2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

Platooning control of autonomous nonholonomic mobile robots in a human-robot coexisting environment

Marco L. Della Vedova, Matteo Rubagotti, Tullio Facchinetti and Antonella Ferrara

***Abstract*— This paper proposes a gradient tracking algorithm based on artificial harmonic potential fields, to support the platooning of a team of nonholonomic mobile robots. The main motivation is the need of dynamically changing the goal-point associated with each mobile robot, in order to guarantee the platoon string stability. Mobile obstacles are taken into account with an approach based on the so-called collision cone, and a time-varying artificial security radius is associated with each obstacle, in order to prevent collisions. In addition, the proposed method ensures recovering of the connectivity between robots forming the platoon, after that one of them goes far away from the others and loses the connection during an obstacle avoidance maneuver. Finally, the so-called interference index has been evaluated, to show the low impact of robot motion on human behaviors.**

***Index Terms*— mobile robots, collision avoidance, path planning, human robot interaction**

I. Introduction

The navigation of autonomous robots is a relevant and widely studied topic in the field of robotics [1]. Most navigation techniques for mobile robots can be divided into two classes: local and global methods. While global navigation aims to find an optimal path to the goal position, local methods mainly deal with the obstacle-avoidance problem. One of most popular techniques for local navigation is the *artificial potential field* method, in which a potential field is defined in the workspace, and the robots navigate according to the resulting artificial forces.

Despite strong efforts have been made in both global and local robot navigation, the particular case of robot navigation in presence of humans and other robots is rarely considered in the field of robotics. A recent work on human-centered robot navigation [2] suggests to take into account not only obstacle avoidance and goal seeking, but also whether a robot interferes with other people or robots, in order to obtain socially acceptable robot movements.

A classical robotics application that must typically deal with a workspace populated by humans is robot platooning. Platooning of autonomous vehicles is studied in many fields, such as urban and extra-urban traffic automation, industrial material movements and military operations [3].

This paper proposes a navigation algorithm for autonomous mobile nonholonomic robots platooning targeted to dynamic environment populated by humans. The navigation algorithm is based on a set of dynamically updated artificial potential fields, each of them associated with a single mobile robot, which allows an on-line trajectory adaptation and a coordinated motion of the entire platoon. By virtue of the distributed nature of the approach, each robot computes locally, at each time instant, its velocity vector reference. Robots and obstacles are represented as moving discs, whose dynamical behavior is given by appropriate models: nonholonomic nonlinear models for the mobile robots, and integrated random walk models for the obstacles, so as to mimic the human pedestrians. Despite its simplicity, this model is suitable for representing a real scenario, in which people and other moving objects can arbitrarily move in the mobile robots workspace.

Additionally, the proposed method also considers the problem of maintaining the communication connectivity between robots, assuming that a broadcast wireless channel is used to allow the interaction among robots. In particular, the enforced mobility strategy guarantees that, if a robot loses the wireless connection with the succeeding robot in the platoon, it will wait the succeeding robot until the connection is re-established.

II. System model

The considered system is composed by a set of objects operating in a two-dimensional workspace described by a xy-axes reference. These objects are n autonomous mobile nonholonomic robots (or "vehicles"), ordered from 1 to n, and m obstacles, labeled from $n+1$ to $n+m$. For the sake of convenience in notations, let us define the two sets $\mathcal{R} \doteq \{1,..,n\}$ and $\mathcal{O} \doteq \{n+1,...,n+m\}$ in order to label the objects. At each time instant, any of the objects within the workspace is characterized by its position $P_i \in \mathbb{R}^2$ and its speed vector $v_i \in \mathbb{R}^2$, and occupies a circular surface, whose center is coinciding with P_i and its radius is $r_i \in \mathbb{R}^+$. Note that this model for obstacle dimension is suitable for representing the space occupied by a human.

The task of the robots is to move in a platoon formation towards the goal-point $G_0 \in \mathbb{R}^2$, avoiding collisions with obstacles and between robots. A platoon consists of one robot leader (i.e. the first one) and $n-1$ robot followers. The i-th robot must follow the $(i-1)$-th. It is difficult to give a formal and exhaustive definition of "platoon formation" in $\mathbb{R}^2$. However, a fundamental property of platooning can be stated as follow: $\forall i \in \mathcal{R} \setminus \{n\}, \forall t \geq 0$, it holds that

$$\|P_i(t) - P_{i+1}(t)\| \leq d^{\max} \tag{1}$$

M. L. Della Vedova, T. Facchinetti and A. Ferrara are with Dpt. of Industial and Information Engineering, University of Pavia, Via Ferrata 1, 27100 Pavia, Italy. {marco.dellavedova, tullio.facchinetti, antonella.ferrara}@unipv.it

M. Rubagotti is with IMT - Institute for Advanced Studies, Piazza San Ponziano 6, 55100 Lucca, Italy. matteo.rubagotti@imtlucca.it

978-1-4577-1096-4/12/$26.00 ©2012 AACC

where the design parameter $d^{\max} \in \mathbb{R}^+$ is the maximum inter-vehicle distance.

The navigation problem consists of defining a method that indicates to each robot the path to follow. Our approach is based on the gradient tracking method, in which the motion of the vehicle is determined on-line: no predetermined trajectory is available, but at each time instant a speed vector reference is generated. The robot model considered in this paper is therefore one that can take as input reference a signal, which can be consider as a time-varying speed vector, i.e. $\bar{v}_i(t)$. The vehicle configuration is represented by the tuple $(P_i(t), \phi(t)) \in \mathbb{R}^3$, in which $\phi_i(t) = \angle v_i(t)$ is the vehicle orientation expressed in radiants. A suitable dynamic model for the robot is the following:

$$\dot{P}_i(t) = \begin{bmatrix} \|v_i(t)\| \cos \phi_i(t) \\ \|v_i(t)\| \sin \phi_i(t) \end{bmatrix} \tag{2a}$$

$$\|\dot{v}_i(t)\| = f_i\left(\bar{v}_i(t)\right) \tag{2b}$$

$$\ddot{\phi}_i(t) = g_i\left(\bar{v}_i(t)\right) \tag{2c}$$

$f_i : \mathbb{R}^2 \to \mathbb{R}^2$ and $g_i : \mathbb{R}^2 \to \mathbb{R}$ being two continuous and bounded functions, which model both the vehicle dynamics and the control action. In this model the control action insists on linear and angular accelerations. We assume as initial conditions: $P_i(0) = P_{0,i}$, $\dot{v}(0) = [0,0]^T$, $\phi_i(0) = \phi_{0,i}$ and $\dot{\phi}(0) = 0$. Moreover, we assume that the initial positions satisfy conditions (1). Model (2) captures the nonholonomic constraint that imposes to the speed vector to be aligned to robot orientation. Notice that the navigation algorithm itself works with holonomic robots as well as with nonholonomic ones.

Regarding obstacles, we assume that they can move arbitrarily and their speed is bounded: $\|v_i(t)\| \leq v^{\max}$, $\forall t, \forall i \in \mathcal{O}$. The navigation algorithm is decentralized. This means that each robot computes its own speed reference $\bar{v}_i$ locally and independently from the other robots. We assume that each robot has the following information on the workspace:

- its ordinal position in the platoon (i.e. i);
- its own current position $P_i(t)$ and speed $v_i(t)$;
- position $P_j(t)$, radius r_j and speed $v_j(t)$ of every other object in the workspace within a given range $\rho \in R^+$, formally $\forall j \neq i$ such that $\|P_j(t) - P_i(t)\| \leq \rho$.

We also assume that the robots can communicate over a wireless medium. The communication is broadcast and it is effective only if the distance between the sender and the receiver is less than a constant value $c \in \mathbb{R}^+$. No multi-hop communication is considered here. The described model is suitable for representing different real-world scenarios as, for instance, the case when there is an external agent that scans the workspace and detects the objects (e.g. a fixed camera located over the workspace in an indoor environment). On the other hand, if the robots directly sense the environment (e.g. with distance sensors) the above-mentioned requirements on available information cannot be satisfied, and the methods proposed in this paper cannot be directly applied.

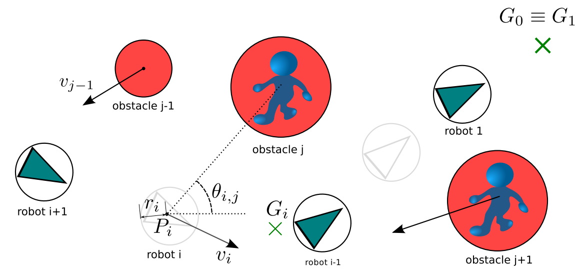

Fig. 1. System model

III. Background Concepts

A. Hill function

The sigmoid function $\sigma : \mathbb{R}^+ \to [0,1]$, defined by (3), is used several times in the reminder of the paper as a smoothing coefficient.

$$\sigma_{p,h}(x) \doteq \frac{p^h}{p^h + x^h} \tag{3}$$

The design parameters $h \geq 2$ and $p > 0$ are positive scalars, used to shape the sigmoid. In particular, note that if $x = p$ then $\sigma(x) = 1/2$. The function σ is known as *Hill function* and it has the property of having the first h derivatives continuous within the domain.

The function $\bar{\sigma} \doteq 1 - \sigma$ will also be used in this paper.

B. Gradient tracking

The gradient tracking is a well known on-line trajectory generator algorithm [4]. It is based on an artificial potential field defined in the workspace. The potential field is the robot internal representation of the workspace, in which the space around obstacles has an artificial repulsive force, while the goal-point has an attractive force. The potential field is harmonic, has a global minimum at the goal-point position, and a global maximum at the obstacle position. In the original method [4], only one static obstacle and one static goal-point are considered. The trajectory of the robot follows the vector opposite to the gradient of the artificial potential field. Considering robot i, to construct the potential field, one obstacle (the closest) is here taken into account and a harmonic dipole potential $U_{i,j} : \mathbb{R}^2 \to \mathbb{R}$ is given by

$$U_{i,j}(p) \doteq \beta_{i,j} \ln \frac{1}{\|p - P_j\|} - \ln \frac{1}{\|p - G_i\|} \tag{4a}$$

$$\beta_{i,j} = \frac{r'_{i,j}}{r'_{i,j} + \|P_j - G_i\|} \tag{4b}$$

where $p \in \mathbb{R}^2$ is a point in the workspace, j is the label of the closest object, $r'_{i,j} \doteq r_i + r_j$, and G_i is the goal-point for robot i. The goal-point location is discussed in the sequel of the paper.

The vector indicating the motion direction, opposite to the gradient of the artificial potential field, is

$$E_{i,j}(p) \doteq -\nabla U_{i,j}(p) \tag{5}$$

The anti-gradient direction $E_{i,j}(p)$ is defined for each point in the workspace: it allows to obtain a continuous trajectory, called *gradient line*.

C. Collision-cone approach for mobile obstacles

In this paper, the collision forecasting is based on consideration regarding the so-called collision cone, a method first introduced for aerospace applications and then applied to robotics [5], [6]. More formally, the term $r'_{i,j}$ in (4) takes the following value:

$$r'_{i,j} \doteq r_i + r_j + \begin{cases} +r''_{i,j} & \text{if } v^r_{i,j} < 0 \\ +0 & \text{if } v^r_{i,j} \geq 0 \end{cases} \tag{6a}$$

$$r''_{i,j} \doteq \min \left\{ r_j \gamma \sigma(\eta_{i,j}),\ \|P_i - P_j\| - r_j - r_i \right\} \tag{6b}$$

$$\eta_{i,j} \doteq \left| v^\theta_{i,j} \right| \left((v^\theta_{i,j})^2 + (v^r_{i,j})^2 \right)^{-\frac{1}{2}} \tag{6c}$$

$$v^r_{i,j} \doteq |v_j| \cos(\angle v_j - \theta_{i,j}) - |v_i| \cos(\angle v_i - \theta_{i,j}) \tag{6d}$$

$$v^\theta_{i,j} \doteq |v_j| \sin(\angle v_j - \theta_{i,j}) - |v_i| \sin(\angle v_i - \theta_{i,j}) \tag{6e}$$

where $\theta_{i,j}$ is the angle between P_i and P_j; σ is a Hill function, whose parameters are considered as tuning parameters; $\gamma \in \mathbb{R}^+$ is a design parameter, which represents the maximum allowed increase of the modified radius, expressed as a percentage of the real radius r_j. Note that it is necessary to limit the value of the modified-radius with (6b) in order to have always the robot location outside the obstacle with the augmented radius. For the sake of readability, the time dependence of v_i, v_j and consequently other terms, included $r'_{i,j}$, has not been made explicit.

D. String stability

The concept of string-stability has been extensively studied for homogeneous vehicle formations. Both centralized and distributed versions of this approach have been investigated [7], [8]. A system that is string-stable has the property that errors on inter-robot distance are not propagated through the platoon. Among other features, this property makes the algorithm scalable. For homogeneous vehicle formations, the spacing errors of a string-stable formation attenuate uniformly down the vehicle chain. This allows to easily choose the proper inter-robot distance [9].

The focus of this paper is to use a distributed inter-robot spacing policy based on information about the position and the speed of the preceding robot in the platoon. Therefore, a string-stable behavior is obtained [10] by adopting the following speed-dependent spacing policy between robots, being $i \in \mathcal{R} \setminus \{1\}$:

$$\bar{d}_{i,i-1}(t) = d_0 + t_h \|v_{i-i}(t)\| \tag{7}$$

in which $\bar{d}_{i,i-1}(t)$ is the desired distance between robot i and robot $i-1$ at time t, $d_0 \in \mathbb{R}^+$ is the desired distance between any two consecutive robots when they are standing, $t_h \in \mathbb{R}^+$ is the *headway time* parameter and it must be chosen depending on the breaking capabilities and the performance of the robot . Notice that (7) is a distributed policy since, requiring information known by robot i, it is implementable locally on robot i.

IV. Gradient tracking support for platooning and mobile obstacles

Each robot autonomously plans its own trajectory. The classical gradient tracking method operates on the current robot position, considering one fixed goal-point and one fixed obstacle. This section describes how the original method has been adapted to the platooning of a set of robots in a workspace with multiple mobile obstacles.

First, all objects in the workspace, being obstacles or other robots, are considered as obstacles by the navigation algorithm of robot i. In this way, the traditional method can be easily adapted to avoid collisions with both obstacles and other robots, without introducing new entities. An exception is done for the preceding robot (i.e. robot $i-1$): it is not consider as an obstacle, it is just used to determine the goal-point location for robot i.

Second, the goal-point of each robot, excluding the platoon leader, dynamically changes location according to the motion of its preceding robot.

A. Goal-point determination

The platooning uses two types of goal-points: the absolute goal-point, that is the location which the whole platoon is moving to, i.e., a reference goal-point for the leader, and the local goal-point associated with each robot. Therefore, the leader goal-point corresponds to the absolute goal-point: $G_1(t) = G_0$, $\forall t$. The goal-point of each remaining robot is dynamically set close to its preceding robot. In other words, the goal-point of robot i, i.e. $G_i(t)$, is a suitably determined location close to robot $i-1$, $\forall i \in \mathcal{R} \setminus \{1\}$. Since $G_i(t)$ can be seen as the desired position of robot i at time t and according to (7), in order to guarantee the string stability of the platoon, $G_i(t)$ must be set at distance $\bar{d}_{i,i-1}$ from the actual position of the preceding robot, i.e. $P_{i-1}(t)$. It follows that

$$\|G_i(t) - P_{i-1}(t)\| = \bar{d}_{i,i-1}(t) \tag{8}$$

Solving (8) for $G_i(t)$, it follows that $G_i(t) \in \Gamma_i(t)$, where $\Gamma_i(t)$ is a circumference in the workspace plane centered in $P_{i-1}(t)$ with a radius $\bar{d}_{i,i-1}(t)$. We refer to $\Gamma_i(t)$ as the *string-stability circumference* for robot i at time t. Therefore, we chose to set the goal-point position on the intersection point $G'_i(t)$ between $\Gamma_i(t)$ and the segment $\overline{P_i(t)P_{i-1}(t)}$. In case $G'_i(t)$ falls inside an obstacle, the actual goal-point $G_i(t)$ is moved on the closest intersection point between the string-stability circumference and the obstacle border. Formally, being j the label of the closest obstacle to robot i at time t, $\Omega_j(t)$ the circumference centered in $P_j(t)$ with radius r_j and $\hat{\Omega}_j(t)$ the disk bounded by $\Omega_j(t)$, one has

$$G_i(t) \doteq \begin{cases} G''_i(t) & \text{if } G'_i(t) \in \hat{\Omega}_j(t) \\ G'_i(t) & \text{otherwise} \end{cases} \tag{9}$$

where $G'_i(t) \doteq \Gamma_i(t) \cap \overline{P_i(t)P_{i-1}(t)}$ and $G''_i(t) \doteq \arg\min_{p \in \Gamma_i(t) \cap \Omega_j(t)} \|p - P_i(t)\|$.

B. Multiple obstacles

An important limitation of the traditional gradient-tracking method is that only one obstacle can be managed. In fact, in case of multiple obstacles, the calculation of the gradient of the potential field in (4) may jeopardize the fundamental property which guarantees that all gradient lines starting outside obstacles remain outside. This is due to the fact that the sum of repulsive contributions may violate such property.

To take into account multiple obstacles, this paper uses a method based on a weighted sum of contributions of directions generated by the traditional gradient-tracking when only one obstacle is considered. This technique has been studied in [5], and is based on Equation (10).

$$\bar{\phi}_i(t) = \frac{\sum_j \mu_{i,j}(t)\varepsilon_{i,j}(t)}{\sum_j \mu_{i,j}(t)} \tag{10}$$

The involved quantities are: the final reference direction $\bar{\phi}_i(t)$, the weight $\mu_{i,j}(t)$ and the direction $\varepsilon_{i,j}(t)$ given by the traditional gradient-tracking when only the obstacle j is considered. Recall that the obstacle can represent a real obstacle or even another robot, excluding the preceding: $j \in \mathcal{O} \cup \mathcal{R} \setminus \{i, i-1\}$.

The weight $\mu_{i,j}(t)$ is inversely proportional to the distance between robot and obstacle j. Taking into account the maximum range ρ in which robots can scan the workspace, it is possible to define $\mu_{i,j}(t)$ as follows:

$$\mu_{i,j}(t) = \begin{cases} (d_{i,j}(t) - r'_{i,j}(t))^{-1} & d_{i,j}(t) \le \rho \\ 0 & d_{i,j}(t) > \rho \end{cases} \tag{11}$$

where

$$d_{i,j}(t) = \|P_i(t) - P_j(t)\| \tag{12}$$

is the distance between robot and obstacle centers.

C. Ghost obstacle

An important issue arises when two obstacles move in such a way that the robot is not able to pass between them. In this case, the robot may stall in an intersection point between the obstacles, which is a local minimum in the potential field. To avoid this situation, the robot considers the involved obstacles as a single obstacle. Such virtual obstacle is called *ghost-obstacle*. The ghost-obstacle overlaps real obstacles and its parameters are used to calculate the desired direction instead of those of real obstacles. Figure 2 shows an example of ghost-obstacle.

Let $j, h \in \mathcal{O} \cup \mathcal{R} \setminus \{i, i-1\}$ be the labels of two obstacle ($j \neq h$), which are too close to allow robot i to pass through. This situation happens when, considering the modified radii:

$$\|P_j(t) - P_h(t)\| \le r'_{i,j} + r'_{i,h} \tag{13}$$

Then obstacles j and h are replaced in the navigation algorithm by the ghost-obstacle g, such that

$$P_g(t) = \frac{P'_{j,h}(t) + P'_{h,j}(t)}{2} \tag{14a}$$

$$r'_{i,g}(t) = \|P'_{j,h}(t) - P_g(t)\| \tag{14b}$$

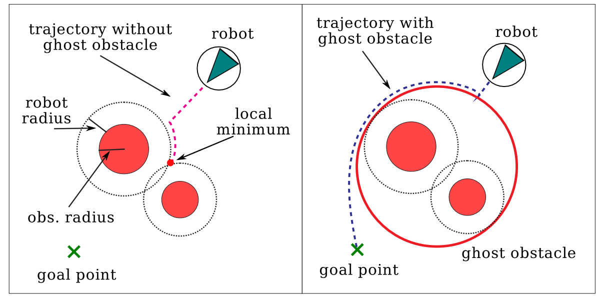

Fig. 2. Ghost-obstacle

being $P'_{j,h}(t)$ one of the two intersection points between the straight line passing by $P_j(t)$ and $P_h(t)$ and the circumference associated to obstacle j, the furthest from $P_h(t)$.

The ghost-obstacle is suddenly created upon the unpredictable motion of obstacles. Therefore, a robot may be within the ghost-obstacle disk when it is created, determining an inconsistent situation. To solve this problem, the following rule is applied: when a robot results to be inside a ghost-obstacle, it manages to move outside along the shortest path. It is easy to observe that the shortest path is the straight line that links the robot position and the ghost-obstacle center, in the opposite direction with respect to the obstacle center.

V. Speed Reference

In the previous section, and in particular in (10), it has been explained how the robot orientation reference $\bar{\phi}_i(t) = \angle \bar{v}_i(t)$ is calculated. This section shows how the modulus of the speed vector reference is generated.

The speed reference modulus $\|\bar{v}_i(t)\|$ for robot i depends on the following quantities:

- the distance between the robot and its goal-point;
- the distance $d_{i,i+1}(t)$ between the robot and its follower, in case the robot is not the last one in the platoon;

We define the speed reference modulus as follow, being $i \in \mathcal{R} \setminus n$:

$$\|\bar{v}_i(t)\| \doteq \begin{cases} \eta_i(t) v^{\max} & d_{i,i+1}(t) < d^{\max} \\ 0 & d_{i,i+1}(t) \ge d^{\max} \end{cases} \tag{15a}$$

$$\eta_i(t) \doteq \bar{\sigma}(\|P_i(t) - G_i(t)\|) \cdot \bar{\sigma}(d^{\max} - d_{i,i+1}(t)) \tag{15b}$$

$$\|\bar{v}_n(t)\| \doteq \bar{\sigma}(\|P_n(t) - G_n(t)\|) \tag{15c}$$

This formula has been studied to have a smooth change of speed in the range $[0, v^{\max}]$. According to (15a), the speed reference is reduced by the reduction factor $\eta_i(t) \in [0, 1]$. In (15b), $\bar{\sigma}$ is a Hill function, whose parameters are tuning parameters. So, (15) states that the speed reference is reduced when the robot approaches the goal-point or its distance from robot $i+1$ is close to the maximum desired inter-robot distance.

The fact that robot i gradually slows down when its follower goes far allows to respect the fundamental property for platooning, stated in (1). Note that, since the robot eventually stops if the followers goes too far (for example because it is the only way to avoid an obstacle), the connectivity can always be recovered. This property holds because, when the connection is lost, robot i is standing and waiting for its follower. Therefore, its location corresponds to the last location received by robot $i+1$, which can catch up with robot i and recover the connection (and the platoon formation).

VI. Decentralized Low-Level Control of the Mobile Robots

The previous sections present the reference generation part of the proposed navigation algorithm. The present section presents the control strategy designed so as to make each robot track its reference trajectory, that will be used in the simulations reported in Section VII. Since this low-level control is decentralized, we drop the index i from notations, for the sake of compactness. The nonholonomic robot model is a particular case of (2), in which both f and g are split into two components: $f = f_r \circ f_c$ and $g = g_r \circ g_c$. The subscript $\cdot_r$ indicates the robot mechanics part of the model, while the subscript $\cdot_c$ indicates the control part.

The robot is driven by two control inputs corresponding to (i) a force $\tau_t \equiv f_c$ acting on the linear acceleration and (ii) a torque $\tau_r \equiv g_c$ acting on the angular acceleration. In particular the dynamic model of the robot is the following:

$$f_r(t) = \frac{\tau_t(t)}{m} - \frac{N_t(t)}{m} \tag{16a}$$

$$g_r(t) = \frac{\tau_r(t)}{J} - \frac{N_r(t)}{J} \tag{16b}$$

where $m, J \in \mathbb{R}$ denote the mass and the moment of inertia of the vehicle with respect to the vertical axis and $N_t, N_r : \mathbb{R}^+ \to \mathbb{R}$ denote the additional dynamics neglected in the modeling phase and the external disturbances.

The proposed control design is a component-wise second-order sliding mode (SOSM) control and it is based on the definition of two sliding variables, one for the orientation control using τ_r and the other for speed control using τ_t. The aim of the SOSM controllers is to steer both sliding variables to zero in finite time in spite of uncertainties, in order to track the reference signals. In particular, the use of SOSM presents robustness features similar to those of traditional first-order sliding mode controllers, while generating a continuous control variable thus attenuating the so-called "chattering" effect, usually associated with the use of sliding mode control, which make this approach particularly appropriate for electro-mechanical systems like mobile robots, where the wear of the different components due to chattering can lead to damages [11].

The selected sliding variables are

$$s_t(t) = \|v(t)\| - \|\bar{v}(t)\| \tag{17a}$$

$$s_r(t) = \alpha(t) - \dot{\alpha}(t) \tag{17b}$$

being $\alpha(t) = \phi(t) - \bar{\phi}(t)$. Therefore, $\tau_t(t) = f_c(f'(s_t(t)))$ and $\tau_r(t) = g_c(g'(s_r(t)))$ are two sliding mode control laws, where f', g' are support functions necessary just for changing the domain. Going into details of the low-level control laws is not in the scope of this paper, so the reader is referred to [11] for further details such as the explicit equations of the control laws and a proof of their stability and convergence properties.

VII. Performance evaluation

This section reports simulation results that show how the proposed method achieves a safe platoon navigation in a human-robot coexisting environment.

A. Model of human movement

In general, modeling pedestrian movements is a difficult task because movements are influenced by a number of factors, such as age, physical conditions, activities, barriers. For this reason, a deterministic model is not suitable for the emulation. Therefore, a simple model will be used to capture the main features of a human walking behavior, while being conservative enough to avoid underestimations. To emulate the complex pedestrian behavior in a simple yet representative way, an integrated random walk model is adopted, borrowing from [12].

B. Interference index

In order to quantify the interference to humans and their safety, the Interference Index ($\mathfrak{II}$) proposed in [2] is adopted. The $\mathfrak{II}$ between robot $i \in \mathcal{R}$ and human $j \in \mathcal{O}$ is defined as follow:

$$\mathfrak{II}_{i,j}(t) = \frac{1 - |\angle v_j(t) - \theta_{i,j}(t)|/\pi}{\|P_i(t) - P_j(t)\| - r_i - r_j} \tag{18}$$

High values of $\mathfrak{II}$ correspond to great interferences of robots on the human behavior. The value of $\mathfrak{II}$ increases when either the distance or the heading angle between robot and human decrease.

C. Case Study 1

In this case study, a platoon composed by 5 robots has to reach the goal-point avoiding randomly moving humans. Robots start from position $P_{0,i} = [0, i-3]^T$. Positions are expressed in meters. The initial robot orientation is generated randomly with a uniform distribution in $[0, 2\pi]$ rad. The global goal-point position is $G_0 = [15, 0]^T$. The security radius surrounding humans is $r_j = 0.5$, $\forall j \in \mathcal{O}$. When the distance between a robot and the center of an obstacle is less than 0.6 m is assumed that a collision has occurred. Initial positions of 4 humans are generated randomly with a uniform distribution within $P_{0,j} \in [7:15, -2:2]$. Moreover, other parameters are: $r_i = 0.3$ m, $\forall i \in \mathcal{R}$, $v^{\max} = 2$ m/s, $d_0 = 0.9$ m, $t_h = 0.1$ s, $d^{\max} = 3$ m, $c = 6$ m.

The goal of this case study is to find the most suitable value for the γ parameter, which is the maximum increment for the modified-radius (according to (6b)). To determine this value, 50 simulations for each value of γ in the set $\{0, 0.2, 0.4, \ldots, 4\}$, for a total of 1050 simulations have been run. The number of simulated humans has been set to 4 for all simulations. Figure 3 shows the results. The chart plots, as a function of γ, the total number of collisions, the time taken to complete the task, the average maximum

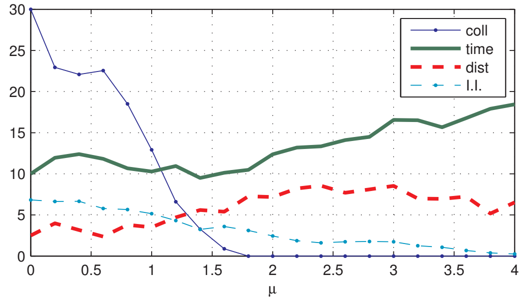

Fig. 3. Effects of maximum increment on modified-radius variation

inter-robot distance and the average $\mathfrak{II}$. A suitable value for the maximum increment of the modified-radius results to be $\gamma = 2.0$, since for this value the number of collision is around zero and there is no point in further increasing γ. This means that the maximum radius doubles the real radius of the obstacle. Moreover, the results show that γ does not significantly affect the time required to complete the task, which is positive since it implies that the collision-avoidance capability is not gained at the price of decreasing the speed of the goal-point reaching process. The $\mathfrak{II}$ decreases when γ increases, as expected. In particular, for $\gamma = 2$ the $\mathfrak{II}$ is about half the $\mathfrak{II}$ registered without modified-radius, i.e. $\gamma = 0$.

D. Case Study 2

In the second case study the connection recovery mode is illustrated. The setup is similar to the one in the previous scenario, but starting locations for robots 4 and 5 are different. They start from far away respect to the other robots, in fact $P_{0,4} = [0, 9]^T$ and $P_{0,5} = [0, 10]^T$. By this initial scenario, we want to emulate a case, in which robot 3 and robot 4 lost the wireless connection. We assume that robot 4 knows the exact position of robot 3, because they are supposed to have been wireless connected before the connection was lost. Looking at Figure 4, it is interesting to observe that robot 3 stays still until $t \approx 3$, when its distance with respect to robot 4 become less than $d^{\max} = 3$. Conversely, robots 1 and 2 start moving because they are close enough to their follower. Then, after a while ($t \approx 1.8$), robot 2 starts slowing down because it approaches the distance limit with robot 3, and so robot 1 does at $t \approx 2.5$. Later, at $t \approx 3$, all the inter-vehicle distances become adequate and after ~ 1 second all the robots travel at cruising speed, which is close to $v^{\max}$. Hence, thanks to the connection recovery rule, the robots act like a chain: the "stop and wait" behavior propagates along the platoon with a delay, as expected.

VIII. Conclusion

A novel on-line navigation algorithm has been discussed in this paper. The proposed method is based on a suitable modification of the gradient tracking method. The goal is to achieve autonomous nonholonomic mobile robots platooning in a dynamic environment with the presence of humans, maintaining at the same time the wireless communication

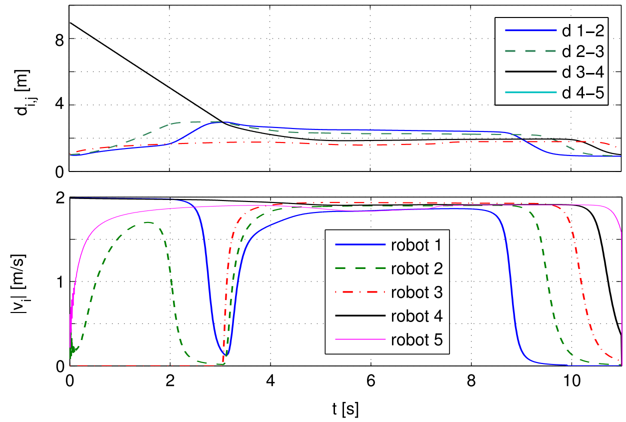

Fig. 4. Inter-vehicle distance (top) and robot speed (bottom) vs time in the connection recovery test

connectivity between the robots. Simulations results, analyzed relying on a suitable Interference Index, allow to conclude that the proposed method is appropriate for robot navigation in a human-robot coexisting environment where many robots and humans (or mobile obstacles in general) are simultaneously present.

References

[1] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion, Theory, Algorithms, and Implementations*. Cambridge, Massachusetts, USA: MIT Press, 2005.

[2] C.-P. Lam, C.-T. Chou, K.-H. Chiang, and L.-C. Fu, "Human-centered robot navigation – towards a harmoniously human-robot coexisting environment," *Robotics, IEEE Trans. on*, vol. 27, no. 1, 2011.

[3] P. Avanzini, B. Thuilot, T. Dallej, P. Martinet, and J. Derutin, "On-line reference trajectory generation for manually convoying a platoon of automatic urban vehicles," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009.

[4] J. Guldner and V. Utkin, "Sliding mode control for gradient tracking and robot navigation using artificial potential fields," *Robotics and Automation, IEEE Trans. on*, vol. 11, no. 2, pp. 247–254, Apr 1995.

[5] M. Rubagotti, M. L. Della Vedova, and A. Ferrara, "Time-optimal sliding-mode control of a mobile robot in a dynamic environment," *Control Theory Applications, IET*, vol. 5, no. 16, pp. 1916 –1924, 3 2011.

[6] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: a collision cone approach," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Trans. on*, vol. 28, no. 5, pp. 562–574, Sep 1998.

[7] P. Seiler, A. Pant, and K. Hedrick, "Disturbance propagation in vehicle strings," *Automatic Control, IEEE Trans. on*, vol. 49, no. 10, pp. 1835 – 1842, 2004.

[8] R. Teo, D. Stipanovic, and C. Tomlin, "Decentralized spacing control of a string of multiple vehicles over lossy datalinks," *Control Systems Technology, IEEE Trans. on*, vol. 18, no. 2, pp. 469 –473, 2010.

[9] E. Shaw and J. Hedrick, "String stability analysis for heterogeneous vehicle strings," in *American Control Conference, 2007. ACC '07*, 2007, pp. 3118 –3125.

[10] D. Yanakiev and I. Kanellakopoulos, "Nonlinear spacing policies for automated heavy-duty vehicles," *Vehicular Technology, IEEE Trans. on*, vol. 47, no. 4, pp. 1365 –1377, nov 1998.

[11] A. Ferrara and M. Rubagotti, "Second-order sliding-mode control of a mobile robot based on a harmonic potential field," *Control Theory Applications, IET*, vol. 2, no. 9, pp. 807 –818, 2008.

[12] G. De Nicolao, A. Ferrara, and L. Giacomini, "Onboard sensor-based collision risk assessment to improve pedestrians' safety," *Vehicular Technology, IEEE Trans. on*, vol. 56, no. 5, pp. 2405 –2413, 2007.