Buffered Uncertainty-Aware Voronoi Cells for Probabilistic Multi-Robot Collision and Deadlock Avoidance

Course Project: Step 3 / 3

Adi Amuzig

September 20, 2022

The Taub Faculty of Computer Science
Technion - Israel Institute of Technology

The Problem

We want to **avoid collisions** between robots as much as possible while employing an autonomous multi-robot system. Collisions might cause the agents' future moves to be delayed, damaged, or even render the mission impossible.

Even if we design a path for each robot that avoids collisions with other robots, each agent will still make errors in its motions, implying that **even if the plan avoids collisions, collisions may occur**.

Another problem that arises is the case of **deadlock**, where two or more agents stop stand still such that they will not collide, which can cause them to never reach their goal locations.

The challenge we're attempting to solve is **how to effectively plan paths and movements in a multi-robot system while accounting for the robots' faults in movement to minimize collisions and deadlock**.

Why is the Problem Interesting and Hard

The topic of multi-robot collision avoidance is essential for many systems, including the three-dimensional environment of a swarm of drones [1, 2] and the limited motions of autonomous cars [3, 4].

Avoiding collisions in a multi-robot system is problematic for various reasons: (1) we need to examine all robots' motions and trajectories simultaneously to avoid collisions, (2) we need a mechanism to quantify location uncertainty and (3) determine how to account for it in path planning, and (4) we strive to optimize the path for all agents to the objective.

What has been done before

We can utilize the reciprocal velocity obstacle (RVO) method [5, 6] when the states are completely known. Where collisions are calculated based on relative velocity. This approach has been developed in a variety of ways [7, 8, 9]. It had also been expanded to incorporate ambiguity about location. For example, a probabilistic RVO technique in which robot state uncertainty is considered to have a Gaussian distribution [10].

Another technique is to use a chance constrained nonlinear MPC (CCNMPC) [11] method to ensure that the likelihood of inter-robot collision is less than a certain threshold. A variant that combines buffered Voronoi cell (BVC) [12] with the threshold collision probability is Buffered Uncertainty-Aware Voronoi Cells (B-UAVC) [13], in which Voronoi edges are moved and buffed according to the robots' locations uncertainty while they move.

Suggested solution

We propose a modification to the B-UAVC approach. All robots will plan their course to the goal based on their future location uncertainty, as computed by a Kalman Filter [14], and the buffered uncertainty-aware Voronoi cells that will come from that location uncertainty.

If the agents reach a point where going straight toward the goal is not possible, then then each agent will attempt to preform a random movement. If after a few tries the agent haven't found a possible direction to travel in, then it will stent still for that turn.

Model

We model our problem using the tuple $\langle n, \mathcal{S}, \mathcal{G}, \mathcal{R}, \mathcal{P} \rangle$ where

- n is the number of robots on the two dimensional $\mathcal{R}^2$ space, where each robot is denoted with $i \in \{1, 2, ..., n\}$,
- $\mathcal{S} = \{s_i \in \mathcal{R}^2\}$ denotes the initial locations of the robots,
- $\mathcal{G} = \{g_i \in \mathcal{R}^2\}$ denotes the goal locations of the robots,
- $\mathcal{R} = \{r_i \in \mathcal{R}^2\}$ denotes each agent's safety radius, and
- $\mathcal{P} = \{P^j\} = \{p_i^j \in \mathcal{R}^2\}$ denotes the true position of the robots at each time j, and is described as a Gaussian distribution with mean μ_i^j and covariance Σ_i^j i.e. $p_i^j = \mathcal{N}(\mu_i^j, \Sigma_i^j)$.

Method

In general, we want to make cells with a low risk of collision if the agent stays inside them while moving toward the goal.

More specifically, our method is based on generating a path based on the robots' initial and goal positions prior to movement. In the planning phase, we have two actions to do for each time step j:

1. **Generate the B-UAVC** - Divide the space using basic Voronoi cells, then shift the edges of the cells using the p_i^j obtained from the KF (a larger uncertainty will push the edges of the cells away from the agent). Following that, we add a buffer to the edges according to r_i, followed by another layer of buffering according to p_i^j.
2. **Select a movement** - A movement is selected for each agent based on the possible actions the agent can preform. The agent can move only while remaining within its own Buffed Voronoi cell.

Method - Pseudo Code

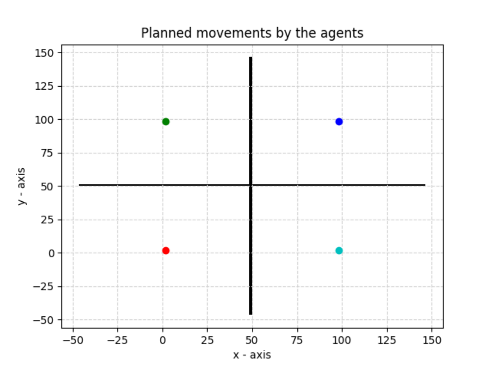
Algorithm 1 *B-UAVC Planner*

Input: Planner - used planner for the robots; $P^0 = \mathcal{S}$ - initial locations of the robots; $\mathcal{G}$ - goal locations of the robots; $\mathcal{R}$ - safety radiuses of the robots.

for j in pathTimeSteps **do**
    for i in n **do**
        $p_i^j \leftarrow \text{KalmanFilter}(p_i^{j-1}, Move_i^j)$
    $P^j \leftarrow [p_1^j, ..., p_n^j]$
    Vor $\leftarrow$ VoronoiCells(P^j)
    UAVor $\leftarrow$ UncertaintyAwareVoronoiCells(Vor, P^j)
    BUAVor $\leftarrow$ BuffedUncertaintyAwareVoronoiCells(UAVor, $\mathcal{R}$)
    for i in n **do**
        $Move_i^j \leftarrow$ Planner(i, BUAVor, $\mathcal{G}$)
        if LegalMove($Move_i^j$, BUAVor) $==$ FALSE **then**
            $Move_i^j \leftarrow$ RandomStep(i, BUAVor, $\mathcal{G}$)
            if LegalMove($Move_i^j$, BUAVor) $==$ FALSE **then**
                $Move_i^j \leftarrow$ NoMove(i, BUAVor, $\mathcal{G}$)
    $Movements^j \leftarrow [Move_1^j, ..., Move_n^j]$

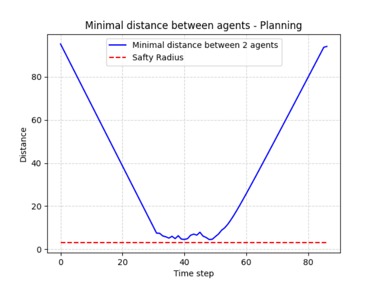
How I evaluated my approach

During the evaluation, I assigned multiple agents the task of exchanging positions. For example, in the diagram below, the red and blue agents are expected to switch places, as are the green and cyan, with the U-BVC in black. I compared the time it took for all agents to reach their objectives to the time it would have taken if collisions had not been accounted for, as well as the histogram of distance between the agents.

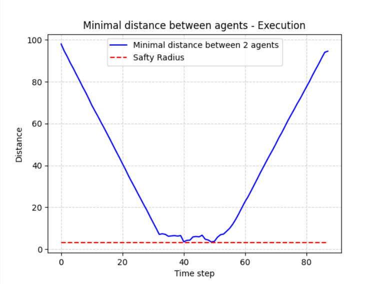


Evaluation Results - Distance Between Agents (1/2)

First, we needed to make sure that the original algorithm was still valid. That is, during the planned and executed paths, no agent crosses the determined safety radius, indicating that they keep a safe distance from one another. This was true of our new algorithm. Below is an illustration of the smallest distances between agents for the previous slide's scenario.



(a) Planned

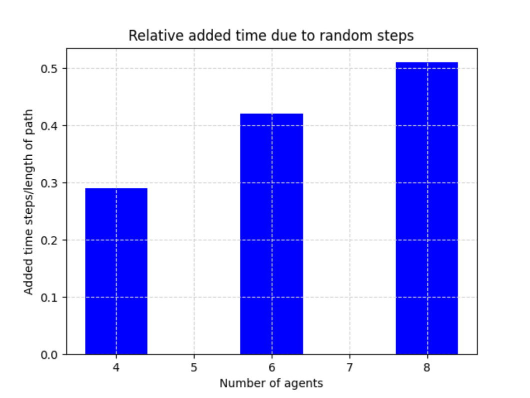


(b) Execution

Minimal distance between every 2 agents along the path

Evaluation Results - Relative Added Time Steps (2/2)

We can observe that as the number of agents increases, so does the proportionate number of additional time steps. Even if we simply utilize eight agents, over half of the time steps involve random steps to avoid collisions.



Conclusion

The proposed technique **improves** on the original method by preventing deadlock while still ensuring the safety radius between the agents.

The suggested technique includes **flaws**, such as the comparatively large number of time steps required for collision avoidance even with a small number of agents.

In future study, a mix of the two approaches can be developed such that the random actions are only done if the deadlock persists.

References i

Jawad Naveed Yasin, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila.
Formation maintenance and collision avoidance in a swarm of drones.
In *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*, pages 1–6, 2019.

Jawad Naveed Yasin, Sherif Abdelmonem Sayed Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, Muhammad Mehboob Yasin, and Juha Plosila.
Energy-efficient formation morphing for collision avoidance in a swarm of drones.
IEEE Access, 8:170681–170695, 2020.

Yew Cheong Hou, Khairul Salleh Mohamed Sahari, Leong Yeng Weng, Hong Kah Foo, Nur Aira Abd Rahman, Nurul Anis Atikah, and Raad Z Homod.
Development of collision avoidance system for multiple autonomous mobile robots.
International Journal of Advanced Robotic Systems, 17(4):1729881420923967, 2020.

Mingyu Wang, Zijian Wang, John Talbot, J Christian Gerdes, and Mac Schwager.
Game-theoretic planning for self-driving cars in multivehicle competitive scenarios.
IEEE Transactions on Robotics, 37(4):1313–1325, 2021.

Jur Van den Berg, Ming Lin, and Dinesh Manocha.
Reciprocal velocity obstacles for real-time multi-agent navigation.
In *2008 IEEE international conference on robotics and automation*, pages 1928–1935. Ieee, 2008.

Paolo Fiorini and Zvi Shiller.
Motion planning in dynamic environments using velocity obstacles.
The international journal of robotics research, 17(7):760–772, 1998.

Jur van den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha.
Reciprocal n-body collision avoidance.
In *Robotics research*, pages 3–19. Springer, 2011.

Daman Bareiss and Jur Van den Berg.
Generalized reciprocal collision avoidance.
The International Journal of Robotics Research, 34(12):1501–1514, 2015.

Javier Alonso-Mora, Paul Beardsley, and Roland Siegwart.
Cooperative collision avoidance for nonholonomic robots.
IEEE Transactions on Robotics, 34(2):404–420, 2018.

Bharath Gopalakrishnan, Arun Kumar Singh, Meha Kaushik, K Madhava Krishna, and Dinesh Manocha.
Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty.
In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1089–1096. IEEE, 2017.

Hai Zhu and Javier Alonso-Mora.
Chance-constrained collision avoidance for mavs in dynamic environments.
IEEE Robotics and Automation Letters, 4(2):776–783, 2019.

Dingjiang Zhou, Zijian Wang, Saptarshi Bandyopadhyay, and Mac Schwager.
Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells.
IEEE Robotics and Automation Letters, 2(2):1047–1054, 2017.

Hai Zhu and Javier Alonso-Mora.
B-uavc: Buffered uncertainty-aware voronoi cells for probabilistic multi-robot collision avoidance.
In *2019 international symposium on multi-robot and multi-agent systems (MRS)*, pages 162–168. IEEE, 2019.

R. E. Kalman.
A new approach to linear filtering and prediction problems.
Transactions of the ASME–Journal of Basic Engineering, pages 35–45, 1960.