Experiments of Formation Control with Collisions Avoidance using the Null-Space-Based Behavioral Control

Gianluca Antonelli Filippo Arrichiello Stefano Chiaverini

Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale
Università degli Studi di Cassino
via G. Di Biasio 43, 03043 Cassino (FR), Italy.
{antonelli,f.arrichiello,chiaverini}@unicas.it

***Abstract*— In this paper an approach to formation control with collisions avoidance of a multi-robot system is presented. The proposed technique, namely the Null-Space-Based Behavioral Control, is a behavior based approach aimed at coordinating a platoon of mobile robots while performing different missions. The overall mission is firstly decomposed in elementary tasks and, for each of them, a motion reference command to each robot is elaborated referring to an inverse kinematic approach. The proposed technique is novel in the way it combines the output required by each task in order to obtain the final motion command for each robot; in details, it uses a hierarchy-based approach that uses the null-space projection to handle multiple, eventually conflicting, tasks. The proposed technique has been experimentally validated while performing a formation control mission with a platoon of 5 Khepera II mobile robots; during the mission a change of formation is commanded that requires the vehicles to avoid collisions among themselves and with external obstacles.**

I. INTRODUCTION

In the latest years, the scientific interest in the field of coordination and cooperation of multi-robot systems is significantly increased. Multi-robot systems, in fact, can accomplish missions physically not executable by a single robot; moreover, a coordinated multi-robot system can increase the flexibility and give fault tolerance properties to the system, it can reduce the execution time with respect to the not-coordinated robots engaged in the same mission or can take advantage of distributed sensing and actuation.

According to the nomenclature in [13], this paper focuses on the problem of *traffic control*, i.e., the path planning of multiple agents/robots/vehicles in a common environment. Path planning must be accomplished for the single robots while taking into account the other robots and the environment; it is, thus, a geometric problem in the space-time configuration. Nevertheless, the paper will not cover, even in the following brief literature survey, the problem of cooperative box-pushing, grasping or foraging.

One of the first works on multi-robot systems is reported in [24], where a successful group behavior is achieved by considering only local sensing for each robot. Reference [18] presents a multi-agent system specifically designed for flying missions, in which all the agents participate to the decision making with a weight depending on their place within the hierarchy; different structures are investigated requiring different levels of communication. It is recognized that a completely decentralized control strategy as the one presented, e.g., in [25], strictly requires communication between the vehicles. Another decentralized motion planner, based on the optimization of a local and a global performance index, is proposed and experimentally validated in [16]. In reference [8] the behavior-based approach is applied to the robots of a platoon in order to fulfill a specific geometric formation; communication delays and sensor requirements are addressed in the experimental section. Structural potential functions are used in [22] within a distributed controller that requires inter-vehicle communication. Experimental results of a distributed controller that makes use of directional visual feedback are given in [20].

One of the most common technique used to achieve coordinated control is the behavioral control; behavior based approaches, widely studied for mobile robotic applications [7], are useful to guide a multi-robot system in an unknown or dynamically chancing environment. These approaches give the system the autonomy to navigate in complex environments avoiding off-line path planning, using sensors to obtain instantaneous information of the environment and increasing flexibility of the system. Among the behavioral approaches, seminal works are reported in the papers [12] and [6], while, lately, behavioral approaches have been applied to the formation control of multi-robot systems as in, e.g. [23], [19] and [9].

Among the multiple approaches proposed in literature, a behavior based approach to control one single mobile robot were presented in [3], [2], namely the Null-Space-based Behavioral (NSB) control. It differs from the other existing methods in the behavioral coordination method, i.e., in the way the outputs of the single elementary behaviors are assembled to compose the final behavior. In particular, the NSB uses a geometric, hierarchy-based composition of the tasks' outputs to obtain the motion reference commands for the robot that allow the system to exhibit robustness with respect to eventually conflicting tasks. In [4], [5], the NSB were finally extended to the control of multi-robot systems. In this paper experimental results with a 5-robots platoon are reported. In detail, the robots are commanded to fulfill a rigid formation mission where the environmental obstacles, and the robots themselves, need to be avoided. The experiments were conducted using a platoon of non-holonomic 5 Khepera II mobile robots and verified the effectiveness of the proposed algorithm.

II. Null-Space-Based Behavioral Control

Generally, a mission involving several robots may requires the accomplishment of several tasks at the same time. A possible technique to handle the tasks' composition has been proposed in [10], [11], which consists in assigning a relative priority to the single task functions by resorting to the task-priority inverse kinematics introduced in [17], [21] for redundant manipulators. Nevertheless, as discussed in [14], in the case of conflicting tasks it might be advisable to use, instead, singularity-robust algorithms that ensure proper functioning of the inverse velocity mapping. Concerning multi-robot systems, the extension of the task-priority-singularity-robust-inverse kinematics algorithm, developed in [14], for multi-robot systems were presented in [4]. In [5] this approach were designed to perform an escort mission, where obstacle avoidance and vehicles failure issues were also taken into account. The basic concepts are recalled in the following.

By defining as $\boldsymbol{\sigma} \in \mathbb{R}^m$ the task variable to be controlled and as $\boldsymbol{p} \in \mathbb{R}^l$ the system configuration, it is:

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{p}) \tag{1}$$

with the corresponding differential relationship:

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}} \boldsymbol{v} = \boldsymbol{J}(\boldsymbol{p}) \boldsymbol{v}\,, \tag{2}$$

where $\boldsymbol{J} \in \mathbb{R}^{m \times l}$ is the configuration-dependent task Jacobian matrix and $\boldsymbol{v} \in \mathbb{R}^l$ is the system velocity. Notice that l depends on the specific robotic system considered, in case of a differential mobile robot $l = 3$, and the term *system configuration* simply refers to the robot position/orientation, for a multi-robot system $l = 3n$ where n is the number of robots, in case of a full actuated underwater vehicle $l = 6$, finally, an anthropomorphic robots can reach very large value of l.

An effective way to generate motion references $\boldsymbol{p}_d(t)$ for the vehicles starting from desired values $\boldsymbol{\sigma}_d(t)$ of the task function is to act at the differential level by inverting the (locally linear) mapping (2); in fact, this problem has been widely studied in robotics (see, e.g., [26] for a tutorial). A typical requirement is to pursue minimum-norm velocity, leading to the least-squares solution:

$$\boldsymbol{v}_d = \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}}_d = \boldsymbol{J}^{\mathrm{T}} \left(\boldsymbol{J} \boldsymbol{J}^{\mathrm{T}} \right)^{-1} \dot{\boldsymbol{\sigma}}_d\,. \tag{3}$$

At this point, the vehicle motion controller needs a reference position trajectory besides the velocity reference; this can be obtained by time integration of $\boldsymbol{v}_d$. However, discrete-time integration of the vehicle's reference velocity would result in a numerical drift of the reconstructed vehicle's position; the drift can be counteracted by a so-called Closed Loop Inverse Kinematics (CLIK) version of the algorithm, namely,

$$\boldsymbol{v}_d = \boldsymbol{J}^{\dagger} \left(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \widetilde{\boldsymbol{\sigma}} \right), \tag{4}$$

where $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains and $\widetilde{\boldsymbol{\sigma}}$ is the task error defined as $\widetilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$.

The Null-Space-based Behavioral control intrinsically requires a differentiable analytic expression of the tasks defined, so that it is possible to compute the required Jacobians. In detail, on the analogy of eq. (4), the single task velocity is computed as

$$\boldsymbol{v}_i = \boldsymbol{J}_i^{\dagger} \left(\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{\Lambda}_i \widetilde{\boldsymbol{\sigma}}_i \right), \tag{5}$$

where the subscript i denotes i-th task quantities. If the subscript i also denotes the degree of priority of the task with, e.g., Task 1 being the highest-priority one, in the same case of 3 tasks we have provided as an example for the other two approaches considered, according to [14] the CLIK solution (4) is modified into

$$\boldsymbol{v}_d = \boldsymbol{v}_1 + \left(\boldsymbol{I} - \boldsymbol{J}_1^{\dagger} \boldsymbol{J}_1 \right) \left[\boldsymbol{v}_2 + \left(\boldsymbol{I} - \boldsymbol{J}_2^{\dagger} \boldsymbol{J}_2 \right) \boldsymbol{v}_3 \right] \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix of proper dimensions. Remarkably, eq. (6) has a nice geometrical interpretation. Each task velocity is computed as if it were acting alone; then, before adding its contribution to the overall vehicle velocity, a lower-priority task is projected onto the null space of the immediately higher-priority task so as to remove those velocity components that would conflict with it.

The Null-Space-based Behavioral control always fulfils the highest-priority task at nonsingular configurations. The lower-priority tasks, on the other hand, are fulfilled only in a sub-space where they do not conflict with the ones having higher priority, that is, each task reaches a sub-optimal condition that optimizes the task respecting the constraints imposed by the highest-priority tasks. Details about the difference with the respect to the cooperative and competitive behavioral approaches can be found in [3].

III. Example of Tasks

According to the behavioral control approaches, the mission of the platoon is decomposed into elementary tasks and, for each of them, a suitable task function should be defined. Thus, in the performed formation control experiments, the mission is decomposed in three elementary tasks: move the barycenter of the platoon, keep a certain formation relative to the barycenter and avoid collisions among vehicles and with external obstacles. In the following, all the task functions will be showed.

A. Task function for platoon barycenter position

The barycenter of a platoon expresses the mean value of the vehicles positions. In a 2-dimensional case the task function is expressed by:

$$\boldsymbol{\sigma}_b = \boldsymbol{f}_b \left(p_1, \ldots, p_n \right) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{p}_i.$$

where $\boldsymbol{p}_i = [\, x_i \quad y_i \,]^{\mathrm{T}}$ is the position of the vehicle i.

By deriving the previous relation it holds:

$$\dot{\boldsymbol{\sigma}}_b = \sum_{i=1}^{n} \frac{\partial \boldsymbol{f}_b \left(\boldsymbol{p} \right)}{\partial \boldsymbol{p}_i} \boldsymbol{v}_i = \boldsymbol{J}_b \left(\boldsymbol{p} \right) \boldsymbol{v}$$

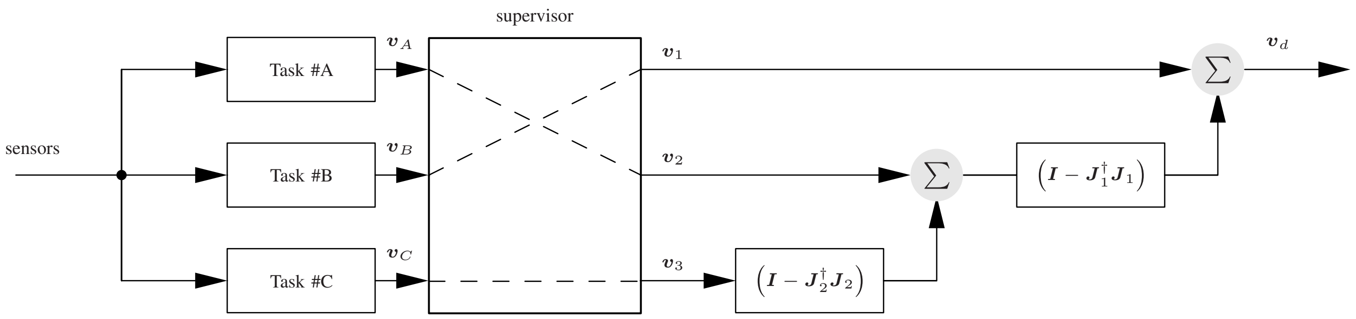

Fig. 1. Sketch of the Null-Space-Based behavioral control in a 3-task example. The supervisor is in charge of changing the relative priority among the tasks.

where the Jacobian matrix $\boldsymbol{J}_b \in \mathbb{R}^{2\times 2n}$ is

$$\boldsymbol{J}_b = \frac{1}{n}\begin{bmatrix} \dots & 1 & 0 & \dots \\ & 0 & 1 & \end{bmatrix}.$$

B. Rigid Formation

The rigid formation task moves the vehicles to a predefined formation relative to the barycenter. The task function is defined as:

$$\boldsymbol{\sigma}_f = \begin{bmatrix} \boldsymbol{p}_1 - \boldsymbol{p}_b \\ \vdots \\ \boldsymbol{p}_n - \boldsymbol{p}_b \end{bmatrix},$$

where $\boldsymbol{p}_i$ are the coordinates of the vehicle i and $\boldsymbol{p}_b = \boldsymbol{\sigma}_b$ are the coordinates of the barycenter. The corresponding Jacobian matrix $\boldsymbol{J}_f \in \mathbb{R}^{2n\times 2n}$ is:

$$\boldsymbol{J}_f = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A} \end{bmatrix}, \tag{7}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1-\frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & \dots & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1-\frac{1}{n} \end{bmatrix} \tag{8}$$

and $\boldsymbol{O}$ is a $2n \times 2n$ null matrix.

The desired value $\boldsymbol{\sigma}_{f,d}$ of the task function describes the shape of the desired formation; that is, once defined the formation, the elements of $\boldsymbol{\sigma}_{f,d}$ represent the coordinates of each vehicle in the barycenter reference frame.

C. Obstacle and collision avoidance

The obstacle avoidance task function is built individually for each vehicle, i.e., it is not an aggregate task function. Each vehicle needs to avoid both environmental obstacles and the other vehicles.

With reference to the generic vehicle in the team, in presence of an obstacle in the advancing direction, the task function has to elaborate a driving velocity, aligned to the vehicle-obstacle direction, that keeps the vehicle at a safe distance d from the obstacle. Therefore, it is:

$$\begin{aligned} \sigma_o &= \|\boldsymbol{p} - \boldsymbol{p}_o\| \\ \sigma_{o,d} &= d \\ \boldsymbol{J}_o &= \hat{\boldsymbol{r}}^{\mathrm{T}}, \end{aligned}$$

where $\boldsymbol{p}_o$ is the obstacle position and

$$\hat{\boldsymbol{r}} = \frac{\boldsymbol{p} - \boldsymbol{p}_o}{\|\boldsymbol{p} - \boldsymbol{p}_o\|}$$

is the unit vector aligned with the obstacle-to-vehicle direction. According to the above choice, eq. (5) simplifies to

$$\boldsymbol{v}_o = \boldsymbol{J}_o^\dagger \lambda_o \widetilde{\boldsymbol{\sigma}}_o = \lambda_o \left(d - \|\boldsymbol{p} - \boldsymbol{p}_o\|\right) \hat{\boldsymbol{r}}. \tag{9}$$

It is worth noting that, being

$$\mathcal{N}(\boldsymbol{J}_o) = \boldsymbol{I} - \hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^{\mathrm{T}},$$

the tasks of lower priority with respect to the obstacle avoidance are only allowed to produce motion components tangent to the circle of radius d and centered in $\boldsymbol{p}_o$ (also called *safety area*), so as to not interfere with the enforcement of the safe distance d. Thus, a robot that is going to collide with a punctual obstacle does slide on its safety area with an instantaneous velocity that depends on the projection along the tangential direction of the velocities elaborated by the other tasks. If the robot is going to frontally collide the obstacle (the velocity elaborated by the other tasks is in the vehicle-obstacle direction), then the projection along the tangential direction is null. This particular situation gives rise to a local minimum that makes the robot stop. Nevertheless, the experimental experience evidenced that the presence of measurement noise allows the vehicle to avoid the local minima; this is, in fact, an unstable stationary point.

The obstacle avoidance function is developed for dot-like obstacles or obstacles that may be conveniently rounded by a circle. In same cases, however, the obstacles may be better represented by straight-lines or convex curves, in this case, $\boldsymbol{p}_o$ represents the coordinates of the closest point of the obstacle to the vehicle at the current time instant. In the experimental results, both dot-like and linear obstacles will be considered.

In the frequent case of multiple obstacles acting simultaneously (e.g., both an obstacle in the environment and the other vehicles of the team) a priority among their avoidance should be defined; a reasonable choice is to assign the currently closest obstacle the highest priority. In critical situations the obstacle avoidance function may give a null-velocity output; this causes delay to the mission or loss of vehicles to the formation but increases safety of the approach.

IV. Experimental Results

The proposed technique has been tested to control a platoon of Khepera II available in the LAI (Laboratorio di Automazione Industriale) of the University of Cassino. The Khepera II (see fig 2), manufactured by K-team [1], are differential-drive, mobile robots with a unicycle-like kinematics with an approximative dimension of 8 cm of diameter. Each of them can communicate trough a Bluetooth module with a remote Linux-based PC where the NSB has been implemented. Since the experiments focus on formation control the vehicles positions are measured resorting to a vision-based system running on another windows-based PC; the Linux-based PC receives the position measurements at a sampling time of 60 ms. The measurement error has an upper bound of $\approx$ 0.5 cm and $\approx$ 1 deg. The NSB elaborates the desired linear velocity for each robot, thus a heading controller is derived from the controller reported in [15] to obtain wheels' desired velocities. The remote PC sends to each vehicle (trough the Bluetooth module) the wheels' desired velocities with a sampling time of $T = 200$ ms. The wheels' controller (on board of each robot) is a PID developed by the manufacturer. A saturation of 20 cm/s and 180 deg/s has been introduced for the linear and angular velocities, respectively. Moreover, the encoders resolution is such that a quantization of $\approx$ 0.8 cm/s and $\approx$ 9 deg/s are experienced.

Fig. 2. Khepera II mobile robot manufactured by K-Team available in the LAI (Laboratorio di Automazione Industriale) with the Bluetooth communication module.

The overall system has been tested while performing a mission with a platoon of 5 Khepera II (see fig 3). The mission of the multi-robot system consists of two steps: in the first step, the robots, starting from a random configuration, have to move their barycenter to a fixed value, reach a certain configuration around it and avoid collisions and static obstacles. In particular, the desired value of the barycenter task function is $\boldsymbol{\sigma}_{b,d} = [80 \quad 150]^{\mathrm{T}}$ cm while the desired formation is a linear formation rotated of -25 degrees respect to the axes x where the robots keep a distance of 30 cm one from the others. The second step of the mission consists of a change of formation, i.e., the robots have to symmetrically invert their positions in the linear formation avoiding collisions among themselves.

The priority of the 3 tasks implemented is:

1) obstacle avoidance
2) barycenter
3) rigid formation

The gain matrix of the barycenter task function is

$$\boldsymbol{\Lambda}_b = \boldsymbol{I}_2$$

while the gain matrix for the rigid formation task function $\boldsymbol{\Lambda}_f$ is

$$\boldsymbol{\Lambda}_f = \boldsymbol{I}_{10}$$

and the gain λ_o of the obstacle avoidance task function is

$$\lambda_o = 0.5$$

while $\sigma_{o,d}$ is equal to 16 cm.

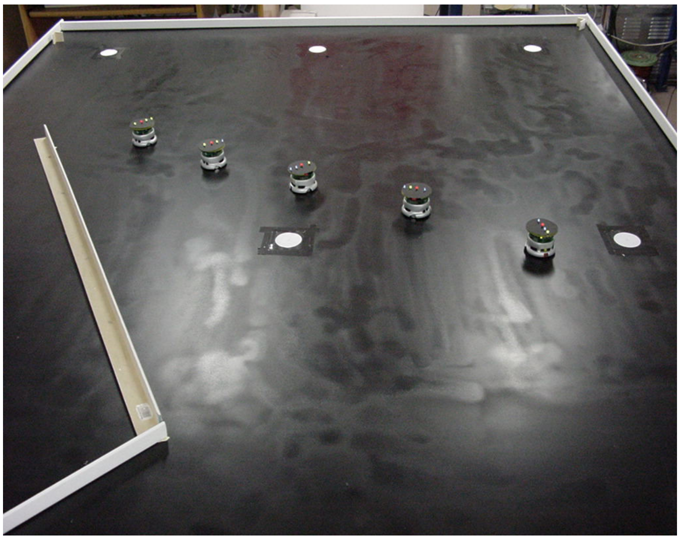

Fig. 3. Khepera II mobile robots while performing the formation control mission in the LAI (Laboratorio di Automazione Industriale).

Figure 5 shows the paths of the robots during the execution of the experiment, while fig. 4 shows several steps of the mission execution including desired values of the obstacle avoidance task functions. The robots start from a random configuration (gray positions of fig. 5) and reach the first configuration in $t \approx 20$ s do avoiding collisions among themselves and with the static obstacle. At $t = 50$ s a new step input is commanded to the platoon by requiring a change in the robots' configuration while keeping the same desired barycenter. Please notice that a quite severe command is given since all the robots tend to cluster close to the barycenter in order to reach their new position in the formation. The algorithm, moreover, works often close to the local minima that might arise with local obstacle avoidance algorithms.

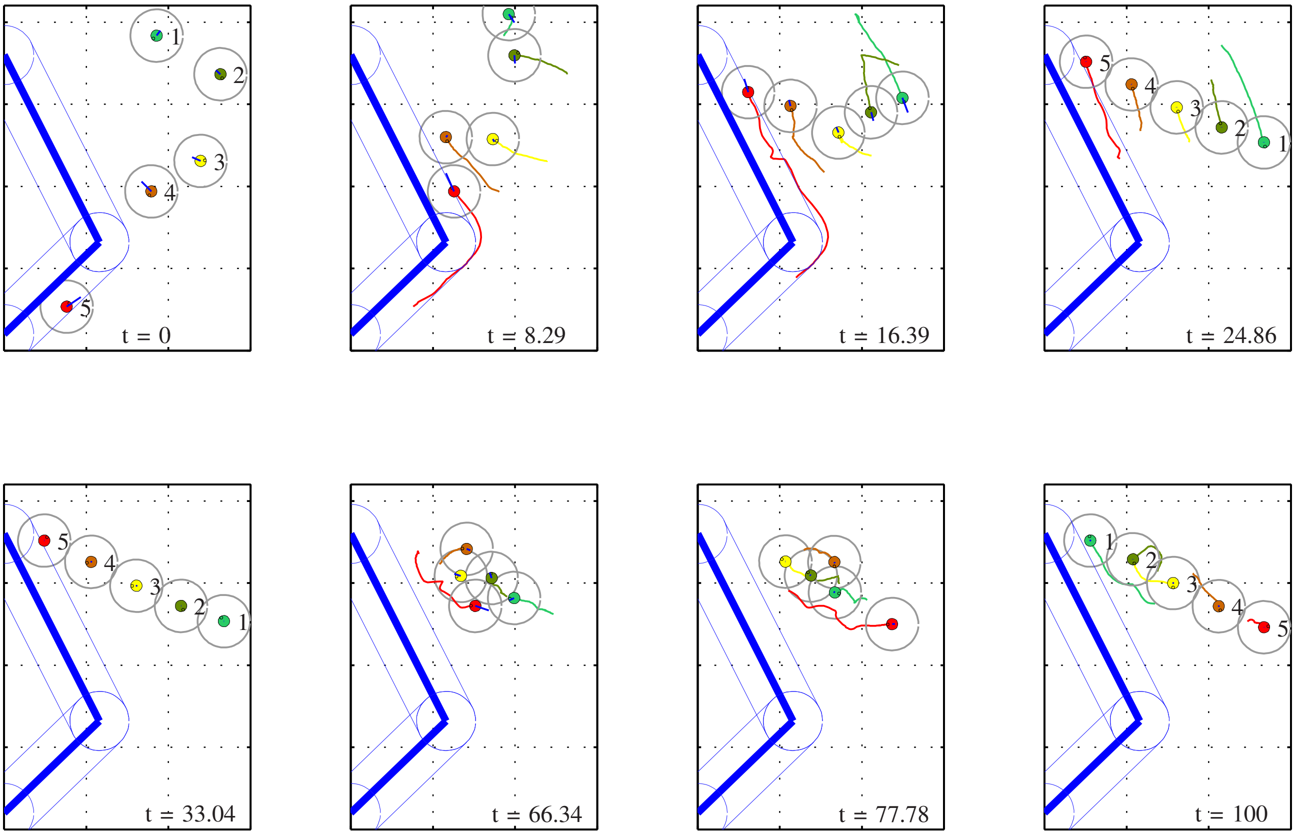

Fig. 4. Snapshots of the experiment. The blue, solid line represent the static obstacle, the blue, solid, thin line denotes its safety area; the robots are represented with colored circle rounded by a gray line denoting the safety area

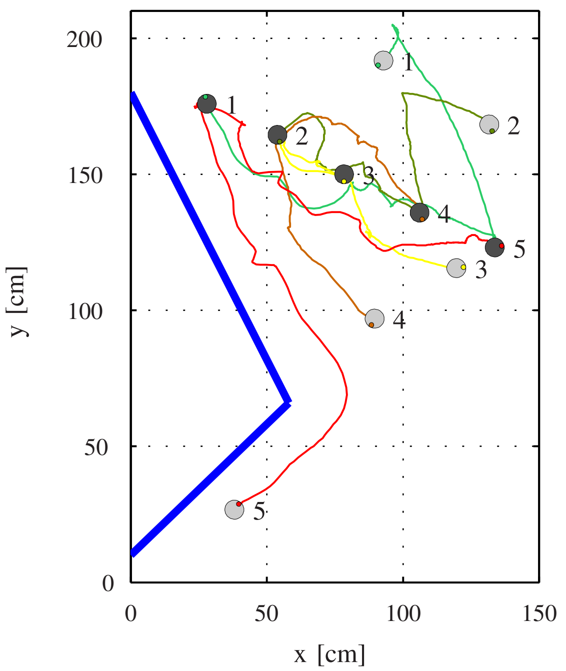

Fig. 5. Paths followed by the Khepera II robots during the experimental mission. Gray, initial positions; black, final positions.

From fig. 5 it can be observed the specific path of robot $n°5$, in the beginning of the mission it correctly avoids the linear, static, obstacle by *sliding* around its safety area. However, it can be observed from the second and third frame of fig 4 that it enters the safety area. This is due to several reason, the closeness of another vehicle, the vehicle dynamics, its non-holonomy and the sampling time.

Figure 6 shows the error $\tilde{\boldsymbol{\sigma}}_f$ of the rigid formation task function (the change of formation occurs as a step input at $t = 50\,\mathrm{s}$), while fig. 7 shows the values $\boldsymbol{\sigma}_b$ and $\boldsymbol{\sigma}_{b,d}$ of the barycenter task function. The convergence to zero of the rigid formation task function error and the convergence to the desired values of the barycenter task function show that both the tasks are successfully performed. It can be noticed that, being both the tasks at lower priority, the errors do not decrease monotonically to zero; when the obstacle avoidance is active, in fact, it is the higher priority task, the only that is certainly fulfilled.

V. CONCLUSION

Experimental results concerning the implementation of the Null-Space-based Behavioral approach to control a platoon of mobile robots were presented. The NSB approach allows to properly handle the outputs of several, eventual conflicting, behaviors/tasks. The experiments were performed in the LAI

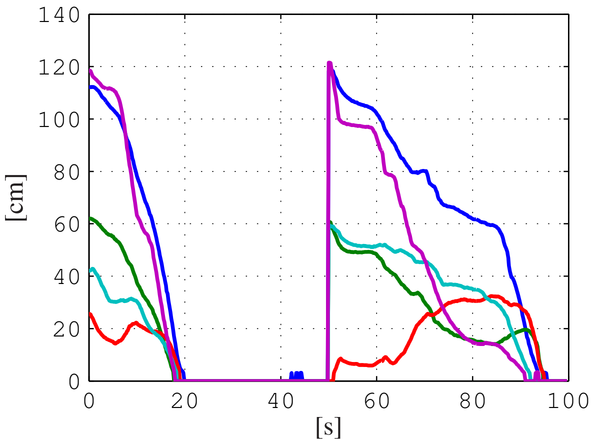

Fig. 6. Rigid formation task function errors: norms of the error components of $\tilde{\boldsymbol{\sigma}}_f$ for each vehicle

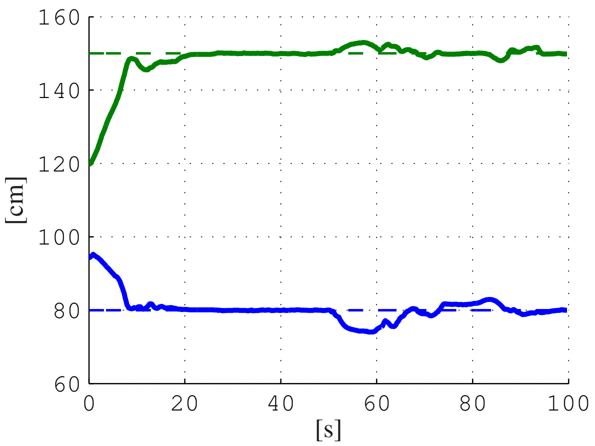

Fig. 7. Average task function: desired values of $\boldsymbol{\sigma}_{b,d}$ (dashed lines) and real values of $\boldsymbol{\sigma}_b$ (solid lines)

(Laboratorio di Automazione Industriale) of the University of Cassino equipped with 7 Khepera II mobile robots, 5 of which were used for the experiments. The algorithm resulted in a successful implementation for dozen of missions requiring the movement in a quite cluttered environment. Future experimental work requires the implementation of other kind of missions such as, e.g., the escort mission, the fault tolerance with respect to, e.g., the failure of one or more vehicles.

References

[1] `http://www.k-team.com/`. K-Team.

[2] G. Antonelli, F. Arrichiello, and S. Chiaverini. Experimental kinematic comparison of behavioral approaches for mobile robots. In *Proceedings of the 16th IFAC World Congress*, Prague, CZ, July 2005.

[3] G. Antonelli, F. Arrichiello, and S. Chiaverini. The null-space-based behavioral control for mobile robots. In *Proceedings of the 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 15–20, Espoo, Finland, June 2005.

[4] G. Antonelli and S. Chiaverini. Kinematic control of a platoon of autonomous vehicles. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 1464–1469, Taipei, TW, Sept. 2003.

[5] G. Antonelli and S. Chiaverini. Fault tolerant kinematic control of platoons of autonomous vehicles. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 3313–3318, New Orleans, LA, April 2004.

[6] R.C. Arkin. Motor schema based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.

[7] R.C. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, MA, 1998.

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

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

[10] B.E. Bishop. On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles. *IEEE Transactions on Systems, Man and Cybernetics*, 33(5):608–615, Sept. 2003.

[11] B.E. Bishop and D.J. Stilwell. On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles. In *Proceedings 2001 IEEE International Conference on Control Applications*, pages 823–828, México City, MX, Sept. 2001.

[12] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

[13] Y. Uny Cao, A.S. Fukunaga, and A.B. Kanhg. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–27, 1997.

[14] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[15] A. De Luca G. Oriolo and M. Vendittelli. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.

[16] Y. Guo and L.E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation*, pages 2612 – 2619, Washington, DC, May 2002.

[17] A.A. Maciejewski. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, 5(6):527–552, 1988.

[18] S. Mandutianu, F. Hadaegh, and P. Elliot. Multi-agent system for formation flying missions. In *Proceedings 2001 IEEE Aerospace Conference*, volume 6, pages 2793–2802, Pasadena, MS, Mar. 2001.

[19] M.J. Mataric. Behavior-based control: Examples from navigation, learning, and group behavior. *J. Experimental and Theoretical Artificial Intelligence*, 9(2-3):323 336, 1997.

[20] F. Michaud, D. Létorneau, M. Guilbert, and J.M. Valin. Dynamic robot formations using directional visual perceptions. In *Proceedings 2002 IEEE/RJS International Conference on Intelligent Robots and Systems*, pages 2740–2745, Lausanne, CH, Oct. 2002.

[21] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal Robotics Research*, 6(2):3–15, 1987.

[22] R. Olfati-Saber and R.M. Murray. Distribute cooperative control fo multiple vehicle formations using structural potential functions. In *15th Ifac World Congress*, Barcelona, Spain, July 2002.

[23] L.E. Parker. On the design of behavior-based multi-robot teams. *Advanced Robotics*, 10(6):547–578, 1996.

[24] C. Reynolds. Flocks, herd and schools: A distributed behavioral model. *Comput. Graph.*, 21(4):25–34, 1987.

[25] D.A. Schoenwald, J.T. Feddema, and F.J. Oppel. Decentralized control of a collective of autonomous robotic vehicles. In *Proceedings 2001 American Control Conference*, volume 3, pages 2087–2092, Arlington, VA, June 2001.

[26] B. Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent Robotic Systems*, 3:201–212, 1990.