Planning with Real-Time Collision Avoidance for Cooperating Agents under Rigid Body Constraints

Nicola Piccinelli
Dept. of Computer Science, Altair Lab.
University of Verona, Verona, Italy
nicola.piccinelli@univr.it

Federico Vesentini
Dept. of Computer Science, Altair Lab.
University of Verona, Verona, Italy
federico.vesentini@univr.it

Riccardo Muradore
Dept. of Computer Science, Altair Lab.
University of Verona, Verona, Italy
riccardo.muradore@univr.it

***Abstract*—In automated warehouses, path planning is a crucial topic to improve automation and efficiency. This kind of planning is usually computed off-line knowing the planimetry of the warehouse and the starting and target points of each agent. However, this global approach is not able to manage unexpected static/dynamic obstacles and other agents moving in the same area. For this reason in multi-robot systems global planners are usually integrated with local collision avoidance algorithms. In this paper we use the Voronoi diagram as global planner and the Velocity Obstacle (VO) method as collision avoidance algorithm. The goal of this paper is to extend such hybrid motion planner by enforcing mechanical constraints between agents in order to execute a task that cannot be performed by a single agent. We will focus on the cooperative task of carrying a payload, such as a bar. Two agents are constrained to move at the end points of the bar. We will improve the original algorithms by taking into account dynamically the constrained motion both at the global and at the collision avoidance level.**

***Index Terms*—Velocity Obstacle, Motion Planning, Collision Avoidance, Cooperative Robotics**

I. INTRODUCTION

Path planning for multi-agent systems is a critical aspect in automated warehouses where efficient delivery and storage are mandatory. Moreover mobile robot cooperation, instead of developing bigger vehicles, can effectively be exploited to delivery cumbersome loads. This cooperative approach increases system robustness, reconfigurability and fault-tolerant capability [10].

Path planning can be solved from two different points of view: local and global. Global planners provides optimal solution at the expense of efficient computational time which inhibits their use in real-time applications [8], [15]. On the other hand, local planning can be solved in efficient ways and can be used in real-time applications where promptness to react to environment changes and fast re-planning are important [3], [4], [6], [14].

Mobile robot cooperation deals with two different sub problems: formation control and cooperative path planning. Formation control aims to satisfy constraints on the relative position of the agents in a fleet [11] without any high-level decision-making architecture [10]. To account the formation control, many approaches are available in the literature and are classified as: leader-follower [12], virtual shapes [9] and behaviour-based approaches [2]. Cooperative path planning aims to provide optimised trajectories for the vehicles of the formation to accomplish a given task knowing the mission starting and target points, the environmental constraints and the required formation shape [7].

In this paper, a hybrid navigation system which uses the Voronoi Diagrams (VD) as a cooperative global planner and the Velocity Obstacle (VO) approach as local collision avoidance mechanism, such as in [13], will be used as a starting point. We will extend the Velocity Obstacle approach in order to handle rigid body constraints between agents during the local navigation and collision avoidance. Our solution is not a proper formation control due to the lack of re-shaping capability: we assume the payload to be rigidly connected to the agents.

The paper is organised as follows. In Section II we recall the Velocity Obstacle problem, the Optimal Reciprocal Collision Avoidance algorithm (ORCA) [16], the Voronoi Diagram and basic rigid body mechanics. In Section III we improve the local VO and the global planning VD to handle navigation formation and optimal path planning for coupled agents. Then in Section IV we explain the implementation details and we show the simulation results. Finally in Section V conclusions are drawn.

II. PROBLEM STATEMENT AND BACKGROUND

In this section, we will introduce the fundamental notions in order to explain how we will manage the problem of transporting a load; i.e. in a cooperative manner, from an initial pose towards a goal pose, and without colliding with other agents and/or environmental obstacles.

The robots/agents are assumed to be holonomic and a steel bar will be the load. The aim is to locate two agents at its end points, forming a system composed by two particles (the agents) under a rigidity constraint (the bar). The goal is to carry a particularly heavy or cumbersome payload exploiting the cooperation capabilities of more than one robot.

A. Voronoi Diagram

As defined in [1], given a finite and discrete set of points $S = \{s_1, s_2, ..., s_k\}$ of $E \subseteq \mathbb{R}^2$, the Voronoi Diagram of S is

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 779813 (SARAS project, www.saras-project.eu) and was partly supported by the project MIUR "Department of Excellence" 2018-2022.

a particular decomposition of E that associates a region *reg(s)* to every point $s \in S$, such that all the points into *reg(s)* are closer to s than any other point of the set S.

Formally, for a generic point $s \in S$ the related plane region is computed as:

$$reg(s) = \bigcap_{q \in S \setminus \{s\}} \{x \in E | ||x - s||_2 \le ||x - q||_2\}$$

where $|| \cdot ||_2$ is the Euclidean distance. VD are suitable for autonomous navigation because moving along the edges of a Voronoi Diagram implies that the robot is as far away as possible from the neighbouring obstacles [15].

B. Velocity Obstacle and ORCA

For a robot A moving in a 2D plane E that has to avoid collisions with another moving obstacle B, the Velocity Obstacle cone $VO^{\tau}_{A|B}$ is the set of velocities for A that will cause a collision with B within time τ. Thus collision avoidance implies that agent A selects as future velocity a vector $v_A^{new} \notin VO^{\tau}_{A|B}$.

When there are multiple obstacles (B_i, $i = 1, \dots, n$), agent A has to compute the following cone

$$VO = \bigcup_{i=1}^{n} VO^{\tau}_{A|B_i}.$$

Optimal Reciprocal Collision Avoidance (ORCA) is a numerical efficient approach to reciprocal n-body collision avoidance proposed in [16]. Suppose that two robots A and B move with current velocities v_A and v_B that will cause a collision within time τ, i.e. $v_{AB} = v_A - v_B \in VO^{\tau}_{A|B}$.

Let u be the vector from v_{AB} to the closest point of the boundary ∂VO of the Velocity Obstacle, i.e.

$$u = \underset{v \in \partial VO}{\operatorname{argmin}} ||v - v_{AB}|| - v_{AB}$$

and let η be the outward normal vector from the point $(v_{AB} + u) \in \partial VO$. The vector u can be interpreted as the minimal speed variation to be applied to v_{AB} in order to obtain $(v_{AB} + u) \notin VO^{\tau}_{A|B}$. The set of all feasible velocities for A is the half-plane $ORCA^{\tau}_{A|B}$ oriented accordingly to η and generated by the line parallel to the tangent of ∂VO at point $v_{AB} + u$ passing through $v_A + \frac{1}{2}u$.

Whenever A has to avoid collisions with a certain number n of other agents $B_1, ..., B_n$ within time τ, the overall ORCA

$$ORCA^{\tau}_A = D(0, v_A^{max}) \cap \bigcap_{i=1}^{n} ORCA^{\tau}_{A|B_i} \tag{1}$$

is a *convex set*. $D(0, v_A^{max})$ is a disc in the space of velocities representing the maximum speed constraint for robot A. Then

$$v_A^{new} = \underset{v \in ORCA^{\tau}_A}{\operatorname{argmin}} ||v - v_A^{pref}||_2$$

where v_A^{pref} is the preferred velocity of A, for example the one that allows to reach the target position in the shortest time. A will reach the new position $P_A^{new} = P_A + v_A^{new}\Delta t$ after Δt seconds.

C. Rigid body constraints

Let (x_A, y_A) be the position vector of agent A and (x_B, y_B) be the position of B in the Euclidean space $E \subseteq \mathbb{R}^2$.

The linked agents A and B form a system with three degrees of freedom: to know the exact configuration of the system, it is sufficient to know the position of one of the agents and the relative orientation angle $\theta \in [0, 2\pi)$ with respect to a fixed reference frame $\mathcal{O}_{x,y}$. The agents have to satisfy the following equations

$$\begin{cases} x_B &= x_A + d\cos(\theta) \\ y_B &= y_A + d\sin(\theta). \end{cases} \tag{2}$$

The velocity vectors $v_A = (\dot{x}_A, \dot{y}_A)$ and $v_B = (\dot{x}_B, \dot{y}_A)$ then satisfy the following equations

$$\begin{aligned} \dot{x}_B &= \dot{x}_A - d\omega\sin\theta \\ \dot{y}_B &= \dot{y}_A + d\omega\sin\theta \end{aligned}$$

where $\omega = \dfrac{d\theta}{dt}$ is the angular velocity.

III. PROPOSED METHOD

To take into account rigid body constraints during collision avoidance and path planning manoeuvres, the following subsections will introduce an adaption of [13] to plan optimal path with rigidly coupled agents and an extension of ORCA for handling non-linear optimisation problems (as it occurs in the present setup).

A. Voronoi Diagrams and rigid body constraints

As in [13], we use Voronoi Diagrams as global planner as well. In our case we have to include some improvements to carry out the planning for constrained agents.

We start by planning the optimal trajectory for a single agent, obtaining the waypoints P^i for $i = 1, ..., m$. For every index i, the segment connecting the waypoints P^i and P^{i+1} is given by $L^i = P^{i+1} - P^i$. Then for every P^i like in [5], we define the waypoints for the constrained system as follows

$$P_A^i = P^i, \tag{3}$$

$$P_B^i = L^{k_{min}} \cap \mathcal{C}(P_A^i, d), \tag{4}$$

where $i = 1, \dots, m$, $\mathcal{C}(P_A^i, d)$ is the circle of radius d centred in P_A^i, and k_{min} is given by $k_{min} = \min\{k \in [i, \dots, m) | L^k \cap \mathcal{C}(P_A^i, d) \neq \{\emptyset\}\}$.

The index k_{min} identifies the index of the first segment L that has non-empty intersection with $\mathcal{C}(P_A^i, d)$. Equation (4) might result to be not satisfied when the payload is approaching its goal position, hence it might happen that, in a certain waypoint P_A^j, there is no k_{min} for which equation (4) is satisfied. This means that the global planner is not able to set the corresponding P_B^j. In that case we have to change our strategy in order to select a proper waypoint for agent B. If all the waypoints P_A^i after P_A^j are such that $||P_A^j - P_A^i||_2 \le d, \quad i = j+1, \dots, m$ (i.e. entirely contained into the circle $\mathcal{C}(P_A^j, d)$) then the global planner sets the positions P_A^i and P_B^i by considering again equations (3) and (4) but switching the subscript indices A and B. This means that the circle will be centred in P_B^i, not in P_A^i. This allows us to stay on the optimal path planned in the beginning.

Figure 1 shows an example of planning: waypoint P_B^{52} was P_A^{52} in origin, but it failed to satisfy equation (4) so it is necessary to rename it as P_B^{52} by swapping subscript indices and to use it as centre of $\mathcal{C}(P_B^{52}, d)$, in order to set P_A^{52}. Then move on to P_B^{53} that allows to position P_A^{53} and the goal is reached.

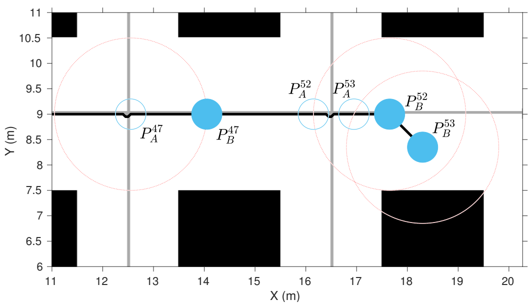

Figure 1. An example of planning for constrained agents A (empty blue circle) and B (filled blue circle). The dotted red circles shows the position constraints between agents for each waypoint.

B. ORCA and rigid body constraints

In order to make ORCA algorithm able to manage collision avoidance for agents rigidly linked, we modified the constraints formulation in the linear programming section of the code for computing the half planes.

First, we "group" the two robots A and B in order to constitute a single entity. Second, we run the "classic" ORCA algorithm to isolate the sets of all the collision avoidance velocities $\overline{ORCA}_A^\tau$ and $\overline{ORCA}_B^\tau$ with respect to any other agent or moving obstacle that doesn't belong to the constrained pair $A - B$. We have

$$\overline{ORCA}_\star^\tau = D(0, v_\star^{max}) \cap \bigcap_{C \neq \{A,B\}} ORCA_{\star|C}^\tau$$

where $\star = \{A, B\}$. Third, we update the new collision avoidance velocities v_A^{new} and v_B^{new} in such a way they satisfy

$$v_A^{new} \in \overline{ORCA}_A^\tau \cap RB, \quad v_B^{new} \in \overline{ORCA}_B^\tau \cap RB$$

where RB is the set $RB = \{(v_A^{new}, v_B^{new}) : ||p_A^{new} - p_B^{new}||_2 - d = 0\}$ with $p_A^{new} = p_A + v_A^{new}\Delta t$ and $p_B^{new} = p_B + v_B^{new}\Delta t$.

In other words, among the all possible collision avoidance velocities for A and B, i.e. those belonging to $\overline{ORCA}_A^\tau$ and $\overline{ORCA}_B^\tau$, robots A and B have to choose velocities v_A^{new} and v_B^{new} that not only guarantee no collisions within time τ, but that preserve the rigid link between them as well. Any other agent C sharing the same workspace avoids collisions accordingly to the original ORCA algorithm.

C. Non-linear programming solver

To obtain the velocities for A and B that satisfy the mechanical constraint and drive the robots toward their final targets, we adopt a non-linear programming solver based on the Augmentend Lagrangian Method (ALM). ALM is a class of methods developed to solve (generally non-linear) constrained optimisation problems. The problem that we want to solve can be formalised as

$$\underset{v_A^{new}, v_B^{new}}{\text{argmin}} \quad ||v_A^{new} - v_A^{pref}||_2 + ||v_B^{new} - v_B^{pref}||_2 \tag{5}$$

$$\text{subject to} \quad ||p_A^{new} - p_B^{new}||_2 - d = 0 \tag{6}$$

$$f_i(v_A^{new}) \leq b_i \quad i = 1, \dots N \tag{7}$$

$$f_j(v_B^{new}) \leq b_j \quad j = 1, \dots M \tag{8}$$

where p_A^{new} and p_B^{new} are the optimal new positions, v_A^{new} and v_B^{new} the optimal new velocities of the constrained agents, v_A^{pref} and v_B^{pref} are the velocities that allow the constrained agents to reach their next waypoints, $d > 0$ is the length of the rigid link between A and B, $f_i(\cdot) \leq b_i$ and $f_j(\cdot) \leq b_j$ are the affine functions defining the $ORCA_{A|C}^\tau$ half-planes, $N = |\mathcal{C}_N \setminus \{B\}|$ and $M = |\mathcal{C}_M \setminus \{A\}|$ where $\mathcal{C}_N$ and $\mathcal{C}_M$ are the set of the agents detected through their sensing systems (e.g. laser scanners, cameras, LIDAR's, etc) by A and B, respectively.

IV. Simulation results

The proposed solutions has been tested in simulation using ROS Kinetic (Robot Operating System), RVO2 library and Alglib 3.14.0 for solving the non-linear optimisation problem. The time step used is of $\Delta t = 50.0\,\text{ms}$, the ALM penalty coefficient and the number of outer iteration for the whole experiments are set to 10 and 500, respectively.

During the simulation tests, the agents (abstracted as circles) are considered for simplicity with the same maximum speed $v_{max} = 0.5\,\text{m/s}$ and bounding circle radius of $0.3\,\text{cm}$. We tested out our solution in a simulated warehouse environment with corridor $2.5\,\text{m}$ wide and with a rigid body constraints of length $d = 1.5\,\text{m}$. Three different scenarios are implemented to validate the proposed approach: Constrained motion, Constrained motion and obstacle, Constrained motion and agent.

A. Constrained motion

In the first scenario we assume two agents A, B moving in a completely known and static environment. They are rigidly coupled with a bar and should move along a pre-computed path as shown in Figure 2. The optimal path is the shortest part from P_{start} to P_{target} over the VD edges which satisfies the rigid constraint. In our case we computed collision free trajectories for A and B with a maximum error smaller than $\sim 4.4\,\text{mm}$.

B. Constrained motion and obstacle

In the second scenario we tested our solution with an obstacle placed over the precomputed paths for both agents A and B. We set a rectangular box of size $1.0\,\text{m} \times 0.5\,\text{m}$ and set the time horizon τ to $1.0\,\text{sec}$. Figure 3 shows the avoiding manoeuvres between the instants 18 and 23. The maximum position error is $5.7\,\text{mm}$.

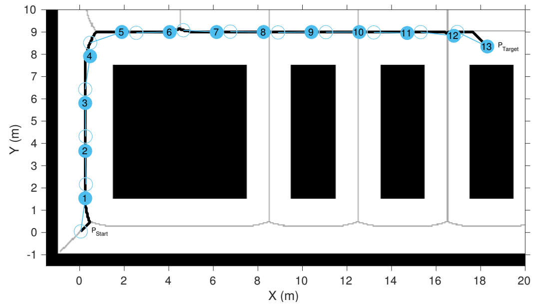

Figure 2. The trajectory for the two coupled agents (agent A empty blue circles, agent B filled blue circles). The thick black line shows the reference path.

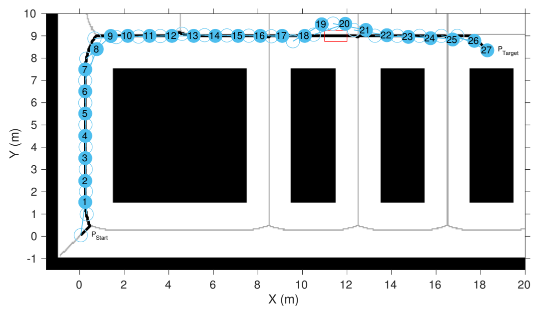

Figure 3. The trajectory for the coupled agents (agent A empty blue circles, agent B filled blue circles). The thick black line shows the reference path. The red rectangles is an unaccounted obstacle over the precomputed trajectory.

C. Constrained motion and agent

In the last scenario we have the coupled agents A and B moving over the precomputed trajectory from P_{start} towards P_{target} following a set of waypoints, and a third robot C is moving in a way of causing a collision. As shown in Figure 4 around discrete instants 7-9, the nominal trajectories of the agents cross, leading to a possible collision. As expected the agents A-B and C made successfully collision avoidance manoeuvres. It is worth highlighting that agent C treats the coupled agents as a single entity and senses the bar as an obstacle. This means that it cannot pass through the payload (constraint) during the collision avoidance manoeuvre. As in the previous scenarios the maximum position error is smaller than a centimetre $1.1\,\mathrm{mm}$ (with respect to the bar length $d = 1.5$m).

V. Conclusion

In this paper we presented an extension of ORCA in order to have collision avoidance when agents are rigidly coupled to execute a cooperative task (like carry a cumbersome load). For the coupled agents we added an extra non-linear constraint to the ORCA method to reduce the set of available velocities to the ones which also satisfies the rigid body dynamics. Moreover we introduced a strategy to plan the optimal path for the constrained agents using Voronoi diagrams. We tested out our solution in simulation in three different scenarios verifying our method against obstacles and other agents.

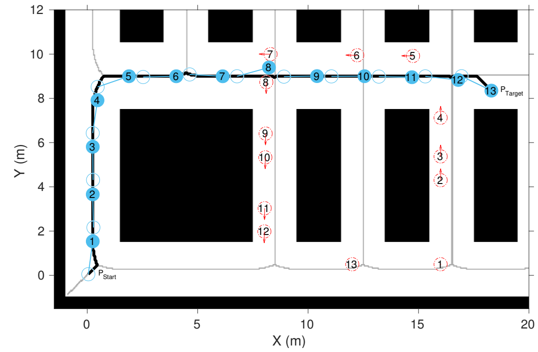

Figure 4. The trajectory for two coupled agents (agent A empty blue circles, agent B filled blue circles), and an extra agent C which is free to move (red circle). The thick black line shows the reference path.

References

[1] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[2] Tucker Balch and Ronald C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[3] Johann Borenstein and Yoram Koren. The Vector Field Histogram—Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.

[4] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, 1998.

[5] Mark Foskey, Maxim Garber, Ming C. Lin, and Dinesh Manocha. A voronoi-based hybrid motion planner. *IEEE International Conference on Intelligent Robots and Systems*, 1:55–60, 2001.

[6] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, 1997.

[7] Javier V. Gómez, Alejandro Lumbier, Santiago Garrido, and Luis Moreno. Planning robot formations with fast marching square including uncertainty conditions. *Robotics and Autonomous Systems*, 61(2):137–152, 2013.

[8] J.J. Kuffner and S.M. LaValle. RRT-connect: An efficient approach to single-query path planning. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2:995–1001, 2000.

[9] M. Anthony Lewis and Kar-Han Tan. High Precision Formation Control of Mobile Robots Using Virtual Structures. *Autonomous Robots*, 4(4):387–403, 1997.

[10] Yuanchang Liu and Richard Bucknall. A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica*, 36(7):1019–1047, 2018.

[11] Kwang Kyo Oh, Myoung Chul Park, and Hyo Sung Ahn. A survey of multi-agent formation control, 2015.

[12] Zhaoxia Peng, Guoguang Wen, Ahmed Rahmani, and Yongguang Yu. Leader-follower formation control of nonholonomic mobile robots based on a bioinspired neurodynamic based approach. *Robotics and Autonomous Systems*, 61(9):988–996, 2013.

[13] Nicola Piccinelli and Riccardo Muradore. Hybrid Motion Planner Integrating Global Voronoi Diagrams and Local Velocity Obstacle Method. In *Proceedings of 16th European Control Conference*, pages 26–31, Limassol, Cyprus, 2018.

[14] Andrey V. Savkin and Michael Hoy. Reactive and the shortest path navigation of a wheeled mobile robot in cluttered environments. *Robotica*, 31(2):323–330, 2013.

[15] Osamu Takahashi and R. J. Schilling. Motion Planning in a Plane Using Generalized Voronoi Diagrams. *IEEE Transactions on Robotics and Automation*, 5(2):143–150, 1989.

[16] Jur Van Den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. *Springer Tracts in Advanced Robotics*, 70(STAR):3–19, 2011.