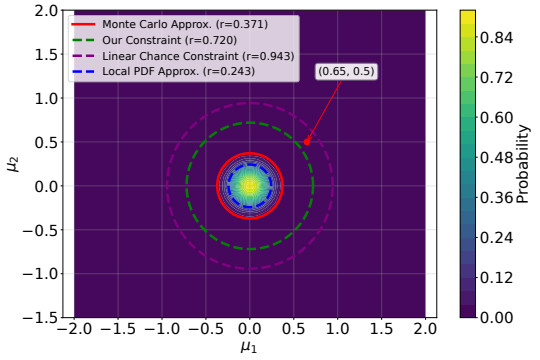


Fig. 4. Constraints boundary of different methods. The probability value at each point represents the integral over a circular region (centered at the origin with a radius equal to the safety distance) of the combined Gaussian distribution whose center is located at that point.

Remark 3: In section V-A, we would utilize this sufficient condition to design constraints among robots and obstacles in cost function. For simplicity, the collision avoidance constraints between object i and object j can be reformulated as follows:

$$m_{ij}^t > \Xi_{ij}, \tag{12}$$

where $m_{ij}^t = \sqrt{(\hat{\mathbf{p}}_{ij}^t)^\top (\Sigma_{ij}^t)^{-1}\hat{\mathbf{p}}_{ij}^t}$, which is the Mahalanobis distance between objects i and j at time t accounting for their uncertainties, and $\Xi_{ij} = \frac{l_{ij}}{\sqrt{\lambda_{min}}} - \Phi^{-1}(\epsilon \cdot (2\Phi(\frac{l_{ij}}{\sqrt{\lambda_{min}}}) - 1)^{1-d})$.

B. Comparison of Collision Avoidance Constraints

Fig. 3 shows the integration regions of our method and the linear chance constraint [7]. The linear chance constraint uses a single hyperplane-bounded region to approximate the elliptical region, which leads to an overestimated collision probability compared to the true value. However, our method approximates the elliptical region by constructing a circle and uses three hyperplanes to approximate this circular region, resulting in a more accurate collision probability.

Furthermore, we compare our collision avoidance constraints to the others, including Monte Carlo, linear chance

constraint [7], [8] and local PDF approximation [19] in numerical simulation. We assume a robot located in origin with covariance diag(0.30 m, 0.30 m)^2 and its radius is 0.10 m and other robots have the same covariance and radius, Fig. 4 is the result of different constraints boundaries with the collision probability threshold ϵ = 0.04. The result of Monte Carlo method is the actual boundary of the safe region (0.37 m). Our boundary (0.72 m) demonstrates closer proximity to the actual boundary compared to linear chance constraint (0.94 m) and since we consider a tighter region to compute collision probability. If there is a robot at (0.65, 0.5), our method would classify it as safe, whereas the methods with linear chance constraint would predict a collision. Furthermore, the boundary generated by local PDF approximation method (0.24 m) is much smaller than actual one, which carries a high collision risk.

V. Stochastic Trajectory Optimization

CCNMPC methods are widely adapted for optimization and most existing CCNMPC methods treat the transformed chance constraints as hard constraints, leading to increased computational complexity and solver instability while reducing computational efficiency. To overcome these limitations, our approach adopts the Model Predictive Path Integral (MPPI) framework [10], which is well-known for its computational efficiency due to its inherent parallel computation capability. Furthermore, MPPI provides a natural way to implement constraints as soft constraints that can be seamlessly incorporated into the cost function, offering both computational advantages and improved numerical stability compared to traditional hard-constrained approaches.

A. Local Planning of Single Robot in MPPI

MPPI is a sampling-based optimal control method that uses Monte Carlo sampling to estimate the optimal control inputs for a system, which includes sampling trajectories, evaluating costs, and updating control inputs based on the sampled trajectories. We design the cost function for the trajectory optimization problem in Eq. (2a):

$$J(\hat{\mathbf{x}}_i, \mathbf{u}_i) = \sum_{t=0}^{T-1} C(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t) + \phi(\hat{\mathbf{x}}_i^T, \mathbf{p}_i^g),$$

where $C(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t)$ consists of four parts: inter-robots collision avoidance constraints c_r, obstacle avoidance constraints c_o, boundary constraints c_o and control constraints c_u, i.e.

$$C(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t) = c_r(\hat{\mathbf{x}}_i^t) + c_o(\hat{\mathbf{x}}_i^t) + c_b(\hat{\mathbf{x}}_i^t) + c_u(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t).$$

1) Inter-robots and obstacle avoidance Constraints: We define the inter-robots collision cost function for robot i as follows:

$$c_r(\hat{\mathbf{x}}_i^t) = \Theta \cdot \mathbf{1}_{\exists j \in \mathcal{I}, m_{ij}^t > \Xi_{ij}, j \neq i}, \tag{13}$$

where Θ is a large positive constant. m_{ij}^t, the probabilistic Mahalanobis distance between robot i and robot j at time t and Ξ_{ij} are defined in Remark 3.

Similarly, the collision avoidance constraints for robot i and obstacle o is defined as:

$$c_o(\hat{\mathbf{x}}_i^t) = \Theta \cdot \mathbf{1}_{\exists o \in \mathcal{O},\ m_{io}^t > \Xi_{io}}. \tag{14}$$

2) Boundary Constraints: To ensure that the robot stays within the workspace $\mathcal{C}$, we define boundary constraints:

$$c_b(\hat{\mathbf{x}}_i^t) = \Theta \cdot \mathbf{1}_{\mathbf{p}_i^t \notin \mathcal{C}}, \tag{15}$$

where $\mathcal{C}$ is the workspace.

3) Control Constraints: The control constraints is defined as follows:

$$c_u(\hat{\mathbf{x}}_i^t, \mathbf{u}_i^t) = \frac{1}{2} \mathbf{u}_i^{t\top} R(\hat{\mathbf{x}}_i^t, t) \mathbf{u}_i^t, \tag{16}$$

where $R(\hat{\mathbf{x}}_i^t, t)$ is a positive definite matrix that penalizes large control inputs.

4) Terminal Cost: To encourage the robot to reach its goal position, we define a terminal cost function:

$$\phi(\hat{\mathbf{x}}_i^t, \mathbf{p}_i^g) = \beta^T \cdot \frac{||\mathbf{p}_i^g - \hat{\mathbf{p}}_i^T||^2}{||\mathbf{p}_i^g - \hat{\mathbf{p}}_i^0||^2}, \tag{17}$$

where β^T is a positive constant.

B. Multi-Robot Planning

To extend the stochastic trajectory optimization algorithm to a multi-robot system, we use a decentralized approach named the Constant Velocity Model Without Communication [7] where each robot independently optimizes its trajectory while considering the predicted trajectories of other robots as dynamic obstacles. This model assumes that other robots move with a constant velocity in the short term. For robot i, we can define the state of the other robot j ($j \neq i$) as $\mathbf{x}_j^t = [\mathbf{p}_j^t, \mathbf{v}_j^t]^\top$. Upon perception obtaining $\hat{\mathbf{x}}_j^t$, the robot i can predict the trajectory of the robot j as follows:

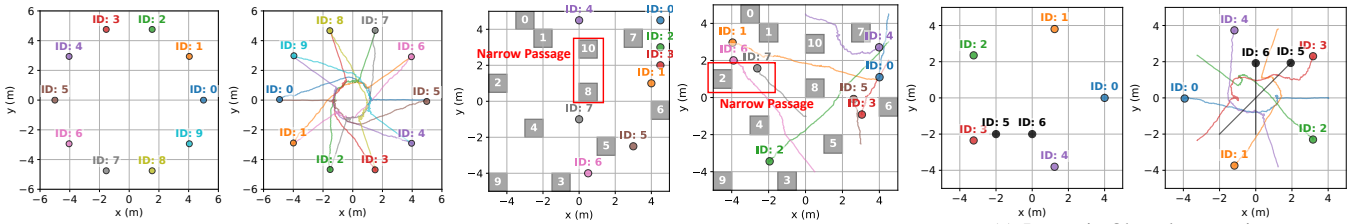
$$\hat{\mathbf{x}}_j^{t+1} = F_j \hat{\mathbf{x}}_j^t, \tag{18}$$

$$\Gamma_j^{t+1} = F_j \Gamma_j^t {F_j}^\top + Q_j^t, \tag{19}$$

where the dynamics matrix $F_j = \left(\begin{smallmatrix} I_d & \Delta t I_d \\ \mathbf{0} & I_d \end{smallmatrix}\right)$, Γ_j^t is the noise of the $\mathbf{x}_j^t$ and Q_j^t is the process noise of the constant velocity model. Each robot i maintains a local copy of the predicted trajectories of other robots $\{\boldsymbol{\tau}_j\}_{j \in \mathcal{I}, j \neq i}$. At each iteration, robot i updates its trajectory $\boldsymbol{\tau}_i$ using the stochastic trajectory optimization algorithm, treating the predicted trajectories of other robots as part of the obstacle set $\mathcal{O}$.

VI. Simulation

To thoroughly validate our approach, we design three representative simulation scenarios encompassing key navigation challenges: (1) a symmetrical scenario examining deadlock resolution in high-density crowds, (2) a static obstacle scenario featuring a constrained narrow passage, and (3) a dynamic obstacle scenario with non-cooperative moving agents. For performance benchmarking, we compare against three established methods: CCNMPC approach [7], B-UAVC [8], and RVO [5].



(a) Symmetrical scenario (b) Static Obstacle scenario (c) Dynamic Obstacle scenario

Fig. 5. Simulation results of our algorithm in different scenarios. Each subfigure consists of two parts: the left part shows the initial positions of robots; the right part shows the final trajectories (solid lines) and goals of robots. Each robot has different color with a unique identifier. In (b), the gray square is a static obstacle. In (c), No. 5 and No. 6 robots (black) are dynamic obstacles.

TABLE I
COLLISION / TIMEOUT / SUCCESS RATE OF DIFFERENT APPROACHES IN SYMMETRICAL SCENARIO WITH 4, 6, 8 AND 10 ROBOTS.

	Ours				**CCNMPC**				**B-UAVC**				**RVO**			
Number of robots	4	6	8	10	4	6	8	10	4	6	8	10	4	6	8	10
Collision rate ↓	**0%**	**0%**	**0%**	**0%**	0%	22%	94%	100%	0%	0%	0%	0%	10%	10%	62%	96%
Timeout rate ↓	**0%**	**0%**	**0%**	**0%**	0%	0%	0%	0%	38%	0%	2%	2%	0%	0%	0%	0%
Success rate ↑	**100%**	**100%**	**100%**	**100%**	100%	78%	6%	0%	62%	100%	98%	98%	90%	90%	38%	4%

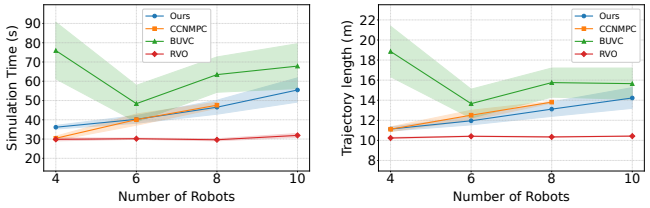
A. Simulation Preparation

To ensure comparability, a consistent parameter configuration is maintained for all scenarios and methods. The time step Δt is 0.10 s. We uniformly adopt single-integrator model among different methods on a two-dimensional plane. Each robot has 0.20 m body radius, 0.40 m/s maximum speed and a detection radius of 1.50 m. Furthermore, the uncertainty covariance of the estimated position Σ = diag(0.10 m, 0.10 m)^2, is applied to all robots and obstacles. In addition, MPPI planning parameters $\Theta = 10^6$ and $\beta^T = 200$. Moreover, we set collision probability threshold ϵ = 0.10 to each robot in both the symmetrical and static scenarios, and to 0.01 for the dynamic scenario. In all scenarios, we run the simulation 50 times. The metrics listed below are what we employ for assessing performance of each method.

- T_{si}: The average time cost to complete the simulation.
- T_{co}: The mean time used by all robots in decision-making.
- d_{rr}: Minimum distance among robots.
- d_{ro}: Minimum distance between robot and obstacle.
- l: The average length of total trajectory.
- Collision (R_{col}) / Timeout (R_{to})/ Success rate (R_{suc}): The number of simulations resulting in collision, time out and successful completion respectively divided by the total number of simulations performed. (A timeout is defined as a simulation that exceeds 1000 time steps.)

B. Symmetrical Scenario

The experimental configuration and final results are shown in Fig. 5(a). The experimental results for the symmetrical scenario, presented in Tab. I and Fig. 6, demonstrate significant performance variations among the evaluated methods. Our method achieves perfect success rates across all test cases and optimization performance, while comparative approaches exhibit distinct limitations. The RVO method performs poorly



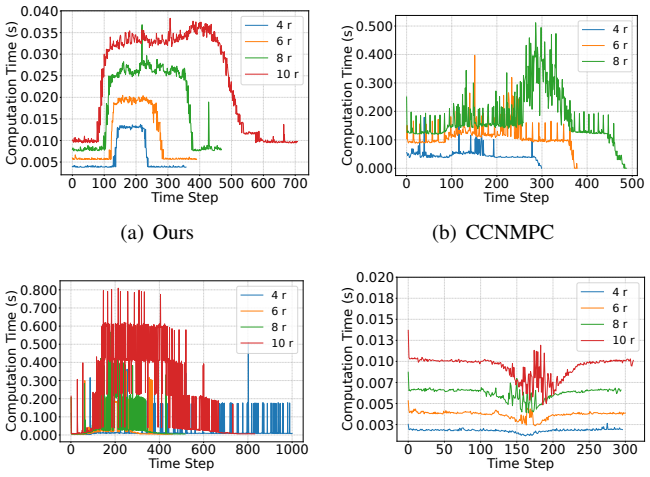
(a) Simulation time comparison (b) Trajectory length comparison

Fig. 6. Performance comparison of different approaches in symmetrical scenario with 4, 6, 8 and 10 robots. (The solid line represents the mean metric value, and the shaded envelope indicates the standard deviation.)

(4% in R_{suc}) due to its inherent inability to account for state uncertainties, resulting in frequent collisions. CCNMPC can guarantee the optimization performance (Fig. 6(b)), but shows notable sensitivity to robot population size (Tab. I), where increasing numbers of robots lead to exponentially growing hard constraints that burden the optimization solver and frequently result in infeasible solutions, as evidenced in Fig. 7. In contrast, B-UAVC maintains stable performance across different robot quantities but suffers from severe deadlock conditions, particularly evident in the 38% timeout rate observed for four-robot configurations. This deadlock phenomenon occurs when robots become trapped at vertices of their mutual sharp-corner safety regions. A plausible explanation is that the robots may enter a locally stable equilibrium state, wherein minimal motion adjustments during replanning produce virtually identical safety regions in successive iterations. This self-reinforcing mechanism maintains a static configuration that ultimately impedes collective advancement.

C. Static Obstacle Scenario

The experimental configuration and final results for the static obstacle scenario are illustrated in Fig. 5(b). Our approach demonstrates superior performance in both success rate and navigation efficiency, as quantified in Tab. II. The



(a) Ours (b) CCNMPC

(c) B-UAVC (d) RVO

Fig. 7. Computation time of different methods in symmetrical scenario. (This figure displays a representative simulation example. CCNMPC do not have the case of 10 robots because of collision.)

TABLE II
PERFORMANCE COMPARISON IN STATIC OBSTACLE SCENARIO.

Metric	Ours	CCNMPC	B-UAVC	RVO
T_{si}(s) ↓	**33.75** ± 4.48	48.35 ± 18.12	47.32 ± 18.39	-
T_{co}(ms) ↓	**2.45** ± 0.08	51.86 ± 9.46	60.58 ± 5.28	-
d_{rr}(m) ↓	**0.60** ± 0.04	0.66 ± 0.03	0.65 ± 0.03	-
d_{ro}(mm) ↓	**1086.6** ± 29.8	1092.5 ± 17.1	1092.5 ± 21.5	-
l(m) ↓	**6.08** ± 0.11	6.26 ± 0.53	6.99 ± 0.73	-
R_{col} ↓	**0%**	0%	0%	100%
R_{to} ↓	**0%**	58%	60%	0%
R_{suc} ↑	**100%**	42%	40%	0%

results reveal that our method maintains safety while permitting closer inter-robot (0.6 m) and robot-obstacle distances (1086 mm) compared to benchmark approaches, as visually confirmed in Fig. 4. Notably, CCNMPC and B-UAVC exhibit comparable performance characteristics in terms of success rate and conservatism, which can be attributed to their shared dependence on chance constraints for uncertainty management - this conservative implementation specifically reduces the navigable area between Obstacle 8 and Obstacle 10, preventing Robot 1 from successfully traversing that critical narrow passage. Robot 6 also struggles to traverse the narrow passage between Obstacle 2 and Robot 7 due to the same reason, which results in a significant number of timeout instances for both CCNMPC and B-UAVC. Furthermore, the RVO method proves inadequate in this uncertain environment, failing to complete the navigation task successfully due to its inherent inability to account for probabilistic constraints.

D. Dynamic Obstacle Scenario

In the dynamic obstacle scenario configuration shown in Fig. 5(c), five robots are initially positioned on a 4-meter radius circle and must navigate to antipodal points while avoiding two non-cooperative dynamic obstacles following fixed linear trajectories. Under the strict collision probability threshold of ϵ = 0.01, our method demonstrates robust performance with a 100% success rate, as quantified in Table III. The CCNMPC approach exhibits significant limitations in this dynamic environment, suffering a 66% collision rate primarily due to the NMPC framework's inability to find feasible solutions when dynamic obstacles approach and compress the solution space under hard chance constraints.Moreover, while CCNMPC demonstrates superior performance in certain metrics, its excessively low success rate (less than 10 in 50 runs) ultimately renders these advantages insignificant in practical applications. In contrast, our MPPI-based method overcomes this limitation through soft constraints with penalty functions, enabling the identification of viable trajectories even in high-risk regions. The B-UAVC method achieves better performance with a 10% collision rate but remains vulnerable when non-cooperative obstacles violate its safety polygon assumptions. The RVO method fails completely.

TABLE III
PERFORMANCE COMPARISON IN DYNAMIC OBSTACLE SCENARIO.

Metric	Ours	CCNMPC	B-UAVC	RVO
T_{si}(s) ↓	41.54 ± 4.27	**37.81** ± 2.52	42.39 ± 4.62	-
T_{co}(ms) ↓	**11.02** ± 0.31	63.99 ± 5.46	72.24 ± 6.74	-
d_{rr}(m) ↓	0.91 ± 0.08	**0.76** ± 0.16	0.79 ± 0.02	-
d_{ro}(m) ↓	0.82 ± 0.05	0.74 ± 0.14	**0.64** ± 0.06	-
l(m) ↓	9.07 ± 0.25	**9.04** ± 0.25	9.81 ± 0.45	-
R_{col} ↓	**0%**	66%	10%	100%
R_{to} ↓	**0%**	8%	0%	0%
R_{suc} ↑	**100%**	26%	90%	0%

VII. EXPERIMENT

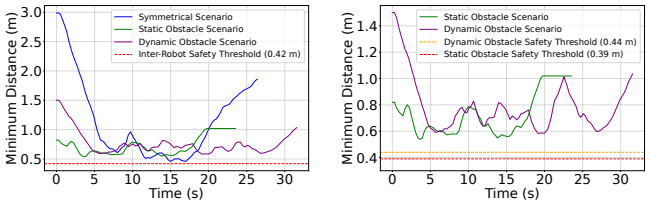
In this section, we prepare three real experiments that correspond to the scenarios in the simulation section VI.

A. Experimental Preparation

We choose a 4 m × 4 m area as the experimental region. Our experimental robots are RoboMaster EP with the radius of 0.21 m (denoting the minimum safe distance between robots is 0.42m), 0.40 m/s maximum speed and a detection radius of 1.50 m. The model of robots is single-integrator. Three static obstacles have the size of 0.3 m × 0.2 m. The dynamic obstacle is a robot with the radius of 0.23 m and the speed of 0.20 m/s, following a straight path. Furthermore, We apply the motion capture system (OptiTrack) to locate the robot. Position data is manually corrupted by additive zero-mean Gaussian noise with a covariance matrix of Σ = diag(0.10 m, 0.10 m)^2, to become estimated position. The time step Δt is 0.20 s.

B. Evaluation Results

Figs. 9, 10, and 11 present snapshots from three experimental scenarios, where elliptical regions around robots and obstacles represent probabilistic Mahalanobis distance constraints. In the symmetrical scenario, all robots successfully avoid collisions under uncertainty while reaching their targets, demonstrating effective deadlock resolution. Our framework enables robots to proactively execute lateral maneuvers, bypassing deadlock zones and recovering navigation efficiency. The static obstacle scenario validates robust navigation in



(a) Minimum inter-robot distance (b) Minimum robot-obstacle distance

Fig. 8. Minimum inter-robot and robot-obstacle distance of our algorithm in different experimental scenarios.

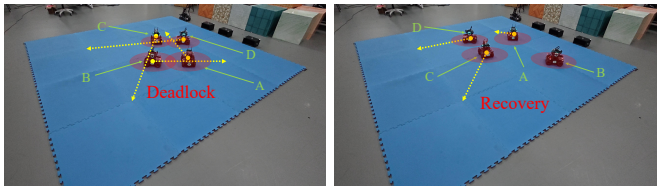


Fig. 9. Snapshots of symmetrical scenario experiment. The label from A to D denotes robots.

constrained environments. Notably, Robot D generates optimal controls to safely traverse a narrow passage between Robot A and a static obstacle, highlighting the method's capability in high-density scenarios. The soft probabilistic constraints with penalty terms ensure collision avoidance without sacrificing motion agility. For dynamic obstacles, robots exhibit enhanced collision avoidance robustness. Unlike traditional methods, our approach prevents entrapment in local minima—robots actively maneuver away from dynamic obstacles and employ lateral movements to resolve deadlocks. Quantitative results in Fig. 8 confirm the maintenance of safe distances throughout all experiments, with inter-robot distances consistently exceeding 0.42 m, static and dynamic obstacle clearances remaining above 0.39 m and 0.44 m respectively, demonstrating the method's reliability across various challenging conditions.

VIII. Conclusion

This letter presents a novel collision avoidance framework combining probabilistic Mahalanobis distance constraints with MPPI control. The proposed method simultaneously enhances navigation safety and operational efficiency through its unique constraint formulation. Extensive simulations and physical experiments validate the framework's effectiveness across three critical scenarios. While the current implementation assumes Gaussian noise distributions, future work will investigate extensions to non-Gaussian uncertainty models to further improve real-world applicability.

References

[1] R. He, Y. Shan, and K. Huang, "Robust cooperative localization with failed communication and biased measurements," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2997–3004, 2024.

[2] F. Vesentini, R. Muradore, and P. Fiorini, "A survey on velocity obstacle paradigm," *Robotics and Autonomous Systems*, vol. 174, p. 104645, 2024.

[3] Y. YunxiaoShan, "Safety planning method for autonomous driving considering localization uncertainty," *JOURNAL OF IMAGE AND GRAPHICS*, vol. 29, no. 11, pp. 3280–3292, 2024.

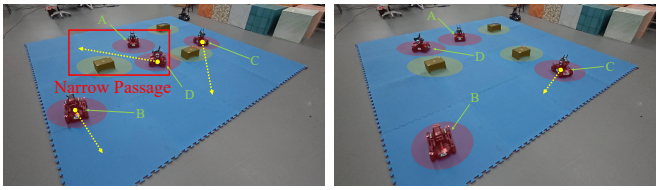


Fig. 10. Snapshots of static obstacle scenario experiment.

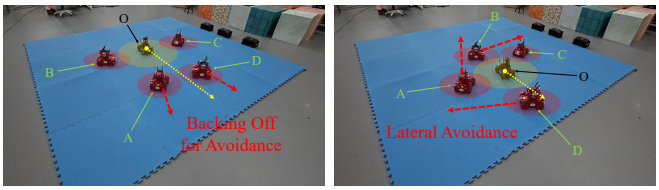


Fig. 11. Snapshots of dynamic obstacle scenario experiment. The label O denotes a dynamic obstacle and others are robots.

[4] Z. ziwei, S. yunxiao, and xundao Zhou, "Bayesian inference-based terrain elevation map completion method for off-road environments," *Journal of Image and Graphics*, 2025.

[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] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using the relative velocity paradigm," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 560–565.

[7] 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.

[8] H. Zhu, B. Brito, and J. Alonso-Mora, "Decentralized probabilistic multi-robot collision avoidance using buffered uncertainty-aware voronoi cells," *Autonomous Robots*, vol. 46, no. 2, pp. 401–420, 2022.

[9] O. de Groot, L. Ferranti, D. M. Gavrila, and J. Alonso-Mora, "Scenario-based motion planning with bounded probability of collision," *The International Journal of Robotics Research*, vol. 44, no. 9, pp. 1507–1525, 2025.

[10] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[11] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5896–5903, 2022.

[12] C. Bai, P. Yan, W. Pan, and J. Guo, "Learning-based multi-robot formation control with obstacle avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 811–11 822, 2021.

[13] K. Oguri, "Chance-constrained control for safe spacecraft autonomy: Convex programming approach," in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 2318–2324.

[14] M. Boggio, C. Novara, and M. Taragna, "Trajectory planning and control for autonomous vehicles: A "fast" data-aided nmpc approach," *European Journal of Control*, vol. 74, p. 100857, 2023.

[15] M. Doff-Sotta and M. Cannon, "Difference of convex functions in robust tube nonlinear mpc," in *2022 ieee 61st conference on decision and control (cdc)*. IEEE, 2022, pp. 3044–3050.

[16] F. Gao, Y. Han, S. E. Li, S. Xu, and D. Dang, "Accurate pseudospectral optimization of nonlinear model predictive control for high-performance motion planning," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1034–1045, 2022.

[17] A. Theurkauf, J. Kottinger, N. Ahmed, and M. Lahijanian, "Chance-constrained multi-robot motion planning under gaussian uncertainties," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 835–842, 2023.

[18] W. Luo, W. Sun, and A. Kapoor, "Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates," *Advances in Neural Information Processing Systems*, vol. 33, pp. 372–383, 2020.

[19] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011.