Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control

Gianluca Antonelli, *Senior Member, IEEE*, Filippo Arrichiello, *Member, IEEE*, and Stefano Chiaverini, *Senior Member, IEEE*

***Abstract*—In this paper, the experimental validation of a behavior-based technique for multirobot systems (MRSs), namely, the Null-Space-based Behavioral (NSB) control, is presented. The NSB strategy, inherited from the singularity-robust task-priority inverse kinematics for industrial manipulators, has been recently proposed for the execution of different formation-control missions with MRSs. In this paper, focusing on the experimental details, the validation of the approach is achieved by performing different experimental missions, in presence of static and dynamic obstacles, with a team of grounded mobile robots available at the Laboratorio di Automazione Industriale of the Università degli Studi di Cassino.**

***Index Terms*—Formation control, mobile robots, multirobot systems (MRSs).**

I. Introduction

In THE LATEST decades, the field of multirobot systems (MRSs) has been object of widespread research interest owing to the several advantages that such systems show with respect to single autonomous vehicles and owing to the technological improvements that have allowed the interaction and the integration among multiple systems. With respect to a single autonomous robot or to a team of noncooperating robots, e.g., an MRS can better perform a mission according to several performance indexes, it can achieve tasks not executable by a single robot such as moving a large object, or it can take advantages of distributed sensing and actuation. Moreover, instead of designing a single powerful robot, a multirobot solution can be easier and cheaper, it can provide flexibility to task execution, and it can make the system tolerant to possible robot faults. The applications of MRSs may involve different fields like industrial, military, and service robotics or research and study of biological systems. Moreover, they may concern largely different kind of missions, e.g., exploration, box pushing, military operation, navigation in unstructured environment, traffic control, or entertainment.

Most of the control approaches devised for autonomous robots make use of biological inspiration. This kind of approaches began after the introduction of the robotics paradigm of behavior-based control [9], [10], [13]. The behavior-based paradigm has been useful for robotic researchers to examine the social characteristics of insects and animals and to apply these findings to the design of MRSs. The most common application is the use of elementary control rules of various biological animals (e.g., ants, bees, birds, and fishes) to reproduce a similar behavior (e.g., foraging, flocking, homing, and dispersing) in cooperative robotic systems. The first works were motivated by computer-graphic applications; in 1986, Reynolds [25] made a computer model for coordinating the motion of animals as bird flocks or fish schools. This pioneering work inspired significant efforts in the study of group behaviors [16], [19], [22], then in the study of multirobot formations [11], [23], [29].

Apart from the behavior-based approaches, different analytical strategies to control MRSs have been proposed in literature. These strategies may differ both for mathematical characteristics and implementation aspects. A flatness-based theory aimed at artificially coupling the motion of the vehicles is presented in [24], while the use of control graphs to address the problem of changing the platoon formation is discussed in [15]. In [12], a formal abstraction-based approach to control a large number of robots required to move as a group has been presented. This approach properly decouples the operational space into two control levels through a proper hierarchical subdivision, and it is verified in simulation with a large number of robots. Reference [27] focuses on formation motion feasibility of multiagent systems, i.e., it focuses on the algebraic conditions that guarantee formation feasibility given the individual agent kinematics. Reference [18] presents experimental results of multirobot coordination controlled by a distributed control strategy based on a circular-pursuit algorithm, while [28] presents experimental results of coalition formation of an MRS while simultaneously performing heterogeneous missions.

Among the multiple approaches proposed in literature, a behavior-based approach, namely, the Null-Space-based Behavioral (NSB) control, has been presented in [5] to control generic robotic systems and in [7] and [8] to control MRSs. The NSB approach takes the advantages of behavior-based approaches in the reactivity to unknown or dynamically changing conditions while, similarly to the analytical approaches, it presents a rigorous mathematical formulation that permits to extrapolate some analytical convergence properties. In [3], [4], [6], first experimental results with MRSs were reported. In this paper, emphasizing the experimental aspects, several missions with a platoon of up to seven Khepera II mobile robots are collected and discussed. Most of the presented experiments are accompanied by a relative video that can be found at the Web page: http://webuser.unicas.it/lai/robotica/video/.

II. NSB Control

Generally, a mission involving several robots may require the accomplishment of several tasks at the same time. A common

Manuscript received April 04, 2008. Manuscript received in final form July 24, 2008. First published April 14, 2009; current version published August 26, 2009. Recommended by Associate Editor F. Caccavale.

The authors are with the Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Università degli Studi di Cassino, 03043 Cassino, Italy (e-mail: antonelli@unicas.it; f.arrichiello@unicas.it; chiaverini@unicas.it).

Digital Object Identifier 10.1109/TCST.2008.2004447

approach is to decompose the overall mission of the system in elementary tasks (or behaviors), to solve them as they were working alone, and, finally, to combine the outputs of the single tasks to obtain the motion commands to each robot. As discussed in [5], the NSB control 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 robots that allow the system to exhibit robustness with respect to eventually conflicting tasks. The basic concepts are recalled in the following, while a complete description can be found in [8].

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 m represents the generic dimension of the specific task, while l depends on the specific robotic system considered, e.g., in case of n mobile robots $l = 2n$, and the term *system configuration* simply refers to the robot positions.

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). Notice that the desired positions/velocities represent the input for the low-level controller. A typical requirement is to pursue minimum-norm velocity, leading to the least squares solution

$$\boldsymbol{v}_d = \boldsymbol{J}^{\dagger} \dot{\boldsymbol{\sigma}}_d \tag{3}$$

where the pseudoinverse Jacobian $\boldsymbol{J}^{\dagger}$ is elaborated as a matrix that verifies the following properties:

$$\boldsymbol{J}\boldsymbol{J}^{\dagger}\boldsymbol{J} = \boldsymbol{J} \quad \boldsymbol{J}^{\dagger}\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\dagger}$$

where $\boldsymbol{J}\boldsymbol{J}^{\dagger}$ and $\boldsymbol{J}^{\dagger}\boldsymbol{J}$ are symmetric. However, when $\boldsymbol{J}$ is a full-rank lower-rectangular matrix, the pseudoinverse Jacobian is simply elaborated as

$$\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\mathrm{T}} (\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1}. \tag{4}$$

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

$$\boldsymbol{v}_d = \boldsymbol{J}^{\dagger} (\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}) \tag{5}$$

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

The NSB 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 (5), the single task velocity is computed as

$$\boldsymbol{v}_i = \boldsymbol{J}_i^{\dagger} (\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i) \tag{6}$$

where the subscript i denotes the ith 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 case of three tasks and according to [14] and [17], the CLIK solution (5) is modified into

$$\boldsymbol{v}_d = \boldsymbol{v}_1 + \left(\boldsymbol{I} - \boldsymbol{J}_1^{\dagger}\boldsymbol{J}_1\right) \left[\boldsymbol{v}_2 + \left(\boldsymbol{I} - \boldsymbol{J}_2^{\dagger}\boldsymbol{J}_2\right) \boldsymbol{v}_3\right] \tag{7}$$

where $\boldsymbol{I}$ is the identity matrix of proper dimensions. Remarkably, (7) 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. Thus, the NSB control always fulfills the highest priority task at nonsingular configurations. The fulfillment of the lower priority tasks should be discussed in a case-by-case basis.

III. Task-Function Definitions

Once recalled the basic concepts concerning the NSB approach, it is necessary to understand which task functions (or elementary behaviors) can be defined and used to control an MRS. Thus, in the following, different task functions expressing global and local behaviors of the team are defined (more details on the single task functions can be found in [5] and [8]).

A. Centroid

The centroid of a platoon expresses the mean value of the vehicles' positions. In a 2-D case, the task function is expressed by

$$\boldsymbol{\sigma}_c = \boldsymbol{f}_c(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{p}_i$$

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

B. Variance

The task function for platoon 2-D variance $\boldsymbol{\sigma}_v \in \mathbb{R}^2$ is defined as

$$\boldsymbol{\sigma}_v = \frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix} (x_i - x_c)^2 \\ (y_i - y_c)^2 \end{bmatrix} \tag{8}$$

where x_c and y_c are the current centroid coordinates.

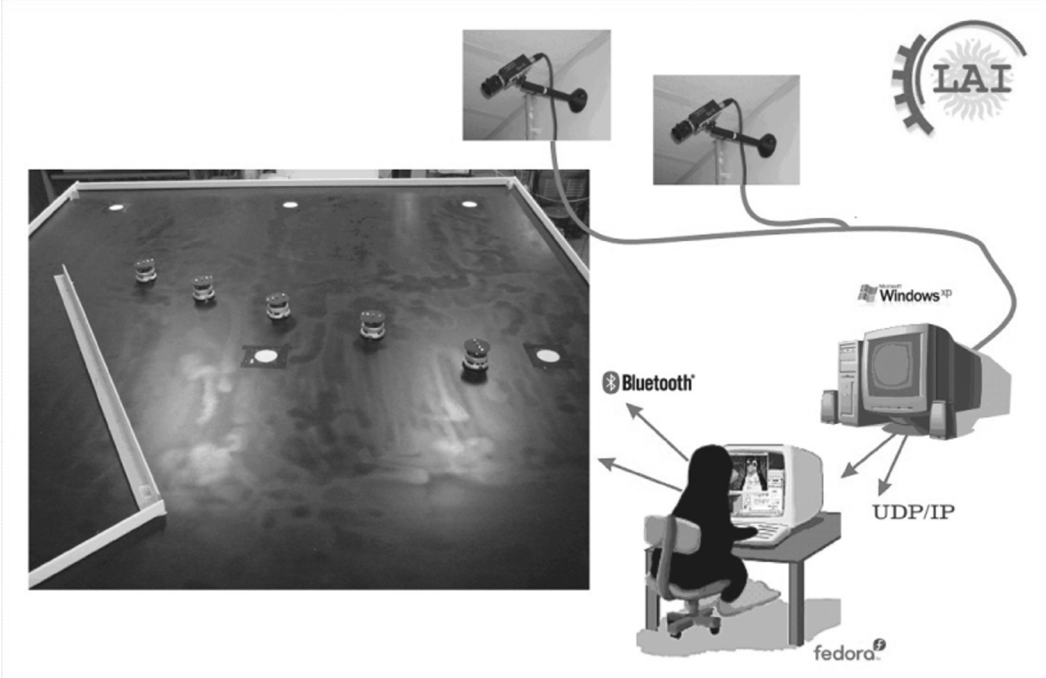

Fig. 1. Experimental setup available at the LAI, Università degli Studi di Cassino.

C. Rigid Formation

The rigid-formation task moves the vehicles to a predefined formation relative to the centroid. The task function $\boldsymbol{\sigma}_f \in \mathbb{R}^{2n}$ is defined as

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

where $\boldsymbol{p}_i$ are the coordinates of the vehicle i and $\boldsymbol{p}_c = \boldsymbol{\sigma}_c$ are the coordinates of the centroid.

D. Obstacle Avoidance

Obstacle avoidance is a crucial task to be followed by each vehicle of the team. The corresponding task function is activated only when the vehicle is approaching an obstacle, and it is designed so that the vehicle *slide* around the obstacle. It is

$$\sigma_o = ||\boldsymbol{p} - \boldsymbol{p}_o||$$

where $\boldsymbol{p}_o$ is the obstacle position and the symbol $||\cdot||$ represents the Euclidean norm.

IV. Experiments With a Team of Khepera II

The NSB approach has been tested in different experimental missions concerning the coordinated control of a team of grounded mobile robots. The objective of the proposed experiments is to coordinate the MRS by simultaneously controlling some of its global parameters like the centroid position, the robots spread in the environment, or the robot displacement. In particular, the performed missions are as follows.

1) To handle the spread of the robots in the environment by controlling their centroid and the variance around it; the robots have to avoid eventual inter-vehicle hitting (two experiments).
2) To impose to the robots a linear formation while avoiding eventual obstacles and inter-vehicle hitting (three experiments).
3) To impose to the robots a circular formation; to stress the algorithm, the vehicles are commanded to switch their position with the *opposite* robot in the circle; in addition, a moving obstacle is thrown into the robots cloud (one experiment).

It is assumed that the robots have limited ranging capabilities with respect to the presence of other robots or obstacles; the corresponding obstacle-avoidance task, thus, is activated only if the robot is close enough to another robot or obstacle.

A. Experimental Setup

The experimental setup available at the Laboratorio di Automazione Industriale (LAI) of the Università degli Studi di Cassino is briefly discussed in the following. The setup is based on a team of seven Khepera II (see Fig. 1), manufactured by K-team [1], that are differential-drive mobile robots with a unicyclelike kinematics of 8 cm of diameter. Each robot communicates via Bluetooth with a remote Linux-based PC, where a Bluetooth dongle, building virtual serial ports, allows the communication with up to seven robots. The remote PC implements the NSB algorithm.

To allow the needed absolute position measurements, a vision-based system using two charge-coupled device cameras, a Matrox Meteor-II frame grabber [2], and the self-developed C++ image-processing functions have been developed. The acquired images are 1024×768 RGB bitmaps. In particular, the upper turrets of each robot have a set of colored LEDs that are used to detect positions, orientations, and identification num-

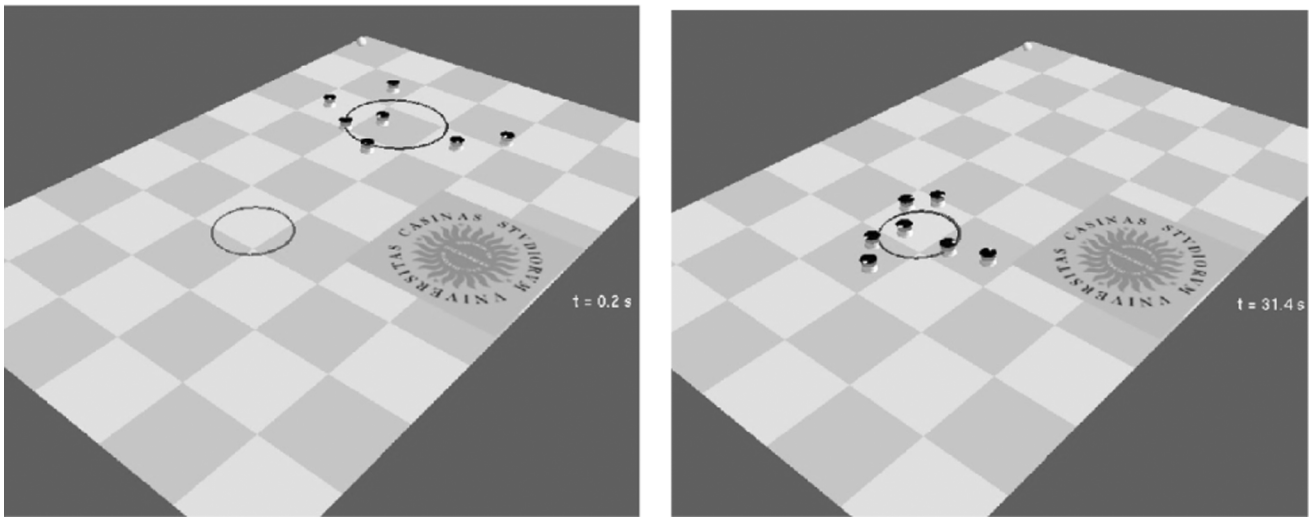

Fig. 2. Mission #1. Snapshots of the (left) initial and (right) final robots configuration. The circles represent the desired and measured centroid positions and variance. In the right plot the circles are superimposed.

bers of each robot. The position measurements are performed at a sampling time of 100 ms while the estimation error has an upper bound of ≈ 0.5 cm and $\approx 1^\circ$. Moreover, the vision system permits to identify static obstacles (e.g., the linear obstacles in Fig. 1) or dynamic obstacles (i.e., a tennis ball) eventually present in the environment. The measurements are sent over the LAN to the Linux-based PC using the UDP/IP protocol.

Following the approach described in the previous sections, the NSB elaborates the desired linear velocity for each robot of the team. Being the Khepera unicycle-like robots, a heading controller has been derived from the controller reported in [20] to obtain the wheels' desired velocities. Thus, the remote Linux-based PC sends to each vehicle (through the Bluetooth module) the wheels' desired velocities with a sampling time of 120 ms. The wheels' controller (onboard of each robot) is a PID developed by the manufacturer. A saturation of 40 cm/s and 180 °/s has been introduced for the linear and angular velocities, respectively. Moreover, the encoders' resolution is such that a quantization of ≈ 0.8 cm/s and $\approx 9^\circ/\text{s}$ are experienced.

The experimental results will be presented in the next sections resorting to a self-developed graphical simulator. The software, developed in C under Linux (and that uses the OpenGL libraries), is used for debugging purposes and as graphical output of the experimental data. To improve the experiments' readings, in the following, the snapshots always represent a graphical representation of the experimental data.

B. Mission 1: Obstacle–Centroid–Variance

The first kind of mission concerns the possibility to control the spread of the team of robots by the position of its centroid and the variance around it. As an example, in Fig. 2, a team of robots, starting from a random configuration, reaches a configuration whose centroid and variance are given. In particular, the circles are centered in the desired and current centroids with radii, respectively, equal to the square roots of the desired and current variances.

The global mission has been decomposed into three tasks, where the order of the tasks represents their order of priority:

1) obstacle avoidance;
2) centroid;
3) bidimensional variance.

The corresponding gains are

$$\lambda_o = 0.5 \quad \mathbf{\Lambda}_c = \boldsymbol{I}_2 \quad \mathbf{\Lambda}_v = \boldsymbol{I}_2$$

where the subscript of the identity matrix denotes its dimension.

In the first experiment, a platoon of six robots, starting from a random configuration, is commanded to move its centroid to a constant desired configuration $\boldsymbol{\sigma}_{c,d} = [80\ 170]^{\mathrm{T}}$ cm, keeping a desired variance of $\boldsymbol{\sigma}_{v,d} = [180\ 180]^{\mathrm{T}}$ cm^2. Fig. 3(a) shows the path followed by the robots (thin lines) and the path of their centroid (larger line); in gray, the starting positions, in black, the final ones. Fig. 3(b) shows the centroid task-function values during all the mission; the robots stop when the distance between the centroid position and its desired position is lower than a threshold value. Fig. 3(c) shows the variance values with respect to the centroid along the axes x and y.

In the second experiment, a platoon of seven robots is commanded to move its centroid to a constant desired configuration $\boldsymbol{\sigma}_{c,d} = [80\ 170]^{\mathrm{T}}$ cm; three different set points for the variance are consecutively assigned as $\boldsymbol{\sigma}_{v,d} = [300\ 300]^{\mathrm{T}}$,$\boldsymbol{\sigma}_{v,d} = [0\ 0]^{\mathrm{T}}$ and $\boldsymbol{\sigma}_{v,d} = [500\ 500]^{\mathrm{T}}$ cm^2. Fig. 4(a) shows several steps of the mission execution, while Fig. 4(b) and 4(c) shows, respectively, the centroid task-function values and the variance values along the axes x and y during all missions. It is worth noticing that the variance value relates to the density of the robots in the team; thus, a small variance value makes the robot stay close to the centroid and a high value makes the robot spread in the environment. Obviously, the value $\boldsymbol{\sigma}_{v,d} = [0\ 0]^{\mathrm{T}}$ cm^2 cannot be reached because the obstacle-avoidance task function, i.e., the highest priority task does not allow the vehicles to have a relative distance lower than 12 cm; this desired value was commanded to test the

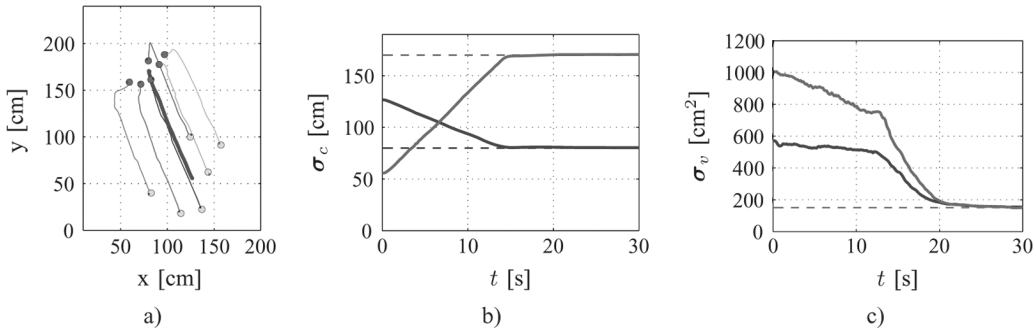

Fig. 3. Mission #1, first experiment. (a) Paths followed by the robots during the mission. (b) Average task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{c,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_c$. (c) Variance task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{v,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_v$.

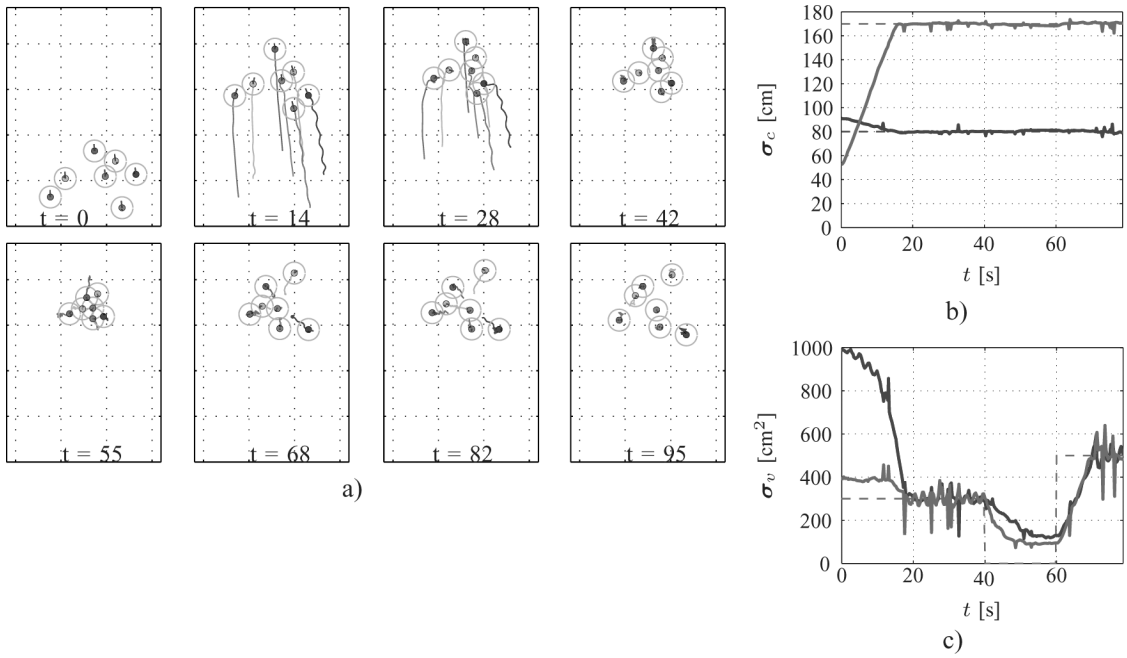

Fig. 4. Mission #1, second experiment. (a) Snapshots: The circles around the robots represent their safety region, the last seconds path is reported. (b) Centroid task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{c,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_c$. (c) Variance task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{v,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_v$.

algorithm in a demanding situation. The corresponding video is named `f_variance.mpg`.

C. Mission 2: Obstacle–Centroid Rigid Formation

In the second mission, the robots are commanded to reach a linear rigid formation around the centroid (see Fig. 5). Three experiments are reported: in the first one, the robots are commanded to assume a linear formation and then switch their relative positions; in the second experiment, the same mission is executed in the presence of two linear static obstacles; and in the last experiment, a moving obstacle (a tennis ball pushed by hand) is thrown over the moving team of robots.

1) Rigid Formation + Change of Formation: The first experiment of the second mission concerns a platoon of five robots starting from a random configuration. The mission consists of two steps: In the first step, the robots are commanded to assume the configuration with a centroid of $\boldsymbol{\sigma}_{c,d} = [60\ 70]^{\mathrm{T}}$ cm and to reach a linear configuration (rotated at $-25^{\circ}/\mathrm{s}$ with respect to the axis x), where each robot has a distance from the others of 30 cm; then, the second step of the mission consists of a position permutation, i.e., the robots are commanded to symmetrically invert their positions in the formation. Obviously, collisions among the robots need to be avoided during all missions.

The priority of the three tasks implemented is as follows:

1) obstacle avoidance;
2) centroid;
3) rigid formation.

The matrix gains are

$$\lambda_o = 0.5 \quad \boldsymbol{\Lambda}_c = \boldsymbol{I}_2 \quad \boldsymbol{\Lambda}_f = \boldsymbol{I}_{10}$$

with $\sigma_{o,d} = 16$cm.

Fig. 6 shows the experimental results; Fig. 6(a) shows some snapshots where the last seconds path and the safety region of the robots are highlighted. The first desired configuration is reached at $t \approx 24$ s. 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 centroid. It can be observed from the plot of the centroid task function [Fig. 6(b)] and

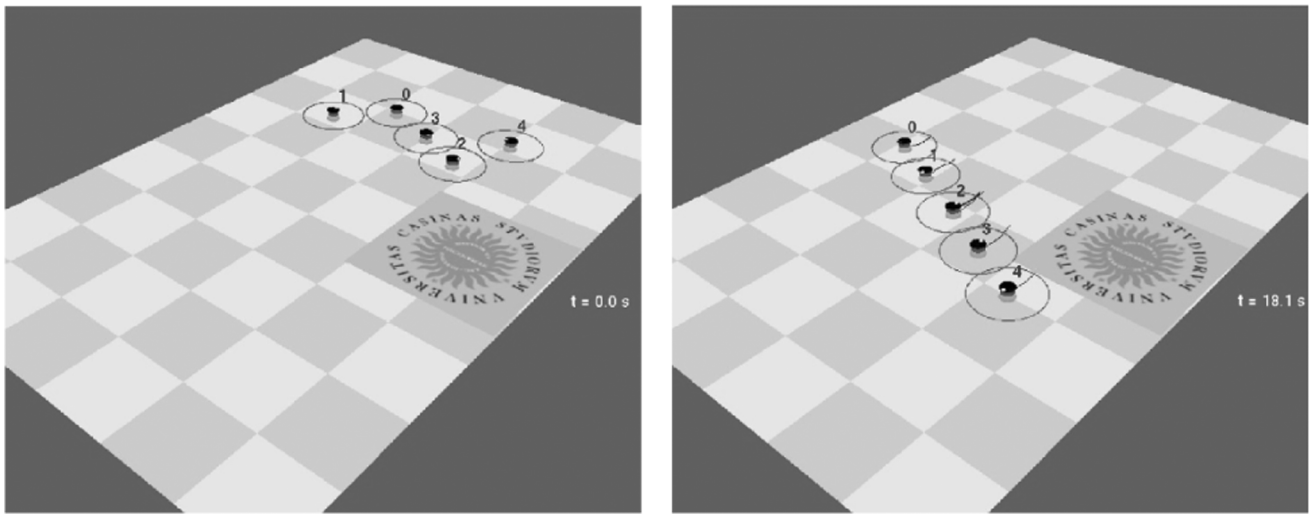

Fig. 5. Mission #2. Snapshots of the (left) initial and (right) final robots' configuration. The circles represent the safety regions for each robot.

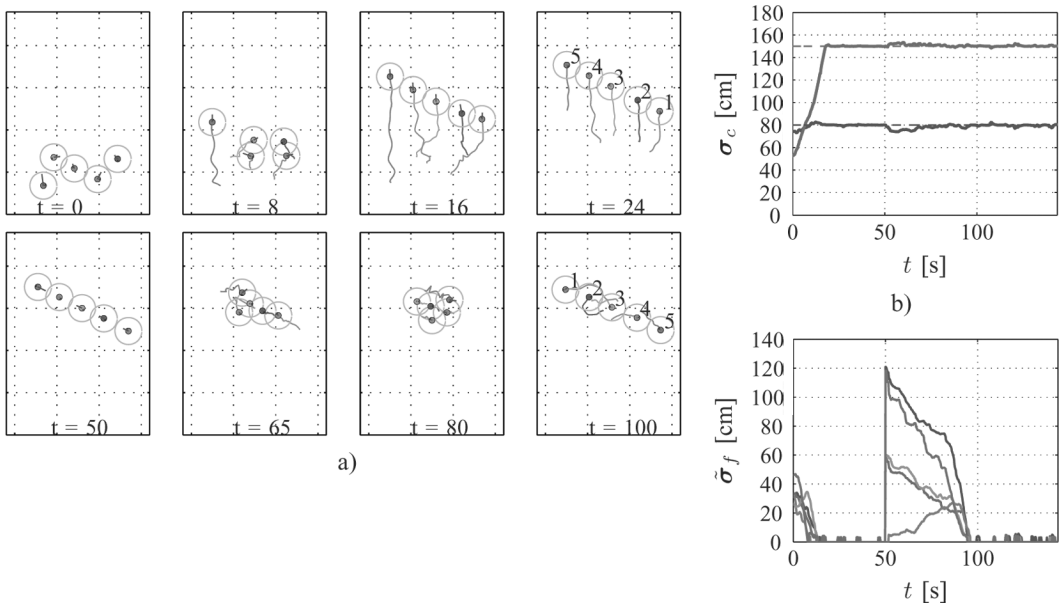

Fig. 6. Mission #2, first experiment. (a) Snapshots: The circles around the robots represent their safety region, the last seconds path is reported. (b) Centroid task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{c,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_c$. (c) Rigid-formation task-function errors: norms of the error components of $\tilde{\boldsymbol{\sigma}}_f$.

of the linear-formation task function [Fig. 6(c)] that the robots correctly reach the new formation avoiding collisions (primary task) and keeping low the centroid error (second priority task) also during the transient. The corresponding video can be found under the name `f_linear_formation.mpg`.

2) Rigid Formation + Linear Static Obstacles: In the second experiment of the second mission, the same rigid-formation mission is commanded but in the presence of two linear static obstacles. Fig. 7 shows several snapshots of the mission execution; the safety regions for the obstacles and the vehicles and the last seconds path are highlighted.

Fig. 8(b) shows the robots' paths during the whole execution of the experiment. The robots start from a random configuration [gray positions shown in Fig. 8(b)], and they reach the first configuration in $t \approx 20$ s, avoiding collisions among themselves and with the static obstacles. At $t = 50$ s, a new step input is commanded to the platoon by requiring a switch in the robots'

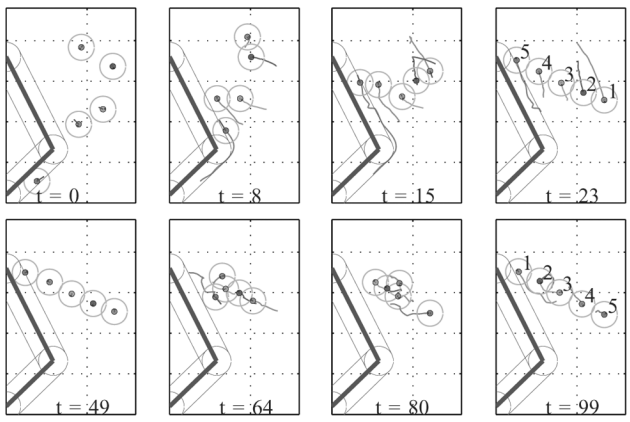

Fig. 7. Mission #2, second experiment. Snapshots: The last seconds path is reported. The thick solid line represent the static obstacle, while the thin line/circle represent the safety areas.

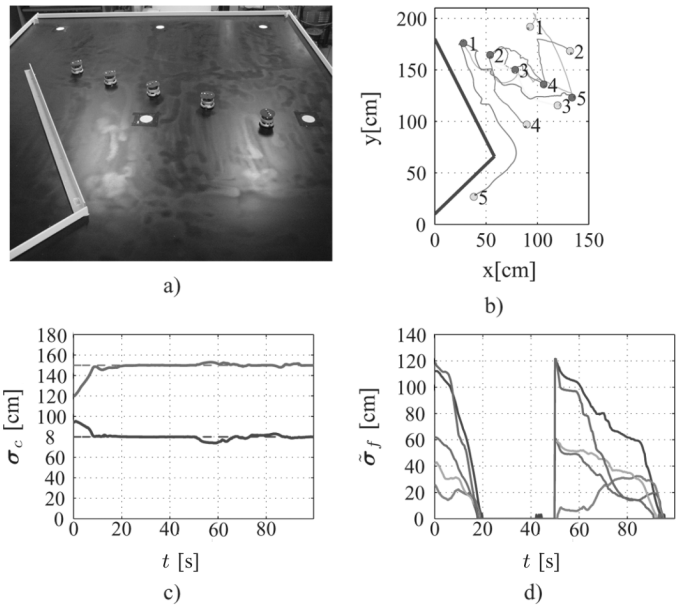

Fig. 8. Mission #2, second experiment. (a) Photograph of the steady-state configuration. (b) Paths followed by the Khepera II robots during the experimental mission. (gray) Initial positions. (black) Final positions. (c) Centroid task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{c,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_c$. (d) Rigid-formation task-function errors: Norms of the error components of $\tilde{\boldsymbol{\sigma}}_f$ for each vehicle.

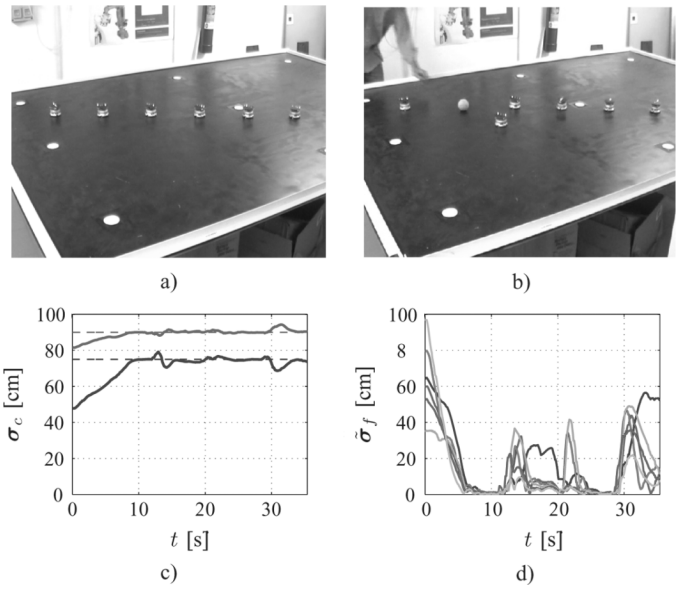

Fig. 9. Mission #2, third experiment. (a) and (b) Two frames of the experiment. (c) Average task function: (dashed lines) Desired values of $\boldsymbol{\sigma}_{c,d}$ and (solid lines) real values of $\boldsymbol{\sigma}_c$. (d) Rigid-formation task-function errors: Norms of the error components of $\tilde{\boldsymbol{\sigma}}_f$ for each vehicle.

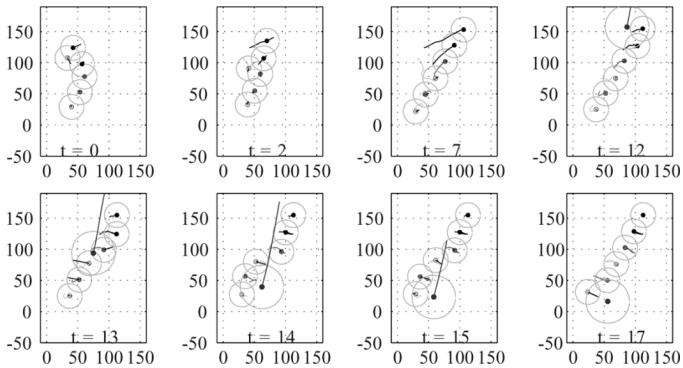

Fig. 10. Mission #2, third experiment. Several steps of the linear-formation mission while a tennis ball is passing through the formation.

positions, keeping the same centroid. The specific path of robot $n^{\circ}5$ is of interest. In the beginning of the mission, it correctly avoids the linear static obstacles by *sliding* around their safety areas; however, it can be observed from the second and third frames shown in Fig. 7 that it enters the safety area. This behavior is due to several reasons: the closeness of another vehicle, the vehicle dynamics, its nonholonomy, and the sampling time.

Fig. 8(c) and (d) shows the task-function values for the centroid and for the linear formation. The convergence to zero of the rigid-formation task-function error and the convergence to the desired values of the centroid task function do show that both tasks are successfully performed. It can be noticed that, where the centroid task is of lower priority with respect to the obstacle avoidance one and where the linear formation is of the lowest priority tasks, the errors do not decrease monotonically to zero when the vehicles need to avoid a hit.

3) Rigid Formation + Dynamic Obstacle: The third experiment of the second mission concern a platoon of six Khepera robots commanded to hold a given centroid with a given linear formation; a tennis ball is thrown over the robots time after time in order to test the control strategy in the presence of dynamic obstacles. The desired position of the platoon centroid is $\boldsymbol{\sigma}_{c,d} = [75\ 90]^{\mathrm{T}}$ cm; the linear formation is rotated at 65° with respect to the axis x, and the robots are commanded to keep a distance of 30 cm away from each other.

The priority of the three tasks implemented are as follows:

1) obstacle avoidance;
2) centroid;
3) linear rigid formation.

The matrix gains are

$$\lambda_o = 0.5 \quad \boldsymbol{\Lambda}_c = \boldsymbol{I}_2 \quad \boldsymbol{\Lambda}_f = 2\boldsymbol{I}_{12}$$

while the safety distance among the robots is 20 cm, and the safety distance from the obstacle is 35 cm.

Fig. 9 shows two frames of the experiment and the values for the centroid and the linear-formation task functions during the whole execution of the experiment. Moreover, Fig. 10 shows some snapshots where the last seconds paths and the safety region of the robots and of the dynamic obstacle are highlighted. The robots reach the desired formation and keep it until the obstacle enters their safety area. When the dynamic obstacle is going through the formation, the robots have to avoid the obstacle to preserve their integrity; thus, they temporarily leave the desired formation and centroid position. Once the obstacle has overtaken the formation, the robots do reach again the desired configuration, avoiding collision among themselves. It is worth noticing that, in the last step of Fig. 10, one of the robot is still close to the obstacle, and it does stay out of the formation; however, the centroid is at the desired value. This behavior is due to the priority order of the tasks. In fact, at first, the robots have

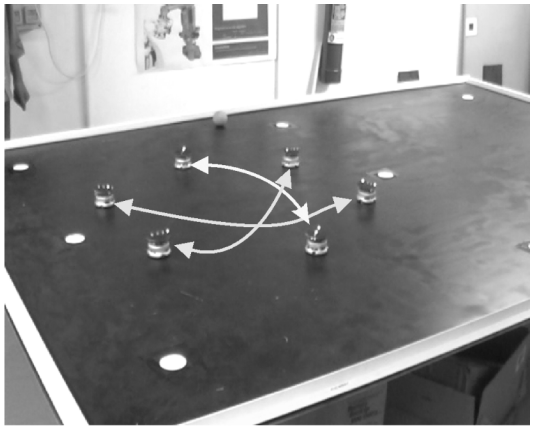

Fig. 11. Mission #3. Circular Permuting Formation.

to avoid collisions, then, in the null space of the obstacle-avoidance task, they have to keep the centroid at the desired value, and finally, as a tertiary task, they have to reach the formation with respect to the centroid. The conflict-resolution policy applied by the NSB permits to guarantee the achievement of the lower priority tasks only if they do not conflict with the higher ones; thus, in the specific configuration, the three task are conflicting, and the last one (rigid formation) can not be achieved. However, after moving the ball away from the last robot, the three tasks do not conflict anymore, and the formation can be reached again. In Fig. 10, only few significant steps of the mission are presented; however, the corresponding video with the complete mission is the file `f_linear.mpg`.

D. Mission 3: Permuting Circular Formation With Dynamic Obstacle

In the third mission, a team of six robots has to dynamically reach circular formations while the relative positions of the robots permute, i.e., once reached one desired configuration, each robot of the team has to exchange its position with the symmetrical one with respect to the center of the circle (see Fig. 11). This mission permits to test the NSB in case of high-*traffic* conditions; in [21], a solution to handle traffic configurations was proposed that is specifically designed for aerial vehicles.

In addition, in this case, the priority of the three tasks implemented are as follows:

1) obstacle avoidance;
2) centroid;
3) rigid formation.

While the gain matrix of the task functions are all identity matrices.

Fig. 12(a) shows several steps of the performed mission. Starting from a linear configuration, the robots reach the circular formation in less the 20 s. Then, once the error of the rigid-formation task function has gone under a threshold value, the robots have to change their relative positions. In particular, each robot has to exchange its position with its symmetrical with respect to the center of the circle [as can be noticed when observing colors and numbers of the robots in the steps 4, 8, and 12 of Fig. 12(a)]. During the change of formation, all the robots converge toward the center of the circle, increasing the risk of collisions and of the incurrence of singular configurations. To significantly stress the algorithm, as shown by the last four steps of Fig. 12(a), a tennis ball passes through the circle while the robot are changing the formation. In addition, in this case, the change is correctly achieved, avoiding collisions among the robots and with the obstacle. The correct achievement of the mission allows us to consider the approach robust to high-traffic condition and to conflict resolution.

Fig. 12(b) and (c) shows, respectively, the errors of the centroid task function and of the rigid-formation task function. It is worth noticing that the centroid error is small during all the missions. However, it is not null because of the nonholonomy of the robots and because the robots primarily have to avoid hitting the eventual obstacle. In an ideal case of omnidirectional robots and ignoring the collision avoidance, the robot should keep the centroid in a constant position, performing all the motions for changing the configuration in the null space of the centroid task. Fig. 12(d) shows that the changes of formations are given as step functions and are correctly achieved also in presence of obstacles. In particular, it is possible to notice that the third change of formation takes longer and is more irregular than the previous ones because the ball is passing through the circle. The corresponding video is the file `f_circular.mpg`.

E. Experimental Outcomes

In [7] and [8], the authors proposed an algorithm for multirobot coordination. The approach turned out to be quite general, and it has permitted to arrange different missions by resorting to very simple tasks functions. The theoretical investigation of the algorithm is still object to research in terms of, e.g., the stability properties or its behavioral interpretation [5], while this paper represents its laboratory experimental validation. A first validation has been achieved by resorting to extensive numerical simulations with nonholonomic vehicles; successively, the implementation on a physical system has been done to lose these ideal conditions (like massless holonomic vehicles, instantaneous communication, absence of noise) and test the algorithm in a realistic scenario. The previously described experiments, in fact, run under several nonideal conditions.

1) Both the vision system and the remote PC run on nonreal-time operative systems, respectively, Fedora 3 and Windows XP at the time of the experiments. The code runs in a soft-real-time version where only the average sampling time can be imposed; several samples long much more than the imposed value of up to 300% the nominal value.
2) The vision system experiences samples where it *loses* the obstacle–robot positions. Although the vision-extraction algorithm is sophisticated, it happens that the vision PC sends null values for some of the robots or the obstacles.
3) The UDP/IP communication layer between vision and the remote PC is not deterministic. The communication flying time experiences the so-called random-sampling phenomenon that, in the better case, is adding white noise to the data.
4) The remote PC and the robots are linked via a Bluetooth communication layer experiencing several troubles. In few cases, one robot had lost the communication for 2–3 s, keeping in memory its last command. When finally recovered the communication, it was able to coordinate again

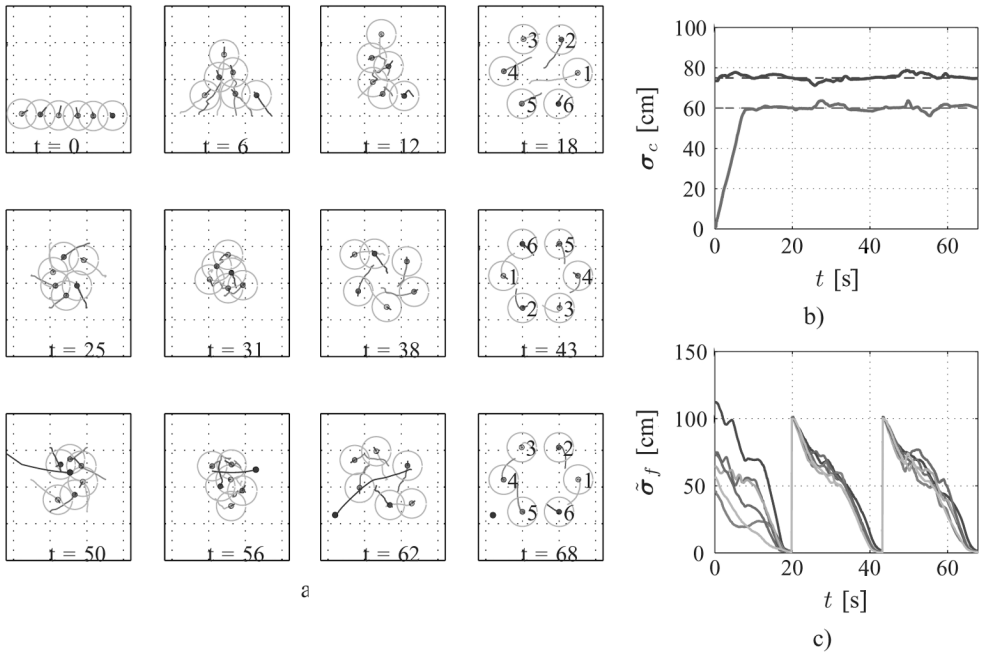

Fig. 12. Mission #3. (a) Multiple steps of the circular-permuting-formation mission while a tennis balls is passing through the circle. (b) Error of the centroid task function during the circular-permuting-formation experiment. (c) Error of the rigid-formation task function during the circular-permuting-formation experiment.

its movements with the remaining robots. Several experiments failed due to a persistent impossibility of the Bluetooth to communicate, in such a case, emergency routines stopped the motion.

5) The presence of nonholonomy and the dynamics imposed the introduction of thresholds to avoid limit cycles around the null tasks' errors. The exact value of the thresholds have been tuned online.
6) The obstacle-avoidance policy is based on a *local* task function, i.e., based on local information. The corresponding task correctly handles the relatively simple approached case studies, i.e., convex obstacles. In a more general case, such as indoorlike environments, a more complex obstacle handling is needed to avoid local minima. This is, however, a characteristic of all the obstacle-avoidance strategies that only use local information.
7) The proposed strategy does not need a large number of parameters to be tuned. A first gross tuning is performed in simulation, resorting to practical considerations, while a second fine tuning is performed on the experimental setup. Over different experiments, the same task has been performed with the same gain. Different tasks generally require different gain values, since they represent different mappings.

Given the constraints above, the success of the experimental implementation was not obvious, and the overall *robustness* of the NSB approach needed to be proven. Further insights could be obtained using proprioceptive sensors to estimate the robot positions instead of a vision system and using exteroceptive sensors to estimate the obstacle positions.

The proposed strategy has been designed and implemented in a centralized architecture. Current research activity is working toward the decentralization objective following two main directions: decentralize the tasks function, avoiding functions, e.g., the centroid, that requires the global information and decentralize via consensus, i.e., trying to elaborate global task function like the centroid via distributed computation algorithms.

V. Conclusion

In this paper, experimental results concerning the implementation of the NSB approach to control a team of mobile robots were presented. The NSB approach allows to properly handle the outputs of several, eventually conflicting, behaviors/tasks; in fact, the overall mission is decomposed into properly defined elementary tasks that are hierarchically arranged so that the higher priority tasks are not influenced by the lower priority ones. The experiments were performed at the LAI, Università degli Studi di Cassino, equipped with seven Khepera II mobile robots. The algorithm resulted in a successful implementation for several missions requiring the movement in a quite cluttered environment. The results prove the effectiveness and the flexibility of the approach in the centralized architecture. Future research will concern the possibility to decentralize the NSB approach by making each robot use only local functions or elaborate global functions, such as the centroid position, through distributed algorithms.

References

[1] K-Team. [Online]. Available: http://www.k-team.com
[2] *"Matrox Imaging Library User Guide,"* Matrox Electron. Syst. Ltd., Dorval, QC, Canada, 2001.
[3] G. Antonelli, F. Arrichiello, S. Chakraborti, and S. Chiaverini, "Experiences of formation control of multi-robot systems with the null-space-based behavioral control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 1068–1073.
[4] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Experiments of formation control with collisions avoidance using the null-space-based behavioral control," presented at the 14th Mediterranean Conf. Control Autom., Ancona, Italy, Jun. 2006.

[5] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *J. Intell. Service Robot.*, vol. 1, no. 1, pp. 27–39, Mar. 2007, Jan. 2008.
[6] G. Antonelli, F. Arrichiello, S. Chiaverini, and K. J. Rao, "Preliminary experiments of formation control using the null-space-based behavioral control," presented at the 8th IFAC Symp. Robot Control, Bologna, Italy, Sep. 2006.
[7] G. Antonelli and S. Chiaverini, "Kinematic control of a platoon of autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2003, pp. 1464–1469.
[8] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1285–1292, Dec. 2006.
[9] R. C. Arkin, "Motor schema based mobile robot navigation," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 92–112, 1989.
[10] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.
[11] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.
[12] C. Belta and V. K. Kumar, "Abstraction and control of groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, Oct. 2004.
[13] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, Mar. 1986.
[14] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.
[15] J. P. Desai, V. Kumar, and J. P. Ostrowski, "Control of changes in formation for a team of mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, May 1999, pp. 1556–1561.
[16] C. R. Kube and H. Zhang, "Collective robotics: From social insects to robots," *Adapt. Behav.*, vol. 2, no. 2, pp. 189–218, 1993.
[17] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Trans. Robot. Autom.*, vol. 23, no. 1, pp. 60–72, Feb. 2007.
[18] J. A. Marshall, T. Fung, M. E. Broucke, G. M. T. D'Eleuterio, and B. A. Francis, "Experiments in multirobot coordination," *Robot. Auton. Syst.*, vol. 54, no. 3, pp. 265–275, Mar. 2006.
[19] M. J. Mataric, "Designing emergent behaviors: From local interaction to collective intelligence," in *Proc. Int. Conf. Simul. Adapt. Beh.: From Animal Animal*, 1992, pp. 432–441.
[20] G. Oriolo, A. De Lucam, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, Nov. 2002.
[21] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 3–11, Mar. 2002.
[22] L. E. Parker, "Designing control laws for cooperative agent teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, Atlanta, GA, May 1993, vol. 3, pp. 582–587.
[23] L. E. Parker, "On the design of behavior-based multi-robot teams," *Adv. Robot.*, vol. 10, no. 6, pp. 547–578, 1996.
[24] S. T. Pledgie, Y. Hao, A. M. Ferreira, S. K. Agrawal, and R. Murphey, "Groups of unmanned vehicles: Differential flatness, trajectory planning, and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, May 2002, pp. 3461–3466.
[25] C. Reynolds, "Flocks, herd and schools: A distributed behavioral model," *Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.
[26] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.*, vol. 3, no. 3, pp. 201–212, Sep. 1990.
[27] P. Tabuada, G. J. Pappas, and P. Lima, "Motion feasibility of multi-agent formations," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 387–392, Jun. 2005.
[28] L. Vig and J. A. Adams, "Multi-robot coalition formation," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 637–649, Aug. 2006.
[29] P. K. C. Wang, "Navigation strategies for multiple autonomous robots moving in formation," *J. Robot. Syst.*, vol. 8, no. 2, pp. 177–195, Apr. 1991.