Safe Distributed Control for Human-Multi-Robot Swarming using Voronoi Partioning

Sylvain Bertrand, Alexandre Eudes, Julien Marzat, Ioannis Sarras
DTIS, ONERA, Université Paris Saclay
F-91123 Palaiseau, France
{firstname.lastname}@onera.fr

***Abstract*—We propose a novel distributed algorithm to control a swarm of unmanned ground vehicles interacting with human operators in presence of obstacles. Each vehicle calculates distributively and dynamically its proper safety zone in which it generates a reference point to be tracked. The main feature of the algorithm is that it is based on purely geometric reasoning through the use of Voronoi partitioning and collision cones, which allows to naturally account for inter-robot, human-robot and robot-obstacle interactions. The effectiveness of the algorithm is illustrated by outdoor field experiments.**

***Index Terms*—Multi-Robot Swarms, Human-Robot Interactions, Distributed Control**

ACKNOWLEDGEMENT

This project has received funding from the European Union's Preparatory Action for Defence Research – PADR programme under grant agreement No 883465 – ARTUS.

I. INTRODUCTION

The deployment of fleets of unmanned vehicles for both civil and defense missions has radically increased in the last years. Recent progresses in vision or laser-based localization and mapping, along with the increase in embedded computational power, have led to the development of mobile and aerial robots of reduced dimensions allowing larger fleets of robotic vehicles to effectively undertake such missions under realistic environmental and communication conditions. The interaction with humans and obstacles or the practical limitations of inter-vehicle communication data links still pose serious challenges that need to be consistently addressed for field deployment of teams of autonomous robots. This requires the synthesis of distributed control algorithms with increased capabilities in terms of autonomy, safety and resilience. Several paradigms have been proposed for distributed fleet control [1]–[3] such as: leader-following, behavioral rules, virtual structure, artificial potential function, graph-rigidity. Most of these methods require that the geometric formation of the fleet is (quasi-)explicitly defined through fixed, desired relative positions or distances to be attained [4], [5], and they cannot incorporate naturally the interactions with a human, be it a pedestrian that has to be avoided or an operator that has to be followed at a distance.

A less rigid behavior, more suitable to human-robot interactions, can be achieved by partioning the space for motion coordination. This is the case of Voronoi diagrams which have been mostly used for allocation and coverage tasks [6]–[8], but also to the fleet navigation problem considered in [9], [10]. These algorithms rely on user-defined navigation functions to compute the centroid of the Voronoi cell of each agent, which is used as the reference position to be tracked by the robots, and collision avoidance is handled directly by the space partioning. An improved version relying on geometric constraints has been proposed by the authors in [11], with a more intuitive and direct management of fleet navigation and collision avoidance behaviors between the vehicles. The work presented in this paper is an extension of the latter method, with the inclusion of obstacle avoidance and interaction with a human operator. Extending the earlier concept of swarm teleoperation by a human operator [12], more advanced interactions can be integrated in control algorithms for human-multi-robot swarming. Different types of interactions can be considered depending on information flows available between the robots and human operators (one-way / two-ways) and the nature of the interactions themselves (physical / non-physical), see e.g. [13] for a good overview. Two levels of interactions are considered in this paper, which allow the operators to be included in a swarm of autonomous vehicles and possibly impose their velocity to the robots for coordinated safe motion.

The problem definition and proposed swarm control method are described in Section II, and field experiments[1] with two mobile robots and a localized human operator are reported in Section III.

II. SWARM CONTROL METHOD

A. Problem definition

The problem under consideration consists in driving a swarm of N unmanned ground vehicles (UGVs) to a waypoint, denoted by $p^* \in \mathbb{R}^2$, and by extension to reach successive waypoints on a given path in a cluttered environment with no prior map available. A typical applicative context is search-and-rescue or tactical missions, where human operators are assisted by a swarm of autonomous robots for transportation of critical resources, wounded persons or communication link maintenance. A fully autonomous behavior is expected

[1] Video available at https://tinyurl.com/OneraHumanRobotSwarm

from the swarm, achieving a requested mode and enforcing safety constraints, with respect to the presence of obstacles and humans, while ensuring an automatic reconfiguration in case of vehicle loss(es). It is assumed that each vehicle is able to estimate its own position with respect to a common global fixed reference frame, into which p^* is defined, and to broadcast it to all other vehicles within communication range. The position of Robot i will be denoted by $p_i \in \mathbb{R}^2$ and the set of its neighbor robot positions by $\mathcal{N}_i = \{p_j \,|\, j = 1..N, j \neq i, \|p_i - p_j\| \leq r_{com}\}$ where $r_{com} > 0$ is the communication range assumed to be constant. The number of neighbors of Robot i will be referred to as $N_i = Card\,\{\mathcal{N}_i\}$.

The human operators are equipped with equivalent localization devices, and are considered as additional vehicles with no control input computed. Two main levels of interaction between the swarm and a human operator have been studied:

- An *Autonomous* mode, in which the swarm has to follow autonomously a predefined path at a given nominal speed reference.
- A *Velocity-Guided* mode, where the swarm follows a predefined path at the same speed as a human operator.

In both cases, all the robots should remain at a desired safety distance from any human operator, other robot and obstacle.

B. Algorithm description

The main idea of the proposed distributed algorithm is that each vehicle computes online a Voronoi partition of the space involving other physical agents (other vehicles, human operators), and virtual (mirror) agents that are added to maintain the coherence of the swarm. A reference position to be tracked by a lower-level controller is then computed by each robot inside its own Voronoi cell. A geometric approach has been preferred for this calculation, which is done by considering lines of sight between the vehicle, the waypoint (for attraction), the boundaries of possible obstacles (for avoidance), and other UGVs or human operators (for collision avoidance). The algorithm allows to obtain different behaviors/patterns (e.g. side-by-side, group, convoy-like) by only modifying the initial relative placement of the vehicles. Finally, the distributed nature of the algorithm also grants robustness to online modification (removing or adding) of the number of entities (robots, humans) in the swarm, while also addressing the mono-robot and 2-robot scenarios.

The main steps of the algorithm are the following.

Step 1: Voronoi partitioning. Each robot i computes a Voronoi partition accounting for the other real agents in the swarm and virtual mirror agents (see left part of Fig. 1). The mirror agents are introduced as a means to guarantee the feasibility of the computation of the Voronoi partition, especially with one and two robots, and to adjust the size and bound of each robot's Voronoi cell. For each real agent located at a distance lower than a predefined threshold, a spacer segment is inserted between the robot and this agent to modify the construction of the Voronoi cell (see right part of Fig. 1). More precisely, the size of the Voronoi cell is reduced in the direction of the other agent(s) with collision risk in order to disable the placement in this area of the reference point to be tracked by the robot.

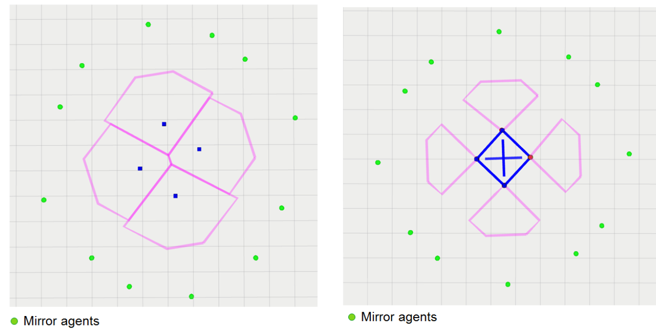

Fig. 1. Voronoi diagram for a swarm with four agents: without spacer segments (left) or with spacer segments (right) to enforce anti-collision between agents

Step 2: Attraction to waypoint. An attraction point P_a^i is defined inside the Voronoi cell C_i of the vehicle i, on the segment directed along the line of sight between the vehicle and the waypoint WP, and limited inside the Voronoi cell C_i (see Fig. 2).

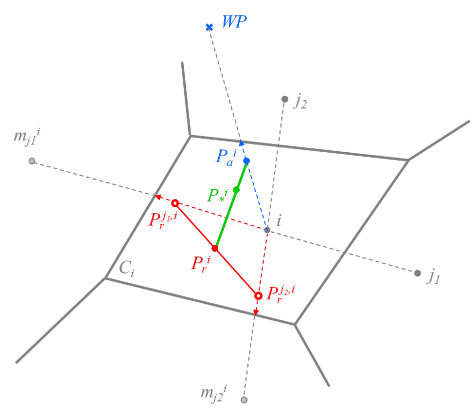

Fig. 2. Illustration of computation by robot i of its reference position P_*^i to be tracked. Case with three robots (i, j_1, j_2) and two mirror agents $m_{j_1}^i$ and $m_{j_2}^i$ added to bound the Voronoi cell.

Step 3: Obstacle avoidance. The distances between the vehicle i and the detected obstacles are evaluated to identify obstacles in proximity which need to be checked for collision risk. A map of obstacles is built online from range measurements provided by onboard sensors of the UGV. For collision risk evaluation, a cylinder model of obstacles (including a safety margin) is considered. A first step then consists in computing a cone englobing each obstacle, with the robot's position as vertex. A test is then realized to check whether the line of sight between the robot and the waypoint intersects at least one of these cones:

- If not, there is no collision risk with any of the obstacles, and direct straight motion to the waypoint is safe for the robot. The attraction point P_a^i computed at Step 2 is still valid and the algorithm proceeds to the next step.

- If there is at least one obstacle with collision risk, the cone of this obstacle is considered. It is enlarged step by step by considering adjacent and intersecting cones related to other obstacles, so as to obtain a larger cone containing a cluster of the obstacles with collision risk (see example on Fig. 3). The same procedure is repeated to build another cone, but this time by considering the waypoint as vertex. The two intersection points between these bounding cones are then computed. They correspond to two intermediate target points for the robot, each of them defining a possible obstacle-free path towards the waypoint. Some heuristics are used at this stage to select the shortest among the two available paths. The target point corresponding to the selected path is considered, instead of the waypoint, to compute a new attraction point P_a^i , in the same way as in Step 2. This new attraction point replaces the one computed in Step 2 and is used instead for the rest of the algorithm.

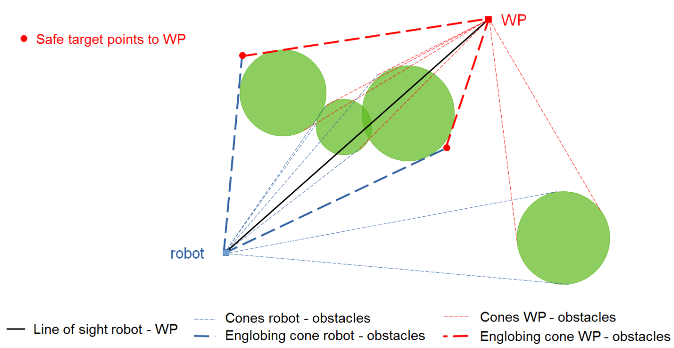

Fig. 3. Example of collision cones and computation of a safe path towards a given waypoint in case of collision risk with obstacles.

Step 4: Collision avoidance with other agents. For each other real agent j (either UGV or human operator) with collision risk (distance criterion), a repulsion point $P_r^{j,i}$ is defined inside the Voronoi cell of the vehicle i, on the segment directed along the line of sight between agent j and agent i, and limited to the Voronoi cell C_i (see Fig. 2). A repulsion point P_r^i is then computed as the mean of all the $P_r^{j,i}$. This additional repulsion term is especially important to avoid collisions with human operators as their motion may not comply with the Voronoi partitioning constraints. The localized agents are masked in the static obstacle map to prevent unnecessary maneuvers.

Step 5a: Computation of reference. The reference position P_*^i that will be tracked by robot i is computed as a weighted mean of the attraction point P_a^i and the repulsion point P_r^i,

$$P_*^i = (1-\beta)P_r^i + \beta P_a^i, \quad (1)$$

where the weighting coefficient $0 \leq \beta \leq 1$ is adapted online depending on the minimum distance to other agents with collision risk. It enables to give more weight on repulsion if some UGVs are very close or more weight on attraction to the waypoint otherwise. If there are no collision risks between the agents ($\beta = 1$), this algorithm results in $P_*^i = P_a^i$, leading to pure attraction to the waypoint.

Step 5b: Velocity modulation. In addition to the computation of the reference point to be tracked by the robot, a speed ratio is also produced by the algorithm. This speed ratio multiplies the speed value that has been chosen initially by the user, so as to provide the current reference speed to the low-level controller. In the *Autonomous* mode, the speed ratio is set to 0.5 in case of collision risk (slow motion in presence of obstacles) and to 1 otherwise (full-speed motion). In the *Velocity-Guided* mode, the speed ratio is set to copy the speed of the human operator, considered as command for the swarm, with a saturation at 1, which means the operator can go faster than the robot's maximal reference speed. In case of collision risk, the robot's speed ratio is saturated at 0.5, while still copying the operator's speed below this value.

III. Field Experiments

Fig. 4. Field experiments. Top: View of testing area with UGVs and obstacles. Bottom: UGVs and operator equipped with portable localization kit.

Experiments were carried out with the following robotic platforms (see Fig. 4):

- 2 UGVs: Robotnik Summit XL, 50 kg, basis 72x61 cm, with localization sensors (stereovision and IMU) and an embedded Intel-NUC CPU.
- 1 operator Portable Localization Kit with localization sensors (stereovision and IMU) and a dedicated embedded Intel-NUC CPU.
- A standard WiFi network connecting all the embedded computers and a ground station for mission supervision.

The localization of the UGVs and the operator were achieved using calibrated stereo cameras and a visual odometry algorithm [14], with a common AprilTag reference acting as the global reference origin (Fig. 4 bottom). One UGV was equipped with a Velodyne 3D LiDAR sensor, while the other UGV used the stereo images to build a real-time equivalent occupancy grid of obstacles. Two experimental results are reported to illustrate the proposed method.

1) UGV–Operator interaction in Autonomous mode: In this test, the human operator is not considered as an obstacle but as an additional agent of the swarm in the Voronoi partition, while the UGVs follow the path autonomously. This test illustrates the way the localized operator can influence the other robots, where in Fig. 5 the operator walks between the two robots and forces them to make room: the red robot stops and the green one is forced to deviate from its trajectory.

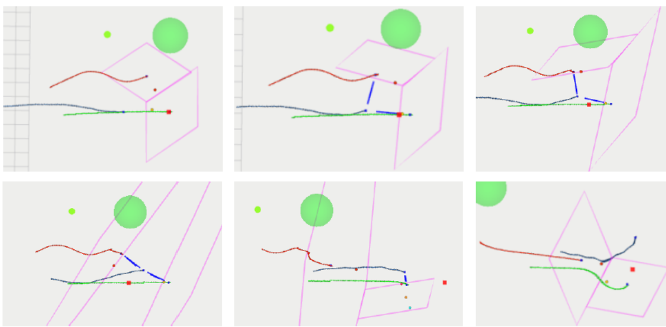

Fig. 5. Autonomous mode with Human Operator interaction. Each vignette shows a situation in time. The time flows from left to right on the top then the bottom rows, where the Voronoi cell of each robot is represented in magenta, the trajectory of each robot is in red and green, the operator is in blue.

2) Velocity-Guided mode: In this test, the robots follow a 8-shaped trajectory (Fig. 6). It can be checked that both robots adapt their velocity to the one of the localized human operator, and that safety distances are respected at all times. In particular, it can be seen that the velocities of the robots and the operator are correctly superposed, except when the operator goes faster than the UGV nominal velocity (e.g., a little after time 80s) or when an obstacle is along the way (at the end of the trajectory). These behaviors are fully consistent with the distributed algorithm and the imposed requirements.

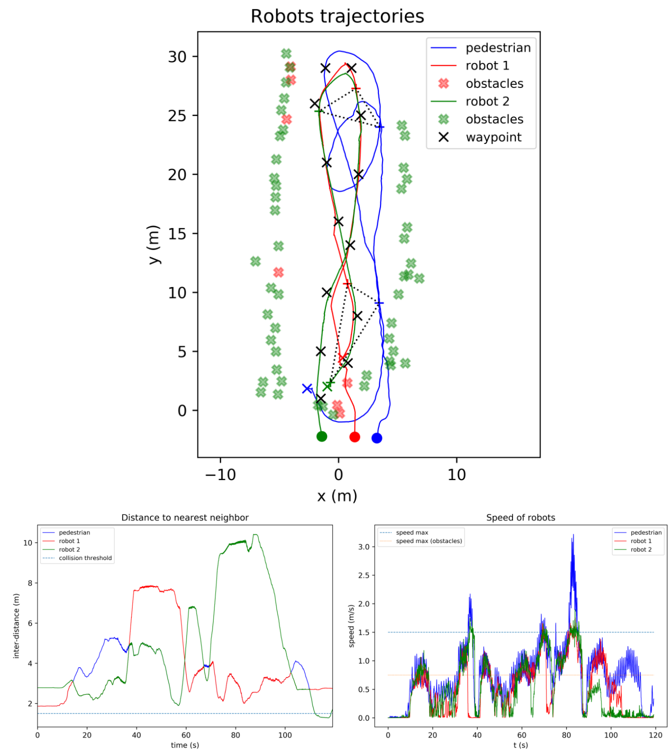

Fig. 6. Velocity-Guided mode for two UGVs interacting with a localized human operator. Top: Trajectories followed by the agents. Bottom left: inter-distances between robots and with operator (safety is always ensured). Bottom right: superposition of robots and operator velocities.

IV. Conclusions and Perspectives

A distributed control algorithm based on Voronoi partitioning and collision cones has been proposed to coordinate the navigation of a swarm of unmanned ground vehicles interacting with a localized human operator in cluttered environments. More elaborate interaction modes are foreseen for future work (e.g., formation split-and-merge or adaptation to multiple operators), as well as larger-scale experiments.

References

[1] R. M. Murray, "Recent research in cooperative control of multivehicle systems," Journal of Dynamic Systems, Measurement, and Control, vol.129, no.5, pp.571-583, 2007.

[2] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multi-agent Networks,* Princeton University Press, 2010.

[3] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications,* Springer, 2010.

[4] K-K. Oh, M-C. Park and H-S. Ahn, "A survey of multi-agent formation control", Automatica, vol.53, pp.424-440, 2015.

[5] K. Fathian, D. I. Rachinskii, M. W. Spong, T. H. Summers and N. R. Gans, "Distributed Formation Control via Mixed Barycentric Coordinate and Distance-Based Approach", American Control Conference, Philadelphia, PA, USA, 2019.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks", IEEE Transactions on Robotics and Automation, vol.20, no.2, pp.243–255, 2004.

[7] K. R. Guruprasad, and P. Dasgupta, "Voronoi Partitioning for Multi-Robot Systems with Limited Range Sensors", IEEE/RSJ IROS, Vilamoura, Portugal, 2012.

[8] J. Hatleskog, S. Olaru, and M. Hovd, "Voronoi-based deployment of multi-agent systems", IEEE Conference on Decision and Control, Miami, USA, pp. 5403-5408, 2018.

[9] M. Lindhé, P. Ogren, and K. H. Johansson, "Flocking with obstacle avoidance: A new distributed coordination algorithm based on Voronoi partitions", IEEE ICRA, Barcelona, Spain, 2005.

[10] Q. Jiang, "An Improved Algorithm for Coordination Control of Multi-Agent System Based on r-limited Voronoi Partitions", IEEE International Conference on Automation Science and Engineering, Shanghai, China, 2006.

[11] S. Bertrand, I. Sarras, A. Eudes and J. Marzat, "Voronoi-based Geometric Distributed Fleet Control of a Multi-Robot System", 16th ICARCV, pp. 85-91, 2020.

[12] M. D. Moniruzzaman, A. Rassau, D. Chai and S. M. Shamsul Islam, "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey", Robotics and Autonomous Systems, vol. 150, p. 103973, 2021.

[13] M. Aggravi, G. Sirignano, P. R. Giordano and C. Pacchierotti, "Decentralized Control of a Heterogeneous Human-Robot Team for Exploration and Patrolling", IEEE Transactions on Automation Science and Engineering, 2021.

[14] M. Sanfourche, V. Vittori, and G. Le Besnerais, "eVO: A realtime embedded stereo odometry for MAV applications," IEEE/RSJ IROS, Tokyo, Japan, pp. 2107–2114, 2013.