Socially-aware Multi-agent Velocity Obstacle based Navigation for Nonholonomic Vehicles

1st Manuel Boldrer
Dept. of Industrial Engineering
University of Trento
Trento, Italy
manuel.boldrer@unitn.it

2nd Luigi Palopoli
Dept. of Inf. Eng. and Comp. Science
University of Trento
Trento, Italy
luigi.palopoli@unitn.it

3rd Daniele Fontanelli
Dept. of Industrial Engineering
University of Trento
Trento, Italy
daniele.fontanelli@unitn.it

***Abstract*—We present an algorithm for collision free and socially-aware navigation of multiple robots in an environment shared with human beings, other robots and with the presence of static obstacles. We formulate the problem as a constrained optimization problem, where the cost function is chosen in order for the robotic agents to exhibit bio-inspired behaviors, such as cooperation inside the group and cohesive motion. Some of the constraints are required to avoid collision between the agents and with other obstacles and emanate from the application of Velocity Obstacle approach. The nonholonomic dynamics of the vehicles, is managed through the application of the feedback linearization technique to map the velocities in the control values. In this paper we propose both centralized solution and a completely decentralized solution. The overall strategies are extensively tested in simulations.**

***Index Terms*—Human behavior, velocity obstacle, distributed control and optimization, multi-agent systems.**

I. Introduction

Service robots moving in environments shared with humans are nowadays increasingly popular [1]. Examples range from the classic robot cleaners [2] to tour guiding systems [3] or personal robots [4, 5]. One of the main problems a robot moving in a space shared with human beings has to face is to exhibit a socially acceptable behavior by taking motion decision that do not intimidate the human bystanders [6]. In this paper, we raise the level of complexity requiring that the robots moving in the environment are themselves a group and need to behave as such. In this case, avoiding collisions with humans moving in a predictable way is not sufficient. The motion has to follow the patterns of a socially inspired group formation. In contrast to other existing solutions [7], this paper addresses the problem using an optimal reactive control law instead of motion planning approaches.

In the literature there are many algorithms for collision free navigation control of multiple agents. In [8] the authors classify the most important papers on the navigation of multi-agent systems in cluttered environments along the following lines: i) reactive methods, i.e., basically potential field based methods and reciprocal collision avoidance methods; ii) decentralized MPC based methods.

The algorithms based on these control approaches are quite often bio-inspired. In particular, animal aggregation examples such as flocking of birds, school of fish, ant colony or swarm of bacteria are based on local coordination rules. In nature, this kind of simple interactions lead to more complex and intelligent emergent behaviors, e.g., aggregation of the group and cooperation at the group level [9].

The Artificial Potential Function (APF) approaches are based on gradient descent methods built on a combination of attractive and repulsive potential fields. The main drawback of these methods is the presence of local minima, where the robot could be trapped. Potential fields are applied, to cite a few, in [10], where a theoretical approach to synthesize distributed flocking algorithms is presented, and in [11], where a geodesic control law is conceived to minimize a notion of misalignment energy related to the flocking problem for a group of nonholonomic agents. Furthermore, [12] adopts potential functions to preserve the connectivity of a multi-agent system network. In [13] the authors present a combination of potential like-functions to model the human motion in complex environments, while in [14] superimposition of potential fields and limit cycles are used to avoid human beings in a socially acceptable manner.

Distributed Model Predictive Control (DMPC) approaches are also widely adopted to deal with the problem of multi-agent coordination. The coupled constrained systems there defined lead to the adoption of basically two strategies: the sequential DMPC, where we have one-directional communication [15]; the iterative DMPC where the controllers negotiate the optimal solution and the communication is bi-directional [16].

The approaches that are closer to the one presented in this paper are the reciprocal collision avoidance (RCA) methods. The RCA methods manage collision avoidance between agents by assuming that each agent takes its share of responsibility in the avoiding maneuver. This model is inspired by ecological observations and was applied to human motion models [17]. In [18] the Velocity Obstacle concept introduced by [19] is extended to manage multi-agent interactions. This approach called Reciprocal Velocity Obstacles (RVO), partitions the responsibility for the avoidance maneuver equally between the agents. However, RVO exhibits an undesired oscillatory behavior in crowded environment conditions. Thus, in [20], the RVO is reformulated as Optimal Reciprocal Collision Avoidance (ORCA). These algorithms, first conceived to work

only for simple integrator models, have been recently extended to nonholonomic systems [21], e.g., car-like and unicycle-like models, by letting the robot to stay close to a holonomic trajectory with predefined bounded tracking errors. Finally, [22] generalizes the reciprocal collision avoidance to generic non-linear dynamics.

In this paper we propose an algorithm based on Velocity Obstacles for the navigation of a group of unicycles that can be generalized for the class of feedback-linearizable vehicles. The approach has been designed for nonholonomic vehicles since many observations reveal that nonholonomic motion is a typical human behavior in natural environments [23]. The core of the paper is the formulation of an optimization problem that captures both collision avoidance issues and the compatibility of the generated motion with the social rules. We discuss and compare both a centralized and a distributed solution for the problem. The approach is remarkably different from the MPC approaches since the high variability imposed by the presence of human beings in the area suggest a "pragmatic" local solution (whereas MPC seeks a globally optimal solution). The optimal local control problem is solved taking into account constraints conceived to mimic the human social behavior, such as a flexible group formation along the path to the desired group goal [24].

The paper is organized as follows. In Section II, background materials and the problem at hand are presented in details. The main results of the paper, including the optimal centralized and decentralized problems, are discussed in Section III, and in Section IV extensive simulation results are deeply analyzed in different scenarios. Section V presents the conclusions of the paper and discusses future research directions.

II. Problem description and background material

In this paper we are dealing with the synthesis of a social-group steering control that allows a team of n robotic agents to move across a cluttered environment (e.g. shared with human beings, other robots and static obstacles) from an initial set of known positions $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ to a final set $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ in a socially-aware manner, i.e. the robot motion has to be compatible with the presence of human beings and also has to be intuitive for them. The solution takes into account the limited maneuverability of the robotic platform. We rely on the availability of a path planner to plan offline opportunistic paths for each agent and then, to react to dynamic interactions, we use the quite known idea of velocity obstacles [19].

The dynamics of each agent is modeled as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix}, \tag{1}$$

where (x, y) is the position of the mid point of the rear axle in the plane of motion, θ is the vehicle yaw, v is the forward velocity, and ω is the angular velocity. The linear velocity v, and the angular velocity ω are the control inputs of the system.

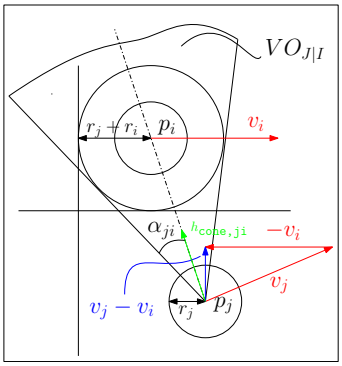

Figure 1. Construction of the set of forbidden velocities $VO_{J|I}$ for agent J. If $v_j \in VO_{J|I}$ a collision between the agents I and J will occur in a finite time.

In order to generate a path for the i–th agent connecting the starting position s_i to the final position e_i, any planning algorithm can be adopted. In this work, we have used the Rapidly-Exploring Random Tree star (RRT*), which is widely used for in both convex and non-convex environments [25]. The main advantages of this planner are its computation time efficiency and the capability of finding the optimal obstacle-free path. Moreover it is simple, easy to implement and converges asymptotically to the shortest path. The typical path generated by RRT* get close to obstacles obstructing the shortest path. As a consequence, during the path planning, the static obstacles are inflated to consider the agents footprint. In this work the RRT* is used as global path planning, i.e. it finds a path in the environment considering only static obstacles, while the presence of the other agents or of dynamic obstacles is dealt with the Velocity Obstacle.

A. Local path planning: Velocity Obstacle

The Velocity Obstacle [19] is a geometric algorithm used for collision avoidance. Since this solution follows the philosophy of local and reactive navigation, we use it for agent to agent and agent to static or dynamic obstacle collision avoidance.

Let I and J be two agents moving in the plane. With reference to (1), let $p_i = [x_i, y_i]^T$ and $p_j = [x_j, y_j]^T$ denote the positions of the agents I and J respectively and let $v_i = [\dot{x}_i, \dot{y}_i]^T$ and $v_j = [\dot{x}_j, \dot{y}_j]^T$ be the current velocities of the agents I and J respectively, as reported in [18] and shown in Figure 1. Considering the agents as disks of rays r_i and r_j, it follows that the cone $VO_{J|I}$ is the set of the velocities that lead to a collision, i.e., if the difference between v_j and v_i belongs to $VO_{J|I}$, then a collision will occur. More formally, let $I \oplus J = \{i + j \mid i \in I, j \in J\}$ be the Minkowski sum of the agents. Furthermore, let $\lambda(p, v) = \{p + tv \mid t > 0\}$, then

$$VO_{J|I} = \{v \mid \lambda(p_j, v - v_i) \cap I \oplus -J \neq 0\}. \tag{2}$$

Note that if the agent are disk-shaped, (2) simplifies to

$$VO_{J|I} = \{v \mid \exists\, t > 0 : t(v - v_i) \in D(p_i - p_j, r_i + r_j)\}, \tag{3}$$

where $D(p, r)$ is an open disk centered in p with radius r. Finally, by construction, the Velocity Obstacle is the same for

the two agents (i.e. symmetry property: $v_j - v_i \in VO_{J|I} \Leftrightarrow v_i - v_j \in VO_{I|J}$) and it is invariant to a common velocity change (i.e. translation invariance: $v_j - v_i \in VO_{J|I} \Leftrightarrow (v_j + \nu) - (v_i + \nu) \in VO_{J|I}$).

B. Feedback linearization description

The solution we are presenting in this paper relies on the assumption that the derivatives of the Cartesian coordinates (x, y) in (1) can be used as independent control inputs. Due to the nonholonomic constraint of the unicycle-like kinematic model (1), this is clearly not the case. A shortcut is to impose such velocities to another reference point of the vehicle that does not lay on the unicycle rear axle. Therefore, let

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} x + b\cos\theta \\ y + b\sin\theta \end{bmatrix}, \qquad b \neq 0. \tag{4}$$

be such a reference point, whose time derivative is

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix} = \begin{bmatrix} \dot{x} - b\dot{\theta}\sin\theta \\ \dot{y} + b\dot{\theta}\cos\theta \end{bmatrix}. \tag{5}$$

By substituting the model (1) in (5), one gets

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix} = \begin{bmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = T(b,\theta) \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{6}$$

Since $b \neq 0$, the matrix $T(b, \theta)$ is invertible and the vehicle control inputs can be uniquely derived as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = T(b,\theta)^{-1} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{b}\sin\theta & \frac{1}{b}\cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix}.$$

III. Optimization problem

The problem of the navigation without collision of multiple agents in a dynamic environment can be seen as an optimization problem. In particular we want to minimize a cost function defined as

$$H = \sum_{i=1}^{n} \|v_i^D - v_i\|, \tag{7}$$

where $n \in \mathbb{R}$ is the number of agents, $v_i^D \in \mathbb{R}^2$ is the vector of the planar desired velocity, i.e. the one constrained by the RRT*, and $v_i \in \mathbb{R}^2$ is the actual velocity decision variable minimizing (7). Let us denote with h_i^D the heading of the desired velocity of the i–th agent along its reference path. Each RRT* defined path is discretized in waypoints. The closest waypoint to the i–th agent is named wp_i^k, while the active waypoint $wp_i^{k+l} = [wp_{i,x}^{k+l}, wp_{i,y}^{k+l}]^T$ is a point ahead of a preview length l. It follows that the desired velocity $v_i^D = \|v_i^D\| h_i^D$, where the module $\|v_i^D\|$ is arbitrarily defined and $h_i^D = [wp_{i,x}^{k+l} - x_i, wp_{i,y}^{k+l} - y_i]^T$. Ideally, if we minimize the cost function (7) the agents will reach the goal position following a path of minimum length. However, there are constraints that have to be satisfied in order to avoid collisions. These constraints are defined by the Velocity Obstacle cones in (3). More precisely, the cones define the regions in the velocity space that lead to a collision in finite time, hence the constraints for the i–th agent can be formulated as

$$\bigcup_{j \in \mathcal{V}_i} \left\langle \frac{v_i - v_j}{\|v_i - v_j\|}, h_{\text{cone},ij} \right\rangle < \cos\alpha_{ij}, \tag{8}$$

$$\bigcup_{o_p \in \mathcal{V}_i} \left\langle \frac{v_i - v_{o_p}}{\|v_i - v_{o_p}\|}, h_{\text{cone},io_p} \right\rangle < \cos\alpha_{io_p}, \tag{9}$$

where $\mathcal{V}_i$ is the visibility set of the i–th agent, comprising the indices of the other agents and of the n_o static or moving obstacles that are closer to the i–th agent less than a threshold value ρ. In particular, Equation (8) defines the set of feasible velocities v_i ensuring no collision between the i–th agent and all the other agents in the group. The expressions for the heading of the cone $h_{\text{cone},ij}$ and its semi-aperture α_{ij} are

$$\begin{aligned} h_{\text{cone},ij} &= \frac{p_j - p_i}{\|p_j - p_i\|}, \\ \alpha_{ij} &= \arctan\left(R_{ij} / \sqrt{\left|(\|p_j - p_i\|)^2 - R_{ij}^2\right|} \right), \end{aligned} \tag{10}$$

where $R_{ij} = r_i + r_j$. Similarly, Equation (9) considers the collision with static and dynamic obstacles, whose cone and semi-aperture are given by (10) substituting j with o_p.

As a consequence, the optimization problem can be written in the compact form

$$\begin{aligned} \arg\min_{\mathcal{W}} \quad & \sum_{i=1}^{n} \|v_i^D - v_i\| \\ s.t. \quad & \forall i = 1, \ldots, n \\ & \bigcup_{j \in \mathcal{V}_i} \left\langle \frac{v_i - v_j}{\|v_i - v_j\|}, h_{\text{cone},ij} \right\rangle < \cos\alpha_{ij}, \\ & \bigcup_{o_p \in \mathcal{V}_i} \left\langle \frac{v_i - v_{o_p}}{\|v_i - v_{o_p}\|}, h_{\text{cone},io_p} \right\rangle < \cos\alpha_{io_p}, \end{aligned} \tag{11}$$

with $\mathcal{W} = \{v_1, \ldots, v_n\}$.

1) Competitive and cooperative behavior: A natural behavior of human beings socially interacting while moving is to coordinate their motion. In particular we want to regulate the competitive/cooperative behavior of social walking. To achieve this goal, we slightly change the cost function (7) in

$$\tilde{H} = \sum_{i=1}^{n} \gamma_i(\alpha_i) \|v_i^D - v_i\| \tag{12}$$

where $\gamma_i(\alpha_i)$ is the sum of the angles α of all the cones associated to the i–th agent, i.e.

$$\gamma_i(\alpha_i) = \sum_{j=1}^{n} \alpha_{ij} + \sum_{p=1}^{n_o} \alpha_{io_p}, \;\; \forall j, o_p \in \mathcal{V}_i, \tag{13}$$

To better clarify the role of the penalty function $\gamma_i(\alpha_i)$, let us consider

$$\bar{\gamma}_i(\alpha_i) = \frac{\gamma_i(\alpha_i)}{\sum_{i=1}^{n} \gamma_i(\alpha_i)}.$$

When $\bar{\gamma}_i$ tends to 1 or, more generally, $\bar{\gamma}_i > 1/n$, the behavior of the i–th agent will be competitive (or selfish) with respect to

the other agents. On the other hand, if $\bar{\gamma}_i < 1/n$, the behavior of the i–th agent will be cooperative (or social). Notice that when an agent has more freedom to move in the space (i.e., the Velocity Obstacle set associated with the agent is small), its behavior will be cooperative by means of (13). Whilst, when the agent moves in a cluttered space, its behavior will be competitive (more on this in Section IV). Therefore, the optimization problem can be rewritten as

$$\begin{aligned} \arg\min_{\mathcal{W}} \quad & \sum_{i=1}^{n} \gamma_i(\alpha_i)\|v_i^D - v_i\| \\ s.t. \quad & \forall i = 1,\dots,n \\ & \bigcup_{j\in\mathcal{V}_i} \left\langle \frac{v_i - v_j}{\|v_i - v_j\|}, h_{\text{cone},ij} \right\rangle < \cos\alpha_{ij}, \\ & \bigcup_{o_p\in\mathcal{V}_i} \left\langle \frac{v_i - v_{o_p}}{\|v_i - v_{o_p}\|}, h_{\text{cone},io_p} \right\rangle < \cos\alpha_{io_p}, \end{aligned} \tag{14}$$

2) Cohesion of the group: Another classical bio-inspired feature regards flock centering (i.e., each agent attempt to stay close to nearby flock-mates). This behavior is, actually, one of the three Reynolds' rules, necessary to simulate flock, herds or schools [26]. The group cohesion may be crucial in tasks which require continuous exchange of information between agents with limited communication capabilities. To confer this feature to the system, we add another term to the cost function (12):

$$\tilde{\tilde{H}} = \sum_{i=1}^{n} \gamma_i(\alpha_i)\|v_i^D - v_i\| + \underbrace{k_i \left\| \tfrac{p_c - p_i}{\|p_c - p_i\|} - v_i \right\|}_{\text{cohesion term}}, \tag{15}$$

where $p_c \in \mathbb{R}^2$ is the group centroid position, and k_i is equal to zero if the distance between the i–th agent and the centroid p_c is greater than a certain threshold, otherwise assumes a positive value. In particular

$$k_i = \begin{cases} 1, & \text{if } \|p_j - p_c\| > t_d, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

where t_d is a user defined threshold. Notice that the cohesion term tends to redirect all the velocities toward the centroid of the group.

3) Nonholonomic constraints and control input saturation: In the previous analysis the kinematic constraints of the unicycle and the saturation of the control inputs are not considered, which may lead to unfeasible desired velocity directions. To solve the problem in the available velocity space $u_i = [v_i, \omega_i]^T$, let us consider the map given by unicycle feedback linearization (6), the set of the allowed controls are defined by the following inequalities:

$$\bigcup_{j\in\mathcal{V}_i} \left\langle \frac{T(b,\theta_i)u_i - T(b,\theta_j)u_j}{\|T(b,\theta_i)u_i - T(b,\theta_j)u_j\|}, h_{\text{cone},ij} \right\rangle < \cos\alpha_{ij}, \tag{17}$$

$$\bigcup_{o_p\in\mathcal{V}_i} \left\langle \frac{T(b,\theta_i)u_i - v_{o_p}}{\|T(b,\theta_i)u_i - v_{o_p}\|}, h_{\text{cone},io_p} \right\rangle < \cos\alpha_{io_p}, \tag{18}$$

$$\underline{u_b} < u_i < \overline{u_b}, \;\; \underline{\dot{u}_b} < \dot{u}_i < \overline{\dot{u}_b}, \tag{19}$$

where $\underline{u_b}$ and $\overline{u_b}$ ($\underline{\dot{u}_b}$ and $\overline{\dot{u}_b}$) are the velocity (acceleration) constraints. Hence, the final optimization problem (11), with the bio-inspired features, becomes:

$$\begin{aligned} \arg\min_{\mathcal{U}} \quad & \sum_{i=1}^{n} \gamma_i(\alpha_i)\|v_i^D - T(b,\theta_i)u_i\| + \\ & + k_i \left\| \tfrac{p_c - p_i}{\|p_c - p_i\|} - T(b,\theta_i)u_i \right\|, \\ s.t. \quad & \forall i = 1,\dots,n, \\ & \bigcup_{j\in\mathcal{V}_i} \left\langle \frac{T(b,\theta_i)u_i - T(b,\theta_j)u_j}{\|T(b,\theta_i)u_i - T(b,\theta_j)u_j\|}, h_{\text{cone},ij} \right\rangle < \cos\alpha_{ij}, \\ & \bigcup_{o_p\in\mathcal{V}_i} \left\langle \frac{T(b,\theta_i)u_i - v_{o_p}}{\|T(b,\theta_i)u_i - v_{o_p}\|}, h_{\text{cone},io_p} \right\rangle < \cos\alpha_{io_p}, \\ & \underline{u_b} < u_i < \overline{u_b}, \;\; \underline{\dot{u}_b} < \dot{u}_i < \overline{\dot{u}_b}. \end{aligned} \tag{20}$$

where $\mathcal{U} = \{u_1,\dots,u_n\}$. In this way, once we have chosen properly the point (x_b, y_b) and the safety radius r_i (i.e. in such a way that the footprint of the agent is encircled by the circumference centered in (x_b, y_b) with radius r_i), we ensure that the collision will not occur as long as the optimization problem is feasible (i.e., the constraints are not violated).

4) Decentralized Optimization: The centralized solution proposed above suffers of two main drawbacks: first the numeric solver receives all the information from all the agents and, second, it computes a unique and complex optimization problem. One way to tackle these problems is to distribute the computation among the available agents. The path followed in this paper is to let each agent solve the optimization problems (one for each agent) in sequence. The order of the sequence builds upon the cooperative concept described in Section III-1: the optimization problems associated to the agents with higher $\gamma_i(\alpha_i)$ are solved at first. More in depth, the steps of the algorithm at each iteration are:

1) Each agent computes its own $\gamma_i(\alpha_i)$ using (13);
2) The order sequence $\mathcal{I} = \{i_1,\dots,i_n\}$ such that $\gamma_{i_j}(\alpha_{i_j}) \geq \gamma_{i_l}(\alpha_{i_l})$, $\forall j < l \leq n$, is computed. Notice that each agent computes just its local ordered sequence.
3) The centroid position is computed by the agents following the distributed Weighted Least Squares [27];
4) Each agent computes its own k_i using (16);
5) Hence the agent $i_l \in \mathcal{I}$ solves the following optimization problem:

$$\begin{aligned} \arg\min_{u_{i_l}} \quad & \|v_{i_l}^D - T(b,\theta_{i_l})u_{i_l}\| + \\ & + k_{i_l} \left\| \tfrac{p_c - p_{i_l}}{\|p_c - p_{i_l}\|} - T(b,\theta_{i_l})u_{i_l} \right\| \\ s.t. \quad & (17), (18), (19), \end{aligned} \tag{21}$$

where the constraints are the same of (20), except that the union of the sets in (17) is computed only for the agents whose index is $i_j \in \mathcal{V}_{i_l}$ and $i_j < i_l$.

IV. Simulation results

The proposed approach has been extensively tested in simulations. The optimization problem in its centralized (20) and decentralized (21) forms is numerically solved with the Matlab function `fmincon` using the interior-point method. In all the simulations, we have assumed that: all the agents have a centered disk of radius $r_i = 0.26$ m; the i–th agent visibility set $\mathcal{V}_i$ in (8) comprises the agent j if $\|p_j - p_i\| \leq \rho = 1$ m and the static obstacle o_p if the relative distance is less than 0.35 m, being 1.25 m for dynamic obstacles. Moreover, the feedback linearization parameter of (4) is $b = 0.20$ m. For the controller tuning constants, we have set the modulus of the desired velocities as $\|v_i^D\| = 1.5$ m/s and the cohesion parameter threshold in (16) as $t_d = 1$. Finally, we have assumed for the constraints in (19) $\underline{u_b} = [0, -3]^T$, $\overline{u_b} = [2.5, 3]^T$, $\underline{\dot{u}_b} = [-6, -29]^T$, $\overline{\dot{u}_b} = [6, 29]^T$. Notice that the bounded values for the control derivatives are quite high, this is because of the parameter b, by increasing its distance from the vehicle axle the bounded values for the control derivatives can be lower and lower by paying in increasing footprint of vehicles.

1) Social behaviors: We first start by showing the effects of the socially compliant behaviors described in Section III-1 and Section III-2. First, to clearly assess the effect of the cooperative/competitive parameter γ_i, we executed three simulations using the centralized approach starting from the same initial conditions of two agents 1 and 2 having different pairs of γ_1 and γ_2. Notice that, to better highlight the cooperative/competitive behavior, γ_1 and γ_2 are constant and do not follow the law described in (13). In Figure 2 the red solid arrows are the output $\bar{v}_i = T(p, \theta_i)\bar{u}_i$ of the optimization problem (20), while the blue dotted arrows represent the desired velocities v_i^D.

In the left picture of Figure 2, agent 1 is the competitive one, while agent 2 is the cooperative one. The velocity of agent 1 slightly differs from the desired velocity computed by the RRT* planner (i.e., agent 1 tends to follow the shortest path) because of the static obstacle . On the other hand, agent 2 deviates from its desired velocity in order to avoid collision with agent 1. In the central picture, the penalty function γ_i is equal for each agent, which gives equal responsibility for the avoiding maneuver. Strictly speaking, this situation corresponds to solve the optimization problem (11). The picture on the right is the opposite situation with respect to the left picture case. In fact, agent 1 is forced to brake, letting agent 2 go forward with its desired velocity. By applying (13) we encourage the behavior depicted on the left. The instant of time is the same for the three cases and can be noticed that the left case leads to a better agents' state to achieve earlier the goal, also promoting cohesion: by considering the performance index $P = \sum_{i=1}^{n} \|p_i - wp_i^e\|^2$, it result for the simulations in Figure 2 that at time $t = 2.7$ s, $P = [1.89, 2.36, 2.13]$ m^2 (respectively for left, center, and right case). Here we show it on a toy example but this concept applies as well to more complex situations with more agents.

The second effect that we want to remark concerns the cohesion term. In Figure 3 we depict two different instants of time t_1 and t_2 resulting from the decentralized optimization solution. At the beginning of the simulation (i.e., at time t_1), it is evident that the group is not cohesive. In particular the circled agent is quite distant from the centroid of the group. Between t_1 and t_2, the circled agent approaches the group forced by the cohesion term, hence, at time t_2, the group results in a more compact configuration. Without the cohesion term the re-entry maneuver undertaken by the circled agent has no reason to be carried out.

To further prove the effectiveness of the decentralized approach in a natural environment, we simulate the trajectories using actual human being trajectories available from the ETH dataset [28]. Since we are using recorded data, the human beings in the scene turn to be highly non-cooperative. A sample trajectory for four agents captured in three different time instants is reported in Figure 4.

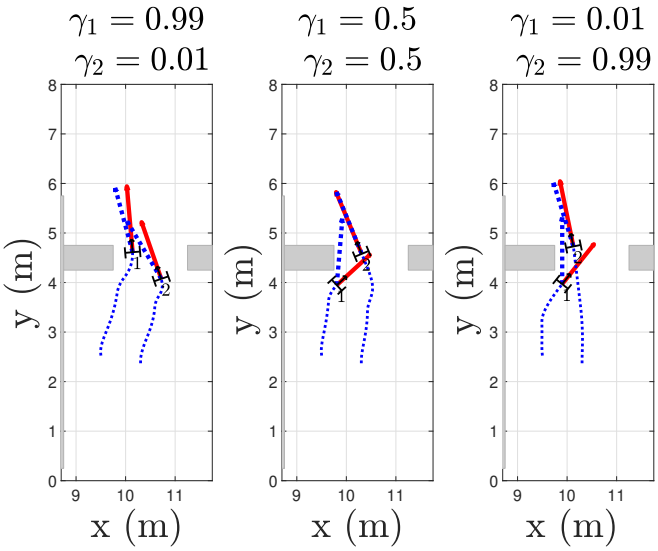

Figure 2. Snapshots at the same time instant $t = 2.1$ s. On the left side, agent 1 is competitive ($\gamma_1 = 0.99$), while agent 2 is cooperative ($\gamma_2 = 0.01$). In the central situation we have $\gamma_1 = \gamma_2 = 0.5$ for the two agents. On the right side, agent 1 is cooperative ($\gamma_1 = 0.01$), while agent 2 is competitive ($\gamma_2 = 0.99$). The blue dashed arrows are the desired velocities v_i^D and the red solid arrows are the optimized velocities.

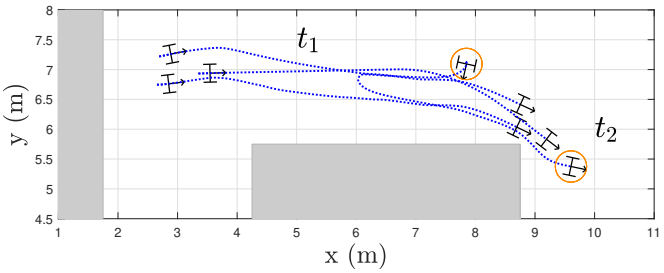

Figure 3. The cohesion term effect is shown here. At the instant t_1, the circled agent is in a remote position with respect to the centroid of the group. The cohesion term introduced in (15) forces the circled agent to join the group, as can be seen at the final time instant t_2.

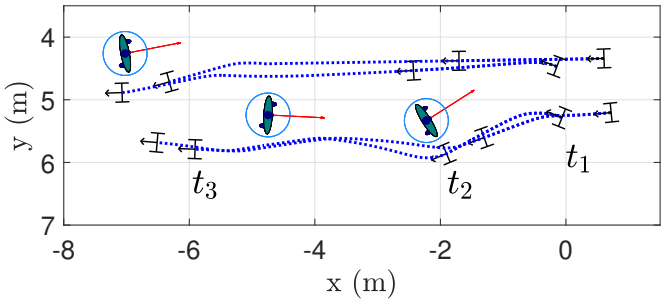

Figure 4. Multi-agent navigation in a human shared environment in three different instant of time using actual human trajectories (ETH dataset [28]).

2) Centralized vs Decentralized optimization: In this subsection we discuss the comparison between the centralized and the decentralized approaches. Table I reports the computation times for the two approaches executing almost 50 iterations.

As it may be expected, the higher is the number of the agents, the more convenient is the decentralized algorithm, which is a further proof of its higher scalability. Recall that the decentralized optimization problem is solved n times the centralized approach but on a simpler problem, moreover on the real testbeds the computation is naturally done in parallel, while in Table I we do not consider a parallel computation. For what concerns the performance in terms of the difference with respect to the desired velocity towards the goal, Figure 5 reports a direct comparison of the centralized and decentralized solutions for a simulated scenario. The squares represent the centralized solution performance while the diamond markers the decentralized. Since the control actions are different, the system evolves differently in the two cases. Moreover, since the optimization problem deals with local information, there is no clear winner between the two solutions. Indeed, at the beginning (first 50 iterations), i.e. when the local information gives a good representation of all the scene, the centralized is clearly more performant. However, due to the presence of unforeseen moving obstacles and the group configuration (see for example the two examples in Figure 5), it is not clear what is the best choice among the two. Similar results can be obtained if the cohesion performance index term in (15) is considered. Instead, if the time to reach the goal is concerned, i.e. the time needed by the last agent in the group to reach the final goal, the results are slightly favorable to the centralized approach (see Table II). As a consequence, the decentralized solution represents a viable and effective solution for the distributed control of social robots.

V. Conclusions

We proposed a control strategy to guide a group of autonomous robots in a complex environment shared with human beings. We adopted the well known RRT* algorithm to plan a global path towards the desired goal destination and the Velocity Obstacle approach to impose constraints on the local interactions. We treated the local control problem as an optimization task and we ensured collision avoidance for nonholonomic multi-vehicle systems with saturated control inputs. We solved the optimization problem both in centralized and decentralized fashion, showing the pros and cons of both approaches. The main novelty of the paper is the addition of human-like socially consistent features to the motion of the group, i.e. the competitive/cooperative behaviors and group cohesion. The overall strategy is successfully tested by simulations on actual maps and in the presence of real human trajectories. Our future work directions will be focused on the experimental validation of the proposed techniques on a team of mobile robots. We are also heading toward the extension of the group management problem for more structured group formations, such as "V" or "U" formations [24].

Table I
Computation times comparison versus the number of agents n for the centralized and the decentralized approaches.

n (-)	Centralized	Decentralized
2	22.531 (s)	16.398 (s)
4	112.646 (s)	75.922 (s)
6	296.670 (s)	167.385 (s)
10	1135.346 (s)	528.413 (s)

Table II
Minimum, maximum and mean values representative of the number of iterations to reach the goal for the last agent in the group.

[-]	min	max	mean
Centralized	249	276	262
Decentralized	246	289	268

References

[1] S. Šabanović, "Robots in society, society in robots," *International Journal of Social Robotics*, vol. 2, no. 4, pp. 439–450, Dec 2010.

[2] F. Vaussard, J. Fink, V. Bauwens, P. Rétornaz, D. Hamel, P. Dillenbourg, and F. Mondada, "Lessons learned from robotic vacuum cleaners entering the home ecosystem," *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 376 – 391, 2014.

[3] R. Triebel, K. Arras, and et. al., *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*. Cham: Springer International Publishing, 2016, pp. 607–622.

[4] K. A. Wyrobek, E. H. Berger, H. F. M. V. der Loos, and J. K. Salisbury, "Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 2165–2170.

[5] F. Ferrari, S. Divan, C. Guerrero, F. Zenatti, R. Guidolin, L. Palopoli, and D. Fontanelli, "Human–Robot Interaction Analysis for a Smart Walker for Elderly: The ACANTO Interactive Guidance System," *International Journal of Social Robotics*, June 2019.

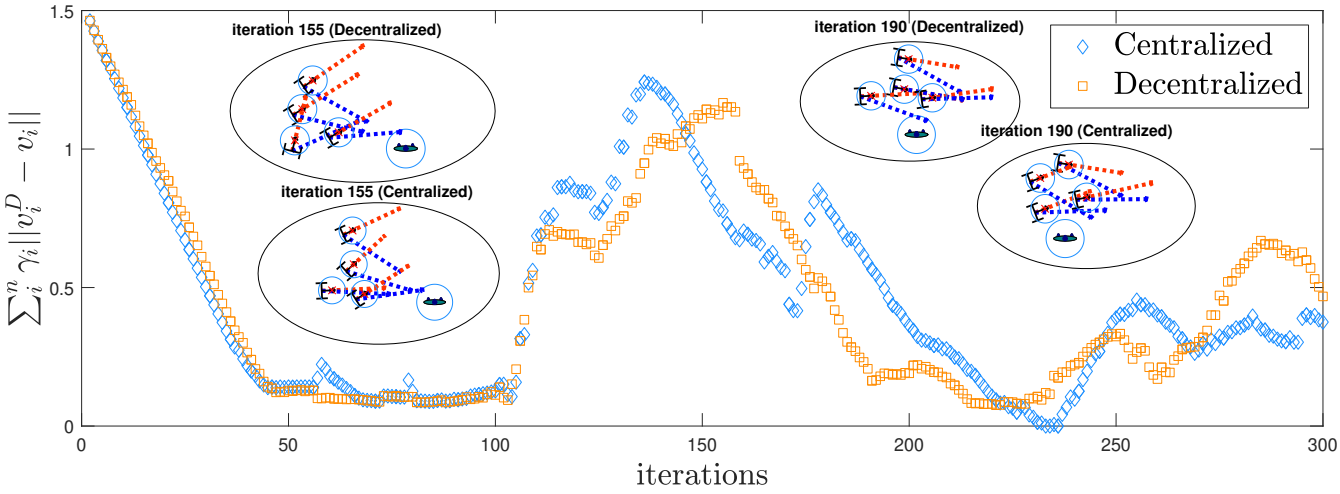

Figure 5. The orange square markers are the performance indexes obtained with the centralized approach, the blue diamond markers are the performance indexes obtained with the decentralized approach. We show also the situations at iteration 155 and 190 in order to understand why the centralized approach could work better than the decentralized and vice versa.

[6] L. P. et al., "Navigation Assistance and Guidance of Older Adults across Complex Public Spaces: the DALi Approach," *Intelligent Service Robotics*, vol. 8, no. 2, pp. 77–92, 2015.

[7] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli, "Reactive Planning for Assistive Robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1276–1283, April 2018.

[8] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[9] G. Flierl, D. Grünbaum, S. Levins, and D. Olson, "From individuals to aggregations: the interplay between behavior and physics," *Journal of Theoretical biology*, vol. 196, no. 4, pp. 397–454, 1999.

[10] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on automatic control*, vol. 51, no. 3, pp. 401–420, 2006.

[11] N. Moshtagh and A. Jadbabaie, "Distributed geodesic control laws for flocking of nonholonomic agents," *IEEE Transactions on Automatic Control*, vol. 52, no. 4, pp. 681–686, 2007.

[12] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863–868, 2007.

[13] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo, "Walking ahead: The headed social force model," *PloS one*, vol. 12, no. 1, p. e0169734, 2017.

[14] M. Boldrer, M. Andreetto, S. Divan, L. Palopoli, and D. Fontanelli, "Socially-aware reactive obstacle avoidance strategy based on limit cycle," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3251–3258, 2020.

[15] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of control*, vol. 80, no. 9, pp. 1517–1531, 2007.

[16] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems*, vol. 22, no. 1, pp. 44–52, 2002.

[17] A. Antonucci and D. Fontanelli, "Towards a Predictive Behavioural Model for Service Robots in Shared Environments," in *IEEE Workshop on Advanced Robotics and its Social Impacts*. Genova, Italy: IEEE, September 2018, pp. 9–14.

[18] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1928–1935.

[19] 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.

[20] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[21] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 203–216.

[22] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.

[23] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, "An optimality principle governing human walking," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 5–14, 2008.

[24] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and

G. Theraulaz, "The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics," *PLoS ONE*, vol. 5, no. 4, p. e10047, April 2010.

[25] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[26] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.

[27] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, April 2005, pp. 63–70.

[28] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 261–268.