Distributed Collision Avoidance of Multiple Robots with Probabilistic Buffered Voronoi Cells

Mingyu Wang[1] and Mac Schwager[2]

Abstract— **This paper introduces Probabilistic Buffered Voronoi Cell (PBVC) collision avoidance for multiple robots using noisy on-board sensing. The work builds upon the previously proposed Buffered Voronoi Cell (BVC) approach. We introduce a probabilistic formulation to construct a family of BVCs with specified safety levels, which take into account uncertainty in sensor measurements among the robots. The safety level of a PBVC represents the probability that the area is contained inside the robot's true (but unknown) BVC. The PBVC provides a set of probabilistically safe positions for the robot to navigate to. Each agent chooses its next position constrained within the PBVC of a desired safety level, while minimizing the deviation from its reference trajectory. We prove a conservative bound on the probability of collision given this reciprocal strategy. We also validate through simulations that the proposed approach achieves safe navigation among multiple robots in challenging scenarios, and provides a significantly lower risk of collision than either the Reciprocal Velocity Obstacles (RVO) method or the Buffered Voronoi Cell (BVC) method when the robots use noisy relative measurements. We also show in experiments with small-scale robotic cars that the algorithm is fast, effective and is useful in real applications.**

I. Introduction

Collision avoidance with dynamic obstacles is a fundamental challenge for research with mobile robots. Each robot senses the environment including static obstacles and dynamic agents using on board sensors, while executing safe commands based on its perception of the surroundings.

In this paper, we build upon the concept of buffered Voronoi cells (BVC) to handle measurement uncertainties. Buffered Voronoi cells were introduced by Zhou et. al. [1] for reciprocal collision avoidance between multiple robots. The algorithm is distributed, meaning each robot only needs to sense the position of its neighbors to compute a safe input and no communication is needed during the process. If all the robots follow the BVC constraints, it is proved in a recursive manner that no collision will occur. The formulation works well for single-integrator robots with perfect knowledge of positions of other robots. However, this approach does not capture uncertainties in measurements.

In this work we introduce a probabilistic formulation to compute BVCs with different safety assurance levels. Safety levels represent the probabilities that a certain area is inside the relevant buffered Voronoi cell with respect to the posterior position distributions of other robots. Then, our method applies a Monte Carlo approach to find safe control actions for single-integrator robots. Given a reference position for the next time step, we apply an efficient sampling algorithm to find the optimal position that complies with the safety assurance and minimally changes the desired input. Many works have addressed the problem of collision avoidance between multiple mobile robots. Some of them take a centralized approach. These methods assume that a central controller knows the state information of all robots and plans the trajectory for all of them simultaneously. The main drawback of these approaches is that the computation time does not scale for a large number of agents. Moreover, they rely on a communication network to broadcast planned trajectories which may not be available.

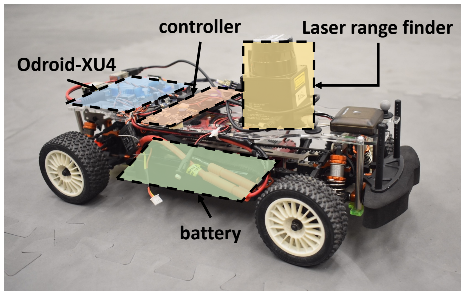


Fig. 1: Small-scale autonomous cars used in experimental validations.

Other approaches address the collision avoidance problem in a decentralized manner. A method using gyroscopic and braking forces is described in [2]. Artificial potential-field based approaches are used for collision avoidance and navigation in [3], [4]. Many attempts have also been made from a geometric perspective into the problem. [5] proposed the concept of local free space for convex-shaped robots based on separating hyperplanes. The approach is adapted for realistic sensor footprints in [6]. Velocity obstacles (VO) were originally proposed in [7]. [8] introduced reciprocal velocity obstacles, extending VO by implicitly assuming reciprocal behavior among robots. Other extensions based on VO also take into account kinematic constraints [9], [10], [11], [12] and different robot geometries [13]. [1], [14], [15] resort to the concept of Voronoi diagrams in developing decentralized collision avoidance algorithms without having velocity information about other robots. In [16], each agent construct risk level sets based on density and motion of objects to navigate through cluttered environments. Reachability analysis is also widely applied to collision avoidance

[1] Mingyu Wang is with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA, mingyuw@stanford.edu.

[2] Mac Schwager are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA, schwager@stanford.edu.

works such as [17], [18], [19], but it is computationally prohibitive for complex dynamic systems.

A emerging line of work tackles collision avoidance using deep learning approaches. In [20], an end-to-end approach is used to map directly from sensor data to steering commands. [21] uses deep reinforcement learning to training a policy network which is shown to provide efficient navigation.

As for sensor uncertainties, [22], [23] proposes probability velocity obstacles considering shape and velocity uncertainty. In [24], the authors combine probabilistic velocity obstacles to a Bayesian occupancy grid and estimate the probability of collision under uncertainty, taking into account position, shape, and velocity uncertainty of obstacles, occlusions and limited sensor range. Our PBVC approach is different from these in that we do not require robots to estimate the velocities of their neighbors, but only their relative positions.

The rest of the paper is organized as follows. We begin by summarizing the background concepts related to BVCs in Section II. We formally define the concept of probabilistic buffered Voronoi cells in Section III. In Section IV, we introduce the collision avoidance approach using PBVCs and describe an efficient algorithm for computing the optimal control input. The simulation and experimental results are provided in Section V.

II. Background and Preliminaries

The method described in this paper combines buffered Voronoi cell (BVC) collision avoidance algorithm with a target tracking framework to deal with perception uncertainty while providing safety assurance. For target tracking, we use a Bayesian filter, such as a particle filter (PF). In this section, we outline the buffered Voronoi cell algorithm which we will use in our proposed probabilistic bufferd Voronoi cell approach. For an introduction of particle filters, the readers are refered to [25].

A. *Buffered Voronoi Cell Collision Avoidance*

In this section we describe the original buffered Voronoi cell (BVC) collision avoidance approach, which was proposed by Zhou et. al. in [1].

Let's consider N disk-shaped robots with center points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N \in \mathbb{R}^2$ and uniform safety radius r_s. Note that we only consider 2D case here for notation simplicity, although the original algorithm could be applied in arbitrary dimensions. We assume that $\forall j \neq i, \mathbf{p}_i \neq \mathbf{p}_j$. The Voronoi cell for agent i with respect to all other agents, denoted by $\bar{\mathcal{V}}_i$, is defined as:

$$\mathcal{V}_i = \{\mathbf{p} \in \mathbb{R}^2 \mid ||\mathbf{p} - \mathbf{p}_i|| \leq ||\mathbf{p} - \mathbf{p}_j||, \forall j \neq i\}, \quad (1)$$

which could also be written as

$$\mathcal{V}_i = \{\mathbf{p} \in \mathbb{R}^2 \mid (\mathbf{p} - \mathbf{p}_i)^T \mathbf{p}_{ij} \leq \frac{||\mathbf{p}_{ij}||^2}{2}, \forall j \neq i\},$$

where $\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i$. By construction, Voronoi tessellation is a partition of the position space and there is no overlap between the interior of any two Voronoi cells.

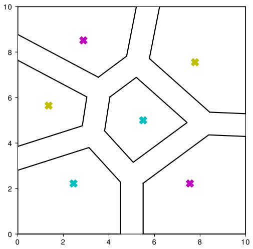


Fig. 2: Illustrations of BVCs for a group of 6 robots.

To consider the footprints of robots, we define the buffered Voronoi cell for agent i, $\bar{\mathcal{V}}_i$ as follows (see Fig. 2):

$$\bar{\mathcal{V}}_i = \{\mathbf{p} \in \mathbb{R}^2 \mid (\mathbf{p} - \mathbf{p}_i)^T \mathbf{p}_{ij} \leq \frac{||\mathbf{p}_{ij}||^2}{2} - r_s ||\mathbf{p}_{ij}||, \forall j\}. \quad (2)$$

Geometrically, buffered Voronoi cells can be interpreted as retracting the edges of Voronoi cells by a safety distance, r_s, so that if the center of a disk-robot is within its buffered Voronoi cell, its extent will be inside its corresponding Voronoi cell.

The concept of buffered Voronoi cells can be used for collision avoidance between a group of robots. We first give the definition of a *collision free configuration* and outline the collision free guarantee given in [1].

***Definition** 1: (Collision Free Configuration)* A group of N robots with identical safety radius r_s is in a collision free configuration if $\forall j \neq i, ||\mathbf{p}_i - \mathbf{p}_j|| \geq 2r_s$.

Being in a collision free configuration is a sufficient condition that, for each robot in the group, its BVC is a nonempty set. Moreover, the center point of each robot is in its corresponding BVC.

***Theorem** 1:* Let us consider a group of N robots in a collision free configuration at time step t. If each robot moves to a point $\mathbf{p}'$ inside its current buffered Voronoi cell, then the new set of positions at time $t+1$ are also in a collision free configuration.

Considering that the interior of any two Voronoi cells are disjoint, each robot can safely navigate in its BVC, given that the other robots also follow this rule. However, this algorithm requires that the robots are single-integrators. In other words, the velocities of the robots are the commanded input and can change instantaneously. For higher-order linear systems, the readers are referred to [14].

Note that Voronoi diagram evolves as each robot's position changes over time. At each time step, each robot plans a trajectory inside its current Voronoi cell and moves to a new position, then we update the Voronoi diagram for the next time step. Applying the above theorem recursively, we can guarantee that there is no collision during the process. Furthermore, Theorem 1 does not require the control method each robot could be using. Thus, it works for a group of heterogeneous robots which have different control policies.

B. Generalized Buffered Voronoi Cells for Convex-shaped Robots

For general convex-shaped robots, the uniform disk-shape assumption in the previous construction can be too restrictive to apply in real applications (such as cars on the road). Thus, generalized buffered Voronoi cells for general convex-shaped robots are introduced in [14]. For two planar robots whose extents are closed convex sets $\mathcal{O}_i \subset \mathbb{R}^2$ and $\mathcal{O}_j \subset \mathbb{R}^2$, the distance between them is:

$$d(\mathcal{O}_i, \mathcal{O}_j) = \min ||a^* - b^*||, \text{ s.t. } a^* \in \mathcal{O}_i, b^* \in \mathcal{O}_j$$

If $d(\mathcal{O}_i, \mathcal{O}_j) > 0$, robot i and j are in a collision-free configuration. Then, the generalized Voronoi cell of i with respect to j is defined as:

$$\mathcal{V}_i = \{\mathbf{p} \mid ||\mathbf{p} - a^*|| \leq ||\mathbf{p} - b^*||\}. \tag{3}$$

Additionally, generalized buffered Voronoi cells are defined using Minkovski difference as:

$$\bar{\mathcal{V}}_i = \mathcal{V}_i \ominus \mathcal{O}_i, \tag{4}$$

where the Minkovski difference between two sets $\mathcal{A} \ominus \mathcal{B}$ is:

$$\mathcal{A} \ominus \mathcal{B} = \{\mathbf{c} \mid \mathbf{c} + \mathcal{B} \subseteq \mathcal{A}\}.$$

Note that, we overload the notation $\mathcal{V}$ and $\bar{\mathcal{V}}$ in this paper since the generalized definition is consistent with the basic definition. When $\mathcal{O}_i$ and $\mathcal{O}_j$ are disks with radius r_s, (3) and (4) reduce to (1) and (2).

Extending Theorem 1 to this case needs extra attention since BVCs are created only considering the footprints of robots at current time step. For single-integrator robots, this issue can be solved by letting $\mathcal{O}$ be the convex hull of all the possible footprints with limited orientations.

III. Probabilistic Buffered Voronoi Cells

In this section we present the proposed probabilistic buffered Voronoi cell algorithm. Unlike the basic buffered Voronoi cell concept, which assumes perfect perception, the probabilistic construction explicitly takes into account the uncertainty in target tracking and computes the level of assurance of Voronoi diagrams. PBVCs use a Monte Carlo approach and have probabilistic Voronoi boundaries with different safety assurance levels. In this section, the robots are assumed to have uniform disk shape.

A. Safety-level Probabilistic Buffered Voronoi Cells

Without making any assumptions on perfect position information of the other robots, we instead have a posterior estimation of the positions $\Pr(\mathbf{p}_{o,t} \mid y_{o,1:t})$ conditioned on all previous measurements. For example, using a PF, the posterior density is denoted as a set of weighted particles $\{\mathbf{x}_t^n, w_t^n\}_{n=1}^{N_p}$. Even though $\mathbf{x}_t^n$ may contain other state information, we are only interested in position and thus ignore other states and let $\mathbf{x}_t^n \in \mathcal{R}^2$.

The idea of a probabilistic BVC is to define probabilistic boundaries. α-boundaries of a Voronoi cell mean that all the points that belong to this α-cell is inside the actual buffered Voronoi cell with at least probability α. The corresponding probability of a point inside a buffered Voronoi cell is computed with respect to the posterior density of other robots.

More formally, we first give the definition of a point with safety level α. For notation simplicity, we omit time step t in this section.

***Definition** 2: (Point with safety level α)* The safety level α for a point $\mathbf{q}$ with respect to robot o is defined as:

$$\alpha_o(\mathbf{q}) = \Pr(\mathbf{q} \in \bar{\mathcal{V}}(\mathbf{p}_o)),$$

where $\mathbf{p}_o$ is the unknown position of robot o. The weighted particles of O other robots are denoted as $\{\mathbf{x}_o^n, w_o^n\}_{n=1}^{N_p(o)}$, $o = 1, \ldots, O$, where $N_p(o)$ is the number of state estimation particles for robot o. Given the posterior particles, we compute the safety level α using a Monte Carlo approach:

$$\alpha_o(\mathbf{q}) = \frac{\sum_{i \in \mathcal{S}_o} w_o^n}{\sum_n w_o^n},$$

where

$$\mathcal{S}_o = \{n \mid \mathbf{q} \in \bar{\mathcal{V}}(\mathbf{x}_o^n)\},$$

and $\bar{\mathcal{V}}(\mathbf{p})$ is the buffered Voronoi cell of controlled robot with respect to a robot at $\mathbf{p}$.

As such, we can define the safety level of point $\mathbf{q}$ with respect to all other O robots as:

$$\alpha(\mathbf{q}) = \prod_{o=1}^{O} \alpha_o(\mathbf{q}).$$

Then, we give the definition of a buffered Voronoi cell with safety level α, which is denoted by $\bar{\mathcal{V}}_\alpha$:

***Definition** 3: (α-level probabilistic buffered Voronoi cells)*

$$\bar{\mathcal{V}}_\alpha = \{\mathbf{q} \mid \alpha(\mathbf{q}) \geq \alpha\}.$$

The definition of α-level buffered Voronoi cell implies that for any point $\mathbf{p} \in \bar{\mathcal{V}}_\alpha$, $\Pr(\mathbf{p} \in \bar{\mathcal{V}}) \geq \alpha$. An illustration of different probability boundaries of PBVCs is shown in Fig. 3. In the figure, each robot maintains a posterior estimation of the positions of all other robots. The probabilistic contours are generated using samples in position space.

Unfortunately, there is no analytical solution for the set $\bar{\mathcal{V}}_\alpha$. Thus we approximate the solution with a sampling-based approach. As the number of samples increase, we gather more information about the probabilistic boundaries of Voronoi cells, as shown in Fig. 3. Next, we introduce a property of probabilistic buffered Voronoi cells which we will use later.

***Lemma** 1: (Monotonically decreasing safety level)* Given a set of particles representing the positional probability of another robot $\{\mathbf{x}^n, w^n > 0\}_{n=1}^{N_p}$. Assume ego agent is at the origin $\mathbf{p}_e = [0, 0]$. Along a line section $u : u = u_o\theta$, where $\theta \geq 0$ and u_o is a unit vector representing the line direction, the safety level of u decreases monotonically with θ.

Proof: Pick a point u' on the line, $u' = u_o\theta'$. Assume the corresponding safety level of this point is α'. Thus we have the following:

$$\alpha' = \frac{\sum_{n \in \mathcal{S}'} w^n}{\sum_n w^n} \tag{5}$$

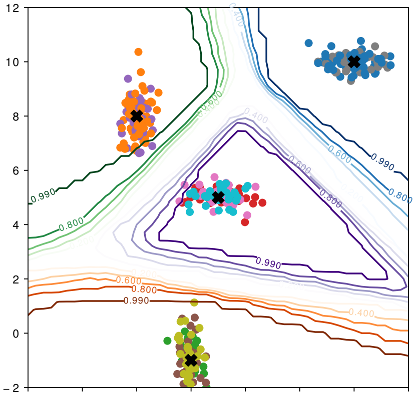


Fig. 3: Contour plot of PBVCs of a group for 4 robots. The true positions of the robots are shown as crosses. Each robot has different perception uncertainty towards the position of 3 other robots (represented by colored dots). Probabilistic boundaries with different safety levels are plotted (correspond to safety levels 0.2, 0.4, 0.6, 0.8, and 0.99).

where $\mathcal{S}' = \{n \mid u' \in \bar{\mathcal{V}}(\mathbf{x}^n)\}$ which is equivalent to $\mathcal{S}' = \{n \mid u'^T\mathbf{x}^n \leq \frac{||\mathbf{x}^n||^2}{2} - r_s||\mathbf{x}^n||\}$.

Now, consider point $u'' = \theta'' u_o$, $\theta'' < \theta'$. So $\frac{u'}{u''} = \beta > 1$. Let α'' be the corresponding safety level $\alpha'' = \frac{\sum_{n \in \mathcal{S}''} w^n}{\sum_n w^n}$. We show that the set $\mathcal{S}'' \supseteq \mathcal{S}'$.

For all n in S', we have

$$\begin{aligned}
& u'^T\mathbf{x}^n \leq \frac{||\mathbf{x}^n||^2}{2} - r_s||\mathbf{x}^n|| \\
\Leftrightarrow\ & \beta u''^T\mathbf{x}^n \leq \frac{||\mathbf{x}^n||^2}{2} - r_s||\mathbf{x}^n|| \\
\Leftrightarrow\ & u''^T\mathbf{x}^n \leq \frac{1}{\beta}\frac{||\mathbf{x}^n||^2}{2} - r_s||\mathbf{x}^n|| \leq \frac{||\mathbf{x}^n||^2}{2} - r_s||\mathbf{x}^n|| \\
\Leftrightarrow\ & u''^T\mathbf{x}^n \leq \frac{||\mathbf{x}^n||^2}{2} - r_s||\mathbf{x}^n||
\end{aligned}$$

Thus, $n \in \mathcal{S}''$. We proved that $\mathcal{S}'' \supseteq \mathcal{S}'$. Since all the weights w^ns are non-negative, we also have $\sum_{n \in \mathcal{S}''} w^n \geq \sum_{n \in \mathcal{S}'} w^n$, thus $\alpha'' \geq \alpha'$. We proved that safety level is non-increasing as θ increases along any direction u_o.

The proof also extends to general cases where multiple neighboring robots present. ■

IV. Collision Avoidance with PBVCs

In this section, we discuss the collision avoidance approach using PBVCs for robots with single integrator dynamics.

A. *Choosing Desired Point with Safety Assurance*

Assume that at any time step, the next desired positional waypoint is $\mathbf{q}_{\text{des}}$, which could be from a trajectory planner. We want to choose a point that conforms with our safety constraint. I.e. the probability we are in our buffered Voronoi cell at next time step is at least α_o. The robot selects a point $\mathbf{q}$ with $\alpha(\mathbf{q}) > \alpha_o$ that is closest to its desired waypoint.

$$\min_{\mathbf{q}} \ ||\mathbf{q} - \mathbf{q}_{\text{des}}||, \tag{7a}$$

$$\text{s.t. } \mathbf{q} \in \bar{\mathcal{V}}_{\alpha_o}. \tag{7b}$$

Since we don't have an analytical solution of $\bar{\mathcal{V}}_{\alpha_o}$, we instead resort to a suboptimal sampling-based approach. We sample M points in the position space and denote the set of points with $\alpha(\mathbf{q}) > \alpha_o$ as $\mathcal{Q}_{\alpha_o}$, which is a sampled approximation of $\bar{\mathcal{V}}_{\alpha_o}$. Thus, the above optimization problem could be rewritten as:

$$\min_{\mathbf{q} \in \mathcal{Q}_\alpha} \ ||\mathbf{q} - \mathbf{q}_{\text{des}}||, \tag{8}$$

for which the optimal solution is denoted as $\mathbf{q}^*(\mathcal{Q}_\alpha)$. Based on this approach, $\forall\ \mathcal{Q}''_\alpha$ s.t. $\mathcal{Q}''_\alpha \supseteq \mathcal{Q}'_\alpha$, $\mathbf{q}^*(\mathcal{Q}''_\alpha)$ is no greater than $\mathbf{q}^*(\mathcal{Q}'_\alpha)$. This implies that, in general, larger number of samples is beneficial to the problem. On the other hand, the computation time increases with the number of samples and the number of particles representing the probability distribution of the obstacle positions.

For real-time control, we need a more efficient sampling approach. Here, we propose an approach that combines bisection method for a modified problem (7) to provide an upperbound, then we solve the approximation problem (8).

B. *Efficient Sampling Method*

Given the exact problem (7) and a desired point $\mathbf{p}_{\text{des}}$, we first validate the safety level of $\mathbf{p}_{\text{des}}$. If the proposed point is safe, then we can continue to execute it. Otherwise, we apply the following approach. We further constrain the search space of (7) to a line, from which we can obtain an upperbound of the original problem.

$$\min_{\theta} \ ||\mathbf{p}_{\text{line}} - \mathbf{q}_{\text{des}}||, \tag{9a}$$

$$\text{s.t. } 0 \leq \theta \leq 1, \tag{9b}$$

$$\mathbf{p}_{\text{line}} = \mathbf{p}_e + \theta(\mathbf{p}_{\text{des}} - \mathbf{p}_e), \tag{9c}$$

$$\mathbf{p}_{\text{line}} \in \bar{\mathcal{V}}_{\alpha_o}, \tag{9d}$$

where $\mathbf{p}_e$ is the position of the ego robot, (9b) and (9c) limit the feasible points on the line section between $\mathbf{p}_e$ and $\mathbf{q}_{\text{des}}$, and (9d) is the safety assurance constraint.

Note that if we plug (9c) in (9a), we have the objective function $(1-\theta)||\mathbf{p}_e - \mathbf{q}_{\text{des}}||$. Ignoring the non-negative constant term, we try to maximize θ. Nonetheless, according to Lemma 1, safety level of $\mathbf{p}_{\text{line}}$ increases with the decrease of θ. So θ has to be less than a threshold to satisfy (9d). We use a line search algorithm based on bisection method. The modified bisection method is outlined in Algorithm 1.

Let $\theta^{\#}$ be the solution to this line search problem. The upperbound of the optimal value is $(1-\theta^{\#})||\mathbf{p}_e - \mathbf{q}_{\text{des}}||$.

Then we solve the approximation problem based on this bound. We uniformly sample M points in the circle centered at $\mathbf{q}_{\text{des}}$ with radius $r = (1-\theta^{\#})||\mathbf{p}_e - \mathbf{q}_{\text{des}}||$. After this, we evaluate the safety level of each point and choose the one that has the minimum objective value from $\mathcal{Q}_{\alpha_o}$.

The efficient sampling process is illustrated in Fig. 4. The grey areas represent PBVCs with different safety levels for

Algorithm 1 Line search for safety point

```
Input: isSafe($\theta$), $\mathbf{p}_e$, $\mathbf{q}_{\mathrm{des}}$, tol
Output: $\theta^{\#}$
    Initialization: $\theta_{\mathrm{s}} = 0$, $\theta_{\mathrm{us}} = 1$, err = 1
while err $\geq$ tol do
    $\theta$ = average($\theta_{\mathrm{s}}$, $\theta_{\mathrm{us}}$)
    $\mathbf{q} = \mathbf{p}_e + \theta(\mathbf{p}_{\mathrm{des}} - \mathbf{p}_e)$
    if isSafe($\mathbf{q}$) then
        $\theta_{\mathrm{s}} = \theta$
    else
        $\theta_{\mathrm{us}} = \theta$
    end if
    err = $\theta_{\mathrm{us}} - \theta_{\mathrm{s}}$
end while
return $\theta$
```

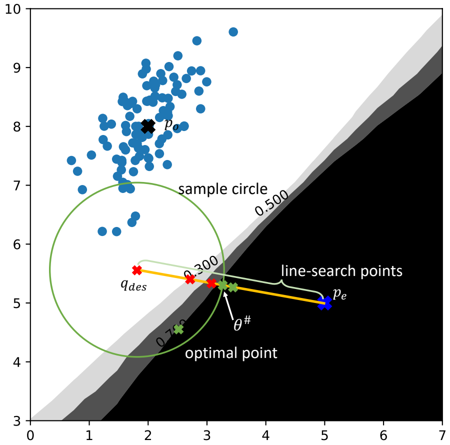


Fig. 4: An efficient sampling method to find next positional waypoint with safety assurance. The crosses along the line represent the bisection search process described in Algorithm 1. After finding $\theta^{\#}$, we uniformly sample in the sample circle to find a safe point closest to $\mathbf{q}_{\mathrm{des}}$.

the controlled robot with respect to another robot at $\mathbf{p}_o$, whose position belief is shown as blue dots. To find a point closest to $\mathbf{q}_{\mathrm{des}}$ with safety level at least 0.7, we first search on a line. The line search process is shown as crosses in red (θ_{us}) and green (θ_s). Then, we uniformly sample in the green circle with radius $(1 - \theta^{\#})$ to solve for $\mathbf{q}^*$.

C. Collision Probability for Multiple Reciprocal Robots

In the above we have shown how to choose the desired waypoint for the next time step for a single robot. The safety level α we computed corresponds to the probability that a single robot will be inside its BVC in next time step. However, this does not directly translate to the probability of a collision between robots. To analyze the probability of collisions, we need to take into account all the relevant robots. In most cases, the robots we considered as obstacles also react to ego agent's action and avoid collisions, exhibiting reciprocal behaviors. Assume that all the robots adopt the proposed collision avoidance strategy with safety level α at time step t, the probability that all the robots are inside their BVCs at the next time step is thus

$$\Pr(\mathbf{p}_i^{t+1} \in \bar{\mathcal{V}}_i^t, \forall i) = \alpha^N.$$

As discussed in [14], all the robots remaining in their BVCs is a sufficient but not necessary condition for no collision. Thus, the collision probability between any two robots $\Pr(Coll)$ is:

$$\begin{aligned}\Pr(Coll) &\leq 1 - \Pr(\mathbf{p}_i^t, \in \bar{\mathcal{V}}_i^t, \forall i) \\ &= 1 - \alpha^N.\end{aligned}$$

For $\alpha = 0.999$ and $N = 3$, the collision probability is upperbounded by 0.0030.

V. Results

A. Simulation

In this section, we present simulation results using the proposed PBVC collision avoidance algorithm. We also compare the performance with reciprocal velocity obstacle (RVO) [8] and basic buffered Voronoi cell collision avoidance approaches.

Firstly, we demonstrate the performance of PBVC collision avoidance algorithm with 3 disk-shaped single-integrator robots navigating to goal positions respectively. The desired position for each robot at next time step is simply v distance away from the current position in the direction of its goal position. Snapshots during the simulation are shown in Fig. 5. The true positions of the robots are represented by crosses. The goal locations of each robot are triangles plotted in the same color. Note that without collision avoidance, the trajectories of the robots intersect. The lines connecting each robot from their initial positions are their traces. Each robot measure the relative distance and bearing angle of the other two robots with Gaussian noise. Particle filters with 200 particles are used for each tracking target. The prediction step only adds noise to the particles without dynamic propagation. All three robots use the PBVC collision avoidance approach with safety level $\alpha = 0.99$. We run the simulation from the start configuration (shown in top left figure in Fig. 5) with small random offset for 300 times. No collision is observed during all the simulations. Due to the sampling nature of the approach, no deadlock occurred.

In the basic BVC and RVO implementation, each robot uses the expected value from particle filters as the "true" position of other robots. In addition, velocities of other robots are estimated as the position difference between time steps for RVO approach. Furthermore, we added an extra buffer of 2% robot radius for both BVC and RVO approaches to account for the measurement uncertainty. For BVC, all of the 300 simulations failed with robot colliding with each other in this scenario. For RVO, 235 simulations ended in collisions out of 300 simulations. A successful simulation using RVO is shown in Fig. 6. The reason is that given the inaccurate estimation, the robots are over-confident about their "safe regions" and thus execute actions that lead to collisions.

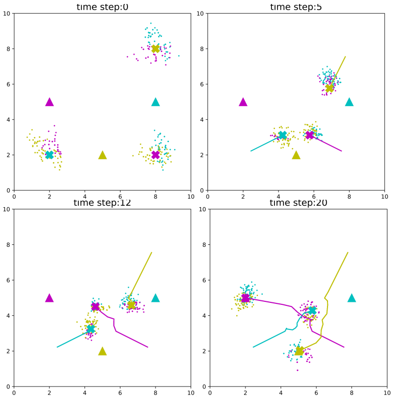


Fig. 5: Snapshots during a PBVC simulation where 3 robots navigate to their goal positions while avoiding collisions with others. True positions and goal positions are shown by crosses and triangles. The colored lines are the traces. No collisions occurred during 300 simulations.

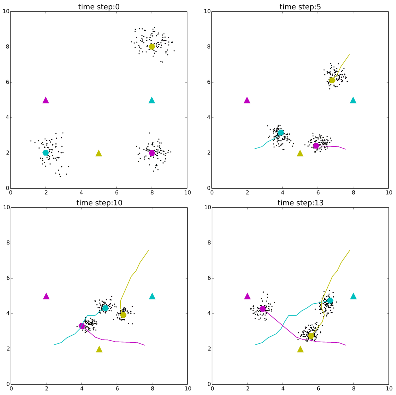
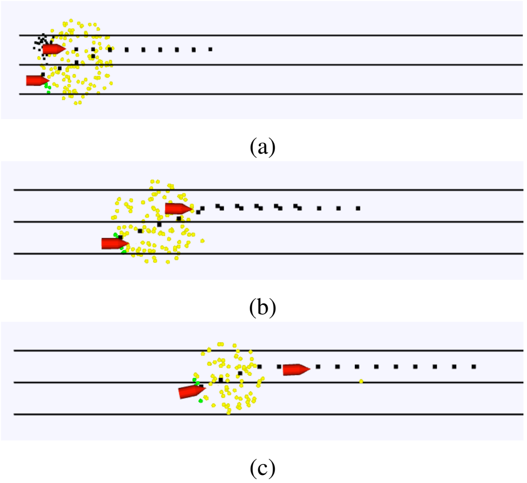


Fig. 6: Snapshots during a successful RVO simulation where 3 robots navigate to their goal positions while avoiding collisions with others.

B. Experiments using scale autonomous cars

The experimental platform we use consists of two 1/10 scale autonomous cars (see Fig. 1) with Pixfalcon flight controllers and Electronic Speed Controllers for steering and motor control. The cars also have Odroid-XU4 computers which is used for estimation of the position of the other car based on measurements from a Hokuyo URG-04LX-UG01 laser range finder. Hokuyo Lidar broadcasts measurements at $10\,\mathrm{Hz}$. An Optitrack motion capture system is also used only for position of the ego car for trajectory planning purposes. In the experiments, the ego car uses PBVC collision avoidance controller and the other car is not reactive to the ego car. The goal of the ego car is to change to the top lane and maintain in that lane. A particle filter runs at $50\,\mathrm{Hz}$. In the case where the other car is outside the sensing range (set to $1.5\,\mathrm{m}$), the ego car will assume there is no other car in the area. Although cars have different dynamics from single-integrator robots, we still show that the proposed approach is applicable. 3 snapshots during the experiment are shown in Fig. 7. Two cars start next to each other in two lanes (top figure) and the bottom car (PBVC controlled car) wants to change to the top lane. The car starting from the top lane is not reactive to the controlled car. The green dashed lines are reference trajectories. We conducted 10 experiments and there is no collision during the experiments.


(a)

(b)

(c)

Fig. 7: Snapshot during experiment with 1/10 scale autonomous cars. The yellow/green dots are samples draw in the position space for deciding the optimal next position.

VI. CONCLUSIONS

In this paper, we presented a distributed collision avoidance algorithm for multiple single-integrator robots that only has noisy position measurements of their neighbors. Each robot maintains a posterior distribution of the positions of other robots using a Bayesian filter. Then, a Monte Carlo approach is used to approximately solve an optimization problem to select the next position point with safety level constraints. In particular, probabilistic buffered Voronoi cells define the position regions of different safety levels that the ego robot can navigate safely. Each robot can compute its PBVCs without communication with other robots. The proposed heuristic efficient sampling approach also allows for real-time implementation of the algorithm. In simulations and experiments, we showed successfully that our approach achieves safe collision avoidance behaviors with only noisy measurements of other robots in challenging scenarios. The executed control input obeys safety assurance level and deviates minimally from desired input.

In future work, we will attempt to incorporate complex dynamics into this approach in order to allow for applications in more realistic settings. We also want to investigate applying this algorithm as a safety guarantee with other trajectory planning frameworks.

ACKNOWLEDGMENT

Toyota Research Institute (”TRI”) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

REFERENCES

[1] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, “Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, pp. 1047 – 1054, 2017.

[2] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, “Collision avoidance for multiple agent systems,” *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.

[3] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, 1992.

[4] S. Paternain, D. E. Koditschek, and A. Ribeiro, “Navigation functions for convex potentials in a space with convex obstacles,” *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 2944–2959, 2018.

[5] O. Arslan and D. E. Koditschek, “Sensor-based reactive navigation in unknown convex sphere worlds,” *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 196–223, 2019.

[6] V. Vasilopoulos, W. Vega-Brown, O. Arslan, N. Roy, and D. E. Koditschek, “Sensor-based reactive symbolic planning in partially known environments,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–5, IEEE, 2018.

[7] 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.

[8] 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*, pp. 1928–1935, IEEE, 2008.

[9] 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*, pp. 3475–3482, IEEE, 2011.

[10] 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.

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

[12] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, “Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5917–5922, IEEE, 2009.

[13] A. Best, S. Narang, and D. Manocha, “Real-time reciprocal collision avoidance with elliptical agents,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 298–305, IEEE, 2016.

[14] M. Wang, Z. Wang, S. Paudel, and M. Schwager, “Safe distributed lane change maneuvers for multiple autonomous vehicles using buffered input cells,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, IEEE, 2018.

[15] A. Pierson and D. Rus, “Distributed target tracking in cluttered environments with guaranteed collision avoidance,” in *2017 International Symposium on Multi-robot and Multi-agent Systems (MRS)*, pp. 83–89, IEEE, 2017.

[16] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, “Navigating congested environments with risk level sets,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, 2018.

[17] G. M. Hoffmann and C. J. Tomlin, “Decentralized cooperative collision avoidance for acceleration constrained vehicles,” in *2008 47th IEEE Conference on Decision and Control*, pp. 4357–4363, IEEE, 2008.

[18] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin, “A scalable framework for real-time multi-robot, multi-human collision avoidance,” *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[19] K. Leung, E. Schmerling, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone, “On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions,” in *International Symposium on Experimental Robotics*, 2018.

[20] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6252–6259, IEEE, 2018.

[21] Y. F. Chen, M. Liu, M. Everett, and J. P. How, “Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 285–292, IEEE, 2017.

[22] B. Kluge and E. Prassler, “Reflective navigation: Individual behaviors and group behaviors,” in *2004 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, pp. 4172–4177, IEEE, 2004.

[23] B. Kluge and E. Prassler, “Recursive probabilistic velocity obstacles for reflective navigation,” in *Field and Service Robotics*, pp. 71–79, Springer, 2003.

[24] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid,” in *2007 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1610–1616, IEEE, 2007.

[25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.